Documentation 8.0

ZABBIX

02.04.2026

Contents

Manuel utilisateur **6**

Bienvenue dans la documentation Zabbix **6**

Premiers pas avec Zabbix . . . 6
Zabbix Cloud . . . 6
Centre de développement . . . 6
Communauté et autres ressources . . . 6
1 Installation et premiers pas . . . 6
1 Installation de l'interface web . . . 7
2 Guides de démarrage rapide . . . 13
3 Installation : guides supplémentaires . . . 33
4 Mise à niveau . . . 73
5 Notes de mise à niveau pour Zabbix 8.0 . . . 88
Problèmes connus . . . 90
Prérequis . . . 98
2 Nouveautés de Zabbix 8.0 . . . 108
Modèles . . . 108
Éléments . . . 109
Plugins . . . 109
Découverte de bas niveau . . . 110
Processus . . . 110
Authentification . . . 111
Widgets . . . 111
Interface web . . . 112
Documentation . . . 113
Nouveautés de Zabbix 8.0.x . . . 113
3 Appliance Zabbix . . . 113
4 Processus Zabbix . . . 117
1 Serveur . . . 117
2 Agent . . . 126
3 Agent 2 . . . 130
4 Proxy . . . 139
5 Passerelle Java . . . 144
6 Sender . . . 148
7 Get . . . 149
8 JS . . . 149
9 Web service . . . 150
5 Configuration . . . 150
1 Hôtes et groupes d'hôtes . . . 161
2 Éléments . . . 180
3 Détection des problèmes avec les déclencheurs . . . 397
4 Événements . . . 448
5 Corrélation d'événements . . . 452
6 Balisage . . . 459
7 Visualisation . . . 462
8 Modèles et groupes de modèles . . . 491
9 Modèles prêts à l'emploi . . . 491
10 Notifications sur les événements . . . 504
11 Macros . . . 555
12 Utilisateurs et groupes d'utilisateurs . . . 570
13 Stockage des secrets . . . 579

14 Rapports planifiés . . . 587
15 Exportation des données . . . 590
6 Surveillance des services . . . 596
1 Arborescence des services . . . 597
2 SLA . . . 601
3 Exemple de configuration . . . 602
7 Surveillance Web . . . 607
1 Éléments de supervision Web . . . 617
2 Scénario réel . . . 618
8 Supervision des machines virtuelles . . . 625
1 Clés d'élément de supervision VMware . . . 627
2 Champs clés de la découverte des machines virtuelles . . . 645
3 exemples JSON pour les éléments VMware . . . 650
4 Exemple de configuration de la supervision VMware . . . 654
9 Maintenance . . . 659
10 Expressions régulières . . . 664
11 Acquittement de problème . . . 670
12 Export/import de configuration . . . 673
1 Groupes de modèles . . . 675
2 Groupes d'hôtes . . . 675
3 Modèles . . . 675
4 Hôtes . . . 692
5 Cartes du réseau . . . 709
6 Types de média . . . 717
13 Découverte . . . 724
1 Découverte du réseau . . . 724
2 Enregistrement automatique d'agent actif . . . 733
3 Découverte de bas niveau . . . 737
14 Supervision distribuée . . . 794
1 Proxys . . . 794
15 Chiffrement . . . 803
1 Par certificat . . . 810
2 Par clés pré-partagées (PSK) . . . 817
3 Dépannage . . . 819
16 Interface web . . . 822
1 Menu . . . 823
2 Sections de l'interface . . . 832
3 Paramètres utilisateur . . . 1092
4 Recherche globale . . . 1097
5 Mode maintenance de l'interface Web . . . 1099
6 Paramètres de la page . . . 1100
7 Définitions . . . 1103
8 Créer votre propre thème . . . 1103
9 Mode Debug . . . 1104
10 Cookies utilisés par Zabbix . . . 1105
11 Fuseaux horaires . . . 1106
12 Réinitialisation du mot de passe . . . 1107
13 Sélecteurs de période et d'hôte . . . 1107
17 Bonnes pratiques . . . 1109
1 Bonnes pratiques de sécurité . . . 1109
2 Bonnes pratiques de configuration . . . 1117
18 API . . . 1119
Référence des méthodes . . . 1124
Annexe 1. Commentaire de référence . . . 1953
Annexe 2. Modifications de la version 7.4 à la version 8.0 . . . 1959
Annexe 3. Modifications dans la version 8.0 . . . 1960
19 Extensions . . . 1960
1 Modules chargeables . . . 1963
3 Modules du frontend . . . 1971
20 Annexes . . . 1972
1 Installation et configuration . . . 1973
2 Configuration des processus . . . 2033
3 Protocoles . . . 2122

4 Éléments . . . 2148
5 Fonctions prises en charge . . . 2186
6 Macros . . . 2186
7 Symboles des unités . . . 2224
8 Syntaxe des périodes de temps . . . 2226
9 Exécution de commande . . . 2227
10 Compatibilité des versions . . . 2228
12 Bibliothèque de liens dynamiques Zabbix sender pour Windows . . . 2229
13 Mise à niveau de la surveillance des services . . . 2230
14 Autres problèmes . . . 2230
16 Exemples d'échappement . . . 2231
21 Guides de référence rapide . . . 2233
Aperçu . . . 2233
1 Surveiller Linux avec l'agent Zabbix . . . 2233
2 Surveiller Windows avec l'agent Zabbix . . . 2237
3 Surveiller Apache via HTTP . . . 2240
4 Surveiller MySQL avec l'agent Zabbix 2 . . . 2245
5 Surveiller VMware avec Zabbix . . . 2252
6 Surveiller le trafic réseau avec Zabbix . . . 2256
7 Surveiller le trafic réseau à l'aide de contrôles actifs . . . 2259
8 Surveiller des sites web avec des éléments Browser . . . 2262
9 Surveiller les certificats de site web avec l'agent Zabbix 2 (passif) . . . 2267
10 Surveiller un commutateur réseau ou un routeur avec Zabbix . . . 2271
11 Surveiller le journal des événements Windows à l'aide de contrôles actifs . . . 2279
Qu'est-ce que Zabbix . . . 2282

Zabbix Cloud **2287**

Zabbix Cloud **2287**
Configurer et gérer Zabbix Cloud . . . 2287
Découvrez Zabbix Cloud . . . 2287
1 Déployer Zabbix dans le cloud . . . 2287
2 Configuration du nœud . . . 2291
Aperçu . . . 2291
Onglet Vue d'ensemble . . . 2292
Onglet Filtres d'accès . . . 2293
Onglet Chiffrement . . . 2294
Onglet Sauvegardes . . . 2295
Onglet Historique . . . 2295
Onglet Maintenance . . . 2296
Onglet Mise à niveau . . . 2297
3 Ajout d'utilisateurs . . . 2299
Introduction . . . 2299
Ajout d'utilisateurs aux organisations . . . 2299
Ajouter des utilisateurs aux nœuds . . . 2300
4 Principales différences de Zabbix Cloud . . . 2301

Centre de développement **2303**

Centre de développement **2303**
Développer sur Zabbix . . . 2303
Plugins . . . 2303
Interfaces de plugins . . . 2309
Bibliothèque Python pour Zabbix . . . 2312
Installation . . . 2312
Guide de démarrage rapide . . . 2312
Utiliser l'API Zabbix . . . 2316
Collecter des données depuis l'agent Zabbix . . . 2319
Envoyer des données au serveur ou au proxy Zabbix . . . 2321
Journalisation de débogage . . . 2324
Modules . . . 2325
Structure des fichiers des modules . . . 2325
Widgets . . . 2333
Tutoriels . . . 2341

Exemples . . . 2367
Modifications du développement des extensions . . . 2368

Pages de manuel Zabbix **2368**
zabbix_agent2 . . . 2368
NOM . . . 2368
RÉSUMÉ . . . 2368
DESCRIPTION . . . 2368
OPTIONS . . . 2368
FICHIERS . . . 2369
VOIR ÉGALEMENT . . . 2369
Index . . . 2369
zabbix_agentd . . . 2370
NOM . . . 2370
RÉSUMÉ . . . 2370
DESCRIPTION . . . 2370
OPTIONS . . . 2370
FICHIERS . . . 2371
VOIR EGALEMENT . . . 2371
Index . . . 2372
zabbix_get . . . 2372
NOM . . . 2372
SYNOPSIS . . . 2372
DESCRIPTION . . . 2372
OPTIONS . . . 2372
EXEMPLES . . . 2373
VOIR ÉGALEMENT . . . 2374
Index . . . 2374
zabbix_js . . . 2374
NOM . . . 2374
SYNOPSIS . . . 2374
DESCRIPTION . . . 2374
OPTIONS . . . 2374
EXEMPLES . . . 2375
VOIR ÉGALEMENT . . . 2375
Indice . . . 2375
zabbix_proxy . . . 2375
NOM . . . 2375
RÉSUMÉ . . . 2375
DESCRIPTION . . . 2376
OPTIONS . . . 2376
FICHIERS . . . 2377
VOIR EGALEMENT . . . 2377
Index . . . 2377
zabbix_sender . . . 2377
NOM . . . 2377
SYNOPSIS . . . 2377
DESCRIPTION . . . 2378
OPTIONS . . . 2378
ETAT DE SORTIE . . . 2380
EXEMPLES . . . 2380
VOIR EGALEMENT . . . 2381
Index . . . 2381
zabbix_server . . . 2381
NOM . . . 2382
RÉSUMÉ . . . 2382
DESCRIPTION . . . 2382
OPTIONS . . . 2382
FICHIERS . . . 2384
VOIR EGALEMENT . . . 2384
Index . . . 2384
zabbix_web_service . . . 2384
NOM . . . 2384

RÉSUMÉ 2384
DESCRIPTION 2384
OPTIONS 2385
FICHIERS 2385
VOIR EGALEMENT 2385
Indice 2385

Mention de copyright **2385**

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 concis, orientés tâches, qui vous accompagnent à travers les bases — de vos premières étapes de configuration 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.

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.

1 Installation et premiers pas

Pour configurer Zabbix à partir de zéro, suivez ce processus :

1. Assurez-vous que les logiciels requis sont installés et en cours d'exécution.
2. Installez Zabbix.
3. Installez l'interface web de Zabbix.
4. Commencez la supervision.

Plus de détails sur chaque étape suivent.

Logiciels requis Zabbix fonctionne bien avec les logiciels suivants (souvent appelés la 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 peut déjà être installé. Dans ce cas, il se peut que vous n'ayez qu'à configurer le serveur de base de données et le serveur web.

Pour des informations plus détaillées, telles que les versions de logiciels prises en charge, consultez Logiciels requis.

Installer Zabbix La manière la **plus simple** d'installer Zabbix est d'utiliser les **paquets de distribution**. 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 d'installer pour commencer un serveur Zabbix, un agent Zabbix 2 et le frontend (interface web). Vous pourrez ajouter des proxys Zabbix et davantage d'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 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 le frontend 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 rapides expliquent avec quels modèles vous devez travailler pour commencer à superviser des ressources courantes.

Vous pouvez commencer par le guide de démarrage rapide pour superviser un serveur Linux.

1 Installation de l'interface web

Cette section fournit des instructions étape par étape pour installer l'interface web de Zabbix.
Suivez ces étapes après avoir installé le backend de Zabbix — serveur Zabbix, agent et base de données.

Le frontend de Zabbix est écrit en PHP ; un serveur web prenant en charge PHP est donc nécessaire pour l'exécuter.

> **Note:**
> Vous pouvez en savoir plus sur la configuration de SSL pour l'interface web Zabbix en consultant ces bonnes pratiques.

Écran de bienvenue

Ouvrez l'URL du frontend Zabbix dans le navigateur. Si vous avez installé Zabbix à partir de paquets, l'URL est :

- pour Apache : *http://<server_ip_or_name>/zabbix*
- pour Nginx : *http://<server_ip_or_name>*

Vous devriez voir le premier écran de l'assistant d'installation du frontend.

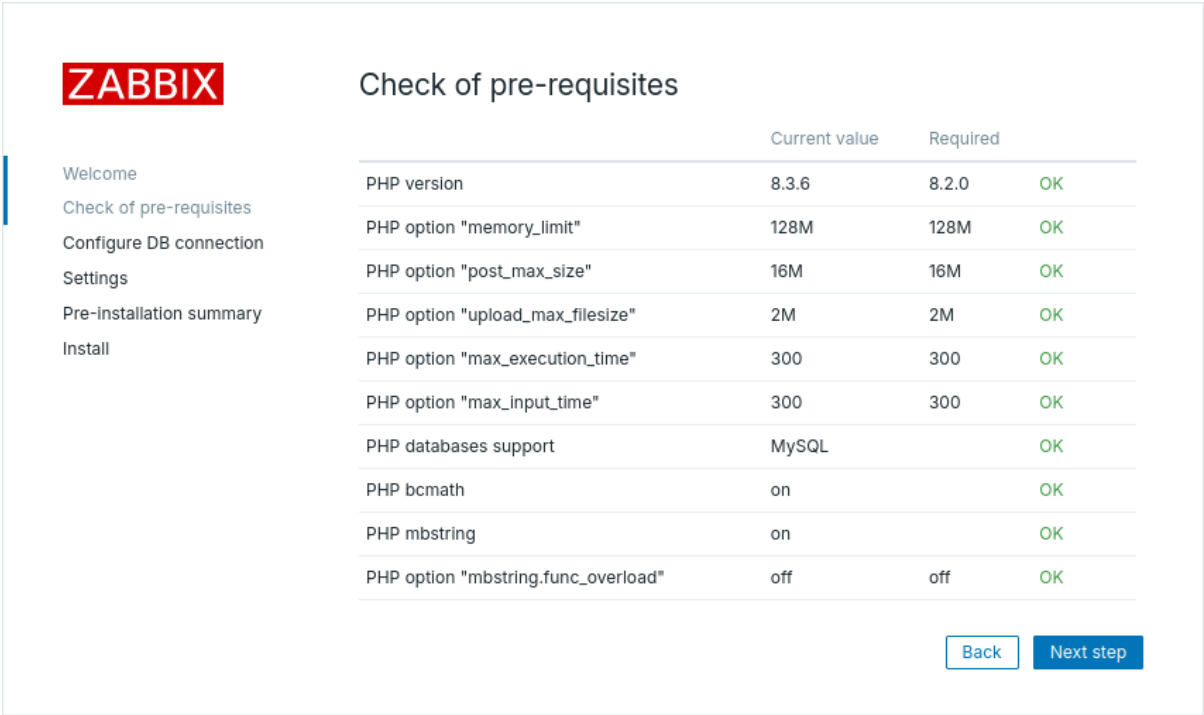
Utilisez le menu déroulant *Default language* pour modifier la langue par défaut du système et poursuivre le processus d'installation dans la langue sélectionnée (facultatif). Pour plus d'informations, consultez Installation de langues supplémentaires du frontend.

Notez que le fait de définir la langue sur *English (en_US)* activera également le format de date/heure américain dans le frontend.



Vérification des prérequis

Assurez-vous que tous les prérequis obligatoires du frontend Zabbix sont remplis.



Prérequis	Valeur minimale	Description
Version de PHP	8.2.0	
Option PHP "memory_limit"	128MB	Dans php.ini : memory_limit = 128M
Option PHP "post_max_size"	16MB	Dans php.ini : post_max_size = 16M
Option PHP "upload_max_filesize"	2MB	Dans php.ini : upload_max_filesize = 2M
Option PHP "max_execution_time"	300 secondes	Dans php.ini : max_execution_time = 300 (les valeurs 0 et -1 sont également autorisées)
Option PHP "max_input_time"	300 secondes	Dans php.ini : max_input_time = 300 (les valeurs 0 et -1 sont également autorisées)

Prérequis	Valeur minimale	Description
Prise en charge des bases de données PHP	Une des suivantes : MySQL, PostgreSQL	Voir Requirements pour la liste de toutes les extensions PHP obligatoires et facultatives. Notez que les prérequis facultatifs non satisfaits sont affichés avec un statut *Warning* rouge, et le processus d'installation peut se poursuivre même s'ils ne sont pas remplis.
PHP bcmath	doit être activé	
PHP mbstring	doit être activé	
Option PHP "mbstring.func_overload"	doit être désactivée	Dans php.ini : mbstring.func_overload = 0
Option PHP "session.auto_start"	doit être désactivée	Dans php.ini : session.auto_start = 0
Option PHP "arg_separator.output"	&	Dans php.ini : arg_separator.output = "&" (la valeur "&" est également autorisée)

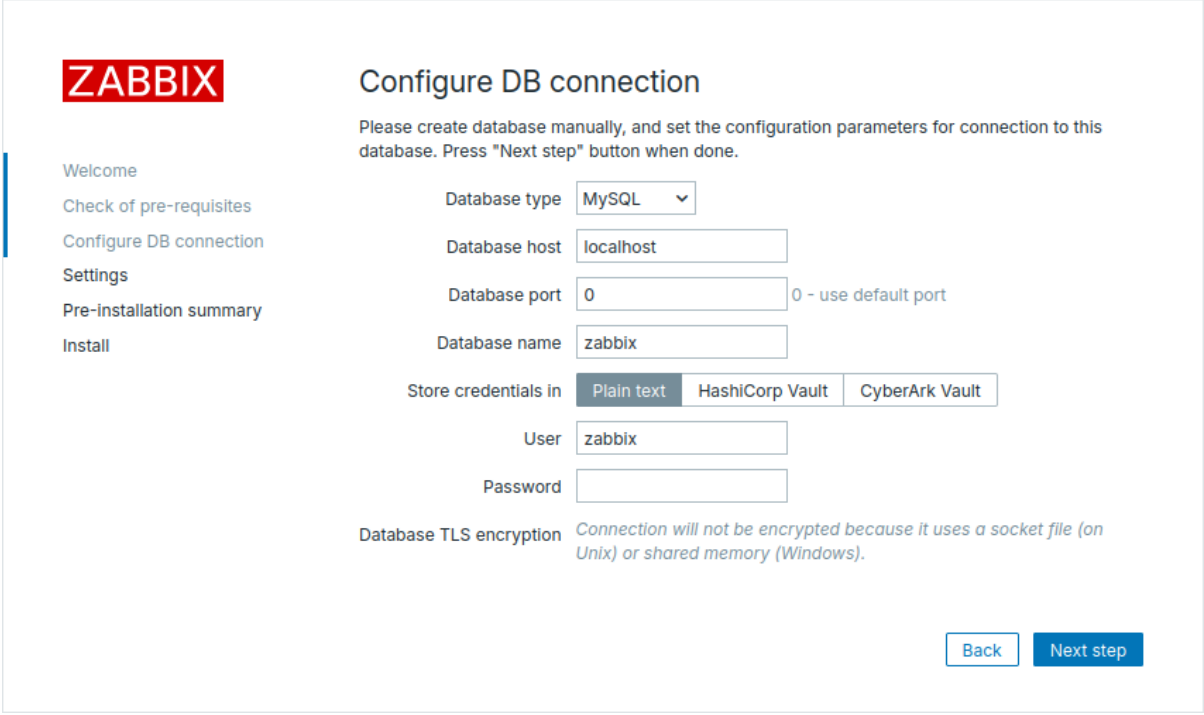
Attention:
Si l'utilisateur Apache ou le groupe d'utilisateurs doit être modifié, vérifiez les autorisations du dossier de session ; sinon, l'installation de Zabbix risque de ne pas pouvoir se poursuivre.

Configurer la connexion à la base de données

Saisissez les informations nécessaires pour vous connecter à la base de données. La base de données Zabbix doit déjà être créée.

Pour MySQL, saisir `localhost` ou laisser le champ *Hôte de la base de données* vide entraîne une connexion via le socket Unix par défaut. Le formulaire d'installation ne propose pas de champ distinct *Socket de la base de données* ; ainsi, pour utiliser un socket personnalisé, configurez-le dans les paramètres du serveur Zabbix (par exemple, avec `DBSocket=` dans zabbix_server.conf). Cela permet de maintenir l'interface web alignée sur les paramètres de connexion entre le serveur et la base de données.

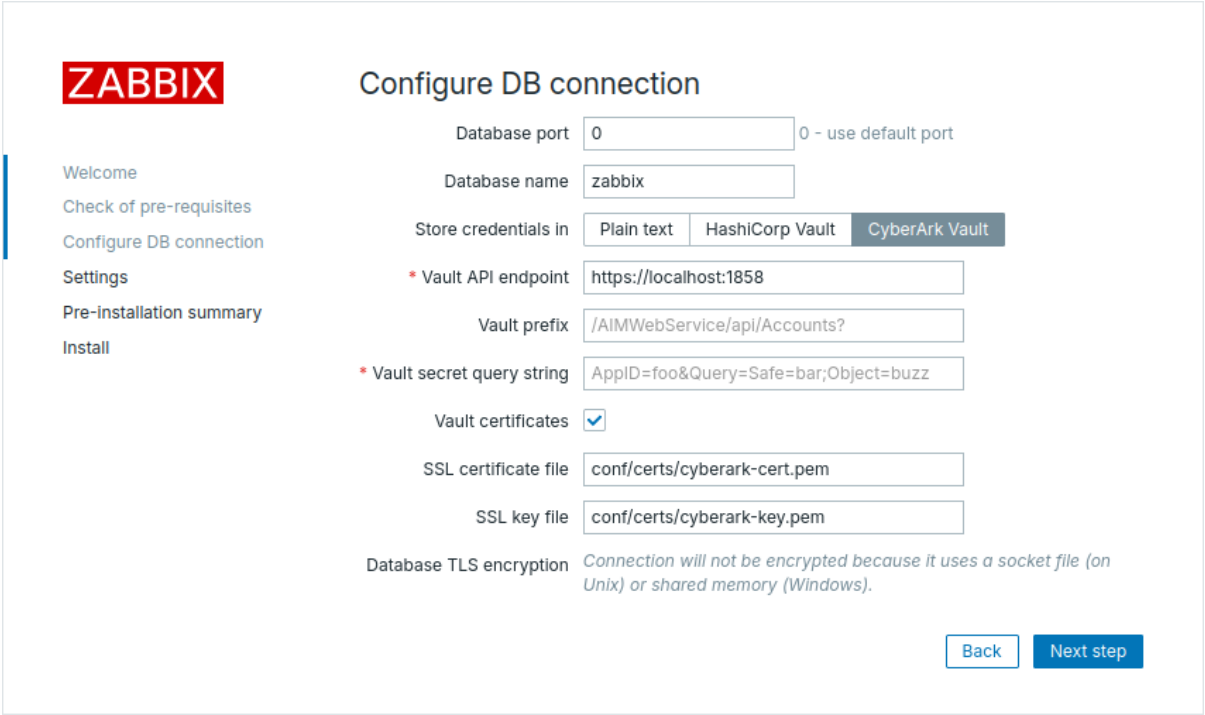
Pour PostgreSQL, si le champ *Hôte de la base de données* est laissé vide, le socket de domaine Unix par défaut est utilisé. Si un chemin de socket est saisi (par exemple, `/var/run/pgbouncer`), ce socket de domaine Unix sera utilisé à la place.



Si l'option *Chiffrement TLS de la base de données* est cochée, des champs supplémentaires pour configurer la connexion TLS à la base de données apparaissent dans le formulaire (MySQL ou PostgreSQL uniquement).

Si *Stocker les identifiants dans* est défini sur HashiCorp Vault ou CyberArk Vault, des paramètres supplémentaires deviennent disponibles :

- pour HashiCorp Vault : point de terminaison de l'API Vault, préfixe du coffre, chemin du secret et jeton d'authentification ;
- pour CyberArk Vault : point de terminaison de l'API Vault, préfixe du coffre, chaîne de requête du secret et certificats. En cochant la case *Certificats Vault*, deux nouveaux champs permettant de spécifier les chemins vers le fichier de certificat SSL et le fichier de clé SSL apparaîtront.

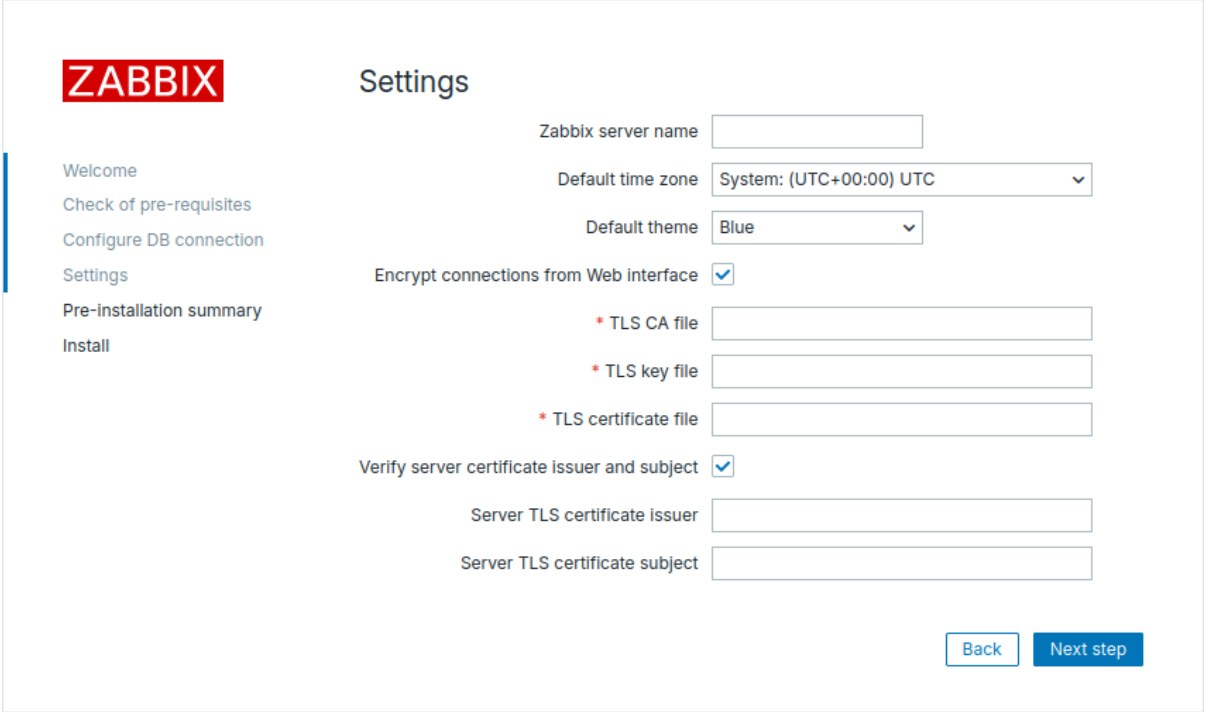


Paramètres

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

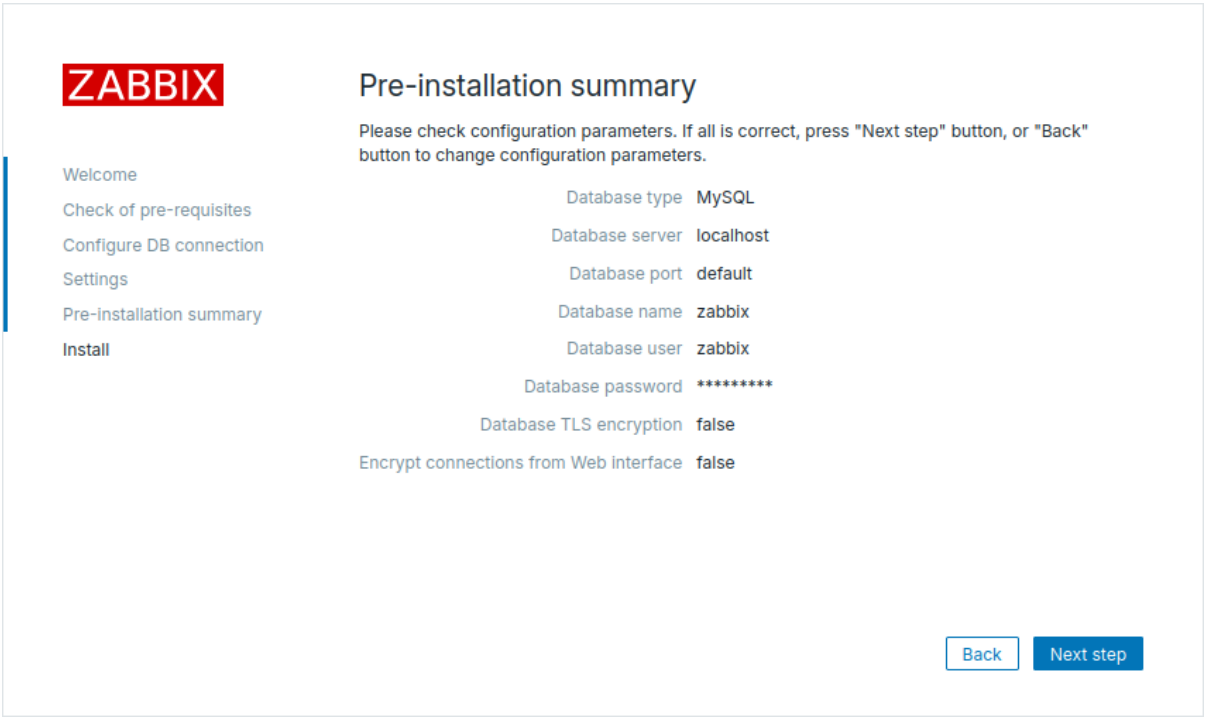
Définissez le fuseau horaire et le thème par défaut pour l'interface web.

Si l'option *Encrypt connections from Web interface* est cochée, des champs supplémentaires pour la configuration de la connexion TLS entre le serveur Zabbix et l'interface web apparaissent dans le formulaire.

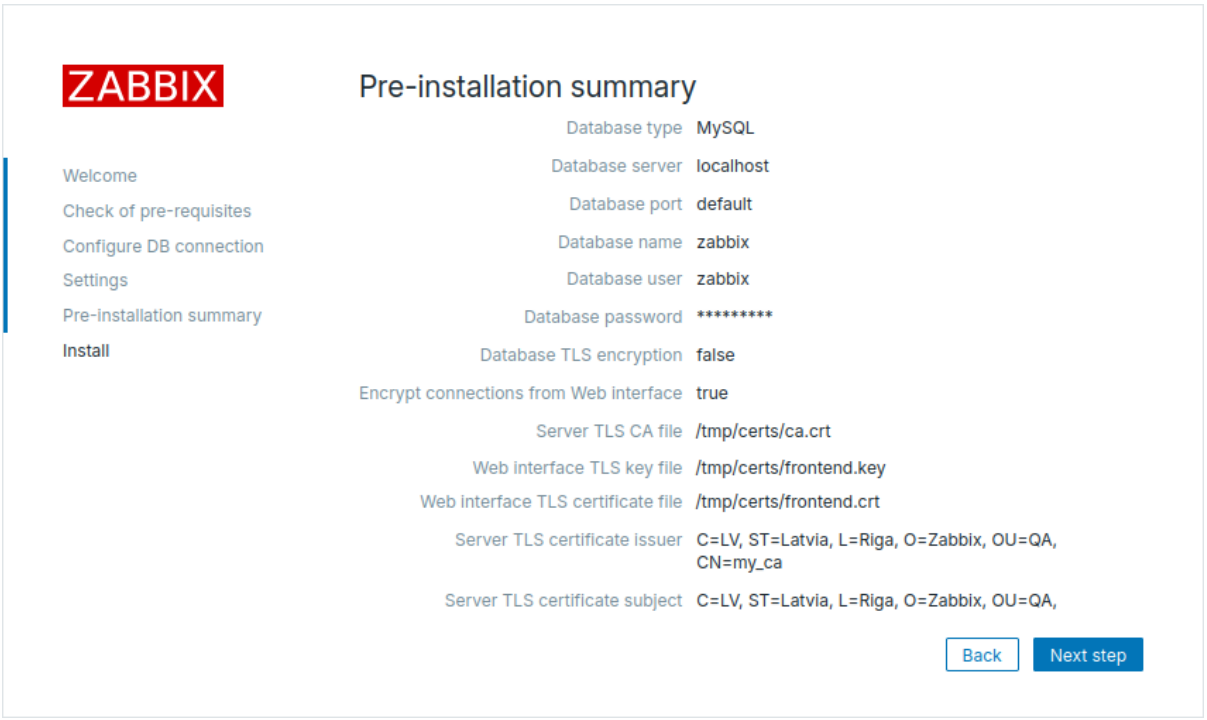


Résumé de la préinstallation

Consultez un résumé des paramètres.

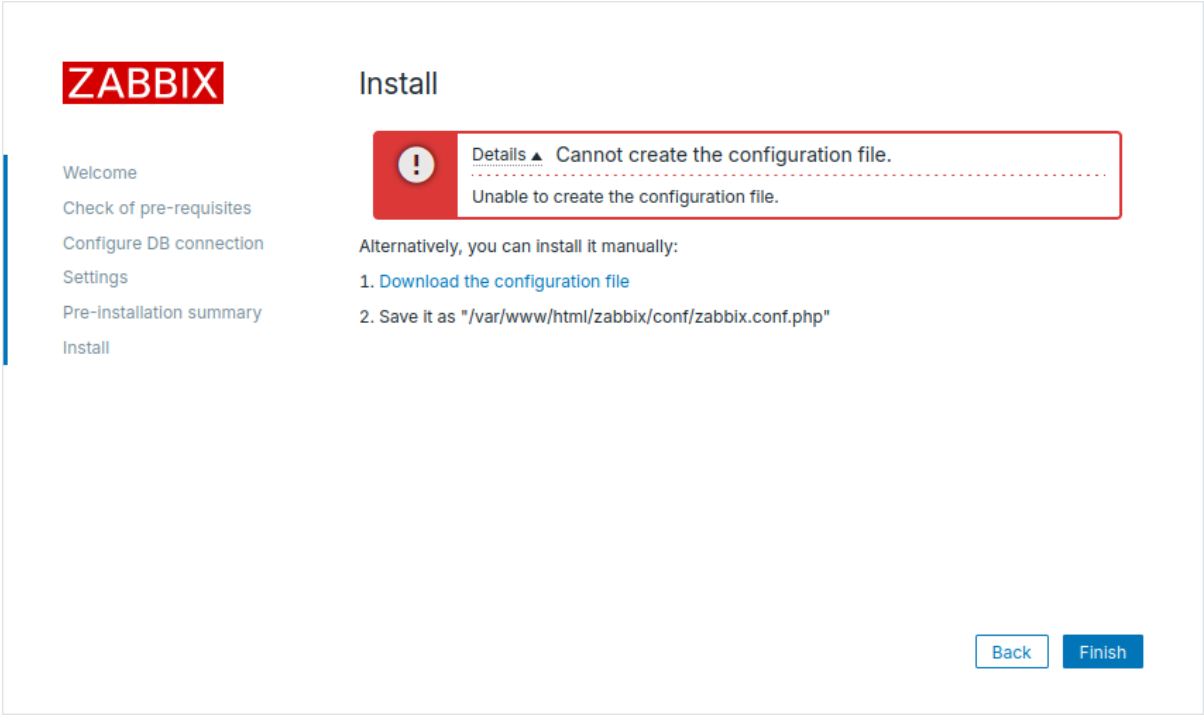


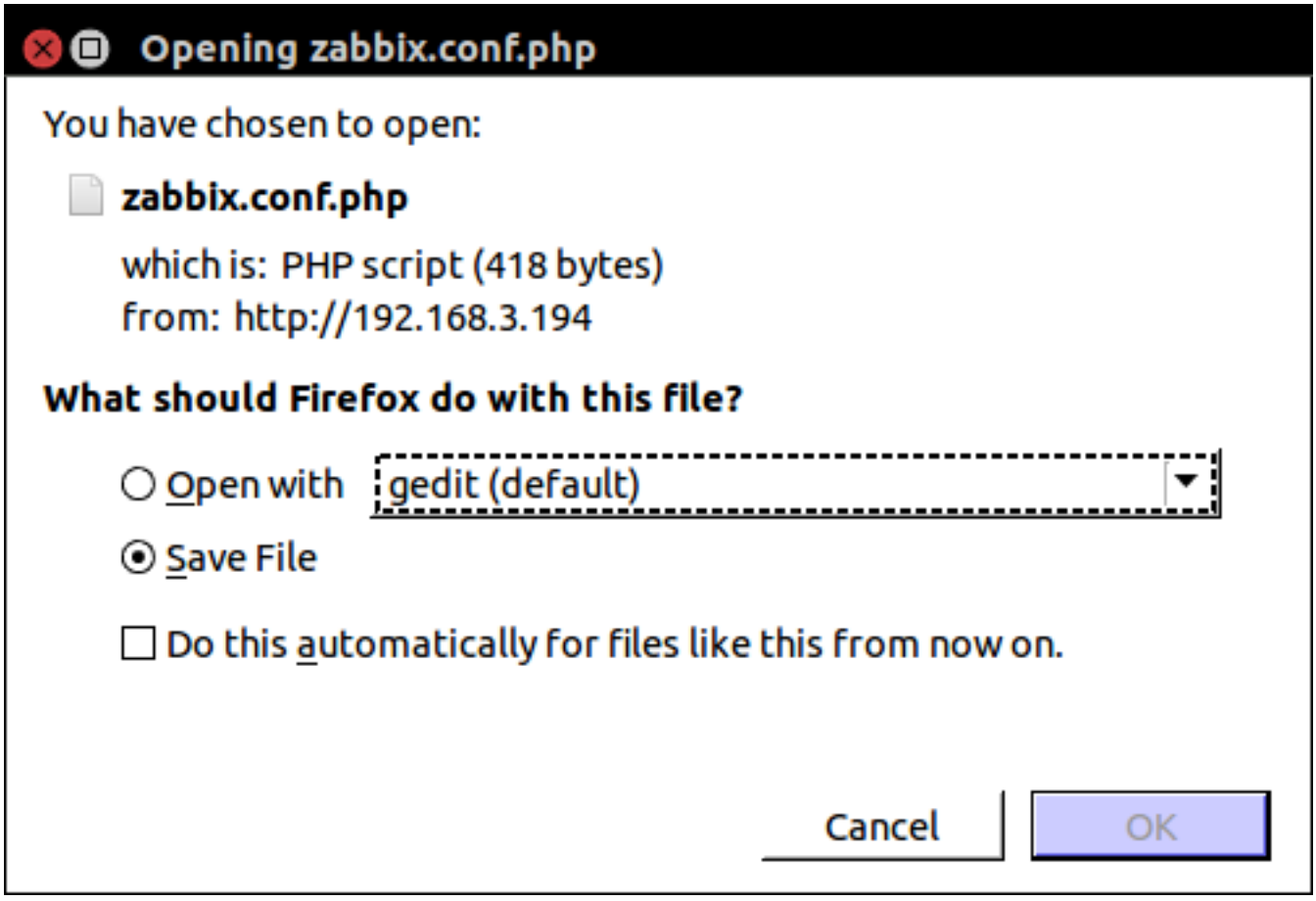
La sous-page affichera les données si une configuration TLS a été ajoutée.



Installation

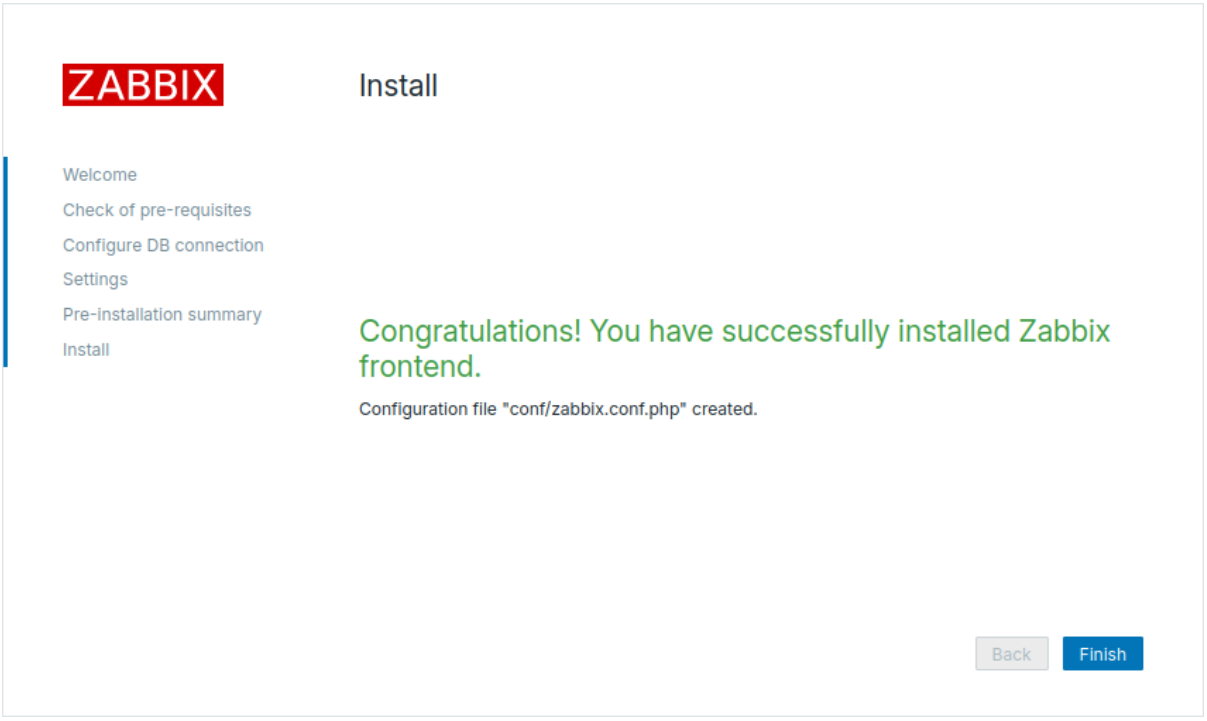
Si vous installez Zabbix à partir des sources, téléchargez le fichier de configuration et placez-le dans conf/ dans le sous-répertoire des documents HTML du serveur web où vous avez copié les fichiers PHP de Zabbix.





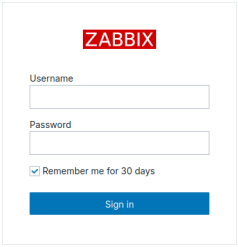
Note:
Si l'utilisateur du serveur web dispose d'un accès en écriture au répertoire conf/, le fichier de configuration sera enregistré automatiquement et il sera possible de passer immédiatement à l'étape suivante.

Terminez l'installation.



Connexion

L'interface web de Zabbix est prête ! Le nom d'utilisateur par défaut est **Admin**, le mot de passe **zabbix**.



Poursuivez avec le guide de supervision Linux.

2 Guides de démarrage rapide

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

Utilisez ces guides rapides pour commencer à superviser des ressources populaires :

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

Les guides de démarrage rapide se concentrent sur l'utilisation des modèles Zabbix officiels pour la supervision.
Un modèle est un ensemble préconfiguré d'éléments de supervision.
Un modèle 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 une supervision prête à l'emploi, vous pouvez également souhaiter personnaliser leurs 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

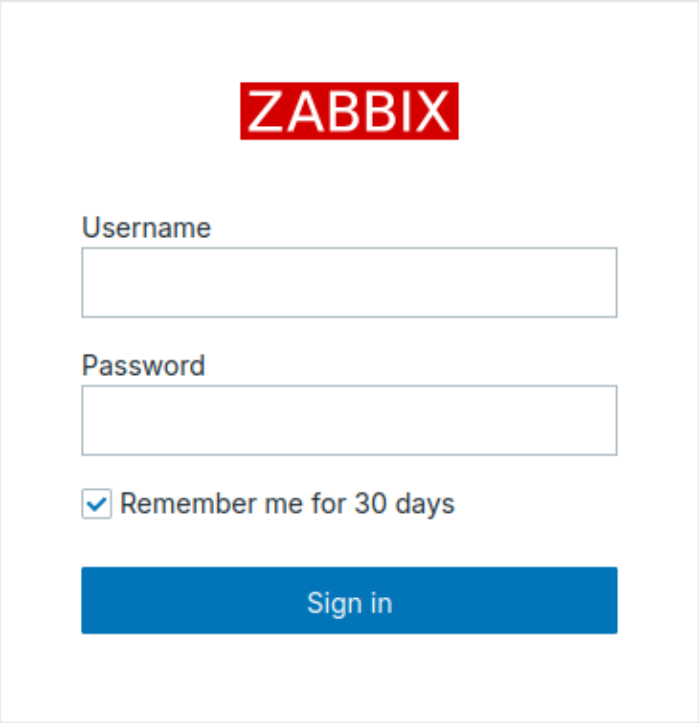
- Nouveau déclencheur
- Réception d'une notification de problème
- Nouveau modèle

1 Connexion et configuration de l'utilisateur

Vue d'ensemble

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

Connexion



Voici l'écran d'accueil de Zabbix. Saisissez le nom d'utilisateur **Admin** avec 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 le mot de passe par défaut du compte Admin immédiatement après la première connexion.

Connexion persistante

Pour rester connecté pendant une durée maximale de 30 jours, sélectionnez *Se souvenir de moi pendant 30 jours* avant de cliquer sur *Se connecter*.

Se souvenir de moi pendant 30 jours activé :

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

Se souvenir de moi pendant 30 jours désactivé :

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

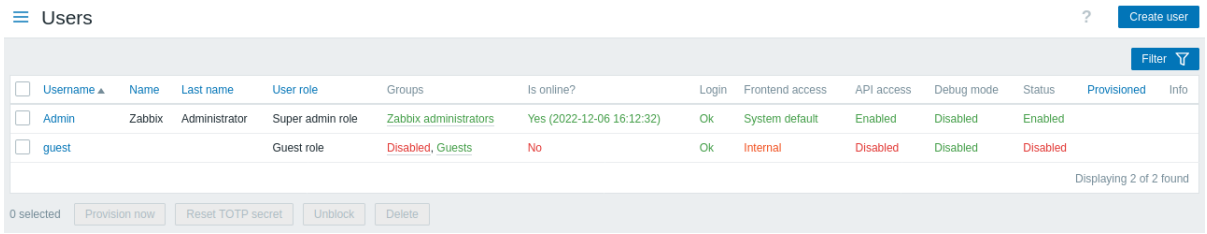
Protection contre les attaques par force brute

En cas de cinq tentatives de connexion consécutives échouées, l'interface Zabbix suspendra les connexions pendant 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.

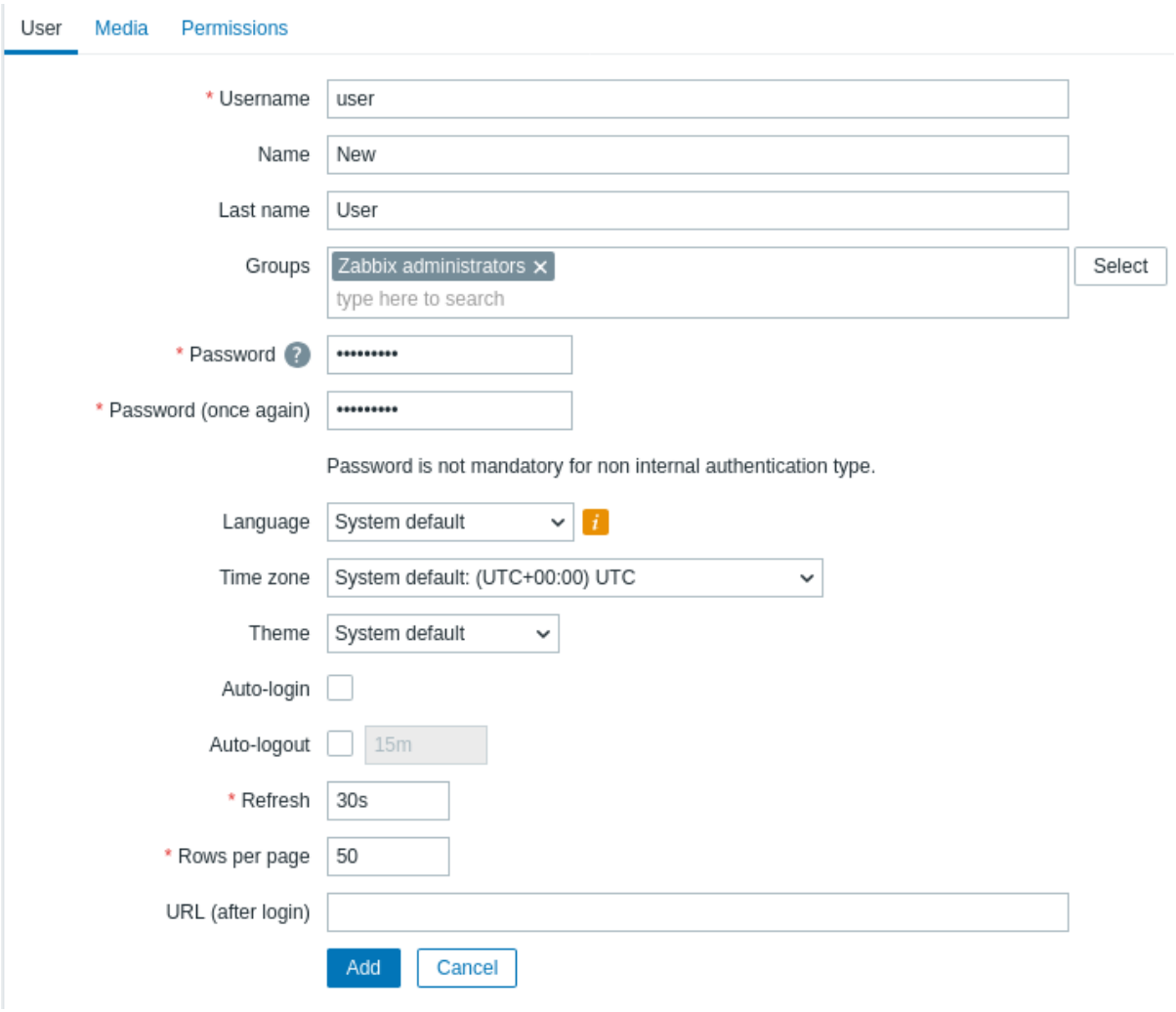
Ajout d'un utilisateur

Pour afficher les informations sur les utilisateurs, accédez à *Utilisateurs* > *Utilisateurs* dans le menu vertical de la barre latérale.



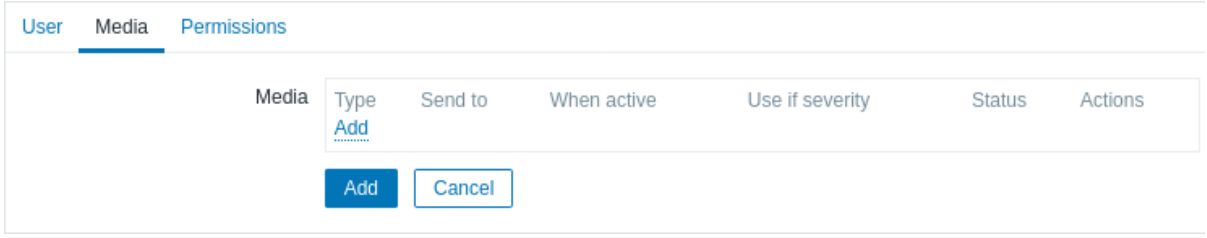
Pour ajouter un nouvel utilisateur, sélectionnez *Créer un utilisateur* dans le coin supérieur droit.

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



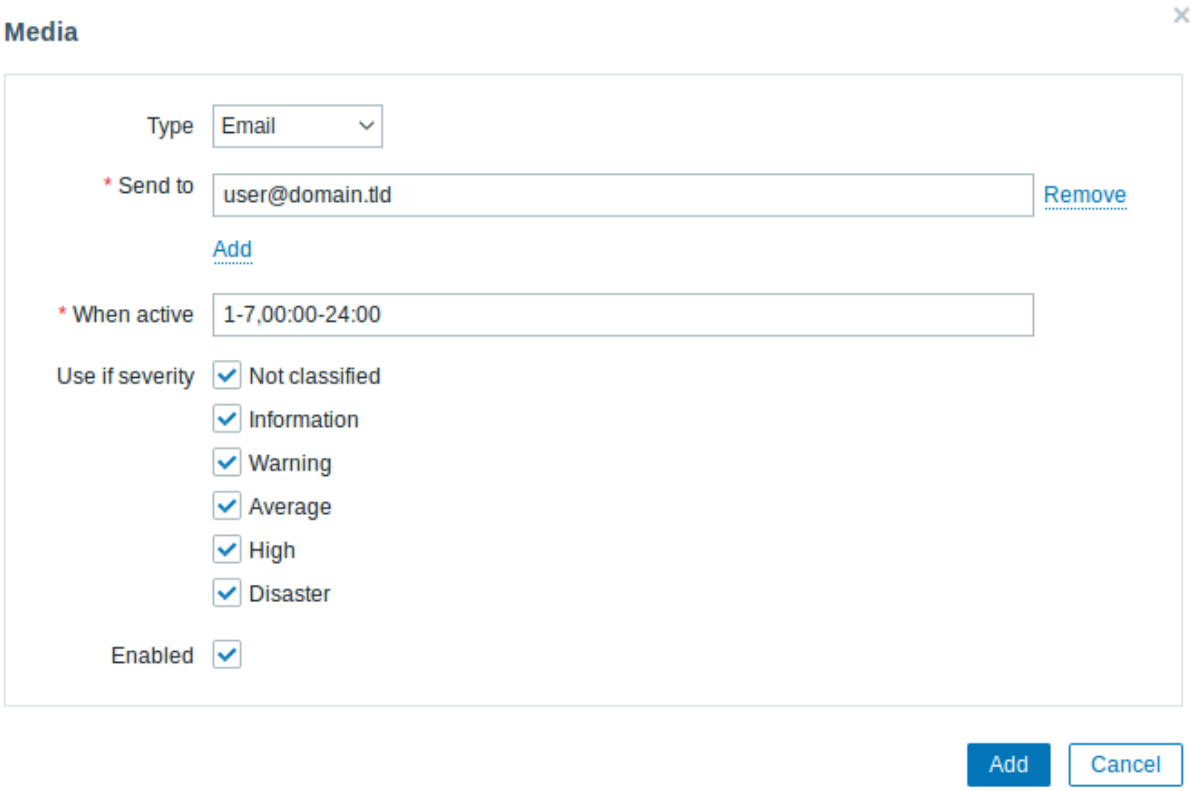
Tous les champs obligatoires sont marqués d'un astérisque rouge. Pour plus de détails sur les champs de saisie de ce formulaire de configuration, consultez la page Paramètres utilisateur.

Par défaut, les nouveaux utilisateurs n'ont aucun média (méthode de diffusion des notifications) défini. Pour en créer un, accédez à l'onglet « Media » et cliquez sur *Ajouter*.



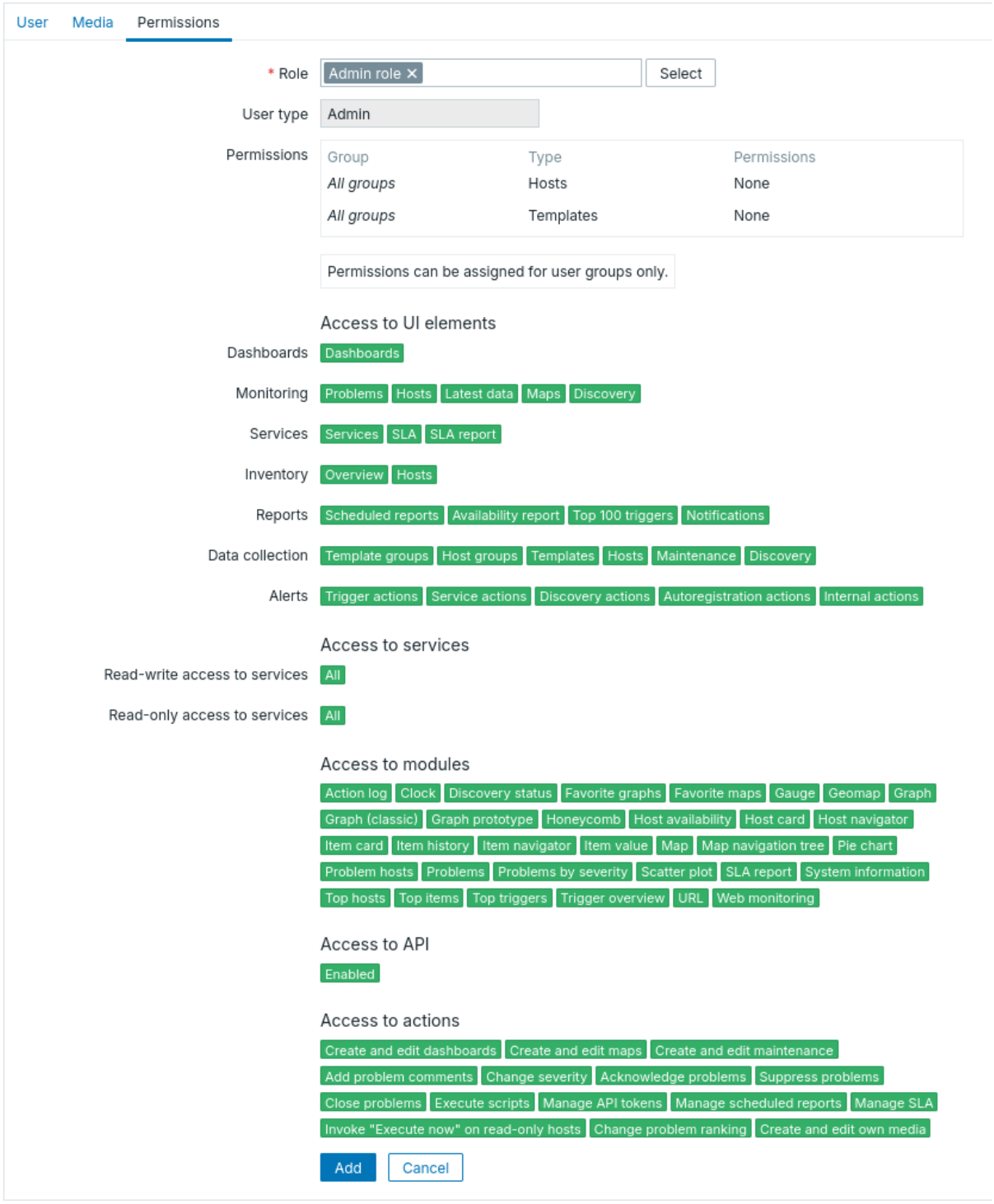
Dans la fenêtre contextuelle, saisissez l'adresse e-mail de l'utilisateur.

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



Cliquez sur *Ajouter* pour enregistrer le média, puis accédez à l'onglet Permissions.

L'onglet Permissions comporte un champ obligatoire, *Rôle*. Le rôle détermine quels éléments du frontend l'utilisateur peut voir et quelles actions il est autorisé à effectuer. Appuyez sur Sélectionner et choisissez l'un des rôles dans la liste. Par exemple, sélectionnez *Rôle administrateur* pour autoriser l'accès à toutes les sections du frontend Zabbix, à l'exception de l'Administration. Par la suite, vous pourrez modifier les autorisations ou créer davantage de rôles utilisateur. Une fois un rôle sélectionné, les autorisations apparaîtront dans le même onglet :

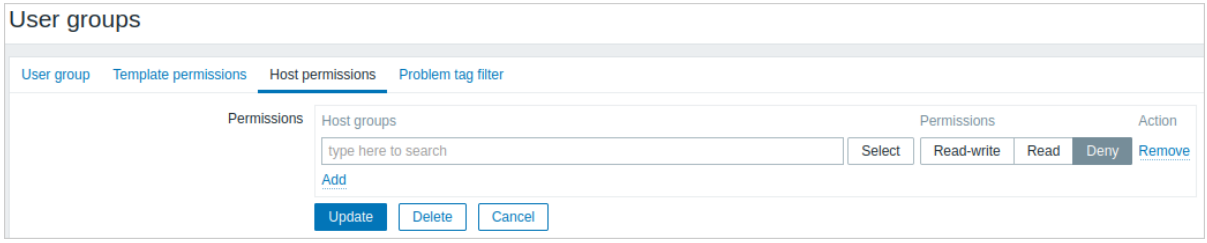


Cliquez sur *Ajouter* dans le formulaire des propriétés de l'utilisateur pour enregistrer l'utilisateur. Le nouvel utilisateur apparaît dans la liste des utilisateurs.



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 des propriétés de *User groups*, allez dans 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 afficher 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, puis 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 les groupes de modèles. Les étapes sont identiques à l'attribution de permissions aux groupes d'hôtes. Une vue d'ensemble des modèles est disponible dans la section New template de ce guide de démarrage rapide.

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

Terminé ! Vous pouvez essayer de vous connecter en utilisant les identifiants du nouvel utilisateur.

2 Nouvel hôte

Vue d'ensemble

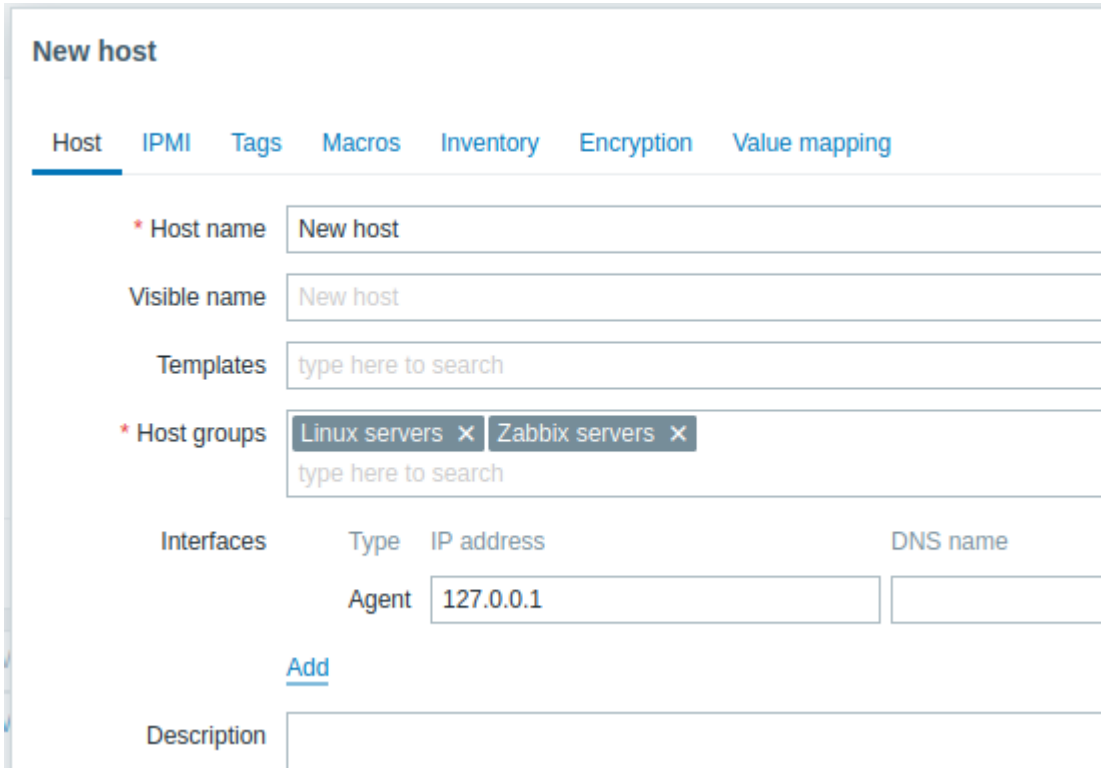
Dans cette section, vous apprendrez à configurer un nouvel hôte.

Dans Zabbix, un hôte est une entité en réseau (physique, virtuelle) que vous souhaitez surveiller. La définition de ce qui peut être un « hôte » dans Zabbix est assez souple. 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 *Collecte de données* > *Hôtes* ainsi que dans les sections du menu *Supervision* > *Hôtes*. Il existe déjà un hôte prédéfini, appelé « Zabbix server », 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 de saisie obligatoires sont marqués d'un astérisque rouge.

Le strict minimum à renseigner ici est :

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 créer un nouveau groupe.

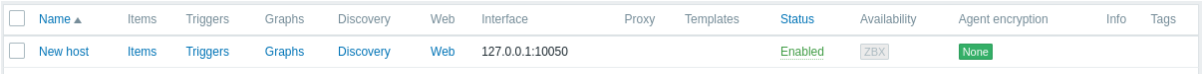
Note:
Toutes les autorisations d'accès sont attribuées aux groupes d'hôtes, et non aux hôtes individuels. C'est pourquoi un hôte doit appartenir à au moins un groupe.

Interfaces : adresse IP

- Bien qu'il ne s'agisse pas techniquement d'un champ obligatoire, une interface d'hôte est nécessaire pour collecter certaines métriques. Pour utiliser les vérifications passives de l'agent Zabbix, indiquez l'adresse IP ou le DNS de l'agent dans ce champ. Notez que vous devez également spécifier 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/le même DNS aux deux endroits.

Autres options peuvent être laissées inchangées, en utilisant leurs valeurs par défaut.

Une fois terminé, cliquez sur *Ajouter*. Votre nouvel hôte devrait être visible dans la liste des hôtes.



La colonne Disponibilité contient des indicateurs de disponibilité de l'hôte pour chaque interface. Nous avons défini une interface d'agent Zabbix ; nous pouvons donc utiliser l'icône de disponibilité de l'agent (avec « ZBX » dessus) pour comprendre la disponibilité de l'hôte :

- ZBX - le statut de l'hôte n'a pas encore été établi ; aucune vérification de métrique n'a encore eu lieu
- ZBX - l'hôte est disponible, une vérification de métrique a réussi
- ZBX - l'hôte n'est pas disponible, une vérification de métrique a échoué (placez le curseur de la souris sur l'icône pour voir le message d'erreur). Il peut y avoir un problème de communication, éventuellement causé par des identifiants d'interface incorrects. Vérifiez que le serveur Zabbix est en cours d'exécution et essayez également d'actualiser la page plus tard.

3 Nouvel élément

Aperçu

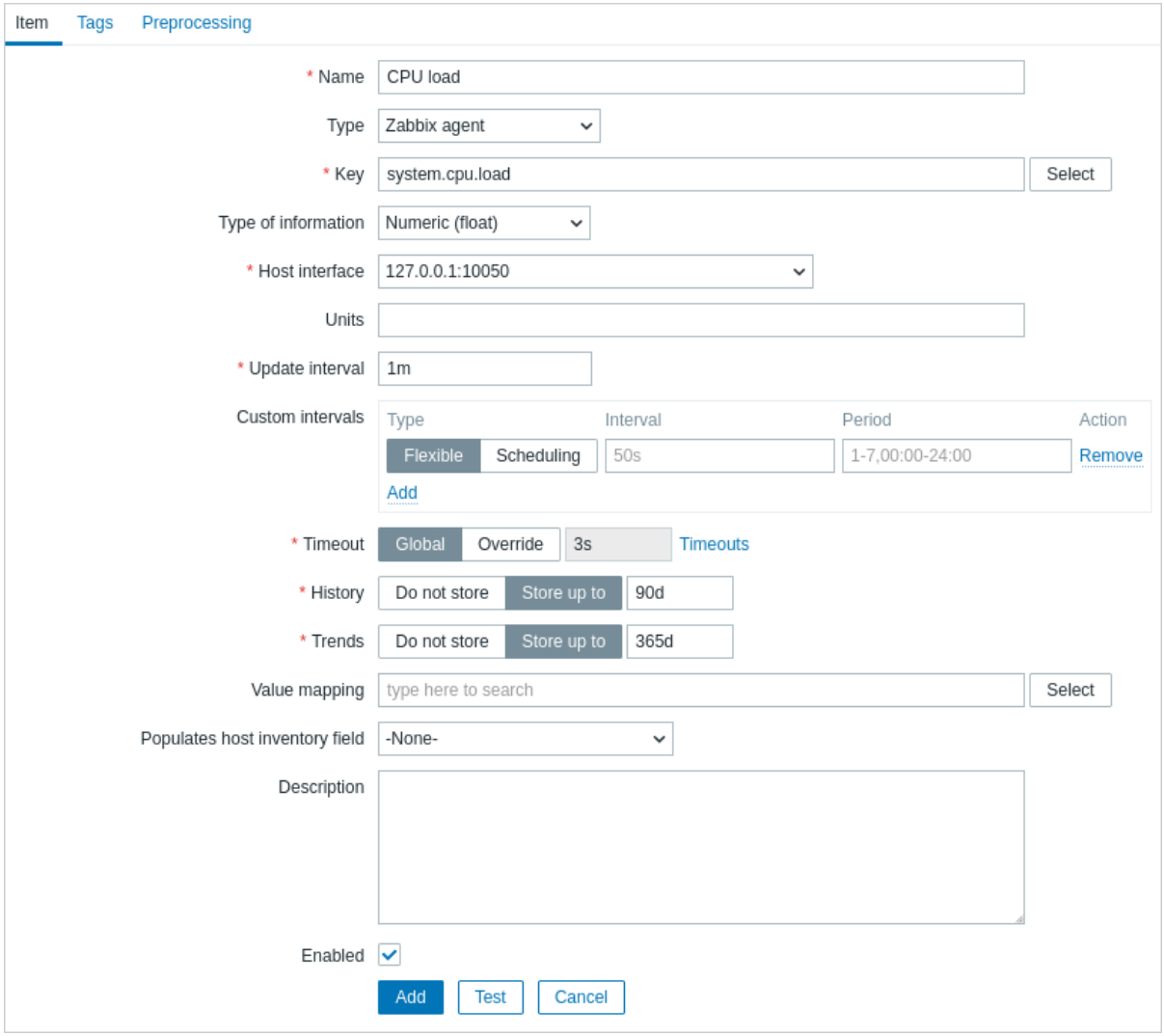
Dans cette section, vous apprendrez à configurer un élément.

Les éléments constituent 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 à partir d'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 « Nouvel hôte » que nous avons créé.

Cliquez sur le lien *Éléments* dans la ligne de « Nouvel hôte », 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 :

Nom

- Saisissez *Charge CPU* comme valeur. Ce sera le 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 qui sera collecté. Cette clé particulière n'est que l'une des clés prédéfinies fournies avec l'agent Zabbix.

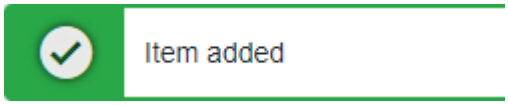
Type d'information

- 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. Il s'agit d'une bonne pratique pour éviter que la base de données ne conserve trop de valeurs historiques.

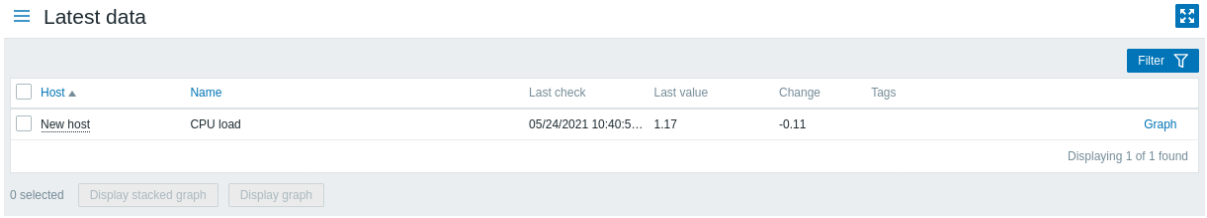
Les autres options peuvent être laissées inchangées, en utilisant leurs valeurs par défaut.

Une fois terminé, cliquez sur *Ajouter*. Le nouvel élément devrait apparaître dans la liste des éléments, et vous devriez voir un message de réussite.



Affichage des données

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



Cela dit, l'arrivée des premières données peut prendre jusqu'à 60 secondes. 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 vous ne voyez aucune valeur 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.

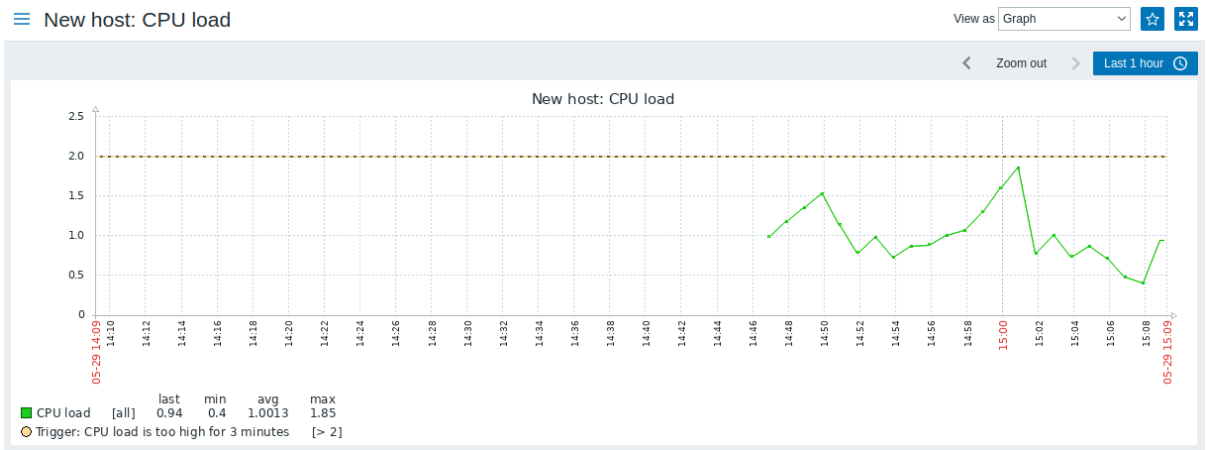
Si vous ne voyez pas d'informations sur l'élément comme dans la capture d'écran, assurez-vous que :

- 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 ;
- le statut de l'hôte est « Monitored » et son icône de disponibilité est verte ;
- l'hôte sélectionné dans le filtre d'hôte est correct ;
- l'élément est activé.

Graphiques

Une fois l'élément en fonctionnement 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 à l'exécution.

Pour afficher le graphique, accédez à *Monitoring* > *Latest data* et cliquez sur le lien « Graph » à côté de l'élément.



4 Nouveau déclencheur

Vue d'ensemble

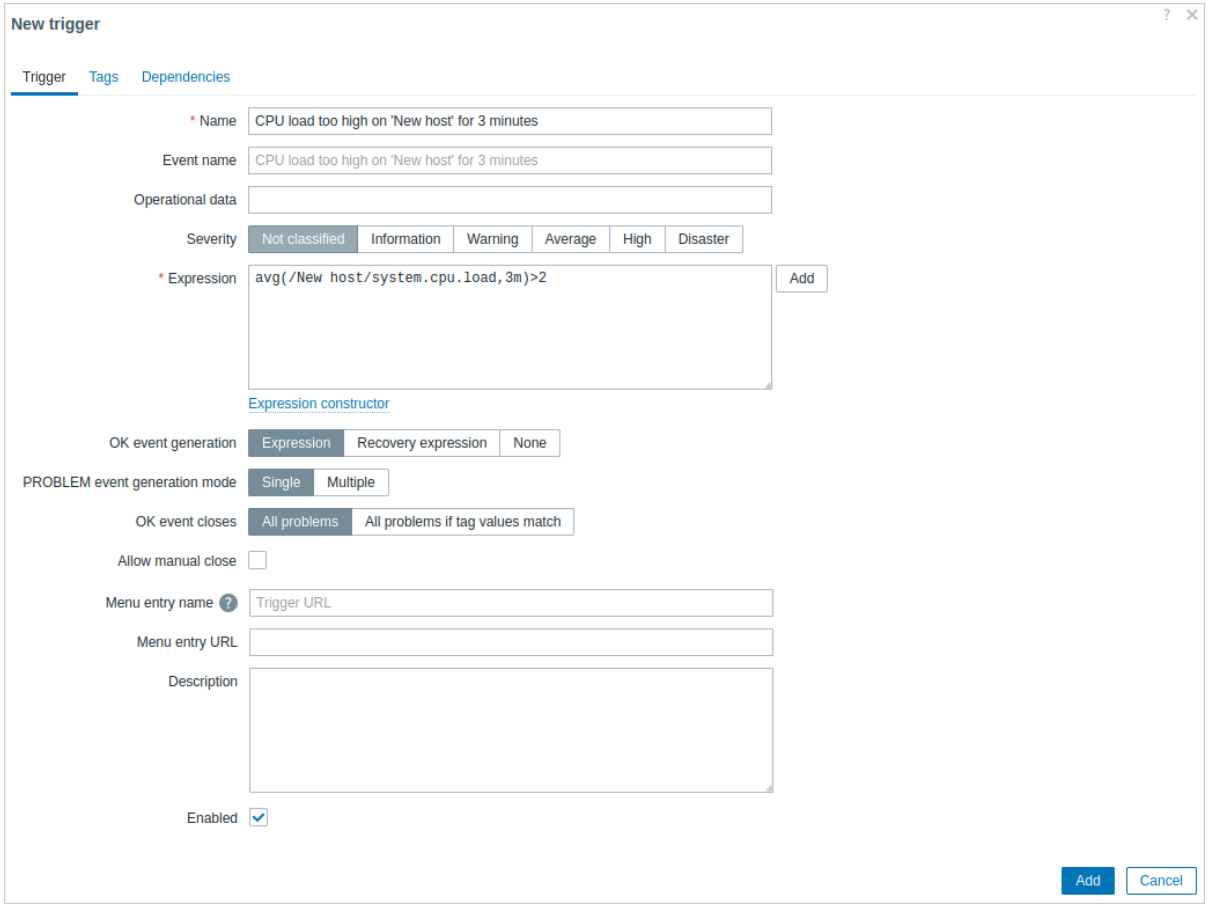
Dans cette section, vous apprendrez à configurer un déclencheur.

Les éléments ne font que collecter 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 « déclenchera » ou passera 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, allez dans *Collecte de données* > *Hôtes*. Cliquez sur le lien *Déclencheurs* dans la ligne de « New host », puis cliquez sur *Créer un déclencheur*. Un formulaire de configuration du déclencheur s'ouvre.



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

Nom

- Saisissez *CPU load 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

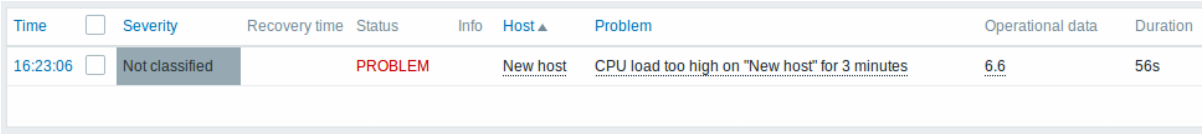
Il s'agit de l'expression du déclencheur. Assurez-vous que l'expression est saisie correctement, jusqu'au dernier symbole. La clé d'élément ici (system.cpu.load) est utilisée pour faire référence à l'élément. Cette expression particulière indique essentiellement que le seuil du problème est dépassé lorsque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

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

Affichage de l'état du déclencheur

Une fois un déclencheur défini, vous souhaiterez peut-être voir son état.

Si la charge CPU a dépassé le niveau de seuil que vous avez défini dans le déclencheur, le problème sera affiché dans *Surveillance* > *Problèmes*.



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

5 Réception d'une notification de problème

Vue d'ensemble

Dans cette section, vous apprendrez à configurer l'alerte sous forme de notifications dans Zabbix.

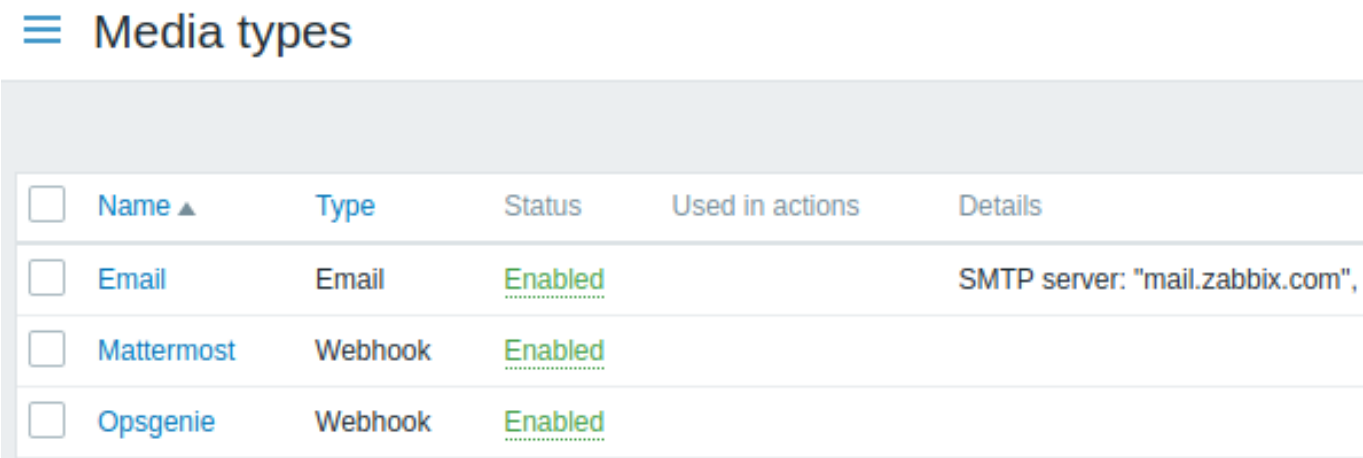
Avec des éléments collectant des données et des déclencheurs conçus pour se « déclencher » en cas de situation problématique, il est également utile de disposer d'un mécanisme d'alerte pour notifier les événements importants même lorsque l'interface web de Zabbix n'est pas activement consultée.

C'est précisément le rôle des notifications. L'e-mail étant le moyen de diffusion le plus courant pour les notifications de problème, nous allons voir comment configurer une notification par e-mail.

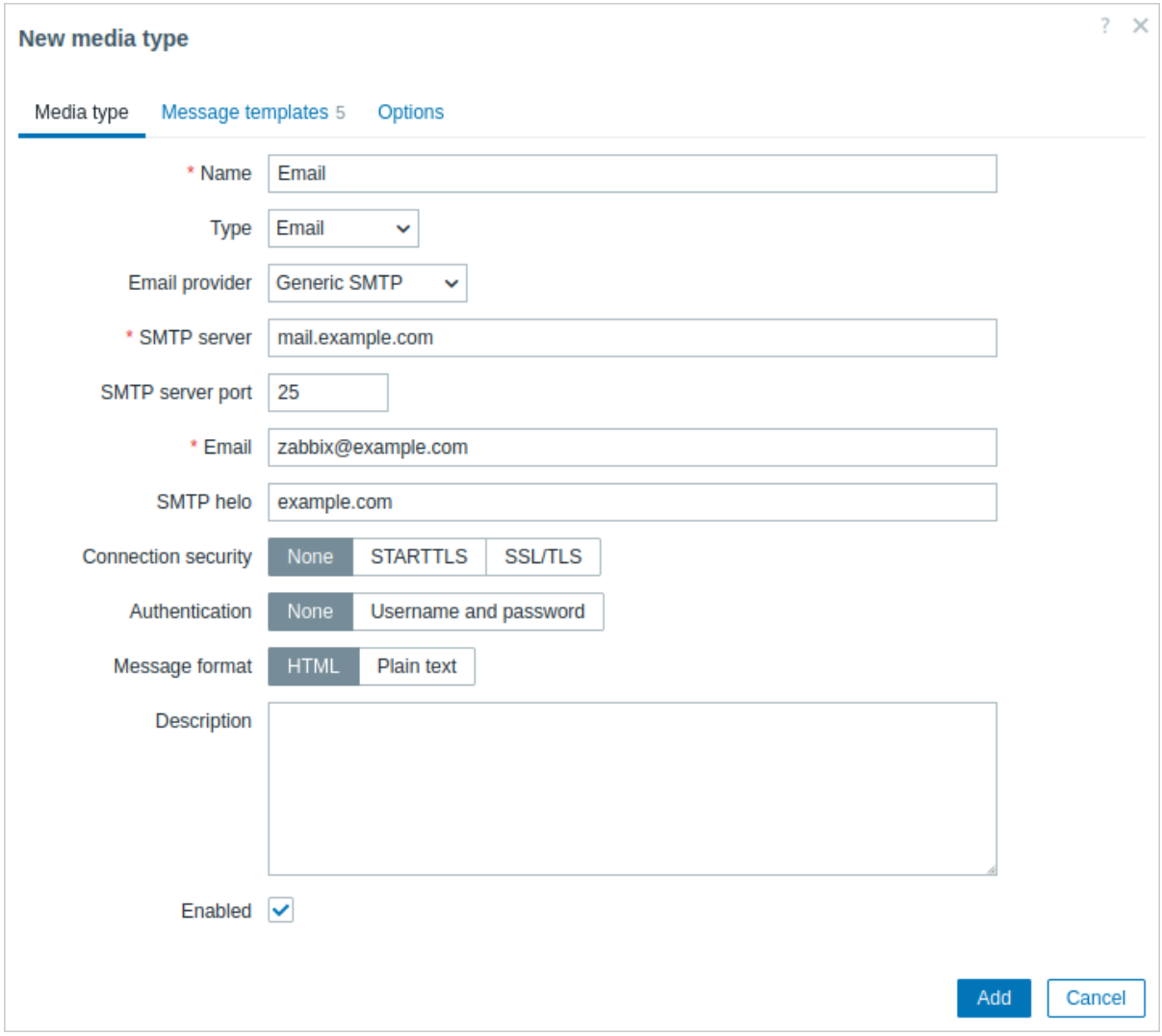
Paramètres de messagerie

Au départ, plusieurs méthodes d'envoi de notification prédéfinies sont disponibles dans Zabbix. Email en fait partie.

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



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



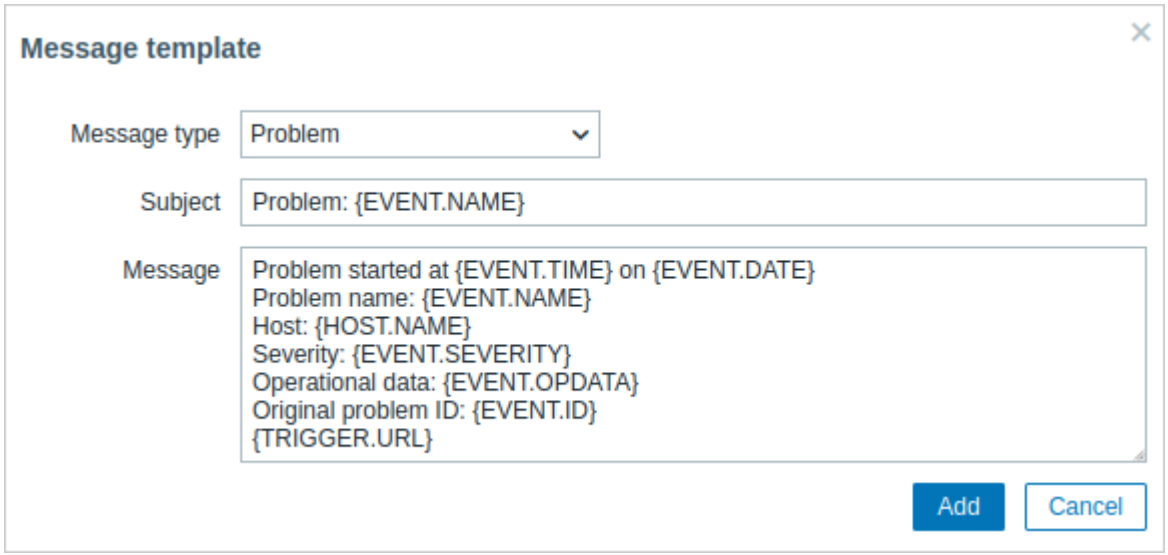
Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge.

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

Note:
La valeur du champ *Email* sera utilisée comme adresse « From » pour les notifications envoyées par Zabbix.

Ensuite, il est nécessaire de définir le contenu du message de problème. Le 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.

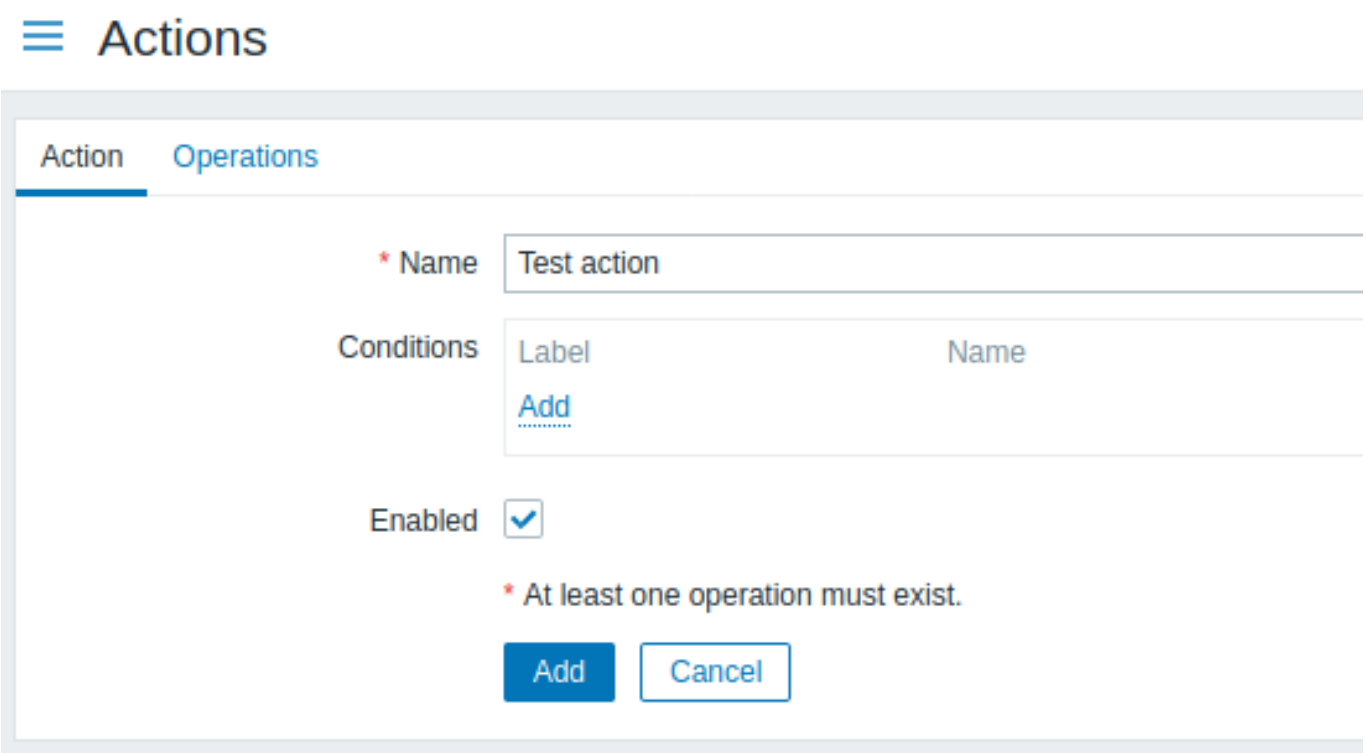


Cliquez sur *Add* lorsque vous êtes prêt, puis enregistrez le formulaire.

Vous avez maintenant configuré *Email* comme type de média opérationnel. Le type de média doit également être associé aux utilisateurs en définissant des adresses d'envoi 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*.

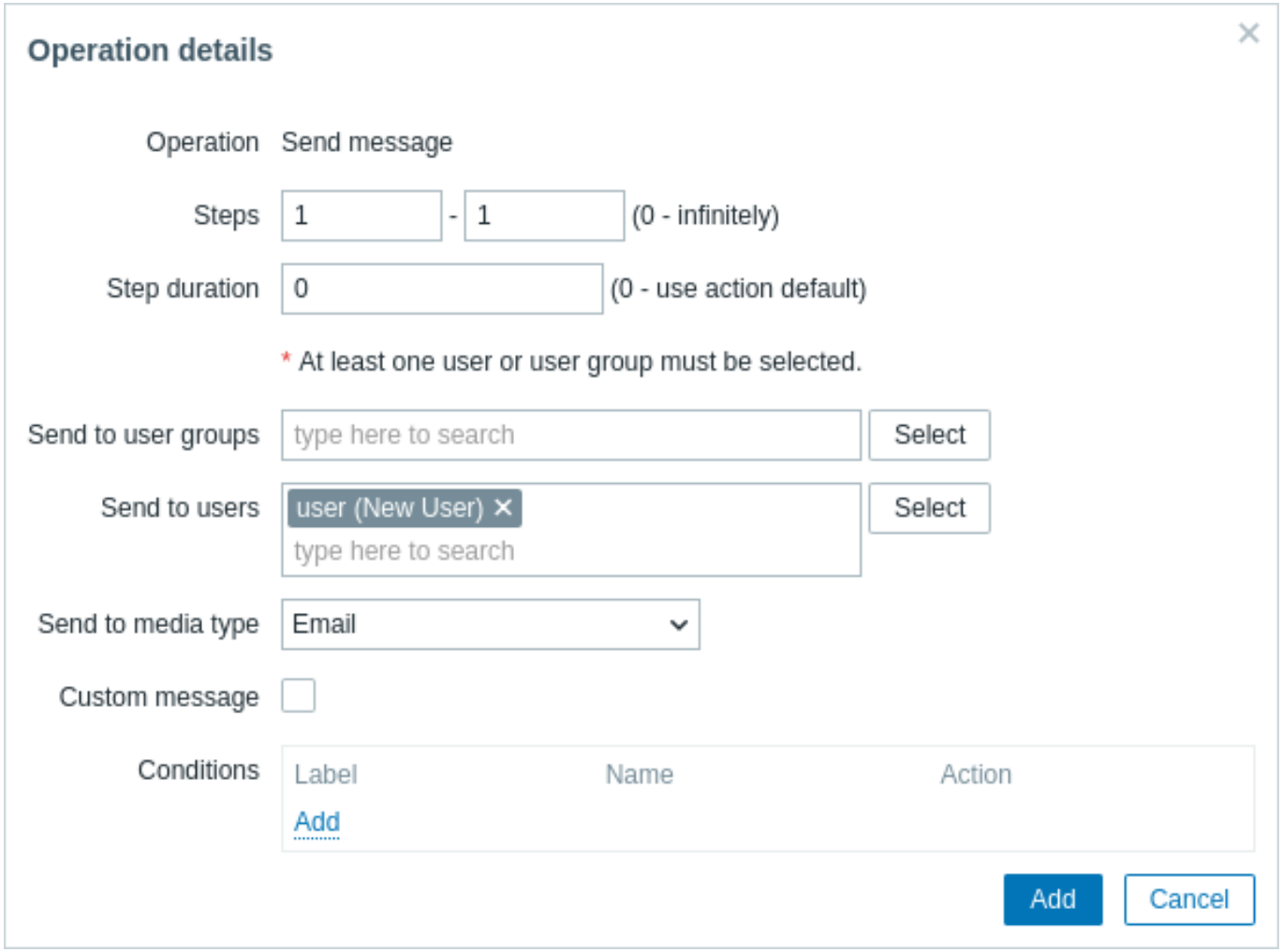


Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge.

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

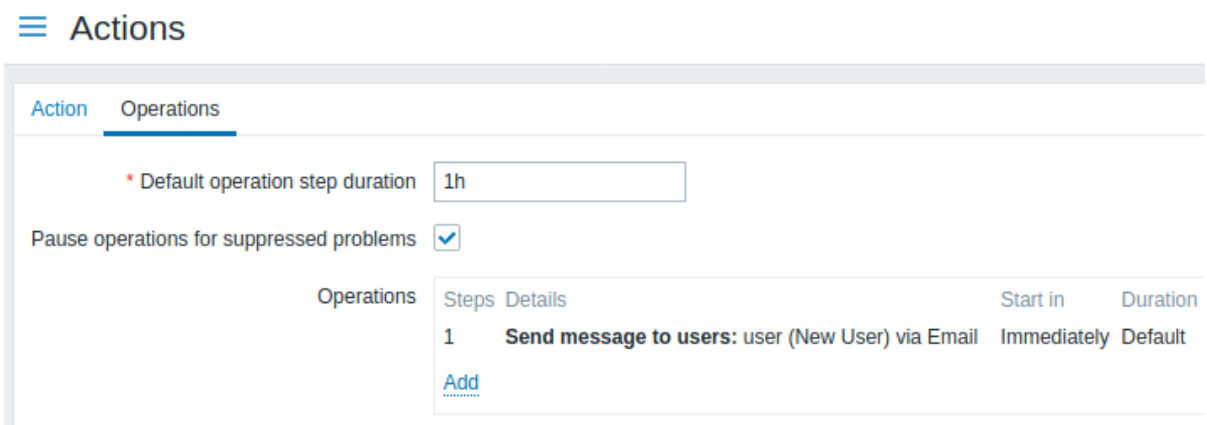
Dans le cas le plus simple, si nous n'ajoutons pas de conditions plus spécifiques, l'action sera exécutée lors de tout changement de déclencheur de « Ok » à « Problème ».

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



Tous les champs de saisie 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 au type de média*. Une fois cela fait, cliquez sur *Ajouter* et l'opération devrait être ajoutée :



C'est tout pour une configuration d'action simple ; cliquez donc sur *Ajouter* dans le formulaire d'action.

Réception d'une notification

Maintenant que l'envoi des notifications est configuré, il serait intéressant d'en recevoir réellement 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 maintenant 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 par e-mail.

Attention:
Si les notifications ne fonctionnent pas :

- vérifiez une nouvelle fois que les paramètres de messagerie et l'action ont tous deux été correctement configurés
- assurez-vous que l'utilisateur que vous avez créé dispose au minimum des droits de lecture sur l'hôte qui a généré l'événement, comme indiqué à l'étape *Adding user*. 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.
- De plus, vous pouvez consulter le journal des actions en allant dans *Reports* > *Action log*.

6 Nouveau modèle

Aperçu

Dans cette section, vous apprendrez à configurer un modèle.

Précédemment, nous avons vu comment configurer un élément, un déclencheur et comment recevoir une notification de problème pour l'hôte.

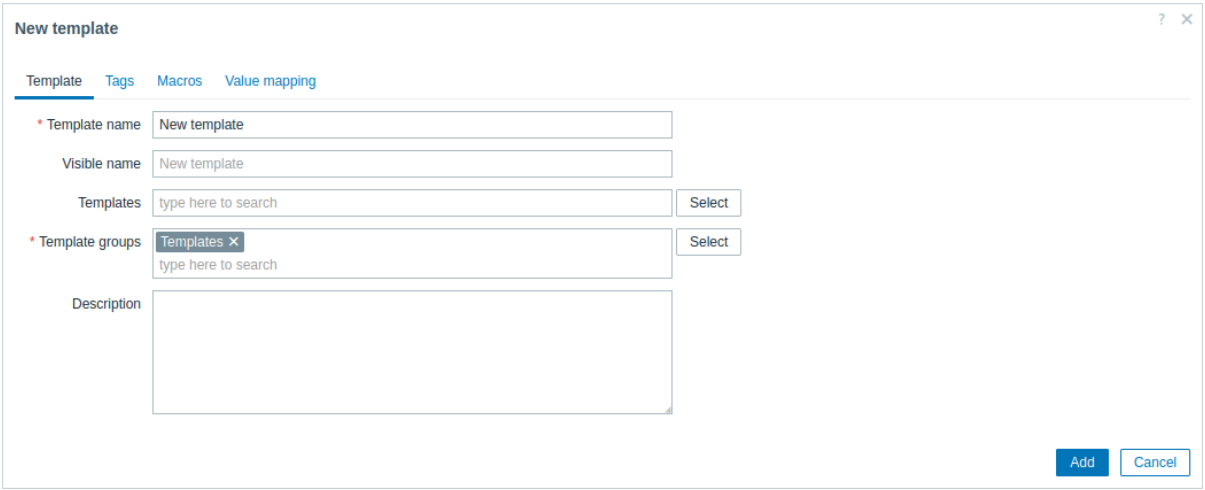
Bien que toutes ces étapes offrent en elles-mêmes une grande flexibilité, cela peut sembler représenter beaucoup d'étapes à effectuer si cela est nécessaire pour, par exemple, un millier d'hôtes. Une certaine automatisation serait utile.

C'est là que les modèles entrent en jeu. Les modèles permettent de regrouper des éléments, des déclencheurs et d'autres entités utiles afin qu'ils puissent être réutilisés encore et encore en les appliquant aux hôtes en une seule étape.

Lorsqu'un modèle est lié à un hôte, l'hôte hérite de toutes les entités du modèle. Ainsi, en pratique, un ensemble de contrôles préconfiguré 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 ouvrira un formulaire de configuration du modèle.



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

Les paramètres obligatoires à saisir ici sont les suivants :

Nom du modèle

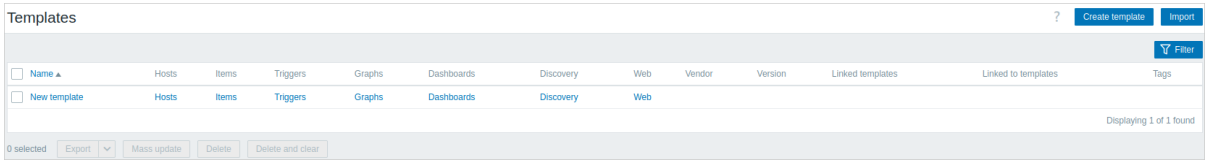
- Saisissez un nom de modèle. Les caractères alphanumériques, les espaces et les traits de soulignement sont autorisés.

Groupes de modèles

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

Note:
Les droits d'accès aux groupes de modèles sont attribués dans la configuration du groupe d'utilisateurs, dans l'onglet **Autorisations des modèles**, de la même manière que les autorisations des hôtes. Tous les droits d'accès sont attribués aux groupes, et non aux modèles individuels ; c'est pourquoi il est obligatoire d'inclure le modèle dans au moins un groupe.

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.



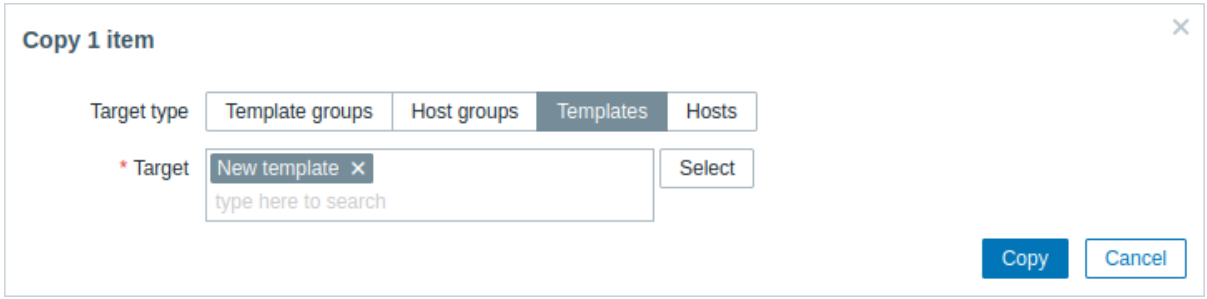
Comme vous pouvez le voir, le modèle est bien présent, mais il ne contient encore rien : aucun élément, déclencheur ou autre entité.

Ajout d'un élément à un modèle

Pour ajouter un élément au modèle, ouvrez la liste des éléments de « New host » en allant dans *Collecte de données* → *Hôtes* puis en cliquant sur *Éléments* à côté de « New host ».

Ensuite :

- Cochez la case de l'élément « CPU Load » dans la liste.
- Cliquez sur *Copier* sous la liste.
- Sélectionnez l'onglet *Modèles*.
- Sélectionnez le modèle vers lequel copier l'élément.



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

- Cliquez sur *Copier*.

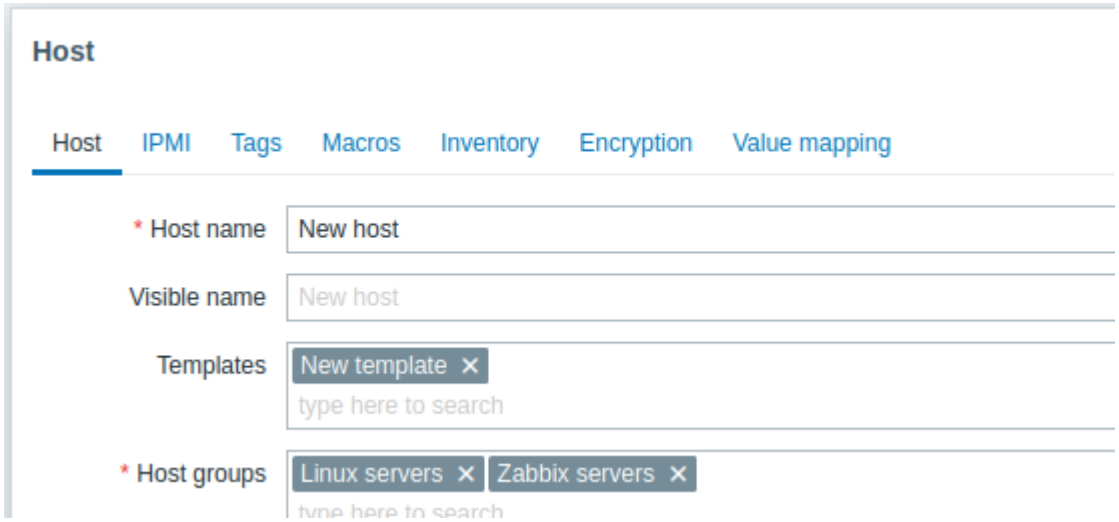
Si vous allez maintenant dans *Collecte de données* > *Modèles*, « New template » devrait contenir un nouvel élément.

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

Lier un modèle à un hôte

Une fois le modèle prêt, il ne reste plus qu'à l'ajouter à un hôte. Pour cela, allez dans *Collecte de données* > *Hôtes*, cliquez sur « New host » pour ouvrir son formulaire de configuration et repérez le champ **Templates**.

Commencez à saisir *New template* dans le champ *Templates*. Le nom du modèle que nous avons créé devrait apparaître dans la liste déroulante. Faites défiler la liste pour le sélectionner. Vérifiez qu'il apparaît bien dans le champ *Templates*.



Cliquez sur *Update* dans le formulaire pour enregistrer les modifications. Le modèle est maintenant ajouté à l'hôte, avec toutes les entités qu'il contient.

De cette manière, il peut également 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 livré avec un ensemble de modèles prédéfinis pour divers systèmes d'exploitation, appareils et applications. Pour commencer la supervision très rapidement, vous pouvez lier celui qui convient à un hôte, mais gardez à l'esprit que ces modèles doivent être ajustés avec précision à votre environnement. Certaines vérifications peuvent ne pas être nécessaires, et les intervalles d'interrogation peuvent être beaucoup trop fréquents.

Davantage d'informations sur les modèles sont disponibles.

Surveillance du serveur web Apache

Pour commencer à surveiller le serveur web Apache, suivez ce processus :

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 avoir le serveur Zabbix et l'interface web installés. 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, des 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 l'emplacement `/server-status`) affichant des données de performance 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, cela signifie que le module de statut est déjà activé. Sinon, vous devez activer le module de statut.

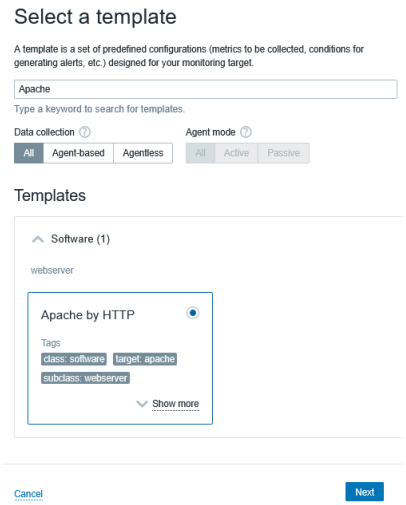
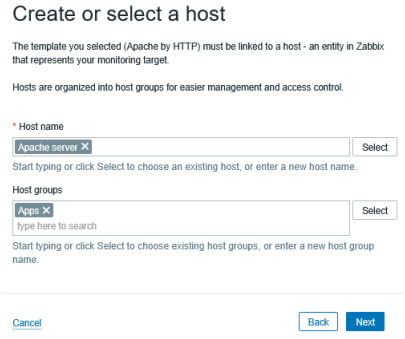
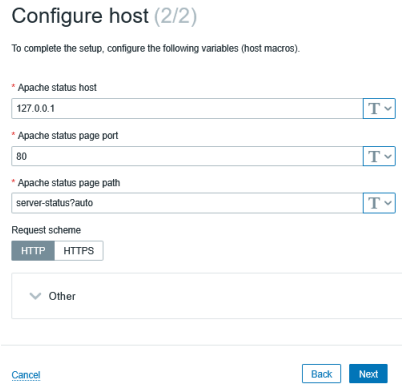
Appliquer le modèle de surveillance Apache

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, lancez l'Assistant hôte (dans *Collecte de données* > *Hôtes*) :

- **Sélectionnez le modèle** (« Apache by HTTP »)
- Créez un hôte Apache et ajoutez-lui un groupe. Cet hôte virtuel contiendra les données liées à Apache.

- Configurez l'hôte. Il est important de saisir l'adresse du serveur web Apache (127.0.0.1 sur une machine locale, adresse IP sur une machine distante).

Étape 1	Étape 2	Étape 3
		
Sélection du modèle.	Création de l'hôte/du groupe d'hôtes.	Configuration de l'hôte.

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

Pour terminer l'Assistant hôte, cliquez sur *Create* puis sur *Finish*.

Consultez les dernières données



Félicitations, la surveillance du serveur web Apache a démarré !

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

Surveillance d'un serveur Linux

Pour commencer à surveiller un serveur Linux, suivez ce processus :

1. Appliquez le modèle de surveillance Linux.
2. Installez l'agent Zabbix sur le serveur.
3. Consultez les dernières données.

Pour commencer à surveiller 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 surveiller. Les instructions d'installation de l'agent seront fournies au cours du processus.

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

Appliquer le modèle de surveillance Linux

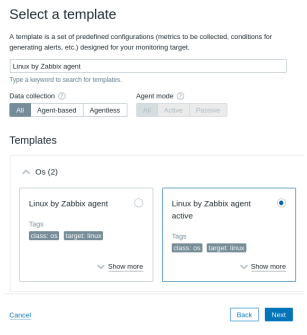
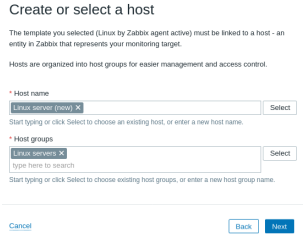
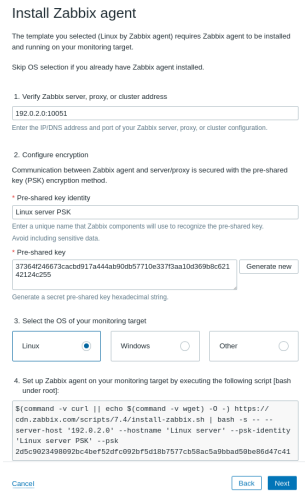
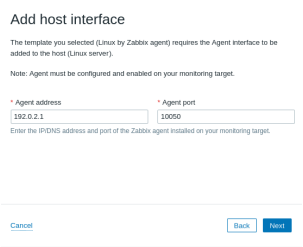
Dans Zabbix, ce modèle est prêt à l'emploi. Il contient des éléments préconfigurés prêts pour la collecte de données, la visualisation et l'analyse.

Pour appliquer le modèle, lancez l'assistant Hôte (dans *Collecte de données* > *Hôtes*) :

- **Sélectionnez le modèle** (« Linux by Zabbix agent active »)
- Créez un hôte pour le serveur Linux, ajoutez-lui un groupe. Cet hôte virtuel contiendra toutes les données liées à la surveillance du serveur Linux.
- 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'identité de la clé pré-partagée et la clé pré-partagée pour vous connecter à l'agent.

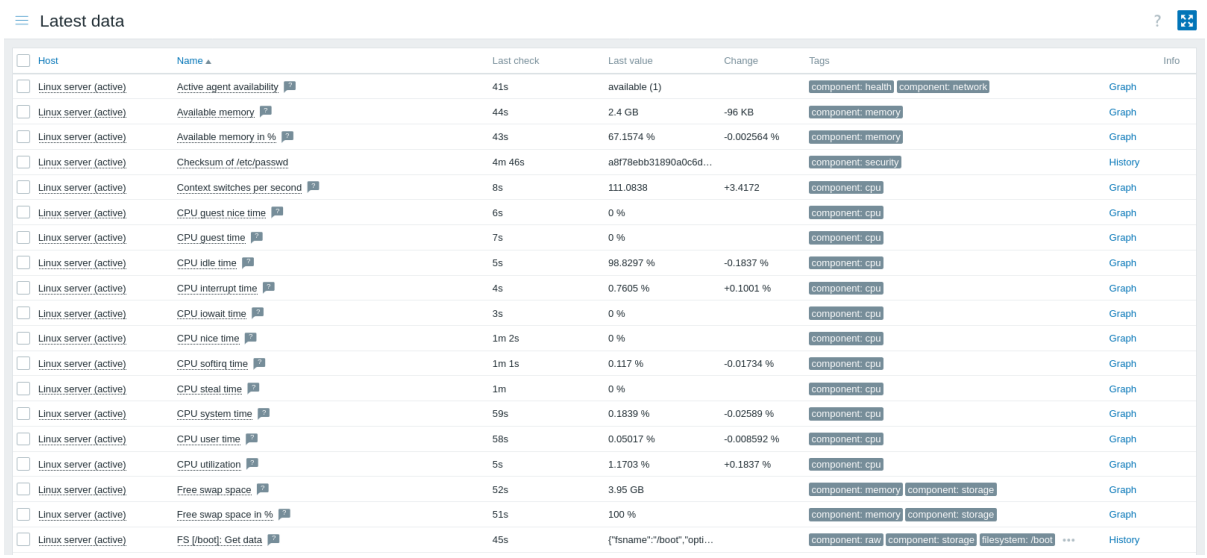
- Terminez la configuration de l'hôte. Saisissez l'adresse IP et le port de l'agent (serveur Linux).

Étape 1	Étape 2	Étape 3	Étape 4
			
Sélection du modèle.	Création de l'hôte/du groupe d'hôtes.	Configuration et installation de l'agent sur l'hôte.	Ajout de l'adresse IP/du port du serveur Linux comme interface de l'agent.

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

Pour terminer l'assistant Hôte, cliquez sur *Créer* puis sur *Terminer*.

Consultez les dernières données



Félicitations, la surveillance du serveur Linux a démarré !

Zabbix propose des options de visualisation et envoie des alertes en cas de problème. 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 ce processus :

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 ainsi que 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 du processus.

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

Créer un utilisateur MySQL limité

Il est recommandé de créer un utilisateur MySQL dont les permissions sont limitées uniquement à celles nécessaires pour surveiller MySQL.

```
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.

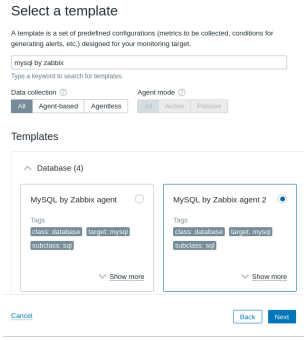
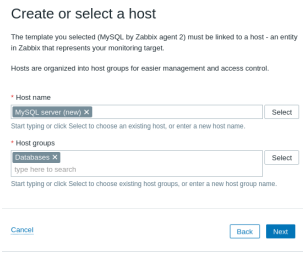
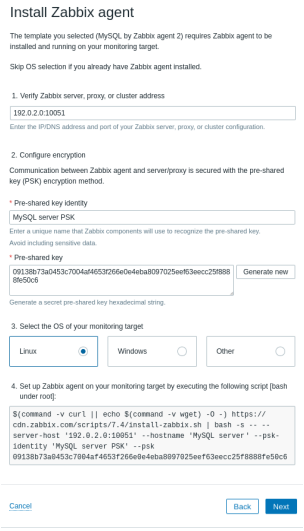
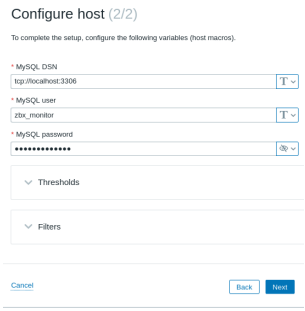
Pour appliquer le modèle, lancez l'assistant Hôte (dans *Collecte de données* > *Hôtes*) :

- **Sélectionnez le modèle** (« MySQL by Zabbix agent 2 »)
- Créez un hôte de serveur MySQL et ajoutez-lui un groupe. Cet hôte virtuel contiendra toutes les données liées à la surveillance du serveur MySQL.
- 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. Elle 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'identité de la clé pré-partagée et la clé pré-partagée pour vous connecter à l'agent.

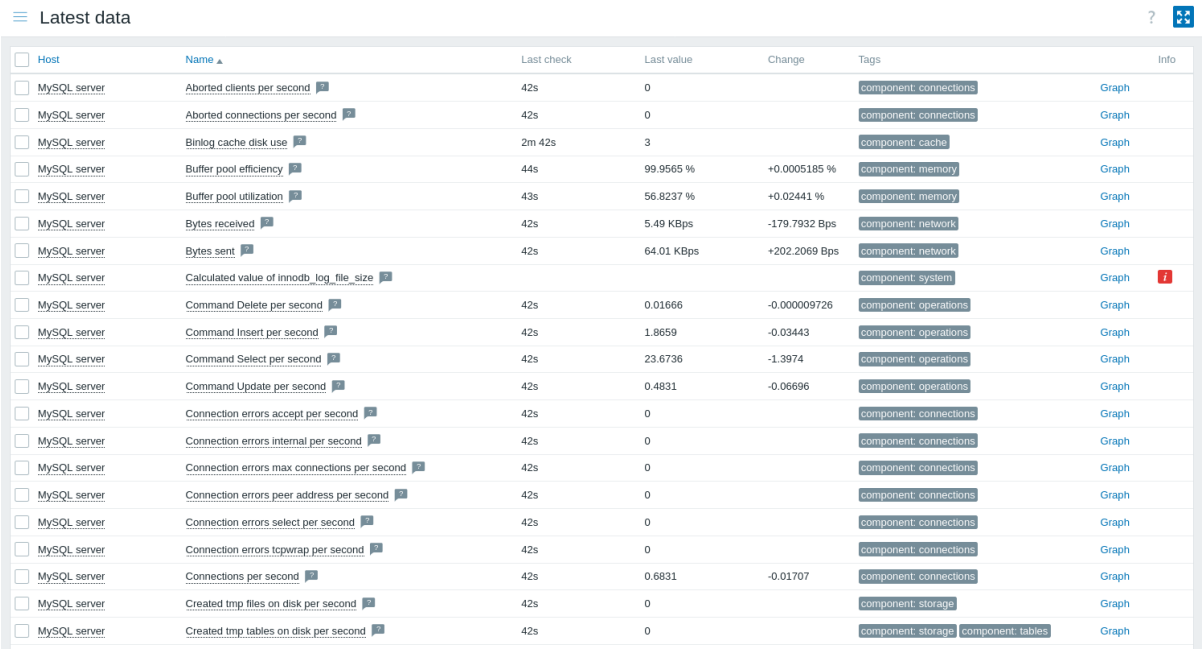
- Terminez la configuration de l'hôte. Il vous sera notamment demandé de saisir le DSN du serveur MySQL, l'utilisateur MySQL et le mot de passe. Indiquez l'utilisateur `zbx_monitor` avec le mot de passe que vous avez créé.

Étape 1	Étape 2	Étape 3	Étape 4
			
Sélection du modèle.	Création de l'hôte/du groupe d'hôtes.	Configuration et installation de l'agent sur l'hôte.	Ajout du DSN, de l'utilisateur et du mot de passe du serveur MySQL.

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

Pour terminer l'assistant Hôte, cliquez sur *Créer* puis sur *Terminer*.

Consulter les dernières données



Host	Name	Last check	Last value	Change	Tags		Info
MySQL server	Aborted clients per second	42s	0		component: connections	Graph	
MySQL server	Aborted connections per second	42s	0		component: connections	Graph	
MySQL server	Binlog cache disk use	2m 42s	3		component: cache	Graph	
MySQL server	Buffer pool efficiency	44s	99.9565 %	+0.0005185 %	component: memory	Graph	
MySQL server	Buffer pool utilization	43s	56.8237 %	+0.02441 %	component: memory	Graph	
MySQL server	Bytes received	42s	5.49 KBps	-179.7932 Bps	component: network	Graph	
MySQL server	Bytes sent	42s	64.01 KBps	+202.2069 Bps	component: network	Graph	
MySQL server	Calculated value of innodb_log_file_size				component: system	Graph	
MySQL server	Command Delete per second	42s	0.01666	-0.000009726	component: operations	Graph	
MySQL server	Command Insert per second	42s	1.8659	-0.03443	component: operations	Graph	
MySQL server	Command Select per second	42s	23.6736	-1.3974	component: operations	Graph	
MySQL server	Command Update per second	42s	0.4831	-0.06696	component: operations	Graph	
MySQL server	Connection errors accept per second	42s	0		component: connections	Graph	
MySQL server	Connection errors internal per second	42s	0		component: connections	Graph	
MySQL server	Connection errors max connections per second	42s	0		component: connections	Graph	
MySQL server	Connection errors peer address per second	42s	0		component: connections	Graph	
MySQL server	Connection errors select per second	42s	0		component: connections	Graph	
MySQL server	Connection errors tcpwrap per second	42s	0		component: connections	Graph	
MySQL server	Connections per second	42s	0.6831	-0.01707	component: connections	Graph	
MySQL server	Created tmp files on disk per second	42s	0		component: storage	Graph	
MySQL server	Created tmp tables on disk per second	42s	0		component: storage component: tables	Graph	

Félicitations, la surveillance du serveur MySQL a démarré !

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

3 Installation : guides supplémentaires

Si vous recherchez la manière la plus simple d'installer Zabbix, veuillez consulter Installation et premiers pas.

Les guides ci-dessous couvrent des méthodes **supplémentaires** ou spécifiques au système 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 du MSI Windows
- Compilation de l'agent Zabbix 2 sous Windows
- Compilation de l'agent Zabbix sous Windows
- Installation de l'agent sur macOS à partir d'un PKG
- Compilation de l'agent Zabbix sur macOS
- Appliance Zabbix

Installation à partir de conteneurs

Aperçu

Cette section décrit comment déployer Zabbix avec Docker ou Docker Compose.

Zabbix fournit officiellement :

- Des images Docker distinctes pour chaque composant Zabbix afin de les exécuter comme des conteneurs portables et autonomes.
- Des fichiers Compose pour définir et exécuter des composants Zabbix multi-conteneurs dans Docker.

Attention:
Depuis Zabbix 6.0, les déclencheurs déterministes doivent être créés pendant l'installation. 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 la variable.
Notez que si elle est exécutée depuis une console, la variable ne sera définie que temporairement et sera supprimée lors du redémarrage d'un Docker. Dans ce cas, laissez votre service SQL en cours d'exécution, arrêtez uniquement le service zabbix-server en exécutant 'docker compose down zabbix-server', puis 'docker compose up -d zabbix-server'.
Vous pouvez également définir cette variable dans le fichier de configuration.

Fichiers source

Les sources des fichiers Docker sont stockées dans le dépôt officiel de Zabbix sur GitHub, où vous pouvez suivre les dernières modifications des fichiers ou bifurquer le projet pour créer vos propres images.

Docker

Zabbix fournit des images basées sur une variété d'images de base de systèmes d'exploitation. Pour obtenir la liste des images de base de systèmes d'exploitation prises en charge pour un composant Zabbix spécifique, consultez la description du composant sur Docker Hub. Toutes les images Zabbix sont configurées pour reconstruire les dernières images si les images de base sont mises à jour.

Installation

Pour obtenir l'image d'un composant Zabbix, exécutez :

```
docker pull zabbix/zabbix-server-mysql
```

Remplacez `zabbix/zabbix-server-mysql` par le nom du dépôt Docker requis.

Cette commande récupérera la dernière version stable du composant Zabbix basée sur le système d'exploitation Alpine Linux. Vous pouvez ajouter des tags au nom du dépôt pour obtenir une image basée sur un autre système d'exploitation ou sur une version majeure ou mineure spécifique de Zabbix.

Les dépôts suivants sont disponibles sur Docker Hub :

Composant	Dépôt Docker
Agent Zabbix	zabbix/zabbix-agent
Serveur Zabbix	
avec prise en charge de MySQL	zabbix/zabbix-server-mysql
avec prise en charge de PostgreSQL	zabbix/zabbix-server-pgsql

Composant		Dépôt Docker
Interface web Zabbix		
	basée sur le serveur web Apache2 avec prise en charge de MySQL	zabbix/zabbix-web-apache-mysql
	basée sur le serveur web Apache2 avec prise en charge de PostgreSQL	zabbix/zabbix-web-apache-pgsql
	basée sur le serveur web Nginx avec prise en charge de MySQL	zabbix/zabbix-web-nginx-mysql
	basée sur le serveur web Nginx avec prise en charge de PostgreSQL	zabbix/zabbix-web-nginx-pgsql
Proxy Zabbix		
	avec prise en charge de SQLite3	zabbix/zabbix-proxy-sqlite3
	avec prise en charge de MySQL	zabbix/zabbix-proxy-mysql
Java gateway Zabbix		zabbix/zabbix-java-gateway

Note:
La prise en charge des traps SNMP est fournie dans un dépôt distinct zabbix/zabbix-snmptraps. Il peut être lié au serveur Zabbix et au proxy Zabbix.

Tags

Les images officielles des composants Zabbix peuvent contenir les tags suivants :

Tag	Description	Exemple
latest	La dernière version stable d'un composant Zabbix basée sur une image Alpine Linux.	zabbix-agent:latest
<OS>-trunk	La dernière build nightly de la version de Zabbix actuellement en cours de développement sur un système d'exploitation spécifique. **<OS>** - le système d'exploitation de base. Valeurs prises en charge : *alpine* - Alpine Linux ; *ltsc2019* - Windows 10 LTSC 2019 (agent uniquement) ; *ol* - Oracle Linux ; *ltsc2022* - Windows 11 LTSC 2022 (agent uniquement) ; *ubuntu* - Ubuntu	zabbix-agent:ubuntu-trunk
<OS>-latest	La dernière version stable d'un composant Zabbix sur un système d'exploitation spécifique. **<OS>** - le système d'exploitation de base. Valeurs prises en charge : *alpine* - Alpine Linux ; *ltsc2019* - Windows 10 LTSC 2019 (agent uniquement) ; *ol* - Oracle Linux ; *ltsc2022* - Windows 11 LTSC 2022 (agent uniquement) ; *ubuntu* - Ubuntu	zabbix-agent:ol-latest

Tag	Description	Exemple
<OS>-X.X-latest	La dernière version mineure d'un composant Zabbix pour une version majeure et un système d'exploitation spécifiques. **<OS>** - le système d'exploitation de base. Valeurs prises en charge : *alpine* - Alpine Linux ; *ltsc2019* - Windows 10 LTSC 2019 (agent uniquement) ; *ol* - Oracle Linux ; *ltsc2022* - Windows 11 LTSC 2022 (agent uniquement) ; *ubuntu* - Ubuntu **X.X** - la version majeure de Zabbix (c.-à-d. *6.0*, *7.4*, *8.0*).	zabbix-agent:alpine-8.0-latest
<OS>-X.X.*	La version mineure spécifique d'un composant Zabbix pour une version majeure et un système d'exploitation spécifiques. **<OS>** - le système d'exploitation de base. Valeurs prises en charge : *alpine* - Alpine Linux ; *ltsc2019* - Windows 10 LTSC 2019 (agent uniquement) ; *ol* - Oracle Linux ; *ltsc2022* - Windows 11 LTSC 2022 (agent uniquement) ; *ubuntu* - Ubuntu **X.X** - la version majeure de Zabbix (c.-à-d. *6.0*, *7.4*, *8.0*). * - la version mineure de Zabbix	zabbix-agent:alpine-8.0.1

Configuration initiale

Après avoir téléchargé les images, démarrez les conteneurs en exécutant la commande `docker run`, suivie d'arguments supplémentaires pour spécifier les variables d'environnement et/ou les points de montage requis. Quelques exemples de configuration sont fournis ci-dessous.

Note:
Pour activer la communication entre les composants Zabbix, certains ports, tels que 10051/TCP pour le serveur Zabbix (trapper), 10050/TCP pour l'agent Zabbix, 162/UDP pour les traps SNMP et 80/TCP pour l'interface web Zabbix, seront exposés à la machine hôte. La liste complète des ports par défaut utilisés par les composants Zabbix est disponible sur la page Requirements. Pour le serveur et l'agent Zabbix, le port par défaut peut être modifié en définissant la variable d'environnement ZBX_LISTENPORT environment variable.

Variables d'environnement

Toutes les images des composants Zabbix fournissent des variables d'environnement pour contrôler la configuration. Les variables d'environnement prises en charge sont répertoriées dans le dépôt du composant.

Ces variables d'environnement correspondent à des options des fichiers de configuration Zabbix, mais avec une méthode de nommage différente. Par exemple, `ZBX_LOGSLOWQUERIES` est équivalent à `LogSlowQueries` dans les fichiers de configuration du server Zabbix ou du proxy Zabbix.

Attention:
Certaines options de configuration (par exemple, `PIDFile` et `LogType`) ne peuvent pas être modifiées.

Les variables d'environnement suivantes sont spécifiques aux composants Docker et n'existent pas dans les fichiers de configuration Zabbix :

Variable	Composants	Valeur par défaut	Description
`DB_SERVER_HOST`	Server Proxy Interface web	`mysql-server` pour MYSQL `postgres-server` pour PostgreSQL	Nom IP ou DNS du serveur MySQL ou PostgreSQL.
`DB_SERVER_PORT`	Server Proxy Interface web	3306 pour MYSQL 5432 pour PostgreSQL	Port du serveur MySQL ou PostgreSQL.

Variable	Composants	Valeur par défaut	Description
MYSQL_USER	Server Proxy Web-interface	zabbix	Utilisateur de la base de données MySQL.
MYSQL_PASSWORD	Server Proxy Interface web	zabbix	Mot de passe de la base de données MySQL.
MYSQL_DATABASE	Server Proxy Interface web	zabbix pour le serveur Zabbix zabbix_proxy pour le proxy Zabbix	Nom de la base de données Zabbix.
POSTGRES_USER	Server Interface web	zabbix	Utilisateur de la base de données PostgreSQL.
POSTGRES_PASSWORD	Server Interface web	zabbix	Mot de passe de la base de données PostgreSQL.
POSTGRES_DB	Server Interface web	zabbix pour le serveur Zabbix zabbix_proxy pour le proxy Zabbix	Nom de la base de données Zabbix.
PHP_TZ	Web-interface	Europe/Riga	Fuseau horaire au format PHP. La liste complète des fuseaux horaires pris en charge est disponible sur php.net.
ZBX_SERVER_NAME	Interface web	Zabbix Docker	Nom visible de l'installation Zabbix sous le logo Zabbix dans le menu vertical de l'interface web.
ZBX_JAVAGATEWAY_ENABLE	Server Proxy	false	Active la communication avec le Java gateway Zabbix pour collecter les contrôles liés à Java.
ZBX_ENABLE_SNMP_TRAPS	Server Proxy	false	Active la fonctionnalité des traps SNMP. Elle nécessite une instance **zabbix-snmptraps** et le volume partagé */var/lib/zabbix/snmptraps* vers le serveur Zabbix ou le proxy Zabbix.

Volumes

Les images permettent de monter des volumes en utilisant les points de montage suivants :

Volume	Description
Zabbix agent	
/etc/zabbix/zabbix_agentd.d	Le volume permet d'inclure des fichiers **.conf* et d'étendre Zabbix agent à l'aide de la fonctionnalité UserParameter
/var/lib/zabbix/modules	Le volume permet de charger des modules supplémentaires et d'étendre Zabbix agent à l'aide de la fonctionnalité LoadModule
/var/lib/zabbix/enc	Le volume est utilisé pour stocker les fichiers liés à TLS. Les noms de ces fichiers sont spécifiés à l'aide des variables d'environnement ZBX_TLSCAFILE, ZBX_TLSCRLFILE, ZBX_TLSKEY_FILE et ZBX_TLSPSKFILE
Zabbix server	
/usr/lib/zabbix/alertscripts	Le volume est utilisé pour les scripts d'alerte personnalisés. Il correspond au paramètre AlertScriptsPath dans zabbix_server.conf
/usr/lib/zabbix/externalscripts	Le volume est utilisé par les vérifications externes. Il correspond au paramètre ExternalScripts dans zabbix_server.conf
/var/lib/zabbix/modules	Le volume permet de charger des modules supplémentaires et d'étendre Zabbix server à l'aide de la fonctionnalité LoadModule
/var/lib/zabbix/enc	Le volume est utilisé pour stocker les fichiers liés à TLS. Les noms de ces fichiers sont spécifiés à l'aide des variables d'environnement ZBX_TLSCAFILE, ZBX_TLSCRLFILE, ZBX_TLSKEY_FILE et ZBX_TLSPSKFILE
/var/lib/zabbix/ssl/certs	Le volume est utilisé comme emplacement des fichiers de certificat client SSL pour l'authentification du client. Il correspond au paramètre SSLCertLocation dans zabbix_server.conf

Volume	Description
var/lib/zabbix/ssl/keys	Le volume est utilisé comme emplacement des fichiers de clé privée SSL pour l'authentification du client. Il correspond au paramètre `SSLKeyLocation` dans zabbix_server.conf
var/lib/zabbix/ssl/ssl_ca	Le volume est utilisé comme emplacement des fichiers d'autorité de certification (CA) pour la vérification du certificat SSL du serveur. Il correspond au paramètre `SSLCALocation` dans zabbix_server.conf
var/lib/zabbix/snmptraps	Le volume est utilisé comme emplacement du fichier snmptraps.log. Il peut être partagé par le conteneur zabbix-snmptraps et hérité à l'aide de l'option Docker volumes_from lors de la création d'une nouvelle instance de Zabbix server. La fonctionnalité de traitement des traps SNMP peut être activée en utilisant un volume partagé et en définissant la variable d'environnement `ZBX_ENABLE_SNMP_TRAPS` sur 'true'
var/lib/zabbix/mibs	Le volume permet d'ajouter de nouveaux fichiers MIB. Les sous-répertoires ne sont pas pris en charge, tous les MIB doivent être placés dans `/var/lib/zabbix/mibs`
Zabbix proxy	
usr/lib/zabbix/externalscripts	Le volume est utilisé par les vérifications externes. Il correspond au paramètre `ExternalScripts` dans zabbix_proxy.conf
var/lib/zabbix/db_data	Le volume permet de stocker les fichiers de base de données sur des périphériques externes. Pris en charge uniquement pour Zabbix proxy avec SQLite3
var/lib/zabbix/modules	Le volume permet de charger des modules supplémentaires et d'étendre Zabbix server à l'aide de la fonctionnalité LoadModule
var/lib/zabbix/enc	Le volume est utilisé pour stocker les fichiers liés à TLS. Les noms de ces fichiers sont spécifiés à l'aide des variables d'environnement `ZBX_TLSCAFILE`, `ZBX_TLSCRLFILE`, `ZBX_TLSKEY_FILE` et `ZBX_TLSPSKFILE`
var/lib/zabbix/ssl/certs	Le volume est utilisé comme emplacement des fichiers de certificat client SSL pour l'authentification du client. Il correspond au paramètre `SSLCertLocation` dans zabbix_proxy.conf
var/lib/zabbix/ssl/keys	Le volume est utilisé comme emplacement des fichiers de clé privée SSL pour l'authentification du client. Il correspond au paramètre `SSLKeyLocation` dans zabbix_proxy.conf
var/lib/zabbix/ssl/ssl_ca	Le volume est utilisé comme emplacement des fichiers d'autorité de certification (CA) pour la vérification du certificat SSL du serveur. Il correspond au paramètre `SSLCALocation` dans zabbix_proxy.conf
var/lib/zabbix/snmptraps	Le volume est utilisé comme emplacement du fichier snmptraps.log. Il peut être partagé par le conteneur zabbix-snmptraps et hérité à l'aide de l'option Docker volumes_from lors de la création d'une nouvelle instance de Zabbix server. La fonctionnalité de traitement des traps SNMP peut être activée en utilisant un volume partagé et en définissant la variable d'environnement `ZBX_ENABLE_SNMP_TRAPS` sur 'true'
var/lib/zabbix/mibs	Le volume permet d'ajouter de nouveaux fichiers MIB. Les sous-répertoires ne sont pas pris en charge, tous les MIB doivent être placés dans `/var/lib/zabbix/mibs`
Interface web Zabbix basée sur le serveur web Apache2	
/etc/ssl/apache2	Le volume permet d'activer HTTPS pour l'interface web Zabbix. Le volume doit contenir les deux fichiers `ssl.crt` et `ssl.key` préparés pour les connexions SSL Apache2
Interface web Zabbix basée sur le serveur web Nginx	

Volume	Description
/etc/ssl/nginx	Le volume permet d'activer HTTPS pour l'interface web Zabbix. Le volume doit contenir les deux fichiers `ssl.crt`, `ssl.key` et `dhparam.pem` préparés pour les connexions SSL Nginx
Zabbix sn-mptraps	
/var/lib/zabbix/snmptraps	Le volume contient le fichier journal `snmptraps.log` nommé avec les traps SNMP reçus
/var/lib/zabbix/mibs	Le volume permet d'ajouter de nouveaux fichiers MIB. Les sous-répertoires ne sont pas pris en charge, tous les MIB doivent être placés dans `/var/lib/zabbix/mibs`

Pour plus d'informations, consultez les dépôts officiels Zabbix dans Docker Hub.

Exemples

Exemple 1

Cet exemple montre comment exécuter le serveur Zabbix avec la prise en charge de la base de données MySQL, l'interface web Zabbix basée sur le serveur web Nginx et le Java gateway Zabbix.

1. Créez un réseau dédié aux conteneurs des composants Zabbix :

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

2. Démarrez une instance vide du serveur MySQL :

```
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=utf8 --collation-server=utf8_bin \
    --default-authentication-plugin=caching_sha2_password
```

3. Démarrez une instance du 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
```

4. Démarrez une instance du serveur Zabbix et liez-la à l'instance du serveur MySQL créée :

```
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 MYSQL_ROOT_PASSWORD="root_pwd" \
    -e ZBX_JAVAGATEWAY="zabbix-java-gateway" \
    --network=zabbix-net \
    -p 10051:10051 \
    --restart unless-stopped \
    -d zabbix/zabbix-server-mysql:alpine-8.0-latest
```

5. Démarrez l'interface web Zabbix et liez-la aux instances du serveur MySQL et du serveur Zabbix créées :

```
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 MYSQL_ROOT_PASSWORD="root_pwd" \
    --network=zabbix-net \
```

```
    -p 80:8080 \
    --restart unless-stopped \
    -d zabbix/zabbix-web-nginx-mysql:alpine-8.0-latest
```

Exemple 2

Cet exemple montre comment exécuter le serveur Zabbix avec la prise en charge de la base de données PostgreSQL, l'interface web Zabbix basée sur le serveur web Nginx et la fonctionnalité de traps SNMP.

1. Créez un réseau dédié aux conteneurs des composants Zabbix :

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

2. Démarrez une instance vide du serveur PostgreSQL :

```
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. Démarrez une instance Zabbix snmptraps :

```
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. Démarrez une instance du serveur Zabbix et liez-la à l'instance du serveur PostgreSQL créée :

```
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 \
    -d zabbix/zabbix-server-pgsql:alpine-8.0-latest
```

5. Démarrez l'interface web Zabbix et liez-la aux instances du serveur PostgreSQL et du serveur Zabbix créées :

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

Exemple 3

Cet exemple montre comment exécuter le serveur Zabbix avec la prise en charge de la base de données MySQL, l'interface web Zabbix basée sur le serveur web Nginx et le Java gateway Zabbix en utilisant `podman` sur Red Hat 8.

1. Créez un nouveau pod nommé `zabbix` avec les ports exposés (interface web, trapper du serveur Zabbix) :

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

2. (facultatif) Démarrez le conteneur de l'agent Zabbix à l'emplacement du 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-74:latest
```

3. Créez le répertoire `./mysql/` sur l'hôte et démarrez Oracle MySQL server 8.4 :

```
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=utf8 --collation-server=utf8_bin \
    --default-authentication-plugin=caching_sha2_password
```

4. Démarrez le conteneur du serveur Zabbix :

```
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 MYSQL_ROOT_PASSWORD="root_pwd" \
                -e ZBX_JAVAGATEWAY="127.0.0.1" \
                --restart=always \
                --pod=zabbix \
                -d registry.connect.redhat.com/zabbix/zabbix-server-mysql-74
```

5. Démarrez le conteneur Java Gateway Zabbix :

```
podman run --name zabbix-java-gateway -t \
    --restart=always \
    --pod=zabbix \
    -d registry.connect.redhat.com/zabbix/zabbix-java-gateway-74
```

6. Démarrez le conteneur de l'interface web Zabbix :

```
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 MYSQL_ROOT_PASSWORD="root_pwd" \
                --restart=always \
                --pod=zabbix \
                -d registry.connect.redhat.com/zabbix/zabbix-web-mysql-74
```

> **Note:**
> Le pod `zabbix` expose le port 80/TCP (HTTP) à la machine hôte depuis le port 8080/TCP du conteneur `zabbix-web-mysql`.

Docker Compose

Il est également possible d'installer Zabbix à l'aide du plugin Docker Compose. Les fichiers Compose permettant de définir et d'exécuter des composants Zabbix multi-conteneurs sont disponibles dans le dépôt Docker officiel de Zabbix sur GitHub.

Attention:
Les fichiers Compose officiels de Zabbix prennent en charge la version 3 de Docker Compose.

Ces fichiers Compose sont fournis à titre d'exemple ; ils sont surchargés. Par exemple, ils contiennent des proxys avec prise en charge à la fois de MySQL et de SQLite3.

Pour obtenir les fichiers Docker Compose fournis par Zabbix, clonez le dépôt :

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

Basculez vers la version requise :

```
git checkout 8.0
```

Configurez les fichiers Compose, puis créez et démarrez les conteneurs :

```
docker compose -f ./docker-compose_v3_alpine_mysql_latest.yaml up
```

Remplacez `docker-compose_v3_alpine_mysql_latest.yaml` dans la commande ci-dessus par le fichier de configuration requis.

Les options suivantes sont disponibles :

File name	Description
`docker-compose_v3_alpine_mysql_latest.yaml`	Les fichiers Compose exécutent la dernière version des composants Zabbix 8.0 sur Alpine Linux avec prise en charge de la base de données MySQL.
`docker-compose_v3_alpine_mysql_local.yaml`	Les fichiers Compose construit localement la dernière version de Zabbix 8.0 et exécute les composants Zabbix sur Alpine Linux avec prise en charge de la base de données MySQL.
`docker-compose_v3_alpine_pgsql_latest.yaml`	Les fichiers Compose exécutent la dernière version des composants Zabbix 8.0 sur Alpine Linux avec prise en charge de la base de données PostgreSQL.
`docker-compose_v3_alpine_pgsql_local.yaml`	Les fichiers Compose construit localement la dernière version de Zabbix 8.0 et exécute les composants Zabbix sur Alpine Linux avec prise en charge de la base de données PostgreSQL.
`docker-compose_v3_ol_mysql_latest.yaml`	Les fichiers Compose exécute la dernière version des composants Zabbix 8.0 sur Oracle Linux avec prise en charge de la base de données MySQL.
`docker-compose_v3_ol_mysql_local.yaml`	Les fichiers Compose construit localement la dernière version de Zabbix 8.0 et exécute les composants Zabbix sur Oracle Linux avec prise en charge de la base de données MySQL.
`docker-compose_v3_ol_pgsql_latest.yaml`	Les fichiers Compose exécute la dernière version des composants Zabbix 8.0 sur Oracle Linux avec prise en charge de la base de données PostgreSQL.
`docker-compose_v3_ol_pgsql_local.yaml`	Les fichiers Compose construit localement la dernière version de Zabbix 8.0 et exécute les composants Zabbix sur Oracle Linux avec prise en charge de la base de données PostgreSQL.
`docker-compose_v3_ubuntu_mysql_latest.yaml`	Les fichiers Compose exécutent la dernière version des composants Zabbix 8.0 sur Ubuntu 24.04 (noble) avec prise en charge de la base de données MySQL.
`docker-compose_v3_ubuntu_mysql_local.yaml`	Les fichiers Compose construit localement la dernière version de Zabbix 8.0 et exécute les composants Zabbix sur Ubuntu 24.04 (noble) avec prise en charge de la base de données MySQL.
`docker-compose_v3_ubuntu_pgsql_latest.yaml`	Les fichiers Compose exécutent la dernière version des composants Zabbix 8.0 sur Ubuntu 24.04 (noble) avec prise en charge de la base de données PostgreSQL.
`docker-compose_v3_ubuntu_pgsql_local.yaml`	Les fichiers Compose construit localement la dernière version de Zabbix 8.0 et exécute les composants Zabbix sur Ubuntu 24.04 (noble) avec prise en charge de la base de données PostgreSQL.

Stockage

Les fichiers Compose sont configurés pour prendre en charge le stockage local sur une machine hôte. Docker Compose créera un répertoire `zbx_env` dans le dossier contenant le fichier compose lorsque vous exécuterez les composants Zabbix à l'aide du fichier compose. Le répertoire contiendra la même structure que celle décrite dans la section Volumes, ainsi qu'un répertoire pour le stockage de la base de données.

Il existe également des volumes en mode lecture seule pour les fichiers `/etc/localtime` et `/etc/timezone`.

Variables d'environnement

Les fichiers de variables ont la structure de nommage suivante : `.env_<type de composant>` et sont situés dans le répertoire *env_vars*. Consultez les variables d'environnement pour plus de détails sur le nommage des variables et la sélection disponible.

Exemples

Exemple 1

```
git checkout 8.0
docker compose -f ./docker-compose_v3_alpine_mysql_latest.yaml up -d
```

La commande téléchargera les dernières images Zabbix 8.0 pour chaque composant Zabbix et les exécutera en mode détaché.

> **Attention:**
> N'oubliez pas de télécharger les fichiers `.env_<type of component>` depuis le dépôt officiel Zabbix sur github.com avec les fichiers compose.

Exemple 2

```
git checkout 8.0
docker compose -f ./docker-compose_v3_ubuntu_mysql_local.yaml up -d
```

La commande téléchargera l'image de base Ubuntu 24.04 (noble), puis construira localement les composants Zabbix 8.0 et les exécutera en mode détaché.

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 vous installez depuis le dépôt Git Zabbix, il est nécessaire d'exécuter d'abord :
>
> ```
> ./bootstrap.sh
> ```

Lors de la configuration des sources pour un serveur ou un proxy Zabbix, vous devez spécifier le type de base de données à utiliser. Un seul type de base de données peut être compilé avec un processus de serveur ou de proxy à la fois.

Pour voir toutes les options de configuration prises en charge, exécutez, dans le répertoire des sources Zabbix extraites :

```
./configure --help
```

Pour configurer les sources pour un serveur Zabbix et un agent, vous pouvez exécuter une commande comme :

```
./configure --enable-server --enable-agent --with-mysql --enable-ipv6 --with-net-snmp --with-libcurl --wit
```

Pour configurer les sources pour un serveur Zabbix (avec PostgreSQL, etc.), vous pouvez exécuter :

```
./configure --enable-server --with-postgresql --with-net-snmp
```

Pour configurer les sources pour 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 pour 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 supervision des machines virtuelles, l'authentification SMTP et les éléments Zabbix agent `web.page.*`. Voir aussi : Prérequis (libcurl).
- `--with-libxml2` - requis pour la supervision des machines virtuelles.

- `--with-libpcre2[=DIR]` - Zabbix compile toujours avec la bibliothèque PCRE2 ; cette option permet uniquement de spécifier un chemin d'installation PCRE2 personnalisé.
- `--with-mysql=/path/to/mysql_config` - spécifie le chemin vers une configuration particulière de bibliothèque cliente MySQL. Utile lorsque plusieurs versions de MySQL ou MariaDB sont installées.
- `--enable-static` - lie les bibliothèques statiquement (non pris en charge sur Solaris). Utilisez cette option si vous prévoyez de distribuer des binaires compilés sur des systèmes ne disposant pas des bibliothèques requises. Non recommandé lors de la compilation du serveur Zabbix. Pour compiler le serveur statiquement, une version statique de chaque bibliothèque externe est requise. Le script de configuration ne vérifie pas cela automatiquement.
- `--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 avec de faibles limites par défaut de pile de thread).

Attention:
Si `./configure` échoue en raison de bibliothèques manquantes ou d'autres problèmes, veuillez consulter 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 aussi :

- Compiler Zabbix avec la prise en charge du chiffrement
- Problèmes de compilation connus

5 Compiler et tout installer

Note:
Si vous installez 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 des démons (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 des démons seront installés dans <prefix>/sbin, tandis que les utilitaires le seront dans <prefix>/bin. Les pages de manuel seront installées dans <prefix>/share.

6 Vérifier et modifier les fichiers de configuration

- modifiez le fichier de configuration de l'agent Zabbix **/usr/local/etc/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.

- modifiez le fichier de configuration du serveur Zabbix **/usr/local/etc/zabbix_server.conf**

Vous devez spécifier le nom de la base de données, l'utilisateur et le mot de passe (le cas échéant).

Le reste des paramètres conviendra avec leurs valeurs par défaut si vous avez une petite installation (jusqu'à dix hôtes supervisés). Vous devriez toutefois modifier les paramètres par défaut si vous souhaitez maximiser les performances du serveur Zabbix (ou du proxy).

- si vous avez installé un proxy Zabbix, modifiez le fichier de configuration du proxy **/usr/local/etc/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 démons

Exécutez zabbix_server 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 web de Zabbix est écrite en PHP ; un serveur web prenant en charge PHP est donc nécessaire pour l'exécuter. L'installation consiste simplement à copier 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 web de 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 pour l'interface web 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 superviser des cibles non couvertes par les plugins intégrés (par exemple, des serveurs ou clusters MongoDB, PostgreSQL et ses forks, 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 et extraire l'archive des sources du plugin chargeable.

Pour compiler le plugin, accédez au répertoire du plugin extrait et exécutez `make` :

```
make
```

> **Note:**
> Une version de Go prise en charge 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
```

Consultez 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 cible de supervision réelle, 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` afin de demander un chemin d'installation autre que le chemin par défaut /usr/local, car l'installation de Java gateway créera toute une arborescence de répertoires, et pas seulement 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.

```
make
```

Vous disposez maintenant d'un fichier zabbix-java-gateway-$VERSION.jar dans src/zabbix_java/bin. Si l'exécution de Java gateway depuis src/zabbix_java dans le répertoire de distribution vous convient, vous pouvez alors passer aux instructions de configuration et d'exécution de Java gateway. Sinon, assurez-vous de disposer de privilèges suffisants et exécutez `make install`.

```
make install
```

Passez à la section 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 requise que si vous souhaitez utiliser les rapports planifiés.

Pour effectuer l'installation à partir des sources, commencez par télécharger puis extraire l'archive des sources.

Pour compiler le service web Zabbix, exécutez le script `./configure` avec l'option `--enable-webservice`.

> **Note:**
> Une version de 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
```

Passez à la configuration pour plus de détails sur la configuration de la génération des rapports planifiés.

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 petite sur certains systèmes

Si Zabbix plante ou se fige en raison de débordements de pile, vous pouvez augmenter la taille de pile par thread à l'aide de l'option `--with-stacksize` lors de la configuration des sources. Ce problème peut survenir sur les systèmes dont les limites par défaut de pile de thread sont faibles, en particulier pendant le prétraitement, où plusieurs threads sont créés.

L'exemple suivant définit la taille de 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

Aperçu

Les paquets officiels en **préversion** de Zabbix 8.0 sont disponibles sur le site web de Zabbix.

Sélectionnez votre système d'exploitation et le composant Zabbix afin de générer des instructions d'installation adaptées à votre environnement. Consultez également les notes d'installation des paquets sur cette page pour obtenir des informations complémentaires importantes, ainsi que la section Installation and setup.

Des paquets sont disponibles pour les distributions Linux suivantes :

- Red Hat Enterprise Linux et ses dérivés : 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 (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 avez précédemment installé Zabbix à partir du dépôt de votre système d'exploitation, consultez les étapes de mise à niveau des paquets Zabbix depuis les 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 si les deux sont installés sur le même hôte.

Si nécessaire, des paquets distincts pour Zabbix agent/agent 2, Zabbix get et Zabbix sender sont disponibles dans le dépôt officiel Zabbix.

Zabbix fournit également des binaires précompilés de Zabbix agent pour les systèmes d'exploitation non Linux ; voir :

- Installation de l'agent Windows à partir d'un MSI
- Installation de l'agent macOS à partir d'un PKG
- 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 server/proxy Zabbix fait que PostgreSQL utilise un socket réseau au lieu d'un socket UNIX local ; voir la configuration SELinux pour les instructions de configuration associées.
- Si vous utilisez TimescaleDB, consultez la section supplémentaire sur la configuration de TimescaleDB.
- Si vous installez le Java gateway Zabbix (pour la supervision des applications JMX), consultez également les instructions de configuration supplémentaires pour les systèmes basés sur RHEL et les systèmes basés sur Debian.
- Pour exécuter l'agent Zabbix en tant que root, voir Exécuter l'agent en tant que root.

Les remarques suivantes s'appliquent à RHEL et à ses dérivés :

- Si vous avez activé le dépôt EPEL pour EL9, qui fournit également des paquets Zabbix, il doit être exclu de la résolution des paquets avant d'installer les paquets Zabbix officiels ; voir Installation accidentelle des paquets Zabbix EPEL.
- Pour installer des paquets Zabbix dans des environnements Red Hat UBI, voir Paquets Zabbix pour RHEL dans les environnements Red Hat UBI.
- Pour utiliser les éléments ICMP ping, des paquets pour `fping` sont également disponibles dans le dépôt officiel Zabbix.

Configuration de SELinux

Zabbix utilise une communication inter-processus basée sur des sockets. Sur les systèmes où Security-Enhanced Linux (SELinux) est activé, il peut être nécessaire d'ajouter des règles SELinux pour permettre à Zabbix de créer/utiliser des sockets de domaine UNIX dans le répertoire SocketDir. Les fichiers de 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.

Si SELinux est activé en mode enforcing, exécutez les commandes suivantes pour autoriser la communication entre le frontend 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 le frontend Zabbix à se connecter à la base de données :

```
setsebool -P httpd_can_network_connect_db on
```

Pour les versions de RHEL antérieures à 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 paquet `zabbix-selinux-policy` prédéfini depuis le dépôt officiel de Zabbix. Ce paquet est fourni pour toutes les versions de systèmes d'exploitation 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.

> **Attention:**
> Pour une sécurité maximale, il est recommandé de définir des paramètres SELinux personnalisés.

Le paquet `zabbix-selinux-policy` contient une politique SELinux de base, permettant à Zabbix de créer et d'utiliser des sockets et d'autoriser la connexion de HTTPd à PostgreSQL (utilisé par le frontend).

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 nécessaires pour une installation ou un fonctionnement normal, mais sont utiles pour le dépannage avancé.

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/stable/sles/15/x86_64/debuginfo/
gpgcheck=1
gpgkey=https://repo.zabbix.com/zabbix/8.0/stable/sles/15/x86_64/debuginfo/repodata/repomd.xml.key
```

```
enabled=0
update=1
```

Une fois activé, installez les paquets :

- Sur RHEL, installez un seul paquet avec les informations de débogage pour tous les composants Zabbix :

```
dnf install zabbix-debuginfo
```

- Sur SUSE, installez les paquets debuginfo spécifiques aux composants :

```
zypper install zabbix-<component>-debuginfo
```

Activation des dépôts des versions instables

Les instructions ci-dessous permettent d'activer les dépôts des versions instables de Zabbix (désactivés par défaut), utilisés pour les versions candidates de publications mineures de Zabbix.

Commencez par installer ou mettre à jour vers la dernière version du paquet zabbix-release. 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.

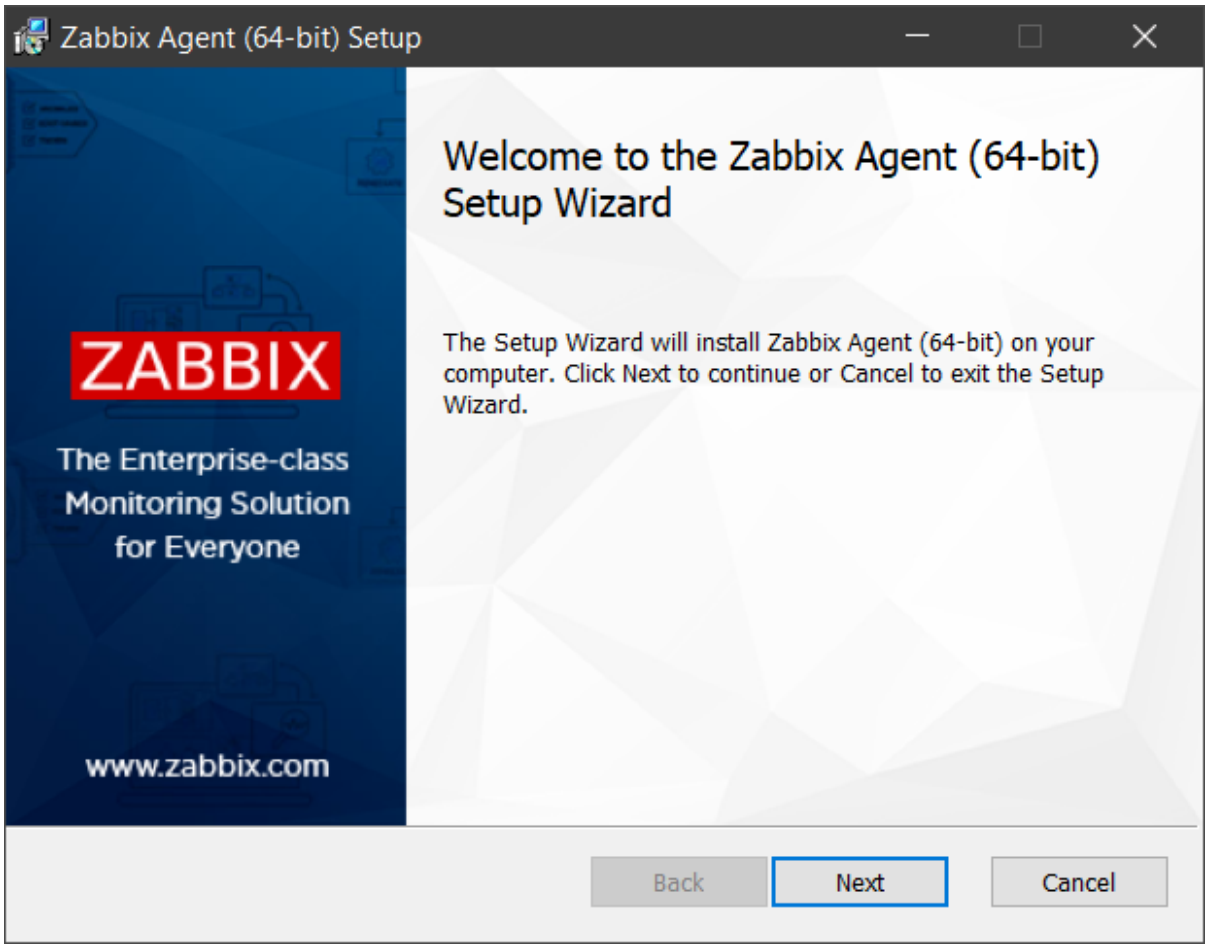
Bien que l'installation à l'aide de packages MSI soit entièrement prise en charge, il est recommandé d'installer au minimum Microsoft .NET Framework 2 pour une gestion correcte des erreurs.

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 à partir de l'assistant d'installation

Les étapes d'installation suivantes s'appliquent à la fois à Zabbix agent et à Zabbix agent 2.

1. Double-cliquez sur le fichier MSI téléchargé pour démarrer l'installation :

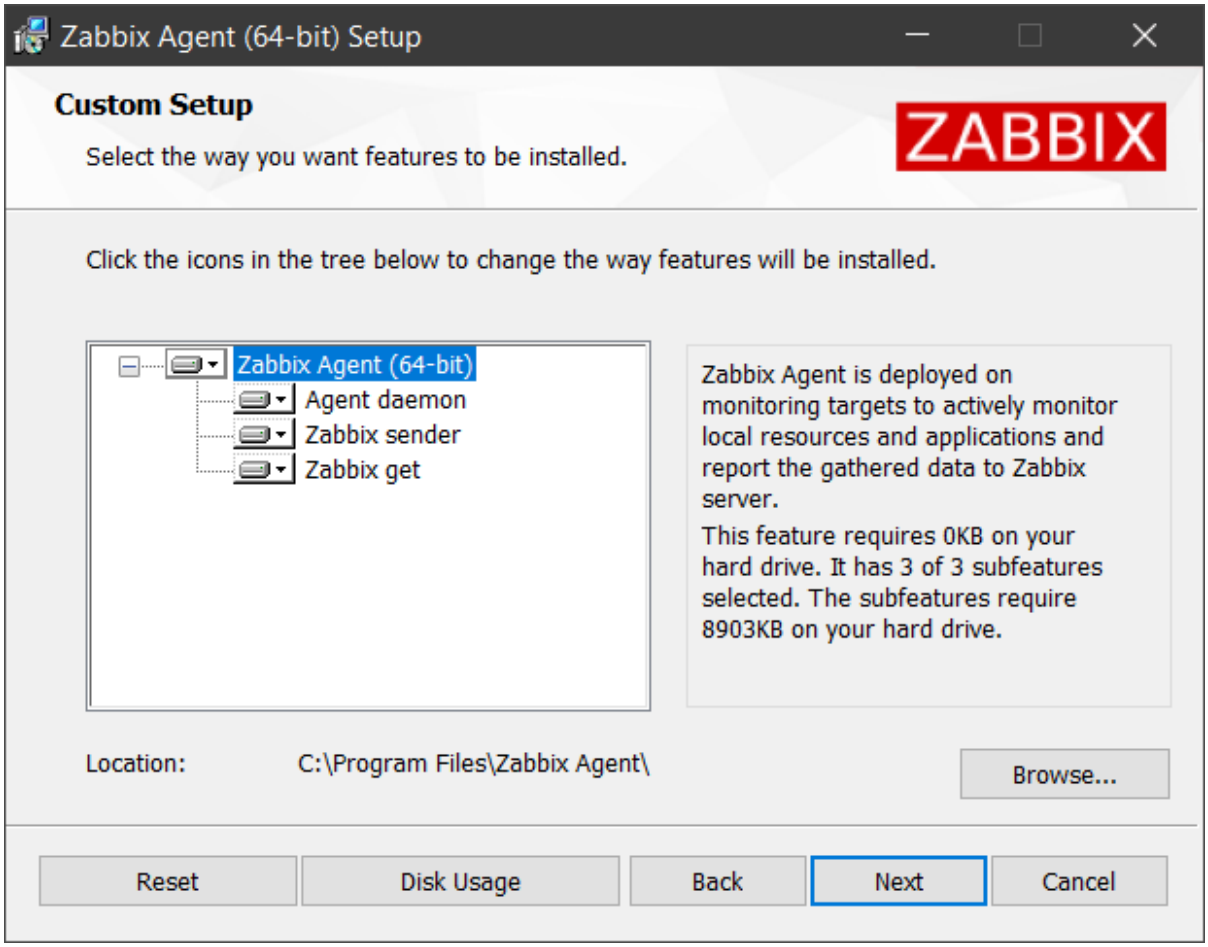


2. Acceptez le contrat de licence utilisateur final :



3. Sélectionnez les composants Zabbix (Agent daemon, Zabbix sender, Zabbix get) à installer :

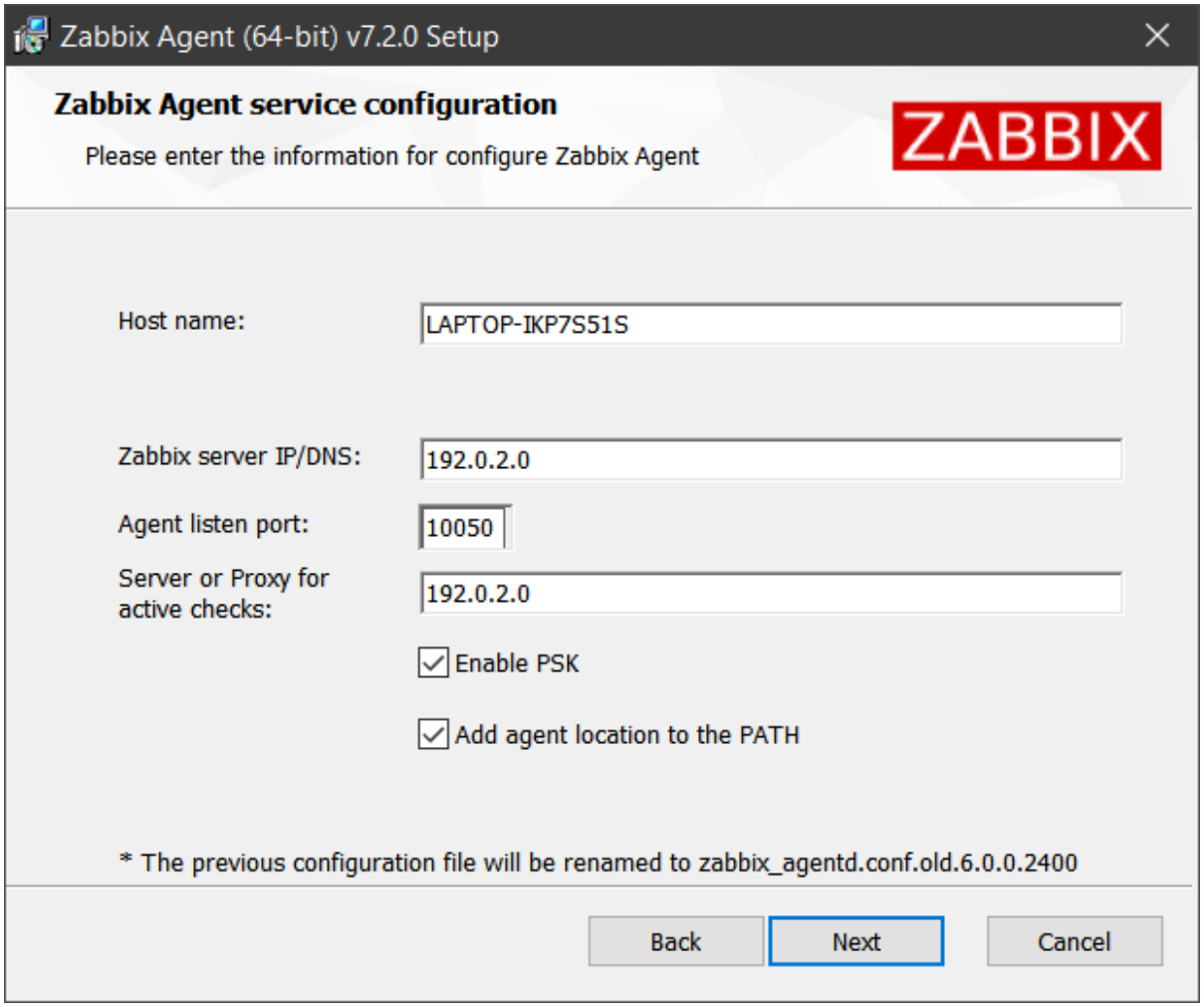
Attention:
Il est recommandé d'utiliser l'emplacement d'installation par défaut fourni par le programme d'installation. L'utilisation d'un emplacement personnalisé sans les autorisations nécessaires peut compromettre la sécurité de l'installation.



4. Configurez les paramètres suivants. Leurs valeurs seront définies dans le fichier de configuration de Zabbix agent :

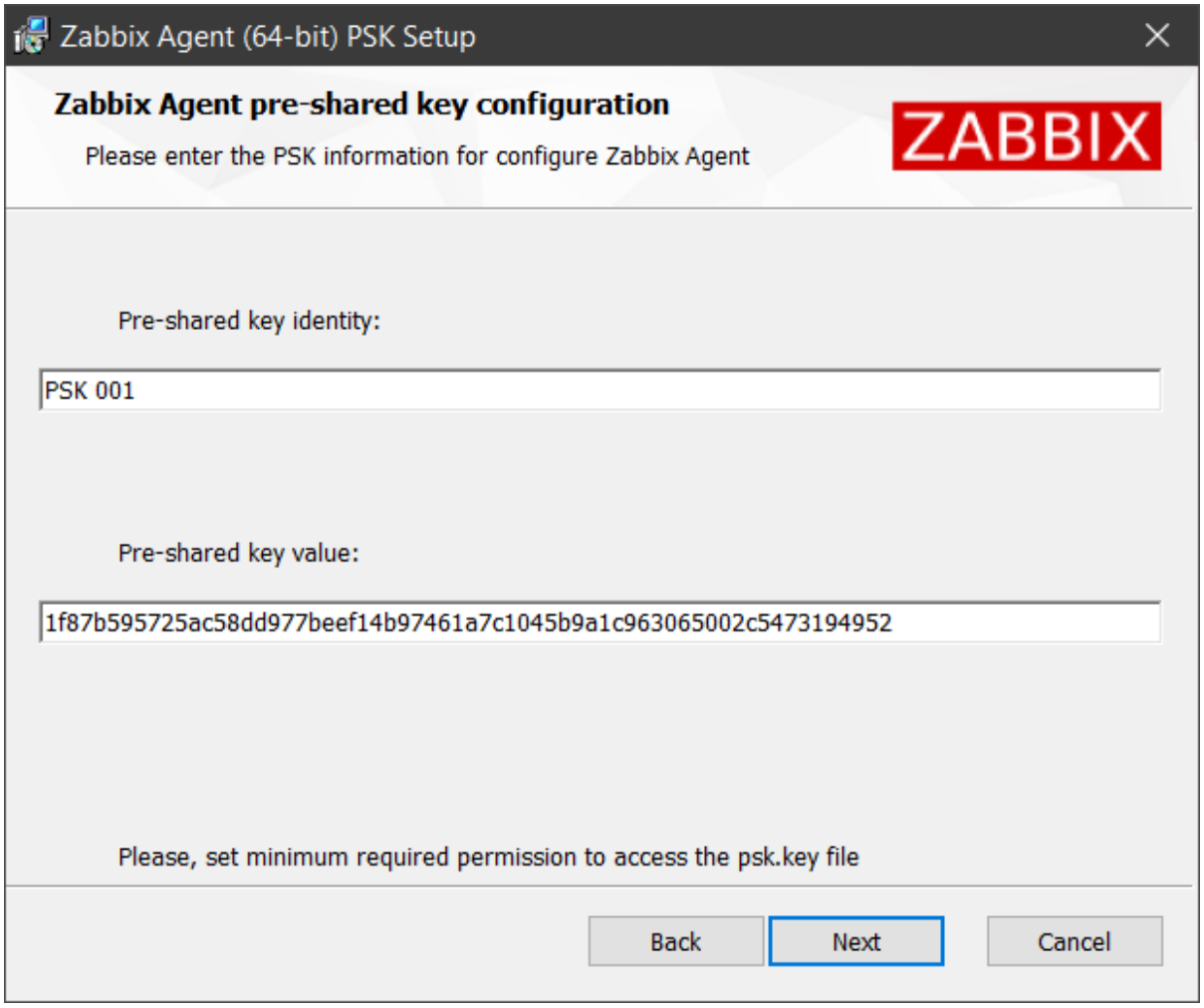
Paramètre	Description
Host name	Le nom d'hôte de la machine sur laquelle Zabbix agent est installé. Définit le paramètre Hostname.
Zabbix server IP/DNS	Une liste d'adresses IP séparées par des virgules, éventuellement en notation CIDR, ou de noms DNS de serveurs Zabbix ou de proxies Zabbix. Ce paramètre est **obligatoire**. Définit le paramètre Server.
Agent listen port	L'agent écoutera sur ce port les connexions provenant du serveur. Définit le paramètre ListenPort.
Server or Proxy for active checks	L'adresse du serveur/proxy Zabbix ou la configuration du cluster à partir de laquelle obtenir les active checks. 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.
Enable PSK	Cochez cette case pour activer la prise en charge de TLS à l'aide de clés pré-partagées. Définit les paramètres TLSConnect et TLSAccept sur `psk`.
Add agent location to the PATH	Cochez cette 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 *Enable PSK* à l'étape précédente. Ces paramètres seront également définis dans le fichier de configuration de Zabbix agent :

Paramètre	Description
Pre-shared key identity	La chaîne d'identité de la clé pré-partagée. Définit le paramètre TLSPSKIdentity.
Pre-shared key value	La valeur de chaîne de la clé pré-partagée. 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 pré-partagée en ajustant les paramètres de sécurité du fichier afin que seul Zabbix agent (ou l'utilisateur exécutant l'agent) puisse le lire.

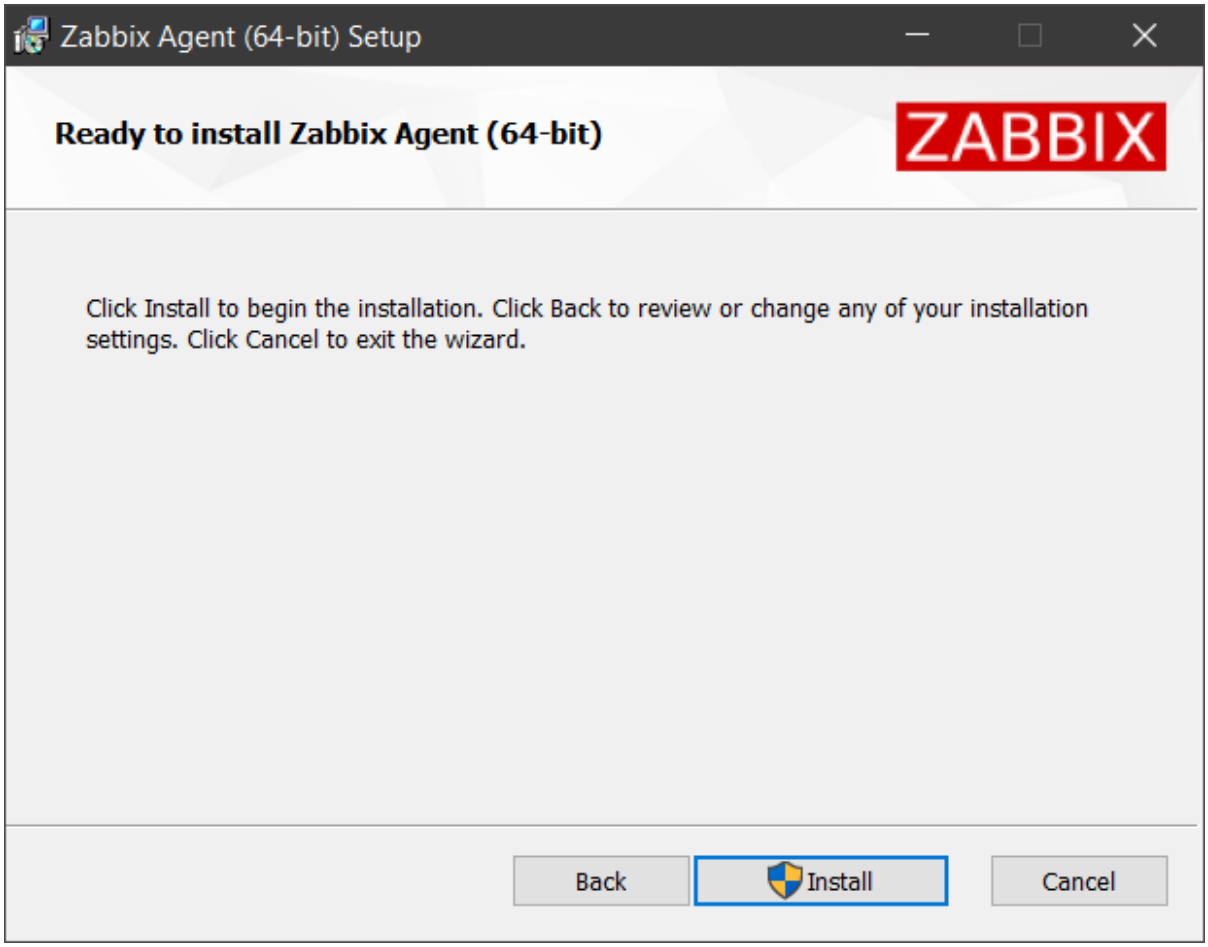


6. Cliquez sur *Install* pour commencer l'installation.

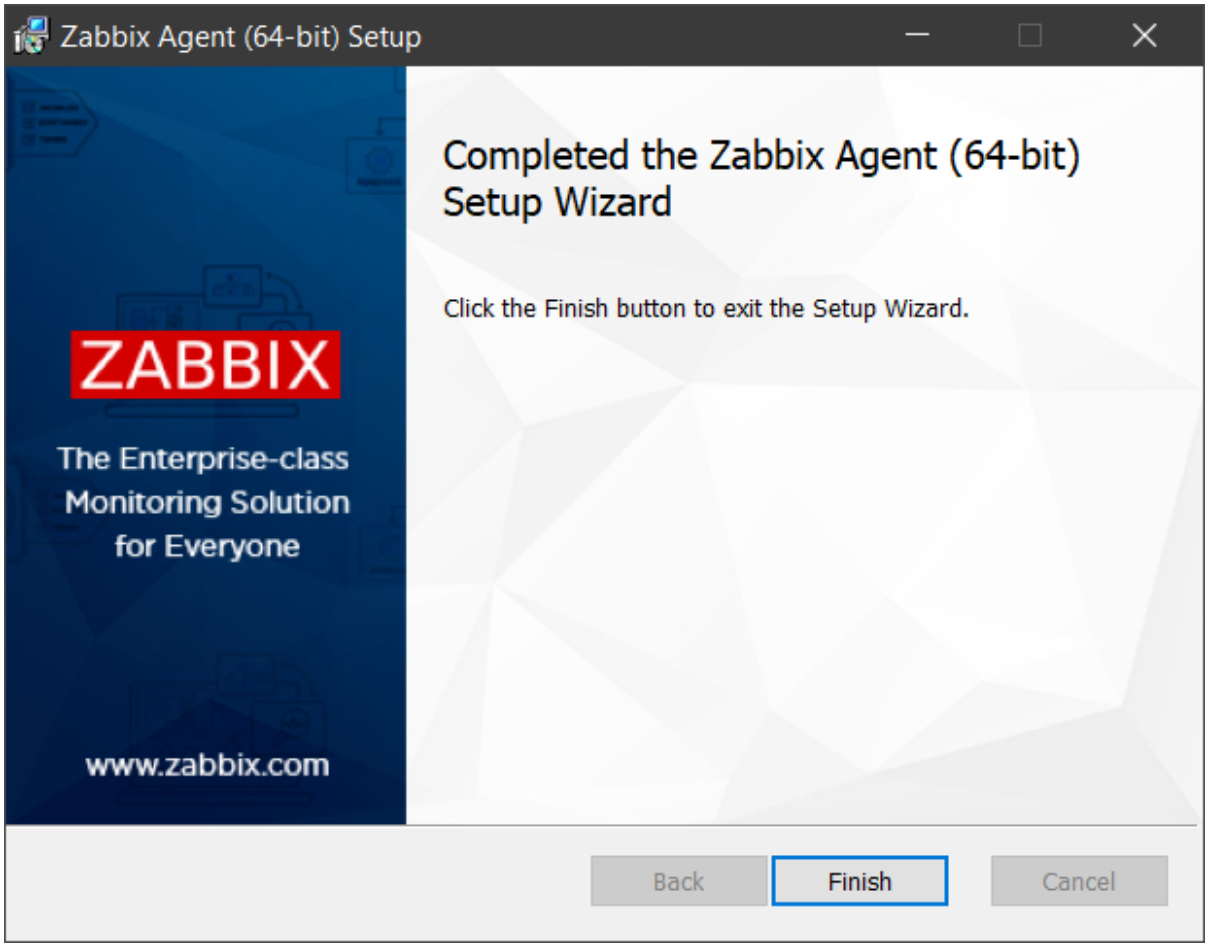
Tous les composants Zabbix sélectionnés ainsi que le fichier de configuration de Zabbix agent seront installés à l'emplacement que vous avez spécifié (par défaut : `C:\Program Files\Zabbix Agent`). Il en va de même pour Zabbix agent 2, sauf que des fichiers de configuration supplémentaires pour ses plugins 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 démarrage automatique différé (ou 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, il vous sera demandé de choisir soit de fermer l'application et d'essayer de la redémarrer, soit de la laisser ouverte, auquel cas un redémarrage sera nécessaire.



7. Cliquez sur le bouton *Finish* 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 assistance et des configurations personnalisées à l'aide de paramètres.

Paramètres pris en charge

Les packages d'installation MSI de l'agent Zabbix 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, reportez-vous à la page Zabbix agent 2 (Windows).

Parameter	Description
ADDDEFAULT	Une liste de composants, séparés par des virgules, à installer avec leur configuration par défaut. Pour plus d'informations, voir 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, voir 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 contrôles de l'agent Zabbix. 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 modèle pour Zabbix agent. Pendant l'installation, ce fichier deviendra le fichier de configuration de l'agent. Le fichier doit contenir au minimum 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 d'un 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, séparée par des points-virgules, de fichiers individuels ou de tous les fichiers d'un répertoire à inclure dans le fichier de configuration de Zabbix agent.
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 les connexions provenant du serveur.
LOGFILE	Le nom du fichier journal de Zabbix agent.
LOGTYPE	Le type de sortie du journal.
NONMSICONFNAME	Le chemin complet vers un fichier de configuration personnalisé pour Zabbix agent. Pendant l'installation, tous les paramètres de configuration d'agent valides présents dans ce fichier (limités à ceux listés dans ce tableau) seront écrits dans le fichier de configuration de l'agent nouvellement créé. Le fichier doit contenir au minimum le paramètre Server. Exemple : `NONMSICONFNAME="C:\full\path\to\example.conf"`

Parameter	Description
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 conservées 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 proxys Zabbix. Ce paramètre est **obligatoire**, sauf lorsque STARTAGENTS est défini sur 0.
SERVERACTIVE	L'adresse du serveur/proxy Zabbix ou la configuration du cluster à partir de laquelle obtenir les contrôles actifs.
SKIP	Utilisez `SKIP=fw` pour empêcher le programme d'installation MSI d'ajouter une règle d'exception du 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 contrôles passifs. Si défini sur 0, les contrôles passifs sont désactivés et l'agent n'écoutera sur aucun port TCP.
STARTUPTYPE	Type de démarrage du service Zabbix agent. Valeurs possibles : **automatic** - démarrer le service automatiquement au démarrage de Windows ; **delayed** - *(par défaut)* retarder le démarrage du service jusqu'à ce que les services démarrés automatiquement aient terminé leur démarrage (disponible sur Windows Vista/Server 2008 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. 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 combien de temps attendre (en secondes) pour établir une connexion et échanger des données avec le proxy ou le serveur Zabbix.
TLSACCEPT	Les connexions entrantes à accepter (utilisé pour les contrôles passifs). Si défini sur `psk`, alors TLSCONNECT sera également défini sur `psk` (sauf indication contraire).
TLSCAFILE	Le chemin complet d'un fichier contenant les certificats de la ou des autorités de certification racines pour la vérification du certificat du pair.
TLSCERTFILE	Le chemin complet d'un fichier contenant le certificat de l'agent ou la chaîne de certificats.
TLSCONNECT	Comment l'agent doit se connecter au serveur ou au proxy Zabbix (utilisé pour les contrôles actifs). Si défini sur `psk`, alors TLSACCEPT sera également défini sur `psk` (sauf indication contraire).
TLSCRLFILE	Le chemin complet d'un fichier contenant les certificats révoqués.
TLSKEYFILE	Le chemin complet d'un fichier contenant la clé privée de Zabbix agent.
TLSPSKFILE	Le chemin complet d'un fichier contenant la clé pré-partagée 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 pré-partagée 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 pré-partagée.
TLSPSKVALUE	La valeur de chaîne de la clé pré-partagée. 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 de certificat autorisé du serveur (proxy).
TLSSERVERCERTSUBJECT	Le sujet de certificat autorisé du serveur (proxy).
UNSAFEUSERPARAMETERS	Autorise le passage de tous les caractères dans les arguments des paramètres définis par l'utilisateur.

Exemples

L'exemple suivant installe l'agent Zabbix avec une configuration personnalisée. Il active également la prise en charge de TLS à l'aide de clés pré-partagées.

```
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=192.0.2.0^
```

```
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 l'agent Zabbix et utilise un **fichier de configuration 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 d'agent valides présents dans ce fichier (limités à ceux répertoriés dans le tableau ci-dessus) seront écrits dans le fichier de configuration de l'agent nouvellement créé. 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 en téléchargement.

La configuration minimale requise du système d'exploitation pour une 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 ainsi que 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 du programme 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 avec leur configuration par défaut. Pour plus d'informations, voir ADDDEFAULT property. 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, voir ADDLOCAL property. Valeurs possibles : `ALL`, `CephPlugin`, `EmberplusPlugin`, `MongodbPlugin`, `MssqlPlugin`, `NvidiagpuPlugin`, `PostgresqlPlugin` Exemple : `ADDLOCAL=MongodbPlugin,MssqlPlugin`
INSTALLFOLDER	Le chemin complet vers un dossier dans lequel 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

Aperçu

Cette page montre comment compiler Zabbix agent 2 à partir des sources sous Windows 10 (64 bits ou 32 bits).

Les versions 32 bits et 64 bits peuvent toutes deux être compilées sur une plateforme 64 bits, mais seule la version 32 bits peut être compilée sur une plateforme 32 bits.

La compilation de Zabbix agent 2 nécessite :

- les outils de compilation MinGW
- le langage de programmation Go
- OpenSSL (pour les fonctionnalités de chiffrement dans Zabbix)
- PCRE2 (Perl Compatible Regular Expressions ; pour les fonctionnalités de correspondance de motifs par expressions régulières dans Zabbix)

Vous pouvez compiler Zabbix agent 2 en utilisant 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 de compiler l'agent.

Attention:
Avant de commencer le processus de compilation, veuillez garder à l'esprit ce qui suit :

- Pour exécuter les commandes, utilisez l'invite de commandes, lancée par un utilisateur disposant de permissions suffisantes pour écrire dans des dossiers protégés. Toutefois, lors de l'installation de OpenSSL et de PCRE2, utilisez le terminal MSYS2.
- Il est recommandé de créer un répertoire de travail dans `C:\Zabbix` pour tous les fichiers source et dossiers de compilation. Toutefois, 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 des instructions pour compiler Zabbix agent avec vcpkg, un gestionnaire de paquets qui simplifie la gestion des dépendances et l'intégration avec les 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`

Ensuite, extrayez-la vers `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 cela 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

#### For 64-bit builds:
set PATH=C:\Zabbix\mingw64\bin;%PATH%
vcpkg install --triplet x64-mingw-static --x-install-root=x64

#### For 32-bit builds:
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 vers `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 :

```
:: Add MinGW and Go to the system `PATH` variable for the current session:
set PATH=C:\Zabbix\mingw64\bin;%PATH%
set PATH=C:\Zabbix\Go\bin;%PATH%

:: Set vcpkg installation path:
set vcpkg="C:\Zabbix\x64\x64-mingw-static"

:: Set linker flags for Crypt32 library:
SET CGO_LDFLAGS="-lCrypt32"

:: Run the build process:
mingw32-make GOFLAGS="-buildvcs=false" ARCH=AMD64 ^
    PCRE2="%vcpkg%" ^
    OPENSSL="%vcpkg%" ^
    all
```

- Pour les compilations 32 bits :

```
:: Add MinGW and Go to the system `PATH` variable for the current session:
set PATH=C:\Zabbix\mingw32\bin;%PATH%
set PATH=C:\Zabbix\Go\bin;%PATH%

:: Set vcpkg installation path:
set vcpkg="C:\Zabbix\x86\x86-mingw-static"

:: Set linker flags for Crypt32 library:
SET CGO_LDFLAGS="-lCrypt32"

:: Run the build process:
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 `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

#### For 64-bit builds:
copy C:\Zabbix\zabbix-8.0.0\bin\win64\zabbix_agent2.exe C:\Zabbix\agent2\

#### For 32-bit builds:
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.win.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.win.conf
```

Si nécessaire, poursuivez avec la compilation des plugins chargeables de Zabbix agent 2.

Compilation des plugins chargeables de Zabbix agent 2

1. Téléchargez les sources du plugin Zabbix correspondant à votre version de Zabbix agent 2 (par exemple, `zabbix-agent2-plugin-postgr`) et extrayez-les 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. Indiquez 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 indiqué 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.

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`

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 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 les droits d'écriture des utilisateurs non administrateurs sur le 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 entraînera une vulnérabilité de sécurité potentielle.

- L'option `no-shared` rend autonomes les bibliothèques statiques OpenSSL libcrypto.lib et libssl.lib, 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 à l'exécution. Cela signifie que les mises à jour d'OpenSSL peuvent ne pas nécessiter la recompilation de l'agent Zabbix ; toutefois, lors de la copie vers 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 cela 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 brut (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** avec Go installé (requis pour configurer Zabbix agent 2) :

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
cd zabbix
./bootstrap.sh
./configure --enable-agent2 --enable-ipv6 --prefix=`pwd`
make dist
```

Cela créera une archive des sources, qui pourra ensuite être copiée sur une machine Windows.

2. Ouvrez l'invite de commandes avec les privilèges administrateur. Ensuite, accédez au répertoire de compilation de Zabbix et compilez Zabbix agent ; veillez à spécifier correctement les répertoires où 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 prise en charge de TLS :
mingw32-make ARCH=AMD64 PCRE2="C:\Zabbix\x64\PCRE2" OPENSSL="C:\Zabbix\x64\OpenSSL-Win64-350-static"
```

```
#### Sans 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 prise en charge de TLS :
mingw32-make ARCH=x86 PCRE2="C:\Zabbix\x86\PCRE2" OPENSSL="C:\Zabbix\x86\OpenSSL-Win64-350-static"

#### Sans prise en charge de TLS :
mingw32-make ARCH=x86 PCRE2="C:\Zabbix\x86\PCRE2"
```

Après la compilation, le binaire de Zabbix agent 2 se trouvera dans `C:\Zabbix\zabbix-8.0.0\bin\win64` (ou `C:\Zabbix\zabbix-8.0.0` 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.win.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.win.conf
```

Si nécessaire, poursuivez avec la compilation des plugins chargeables de Zabbix agent 2.

Compilation de l'agent Zabbix sous Windows

Aperçu

Cette page montre 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 dans Zabbix)
- PCRE2 (Perl Compatible Regular Expressions ; pour les fonctionnalités de correspondance de motifs par 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 de compiler l'agent.

Selon vos besoins de supervision, des bibliothèques supplémentaires peuvent être nécessaires. Pour plus d'informations, consultez Configuration requise.

> **Attention:**
> Avant de commencer le processus de compilation, veuillez garder à l'esprit ce qui suit :

> - Pour exécuter les commandes, utilisez x64 Native Tools Command Prompt (inclus dans Build Tools for Visual Studio 2022), lancé par un utilisateur disposant de permissions suffisantes pour écrire dans des dossiers protégés.
> - Il est recommandé de créer un répertoire de travail dans `C:\Zabbix` pour tous les fichiers source et 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 avec les 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 les outils requis pour compiler l'agent avec vcpkg :

- Compilateur C (Microsoft Visual C++)
- Outil en ligne de commande NMake
- Gestionnaire de paquets vcpkg
- x64 Native Tools Command Prompt

2. Initialisez vcpkg et installez les dépendances requises pour compiler l'agent Zabbix (notez que cela 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 où OpenSSL et PCRE2 sont installés :

```
:: Set vcpkg installation path:
set vcpkg=C:\Program Files\Zabbix\x64\x64-windows-static

:: Run the build process:
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. Compilez l'agent Zabbix en exécutant le script :

```
build.bat
```

Après la compilation, les binaires des composants Zabbix se trouveront 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_agent.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_agent.exe C:\Zabbix\agent\
copy C:\Zabbix\zabbix-8.0.0\conf\zabbix_agent.win.conf C:\Zabbix\agent\

C:\Zabbix\agent\zabbix_agent.exe -c C:\Zabbix\agent\zabbix_agent.win.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 *Développement Desktop en C++*, qui inclut les outils nécessaires pour compiler l'agent manuellement :

- Compilateur C (Microsoft Visual C++)
- Outil en ligne de commande NMake
- Invite de commandes x64 Native Tools

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). Pendant 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 NASM.

4. Téléchargez l'archive des sources 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 non administrateurs. Sinon, l'agent chargera les paramètres SSL à partir d'un chemin pouvant être modifié par des utilisateurs non privilégiés, ce qui entraînera une vulnérabilité de sécurité potentielle.

- 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 à l'exécution. Cela signifie que les mises à jour d'OpenSSL peuvent ne pas nécessiter de recompilation de l'agent Zabbix ; toutefois, lors de la copie vers 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.

6. Compilez OpenSSL et exécutez les tests (notez que cela peut prendre un certain temps) :

Attention:
Exécutez les tests sans privilèges administratifs ; sinon, cela peut entraîner des résultats inattendus ou des risques de sécurité. Si certains tests échouent, consultez la documentation OpenSSL pour le dépannage.

```
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, 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 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 brut (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 --enable-agent --enable-ipv6 --prefix=`pwd`
make dist
```

Cela créera une archive des sources, qui pourra ensuite être copiée sur une machine Windows.

2. Accédez au répertoire de compilation de Zabbix et compilez l'agent Zabbix (ou d'autres composants) ; assurez-vous de spécifier correctement les répertoires où OpenSSL et PCRE2 sont installés :

```
cd C:\Zabbix\zabbix-8.0.0\build\win32\project

#### Avec 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 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
```

Après la compilation, les binaires des composants Zabbix se trouveront 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_agent.exe` et son fichier de configuration dans un dossier dédié (par exemple, `C:\Zabbix\agent`), puis lancez 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.win.conf C:\Zabbix\agent\

C:\Zabbix\agent\zabbix_agentd.exe -c C:\Zabbix\agent\zabbix_agentd.win.conf -f
```

Installation de l'agent macOS à partir d'un PKG

Aperçu

L'agent Zabbix peut être installé sur macOS à l'aide de paquets d'installation PKG, disponibles en téléchargement.

Les paquets 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 un terminal et exécutez la commande suivante :

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

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 les binaires de l'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 les binaires de l'agent Zabbix ou à d'autres fins.

Si les binaires de l'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
```

4 Mise à niveau

Avant d'effectuer 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.

Aperçu

Des instructions de mise à niveau étape par étape 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 haute disponibilité (HA), voir Mise à niveau d'un cluster HA
- Pour la base de données TimescaleDB, voir Mise à niveau du schéma TimescaleDB

Composants à mettre à niveau

La mise à niveau des agents Zabbix est recommandée, mais n'est pas obligatoire.

La mise à niveau des proxys Zabbix est fortement recommandée. Zabbix server prend entièrement en charge les proxys de la même version majeure que le serveur. Zabbix server prend également en charge les proxys qui ne sont **pas plus anciens** que la

version de la précédente publication LTS de Zabbix server, mais avec des fonctionnalités limitées (collecte de données, exécution de commandes distantes, vérifications immédiates des valeurs d'élément). La mise à jour de la configuration est également désactivée, et les proxys obsolètes ne fonctionneront qu'avec une ancienne configuration.

Attention:
Les proxies plus anciens que la version de la précédente publication LTS de Zabbix server ou plus récents que la version majeure de Zabbix server ne sont pas pris en charge. Zabbix server ignorera les données provenant de proxies non pris en charge et toute communication avec Zabbix server échouera avec un avertissement. Pour plus d'informations, voir Compatibilité des versions.

Afin de minimiser les temps d'arrêt et les pertes de données pendant la mise à niveau, il est recommandé d'arrêter, de mettre à niveau, puis de démarrer Zabbix server, puis d'arrêter, de mettre à niveau et de démarrer les proxies Zabbix l'un après l'autre. Pendant l'arrêt de Zabbix server, les proxies en cours d'exécution continueront la collecte des données. Une fois le serveur démarré et opérationnel, les proxies obsolètes enverront les données au serveur plus récent (la configuration du proxy ne sera toutefois pas mise à jour) et resteront partiellement fonctionnels. Toutes les notifications concernant des problèmes survenus pendant l'arrêt de Zabbix server ne seront générées qu'après le démarrage du serveur mis à niveau.

Si Zabbix proxy est démarré pour la première fois et que le fichier de base de données SQLite est manquant, le proxy le crée automatiquement.

Notez que si Zabbix proxy 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 de Zabbix proxy est antérieure à la version du fichier de base de données, Zabbix consignera une erreur et s'arrêtera.

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 à partir des versions **2.0**.x. Pour effectuer une mise à niveau depuis des versions antérieures, consultez la documentation Zabbix pour la version 2.0 et les versions précédentes.

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 de mise à niveau complètes	Principaux changements entre les versions
7.4.x	Pour : Zabbix 8.0	Versions minimales requises des bases de données relevées. Version minimale requise de PHP relevée 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 abandonnée.
7.0.x	Pour : Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	La prise en charge d'Oracle DB a été abandonnée.
6.4.x	Pour : Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Version minimale requise de PHP relevée de 7.4.0 à 8.0.0. Pollers asynchrones pour les vérifications agent, agent HTTP et SNMP walk[oid]. Table de base de données distincte pour les proxies. Emplacement par défaut du fichier de configuration de l'agent Windows modifié. Oracle DB obsolète. Ancien type de valeur numérique (float) supprimé.
6.2.x	Pour : Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Version minimale requise de MySQL relevée 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, Zabbix proxy avec SQLite3 supprime automatiquement l'ancienne version de la base de données (avec tout l'historique) et en crée une nouvelle.

Mise à niveau depuis	Lire les notes de mise à niveau complètes	Principaux changements entre les versions
6.0.x LTS	Pour : Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Version minimale requise de PHP relevée de 7.2.5 à 7.4.0. La supervision des services a été considérablement remaniée. Des 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*. Voir Database creation scripts pour savoir comment définir la 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	Versions minimales requises des bases de données relevées. Le serveur/proxy ne démarrera pas si la base de données est obsolète. Enregistrements du journal d'audit perdus en raison d'un changement de 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	Versions minimales requises des bases de données relevées. Les éléments agrégés ont été supprimés en tant que type distinct.
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	Version minimale requise de PHP relevée de 7.2.0 à 7.2.5. L'algorithme de hachage des mots de passe a été remplacé de MD5 par 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 abandonnée. Version minimale requise de PHP relevée de 5.4.0 à 7.2.0. Versions minimales requises des bases de données relevées. Répertoire des fichiers PHP de Zabbix 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.

Mise à niveau depuis	Lire les notes de mise à niveau complètes	Principaux changements entre les versions
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 proxies plus anciens ne peuvent plus signaler de données à un serveur mis à niveau. Les agents plus récents ne pourront plus fonctionner avec un ancien serveur Zabbix.
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 abandonnée et l'en-tête est obligatoire. Les agents Zabbix de version antérieure à 1.4 ne sont plus pris en charge. Le paramètre Server dans la configuration du proxy passif est désormais obligatoire.
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 backend a été abandonnée pour le serveur/frontend 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. Le Java gateway Zabbix doit être mis à niveau pour prendre en charge la nouvelle fonctionnalité.
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.

Mise à niveau depuis	Lire les notes de mise à niveau complètes	Principaux changements entre les versions
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	Version minimale requise de PHP relevée de 5.3.0 à 5.4.0. Le paramètre d'agent LogFile doit être spécifié.
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	Version minimale requise de PHP relevée 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 le classement utf8_bin sont requis pour que le serveur Zabbix fonctionne correctement avec une base de données MySQL. Voir database creation scripts. 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 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/apache2/conf-enabled/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, désinstallez votre paquet de dépôt Zabbix actuel :

```
rm -Rf /etc/apt/sources.list.d/zabbix.list
```

Il se peut également que vous deviez supprimer manuellement les 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 derniers correctifs de sécurité ou de bogues.

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 lien correct 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, vous devriez envisager de mettre à niveau votre OS. Pour obtenir la liste des composants inclus, consultez Paquets Zabbix.

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 lien correct 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, vous devriez envisager de mettre à niveau votre OS. Pour obtenir la liste des composants inclus, consultez Paquets Zabbix.

Vous pouvez voir une invite concernant la configuration du dépôt Zabbix :

```
Configuration file '/etc/apt/sources.list.d/zabbix.list'
==> Supprimé (par vous ou par un script) depuis l'installation.
==> Le distributeur du paquet a fourni une version mise à jour.
Que souhaitez-vous faire à ce sujet ? Vos options sont :
Y or I  : installer la version du responsable du paquet
N or O  : conserver votre version actuellement installée
D       : afficher les différences entre les versions
Z       : démarrer un shell pour examiner la situation
L'action par défaut est de conserver votre version actuelle.
*** zabbix.list (Y/I/N/O/D/Z) [default=N] ?
```

Saisissez `Y` (ou `I`) pour installer la version du responsable du paquet pour la 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*.

Vous pouvez voir une invite concernant la configuration du serveur Zabbix (ou du proxy) :

```
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, puis décidez si vous souhaitez installer la version du responsable du paquet (`Y` ou `I`).

Ensuite, pour mettre à niveau le frontend Zabbix avec Apache et redémarrer Apache, exécutez :

```
apt install zabbix-apache-conf
systemctl restart apache2
```

6 Vérifier les paramètres de configuration des composants

Consultez les notes de mise à niveau correspondantes 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).

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 d'assurer la compatibilité avec les paquets les plus récents et d'inclure les derniers correctifs de sécurité ou de bogues.

Sur **RHEL 10**, exécutez :

```
rpm -Uvh https://repo.zabbix.com/zabbix/8.0/release/rhel/10/noarch/zabbix-release-latest.el10.noarch.rpm
```

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 ses dérivés, remplacez le lien ci-dessus par le lien approprié 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 de paquets, il convient d'envisager une mise à niveau de 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 de nouvelles métadonnées depuis les dépôts, puisque les anciennes métadonnées auront été supprimées.

Voir aussi : Problèmes connus pour la mise à jour du 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-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 l'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 `dnf install zabbix-agent2` peut entraîner une erreur. Pour plus d'informations, consultez *Problèmes connus*.

Ensuite, pour mettre à niveau le frontend Zabbix avec Apache et redémarrer Apache, exécutez :

```
dnf install zabbix-apache-conf
systemctl restart httpd
```

6 Vérifier les paramètres de configuration des composants

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

Aperçu

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, assurez-vous de lire les **notes de mise à niveau** correspondantes !

Vous pouvez également consulter les prérequis pour la version 8.0.

Note:
Il peut être utile d'exécuter deux sessions SSH parallèles pendant la mise à niveau, en effectuant les étapes de mise à niveau dans l'une et en surveillant 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 erreurs éventuelles en temps réel. Cela peut être essentiel pour les instances de production.

Procédure de mise à jour du serveur

1 Arrêt du serveur Zabbix

Arrêter le serveur Zabbix pour être sûr qu'aucune nouvelle donnée ne soit insérée en base de données.

2 Sauvegarde de la base de données Zabbix existante

Il s'agit d'une étape très importante. Vérifiez que vous disposez d'une sauvegarde de votre base de données. Cela aidera si la procédure de mise à jour échoue (manque d'espace disque, arrêt inopiné ou tout autre problème imprévu).

3 Sauvegarde des fichiers de configuration, des fichiers PHP et des binaires Zabbix

Faites 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 Vérifier 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 optionnels, consultez la page Nouveautés.

6 Démarrer les nouveaux binaires Zabbix

Démarrez les nouveaux binaires. Vérifiez les fichiers de logs pour voir si les fichiers binaires ont démarré avec succès.

Le serveur Zabbix mettra automatiquement à jour la base de données. Au démarrage, le serveur Zabbix signale les versions de base de données actuelles (obligatoires et facultatives) et requises. Si la version actuelle est antérieure à la version requise, le serveur Zabbix exécute automatiquement les correctifs requis de mise à jour de base de données. La progression (en pourcentage) de la mise à jour de la base de données est écrite dans le fichier de logs du serveur Zabbix. Lorsque la mise à jour est terminée, un message "Mise à jour de la base de données entièrement terminée" est écrit dans le fichier de logs. Si l'un des correctifs de mise à jour échoue, le serveur Zabbix ne démarre pas. Le serveur Zabbix ne démarrera pas non plus si la version de base de données actuelle est plus récente que celle requise. Le serveur Zabbix démarre uniquement si la version de base de données obligatoire actuelle 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 des autorisations nécessaires (créer une table, supprimer une table, créer un index, supprimer un index) ;
- Assurez-vous d'avoir suffisamment d'espace disque libre.

7 Installer la nouvelle interface web de Zabbix

La version minimale requise de PHP est 8.2.0. Mettez à jour si nécessaire et suivez les instructions d'installation.

8 Effacer les cookies et le cache du navigateur

Après la mise à jour, 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 Vérifier les paramètres de configuration du proxy

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 optionnels, consultez la page Nouveautés.

5 Démarrer le nouveau proxy Zabbix

Démarrez le nouveau proxy Zabbix. Vérifiez les fichiers journaux pour voir si 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 manière similaire à celle 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 ne devez mettre à niveau les agents que si cela est nécessaire pour accéder à la nouvelle fonctionnalité.

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 de nouveaux binaires d'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 Vérifier les paramètres de configuration de l'agent

Veillez à consulter les Notes de mise à niveau afin de vérifier si des modifications des paramètres de configuration sont nécessaires.

Pour les nouveaux paramètres facultatifs, consultez la page Nouveautés.

5 Démarrage du nouvel agent Zabbix

Démarrez le nouvel agent Zabbix. Vérifiez les fichiers de logs 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'exécuter les mêmes actions pour le serveur/proxy/agent que lors d'une mise à niveau entre versions majeures. La seule différence est que, lors d'une mise à niveau entre 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 :

```
\\\
```

2. La mise à niveau de la base de données se **termine** et le serveur Zabbix continue de fonctionner.
3. Calc_228 renvoie toujours 1 tandis que Calc_232 renvoie 0.
4. Les déclencheurs ont été mis à niveau comme suit :

Trig_228 :

```
find(/Zabbix server/vfs.file.contents[/tmp/ONE],1m:now,"regexp","\\\
```

Attention:
Avant de commencer la mise à niveau, vérifiez que les utilisateurs disposent des autorisations nécessaires sur la base de données à des fins de mise à niveau.

Pour les mises à niveau depuis Zabbix 6.0 ou une version antérieure, des déclencheurs déterministes devront 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 base de données pour savoir comment définir la variable.
Notez que si l'exécution se fait depuis une console, la variable ne sera définie que temporairement et sera supprimée lorsqu'un Docker sera redémarré. Dans ce cas, laissez votre service SQL en cours d'exécution, arrêtez uniquement le service zabbix-server en exécutant 'docker compose down zabbix-server', puis 'docker compose up -d zabbix-server'.
Vous pouvez également définir cette variable dans le fichier de configuration.

Selon la taille de la base de données, la mise à niveau vers la version 8.0 peut prendre assez longtemps.

Mise à niveau d'une image Zabbix

Les étapes ci-dessous peuvent être utilisées pour mettre à niveau n'importe quel composant Zabbix. Remplacez `zabbix-server-mysql` par le nom d'image du composant requis.

1. Vérifiez la version actuelle de l'image :

```
docker inspect -f '{{ .Config.Image }}' zabbix-server-mysql
```

2. Récupérez la version d'image souhaitée, par exemple :

```
docker pull zabbix/zabbix-server-mysql:alpine-8.0-latest
```

`zabbix/zabbix-server-mysql:alpine-8.0-latest` récupérera la dernière version mineure publiée de Zabbix server 8.0 avec prise en charge de MySQL, basée sur Alpine Linux. Remplacez-la par le nom du dépôt Docker et la combinaison de tags dont vous avez besoin. Consultez Installation à partir de conteneurs pour obtenir la liste des options disponibles.

3. Arrêtez le conteneur :

```
docker stop zabbix-server-mysql
```

4. Supprimez le conteneur :

```
docker rm zabbix-server-mysql
```

5. Lancez le conteneur mis à jour en exécutant la commande `docker run`, suivie d'arguments supplémentaires pour spécifier les variables d'environnement et/ou les points de montage requis.

Exemples de configuration

Zabbix server avec 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 MYSQL_ROOT_PASSWORD="root_pwd" \
    -e ZBX_JAVAGATEWAY="zabbix-java-gateway" \
    --network=zabbix-net \
    -p 10051:10051 \
    --restart unless-stopped \
    -d zabbix/zabbix-server-mysql:alpine-8.0-latest
```

Zabbix server avec PostgreSQL :

```
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 \
    -d zabbix/zabbix-server-pgsql:alpine-8.0-latest
```

Serveur MySQL :

```
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 \
```

D'autres exemples de configuration, y compris des exemples pour d'autres composants Zabbix, sont disponibles sur la page Installation à partir de conteneurs.

6. Vérifiez la mise à jour :

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

Fichiers Compose

Suivez les instructions de mise à niveau de cette section si vous avez installé Zabbix à l'aide d'un fichier compose.

1. Vérifiez la version actuelle de l'image :

```
docker inspect -f '{{ .Config.Image }}' zabbix-server-mysql
```

2. Récupérez les dernières mises à jour depuis le dépôt GitHub et basculez vers la branche requise :

```
git pull
git checkout 8.0
```

3. Démarrez les composants Zabbix à l'aide du nouveau fichier compose :

```
docker-compose -f ./docker-compose_v3_alpine_mysql_latest.yaml up -d
```

4. Vérifiez la mise à jour :

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

Consultez Installation à partir de conteneurs pour plus de détails, y compris les listes des variables d'environnement prises en charge et des points de montage de volumes.

5 Notes de mise à niveau pour Zabbix 8.0

Ces notes concernent la mise à niveau de Zabbix 7.4.x vers Zabbix 8.0.0.

Toutes les notes sont regroupées en :

- **Changements incompatibles** - modifications susceptibles de perturber les installations existantes et autres informations critiques liées au processus de mise à niveau
- **Autres** - toutes les informations restantes décrivant les changements de fonctionnalité dans Zabbix

Voir aussi :

- Procédure de mise à niveau pour toutes les informations pertinentes sur la mise à niveau depuis des versions antérieures à Zabbix 8.0.0 ;
- Mise à niveau d'un cluster HA pour des instructions sur la mise à niveau des serveurs dans un cluster **haute disponibilité** (HA).

Changements majeurs Versions des bases de données

Les versions minimales requises des bases de données ont été relevées :

- MySQL/Percona : 8.0.30 → 8.4.0
- MariaDB : 10.5.00 → 10.11.00

- PostgreSQL : 13.0 → 15.0
- TimescaleDB : 2.13.0 → 2.20.0

Plugins

Le plugin Ceph pour l'agent Zabbix 2 est désormais un plugin chargeable et nécessite des étapes d'installation supplémentaires. Consultez le readme du plugin Ceph pour plus de détails.

Le schéma `tcp://` est obsolète dans le plugin MongoDB, mais il est conservé pour assurer la rétrocompatibilité avec les configurations existantes.

Version minimale de PHP requise

La version minimale de PHP requise a été relevée de 8.0.0 à 8.2.0.

Liste étendue de UnsafeUserParameters

Le caractère % a été ajouté à la liste `UnsafeUserParameters` pour Zabbix agent et Zabbix agent 2.

Autres Deprecated macros dropped

The support for the following built-in macros has now been dropped:

- {ACK.DATE}
- {ACK.MESSAGE}
- {ACK.TIME}
- {EVENT.ACK.HISTORY}
- {HOSTNAME<1-9>}
- {IPADDRESS<1-9>}
- {PROFILE.*}
- {TRIGGER.COMMENT}
- {TRIGGER.KEY}
- {STATUS}
- {USER.ALIAS}

Suppression de la validation JSON et XML des éléments d'agent HTTP

La validation JSON et XML a été supprimée du champ *Corps de la requête* dans les éléments HTTP agent. Le champ ne vérifie plus si la valeur est un XML ou un JSON valide, et la sélection de *Données XML* comme *Type du corps de la requête* ne nécessite plus la bibliothèque libxml2.

Liens désactivés masqués dans les hôtes surveillés

Les liens *Graphiques*, *Tableaux de bord* et *Web* désactivés ne sont plus affichés dans la liste des hôtes surveillés.

Type de données JSON

Zabbix prend désormais en charge JSON comme type de données pour les valeurs d'élément. Si vous utilisez TimescaleDB, la nouvelle hypertable `history_json` (utilisée pour stocker les valeurs JSON) doit être configurée manuellement. Si vous utilisez Elasticsearch, la valeur par défaut du paramètre de configuration du serveur Zabbix `HistoryStorageTypes` inclut désormais `json`.

Removed radio button in the New tag filter form

The radio button that required choosing between *All tags* and *Tag list* when creating a new tag filter has been removed.

Item values displayed as HTML using iframes

Item data in the Item history widget, when displayed as HTML-formatted text, is now isolated in iframes.

Template compatibility with Host Wizard

All out-of-the-box templates are now compatible with the Host Wizard. To upgrade them, see Template upgrade.

Modifications des modèles

Cette page répertorie toutes les modifications apportées aux modèles standard fournis avec Zabbix.

La mise à niveau vers la dernière version de Zabbix ne mettra pas automatiquement à niveau les modèles utilisés. Il est recommandé de modifier les modèles dans les installations existantes en téléchargeant les derniers modèles depuis le dépôt Git Zabbix et en les important manuellement dans Zabbix.

Si des modèles portant les mêmes noms existent déjà, les options *Supprimer les éléments manquants* doivent être cochées lors de l'importation afin d'obtenir une importation propre. De cette manière, 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).

> **Note:**
> Veuillez noter que depuis Zabbix 6.0, tous les modèles suivent un format mis à jour, ce qui peut avoir un impact sur l'importation des modèles antérieurs à la version 6.0. Pour plus d'informations, consultez Modifications des modèles dans la version 6.0.

Modifications dans la version 8.0.0

Notes de mise à niveau pour Zabbix 8.0.x

Cette page fournit les notes de mise à niveau regroupées pour les versions mineures de la version majeure de Zabbix.

Voir également les notes de mise à niveau de la version majeure.

Notes de mise à niveau pour 8.0.1 Cette version n'est pas encore publiée.

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).

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.

Timescale DB : utilisation élevée de la mémoire avec un grand nombre de partitions

Les versions 9.6 à 12 de PostgreSQL utilisent trop de mémoire lors de la mise à jour de tables comportant un grand nombre de partitions. Ce problème se manifeste lorsque Zabbix met à jour les tendances sur des systèmes avec TimescaleDB si les tendances sont divisées en segments relativement petits (par exemple, 1 jour). Cela entraîne la présence de centaines de segments dans les tables de tendances avec les paramètres de housekeeping par défaut, une situation dans laquelle PostgreSQL risque fortement de manquer de mémoire.

Le problème a été résolu à partir de Zabbix 5.0.1 pour les nouvelles installations avec TimescaleDB, mais si TimescaleDB a été configuré avec Zabbix avant cela, veuillez consulter ZBX-16347 pour les notes de migration.

Timescale DB 2.5.0 : la politique de compression peut échouer sur les tables contenant des entiers

Ce problème se manifeste lorsque TimescaleDB 2.5.0/2.5.1 est utilisé. Il a été résolu à partir de TimescaleDB 2.5.2.

Pour plus d'informations, veuillez consulter TimescaleDB Issue #3773.

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 utilisée.

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 faire planter OpenIPMI

Il existe un bogue dans la bibliothèque OpenIPMI utilisée par Zabbix pour interroger les données IPMI, qui peut être déclenché par des réponses spécialement conçues provenant d'un périphérique non fiable.
Un périphérique IPMI non fiable peut envoyer des données malveillantes provoquant le plantage de la bibliothèque OpenIPMI, ce qui peut à son tour entraîner l'arrêt du processus Zabbix server 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 Zabbix server ou Zabbix proxy 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 bogue en amont. Configuration recommandée :

Connecteur PostgreSQL, SQLite ou Oracle → pilote MariaDB ou MySQL unixODBC Connecteur MariaDB → pilote MariaDB unixODBC Connecteur MySQL → pilote MySQL unixODBC

Consultez 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 surveiller des bases de données Oracle avec différentes versions d'Oracle Instant Client pour Linux provoque le plantage de Zabbix server. Voir aussi : ZBX-18402, ZBX-20803.
- Si vous utilisez le pilote FreeTDS UnixODBC, vous devez faire précéder une requête SQL de l'instruction 'SET NOCOUNT ON' (par exemple, `SET NOCOUNT ON DECLARE @strsql NVARCHAR(max) SET @strsql = ....`). Sinon, l'élément de surveillance de base de données dans Zabbix ne parviendra pas à récupérer les informations et renverra l'erreur "SQL query returned empty result".
  Consultez 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 aléatoire des paramètres régionaux du frontend

Il a été observé que les paramètres régionaux du frontend peuvent changer sans logique apparente, c.-à-d. que certaines pages (ou parties de pages) s'affichent dans une langue, tandis que d'autres pages (ou parties de pages) s'affichent dans une autre langue. En général, le problème peut apparaître lorsqu'il y a plusieurs utilisateurs, dont certains utilisent un paramètre régional, tandis que d'autres en utilisent un autre.

Une solution de contournement connue consiste à désactiver le multithreading dans PHP et Apache.

Le problème est lié à la manière dont le paramètre régional est défini dans PHP : les informations de paramètre régional 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 le paramètre régional soit modifié dans un autre thread, ce qui change la manière dont les données peuvent être traitées dans le thread Zabbix.

Pour plus d'informations, veuillez consulter les rapports de problème associés :

- ZBX-10911 (Problème de changement aléatoire des paramètres régionaux du frontend)
- ZBX-16297 (Problème de traitement des nombres dans les graphiques utilisant la fonction `bcdiv` des fonctions BC Math)

Graphiques

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 web (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 service 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 ; une solution de contournement possible consiste donc 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'homme non entre guillemets sont automatiquement converties en horodatages Unix.

Assistant de configuration sur SUSE avec NGINX et php-fpm

L'assistant de configuration du frontend 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 :

- Définissez l'option ProtectSystem sur 'true' au lieu de 'full' dans l'unité systemd de php-fpm.
- Enregistrez 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. Même si PCRE est toujours pris en charge, les paquets d'installation de 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. Bien qu'il offre de nombreux avantages, le passage à PCRE2 peut rendre certains motifs regexp PCRE existants invalides ou les faire se comporter différemment. Cela affecte en particulier le motif *^[\w-\.]*. Afin de rendre à nouveau cette regexp valide sans en modifier la sémantique, remplacez l'expression par *^[-\w\.]* . Cela est dû au fait 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 paquet de la version actuelle et le reconfigurer comme décrit dans les instructions de téléchargement, dans la section *e. Configurer PHP pour l'interface web Zabbix*.

Vous pouvez également 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;
}
```

Ensuite, remplacez 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, 𝒸, 𝓅). 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.

Accès aux éléments de l'interface utilisateur avec MariaDB 10.5.1–10.5.9

L'accès à l'interface web de Zabbix avec un rôle autre que Super Admin peut entraîner l'affichage du message : « Une erreur système s'est produite. Veuillez contacter l'administrateur Zabbix. ». Ce problème affecte les installations utilisant les versions de MariaDB 10.5.1 à 10.5.9.

Pour éviter ce problème, mettez à jour MariaDB vers une version postérieure à 10.5.9. Pour plus de détails, consultez ZBX-25746.

Analyse de l'utilisation excessive de la mémoire avec tcmalloc

Si vous soupçonnez que votre installation Zabbix utilise trop de mémoire, vous pouvez utiliser la fonctionnalité de profilage mémoire de tcmalloc pour examiner 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 réel du processus (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.).
- 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.

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 démarrer. 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 (GiB)	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] 1 métrique = 1 élément + 1 déclencheur + 1 graphique
 [2] Exemple avec des instances Amazon EC2 à usage général, utilisant une architecture ARM64 ou x86_64 ; un type d'instance approprié, comme Compute/Memory/Storage optimised, doit être sélectionné lors de l'évaluation et des tests de l'installation Zabbix avant son déploiement dans l'environnement de production.

Note:
La configuration réelle dépend fortement du nombre d'éléments actifs et des intervalles d'actualisation (voir la section database size 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.

Plateforme	Serveur	Agent	Agent 2	Commentaires
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

S'il est indiqué comme obligatoire, le logiciel/la bibliothèque requis(e) est strictement nécessaire. Les éléments optionnels sont nécessaires pour prendre en charge certaines fonctions spécifiques.

Logiciel	Statut obligatoire	Versions prises en charge	Commentaires
MySQL/Percona	L'un de	8.4.0-9.5.X	Requis si MySQL (ou Percona) est utilisé comme base de données backend de Zabbix. Le moteur InnoDB est requis. Nous recommandons d'utiliser la bibliothèque C API (libmysqlclient) pour compiler le serveur/le proxy.
MariaDB		10.11.00-12.0.X	Le moteur InnoDB est requis. La version recommandée est 11.4. Nous recommandons d'utiliser la bibliothèque MariaDB Connector/C pour compiler le serveur/le proxy. Voir aussi : Interblocages possibles avec MariaDB et Accès aux éléments de l'interface avec MariaDB 10.5.1–10.5.9.
PostgreSQL		15.0-18.X	Requis si PostgreSQL est utilisé comme base de données backend de Zabbix. Selon la taille de l'installation, il peut être nécessaire d'augmenter la propriété de configuration PostgreSQL *work_mem* (4MB étant la valeur par défaut), afin que la quantité de mémoire utilisée par la base de données pour une opération particulière soit suffisante et que l'exécution des requêtes ne prenne pas trop de temps.
TimescaleDB pour PostgreSQL		2.20.X-2.25.X	Requis si TimescaleDB est utilisé comme extension de base de données PostgreSQL. Assurez-vous d'installer TimescaleDB Community Edition, qui prend en charge la compression. Notez que PostgreSQL 15 est pris en charge à partir de TimescaleDB 2.10. Vous pouvez également consulter la documentation TimescaleDB pour plus de détails concernant la compatibilité des versions de PostgreSQL et de TimescaleDB.
SQLite	Optionnel	3.3.5-3.34.X	SQLite est uniquement pris en charge avec les proxys Zabbix. Requis si SQLite est utilisé comme base de données de proxy Zabbix.

Logiciel	Statut obligatoire	Versions prises en charge	Commentaires
Elasticsearch		7.X	Elasticsearch est uniquement pris en charge avec les serveurs Zabbix, exclusivement pour le stockage des données d'historique. La prise en charge d'Elasticsearch est actuellement expérimentale. Voir aussi les logiciels requis pour server/proxy.
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`, faisant partie de 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, pour une expérience optimale, nous recommandons d'utiliser des bases de données installées à partir des dépôts officiels des développeurs de bases de données.

Interface web

La largeur minimale d'écran prise en charge pour l'interface web Zabbix est de 1200px.

Si un composant est indiqué comme obligatoire, le logiciel/la bibliothèque requis(e) est strictement nécessaire. Les composants optionnels sont nécessaires pour prendre en charge certaines fonctions spécifiques.

Logiciel	Statut obligatoire	Versions prises en charge	Commentaires
PHP	Oui	8.2.0 - 8.4.X	
Apache	L'un de	2.4 ou ultérieure	
Nginx		1.20 ou ultérieure	
MySQL	L'un de	Voir Logiciels externes tiers de l'environnement	
PostgreSQL			
Extensions PHP			
mysqli	Oui		Requis si MySQL est utilisé comme base de données backend de Zabbix.
pgsql			Requis si PostgreSQL est utilisé comme base de données backend de 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 ultérieure	php-gd (si fourni en tant que paquet séparé 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 possible chevauchement du texte dans les graphiques pour certaines langues de l'interface web.
libxml		2.6.15 ou ultérieure	php-xml (si fourni en tant que paquet séparé par le distributeur)

Logiciel	Statut obligatoire	Versions prises en charge	Commentaires
xmlwriter			php-xmlwriter (si fourni en tant que paquet séparé par le distributeur)
xmlreader			php-xmlreader (si fourni en tant que paquet séparé par le distributeur)
ctype			php-ctype (*--enable-ctype*)
session			php-session (si fourni en tant que paquet séparé 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 ultérieure	php-curl ; requis pour l'AMF Duo Universal Prompt et l'authentification SMTP.

Bibliothèques frontend tierces fournies avec Zabbix :

Bibliothèque	Statut obligatoire	Version minimale	Commentaires
jQuery JavaScript Library	Oui	3.6.0	Bibliothèque JavaScript qui simplifie le processus de développement multiplateforme entre navigateurs.
jQuery UI		1.12.1	Ensemble d'interactions d'interface utilisateur, d'effets, de widgets et de thèmes construit au-dessus de jQuery.
SAML PHP Toolkit		4.0.0	Boîte à outils PHP qui ajoute la prise en charge de l'authentification SAML 2.0 afin de permettre la connexion à 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, MySQL et PostgreSQL.

Attention:
Pour des polices autres que la police DejaVu par défaut, la fonction PHP imagerotate peut être requise. Si elle est absente, ces polices peuvent être affichées incorrectement lors de l'affichage d'un graphique. Cette fonction n'est disponible que si PHP est compilé avec GD intégré, ce qui n'est pas le cas dans Debian et d'autres distributions.

Bibliothèques tierces utilisées pour l'écriture et le débogage du code de l'interface web 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 des bibliothèques requises.
PHPUnit		8.5.29	Framework de tests unitaires PHP pour tester l'interface web Zabbix.
SASS		3.4.22	Langage de script de 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

Si un élément est indiqué comme obligatoire, le logiciel/la bibliothèque requis est strictement nécessaire. Les éléments optionnels 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 Perl (PCRE). PCRE2 v10.x est pris en charge.
libevent		Requis pour la communication inter-processus. Version 2.0.10 ou supérieure.
libevent-pthreads		Requis pour la communication inter-processus.
libpthread		Requis pour la prise en charge des mutex et des verrous lecture-écriture (peut faire partie de libc).
libresolv		Requis pour la résolution DNS (peut faire partie de libc).
libiconv		Requis pour l'encodage/la conversion 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 de	Requis si MySQL est utilisé.
libmariadb		Requis si MariaDB est utilisé.
libpq5		Requis si PostgreSQL est utilisé ; la version de *libpq5* doit correspondre à celle de la base de données PostgreSQL utilisée ou lui être supérieure.
libsqlite3		Requis si SQLite est utilisé. Requis uniquement pour le proxy Zabbix.
libOpenIPMI	Non	Requis pour la prise en charge d'IPMI. Requis uniquement par le serveur Zabbix.
libssh2 or *libssh*		Requis pour les vérifications SSH. Version 1.8.0 ou supérieure (libssh2) ; 0.9.0 ou supérieure (libssh).
libcurl		Requis pour les fonctionnalités suivantes : - les éléments de surveillance Web, de surveillance VMware et agent HTTP (pour tous : version 7.19.1 ou supérieure) ; - les éléments web.page.* de l'agent Zabbix (version 7.19.1 ou supérieure ; voir aussi les exigences de l'agent) ; - l'authentification SMTP (Basic : version 7.20.0 ou supérieure ; OAuth : version 7.33 ou supérieure ; voir aussi les exigences du frontend) ; - Elasticsearch (version 7.28.0 ou supérieure). La version 7.28.0 ou supérieure est recommandée pour toutes les fonctionnalités. Pour utiliser les fonctionnalités cURL mises à niveau pour les éléments `web.page.*`, redémarrez le serveur/proxy Zabbix. Pour l'authentification SMTP, utilisez le paquet `libcurl-full` à l'exécution.
libxml2		Requis pour la surveillance VMware et le prétraitement XML XPath.
net-snmp		Requis pour la prise en charge de SNMP. Version 5.3.0 ou supérieure. La prise en charge des protocoles de chiffrement forts (AES192/AES192C, AES256/AES256C) est disponible à partir de la bibliothèque net-snmp 5.8 ; sur les systèmes basés sur RHEL 8+, il est recommandé d'utiliser net-snmp 5.8.15 ou une version ultérieure.
libunixodbc		Requis pour la surveillance de base de données.
libgnutls or *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 mise en cache des requêtes DNS (cache de requêtes activé par défaut) nécessite c-ares 1.26.0 ou une version ultérieure.

Agent

Exigence	Statut obligatoire	Description
libpcre2	Oui	La bibliothèque PCRE2 est requise pour la prise en charge des expressions régulières compatibles Perl (PCRE). PCRE2 v10.x est pris en charge. Requise pour la surveillance des journaux. Également requise sous Windows.
libpthread		Requise pour la prise en charge des mutex et des verrous 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 l'encodage/la conversion du format du texte en UTF-8 dans les éléments de journal, le contenu de fichier, les éléments de regex de fichier et regmatch (peut faire partie de libc). Non requise sous Windows.
libgnutls ou *libopenssl*	Non	Requise si le chiffrement est utilisé. Versions minimales : *libgnutls* - 3.1.18, *libopenssl* - 1.0.1 Sous Microsoft Windows, OpenSSL 1.1.1 ou une 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 web.page.* de l'agent Zabbix. 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 (la version 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

Exigence	Statut obligatoire	Description
Go	Oui	Requis pour compiler l'agent Zabbix 2 et ses plugins à partir des sources. Go 1.24.10 ou version ultérieure est pris en charge. Voir go.dev pour les instructions d'installation. Les bibliothèques Go utilisées par l'agent Zabbix 2 et ses plugins sont répertoriées dans le dépôt Git de Zabbix (les bibliothèques marquées comme `indirect` dans le dépôt sont des dépendances d'autres bibliothèques requises) : - Agent Zabbix 2 - Prise en charge des plugins - PostgreSQL - MongoDB - MSSQL - Ember+ - GPU NVIDIA - Plugin d'exemple
libpcre2	Oui	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.
libopenssl	Non	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 de 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 à partir du dépôt des sources ou d'une archive, les dépendances nécessaires sont déjà incluses dans l'arborescence des sources.

Si vous avez obtenu Zabbix à partir du paquet de votre distribution, les dépendances nécessaires sont déjà fournies par le système de 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 fournis avec Java gateway dans le code source d'origine :

Library	Mandatory status	Minimum version	Comments
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 org.json extraite du SDK Android.
logback-classic		1.5.16	
logback-core		1.5.16	
slf4j-api		2.0.16	

Java gateway peut être compilé à l'aide d'Oracle Java ou de l'OpenJDK open source (version 1.6 ou ultérieure). Les paquets fournis par Zabbix sont compilés avec OpenJDK. Le tableau suivant répertorie les paquets OpenJDK utilisés pour compiler les paquets Zabbix selon la distribution :

Distribution	OpenJDK package
AlmaLinux 9	`java-11-openjdk-devel` (amd64: 11.0.19.0.7-4; arm64: 11.0.20.0.8-3)
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 9	`java-11-openjdk-devel` (amd64, arm64: 11.0.18.0.10-3)
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 12	`default-jdk-headless` (amd64, arm64: 2:1.17-74)
Debian 11	`default-jdk-headless` (amd64: 2:1.11-72)
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 9	`java-11-openjdk-devel` (amd64: 11.0.19.0.7-4.0.1; arm64: 11.0.20.0.8-2.0.1)
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)
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 9	`java-11-openjdk-devel` (amd64: 11.0.19.0.7-4; arm64: 11.0.20.0.8-3)
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 9	`java-11-openjdk-devel` (amd64: 11.0.19.0.7-4; arm64: 11.0.20.0.8-3)
Rocky Linux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.372.b07-4; arm64: 1.8.0.382.b05-2)
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 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)

Numéros de port par défaut

La liste suivante des ports ouverts par composant s'applique à la configuration par défaut :

Composant Zabbix	Numéro de port	Protocole	Type de connexion
Agent Zabbix	10050	TCP	à la demande
Agent Zabbix 2	10050	TCP	à la demande
Serveur Zabbix	10051	TCP	à la demande
Proxy Zabbix	10051	TCP	à la demande
Zabbix Java gateway	10052	TCP	à la demande
Service web Zabbix	10053	TCP	à la demande
Frontend Zabbix	80	HTTP	à la demande
	443	HTTPS	à la demande
Trappeur Zabbix	10051	TCP	à la demande

Note:
Les numéros de port doivent être ouverts dans le pare-feu pour permettre les communications Zabbix. Les connexions TCP sortantes ne nécessitent généralement pas de règles explicites dans le pare-feu.

Taille de la base de données

Les données de configuration de Zabbix nécessitent une quantité fixe d'espace disque et n'augmentent pas beaucoup.

La taille de la base de données Zabbix dépend principalement des variables suivantes, 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 surveillés avec un intervalle d'actualisation 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 et le type de valeurs reçues (nombres à virgule flottante, 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/log est impossible à prévoir exactement, mais vous pouvez compter sur environ 500 octets par valeur.

- Paramètre du housekeeper pour les tendances

Zabbix conserve, pour chaque élément, un ensemble de valeurs max/min/moyenne/nombre sur 1 heure 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 pour chaque total. Supposons que nous souhaitions conserver les données de tendance 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 résolu, un enregistrement event_recovery est créé. Normalement, la plupart des événements seront résolus, nous pouvons donc supposer un enregistrement event_recovery par événement. Cela signifie 80 octets supplémentaires par événement.

Les événements peuvent éventuellement 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. Chacun nécessitera donc un espace disque supplémentaire de #tags * 100 octets.

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 pouvant être utilisées pour calculer l'espace disque requis pour le système Zabbix :

Paramètre	Formule de l'espace disque requis (en octets)
Configuration Zabbix	Taille fixe. Normalement 10MB ou moins.
Historique	days*(items/refresh rate)*24*3600*bytes items : nombre d'éléments days : nombre de jours de conservation de l'historique refresh rate : intervalle d'actualisation moyen des éléments bytes : nombre d'octets nécessaires pour conserver une seule valeur, dépend du moteur de base de données, normalement ~90 octets.
Tendances	days*(items/3600)*24*3600*bytes items : nombre d'éléments days : nombre de jours de conservation de l'historique bytes : nombre d'octets nécessaires pour conserver une seule tendance, dépend du moteur de base de données, normalement ~90 octets.
Événements	days*events*24*3600*bytes events : nombre d'événements par seconde. Un (1) événement par seconde dans le pire des cas. days : nombre de jours de conservation de l'historique bytes : nombre d'octets nécessaires pour conserver un seul événement, dépend du moteur de base de données, normalement ~330 + nombre moyen de tags par événement * 100 octets.

Ainsi, l'espace disque total requis peut être calculé comme suit :

Configuration + Historique + Tendances + Événements

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, selon les 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.

Exigences réseau

La liste suivante des ports ouverts par composant s'applique à la configuration par défaut.

Composants	Port
Frontend	http sur 80, https sur 443
Serveur	10051 (à utiliser avec le proxy/les agents actifs)
Proxy actif	10051
Proxy passif	10051
Agent2	10050
Trapper	
Java gateway	10052
Service Web	10053

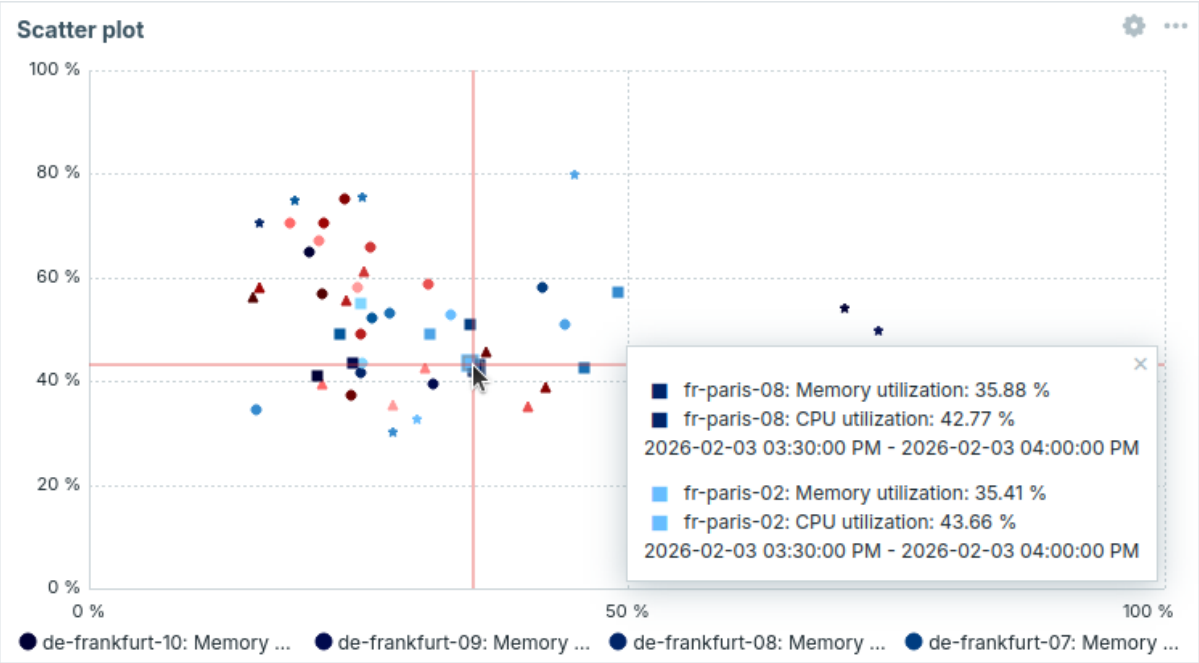
Note:
Les numéros de port doivent être ouverts dans le pare-feu afin de permettre les communications externes avec Zabbix. Les connexions TCP sortantes ne nécessitent généralement pas de paramètres explicites dans le pare-feu.

2 Nouveautés de Zabbix 8.0

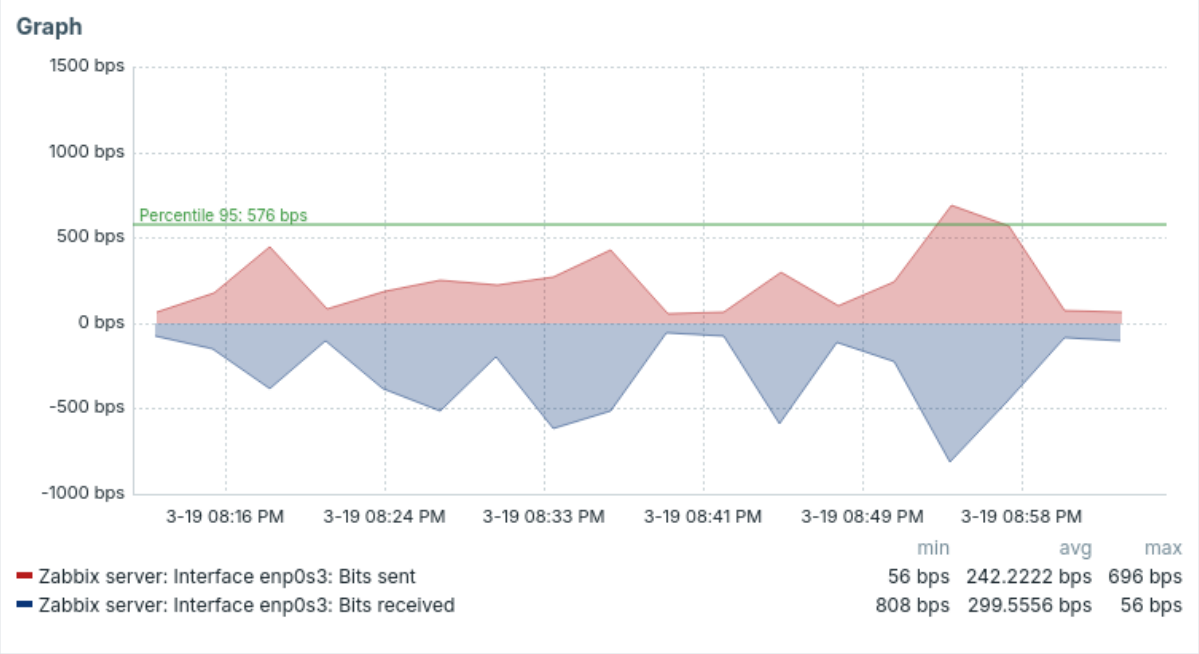
Zabbix 8.0.0 fait suite à Zabbix 7.4.0, en ajoutant de nouvelles fonctionnalités et améliorations.

Consultez les changements incompatibles pour cette version.

Widget de nuage de points Le widget Scatter plot a été ajouté aux widgets de tableau de bord. Il affiche la relation entre deux métriques en traçant des points de données individuels le long des axes X et Y. Cela permet de révéler des motifs, des regroupements, des corrélations et des valeurs aberrantes dans l'ensemble de données.



Y-axis inversion for the Graph widget Graphs in the Graph widget can now be displayed with inverted Y-axis values. A new *Invert values* setting allows you to multiply Y-axis values by −1 without altering the original data.



See also: Graph widget enhancements.

Modèles

Nouveaux modèles

- Aruba CX 8300s by SNMP, un modèle fournissant une supervision basée sur SNMP pour la série de commutateurs Aruba CX 8300.
- L'ensemble de modèles AWS by HTTP a été complété par le modèle *AWS Backup Vault by HTTP*.
- L'ensemble de modèles Azure by HTTP a été complété par le modèle *Azure Sentinel by HTTP*.
- Ciena 3906 by SNMP, un modèle pour superviser les appareils Ciena 3906.
- Cisco Secure Firewall Threat Defense by HTTP, un modèle offrant des capacités de supervision pour les appareils Cisco Secure Firewall Threat Defense à l'aide de l'API REST.
- Cradlepoint NCM v2 by HTTP, un ensemble de modèles pour superviser Cradlepoint NCM v2 et ses appareils via HTTP.
- Domain RDAP by HTTP, un modèle pour superviser les données d'enregistrement de domaine via le protocole RDAP.
- MariaDB by ODBC, un modèle pour superviser les bases de données MariaDB via ODBC.
- Microsoft Hyper-V Failover Cluster by SSH et Microsoft Hyper-V Standalone by SSH, des modèles pour superviser les clusters Microsoft Hyper-V et les hôtes autonomes via SSH.
- OpenAI Platform by HTTP, un modèle pour superviser la plateforme de développement d'OpenAI.
- Ribbon SBC Edge by HTTP, un modèle pour superviser les appareils Ribbon SBC Edge (anciennement SWe Lite) via HTTP.
- Ribbon SBC SWe Core by HTTP, un modèle pour superviser les appareils SBC SWe Core via HTTP, et Ribbon SBC SWe CE by HTTP, un modèle pour superviser les instances Ribbon SBC SWe Call Engine (CE) via HTTP.
- Stormshield SNS by SNMP, un modèle pour superviser les appareils Stormshield Network Security (SNS) via SNMP.
- VeloCloud SD-WAN Edge by HTTP, un modèle pour superviser les appareils VeloCloud SD-WAN Edge via HTTP.
- Vyatta Virtual Router by SNMP, un modèle pour superviser le routeur virtuel Vyatta 1908e.

Modèles mis à jour

- Ciena 3906 by SNMP a été mis à jour pour inclure des éléments de système de fichiers et de charge CPU.
- L'ensemble de modèles GCP by HTTP a été complété par le modèle *GCP Cloud Run Service by HTTP*.
- GitHub organization by HTTP a été mis à jour pour inclure des éléments de supervision de Microsoft Copilot.
- Proxmox VE by HTTP a été mis à jour avec la fonctionnalité LLD imbriquée. De plus, le format d'unité des éléments affichant des pourcentages a été ajusté pour plus de clarté.
- Microsoft 365 reports by HTTP a été mis à jour pour inclure des éléments de supervision de Microsoft Copilot.
- MySQL by Zabbix agent, MySQL by Zabbix agent 2, MySQL by Zabbix agent active, MySQL by Zabbix agent 2 active et MySQL by ODBC ont été mis à jour pour prendre en charge à la fois `SHOW SLAVE STATUS` (ancienne syntaxe) et `SHOW REPLICA STATUS` (nouvelle syntaxe).
- MySQL by ODBC a également été mis à jour avec de nouvelles métriques, des règles de découverte pour les tables et les réplicas, ainsi qu'un tableau de bord amélioré.
- VeloCloud SD-WAN by HTTP, précédemment *VMWare SD-WAN VeloCloud by HTTP*, a été renommé et mis à jour pour s'aligner sur la dernière plateforme VeloCloud SD-WAN.
- Vyatta Virtual Router by SNMP a été mis à jour avec de nouveaux éléments OID et des améliorations du tableau de bord.

Éléments

Type de données JSON Zabbix prend désormais en charge JSON comme type de données pour les valeurs d'élément.

Auparavant, les valeurs JSON étaient collectées par des éléments de type texte et stockées sous forme de chaînes avec une limite de 64KB. Désormais, Zabbix peut stocker les valeurs JSON nativement avec une limite de 128MiB, et également rejeter les valeurs JSON non valides (par exemple, contenant des clés non entre guillemets, des virgules finales ou des crochets non correspondants).

Le type de données JSON est pris en charge par tous les types d'éléments et prototypes d'éléments (sauf *Calculated*), et est disponible dans l'exportation des données en temps réel et les connecteurs. Les valeurs JSON peuvent être stockées dans toutes les bases de données prises en charge et dans Elasticsearch. Si vous utilisez TimescaleDB, veuillez consulter les notes de mise à niveau.

Notez que les éléments JSON ne peuvent pas être utilisés dans les déclencheurs ; toutefois, vous pouvez extraire des champs JSON avec des éléments dépendants ayant un type de données non JSON et les utiliser dans des déclencheurs.

Les éléments qui renvoient une chaîne JSON (`net.if.discovery`, `vfs.file.get`, etc.) restent des éléments de type texte ; toutefois, vous pouvez les convertir en JSON si nécessaire.

Pour plus de détails et les limites des données JSON, consultez la configuration des éléments.

Paramètre *type* pour la découverte des disques S.M.A.R.T L'élément smart.disk.discovery (plugin S.M.A.R.T. de l'agent Zabbix 2) accepte désormais un paramètre **type** facultatif permettant de spécifier une valeur à rechercher lors de l'analyse des disques.

Plugins

Plugin Ceph Ce plugin fonctionne désormais selon deux modes :

- **native** - Ce mode utilise la bibliothèque go-ceph pour communiquer directement avec le cluster Ceph à l'aide de l'API Ceph native (protocole msgr2). Il s'agit du mode recommandé pour les installations Ceph modernes, mais il est pris en charge **uniquement sous Linux** et à partir de Ceph 16.
- **restful** (obsolète) - Ce mode utilise l'API RESTful de Ceph pour la communication. Il s'agit du mode par défaut pour assurer la rétrocompatibilité, mais il ne fonctionnera pas avec Ceph version 20 (Tentacle) ou ultérieure en raison de la suppression du module mgr/restful.

Le mode à utiliser est déterminé par la valeur du paramètre mode (native/restful) :

- `Plugins.Ceph.Default.Mode=native` - définit le mode natif pour le plugin
- `Plugins.Ceph.Sessions.<SessionName>.Mode=native` - définit le mode natif pour la session nommée

Notez que l'ensemble des identifiants utilisateur diffère selon le mode et qu'ils ne sont pas compatibles entre eux. Le paramètre `Plugins.Ceph.InsecureSkipVerify` est ignoré en mode natif, car la sécurité de la connexion est définie côté cluster Ceph par le protocole msgr2 (sécurisé par défaut).

Notez que le plugin Ceph pour Zabbix agent 2 est désormais un plugin chargeable et nécessite des étapes d'installation supplémentaires. Cela est dû à l'exigence du paquet librados (pour le mode natif). Consultez le readme du plugin Ceph pour plus de détails.

Plugin MongoDB Le plugin MongoDB de Zabbix agent 2 offre plusieurs améliorations : - Prise en charge complète de l'analyse des URI MongoDB conventionnelles, avec prise en charge des schémas `mongodb://` et `mongodb+srv://`. - Découverte automatique des nœuds d'un ensemble de réplicas MongoDB, améliorant la capacité à superviser les clusters MongoDB. - Prise en charge de l'authentification x509, permettant des connexions sécurisées à MongoDB à l'aide de certificats client.

Plugin Oracle Le plugin Oracle de Zabbix agent 2 prend désormais en charge les connexions chiffrées aux bases de données Oracle à l'aide du protocole TCPS (TLS). Cela permet de superviser des instances Oracle via des sockets sécurisés par TLS, améliorant ainsi la sécurité de la supervision à distance.

Plugin Redis — prise en charge de TLS et validation au démarrage La prise en charge de TLS a été ajoutée au plugin Redis pour Zabbix agent 2.

La validation au démarrage de la configuration TLS du plugin a été mise en œuvre et les messages de validation/d'erreur ont été améliorés. Une logique de configuration invalide (par exemple : utilisation du type de connexion `verify_full` sans spécifier `TLSCAFile`) peut empêcher Zabbix agent 2 de démarrer.

Mode d'exécution de test pour les plugins chargeables Les plugins chargeables peuvent désormais être lancés en mode test à l'aide de l'option `-t` (`--test`), en passant une clé d'élément comme argument. Dans ce mode, le plugin s'exécute à des fins de débogage et de développement, et les fichiers de configuration du plugin sont ignorés.

Découverte de bas niveau

Case à cocher Convert to JSON dans les formulaires de découverte Une nouvelle case à cocher *Convert to JSON* a été ajoutée au formulaire de règle de découverte de bas niveau et au formulaire de prototype de découverte, qui s'affiche si « HTTP agent » est sélectionné dans la liste déroulante *Type*. Cette option permet de marquer les données récupérées pour une conversion automatique en JSON avant tout traitement ultérieur.

Prise en charge des macros pour la découverte de bas niveau imbriquée Les macros de découverte de bas niveau sont désormais prises en charge dans les règles de découverte de bas niveau imbriquées, dans :

- les paramètres de prétraitement JSONPath
- le champ JSONPath pour les macros LLD personnalisées

Balises modifiables pour les déclencheurs créés par la découverte de bas niveau Les déclencheurs créés à partir de prototypes de déclencheur peuvent désormais recevoir des balises ajoutées manuellement. Les balises héritées des prototypes de déclencheur sont toujours appliquées automatiquement. Les balises ajoutées manuellement peuvent être modifiées sur les déclencheurs découverts et seront incluses dans les tableaux de balises d'événement et disponibles pour les fonctionnalités prenant en charge les balises, telles que les filtres, les widgets de tableau de bord et les macros de notification.

Processus

Mise en cache et réutilisation de l'EngineID SNMPv3 Zabbix met désormais en cache les correspondances EngineID SNMPv3 → IP et tentera de réutiliser les EngineID mis en cache pour les vérifications SNMPv3 suivantes, réduisant ainsi le trafic de sondage et améliorant les performances du poller. Si un EngineID réutilisé ne répond pas, le poller revient à une sonde EngineID et peut supprimer les entrées obsolètes après des modifications d'interface ou des échecs persistants.

Limitation affinée du proxy pendant la récupération du cache d'historique La logique de limitation du proxy a été affinée afin d'améliorer la stabilité du serveur pendant la récupération du cache d'historique.
Lorsque l'utilisation du cache d'historique atteint le seuil de limitation, le serveur continue de ne plus accepter les données du proxy comme auparavant.
Lorsque l'utilisation du cache redescend à 60 %, le serveur commence à traiter la liste de limitation, mais peut encore rejeter les envois du proxy contenant de très gros lots (environ plus de 10 k enregistrements) jusqu'à ce que la pression sur le cache diminue davantage.
Cette modification réduit le risque de surcharges répétées du cache pendant la récupération du serveur.

Valeur maximale du délai d'attente augmentée pour zabbix_get et zabbix_js La valeur maximale du paramètre `timeout` des utilitaires en ligne de commande zabbix_get et zabbix_js a été portée à 600 secondes.

Nettoyage des événements par le housekeeper pour les déclencheurs supprimés Lorsqu'un déclencheur est supprimé, le housekeeper supprime désormais également tous les événements associés aux problèmes de ce déclencheur. Auparavant, seuls les problèmes du déclencheur étaient supprimés, tandis que les événements n'étaient supprimés qu'après l'expiration de la *période de conservation des données des déclencheurs* du housekeeper.

Fermeture manuelle — les tags de déclencheur sont hérités par les événements de rétablissement Les événements de rétablissement créés après une fermeture manuelle héritent des tags du déclencheur en plus des tags de l'élément et de l'hôte. Ces tags sont présents dans le tableau des tags de l'événement et sont disponibles pour les macros de notification telles que `{EVENT.RECOVERY.TAGS}` et `{EVENT.RECOVERY.TAGSJSON}`.

Mise en cache des requêtes DNS pour les démons Zabbix Zabbix server, Zabbix proxy et Zabbix agent prennent désormais en charge l'utilisation du résolveur c-ares pour toutes les requêtes DNS, offrant la mise en cache des requêtes DNS et un basculement amélioré du résolveur en cas de défaillance lorsque Zabbix est compilé avec `--with-ares`. La mise en cache des requêtes DNS nécessite c-ares 1.26.0 ou une version ultérieure.

Prise en charge de la compilation avec c-ares sous Windows L'agent Zabbix peut désormais être compilé avec le résolveur c-ares sous Microsoft Windows. c-ares peut être installé via vcpkg, et la compilation de l'agent prend en charge soit `ARES=<vcpkg prefix>`, soit des chemins `ARESINCDIR`/`ARESLIBDIR` distincts pour les répertoires d'inclusion et de bibliothèque.

Authentification

Certificats SAML importables pour l'authentification unique Les utilisateurs Super admin peuvent désormais importer directement des certificats et des clés privées dans l'interface pour la configuration de SAML. Trois nouveaux champs ont été ajoutés à *Administration* > *Authentication* > *SAML* :

- *Certificat IdP* - certificat X.509 présenté par le fournisseur d'identité
- *Certificat SP* - certificat du fournisseur de services utilisé pour les échanges SAML
- *Clé privée SP* - clé privée correspondant au certificat SP

Ces contrôles permettent d'ajouter de nouvelles valeurs ou de modifier les valeurs existantes depuis l'onglet des paramètres SAML. Les certificats et les clés privées sont validés avant d'être enregistrés dans le backend de stockage choisi ; les valeurs invalides ou mal formées sont rejetées avec un message d'erreur explicatif.

Widgets

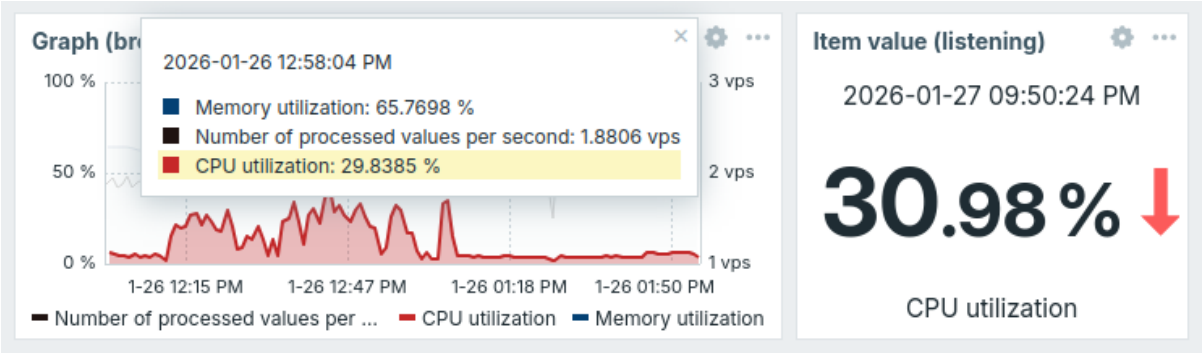
Regroupement des marqueurs d'hôtes sur la carte géographique Le widget Geomap prend désormais en charge un paramètre *Clustering* permettant de contrôler la manière dont les marqueurs d'hôtes proches sont regroupés en un seul marqueur avec un compteur. En définissant le niveau de zoom de la carte pour le regroupement, vous pouvez conserver la lisibilité des grandes cartes et maintenir une visibilité précise lors d'un zoom avant.

Colonnes agrégées dans Top items Le widget Top items peut désormais regrouper des modèles d'éléments dans une seule colonne ou ligne agrégée, ce qui facilite la présentation de métriques combinées (par exemple : le trafic entrant total sur toutes les interfaces réseau d'un hôte). Trois nouvelles options ont été ajoutées lors de la configuration des colonnes d'éléments : *Aggregate columns*, *Column aggregation function* et *Combined column name*.

Améliorations du widget Graph Le widget Graph offre plusieurs améliorations en matière de convivialité :

- Les éléments de l'infobulle des données du graphique, qui apparaît lorsque vous survolez le graphique, sont triés par valeur dans l'ordre décroissant.
- Lorsque vous survolez un élément dans l'infobulle, son graphique est mis en évidence tandis que les autres sont estompés.
- La sélection d'un élément dans l'infobulle diffuse ses données vers d'autres widgets prenant en charge leur réception.
- La limite du nombre d'éléments affichés dans l'infobulle a été supprimée.

- Un nouveau paramètre *Host names in labels* vous permet de choisir si les noms d'hôte sont affichés dans l'infobulle et dans la légende du graphique.



Interface web

Nouvelles polices et polices intégrées Zabbix inclut désormais de nouvelles polices intégrées qui améliorent la lisibilité, se chargent plus rapidement et permettent d'afficher des langues supplémentaires dans l'interface avec un impact minimal sur la mise en page. Comme elles sont intégrées, ces polices garantissent également une apparence cohérente sur tous les systèmes.

Nouvelles polices :

Linux by Zabbix agent: Available memory in %: Memory utilization	Triggers 1	vm.memory.utilization
Linux by Zabbix agent: Total memory	Triggers 1	vm.memory.size[total]
Linux by Zabbix agent: Total swap space	Triggers 1	system.swap.size[,total]

Anciennes polices :

Linux by Zabbix agent: Available memory in %: Memory utilization	Triggers 1	vm.memory.utilization
Linux by Zabbix agent: Total memory	Triggers 1	vm.memory.size[total]
Linux by Zabbix agent: Total swap space	Triggers 1	system.swap.size[,total]

Les nouvelles polices sont utilisées dans presque tous les thèmes. Si nécessaire, les thèmes *Blue (classic)* et *Dark (classic)* nouvellement ajoutés sont disponibles avec les anciennes polices.

La famille de polices utilisée pour le contenu à chasse fixe et les graphiques reste inchangée.

Validation en ligne Les formulaires suivants dans l'interface web ont été ajoutés à l'ensemble des formulaires prenant en charge la validation en ligne :

- Jeton d'API
- Authentification
- Enregistrement automatique
- Connecteur
- Cartes géographiques
- Groupe d'hôtes
- Prototypes d'hôte
- Mappage d'icônes
- Images
- Maintenance
- Type de média
- Notifications
- Profil
- Proxy
- Expressions régulières
- Service
- SLA
- Groupe de modèles

- Macros utilisateur globales
- Mettre à jour le problème
- Utilisateurs

Les erreurs de saisie sont affichées immédiatement après le remplissage des champs, ce qui améliore la convivialité et réduit les erreurs de configuration.

Formulaires modaux Dans la configuration de la découverte de bas niveau, le formulaire de configuration du prototype d'hôte s'ouvre désormais dans une fenêtre modale (pop-up).

Balises héritées visibles dans les hôtes, modèles, éléments et déclencheurs Les balises héritées sont désormais affichées et renvoyées de manière cohérente dans les modèles, hôtes, éléments, scénarios web et déclencheurs. Les balises héritées des chaînes modèle/hôte sont visibles dans *Monitoring > Latest data*. Le filtrage par balises héritées est disponible partout où le filtrage par balises est pris en charge — y compris dans les sections *Monitoring* > *Latest data* et *Data collection*, ainsi que dans tous les widgets de tableau de bord qui permettent le filtrage par balises d'hôte, d'élément, de déclencheur ou de scénario web — afin que la sélection basée sur les balises et le sous-filtrage se comportent de la même manière, quel que soit l'endroit où une balise a été définie.

L'onglet *Tags* des formulaires de configuration des modèles, hôtes et prototypes d'hôte expose désormais un bouton radio permettant de choisir la manière dont les balises sont présentées : les modèles affichent *Template tags* / *Inherited and template tags*, et les hôtes ainsi que les prototypes d'hôte affichent *Host tags* / *Inherited and host tags*. Les balises héritées sont distinguées visuellement par une nouvelle icône de document contourée à côté de l'étiquette de la balise.

Les widgets Graph et Pie chart incluent un nouveau paramètre *Item tags*.

Positionnement des infobulles Les infobulles peuvent désormais être repositionnées en les faisant glisser vers un nouvel emplacement. Cela s'applique, par exemple, à l'infobulle du Graph, ainsi qu'aux infobulles avec descriptions dans les sections Latest Data ou Problems.

Documentation

Pages de documentation consolidées pour les versions mineures La documentation des versions mineures d'une version majeure de Zabbix sera désormais regroupée dans des pages de documentation uniques pour les nouvelles fonctionnalités et les notes de mise à niveau respectivement.

Nouveautés de Zabbix 8.0.x

Cette page fournit des informations regroupées sur les nouvelles fonctionnalités incluses dans les versions mineures de la version majeure de Zabbix.

Voir également Nouveautés de la version majeure.

Pour les modifications apportées aux modèles existants et les informations sur les nouveaux modèles couvrant divers appareils, services et solutions partenaires, consultez Modifications des modèles.

Nouveautés de Zabbix 8.0.1 Cette version n'est pas encore publiée.

3 Appliance Zabbix

Aperçu L'appliance Zabbix permet de déployer instantanément le serveur Zabbix et l'interface web, au lieu de les configurer manuellement ou de réutiliser un serveur existant pour Zabbix.

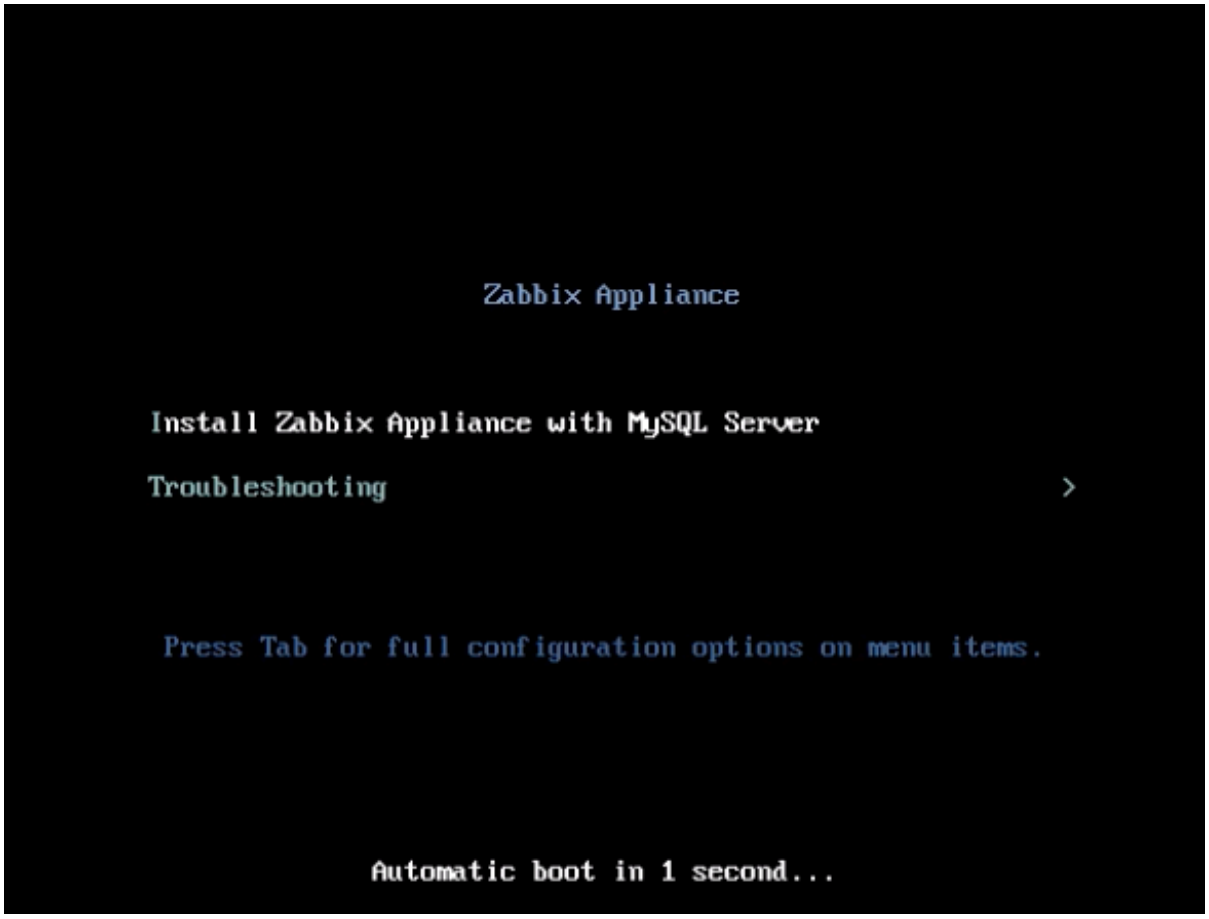
L'appliance est basée sur AlmaLinux 8 (x86_64) et contient un serveur Zabbix préconfiguré fonctionnant sur MySQL ainsi qu'une interface web fonctionnant sur le serveur web Nginx.

> **Attention:**
> Cette appliance est conçue pour évaluer Zabbix. Son utilisation dans des environnements de production critiques est déconseillée.

Les images de l'appliance sont disponibles en téléchargement dans les formats suivants :

- CD/DVD d'installation (.iso)
- VMware (.vmx) - voir les notes
- Format de virtualisation ouvert (.ovf)
- Microsoft Hyper-V (.vhd/.vhdx) - voir les notes
- KVM, Parallels, QEMU, clé USB, VirtualBox, Xen (.raw) - voir les notes
- 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.
2. Configurez les paramètres réseau de la machine virtuelle pour 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 accédez à 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

Frontend 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 celui 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 sous le compte root.

Serveur Zabbix :

- Nom d'utilisateur : zabbix_srv
- Mot de passe : le mot de passe est stocké dans `/etc/zabbix/zabbix\_server.conf`

Frontend Zabbix :

- Nom d'utilisateur : zabbix_web
- Mot de passe : le mot de passe est stocké dans `/etc/zabbix/web/zabbix.conf.php`

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 au frontend

Le frontend 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 que root user 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 :

- Connectez-vous en tant qu'utilisateur root.
- Exécutez les commandes suivantes en remplaçant les valeurs par vos adresses IP personnalisées :

```
nmcli connection modify eth0 ipv4.addresses 192.168.1.10/24    # Adresse IP de l'appliance/préfixe CIDR
nmcli connection modify eth0 ipv4.gateway 192.168.1.1     # Adresse IP de la passerelle
nmcli connection modify eth0 ipv4.dns 8.8.8.8    # Adresse IP du serveur DNS
nmcli connection modify eth0 ipv4.method manual
systemctl restart network
```

Vous pouvez également ouvrir le fichier `/etc/NetworkManager/system-connections/eth0.nmconnection` et ajouter les lignes suivantes :

```
[ipv4]
address1=192.168.1.10/24,192.168.1.1
dns=8.8.8.8
method=manual
```

Après avoir enregistré le fichier modifié, exécutez la commande `systemctl restart network`.

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 se trouvent dans `/etc/zabbix`
- Les fichiers journaux du serveur, du proxy et de l'agent Zabbix se trouvent dans `/var/log/zabbix`
- L'interface web 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 une utilisation dans 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. Sinon, une fois la conversion terminée, vous pouvez 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)* lorsque vous essayez de vous connecter à l'interface web, cela peut indiquer que l'installation est toujours en cours.



Si l'erreur persiste après quelques minutes d'attente, ou si vous observez tout 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 d'essayer de réparer manuellement une installation défectueuse, car cela peut entraîner d'autres complications.

4 Processus Zabbix

Veuillez utiliser la barre latérale pour accéder au contenu de la section des processus Zabbix.

1 Serveur

Aperçu

Le serveur Zabbix est le processus central du logiciel Zabbix.

Le serveur effectue l'interrogation et la réception des données, calcule les déclencheurs et envoie des notifications aux utilisateurs. Il s'agit du composant central auquel les agents et les proxies Zabbix transmettent les données sur la disponibilité et l'intégrité des systèmes. Le serveur peut également vérifier à distance les services réseau (tels que les serveurs web et les serveurs de messagerie) à l'aide de contrôles de service simples.

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 est divisé en trois composants distincts ; il s'agit de : serveur Zabbix, interface web et stockage de 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 de la base de données. Ensuite, environ une fois par minute, le serveur Zabbix interroge la table items pour obtenir la 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 les modifications effectuées dans l'interface Zabbix n'apparaissent dans la section des dernières données.

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
```

Démarrage manuel

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 :

```
-c --config <file>              chemin vers le fichier de configuration (la valeur par défaut est /usr/loc
-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 cette aide
-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 d'exécution

Options de contrôle d'exécution :

Option	Description	Cible
config_cache_reload	Recharger le cache de configuration. Ignoré si le cache est en cours de chargement.	
history_cache_clear=<**target**>	Vider le cache d'historique pour l'élément spécifié par son ID. Affecte toutes les valeurs de l'élément, sauf la première et la dernière.	**target** - ID de l'élément
diaginfo[=<**section**>]	Collecter des informations de diagnostic dans le fichier journal du serveur.	**historycache** - statistiques du cache d'historique **valuecache** - statistiques du cache de valeurs **preprocessing** - statistiques du gestionnaire de prétraitement **alerting** - statistiques du gestionnaire d'alertes **lld** - statistiques du gestionnaire LLD **locks** - liste des mutex (vide sur les systèmes *BSD*) **connector** - statistiques des connecteurs avec la plus grande file d'attente
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 veille 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 proxys
secrets_reload	Recharger les secrets depuis Vault.	
service_cache_reload	Recharger le cache du gestionnaire de services.	
snmp_cache_reload	Recharger le cache SNMP — effacer 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 de problèmes SNMP.	
housekeeper_execute	Démarrer la procédure de nettoyage. Ignoré si la procédure de nettoyage est actuellement en cours.	

Option	Description	Cible
trigger_housekeeper_execute	Exécuter la procédure de nettoyage des déclencheurs pour les services afin de supprimer les problèmes causés par des déclencheurs qui ont depuis été supprimés, y compris les problèmes de service générés par ces problèmes (considérés comme résolus au moment du nettoyage). Notez que, tant que la procédure de nettoyage n'est pas lancée, les problèmes causés par des déclencheurs désormais 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 de service basées sur des déclencheurs fréquemment découverts/supprimés, envisagez d'augmenter la fréquence de la procédure de nettoyage des déclencheurs en ajustant le paramètre de configuration du serveur ProblemHousekeepingFrequency. Ignoré si la procédure de nettoyage 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 ex., poller) Voir tous les types de processus du serveur. **process type,N** - Type de processus et numéro (par ex., poller,3) **pid** - Identifiant du processus (1 à 65535). Pour des valeurs plus grandes, spécifiez la cible comme '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*.	
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 ex. history syncer) Types de processus pris en charge comme cibles de 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, preprocessing worker, proxy poller, self-monitoring, service manager, snmp trapper, task manager, timer, trapper, unreachable poller, vmware collector **process type,N** - Type de processus et numéro (par ex., history syncer,1) **pid** - Identifiant du processus (1 à 65535). Pour des valeurs plus grandes, spécifiez la cible comme 'process type,N'. **scope** - `rwlock`, `mutex`, `processing` peuvent être utilisés avec le type et le numéro du processus (par ex., history syncer,1,processing) ou tous les processus du type (par ex., history syncer,rwlock)

Option	Description	Cible
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 ex. history syncer) Types de processus pris en charge comme cibles de profilage : voir `prof_enable` **process type,N** - Type de processus et numéro (par ex., history syncer,1) **pid** - Identifiant du processus (1 à 65535). Pour des valeurs plus grandes, spécifiez la cible comme 'process type,N'.

Exemple d'utilisation du contrôle d'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 d'exécution pour recharger la configuration du proxy :

```
### Recharger la configuration de tous les proxys :
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 d'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 d'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 d'exécution pour recharger le cache SNMP :

```
zabbix_server -R snmp_cache_reload
```

Attention:
Lorsqu'une interface SNMPv3 est mise à jour via l'interface utilisateur de Zabbix, Zabbix recharge automatiquement les nouveaux identifiants SNMPv3 de cette interface dans la plupart des cas ; utilisez `-R snmp_cache_reload` uniquement si l'interrogation échoue toujours après les modifications des identifiants (par exemple, en raison d'incohérences engineBoots/engineID ou d'appareils non conformes à la RFC), ou lorsque vous devez forcer un effacement global du cache SNMP à des fins de dépannage.

Exemple d'utilisation du contrôle d'exécution pour déclencher l'exécution du housekeeper :

```
zabbix_server -c /usr/local/etc/zabbix_server.conf -R housekeeper_execute
```

Exemples d'utilisation du contrôle d'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 en tant qu'utilisateur non root. Il s'exécutera avec n'importe quel utilisateur non root sous lequel il est démarré. Vous pouvez donc exécuter le serveur avec n'importe quel utilisateur non root sans aucun problème.

Si vous essayez de l'exécuter en tant que « root », il basculera vers l'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 sont exécutés sur la même machine, il est recommandé d'utiliser un utilisateur différent pour exécuter le serveur et pour exécuter l'agent. Sinon, si les deux sont exécutés avec 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

Consultez les options du fichier de configuration pour plus de détails 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` - collecteur 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 d'éléments
- `connector manager` - processus gestionnaire des connecteurs
- `connector worker` - processus de traitement des requêtes du connector manager
- `discovery manager` - processus gestionnaire de la découverte des périphériques
- `discovery worker` - processus de traitement des tâches de découverte du discovery manager
- `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 anciennes données historiques
- `http agent poller` - processus de collecte asynchrone pour les vérifications HTTP avec un thread de travail
- `http poller` - collecteur de supervision web
- `icmp pinger` - collecteur pour les vérifications icmpping
- `internal poller` - collecteur pour les vérifications internes
- `ipmi manager` - gestionnaire des collecteurs IPMI
- `ipmi poller` - collecteur pour les vérifications IPMI
- `java poller` - collecteur pour les vérifications Java
- `lld manager` - processus gestionnaire des tâches de découverte de bas niveau
- `lld worker` - processus de travail des tâches de découverte de bas niveau
- `odbc poller` - collecteur pour les vérifications ODBC
- `poller` - collecteur 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` - collecteur pour les proxies passifs
- `proxy group manager` - gestionnaire de l'équilibrage de charge et de la haute disponibilité des proxies
- `report manager`- gestionnaire des tâches de génération de rapports planifiés
- `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 des informations sur les problèmes, les tags de problème et la récupération des problèmes depuis history syncer, task manager et alert manager
- `snmp poller` - processus de collecte asynchrone pour les vérifications SNMP avec un thread de travail (éléments `walk[OID]` et `get[OID]` uniquement)

- `snmp trapper` - trapper pour les traps SNMP
- `task manager` - processus d'exécution à distance des tâches demandées par d'autres composants (par ex. fermer un problème, acquitter un problème, vérifier la valeur d'un élément maintenant, fonctionnalité de commande à distance)
- `timer` - temporisateur pour le traitement des maintenances
- `trapper` - trapper pour les vérifications actives, les traps, la communication avec les proxies
- `trigger housekeeper` - processus de suppression des problèmes générés par des déclencheurs qui ont été supprimés
- `unreachable poller` - collecteur pour les périphériques injoignables
- `vmware collector` - collecteur de données VMware chargé de la collecte des données depuis les 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 permissions de lecture-écriture pour le propriétaire du fichier uniquement. De plus, le fichier est lisible par le groupe propriétaire. Toutes les autres permissions sont refusées.

Différents types de processus du serveur Zabbix peuvent être supervisés à l'aide de l'élément interne **zabbix[process,<type>,<mode>,<st**
item.

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 temporisateurs.

Les durées, dans « processed...in N (<timings>) sec », sont :

- Temps consacré à l'écriture des valeurs d'élément dans la base de données ;
- Temps consacré à la mise à jour des données d'élément (état, erreurs, inventaire de l'hôte, etc.) ;
- Temps consacré à l'écriture des tendances dans la base de données ;
- Temps consacré au calcul des déclencheurs ;
- Temps consacré au traitement des événements et des actions.

Housekeeping procedure

The `housekeeper` process periodically removes outdated data (item history and trends, user sessions, events, etc.), as well as data left behind by deleted objects. It runs in cycles, with frequency and per-cycle deletion limit determined by `HousekeepingFrequency` and `MaxHousekeeperDelete`. Any data not removed in one cycle is carried over to the next. Automatic housekeeping can be enabled and configured in *Administration* > *Housekeeping*.

For removing data left behind by deleted objects, the `housekeeper` process relies on tasks from the `housekeeper` table, which is populated whenever an object is deleted. For example, when you delete a host, Zabbix also deletes its items, but not their history, trends, or problems. Instead, database triggers populate the `housekeeper` table with tasks consisting of these fields:

- `housekeeperid` - task ID
- `object` - object type (0 - item; 1 - trigger; 2 - service; 3 - discovered host; 4 - discovered service)
- `objectid` - object ID (helps the housekeeper find object-related data)

For example, deleting a host with two items and one trigger populates the `housekeeper` table as follows:

```
+---------------+--------+----------+
| housekeeperid | object | objectid |
+---------------+--------+----------+
|             1 |      1 |    28724 |
|             2 |      0 |    59396 |
|             3 |      0 |    59397 |
+---------------+--------+----------+
```

Database triggers populate the `housekeeper` table without checking for object-related data; that check is done by the `housekeeper` process.

Each task results in one or more `housekeeper` operations that depend on the object type:

- For items (including LLD rules) - removes data from all history and trends tables (`history`, `history_str`, `history_log`, `history_uint`, `history_text`, `history_bin`, `history_json`, `trends`, `trends_uint`) that contain values for those items. Also, checks the `problems` table and removes outdated internal events associated with those items.

- For triggers - checks event-related tables (`problem`, `event_symptom`, `event_recovery`, `events`) and removes outdated events that are associated with those triggers, and also notifies the `service manager` process about removed events.

Note:
A separate `trigger housekeeper` process handles a narrower task—removing problems and events that have no known source trigger. Its run frequency is controlled by `ProblemHousekeepingFrequency`.

 Until the trigger housekeeping procedure is started, problems caused by triggers that have since been deleted might still generate service problems and assign them to services. If your setup involves many service status calculation rules based on frequently discovered/undiscovered triggers, consider increasing the frequency of the housekeeping procedure by adjusting the `ProblemHousekeepingFrequency` server configuration parameter.

- For services - checks the `problems` table and removes outdated service events, as well as outdated service problems, thus resolving them at the time of housekeeping.
- For network discovery - removes outdated discovery events from the `problems` table.

The housekeeper removes only those events that are not associated with problems. For example, an outdated problem/recovery event will not be removed if it's associated to an open problem. When the housekeeper removes outdated entities, it first removes problems, then events.

Note:
Tables using `partition` mode (TimescaleDB partitioned tables), are skipped; only tables using `regular` mode are processed.

Plateformes prises en charge

En raison des exigences de sécurité et du caractère critique du fonctionnement du serveur, 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 principales versions du marché.

Le serveur Zabbix est testé sur les plateformes 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 modernes de type Unix utilisent des paramètres régionaux UTF-8 par défaut ; toutefois, sur certains systèmes, il peut être nécessaire de les définir explicitement.

1 Haute disponibilité

Aperçu

La haute disponibilité (HA) est généralement requise dans les infrastructures critiques qui ne peuvent se permettre pratiquement aucun temps d'arrêt. Ainsi, pour tout service susceptible d'être en erreur, une option de basculement doit être en place pour prendre le relais en cas d'échec du service actuel.

Zabbix propose une solution haute disponibilité **native** facile à configurer et ne nécessitant aucune expertise préalable en HA. Zabbix HA natif peut être utile pour une couche supplémentaire de protection contre les pannes logicielles/matérielles du serveur Zabbix ou pour avoir moins de temps d'arrêt en raison de la maintenance.

En mode haute disponibilité Zabbix, plusieurs serveurs Zabbix sont exécutés en tant que nœuds dans un cluster. Pendant qu'un serveur Zabbix du cluster est actif, d'autres sont en veille, prêts à prendre le relais si nécessaire.



Le passage à Zabbix HA est sans engagement. Vous pouvez revenir à un fonctionnement autonome à tout moment.

Voir aussi : Détails de la mise en œuvre

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 du cluster HA.

Il s'agit d'un identifiant de nœud unique (par exemple, `zabbix-node-01`) auquel le serveur fera référence dans les configurations des agents et des proxys. Si vous ne spécifiez pas HANodeName, le serveur sera démarré en mode autonome.

- Le paramètre **NodeAddress** doit être spécifié pour chaque nœud.

Le paramètre NodeAddress (adresse:port) sera utilisé par le frontend 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 apporté des modifications aux fichiers de configuration. Ils seront désormais démarrés en tant que nœuds de cluster. Le nouvel état des serveurs peut être consulté dans *Rapports* → *Informations système* et également en exécutant :

```
zabbix_server -R ha_status
```

Cette commande d'exécution consignera l'état actuel du cluster HA dans le journal du serveur Zabbix (et sur stdout) :

```
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 web

Assurez-vous que l'adresse:port du serveur Zabbix n'est **pas définie** dans la configuration de l'interface web (située dans `conf/zabbix.conf.php` du répertoire des fichiers de l'interface web).

```
// Uncomment and set to desired values to override Zabbix hostname/IP and port.
// $ZBX_SERVER                  = '';
// $ZBX_SERVER_PORT             = '';
```

L'interface web 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 de cluster HA (serveurs) doivent être répertoriés dans la configuration du proxy Zabbix passif ou actif.

Pour un proxy passif, le nom des nœuds doivent être listés dans le paramètre Server du proxy, séparés par une **virgule**.

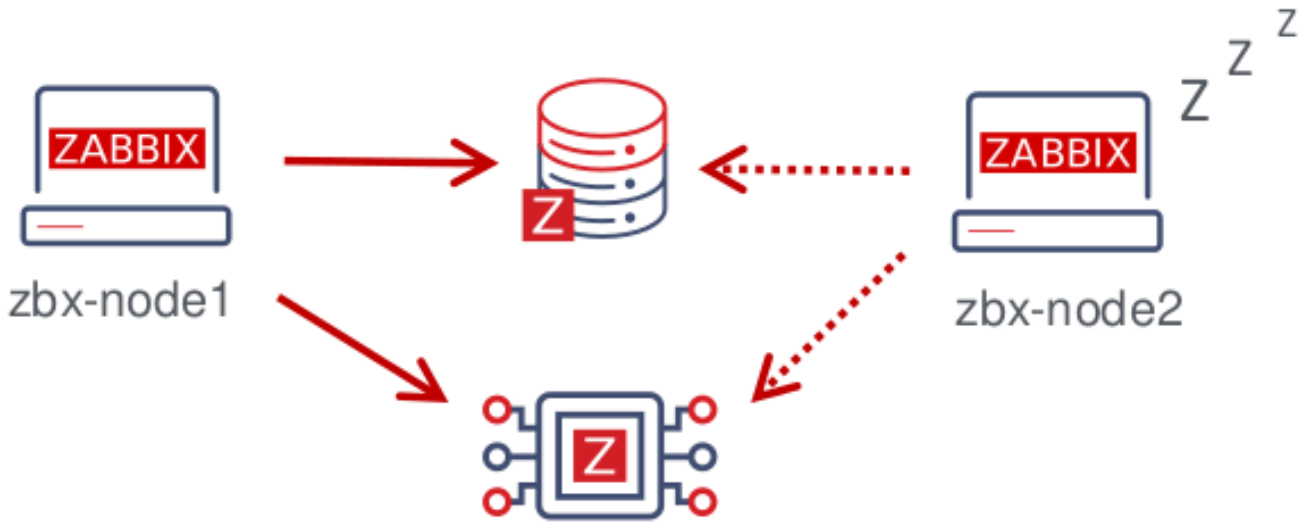
```
Server=zabbix-node-01,zabbix-node-02
```

Pour un proxy actif, le nom des nœud doivent être répertoriés dans le paramètre Server du proxy, séparés par un **point-virgule**.

```
Server=zabbix-node-01;zabbix-node-02
```

Configuration des agents

Les nœuds de cluster HA (serveurs) doivent être répertoriés dans la configuration de l'agent Zabbix ou de l'agent Zabbix 2.



Pour activer les vérifications passives, les noms de nœud doivent être répertoriés dans le paramètre du serveur, séparés par une **virgule**.

```
Serveur=zabbix-node-01,zabbix-node-02
```

Pour activer les vérifications actives, les noms de nœud doivent être répertoriés dans le paramètre ServerActive. Notez que pour les vérifications actives, les nœuds doivent être séparés par une virgule de tout autre serveur, tandis que les nœuds eux-mêmes doivent être séparés par un **point-virgule**, par exemple :

```
ServeurActive=zabbix-node-01;zabbix-node-02
```

Basculement vers le nœud de secours

Zabbix basculera automatiquement vers un autre nœud si le nœud actif s'arrête. Il doit y avoir au moins un nœud en état de secours pour que le basculement puisse avoir lieu.

À quelle vitesse le basculement se produira-t-il ? Tous les nœuds mettent à jour leur heure de dernier accès (et leur état, s'il a changé) toutes les 5 secondes. Ainsi :

- Si le nœud actif s'arrête et parvient à signaler son état comme « arrêté », un autre nœud prendra le relais dans les **5 secondes**.
- Si le nœud actif s'arrête/devient indisponible sans pouvoir mettre à jour son état, les nœuds de secours 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 d'exécution :

- `ha_status` - consigner l'état du cluster HA dans le journal du serveur Zabbix (et sur stdout)
- `ha_remove_node=target` - supprimer un nœud HA identifié par son <target> - nom ou ID du nœud (le nom/l'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/de secours ne peuvent pas être supprimés.

- `ha_set_failover_delay=delay` - définir le délai de basculement HA (entre 10 secondes et 15 minutes ; les suffixes de temps sont pris en charge, par exemple 10s, 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 d'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 de haute disponibilité.

Désactivation du cluster HA

Pour désactiver un cluster de haute disponibilité :

- faites des copies de sauvegarde des fichiers de configuration
- arrêtez les nœuds de secours
- supprimez le paramètre HANodeName du serveur primaire actif
- redémarrez le serveur primaire (il démarrera en mode autonome)

Mise à niveau du cluster HA

Pour effectuer une mise à niveau de version majeure des nœuds HA :

- arrêtez tous les nœuds ;
- créez une sauvegarde complète de la base de données ;
- 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 défaillante.
- sélectionnez un seul nœud qui effectuera la mise à niveau de la base de données, modifiez sa configuration en mode autonome en commentant HANodeName et mettez-le à niveau ;
- assurez-vous que la mise à niveau de la base de données est entièrement terminée (*Informations système* doit indiquer que le serveur Zabbix est en cours d'exécution) ;
- redémarrez le nœud en mode HA ;
- 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).

Pour une mise à niveau de version mineure, il suffit de mettre à niveau le premier nœud, de s'assurer qu'il a bien été mis à niveau et qu'il est en cours d'exécution, puis de lancer la mise à niveau sur le nœud suivant.

Détails de l'implémentation

Le cluster de haute disponibilité (HA) est une solution optionnelle et il est pris en charge pour le serveur Zabbix. La solution HA native est conçue pour être simple à utiliser, elle fonctionne entre plusieurs sites et n'a pas d'exigences spécifiques concernant 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 correspond 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 exécute 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 pas sur les ports ; il utilise un minimum de connexions à la base de données.

Les nœuds actifs et en veille mettent tous deux à jour leur heure du dernier accès toutes les 5 secondes. Chaque nœud en veille surveille l'heure du dernier accès du nœud actif. Si l'heure du dernier accès du nœud actif dépasse le nombre de secondes défini par le « failover delay », le nœud en veille bascule lui-même en nœud actif et attribue le statut « indisponible » 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 le statut des nœuds en veille : si l'heure du dernier accès d'un nœud en veille dépasse le nombre de secondes du « failover delay », le statut « indisponible » est attribué au nœud en veille.

Les nœuds sont conçus pour être compatibles entre les versions mineures de Zabbix.

2 Agent

Aperçu

L'agent Zabbix est déployé sur une cible de supervision afin de surveiller activement les ressources locales et les applications (disques durs, mémoire, statistiques du processeur, etc.).

L'agent collecte localement les informations de fonctionnement 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 de manière inattendue), 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 des informations statistiques.

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 ex. 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). Il récupère d'abord une liste d'éléments de supervision depuis le serveur (charge CPU, mémoire disponible, etc.), puis collecte les données requises et lui renvoie périodiquement de nouvelles valeurs.

Le type de vérification de l'agent est configuré en sélectionnant le type d'élément de supervision correspondant. L'agent Zabbix traite les éléments de type « Zabbix agent » ou « Zabbix agent (active) ».

Plateformes prises en charge

Pour connaître les plateformes 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 qui est surveillé.

Installation

L'agent Zabbix peut être installé sur des 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 de votre système d'exploitation et la version du système d'exploitation), puis suivez les instructions fournies.
- Sources Zabbix - téléchargez les fichiers source et compilez l'agent Zabbix en configurant les sources avec l'option `--enable-agent`.

Attention:
En général, les agents Zabbix 32 bits fonctionneront sur des systèmes 64 bits, mais peuvent échouer dans certains cas.

Note:
Des binaires précompilés de l'agent Zabbix sont disponibles en téléchargement pour macOS, IBM AIX, FreeBSD, OpenBSD et Solaris. Des binaires hérités, compatibles avec la version actuelle de Zabbix server/proxy, sont disponibles pour NetBSD et HP-UX.

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
```

Démarrage manuel

Vous pouvez démarrer l'agent Zabbix en localisant le binaire `zabbix_agentd` et en l'exécutant directement ; par exemple :

```
zabbix_agentd
```

Agent sur les systèmes Windows

L'agent Zabbix sur Windows s'exécute en tant que service Windows.

Installation

L'agent Zabbix peut être installé sur Windows à l'aide de l'une des méthodes suivantes :

- Binaires précompilés de l'agent Zabbix - téléchargez le package d'installation MSI de l'agent Zabbix et suivez les instructions de la page Installation de l'agent Windows à partir d'un MSI.
- Sources Zabbix - téléchargez les fichiers source et suivez les instructions de la page Compilation de l'agent Zabbix sous Windows.

Consultez la page Agent Zabbix sur Microsoft Windows pour plus de détails sur l'installation de l'agent Zabbix (à partir d'une archive ZIP) en tant que service Windows.

Options

Il est possible d'exécuter plusieurs instances de l'agent sur un hôte. Une seule instance peut utiliser le fichier de configuration par défaut ou un fichier de configuration spécifié dans la ligne de commande. En cas d'instances multiples, chaque instance d'agent doit avoir son propre fichier de configuration (l'une des instances peut utiliser le fichier de configuration par défaut).

Les paramètres de ligne de commande suivants peuvent être utilisés avec l'agent Zabbix :

Parameter	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 autre que celui par défaut. Sous UNIX, la valeur par défaut est /usr/local/etc/zabbix_agentd.conf ou celle définie par les variables de compilation *--sysconfdir* ou *--prefix* Sous Windows, la valeur par défaut est C:\Program Files\Zabbix Agent\zabbix_agentd.conf
-f --foreground	Exécuter l'agent Zabbix au premier plan (par défaut : true).
-p --print	Afficher les éléments connus et quitter. *Remarque* : 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 et quitter. *Remarque* : 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 et quitter.
-h --help	Afficher les informations d'aide.
-V --version	Afficher le numéro de version.
Agent UNIX uniquement	
-R --runtime-control <option>	Exécuter des fonctions administratives. Voir le contrôle d'exécution.
Agent Windows uniquement	
-m --multiple-agents	Utiliser plusieurs instances d'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.
-S --startup-type <value>	Définir le type de démarrage du service de l'agent Zabbix Windows. Valeurs autorisées : `automatic` - *(par défaut)* démarrer le service automatiquement au démarrage de Windows ; `delayed` - retarder le démarrage du service jusqu'à ce que les services démarrés automatiquement aient terminé leur démarrage (disponible sur 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 Windows en tant que service.
-d --uninstall	Désinstaller le service de l'agent Zabbix Windows.
-s --start	Démarrer le service de l'agent Zabbix Windows.
-x --stop	Arrêter le service de l'agent Zabbix Windows.

Exemples spécifiques d'utilisation des paramètres de ligne de commande :

- affichage de tous les éléments intégrés de l'agent avec leurs valeurs
- test d'un paramètre utilisateur avec la clé "mysql.ping" définie dans le fichier de configuration spécifié
- installation d'un service "Zabbix Agent" pour Windows en utilisant le chemin par défaut vers le fichier de configuration C:\Program Files\Zabbix Agent\zabbix_agentd.conf

- installation d'un service "Zabbix Agent [Hostname]" pour Windows en utilisant le fichier de configuration zabbix_agentd.conf situé dans le même dossier que l'exécutable de l'agent et en rendant le nom du service unique en l'étendant avec la valeur Hostname du fichier de configuration
- modification du type de démarrage d'un service "Zabbix Agent" installé pour Windows en utilisant le fichier de configuration zabbix_agentd.conf 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 d'exécution

Avec les options de contrôle d'exécution, vous pouvez modifier le niveau de journalisation des processus de l'agent.

Option	Description	Cible
log_level_increase[=target]	Augmenter le niveau de journalisation. Si la cible n'est pas spécifiée, tous les processus sont affectés.	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 de processus et numéro (par exemple, listener,3) **pid** - identifiant du processus (1 à 65535). Pour des valeurs plus grandes, spécifiez la cible comme 'process-type,N'.
log_level_decrease[=target]	Diminuer le niveau de journalisation. Si la cible n'est pas spécifiée, tous les processus sont affectés.	
userparameter_reload	Recharger 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 avec le PID 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 d'exécution n'est pas pris en charge sur OpenBSD, NetBSD et Windows.

Types de processus de l'agent

- `active checks` - processus effectuant des contrôles actifs
- `collector` - processus de collecte des données
- `listener` - processus à l'écoute des contrôles passifs

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 autorisations 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 autorisations sont refusées.

Utilisateur du processus

L'agent Zabbix sous UNIX 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 l'agent avec 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 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

Pour plus de détails sur la configuration de l'agent Zabbix, consultez les options du fichier de configuration pour zabbix_agentd ou l'agent 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 modernes de type Unix utilisent UTF-8 par défaut pour les paramètres régionaux ; toutefois, sur certains systèmes, il peut être nécessaire de le définir explicitement.

Code de sortie

L'agent Zabbix renvoie 0 en cas de sortie réussie et 1 en cas d'échec.

3 Agent 2

Aperçu

Zabbix agent 2 est une nouvelle génération de Zabbix agent, écrite en Go (avec une partie du code C réutilisée depuis Zabbix agent). Il est conçu pour :

- Réduire le nombre de connexions TCP.
- Offrir une concurrence des contrôles améliorée.
- Être facilement extensible avec des plugins, qui fournissent des contrôles simples avec un minimum de code et prennent en charge des contrôles complexes composés de scripts de longue durée d'exécution ainsi que la collecte autonome de données avec des rapports périodiques.
- Servir de remplacement à Zabbix agent, en prenant en charge toutes les fonctionnalités précédentes.

Vérifications passives et actives

Zabbix agent 2 prend en charge les vérifications passives et actives, de manière similaire à Zabbix agent. De plus, les vérifications actives de Zabbix agent 2 prennent en charge les intervalles flexibles/de planification 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 aléatoire conditionnel dans l'intervalle de mise à jour de l'élément afin d'éviter des pics d'utilisation des ressources. Pour exécuter immédiatement après le redémarrage de l'agent les vérifications actives qui n'ont pas d'intervalle de mise à jour *Scheduling* de planification, définissez le paramètre `ForceActiveChecksOnStart` (au niveau global) ou `Plugins.<Plugin name>.System.ForceActiveChecksOnStart` (n'affecte que les vérifications du plugin spécifique) dans le fichier de configuration. Le paramètre au niveau du plugin, s'il est défini, 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 le paramètre de configuration *Plugins*.

Plateformes prises en charge

Pour les plateformes 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 votre système d'exploitation et la version du système d'exploitation), puis suivez les instructions.
- Sources Zabbix - téléchargez les fichiers source et compilez l'agent en le configurant avec l'option `--enable-agent2`.

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.

Si installé en tant que paquet

Zabbix agent 2 s'exécute comme un 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 démonisation 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
```

Démarrage manuel

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 comme un processus autonome ; cependant, il peut également être exécuté en tant que service Windows.

Installation

Zabbix agent 2 peut être installé sur Windows à l'aide de l'une des méthodes suivantes :

- Binaires précompilés de Zabbix agent - téléchargez le package d'installation MSI de l'agent et suivez les instructions de la page Installation de l'agent Windows à partir d'un MSI.
- Sources de Zabbix - téléchargez les fichiers source et suivez les instructions de la page Compilation de Zabbix agent 2 sous 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 :

Parameter	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 par défaut. Sur UNIX, la valeur par défaut est /usr/local/etc/zabbix_agent2.conf ou celle définie par les variables de compilation *--sysconfdir* ou *--prefix* Sur Windows, la valeur par défaut est C:\Program Files\Zabbix Agent 2\zabbix_agent2.conf
-f --foreground	Exécuter Zabbix agent au premier plan (par défaut : true).
-p --print	Afficher les éléments connus et quitter. *Remarque* : 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 et quitter. *Remarque* : 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 et quitter.
-h --help	Afficher les informations d'aide et quitter.
-v --verbose	Afficher les informations de débogage. Utilisez cette option avec les options -p et -t.

Parameter	Description
-V --version	Afficher la version de l'agent et les informations de licence.
-R --runtime-control <option>	Exécuter des fonctions administratives. Voir runtime control.
Agent Windows uniquement	
-m --multiple-agents	Utiliser plusieurs instances d'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.
-S --startup-type <value>	Définir le type de démarrage du service de l'agent Zabbix Windows. Valeurs autorisées : `automatic` - *(par défaut)* démarrer le service automatiquement au démarrage de Windows ; `delayed` - retarder le démarrage du service après la fin du démarrage des services lancés automatiquement ; `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 Windows en tant que service.
-d --uninstall	Désinstaller le service de l'agent Zabbix Windows.
-s --start	Démarrer le service de l'agent Zabbix Windows.
-x --stop	Arrêter le service de l'agent Zabbix 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 C:\Program Files\Zabbix Agent 2\zabbix_agent2.conf
- modifier le type de démarrage d'un service "Zabbix Agent" installé pour Windows en utilisant le fichier de configuration zabbix_agent2.conf situé dans le même dossier que l'exécutable de l'agent

```
zabbix_agent2 --print
zabbix_agent2 -t "mysql.ping" -c /etc/zabbix/zabbix_agentd.conf
zabbix_agent2.exe -i
zabbix_agent2.exe -c zabbix_agent2.conf -S delayed
```

Contrôle d'exécution

Le contrôle d'exécution fournit certaines options de contrôle à distance.

Option	Description
log_level_increase	Augmenter le niveau de journalisation.
log_level_decrease	Diminuer le niveau de journalisation.
metrics	Lister les métriques disponibles.
version	Afficher la version de l'agent.
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 d'exécution.

Exemples :

- augmentation du niveau de journalisation pour l'agent 2
- afficher les options de contrôle d'exécution

```
zabbix_agent2 -R log_level_increase
zabbix_agent2 -R help
```

Fichier de configuration

Les paramètres de configuration de l'agent 2 sont pour la plupart compatibles avec Zabbix agent, à quelques exceptions près.

Nouveaux paramètres	Description
ControlSocket	Le chemin du socket de contrôle d'exécution. L'agent 2 utilise un socket de contrôle pour les commandes d'exécution.
EnablePersistentBuffer, *PersistentBufferFile*, *PersistentBufferPeriod*	Ces paramètres sont utilisés pour configurer le stockage persistant sur l'agent 2 pour les éléments actifs.
ForceActiveChecksOnStart	Détermine si l'agent doit effectuer les contrôles actifs immédiatement après le 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 contrôles pouvant être exécutés en même temps.
StatusPort	Le port sur lequel l'agent 2 écoutera les requêtes d'état HTTP et affichera une liste des plugins configurés ainsi que certains paramètres internes
Paramètres supprimés	**Description**
AllowRoot, *User*	Non pris en charge, car la démonisation n'est pas prise en charge.
LoadModule, *LoadModulePath*	Les modules chargeables ne sont pas pris en charge.
StartAgents	Ce paramètre était utilisé dans Zabbix agent pour augmenter la concurrence des contrôles passifs ou les désactiver. Dans l'agent 2, la concurrence est configurée au niveau du plugin et peut être limitée par un paramètre de capacité.

Pour plus de détails, consultez les options du fichier de configuration pour zabbix_agent2.

Codes de sortie

Zabbix agent 2 peut également être compilé avec d'anciennes versions 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.

Agent vs agent 2 comparison

This section describes the differences between the Zabbix agent and the Zabbix agent 2.

Parameter	Zabbix agent	Zabbix agent 2
Programming language	C	Go with some parts in C
Daemonization	yes	by systemd only (yes on Windows)
Supported extensions	Custom loadable modules in C.	Custom plugins in Go.
Requirements		
Supported platforms	Linux, IBM AIX, FreeBSD, NetBSD, OpenBSD, HP-UX, Mac OS X, Solaris: 9, 10, 11, Windows: all desktop and server versions since XP	Linux, Windows: all desktop and server versions, on which a supported Go version can be installed.
Supported crypto libraries	GnuTLS 3.1.18 and newer OpenSSL 1.0.1, 1.0.2, 1.1.0, 1.1.1, 3.0.x LibreSSL - tested with versions 2.7.4, 2.8.2 (certain limitations apply, see the Encryption page for details).	Linux: OpenSSL 1.0.1 and later. MS Windows: OpenSSL 1.1.1 or later. The OpenSSL library must have PSK support enabled. LibreSSL is not supported.
Monitoring processes		
Processes	A separate active check process for each server/proxy record.	Single process with automatically created threads. The maximum number of threads is determined by the GOMAXPROCS environment variable.

Parameter	Zabbix agent	Zabbix agent 2
Metrics	**UNIX**: see a list of supported items. **Windows**: see a list of additional Windows-specific items.	**UNIX**: All metrics supported by Zabbix agent. Additionally, the agent 2 provides Zabbix-native monitoring solution for: Docker, Memcached, MySQL, PostgreSQL, Redis, systemd, and other monitoring targets - see a full list of agent 2 specific items. **Windows**: All metrics supported by Zabbix agent, and also net.tcp.service* checks of HTTPS, LDAP. Additionally, the agent 2 provides Zabbix-native monitoring solution for: PostgreSQL, Redis.
Concurrency	Active checks for single server are executed sequentially.	Checks from different plugins or multiple checks within one plugin can be executed concurrently.
Third-party traps	no	yes
Additional features		
Persistent storage	no	yes
Persistent files for log*[] metrics	yes (only on Unix)	no
Log data upload	Can be performed during log gathering to free the buffer.	Log gathering is stopped when the buffer is full, therefore the BufferSize parameter must be at least MaxLinesPerSecond x 2.
Changes user at runtime	yes (Unix-like systems only)	no (controlled by systemd)
User-configurable ciphersuites	yes	no

See also:

- *Zabbix processes description*: Zabbix agent, Zabbix agent 2
- *Configuration parameters*: Zabbix agent UNIX / Windows, Zabbix agent 2 UNIX / Windows

Built-in plugins

These plugins are built into Zabbix agent 2 and are available out-of-the-box. Click on the plugin name to go to the plugin repository with additional information.

Plugin name	Description	Supported item keys	Comments
Agent	Metrics of the Zabbix agent being used.	agent.hostname, agent.ping, agent.version	Supported keys have the same parameters as Zabbix agent keys.
CPU	System CPU monitoring (number of CPUs/CPU cores, discovered CPUs, utilization percentage).	system.cpu.discovery, system.cpu.num, system.cpu.util	Supported keys have the same parameters as Zabbix agent keys.
Docker	Monitoring of Docker containers.	docker.container_info, docker.container_stats, docker.containers, docker.containers.discovery, docker.data_usage, docker.images, docker.images.discovery, docker.info, docker.ping	See also: Configuration parameters

Plugin name	Description	Supported item keys	Comments
File	File metrics collection.	vfs.file.cksum, vfs.file.contents, vfs.file.exists, vfs.file.md5sum, vfs.file.regexp, vfs.file.regmatch, vfs.file.size, vfs.file.time	Supported keys have the same parameters as Zabbix agent keys.
Kernel	Kernel monitoring.	kernel.maxfiles, kernel.maxproc	Supported keys have the same parameters as Zabbix agent keys.
Log	Log file monitoring.	log, log.count, logrt, logrt.count	Supported keys have the same parameters as Zabbix agent keys. See also: Plugin configuration parameters (Unix/Windows)
Memcached	Memcached server monitoring.	memcached.ping, memcached.stats	
Modbus	Reads Modbus data.	modbus.get	Supported keys have the same parameters as Zabbix agent keys.
MQTT	Receives published values of MQTT topics.	mqtt.get	To configure an encrypted connection to the MQTT broker, specify TLS parameters in the agent configuration file as a named session or default parameters. Currently TLS parameters cannot be passed as item key parameters.
MySQL	Monitoring of MySQL and its forks.	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	To configure an encrypted connection to the database, specify TLS parameters in the agent configuration file as a named session or default parameters. Currently TLS parameters cannot be passed as item key parameters.
NetIf	Monitoring of network interfaces.	net.if.collisions, net.if.discovery, net.if.in, net.if.out, net.if.total	Supported keys have the same parameters as Zabbix agent keys.

Plugin name	Description	Supported item keys	Comments
Oracle	Oracle Database monitoring.	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	Install the Oracle Instant Client before using the plugin.
Proc	Process CPU utilization percentage.	proc.cpu.util	Supported key has the same parameters as Zabbix agent key.
Redis	Redis server monitoring.	redis.config, redis.info, redis.ping, redis.slowlog.count	To configure an encrypted connection to Redis, specify TLS parameters in the agent configuration file as a named session or default parameters. TLS parameters cannot be passed as item key parameters. Note that an incorrect or otherwise invalid TLS configuration can prevent Zabbix agent 2 from starting, so verify certificate files, permissions and paths before enabling TLS.

Smart	S.M.A.R.T. monitoring.	smart.attribute.discovery, smart.disk.discovery, smart.disk.get	The minimum required smartctl version is 7.1.
			Sudo/root access rights to smartctl are required for the user executing Zabbix agent 2. The plugin uses only the following commands:
			`/usr/sbin/smartctl -a *`
			`/usr/sbin/smartctl --scan *`
			`/usr/sbin/smartctl -j -V`
			
			Supported keys can be used with Zabbix agent 2 only on Linux/Windows, both as a passive and active check.
			See also:
			Configuration parameters

SW	Listing of installed packages.	system.sw.packages, system.sw.packages.get	The supported keys have the same parameters as Zabbix agent key.
Swap	Swap space size in bytes/percentage.	system.swap.size	Supported key has the same parameters as Zabbix agent key.
SystemRun	Runs specified command.	system.run	Supported key has the same parameters as Zabbix agent key. See also: Plugin configuration parameters (Unix/Windows)

Plugin name	Description	Supported item keys	Comments
Systemd	Monitoring of systemd services.	systemd.unit.discovery, systemd.unit.get, systemd.unit.info	
TCP	TCP connection availability check.	net.tcp.port	Supported key has the same parameters as Zabbix agent key.
UDP	Monitoring of the UDP services availability and performance.	net.udp.service, net.udp.service.perf	Supported keys have the same parameters as Zabbix agent keys.
Uname	Retrieval of information about the system.	system.hostname, system.sw.arch, system.uname	Supported keys have the same parameters as Zabbix agent keys.
Uptime	System uptime metrics collection.	system.uptime	Supported key has the same parameters as Zabbix agent key.
VFSDev	VFS metrics collection.	vfs.dev.discovery, vfs.dev.read, vfs.dev.write	Supported keys have the same parameters as Zabbix agent keys.
WebCertificate	Monitoring of TLS/SSL website certificates.	web.certificate.get	
WebPage	Web page monitoring.	web.page.get, web.page.perf, web.page.regexp	Supported keys have the same parameters as Zabbix agent keys.
ZabbixAsync	Asynchronous metrics collection.	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	Supported keys have the same parameters as Zabbix agent keys.
ZabbixStats	Zabbix server/proxy internal metrics or number of delayed items in a queue.	zabbix.stats	Supported keys have the same parameters as Zabbix agent keys.
ZabbixSync	Synchronous metrics collection.	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.	Supported keys have the same parameters as Zabbix agent keys.

Loadable plugins

Loadable plugins are not available out-of-the-box for Zabbix agent 2 and need to be installed separately:

- On Linux, you can use packages (e.g., install Ember+ on Ubuntu with `apt install zabbix-agent2-plugin-ember-plus`) or build plugins
- On Windows, you can install from MSI or build plugins

Attention:
Before installing a plugin, please check its README file. It may contain specific requirements and installation instructions.

Click on the plugin name to go to the plugin repository, which contains the plugin README file with additional information.

Plugin name	Description	Supported item keys	Comments
Ceph	Ceph monitoring.	ceph.df.details, ceph.osd.stats, ceph.osd.discovery, ceph.osd.dump, ceph.ping, ceph.pool.discovery, ceph.status	See also Ceph plugin configuration parameters.
Ember+	Monitoring of Ember+.	ember.get	See also Ember+ plugin configuration parameters.
MongoDB	Monitoring of MongoDB servers and clusters (document-based, distributed database).	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	To configure encrypted connections to the database, specify TLS parameters in the agent configuration file as a named session parameters. Currently TLS parameters cannot be passed as item key parameters. See also MongoDB plugin configuration parameters.
MSSQL	Monitoring of MSSQL database.	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	To configure an encrypted connection to the database, specify TLS parameters in the agent configuration file as a named session or default parameters. Currently TLS parameters cannot be passed as item key parameters. See also MSSQL plugin configuration parameters.

Plugin name	Description	Supported item keys	Comments
NVIDIA GPU	Monitoring of NVIDIA GPU.	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	See also NVIDIA GPU plugin configuration parameters.
PostgreSQL	Monitoring of PostgreSQL and its 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	To configure encrypted connections to the database, specify TLS parameters in the agent configuration file as a named session or default parameters. Currently TLS parameters cannot be passed as item key parameters. See also PostgreSQL plugin configuration parameters.

Note:
Loadable plugins, when launched with:
 - `-V --version` - print plugin version and license information;
 - `-h --help` - print help information.
 - `-t, --test <item key>` — launch plugin for testing (plugin config ignored).

4 Proxy

Aperçu

Le proxy Zabbix est un processus qui peut collecter des données de supervision à partir d'un ou de plusieurs équipements supervisés et envoyer ces informations au serveur Zabbix, agissant essentiellement pour le compte 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 avantageux pour répartir la charge d'un seul serveur Zabbix. Si seuls les proxies collectent les données, le traitement sur le serveur devient moins gourmand en CPU et en E/S disque.

Un proxy Zabbix est la solution idéale pour la supervision centralisée de sites distants, de succursales et de réseaux sans administrateurs locaux.

Le proxy Zabbix nécessite une base de données distincte.

> **Attention:**
> Notez que les bases de données prises en charge avec le proxy Zabbix sont SQLite, MySQL et PostgreSQL.

Voir aussi : Utilisation des proxies dans un environnement distribué

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
```

Démarrage manuel

Si ce qui précède ne fonctionne pas, vous devez le démarrer manuellement. Recherchez le chemin vers le binaire zabbix_proxy et exécutez :

```
zabbix_proxy
```

Vous pouvez utiliser les paramètres de ligne de commande suivants avec Zabbix proxy :

```
-c --config <file>              chemin vers le fichier de configuration
-f --foreground                 exécuter Zabbix proxy au premier plan
-R --runtime-control <option>   effectuer des fonctions administratives
-T --test-config                valider le fichier de configuration et quitter
-h --help                       afficher cette aide
-V --version                    afficher le numéro de version
```

Exemples d'exécution de Zabbix proxy avec des paramètres de ligne de commande :

```
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf
zabbix_proxy --help
zabbix_proxy -V
```

Contrôle d'exécution

Options de contrôle d'exécution :

Option	Description	Cible
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 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, sauf la première et la dernière.	**target** - ID de l'élément
diaginfo[=<**section**>]	Collecter 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	Recharger le cache SNMP — effacer les propriétés du moteur SNMP (temps du moteur, démarrages du moteur, ID du moteur, informations d'identification) 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 nettoyage. Ignoré si la procédure de nettoyage est actuellement en cours.	
log_level_increase[=<**target**>]	Augmenter le niveau de journalisation, affecte tous les processus si target n'est pas spécifié. Non pris en charge sur les systèmes *BSD*.	**type de processus** - Tous les processus du type spécifié (par exemple, poller) Voir tous les types de processus proxy. **type de processus,N** - Type et numéro du processus (par exemple, poller,3) **pid** - Identifiant du processus (1 à 65535). Pour des valeurs plus grandes, spécifiez target comme 'type de processus,N'.
log_level_decrease[=<**target**>]	Diminuer le niveau de journalisation, affecte tous les processus si target n'est pas spécifié. Non pris en charge sur les systèmes *BSD*.	
prof_enable[=<**target**>]	Activer le profilage. Affecte tous les processus si target n'est pas spécifié. Le profilage activé fournit les détails de tous les rwlocks/mutexes par nom de fonction.	**type de processus** - Tous les processus du type spécifié (par exemple, history syncer) Voir tous les types de processus proxy. **type de processus,N** - Type et numéro du processus (par exemple, history syncer,1) **pid** - Identifiant du processus (1 à 65535). Pour des valeurs plus grandes, spécifiez target comme 'type de processus,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 d'un type donné (par exemple, history syncer,rwlock)
prof_disable[=<**target**>]	Désactiver le profilage. Affecte tous les processus si target n'est pas spécifié.	**type de processus** - Tous les processus du type spécifié (par exemple, history syncer) Voir tous les types de processus proxy. **type de processus,N** - Type et numéro du processus (par exemple, history syncer,1) **pid** - Identifiant du processus (1 à 65535). Pour des valeurs plus grandes, spécifiez target comme 'type de processus,N'.

Exemple d'utilisation du contrôle d'exécution 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 d'exécution 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 d'exécution 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 d'exécution pour recharger le cache SNMP :

```
zabbix_proxy -R snmp_cache_reload
```

Attention:
Lorsqu'une interface SNMPv3 est mise à jour via l'interface utilisateur Zabbix, Zabbix recharge automatiquement les nouvelles informations d'identification SNMPv3 pour cette interface dans la plupart des cas ; utilisez `-R snmp_cache_reload` uniquement si l'interrogation échoue toujours après les modifications des informations d'identification (par exemple, en raison d'incohérences engineBoots/engineID ou d'appareils non conformes à la RFC), ou lorsque vous devez forcer un effacement global du cache SNMP à des fins de dépannage.

Exemple d'utilisation du contrôle d'exécution pour déclencher l'exécution du nettoyage :

```
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R housekeeper_execute
```

Exemples d'utilisation du contrôle d'exécution 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"
```

Utilisateur du processus

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 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 proxy en tant que 'root' que si vous modifiez en conséquence le paramètre 'AllowRoot' dans le fichier de configuration du proxy.

Fichier de configuration

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 collecte asynchrone pour les vérifications passives avec un thread de travail
- `availability manager` - processus de mise à jour de la disponibilité des hôtes
- `browser poller` - collecteur pour les vérifications des éléments de navigateur
- `configuration syncer` - processus de gestion du cache en mémoire des données de configuration
- `data sender` - expéditeur de données du proxy
- `discovery manager` - processus gestionnaire pour la découverte des périphériques
- `discovery worker` - processus de traitement des tâches de découverte provenant du gestionnaire de découverte
- `history syncer` - processus d'écriture de l'historique dans la base de données
- `housekeeper` - processus de suppression des anciennes données historiques
- `http agent poller` - processus de collecte asynchrone pour les vérifications HTTP avec un thread de travail
- `http poller` - collecteur de supervision web

- `icmp pinger` - collecteur pour les vérifications icmpping
- `internal poller` - collecteur pour les vérifications internes
- `ipmi manager` - gestionnaire des collecteurs IPMI
- `ipmi poller` - collecteur pour les vérifications IPMI
- `java poller` - collecteur pour les vérifications Java
- `odbc poller` - collecteur pour les vérifications ODBC
- `poller` - collecteur 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
- `self-monitoring` - processus de collecte des statistiques internes du serveur
- `snmp poller` - processus de collecte asynchrone pour les vérifications SNMP avec un thread de travail (éléments `walk[OID]` et `get[OID]` uniquement)
- `snmp trapper` - trapper pour les traps SNMP
- `task manager` - processus d'exécution à distance des tâches demandées par d'autres composants (par ex. fermer un problème, acquitter un problème, vérifier la valeur d'un élément maintenant, fonctionnalité de commande à distance)
- `trapper` - trapper pour les vérifications actives, les traps, la communication du proxy
- `unreachable poller` - collecteur pour les périphériques injoignables
- `vmware collector` - collecteur de données VMware chargé de la collecte des données depuis les 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 permissions de lecture-écriture pour le propriétaire du fichier uniquement. De plus, le fichier est lisible par le groupe propriétaire. Toutes les autres permissions sont refusées.

Différents types de processus du proxy Zabbix peuvent être supervisés à l'aide de l'élément interne **zabbix[process,<type>,<mode>,<stat**
item.

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 les suivantes :

- Temps consacré à l'écriture des valeurs d'élément dans la base de données ;
- Temps consacré à la mise à jour des données de l'élément (état, erreurs).

Housekeeping procedure

Zabbix proxy has the `housekeeper` process that removes outdated item history and trends. It runs in cycles, with frequency determined by `HousekeepingFrequency`, and the per-cycle deletion limit determined by `ProxyLocalBuffer` and `ProxyOfflineBuffer`. Unlike in the Zabbix server housekeeping procedure, the proxy `housekeeper` process does not use the `housekeeper` table—it deletes all outdated data once per housekeeping cycle.

Plateformes prises en charge

Le proxy Zabbix fonctionne sur la même liste de plateformes prises en charge que le serveur Zabbix.

Tampon mémoire

Le tampon mémoire permet de stocker les nouvelles données (valeurs d'éléments, découverte réseau, auto-enregistrement 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 depuis 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, cela reste le comportement par défaut après la mise à niveau vers Zabbix 7.0.

Pour des performances optimisées, 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 dans la base de données les données non envoyées 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 sera utilisé, cependant, il n'existe aucune protection contre la perte de données. Si le proxy est arrêté ou si la mémoire est saturée, les données non envoyées seront perdues.

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 conflictuelle, le proxy affichera une erreur et ne démarrera pas, par exemple si :

- ProxyBufferMode est défini sur « hybrid » ou « memory » et ProxyMemoryBufferSize vaut « 0 » ;
- ProxyBufferMode est défini sur « hybrid » ou « memory » et ProxyLocalBuffer n'est pas « 0 ».

Paramètres régionaux

Notez que le proxy nécessite des paramètres régionaux UTF-8 afin que certains éléments textuels puissent être interprétés correctement. La plupart des systèmes modernes de type Unix utilisent des paramètres régionaux UTF-8 par défaut ; toutefois, sur certains systèmes, il peut être nécessaire de les définir explicitement.

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.

5 Passerelle Java

Aperçu

Le Java gateway Zabbix peut être installé à partir du code source ou de paquets.

La prise en charge native de la supervision des applications JMX existe sous la forme d'un démon Zabbix appelé « Zabbix Java gateway ». Zabbix Java gateway est un démon 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 n'a pas besoin d'un logiciel supplémentaire installé ; 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 ne peut être utilisé que comme un « proxy passif ». Contrairement à Zabbix proxy, il peut également être utilisé depuis Zabbix proxy (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 ; ainsi, 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'un autre type, seuls les éléments **JMX agent** seront transmis à Java gateway pour récupération.

Lorsqu'un élément doit être mis à jour via le Java gateway, le serveur Zabbix ou le proxy se connecte au Java gateway et demande la valeur, que le Java gateway récupère ensuite et renvoie au serveur ou au proxy. Ainsi, le Java gateway ne met en cache aucune valeur.

Le serveur Zabbix ou le proxy dispose d'un type spécifique de processus qui se connectent au Java gateway, contrôlé par l'option **StartJavaPollers**. En interne, le Java gateway démarre plusieurs threads, contrôlés par l'option **START_POLLERS**. Côté serveur, si une connexion prend plus de **Timeout** secondes, elle sera interrompue, mais le Java gateway peut encore être occupé à récupérer la valeur du compteur JMX. Pour résoudre ce problème, il existe l'option **TIMEOUT** dans le Java gateway, qui permet de définir un délai d'expiration pour les opérations réseau JMX.

Le serveur ou le proxy Zabbix essaiera de regrouper autant que possible les requêtes vers une même cible JMX (ceci étant influencé par les intervalles des éléments) et de les envoyer au Java gateway via une seule connexion afin d'améliorer les 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 superviser des applications Java basées sur Wildfly avec le Java gateway de Zabbix, veuillez installer le dernier fichier jboss-client.jar disponible sur la page de téléchargement Wildfly.

2 Installation 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 exécution 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 du système :

RHEL 7 et versions ultérieures :

```
systemctl enable zabbix-java-gateway
```

Versions de RHEL antérieures à 7 :

```
chkconfig --level 12345 zabbix-java-gateway on
```

Configuration du serveur pour utiliser la passerelle Java

Avec la passerelle Java opérationnelle, vous devez indiquer au serveur Zabbix où trouver la passerelle Java Zabbix. Pour ce faire, indiquez les paramètres JavaGateway et JavaGatewayPort dans le fichier de configuration du serveur. Si l'hôte sur lequel l'application JMX est en cours d'exécution est surveillé par le proxy Zabbix, spécifiez les paramètres de connexion dans le fichier de configuration du proxy à la place de celui du serveur.

```
JavaGateway=192.168.3.14
JavaGatewayPort=10052
```

Par défaut, le serveur ne démarre aucun processus lié à la surveillance JMX. Toutefois, si vous souhaitez l'utiliser, vous devez spécifier le nombre d'instances prédéfinies de pollers Java. Vous faites cela de la même manière que vous spécifiez les pollers et les trappers.

```
StartJavaPollers=5
```

N'oubliez pas de redémarrer le serveur ou le proxy, une fois que vous avez fini de les configurer.

Déboggage de la passerelle Java

Le fichier journal de la passerelle Java Zabbix est :

```
/var/log/zabbix/zabbix_java_gateway.log
```

Si vous souhaitez augmenter le niveau de journalisation, éditez le fichier :

```
/etc/zabbix/zabbix_java_gateway_logback.xml
```

et modifiez `level="info"` en "debug" ou encore "trace" (pour un dépannage en profondeur) :

```
<configuration scan="true" scanPeriod="15 seconds">
[...]
      <root level="info">
              <appender-ref ref="FILE" />
      </root>

</configuration>
```

Supervision JMX

Voir la page Supervision JMX pour plus de détails.

3 Configuration depuis les packages Debian/Ubuntu

Aperçu

Si installé à partir de paquets, les informations suivantes vous aideront à configurer le Java gateway.

Configuration et exécution 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 du système :

```
systemctl enable zabbix-java-gateway
```

Configuration du serveur pour utiliser la passerelle Java

Avec la passerelle Java opérationnelle, vous devez indiquer au serveur Zabbix où trouver la passerelle Java Zabbix. Pour ce faire, indiquez les paramètres JavaGateway et JavaGatewayPort dans le fichier de configuration du serveur. Si l'hôte sur lequel l'application JMX est en cours d'exécution est surveillé par le proxy Zabbix, spécifiez les paramètres de connexion dans le fichier de configuration du proxy à la place de celui du serveur.

```
JavaGateway=192.168.3.14
JavaGatewayPort=10052
```

Par défaut, le serveur ne démarre aucun processus lié à la surveillance JMX. Toutefois, si vous souhaitez l'utiliser, vous devez spécifier le nombre d'instances prédéfinies de pollers Java. Vous faites cela de la même manière que vous spécifiez les pollers et les trappers.

```
StartJavaPollers=5
```

N'oubliez pas de redémarrer le serveur ou le proxy, une fois que vous avez fini de les configurer.

Déboggage de la passerelle Java

Le fichier journal de la passerelle Java Zabbix est :

```
/var/log/zabbix/zabbix_java_gateway.log
```

Si vous souhaitez augmenter le niveau de journalisation, éditez le fichier :

```
/etc/zabbix/zabbix_java_gateway_logback.xml
```

et modifiez `level="info"` en "debug" ou encore "trace" (pour un dépannage en profondeur) :

```
<configuration scan="true" scanPeriod="15 seconds">
[...]
      <root level="info">
              <appender-ref ref="FILE" />
      </root>

</configuration>
```

Supervision JMX

Voir la page Supervision JMX pour plus de détails.

Installation à partir des sources

Aperçu

Si installé à partir des sources, les informations suivantes vous aideront à configurer le Java gateway de Zabbix.

Aperçu 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
```

Le fichier JAR de Java gateway lui-même.

```
lib/logback-core-1.5.16.jar
lib/logback-classic-1.5.16.jar
lib/slf4j-api-2.0.16.jar
lib/android-json-4.3_r3.1.jar
```

Dépendances de Java gateway : Logback, SLF4J et la bibliothèque Android JSON.

```
lib/logback.xml
lib/logback-console.xml
```

Fichiers de configuration pour Logback.

`shutdown.sh`
`startup.sh`

Scripts utilitaires pour démarrer et arrêter Java gateway.

`settings.sh`

Fichier de configuration utilisé 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 port différent, vous pouvez le spécifier dans le script settings.sh. Consultez la description du fichier de configuration de Java gateway pour savoir comment spécifier cette option et d'autres options.

> **Warning:**
> Le port 10052 n'est pas enregistré auprès de l'IANA.

Une fois les paramètres définis comme souhaité, vous pouvez démarrer Java gateway en exécutant le script de démarrage :

```
./startup.sh
```

De même, lorsque vous n'avez plus besoin de Java gateway, exécutez le script d'arrêt 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 la passerelle Java

Avec la passerelle Java opérationnelle, vous devez indiquer au serveur Zabbix où trouver la passerelle Java Zabbix. Pour ce faire, indiquez les paramètres JavaGateway et JavaGatewayPort dans le fichier de configuration du serveur. Si l'hôte sur lequel l'application JMX est en cours d'exécution est surveillé par le proxy Zabbix, spécifiez les paramètres de connexion dans le fichier de configuration du proxy à la place de celui du serveur.

```
JavaGateway=192.168.3.14
JavaGatewayPort=10052
```

Par défaut, le serveur ne démarre aucun processus lié à la surveillanceJMX. Toutefois, si vous souhaitez l'utiliser, vous devez spécifier le nombre d'instances prédéfinies de pollers Java. Vous faites cela de la même manière que vous spécifiez les pollers et les trappers.

```
StartJavaPollers=5
```

N'oubliez pas de redémarrer le serveur ou le proxy, une fois que vous avez fini de les configurer.

Débogage de la passerelle Java

Au cas où il y aurait un problème avec la passerelle Java ou si un message d'erreur que vous voyez à propos d'un élément dans l'interface n'est pas assez descriptif, vous voudrez peut-être jeter un coup d'œil au fichier journal de la passerelle Java.

Par défaut, la passerelle Java consigne 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 de disposer d'informations de niveau "debug". Pour augmenter le niveau de journalisation, modifiez le fichier lib/logback.xml et modifiez l'attribut de niveau du tag <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 la passerelle Java Zabbix après la modification du fichier logback.xml. Les modifications apportées dans logback.xml sont automatiquement prises en compte. Lorsque vous avez terminé le débogage, vous pouvez rétablir le niveau de journalisation sur "info".

Si vous souhaitez vous connecter à un autre fichier ou à un support complètement différent, tel que base de données, ajustez le fichier logback.xml en fonction de vos besoins. Voir le manuel de Logback pour plus de détails.

Parfois, à des fins de débogage, il est utile de démarrer la passerelle Java en tant qu'application console plutôt qu'un démon. Pour cela, commentez la variable PID_FILE dans settings.sh. Si PID_FILE est omis, le script startup.sh démarre la passerelle Java en tant qu'application console et oblige Logback à utiliser à la place le fichier lib/logback-console.xml, qui non seulement renvoie ses logs à la console, mais dispose également du niveau de journalisation "debug".

Enfin, notez que, puisque la passerelle Java utilise SLF4J pour la journalisation, vous pouvez remplacer Logback par la structure de votre choix en plaçant un fichier JAR approprié dans le répertoire lib. Voir le manuel de SLF4J pour plus de détails.

Supervision JMX

Voir la page Supervision JMX pour plus de détails.

6 Sender

Aperçu

Zabbix sender est un utilitaire en ligne de commande qui peut être utilisé pour envoyer des données de performance au serveur Zabbix afin qu'elles soient traitées.

Cet utilitaire est généralement utilisé dans des scripts utilisateur de longue durée d'exécution pour l'envoi périodique de données de disponibilité et de performance.

Pour envoyer des résultats directement au serveur Zabbix ou au proxy, un élément de type trapper item doit être configuré.

Voir également la bibliothèque Python pour Zabbix, qui intègre une fonctionnalité permettant d'agir comme Zabbix sender.

Exécution de Zabbix sender

Voici un exemple d'exécution de Zabbix sender sous UNIX :

```
cd bin
./zabbix_sender -z zabbix -s "Linux DB3" -k db.connections -o 43
```

où :

- z - hôte du serveur Zabbix (une adresse IP peut également être utilisée)
- s - nom technique de l'hôte supervisé (tel qu'enregistré dans le frontend Zabbix)
- k - clé de l'élément
- o - valeur à envoyer

Attention:
Les options qui contiennent 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, l'émetteur essaie d'envoyer les données aux autres adresses. Si l'envoi de données par lots échoue vers une adresse, les lots suivants ne sont pas envoyés à cette adresse.

Zabbix sender accepte les chaînes en encodage UTF-8 (à la fois pour les systèmes de type UNIX et Windows) sans marque d'ordre des octets (BOM) au début du fichier.

Zabbix sender sous Windows peut être exécuté de manière similaire :

```
zabbix_sender.exe [options]
```

Les scénarios d'envoi en temps réel de zabbix_sender rassemblent plusieurs valeurs qui lui sont transmises à de courts intervalles et les envoient au serveur via une seule connexion. Une valeur qui n'est pas espacée de plus de 0,2 seconde par rapport à la valeur précédente peut être placée dans la même pile, mais le temps d'interrogation maximal reste de 1 seconde.

Note:
Zabbix sender s'arrêtera si une entrée de paramètre invalide (ne respectant pas la notation *paramètre=valeur*) est présente dans le fichier de configuration spécifié.

Exécution de Zabbix sender avec la découverte de bas niveau

Exemple d'exécution de Zabbix sender pour envoyer une valeur au format JSON pour la découverte de bas niveau :

```
./zabbix_sender -z 192.168.1.113 -s "Zabbix server" -k trapper.discovery.item -o '[{"{#FSNAME}":"/","{#FST
```

Pour que cela fonctionne, la règle de découverte de bas niveau doit avoir un type d'élément Zabbix trapper (dans cet exemple, avec la clé `trapper.discovery.item`).

7 Get

Aperçu

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.

Cet utilitaire est généralement utilisé pour le dépannage des agents Zabbix.

Voir également la bibliothèque Python pour Zabbix, qui intègre une fonctionnalité permettant d'agir comme Zabbix get.

Exécution de Zabbix get

Voici un exemple d'exécution de Zabbix get sous UNIX pour obtenir la valeur de la charge du processeur depuis l'agent :

```
cd bin
./zabbix_get -s 127.0.0.1 -p 10050 -k system.cpu.load[all,avg1]
```

Voici un 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 qu'ici, la clé d'élément contient un espace, des guillemets sont donc utilisés pour indiquer la clé d'élément à l'interpréteur de commandes. Les guillemets ne font pas partie de la clé d'élément ; ils seront supprimés par l'interpréteur de commandes 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>             Spécifier le nom d'hôte ou l'adresse IP d'un hôte
-p --port <port number>                 Spécifier le numéro de port de l'agent exécuté sur l'hôte (par défa
-I --source-address <IP address>        Spécifier l'adresse IP source
-t --timeout <seconds>                  Spécifier le délai d'attente. Plage valide : 1-600 secondes (par dé
-k --key <item key>                     Spécifier la clé de l'élément pour lequel récupérer la valeur
-P --protocol <value>                   Protocole utilisé pour communiquer avec l'agent. Valeurs :
                                            auto - se connecter en utilisant le protocole JSON, puis reven
                                            json - se connecter en utilisant le protocole JSON
                                            plaintext - se connecter en utilisant le protocole en texte br
-h --help                               Afficher ce message d'aide
-V --version                            Afficher le numéro de version

--tls-connect <value>                   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 <CA file>                 Chemin complet vers un fichier contenant les certificats de la ou
--tls-crl-file <CRL file>               Chemin complet vers 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 vers un fichier contenant le certificat ou la chaîn
--tls-key-file <key file>               Chemin complet vers un fichier contenant la clé privée
--tls-psk-identity <PSK-identity>       Chaîne unique sensible à la casse utilisée pour identifier la clé
--tls-psk-file <PSK-file>               Chemin complet vers un fichier contenant la clé pré-partagée
--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 supérieur) ou chaîne de
```

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]
```

8 JS

Vue d'ensemble

zabbix_js est un utilitaire en ligne de commande qui peut être utilisé pour tester des scripts embarqués.

Cet utilitaire exécute un script utilisateur avec un paramètre de type chaîne et affiche le résultat. Les scripts sont exécutés à l'aide du moteur de script embarqué de Zabbix.

En cas d'erreurs de compilation ou d'exécution, zabbix_js affiche l'erreur sur stderr et se termine avec le code 1.

Utilisation

```
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
```

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 com
-i, --input input-file            Spécifie le nom du fichier du paramètre d'entrée. Si '-' est spécifié co
-p, --param input-param           Spécifie le paramètre d'entrée.
-l, --loglevel log-level          Spécifie le niveau de journalisation.
-t, --timeout timeout             Spécifie le délai d'expiration en secondes. Plage valide : 1-600 seconde
-h, --help                        Affiche les informations d'aide.
-V, --version                     Affiche le numéro de version.
-w <webdriver url>                Active la surveillance du navigateur.
```

Exemple :

```
zabbix_js -s script-file.js -p example
```

9 Web service

Aperçu

Le service web Zabbix est un processus utilisé pour communiquer avec des services web externes. Actuellement, le service web Zabbix est utilisé pour générer et envoyer des rapports planifiés, avec l'ajout de fonctionnalités supplémentaires prévu 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 les 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 `zabbix_web_service.conf`.

> **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.

5 Configuration

Veuillez utiliser la barre latérale pour accéder au contenu de la section Configuration.

1 Configuration d'un modèle

Aperçu

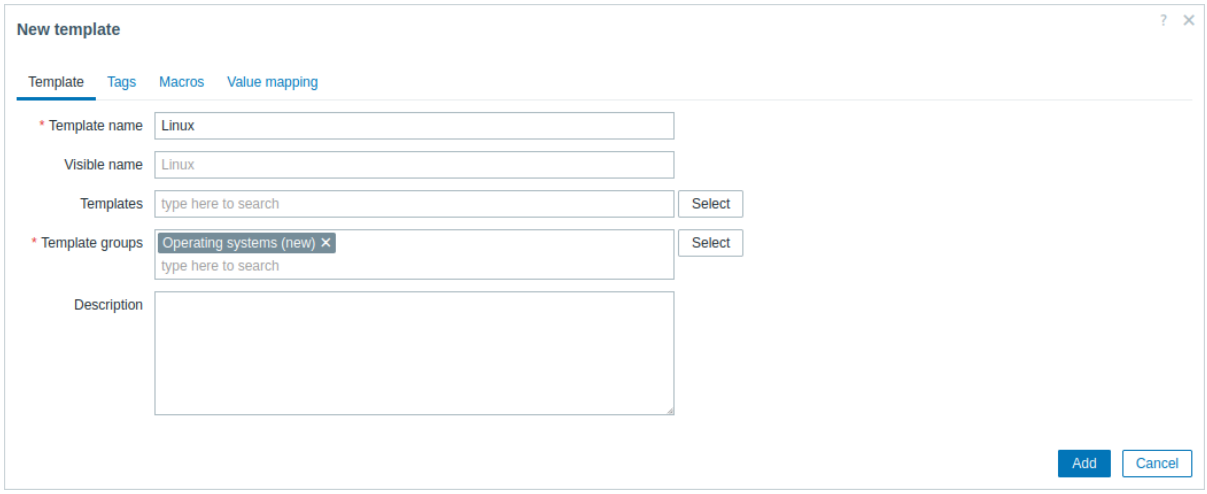
La configuration d'un modèle nécessite d'abord de créer un modèle en définissant ses paramètres généraux, puis d'y ajouter des entités (éléments, déclencheurs, graphiques, etc.).

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.

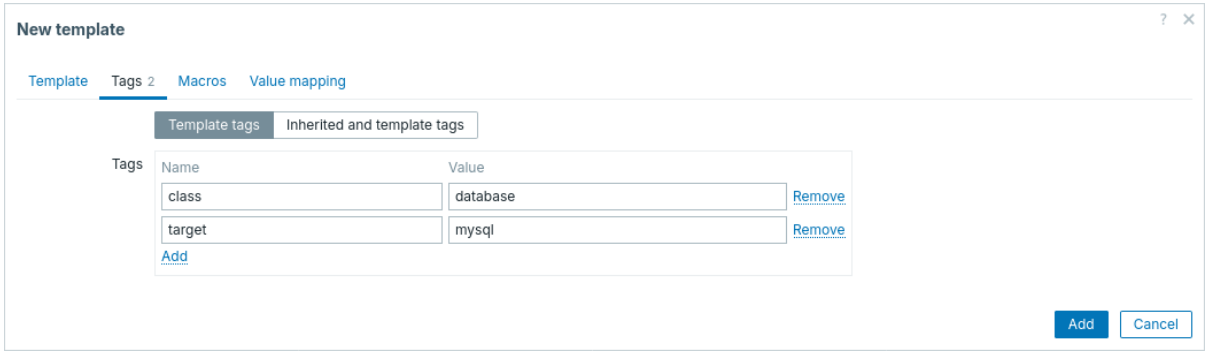


Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge.

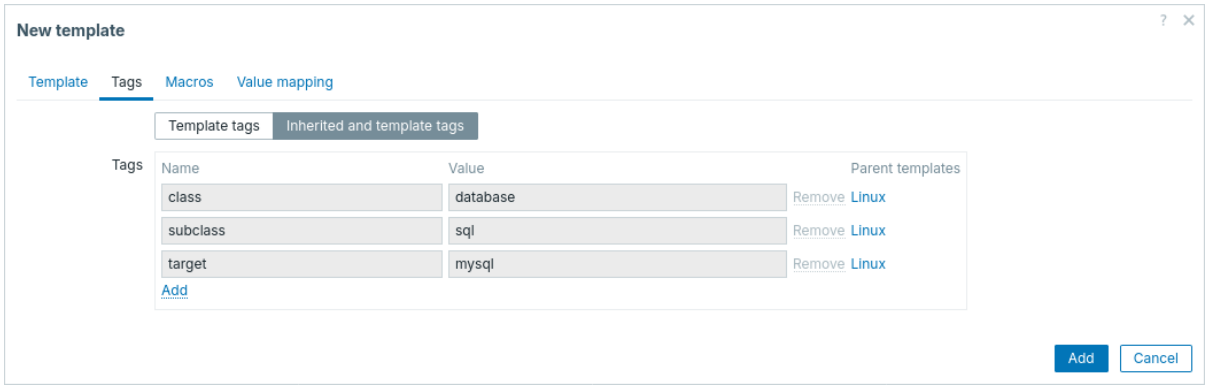
Attributs du modèle :

Parameter	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 soit un nom de groupe de modèles dans le champ *Groupe de modèles* (puis sélectionnez-en un dans la liste proposée), soit appuyez sur le bouton *Sélectionner* 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 pour afficher les modèles appartenant à ce groupe. Les modèles sélectionnés seront liés lorsque la configuration sera enregistrée ou mise à jour. Pour délier un modèle, utilisez l'une des deux options du bloc *Modèles* : *Délier* - délier le modèle, mais conserver ses entités (éléments, déclencheurs, etc.) ; *Délier et supprimer* - délier le modèle et supprimer toutes ses entités (éléments, déclencheurs, etc.).
Groupes de modèles	Groupes de modèles auxquels le modèle appartient.
Description	Description du modèle.
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 web de Zabbix. 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, 7.0-5, 8.0-0, 8.0-3.

L'onglet **Tags** vous permet de définir des tags au niveau du modèle. Tous les problèmes des hôtes liés à ce modèle seront étiquetés avec les valeurs saisies ici.



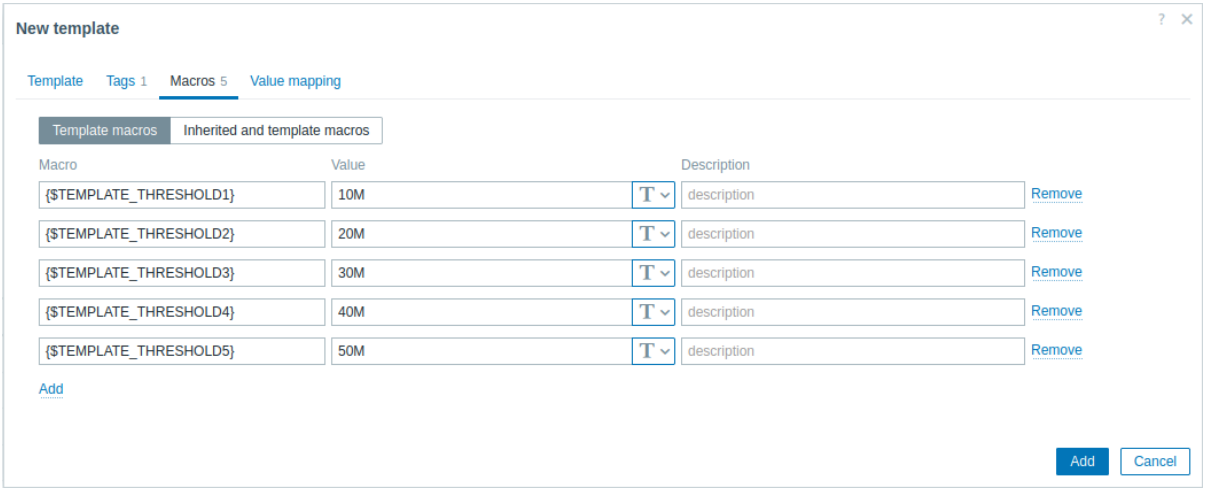
Si vous sélectionnez l'option *Tags hérités et du modèle*, vous pouvez également afficher les tags des modèles liés que le modèle héritera.



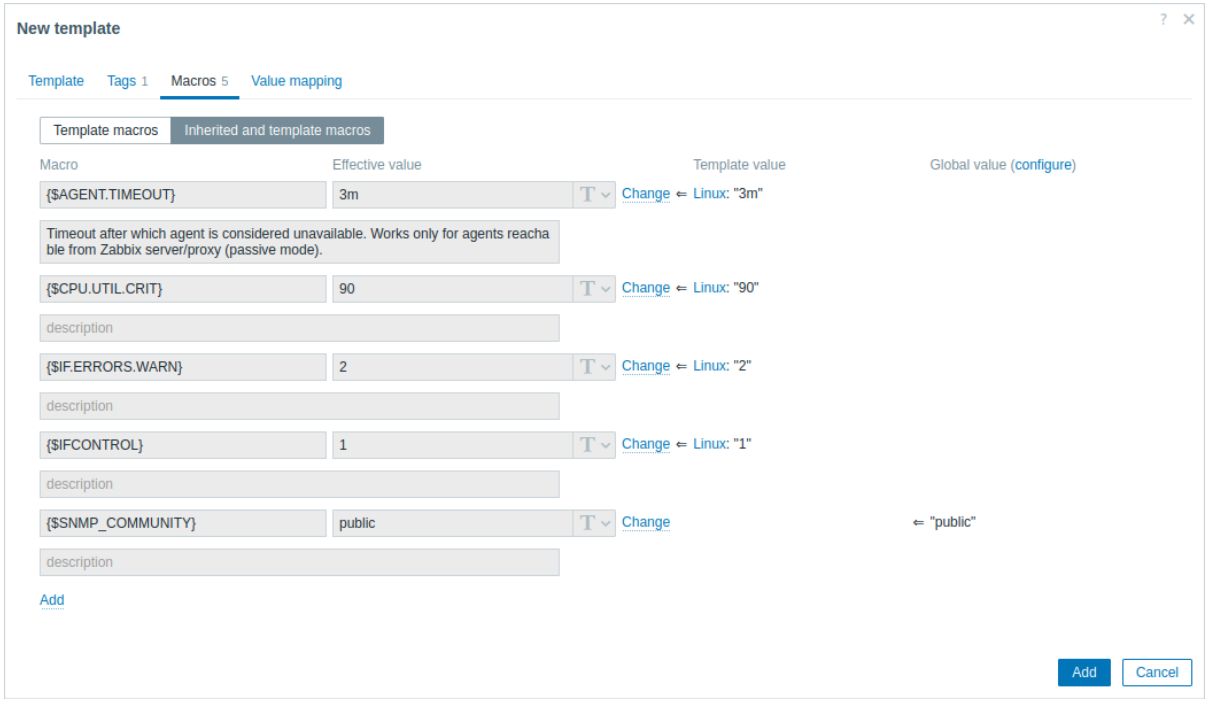
Pour plus de commodité, des liens vers les modèles respectifs sont fournis.

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 tags.

L'onglet **Macros** vous 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. L'ajout d'une description est également pris en charge.



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 respectifs, ainsi qu'un lien vers la configuration des macros globales, sont fournis. Il est également possible de modifier une macro de modèle lié ou une macro globale au niveau du modèle, ce qui crée effectivement une copie de la macro sur le modèle.

L'onglet **Correspondance des valeurs** permet de configurer une représentation conviviale des données d'élément dans les correspondances de valeurs.

Boutons :

Add	Ajouter le modèle. Le modèle ajouté doit apparaître dans la liste.
Update	Mettre à jour les propriétés d'un modèle existant.
Clone	Créer un autre modèle basé sur les propriétés du modèle actuel. Cela **inclut** les entités (éléments, déclencheurs, etc.) à la fois héritées des modèles liés et directement rattachées 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	Supprimer le modèle ; les entités du modèle (éléments, déclencheurs, etc.) restent sur les hôtes liés.
Delete and clear	Supprimer le modèle et toutes ses entités des hôtes liés.
Cancel	Annuler la modification des propriétés du modèle.

Ajout d'éléments, de déclencheurs et 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.

Il existe deux façons d'ajouter des éléments au modèle :

1. Pour créer de nouveaux éléments, suivez les instructions de Création d'un élément.

2. Pour ajouter des éléments existants au modèle :

- Accédez à *Collecte de données* > *Hôtes* (ou *Modèles*).

- Cliquez sur *Éléments* dans la ligne de l'hôte/modèle requis.
- Cochez les cases des éléments que vous souhaitez ajouter au modèle.
- Cliquez sur *Copier* sous la liste des éléments.
- 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 s'effectue 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 instructions de configuration des tableaux de bord.

Attention:
Lors de la configuration de 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 portent 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 *Groupes d'hôtes*, *Exclure les groupes d'hôtes* et *Hôtes* dans le widget *Problèmes* ne sont pas disponibles, le paramètre *Groupes d'hôtes* dans le widget *Disponibilité de l'hôte* n'est pas disponible, et le paramètre *Afficher les hôtes en maintenance* est renommé en *Afficher les données en maintenance*, etc. Pour plus d'informations sur la disponibilité des paramètres dans les widgets des tableaux 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* sur la ligne du modèle.
2. Configurez un scénario web en suivant la méthode habituelle de configuration des scénarios web.

2 Configuration d'un groupe de modèles

Overview

Template groups are used for the logical grouping of templates and assigning user permissions to them.

Each template must have at least one template group assigned. A template may belong to multiple template groups, and each template group may contain multiple templates.

Note that in Zabbix, all permissions are based on groups: user groups, host groups, and template groups. So, even if a single user needs access to a single template, it is granted by adding the user to a user group that has permission to access the template group containing that template.

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 web de Zabbix.

Option un :

- Allez à : *Collecte de données* → *Groupes de modèles*
- Cliquez sur *Créer un groupe de modèles* dans le coin supérieur droit de l'écran
- Saisissez le nom du groupe dans le formulaire



Option deux : lors de la configuration d'un modèle, saisissez un nom de groupe inexistant dans le champ *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 du groupe. Il n'est pas possible de supprimer un groupe de modèles s'il s'agit du 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`.

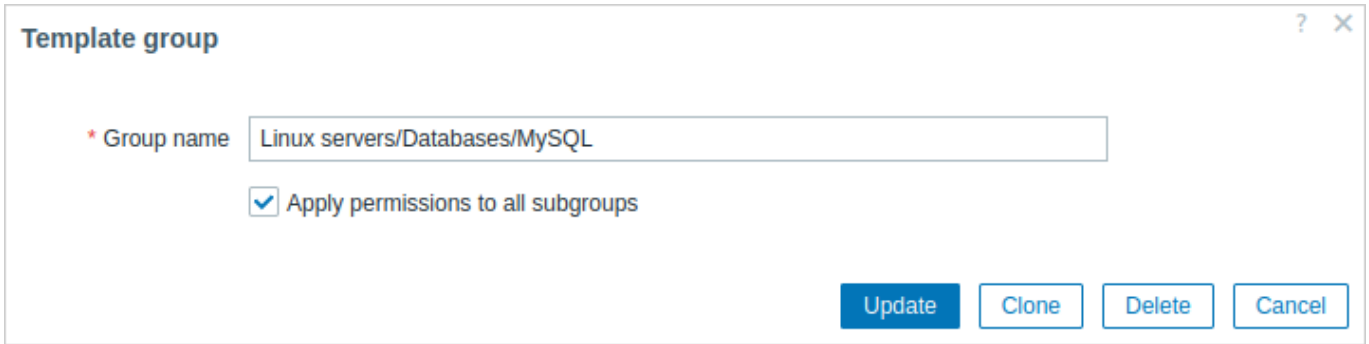
Lors de la création d'un sous-groupe, l'utilisation de barres obliques au début ou à la fin, ou de plusieurs barres obliques consécutives, n'est pas autorisée. L'échappement de « / » n'est pas pris en charge.

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.

Autorisations pour les groupes de modèles

- Lors de la création d'un sous-groupe dans un groupe de modèles parent existant (par exemple, création de `Linux servers/Databases` lorsque `Linux servers` existe déjà), les autorisations du groupe d'utilisateurs sur le sous-groupe sont héritées du parent.
- Lors de la création d'un groupe de modèles parent pour un sous-groupe existant (par exemple, création de `Linux servers` lorsque `Linux servers/Databases` existe déjà), aucune autorisation n'est définie sur le parent.

Lors de la modification de n'importe quel groupe de modèles, vous pouvez également définir une option supplémentaire, *Appliquer les autorisations à tous les sous-groupes*.



En cochant cette case puis en cliquant sur *Mettre à jour*, le même niveau d'autorisations sera appliqué à tous les sous-groupes actuels et futurs du groupe de modèles en cours de modification.

Ainsi, si des groupes d'utilisateurs ont reçu des autorisations différentes pour les sous-groupes du groupe de modèles en cours de modification, le fait de cocher cette case accordera à tous les sous-groupes actuels et futurs les mêmes autorisations 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 autorisations existantes. Toute modification effectuée via cette option ne peut être annulée que manuellement.

3 Liaison/dissociation

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.

Liaison d'un modèle

Pour lier un modèle à l'hôte, procédez comme suit :

1. Allez dans *Data collection* → *Hosts*.
2. Cliquez sur l'hôte requis.
3. Commencez à saisir le nom du modèle dans le champ *Templates*. Une liste des modèles correspondants s'affichera ; faites défiler pour sélectionner. Vous pouvez également cliquer sur *Select* à côté du champ et sélectionner un ou plusieurs modèles dans la liste d'une fenêtre contextuelle.
4. Cliquez sur *Add/Update* dans le formulaire des attributs de l'hôte.

L'hôte disposera alors de toutes les entités du modèle. Cela inclut les éléments de données, 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 de données avec la même clé d'élément. De plus, comme les déclencheurs et les graphiques utilisent des éléments de données, ils ne peuvent pas non plus être liés à un seul hôte à partir de plusieurs modèles s'ils utilisent des clés d'élément identiques.

Lorsque des entités (éléments de données, déclencheurs, etc.) sont ajoutées à partir du modèle :

- les entités identiques déjà existantes 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 ;
- toutes les entités liées directement qui, avant la liaison du modèle, existaient uniquement sur l'hôte restent inchangées.

Dans les listes, toutes les entités du modèle sont désormais préfixées par le nom du modèle, indiquant qu'elles appartiennent à ce modèle particulier. Le nom du modèle lui-même (en texte gris) est un lien permettant d'accéder à la liste de ces entités au niveau du modèle.

Note:
Pour certains éléments de données, tels que les contrôles externes, les contrôles d'agent HTTP, les contrôles simples, les contrôles SSH et les contrôles Telnet, une interface d'hôte est facultative. Si, au moment de la liaison d'un modèle, l'hôte n'a pas d'interface définie, ces éléments de données seront ajoutés sans interface. Si vous ajoutez une interface d'hôte ultérieurement, elle ne sera pas attribuée automatiquement aux éléments de données déjà existants. Pour attribuer en une seule fois la nouvelle interface d'hôte à tous les éléments de données du modèle, déliez le modèle de l'hôte, puis liez-le à nouveau. Pour conserver l'historique des éléments de données, utilisez l'option *Unlink* et non *Unlink and clear*.

Si une entité n'est pas préfixée par le nom du modèle, cela signifie qu'elle existait sur l'hôte auparavant et qu'elle n'a pas été ajoutée par le modèle.

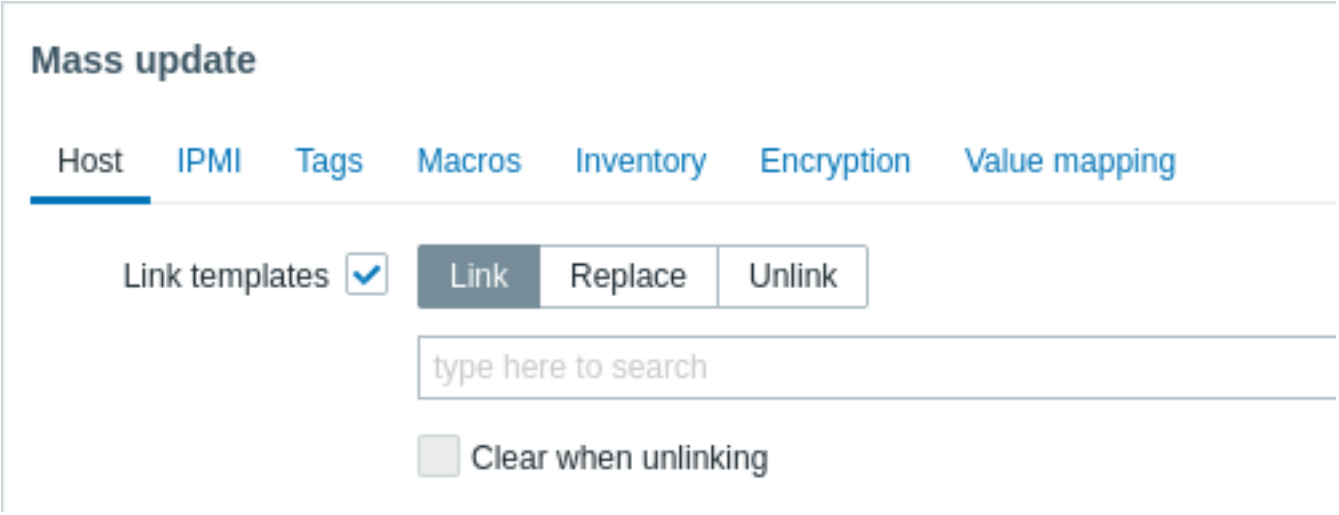
Critères d'unicité des entités

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.

Lier des modèles à plusieurs hôtes

Pour mettre à jour l'association de modèles pour plusieurs hôtes, dans *Collecte de données* → *Hôtes*, sélectionnez certains hôtes en cochant leurs cases, puis cliquez sur **Mise à jour de masse** sous la liste, puis sélectionnez *Lier des modèles* :



Pour lier des modèles supplémentaires, commencez à saisir le nom du modèle dans le champ d'auto-complétion jusqu'à ce qu'une liste déroulante apparaisse avec les modèles correspondants. Faites simplement défiler la liste pour sélectionner le modèle à lier.

L'option *Remplacer* permet de lier un nouveau modèle tout en dissociant tout modèle précédemment lié aux hôtes. L'option *Dissocier* permet de spécifier quels modèles dissocier. L'option *Effacer lors de la dissociation* permet non seulement de dissocier tous les modèles précédemment liés, mais aussi de supprimer toutes les entités héritées de ceux-ci (éléments, déclencheurs, etc.).

Note:
Zabbix propose un vaste ensemble de modèles prédéfinis. Vous pouvez les utiliser comme référence, mais évitez de les utiliser tels quels en production, car ils peuvent contenir trop d'éléments et interroger les données trop fréquemment. Si vous souhaitez les utiliser, ajustez-les afin qu'ils correspondent 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.

Dissociation d'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 *Templates*.
3. Cliquez sur *Unlink* ou *Unlink and clear* à côté du modèle pour le dissocier.
4. Cliquez sur *Update* dans le formulaire des attributs de l'hôte.

Le choix de l'option *Unlink* supprimera simplement l'association avec le modèle, tout en conservant toutes ses entités sur l'hôte. 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 mappages de valeurs et les tags hérités des modèles liés seront également supprimés.

Le choix de l'option *Unlink and clear* supprimera à la fois l'association avec le modèle et toutes ses entités (éléments, déclencheurs, etc.).

4 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 *Templates*.
3. Cliquez sur *Select* pour ouvrir la fenêtre contextuelle *Templates*.
4. Dans la fenêtre contextuelle, choisissez les modèles requis, puis cliquez sur *Select* pour ajouter les modèles à la liste.
5. Cliquez sur *Add* ou *Update* 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 *Unlink* ou *Unlink and clear* dans le formulaire de configuration du modèle, puis cliquez sur *Update*.

L'option *Unlink* supprimera simplement l'association avec le modèle lié, sans supprimer aucune de ses entités (éléments, déclencheurs, etc.).

L'option *Unlink and clear* supprimera à la fois l'association avec le modèle lié, ainsi que toutes ses entités (éléments, déclencheurs, etc.).

5 Mise à jour en masse

Aperçu

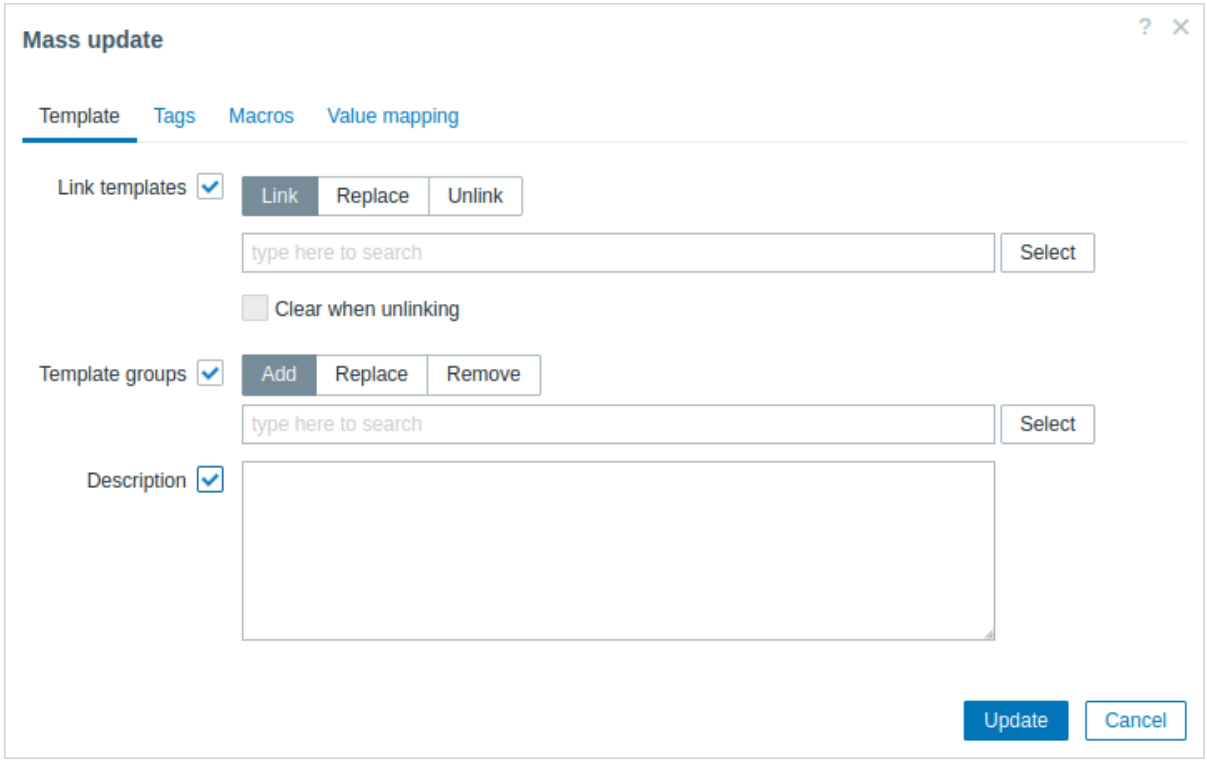
Parfois, vous souhaiterez peut-être modifier un attribut pour plusieurs modèles à la fois. Au lieu d'ouvrir chaque modèle individuel pour le modifier, vous pouvez utiliser la fonction de mise à jour en masse pour cela.

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 *Correspondance des valeurs*).
4. Cochez les cases des attributs à mettre à jour et saisissez une nouvelle valeur pour ceux-ci.

L'onglet **Template** contient les options générales de mise à jour de masse des modèles.



Les options suivantes sont disponibles lors de la sélection du bouton correspondant pour la mise à jour *Link templates* :

- *Link* - permet de spécifier quels modèles supplémentaires lier ;
- *Replace* - permet de spécifier quels modèles lier tout en déliant en même temps tous les modèles précédemment liés ;
- *Unlink* - permet de spécifier quels modèles délier.

Pour spécifier les modèles à lier/délier, commencez à saisir le nom du modèle dans le champ d'auto-complétion jusqu'à ce qu'une liste déroulante apparaisse en proposant les modèles correspondants. Faites simplement défiler la liste pour sélectionner les modèles requis.

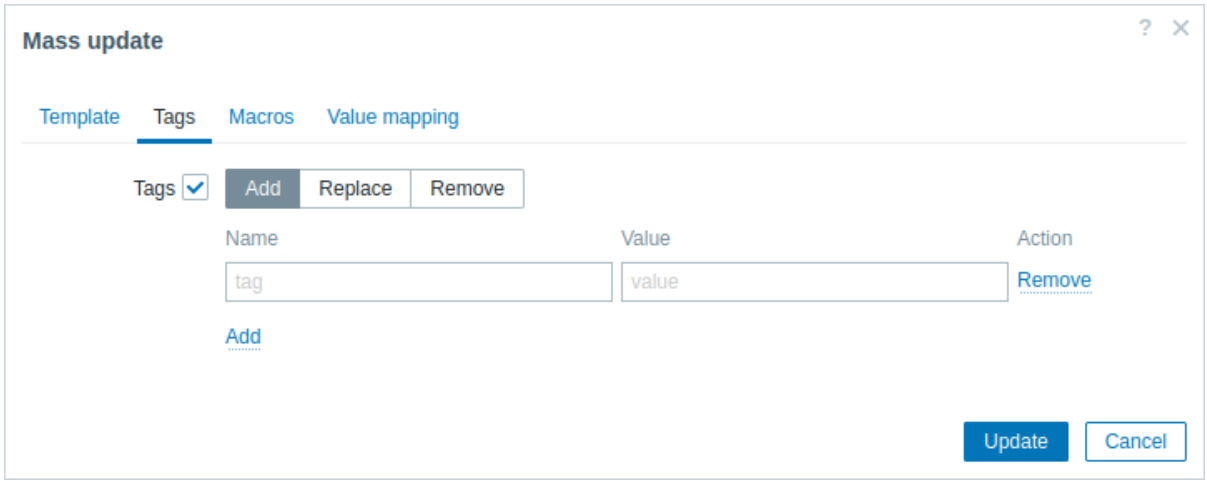
L'option *Clear when unlinking* permet de délier tous les modèles précédemment liés, ainsi que de supprimer tous les éléments hérités de ceux-ci (éléments, déclencheurs, graphiques, etc.).

Les options suivantes sont disponibles lors de la sélection du bouton correspondant pour la mise à jour *Template groups* :

- *Add* - permet de spécifier des groupes de modèles supplémentaires parmi les groupes existants ou de saisir des groupes de modèles entièrement nouveaux pour les modèles ;
- *Replace* - supprimera le modèle de tous les groupes de modèles existants et les remplacera par celui ou ceux spécifiés dans ce champ (groupes de modèles existants ou nouveaux) ;
- *Remove* - supprimera des groupes de modèles spécifiques des modèles.

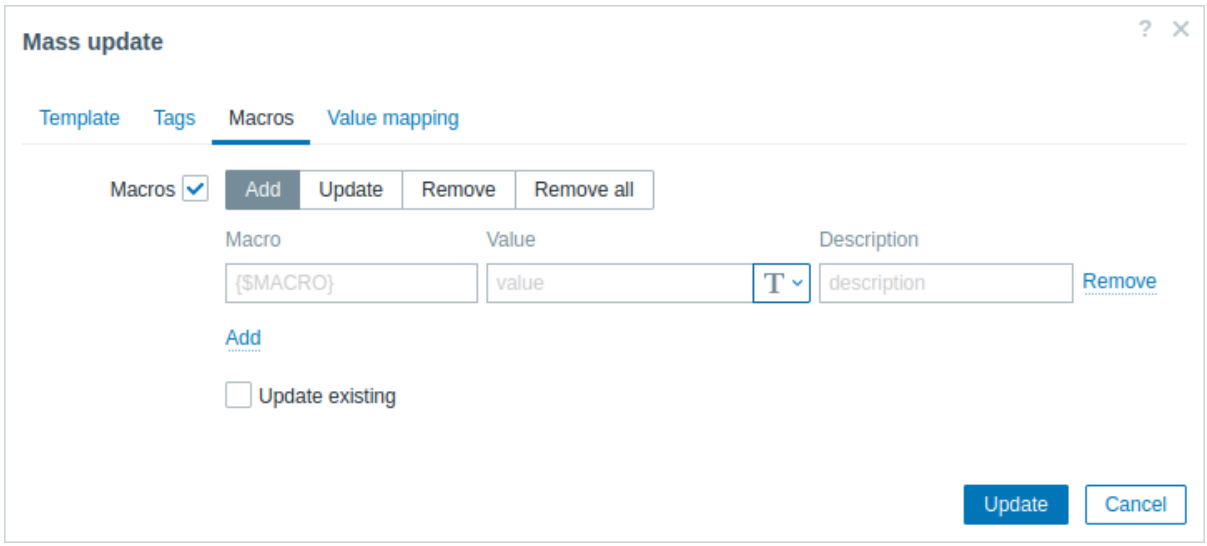
Ces champs prennent en charge l'auto-complétion : commencer à y saisir du texte affiche une liste déroulante des groupes de modèles correspondants. Si le groupe de modèles est nouveau, il apparaît également dans la liste déroulante et est indiqué par *(new)* après la chaîne. Faites simplement défiler la liste pour 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.

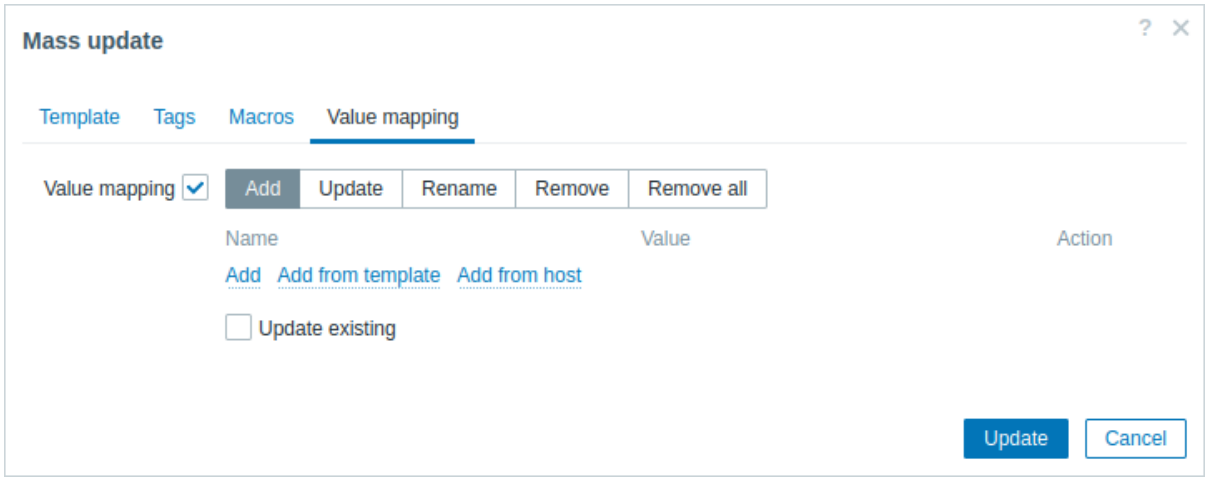
L'onglet **Macros** permet de mettre à jour en masse les macros au niveau des modèles.



Les options suivantes sont disponibles lors de la sélection du bouton correspondant pour la mise à jour des macros :

- *Add* - permet de spécifier des macros utilisateur supplémentaires pour les modèles. Si la case *Update existing* 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(s) modèle(s), elle ne sera pas mise à jour.
- *Update* - remplacera les valeurs, les types et les descriptions des macros spécifiées dans cette liste. Si la case *Add missing* est cochée, une macro qui n'existait pas auparavant sur un modèle sera ajoutée comme nouvelle macro. Si elle n'est pas cochée, seules les macros qui existent déjà sur un modèle seront mises à jour.
- *Remove* - supprimera les macros spécifiées des modèles. Si la case *Except selected* est cochée, toutes les macros sauf celles spécifiées dans la liste seront supprimées. Si elle n'est pas cochée, seules les macros spécifiées dans la liste seront supprimées.
- *Remove all* - supprimera toutes les macros utilisateur des modèles. Si la case *I confirm to remove all macros* n'est pas cochée, une nouvelle fenêtre contextuelle s'ouvrira pour demander la confirmation de la suppression de toutes les macros.

L'onglet **Correspondance des valeurs** permet de mettre à jour en masse les correspondances des valeurs.



Des boutons avec les options suivantes sont disponibles pour la mise à jour des correspondances des valeurs :

- *Ajouter* - ajouter des correspondances des valeurs aux modèles. Si vous cochez *Mettre à jour l'existant*, toutes les propriétés de la correspondance des valeurs portant ce nom seront mises à jour. Sinon, si une correspondance des valeurs portant ce nom existe déjà, elle ne sera pas mise à jour.
- *Mettre à jour* - mettre à jour les correspondances des valeurs existantes. Si vous cochez *Ajouter les éléments manquants*, une correspondance des valeurs qui n'existait pas auparavant sur un modèle sera ajoutée comme nouvelle correspondance des valeurs. Sinon, seules les correspondances des valeurs qui existent déjà sur un modèle seront mises à jour.
- *Renommer* - attribuer un nouveau nom à une correspondance des valeurs existante.
- *Supprimer* - supprimer les correspondances des valeurs spécifiées des modèles. Si vous cochez *Sauf la sélection*, toutes les correspondances des valeurs seront supprimées **à l'exception** de celles qui sont spécifiées.

- *Tout supprimer* - supprimer toutes les correspondances des valeurs des modèles. Si la case à cocher *Je confirme la suppression de toutes les correspondances des valeurs* n'est pas cochée, une nouvelle fenêtre contextuelle s'ouvrira pour demander la confirmation de la suppression.

Les options *Ajouter depuis un modèle* et *Ajouter depuis un hôte* sont disponibles pour les opérations d'ajout/mise à jour des correspondances des valeurs.
Elles permettent de sélectionner des correspondances des valeurs depuis un modèle ou un hôte respectivement.

Une fois toutes les modifications requises effectuées, cliquez sur *Mettre à jour*. Les attributs seront alors mis à jour en conséquence pour tous les modèles sélectionnés.

1 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.

1 Assistant d'hôte

Aperçu

L'assistant d'hôte est une interface guidée, étape par étape, permettant de configurer votre cible de supervision (appareil, 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 Zabbix agent ou agent 2
- Ajout d'une interface d'hôte
- Application d'une configuration supplémentaire à votre cible de supervision ou à l'hôte Zabbix (si le modèle l'exige)

Pour accéder à l'assistant d'hôte dans l'interface web de Zabbix, procédez comme suit :

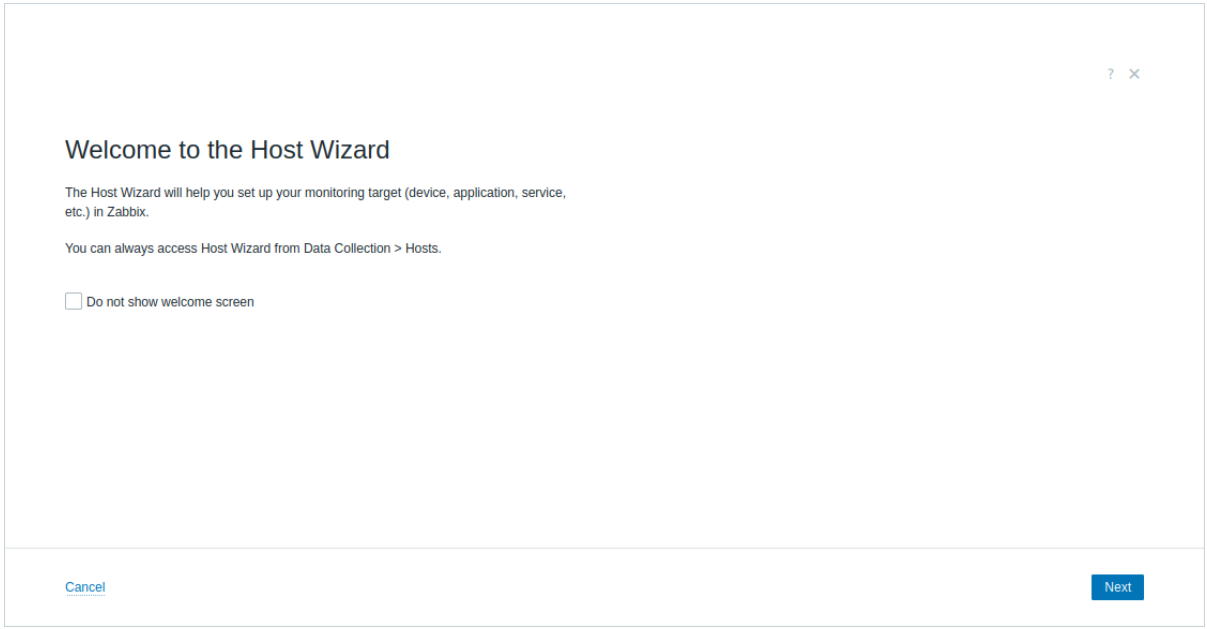
- Allez dans : *Collecte de données* > *Hôtes*
- Cliquez sur *Assistant d'hôte* dans le coin supérieur droit de l'écran
- Suivez les instructions de l'assistant

Vous pouvez également modifier un hôte existant en cliquant sur son nom (ou sur l'icône à trois points située à côté) dans différentes sections de l'interface web 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 également utiliser l'approche classique pour créer et configurer un hôte.

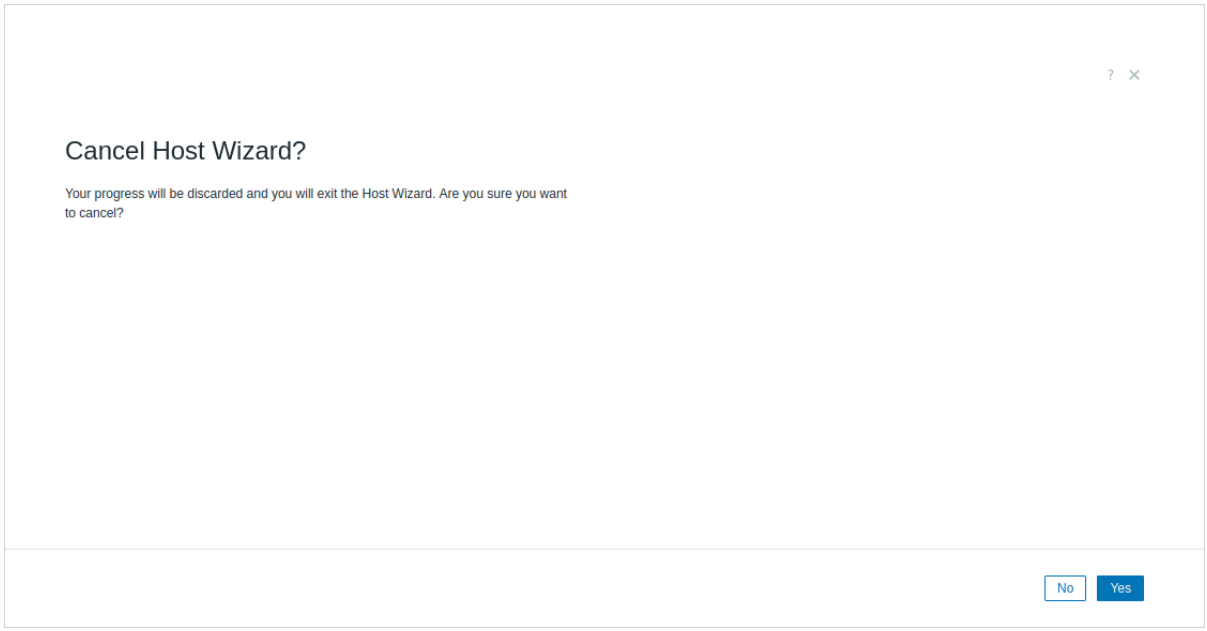
Bienvenue dans l'assistant d'hôte

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 essayez de fermer l'assistant en cliquant sur l'icône de fermeture ou en appuyant sur Échap, un écran de confirmation s'affichera :



Cliquer sur *Oui* fermera l'assistant d'hôte sans enregistrer votre progression. Cliquer sur *Non*, sur Échap 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 — un ensemble de configurations prédéfinies (métriques à collecter, conditions de génération d'alertes, etc.) conçu pour votre cible de supervision.

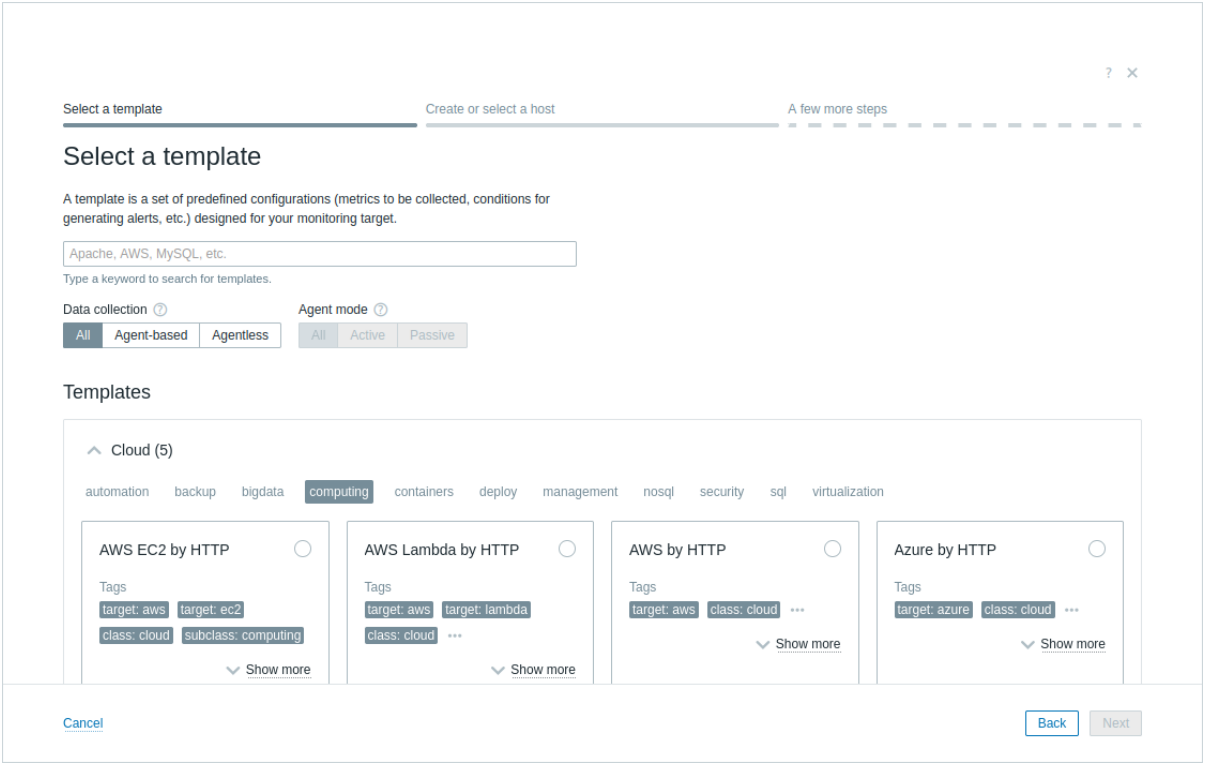
Les modèles prêts à l'emploi de Zabbix offrent diverses configurations de supervision prédéfinies pour les systèmes d'exploitation, les applications, les bases de données, les équipements réseau, les services, et bien plus encore.

À cette étape, vous pouvez :

- Parcourir les modèles par classe (Cloud, Database, Network, etc.) et sous-classe (par exemple, automation, backup), qui sont basées sur les tags des modèles.
- Rechercher des modèles par mots-clés dans le nom du modèle ou dans les noms et valeurs des tags du modèle.
- Filtrer les modèles par méthode de collecte des données (avec agent ou sans agent ; les modèles avec agent incluent au moins un élément Zabbix agent).
- Filtrer les modèles par mode d'agent (actif ou passif ; les modèles actifs incluent au moins un élément Zabbix agent (active), tandis que les modèles passifs incluent au moins un élément passif ; pour plus de détails, voir Contrôles d'agent passifs et

actifs).

- 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, vous verrez le message suivant : *Some templates (n) are incompatible with the Host Wizard. See how to update them. Custom templates are not supported.* 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 ; voir les instructions de mise à niveau.

Le modèle que vous sélectionnez déterminera les étapes suivantes dans 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 host, c'est-à-dire une entité dans Zabbix qui représente 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 d'alertes), nécessaires à la supervision.

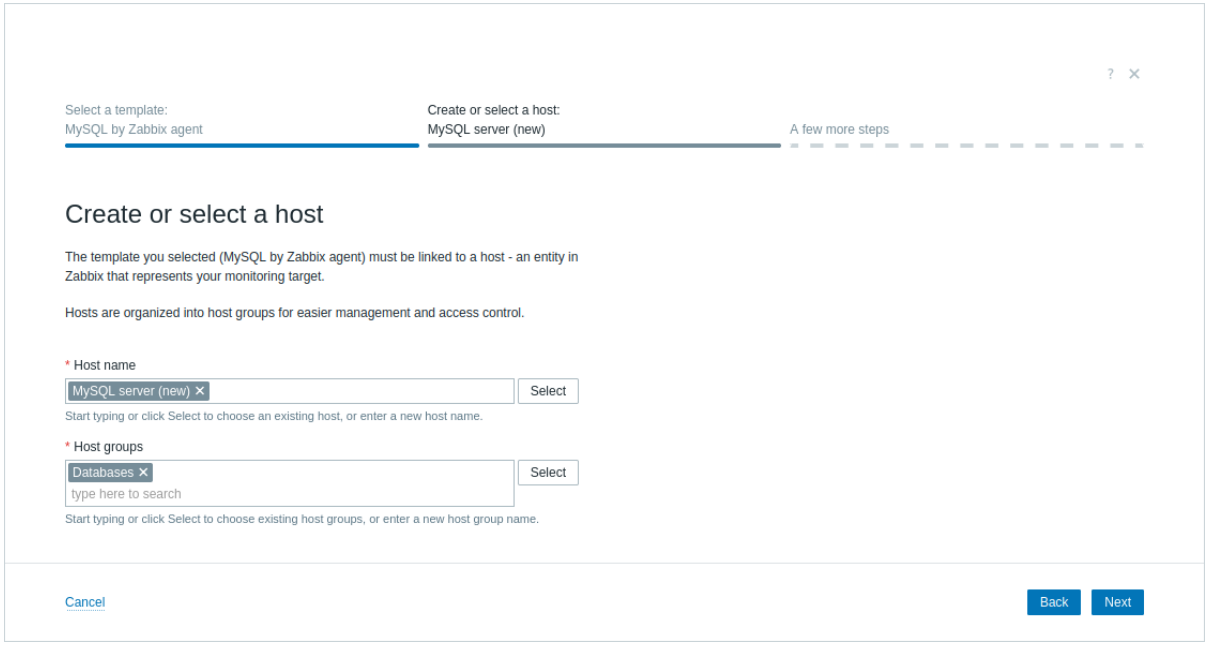
Chaque hôte doit également appartenir à au moins un groupe d'hôtes, utilisé pour organiser les hôtes et leur attribuer des permissions utilisateur.

À cette étape, vous pouvez :

- Créer un nouvel hôte et un nouveau groupe d'hôtes en saisissant leurs noms.
- Créer un nouvel hôte en saisissant son nom, puis sélectionner un groupe d'hôtes existant auquel le nouvel hôte sera attribué.
- Sélectionner un hôte existant ; vous pouvez également attribuer cet hôte à des groupes d'hôtes supplémentaires sans le retirer de ses groupes actuels.

Note:
La sélection d'un hôte existant ne supprime ni n'écrase 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 *MySQL server* pour représenter le serveur MySQL installé localement que vous souhaitez superviser. Le modèle *MySQL by Zabbix agent* précédemment sélectionné sera lié à cet hôte. En outre, vous pouvez sélectionner le groupe d'hôtes existant *Databases* (ou en créer un nouveau, par exemple *MySQL servers*) 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 l'agent (tels que *MySQL by Zabbix agent*) nécessitent l'installation de Zabbix agent ou agent 2 — un processus déployé sur votre cible de supervision afin de surveiller activement les ressources locales et les applications (si le modèle l'exige).

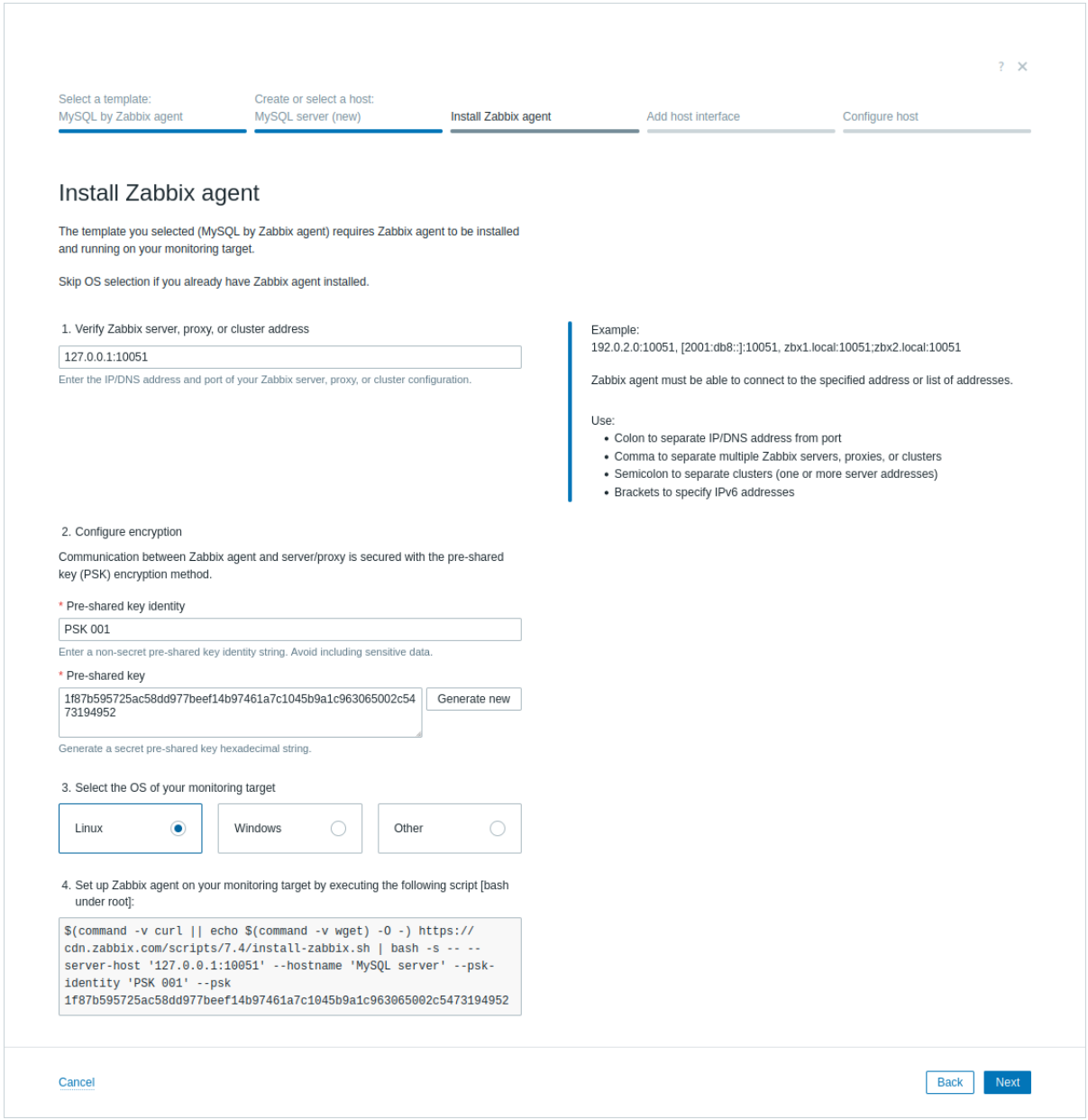
À cette étape :

- Vérifiez l'adresse du serveur, du proxy ou du cluster Zabbix (par exemple, 192.0.2.0:10051). Cette valeur sera utilisée pour configurer le paramètre Server et ServerActive pour l'agent Zabbix.
- Saisissez une nouvelle identité de clé pré-partagée unique et non secrète (par exemple, *PSK 001* ou *mysql-agent-psk1*), puis générez la clé pré-partagée. Ces valeurs seront utilisées pour configurer le chiffrement par clé pré-partagée (PSK) pour l'hôte Zabbix (par exemple, *serveur MySQL*) et l'agent Zabbix.

Attention:
Lors de la modification d'un hôte existant, la configuration 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 y exécutant le script fourni ou en suivant les instructions d'installation.

Par exemple, vous pouvez définir *Pre-shared key identity* sur *PSK 001* et générer une nouvelle *Pre-shared key*. Ensuite, vous pouvez sélectionner *Linux* comme système d'exploitation et exécuter le script fourni sur ce système. Après avoir installé l'agent Zabbix, revenez à l'assistant d'hôte.

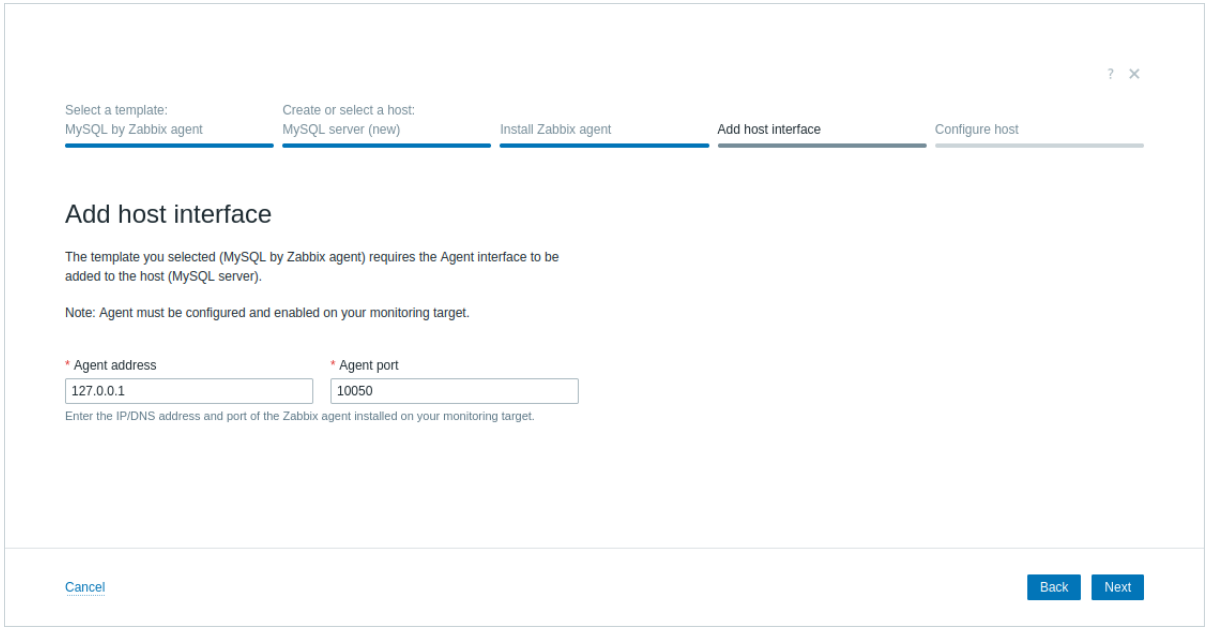


Ajouter une interface d'hôte

L'interface d'hôte définit la manière dont 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 des données requise par le modèle sélectionné.

À cette étape, saisissez les détails de l'interface d'hôte requis par le modèle sélectionné.

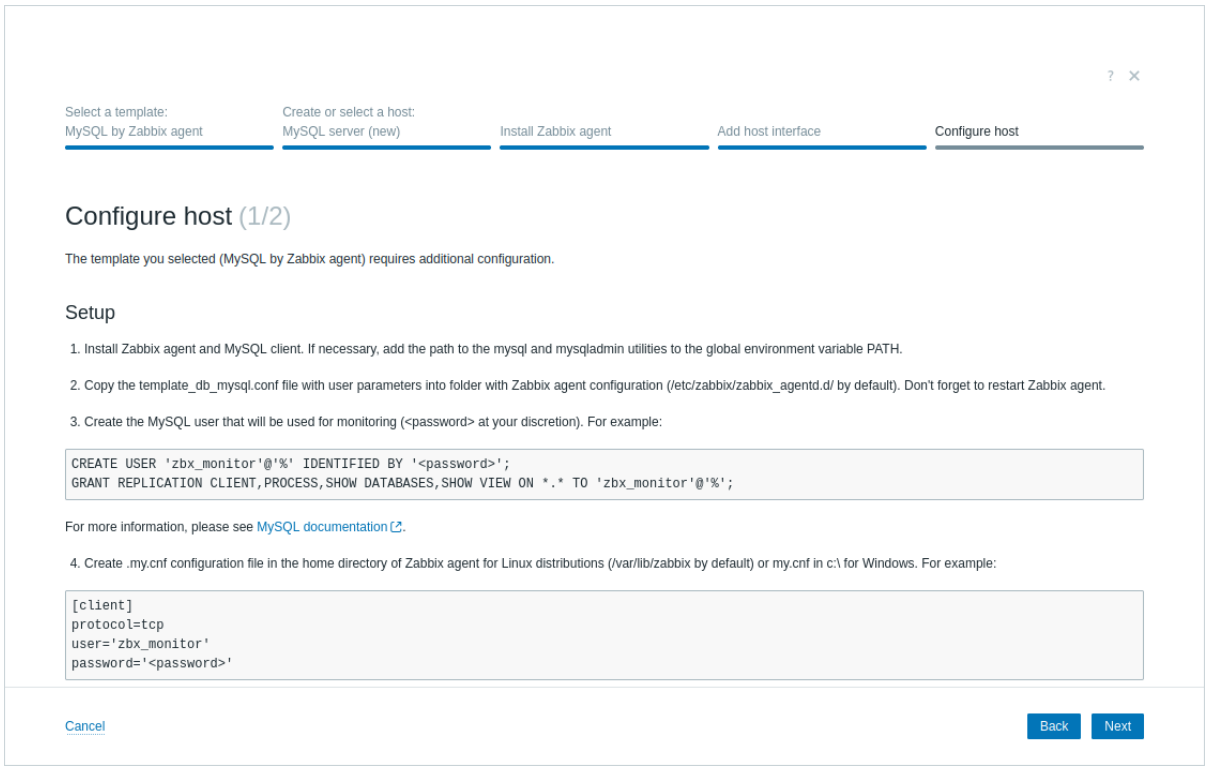
Par exemple, vous pouvez utiliser l'adresse d'agent par défaut (127.0.0.1) et le port (10050) si le serveur Zabbix, l'agent Zabbix et le serveur MySQL s'exécutent tous localement sur la même machine.



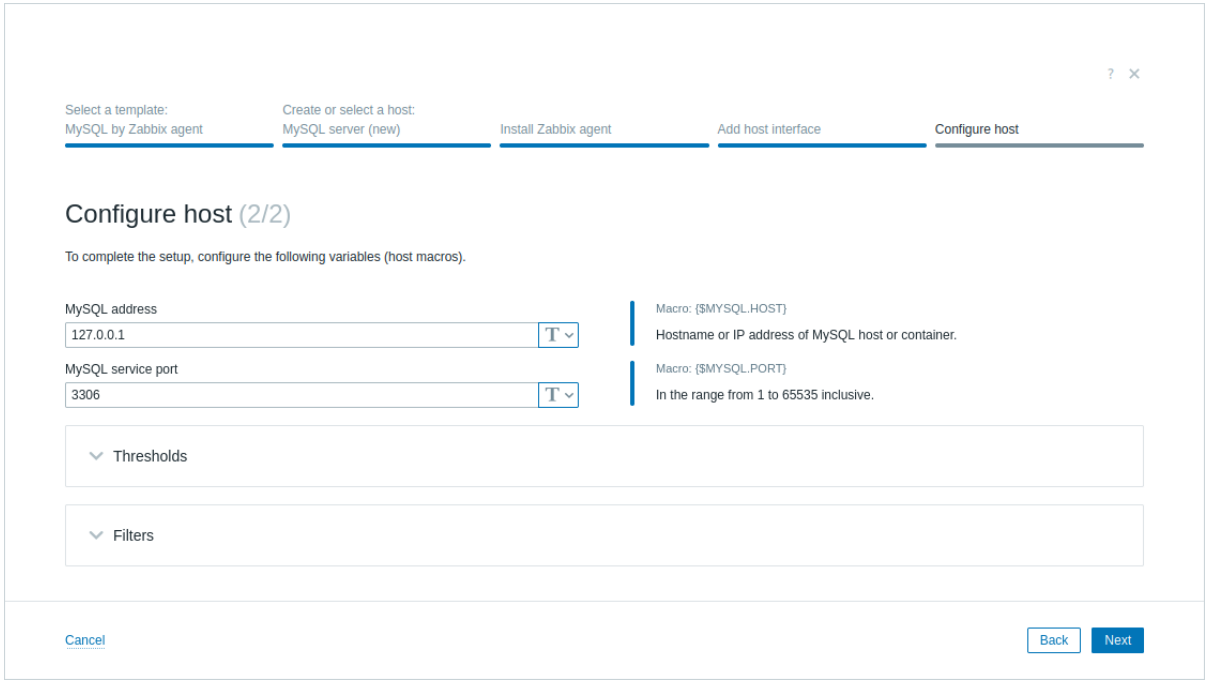
Configurer l'hôte

Certains modèles nécessitent une configuration supplémentaire avant que l'hôte puisse être créé. Cela peut inclure :

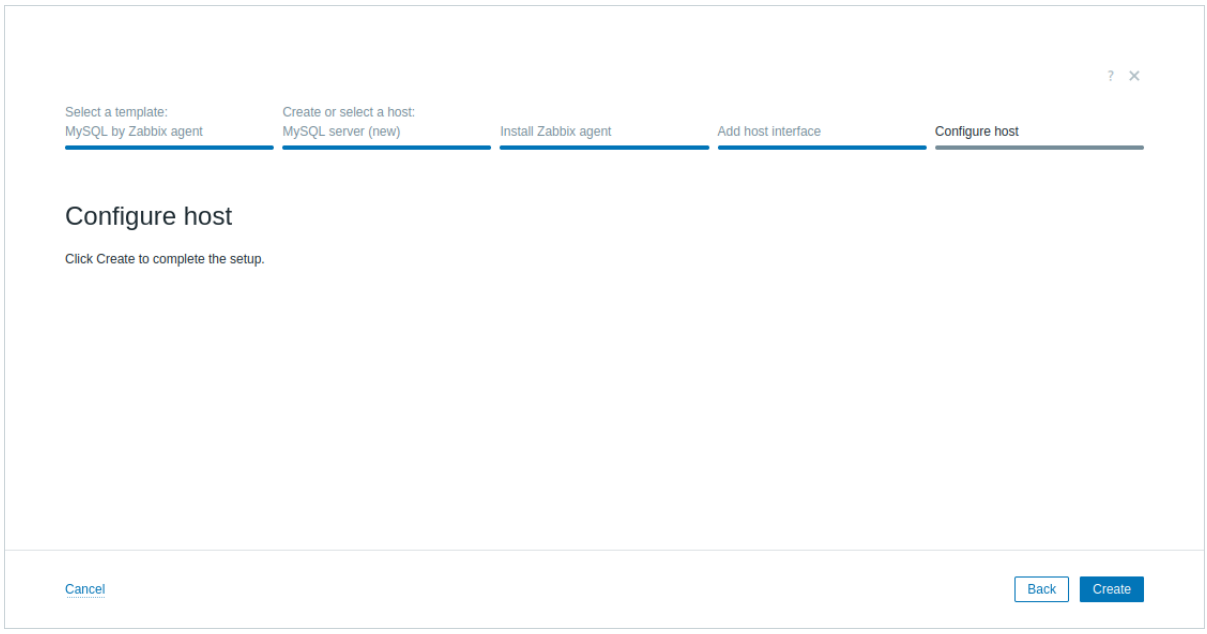
- La configuration manuelle de votre cible de supervision (par exemple, l'activation de services spécifiques, la création d'utilisateurs de service ou l'octroi d'autorisations) :



- La définition de macros d'hôte — variables qui contrôlent le comportement des éléments, les paramètres de connexion, les identifiants d'authentification et d'autres paramètres :



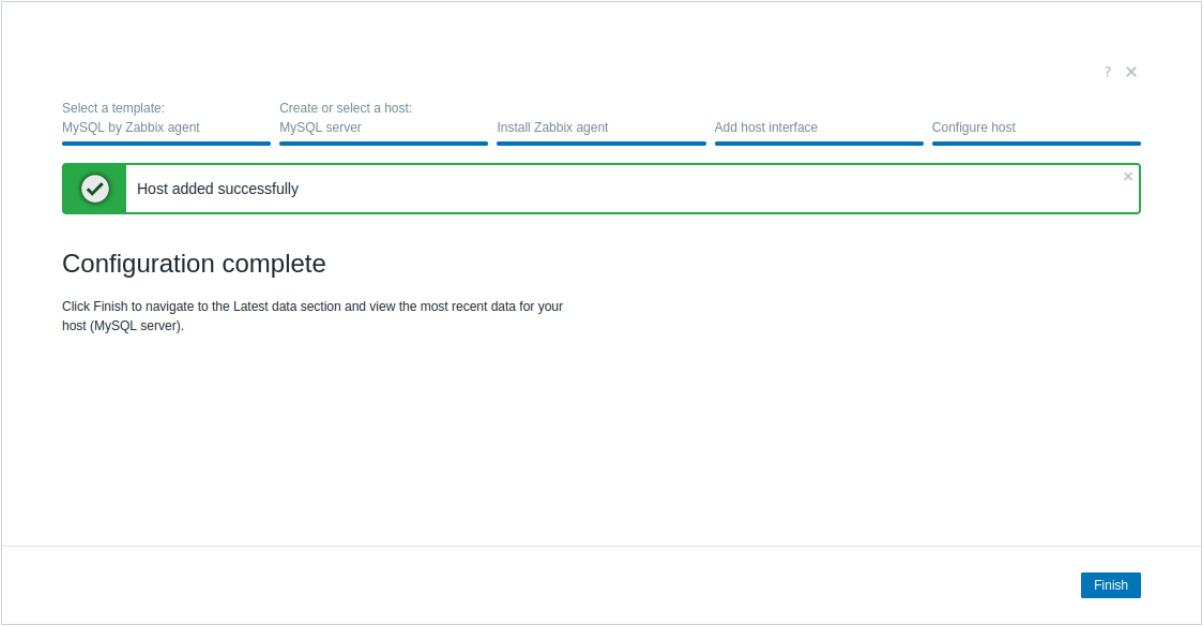
Après avoir terminé toutes les étapes de configuration requises, cliquez sur *Create* pour ajouter l'hôte à Zabbix (ou sur *Update* pour mettre à jour la configuration d'un hôte existant) :



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.



2 Configuration d'un hôte

Aperçu

Pour configurer un hôte dans l'interface web de Zabbix, procédez comme suit :

- Accédez à : *Data collection* > *Hosts* ou *Monitoring* > *Hosts*
- Cliquez sur *Create host* dans le coin supérieur droit de l'écran (ou sur le nom de l'hôte pour modifier un hôte existant)
- Saisissez les paramètres de l'hôte dans le formulaire

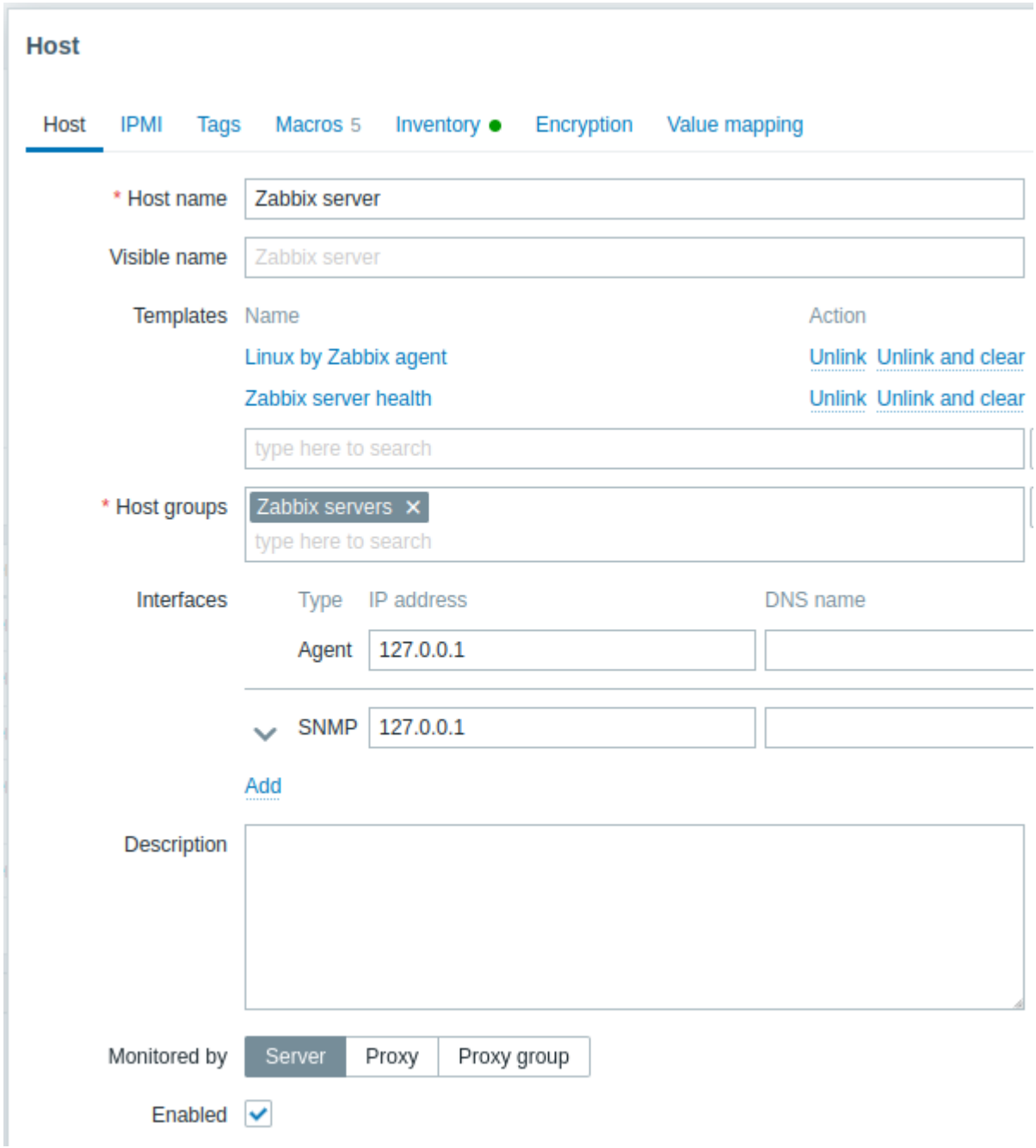
Vous pouvez également utiliser le bouton *Clone* dans le formulaire de configuration d'un hôte existant pour créer un nouvel hôte. Cet hôte aura toutes les propriétés de l'hôte existant, y compris les modèles liés, les entités (éléments, déclencheurs, etc.) de ces modèles, ainsi que les entités directement rattachées à l'hôte existant.

Notez que lorsqu'un hôte est cloné, il conserve toutes les entités de modèle telles qu'elles sont à l'origine 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 de bas niveau) ne sera pas clonée vers le nouvel hôte ; elles seront au contraire identiques à celles du modèle.

Vous pouvez également utiliser l'assistant 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 de saisie obligatoires sont marqués d'un astérisque rouge.

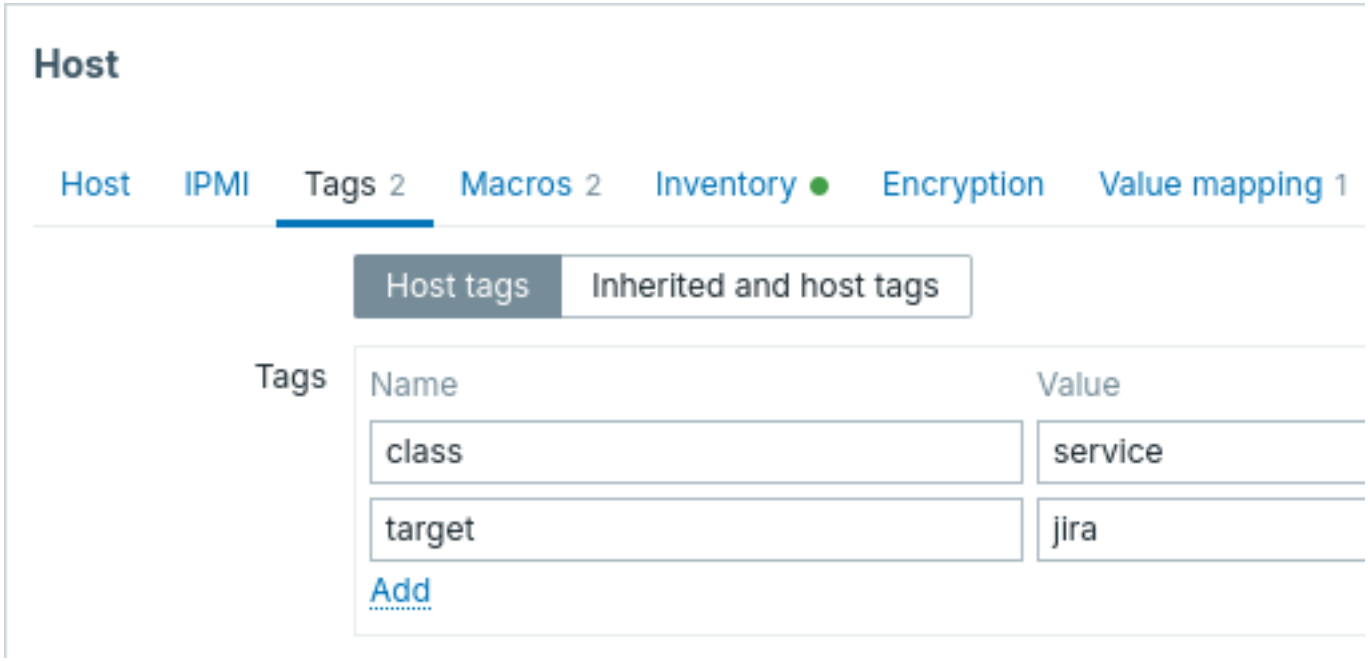
Paramètre	Description
Nom de l'hôte	Saisissez un nom d'hôte unique. Les caractères alphanumériques, les espaces, les points, les tirets et les traits de soulignement sont autorisés. Cependant, les espaces en début et en fin sont interdits. *Remarque :* avec l'agent Zabbix exécuté sur l'hôte que vous 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 dans le paramètre est nécessaire au traitement des vérifications actives.
Nom visible	Saisissez un nom visible unique pour l'hôte. Si vous définissez ce nom, c'est celui qui sera affiché dans les listes, les cartes, etc., au lieu 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 le nom du modèle dans le champ de saisie. Une liste des modèles correspondants apparaîtra ; faites défiler pour sélectionner. 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 lorsque le formulaire de configuration de l'hôte sera enregistré ou mis à jour. 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 de modèles listés sont des liens cliquables menant au formulaire de configuration du modèle.
Groupes d'hôtes	Sélectionnez les groupes d'hôtes auxquels l'hôte appartient. Un hôte doit appartenir à au moins un groupe d'hôtes. Un nouveau groupe peut être créé et lié à l'hôte en ajoutant un nom de groupe inexistant.
Interfaces	Plusieurs types d'interfaces d'hôte sont pris en charge pour un hôte : *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 celles-ci. L'« IP » ou le « DNS » d'une interface SNMP est également utilisé pour les traps SNMP. Pendant la correspondance, seul l'« IP » ou le « DNS » sélectionné dans l'interface de l'hôte est utilisé. Voir Configuration de la supervision SNMP pour plus de détails 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	Cliquer sur le bouton correspondant indiquera au serveur Zabbix quoi 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 : 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 proxys** - l'hôte est supervisé par un groupe de proxys.
Proxy	Le nom du proxy attribué est affiché (uniquement si le serveur Zabbix en a attribué un à partir du groupe de proxys sélectionné). Ce champ n'est affiché que si l'hôte est supervisé par un groupe de proxys.
Activé	Lorsque la case est cochée, l'hôte est activé - prêt pour la supervision. Lorsque la case est décochée, l'hôte est désactivé - non supervisé : Pour les demandes de données passives initiées par le serveur/proxy Zabbix (par exemple, agent Zabbix, agent SNMP, vérifications simples), la supervision est désactivée après la synchronisation de la configuration. Les déclencheurs et 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) pendant lequel l'agent Zabbix reçoit l'information indiquant que l'hôte a été désactivé. Pendant ce bref intervalle, l'hôte continuera à collecter localement les données pour les vérifications actives et tentera de les envoyer au serveur/proxy ; toutefois, comme l'hôte est marqué comme *Désactivé*, le serveur/proxy rejettera les données. Lorsque vous désactivez l'hôte, ses éléments sont immédiatement supprimés du cache d'historique (à l'exception de leurs dernières valeurs, qui sont conservées pour les journaux).

L'onglet **IPMI** contient les attributs de gestion IPMI.

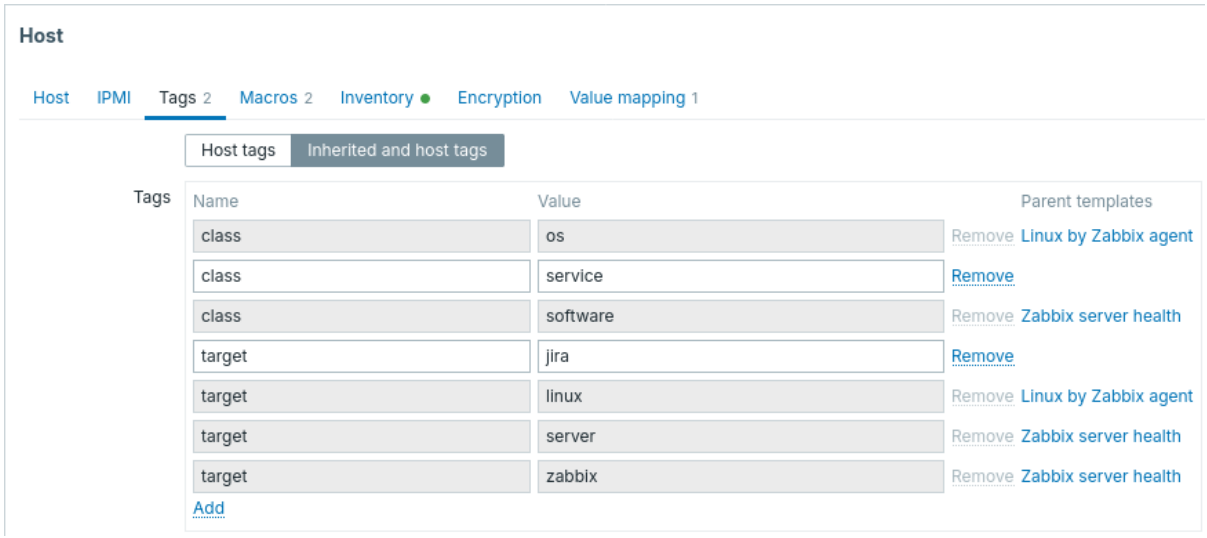
Paramètre	Description
Algorithme d'authentification	Sélectionnez l'algorithme d'authentification.
Niveau de privilège	Sélectionnez le niveau de privilège.
Nom d'utilisateur	Nom d'utilisateur pour l'authentification. Les macros utilisateur peuvent être utilisées.
Mot de passe	Mot de passe pour l'authentification. Les macros utilisateur peuvent être utilisées.

L'onglet **Balises** vous permet de définir des balises au niveau de l'hôte. Tous les problèmes de cet hôte seront balisés avec les valeurs saisies ici.



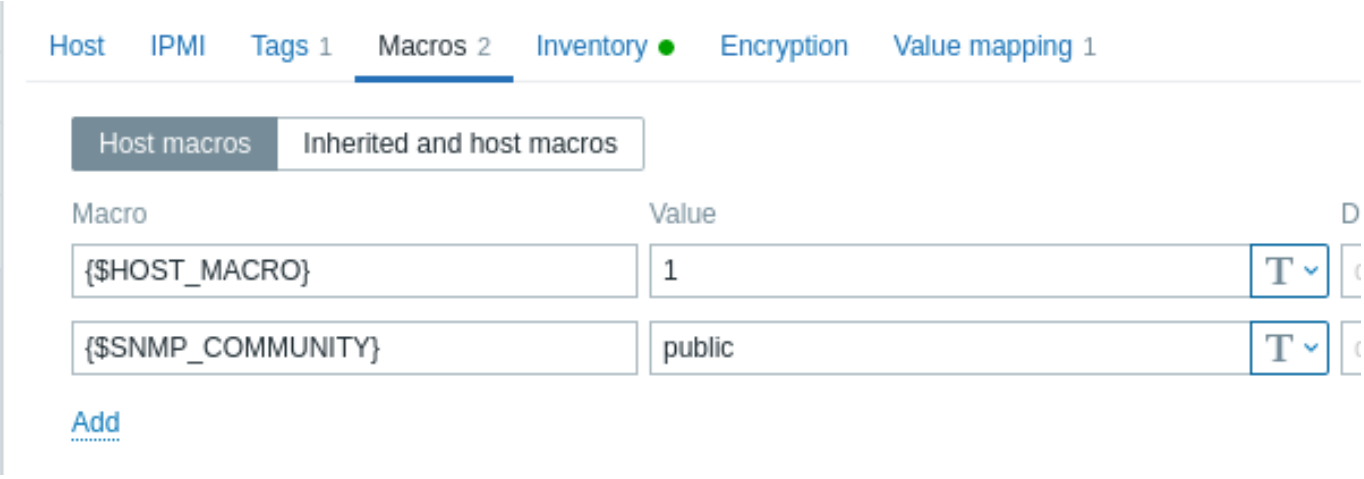
Les macros utilisateur, les macros {INVENTORY.*}, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} et {HOST.ID} sont prises en charge dans les balises.

Vous pouvez également afficher ici les balises au niveau du modèle si vous sélectionnez l'option *Balises héritées et de l'hôte*. C'est là que toutes les balises utilisateur définies pour l'hôte sont affichées ainsi que leur origine.

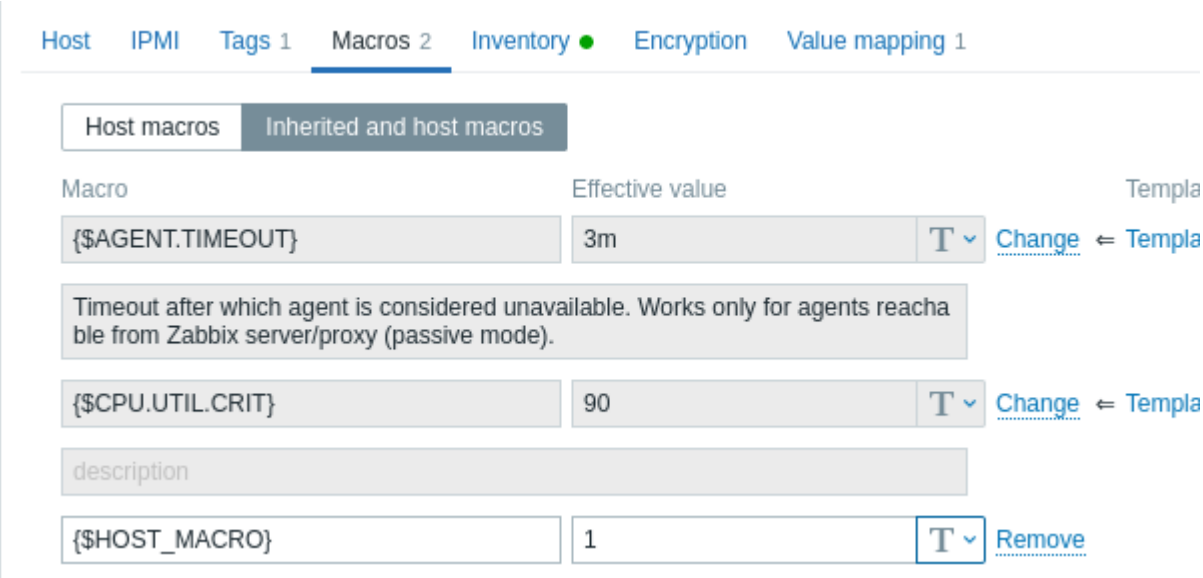


Pour plus de commodité, des liens vers les modèles correspondants sont fournis.

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 en texte brut, texte secret ou secret Vault. L'ajout d'une description est également pris en charge.

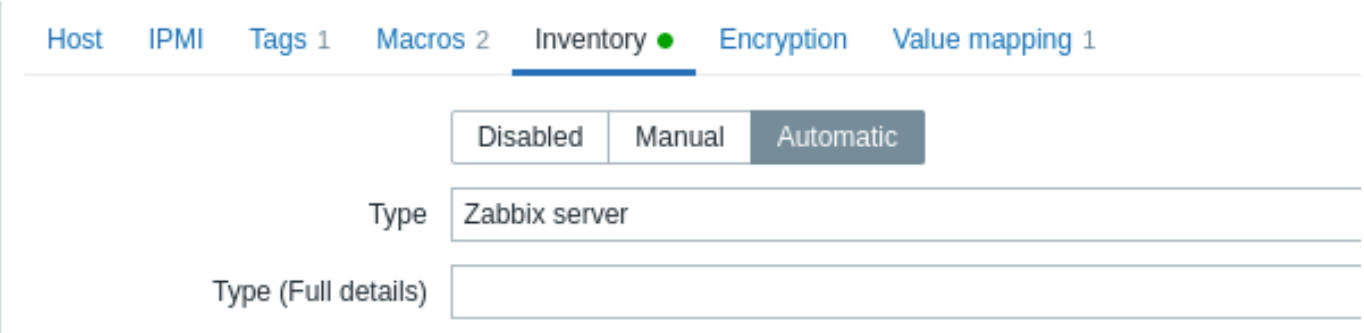


Vous pouvez également afficher ici les macros utilisateur au niveau du modèle et au niveau global si vous sélectionnez l'option *Macros héritées et de l'hôte*. C'est là que toutes les macros utilisateur définies pour l'hôte sont affichées avec la valeur à laquelle elles se résolvent ainsi que leur origine.



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/globale au niveau de l'hôte, créant ainsi effectivement une copie de la macro sur l'hôte.

L'onglet **Inventaire** vous permet de saisir manuellement les informations d'inventaire pour 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 (manuel ou automatique), un point vert est affiché avec le nom de l'onglet.

Chiffrement

L'onglet **Chiffrement** vous permet d'exiger des connexions chiffrées avec l'hôte.

Paramètre	Description
Connexions vers l'hôte	Définit comment le serveur ou le proxy Zabbix se connecte à l'agent Zabbix sur un hôte : sans chiffrement (par défaut), à l'aide d'une PSK (clé pré-partagée) ou d'un certificat.
Connexions depuis l'hôte	Sélectionnez le type de connexions autorisées depuis l'hôte (c'est-à-dire depuis l'agent Zabbix et Zabbix sender). Plusieurs types de connexion peuvent être sélectionnés en même temps (utile pour les tests et le basculement vers un autre type de connexion). La valeur par défaut est « Sans chiffrement ».
Émetteur	Émetteur autorisé du certificat. Le certificat est d'abord validé par l'AC (autorité de certification). S'il est valide et signé par l'AC, le champ *Émetteur* peut alors être utilisé pour restreindre davantage les AC autorisées. 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.
Sujet	Sujet autorisé du certificat. Le certificat est d'abord validé par l'AC. S'il est valide et signé par l'AC, le champ *Sujet* peut alors être utilisé pour n'autoriser qu'une seule valeur de la chaîne *Sujet*. Si ce champ est vide, tout certificat valide signé par l'AC configurée est accepté.
Identité PSK	Chaîne d'identité de la clé pré-partagée. Ne mettez pas d'informations sensibles dans l'identité PSK, car elle est transmise sans chiffrement sur le réseau afin d'indiquer au destinataire quelle PSK utiliser.
PSK	Clé pré-partagée (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

Correspondance des valeurs

L'onglet **Correspondance des valeurs** permet de configurer une représentation conviviale des données d'élément dans les correspondances de valeurs.

3 Configuration d'un groupe d'hôtes

Aperçu

Les groupes d'hôtes sont utilisés pour le regroupement logique des hôtes et pour leur attribuer des autorisations utilisateur.

Chaque hôte doit avoir au moins un groupe d'hôtes attribué. Un hôte peut appartenir à plusieurs groupes d'hôtes, et chaque groupe d'hôtes peut contenir plusieurs hôtes.

Notez que dans Zabbix, toutes les autorisations sont basées sur des groupes : les groupes d'utilisateurs, les groupes d'hôtes et les 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 disposant de l'autorisation d'accéder au groupe d'hôtes contenant cet hôte.

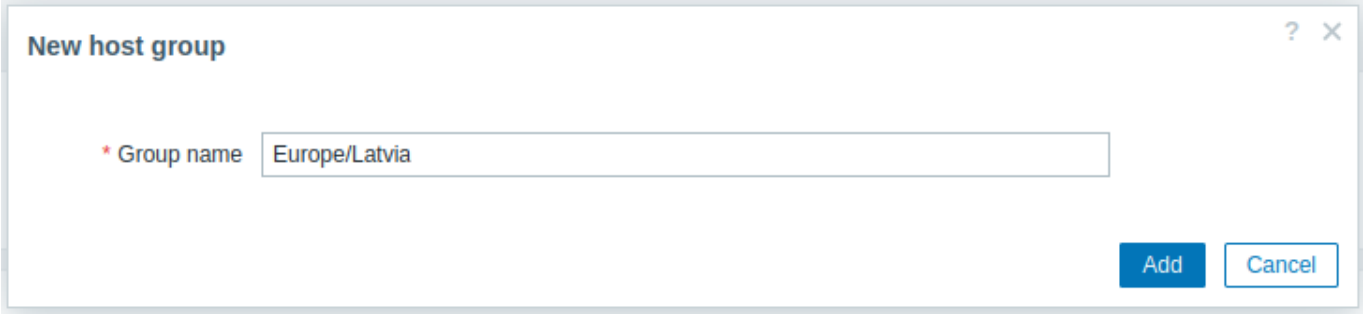
Configuration

> **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 web de Zabbix.

Option 1 :

- Allez dans : *Collecte de données* → *Groupes d'hôtes*
- Cliquez sur *Créer un groupe d'hôtes* dans le coin supérieur droit de l'écran
- Saisissez le nom du groupe dans le formulaire



Option 2 : lors de la configuration d'un hôte, saisissez un nom de groupe inexistant dans le champ *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 s'il s'agit du 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`.

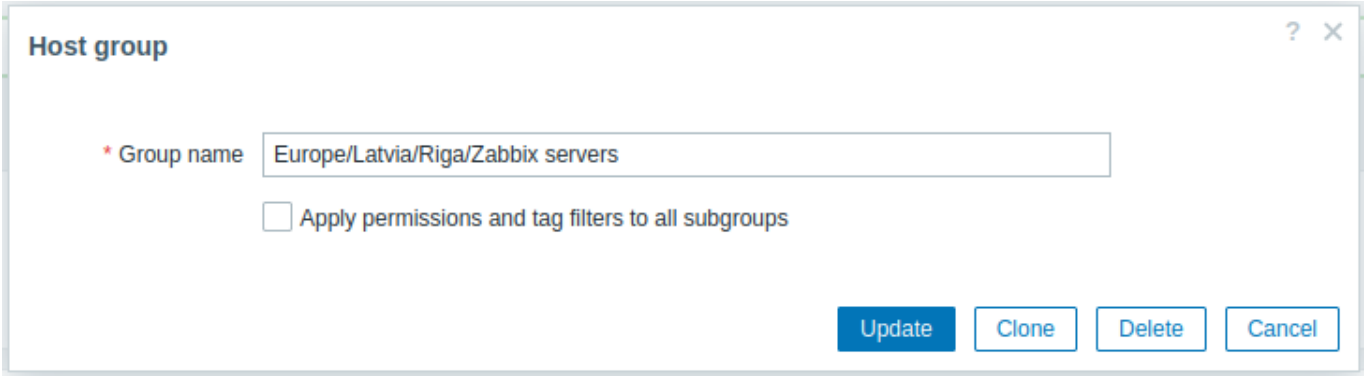
Lors de la création d'un sous-groupe, l'utilisation de barres obliques au début ou à la fin, ou de plusieurs barres obliques consécutives, n'est pas autorisée. L'échappement de « / » n'est pas pris en charge.

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 dans un groupe d'hôtes parent existant (par exemple, création de `Europe/Latvia` alors que `Europe` existe déjà), les autorisations du groupe d'utilisateurs sur le sous-groupe sont héritées du parent.
- Lors de la création d'un groupe d'hôtes parent pour un sous-groupe existant (par exemple, création de `Europe` alors que `Europe/Latvia` existe déjà), aucune autorisation n'est définie sur 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 tags à tous les sous-groupes*.



En cochant cette case et en cliquant sur *Mettre à jour*, le même niveau d'autorisations et les mêmes filtres de tags seront appliqués à tous les sous-groupes actuels et futurs du groupe d'hôtes en cours de modification.

Ainsi, si des groupes d'utilisateurs ont reçu 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 utilisateur et les mêmes autorisations basées sur les tags 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 autorisations existantes. Toute modification effectuée via cette option ne peut être annulée que manuellement.

4 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 détails tels que le type d'appareil, le numéro de série, l'emplacement, la personne responsable, les URL, etc. — des données qui 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 comme un lien cliquable dans la section *Inventaire*.

Mode automatique

L'inventaire de l'hôte peut également être renseigné automatiquement. Pour que cela fonctionne, lors de la configuration d'un hôte, le mode d'inventaire dans l'onglet *Inventaire* doit être défini sur *Automatique*.

Vous pouvez ensuite configurer les éléments d'hôte afin de renseigner n'importe quel champ de l'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.

Le mode d'inventaire par défaut pour les nouveaux hôtes est sélectionné en fonction du paramètre *Mode d'inventaire d'hôte par défaut* dans *Administration* → *Général* → *Autre*.

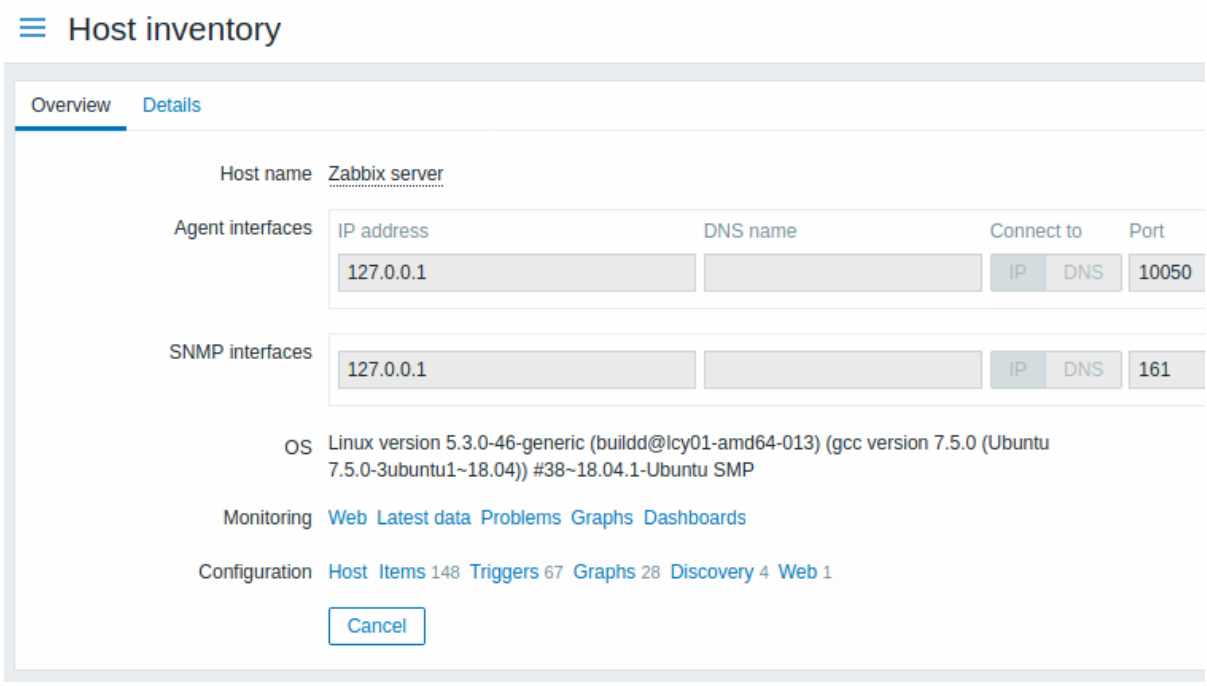
Pour les hôtes ajoutés par des actions de découverte de réseau ou d'enregistrement automatique, il est possible de définir une opération *Définir le mode d'inventaire des hôtes* en sélectionnant le mode manuel ou automatique. Cette opération remplace le paramètre *Mode d'inventaire de l'hôte par défaut*.

Aperçu de l'inventaire

Les détails de toutes les données d'inventaire existantes sont disponibles dans le menu *Inventory*.

Dans *Inventory* → *Overview*, vous pouvez obtenir un nombre d'hôtes selon différents champs de l'inventaire.

Dans *Inventory* → *Hosts*, vous pouvez voir tous les hôtes disposant d'informations d'inventaire. En cliquant sur le nom de l'hôte, les détails de l'inventaire s'affichent dans un formulaire.



L'onglet **Overview** affiche :

Parameter	Description
Host name	Nom de l'hôte. Un clic sur le nom ouvre un menu contenant 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*, *Latest data*, *Problems*, *Graphs*, *Dashboards*.
Configuration	Liens vers les sections de configuration pour cet hôte : *Host*, *Items*, *Triggers*, *Graphs*, *Discovery*, *Web*. Le nombre d'entités configurées est indiqué après chaque lien.

L'onglet **Details** 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}, numéro de téléphone {INVENTORY.POC.PRIMARY.PHONE.A1}."

Pour plus de détails, consultez la page Macros prises en charge.

5 Mise à jour en masse

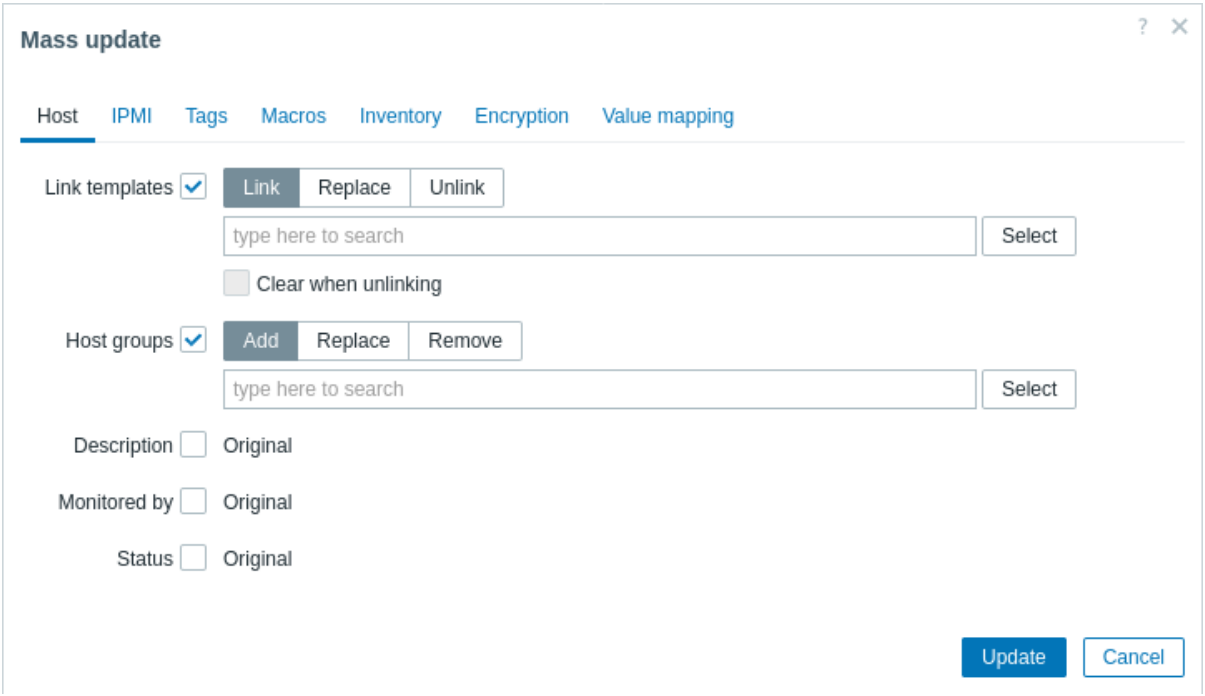
Aperçu

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 une seule fois, 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*, *Tags*, *Macros*, *Inventaire*, *Chiffrement* ou *Correspondance 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 de l'association de **modèles** :

- *Lier* - spécifier quels modèles supplémentaires lier
- *Remplacer* - spécifier quels modèles lier tout en dissociant tout modèle qui était auparavant lié aux hôtes
- *Dissocier* - spécifier quels modèles dissocier

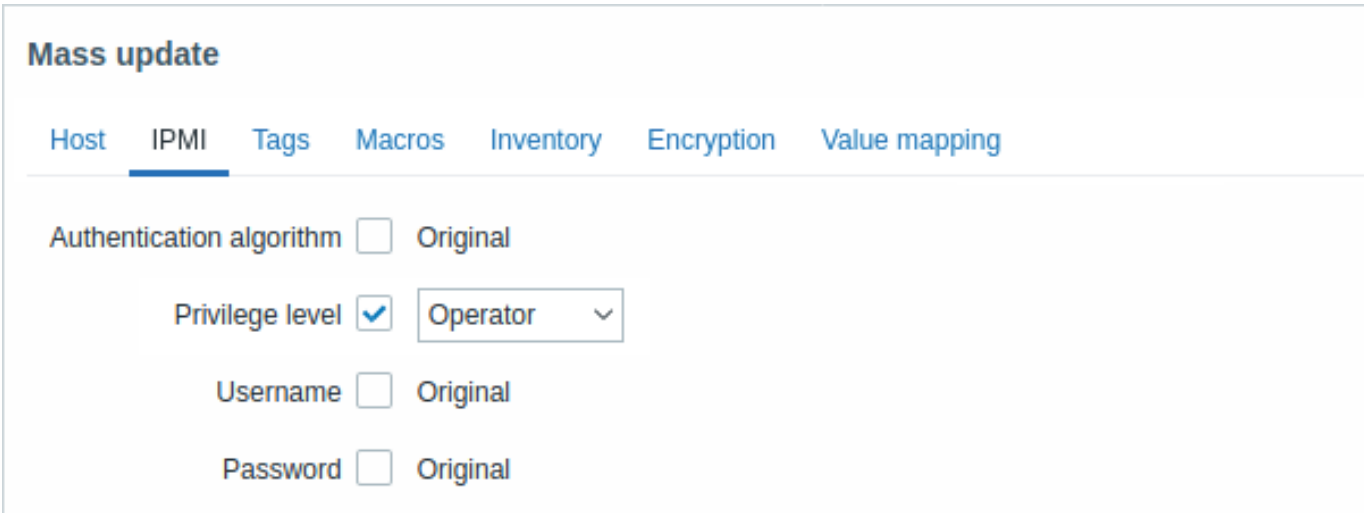
Pour spécifier les modèles à lier/dissocier, commencez à saisir le nom du modèle dans le champ d'auto-complétion jusqu'à ce qu'une liste déroulante apparaisse proposant les modèles correspondants. Faites simplement défiler la liste pour sélectionner le modèle requis.

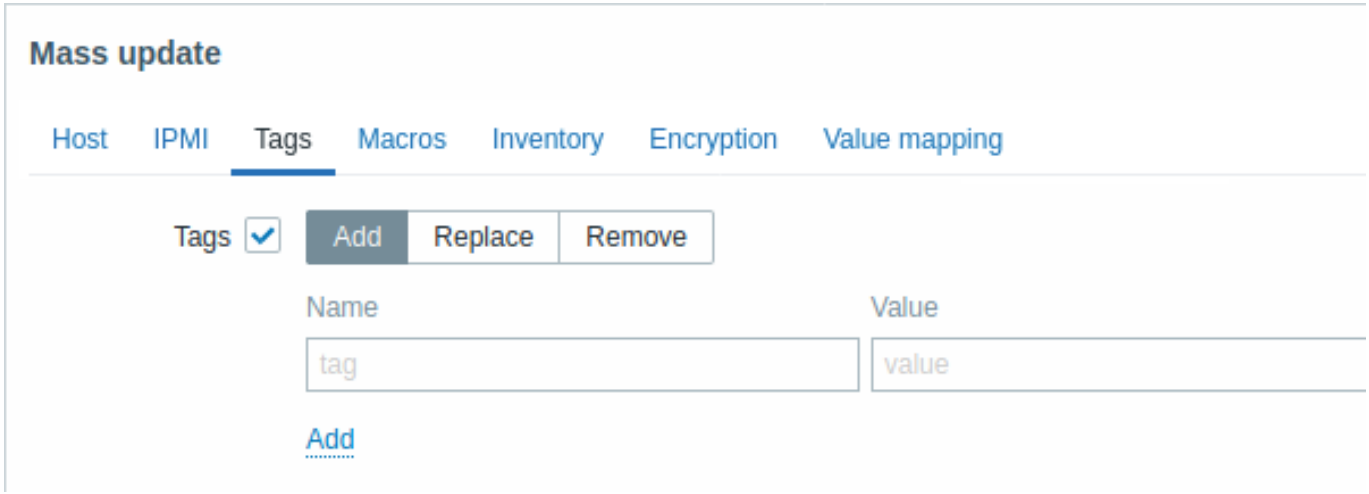
L'option *Effacer lors de la dissociation* permettra non seulement de dissocier les modèles précédemment liés, mais aussi de supprimer tous les éléments hérités de ceux-ci (items, déclencheurs, etc.).

Les options suivantes sont disponibles lors de la sélection du bouton correspondant pour la mise à jour des **groupes d'hôtes** :

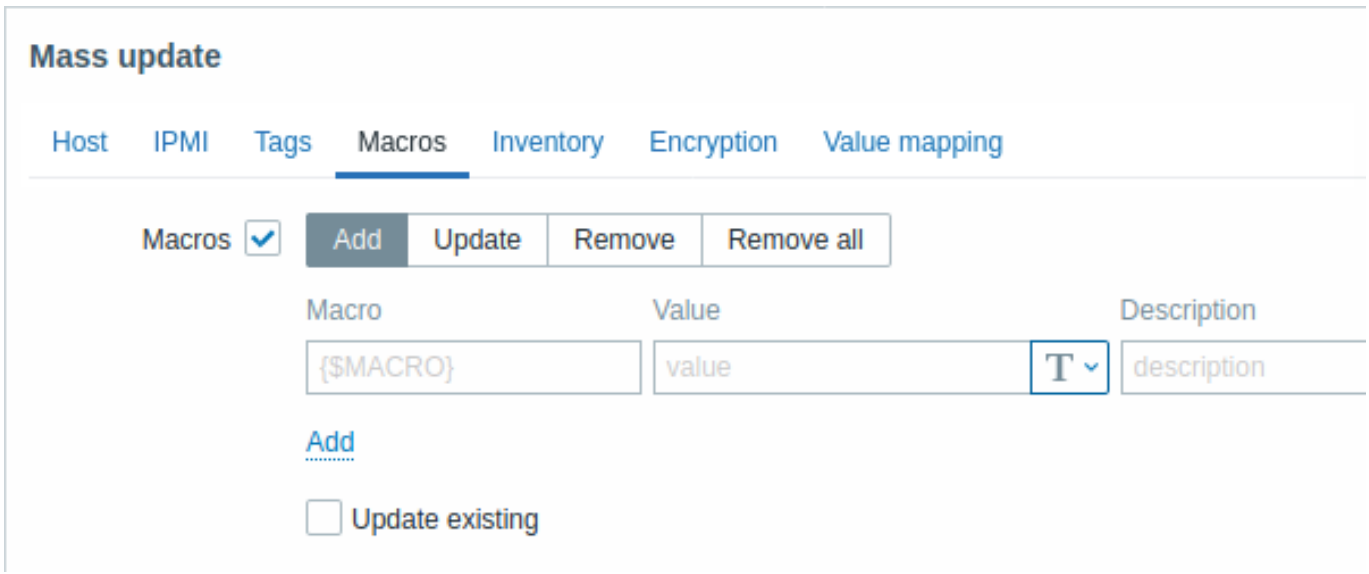
- *Ajouter* - permet de spécifier des groupes d'hôtes supplémentaires parmi les groupes existants ou de saisir des groupes d'hôtes entièrement nouveaux pour les hôtes
- *Remplacer* - supprimera l'hôte de tous les groupes d'hôtes existants et les remplacera par celui/ceux spécifié(s) dans ce champ (groupes d'hôtes existants ou nouveaux)
- *Supprimer* - supprimera des groupes d'hôtes spécifiques des hôtes

Ces champs utilisent l'auto-complétion : lorsque vous commencez à y saisir du texte, une liste déroulante des groupes d'hôtes correspondants est proposée. Si le groupe d'hôtes est nouveau, il apparaît également dans la liste déroulante et est indiqué par *(new)* après la chaîne. Faites simplement défiler la liste pour le sélectionner.



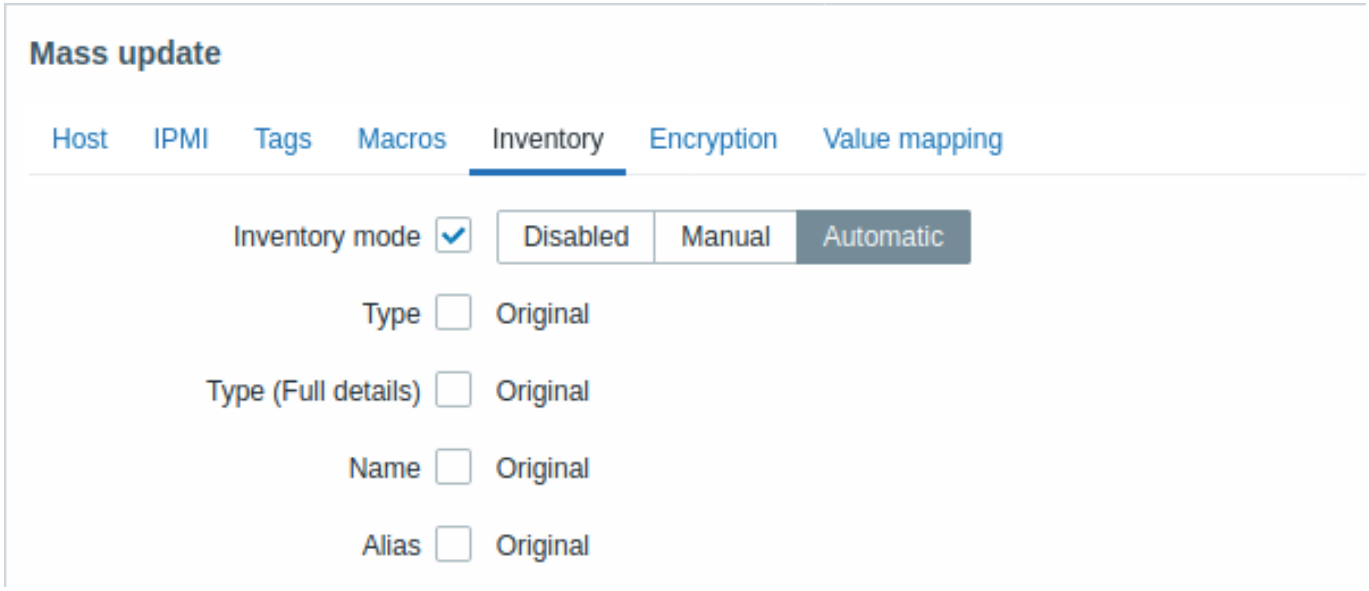


Les macros utilisateur, les macros {INVENTORY.*}, les macros {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} et {HOST.ID} sont prises en charge dans les tags. Notez que les tags portant le même nom mais des valeurs différentes ne sont pas considérés comme des « doublons » et peuvent être ajoutés au même hôte.

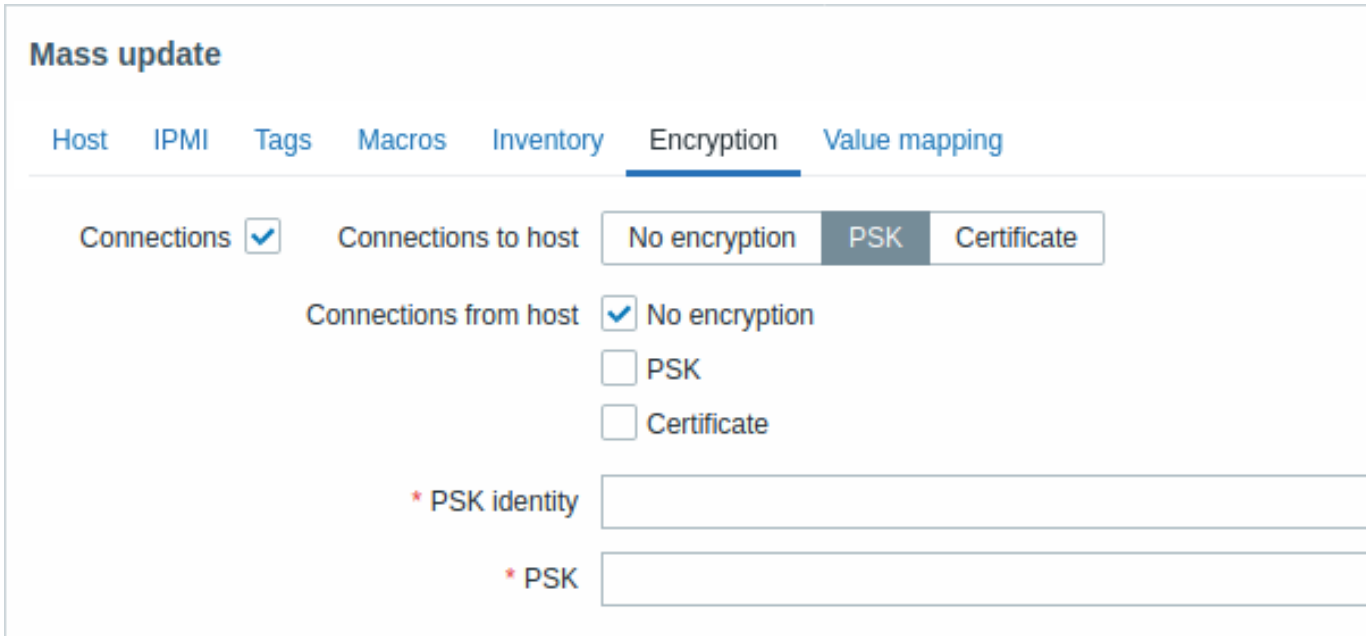


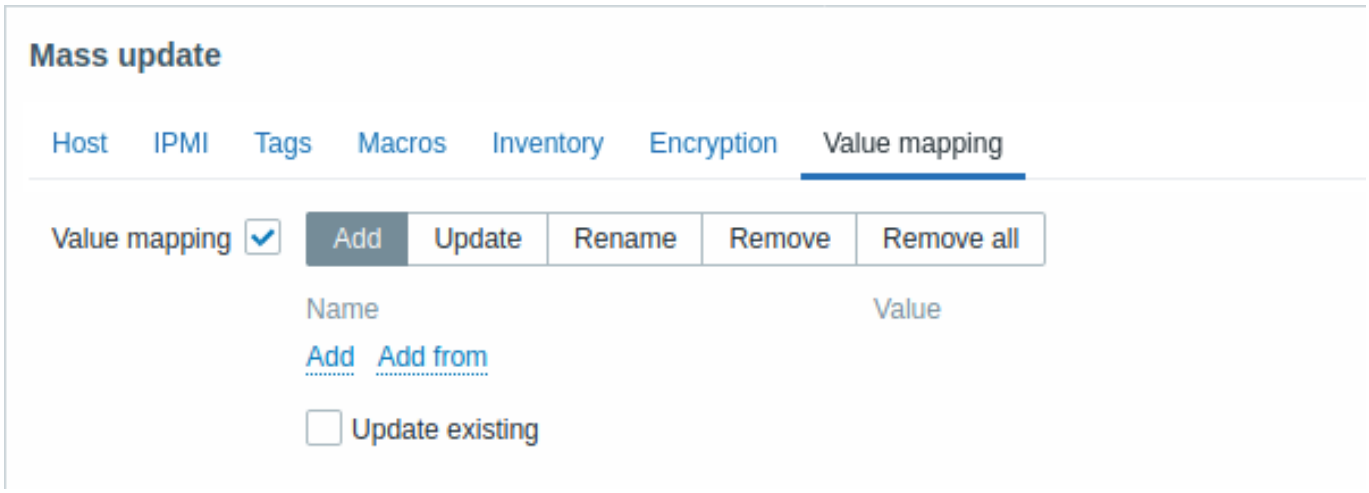
Les options suivantes sont disponibles lors de la sélection du bouton correspondant pour la mise à jour des macros :

- *Ajouter* - permet de spécifier des macros utilisateur supplémentaires pour les hôtes. Si la case *Mettre à jour l'existant* 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(s) hôte(s), elle ne sera pas mise à jour.
- *Mettre à jour* - remplacera les valeurs, types et descriptions des macros spécifiées dans cette liste. Si la case *Ajouter les manquants* est cochée, une 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* - supprimera les macros spécifiées des hôtes. Si la case *Sauf sélection* est cochée, toutes les macros sauf celles spécifiées dans la liste seront supprimées. Si elle n'est pas cochée, seules les macros spécifiées dans la liste seront supprimées.
- *Tout supprimer* - supprimera 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 en masse les champs d'inventaire, le *Mode d'inventaire* doit être défini sur « Manuel » ou « Automatique ».





Des boutons avec les options suivantes sont disponibles pour la mise à jour des correspondances de valeurs :

- *Ajouter* - ajouter des correspondances de valeurs aux hôtes. Si vous cochez *Mettre à jour l'existant*, toutes les propriétés de la correspondance de valeurs portant ce nom seront mises à jour. Sinon, si une correspondance de valeurs portant ce nom existe déjà, elle ne sera pas mise à jour.
- *Mettre à jour* - mettre à jour les correspondances de valeurs existantes. Si vous cochez *Ajouter les manquants*, une corre-

spondance de valeurs qui n'existait pas auparavant sur un hôte sera ajoutée comme nouvelle correspondance de valeurs. Sinon, seules les correspondances de valeurs qui existent déjà sur un hôte seront mises à jour.

- *Renommer* - donner un nouveau nom à une correspondance de valeurs existante
- *Supprimer* - supprimer les correspondances de valeurs spécifiées des hôtes. Si vous cochez *Sauf sélection*, toutes les correspondances de valeurs seront supprimées **sauf** celles qui sont spécifiées.
- *Tout supprimer* - supprimer toutes les correspondances de valeurs des hôtes. Si la case *Je confirme la suppression de toutes les correspondances 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.

2 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.

1 Création d'un élément

Aperçu

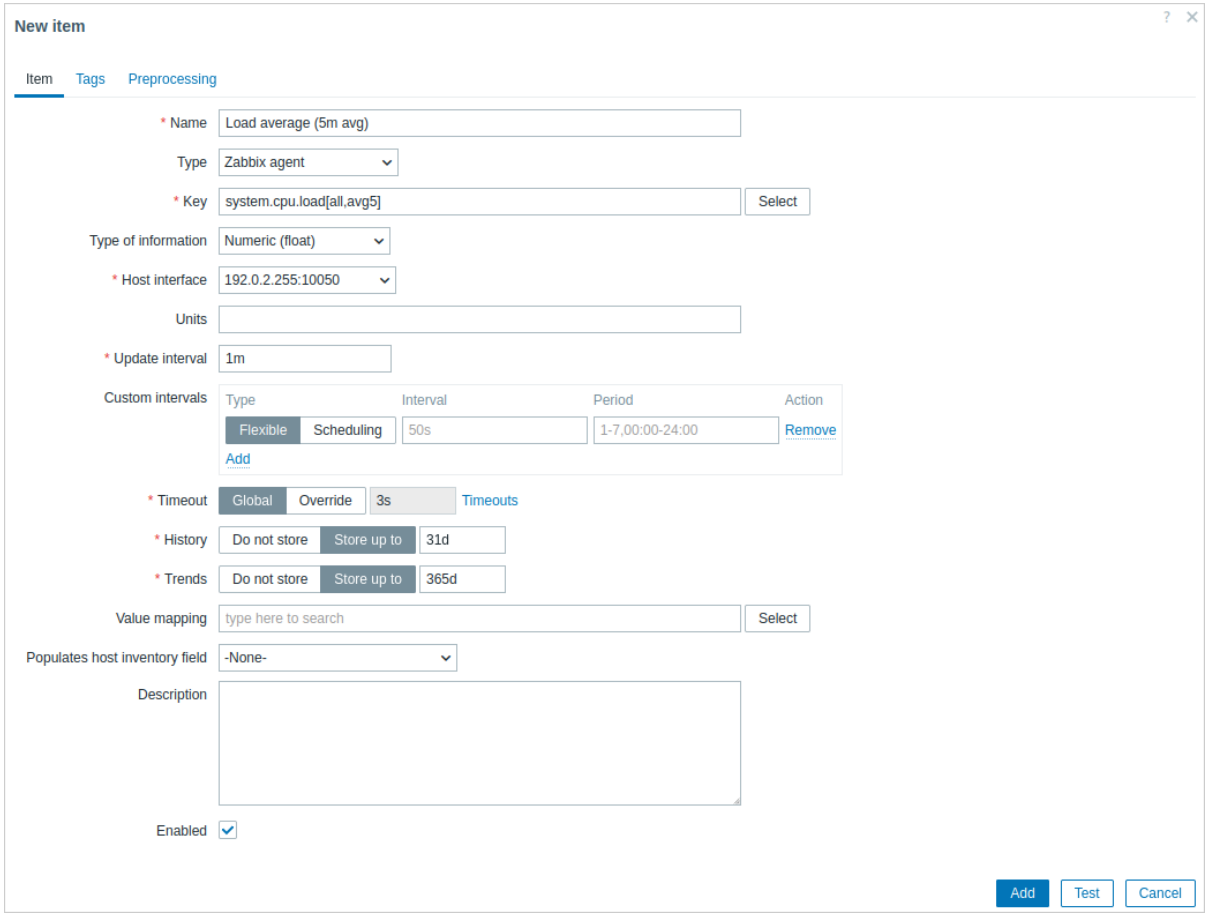
Pour créer un élément dans l'interface web de Zabbix, procédez comme suit :

- Allez dans : *Collecte de données* > *Hôtes*
- Cliquez sur *Éléments* sur la ligne de l'hôte
- Cliquez sur *Créer un élément* dans le coin supérieur droit de l'écran
- Saisissez les paramètres de l'élément dans le formulaire

Vous pouvez également créer un élément en ouvrant un élément existant, en appuyant 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 de saisie obligatoires sont marqués d'un astérisque rouge.

Paramètre	Description
Nom	Nom de l'élément. Les macros utilisateur sont prises en charge.
Type	Type d'élément. Voir les sections individuelles des types d'élément.
Clé	Clé de l'élément (jusqu'à 2048 caractères). Les clés d'élément prises en charge se trouvent dans les sections de 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 Zabbix server. Voir aussi : le format de clé correct.
Type d'information	Le type de données utilisé pour valider la valeur de l'élément et la stocker dans la base de données, après d'éventuelles conversions : **Numérique (non signé)** - entier non signé 64 bits ; **Numérique (flottant)** - nombre à virgule flottante 64 bits ; **Caractère** - texte court ; **Journal** - texte long avec ou sans propriétés de journal (horodatage, source, gravité, ID d'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 déclencheur. Voir les limites des données d'élément pour les limites de stockage par type de données et moteur de base de données. Pour les éléments qui renvoient des données dans un seul format spécifique, 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 64 bits, donc *Numérique (flottant)* est sélectionné automatiquement.
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.

Paramètre	Description
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 spécifié. Symboles d'unité pris en charge avec formatage spécial (et exemples de valeur d'élément reçue → valeur affichée) : **B** - octets (1024 → 1 KB) **Bps** - octets par seconde (1024 → 1 KBps) **s** - secondes, affichées en utilisant jusqu'à trois plus grandes unités de temps non nulles (881764 → 10d 4h 56m) **uptime** - temps écoulé au format hh:mm:ss ou N jours, hh:mm:ss (881764 → 10 jours, 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 *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 un préfixe correspondant (5000 → 5 KHz, 881764 → 881.76 KW). Si le symbole d'unité est préfixé par **!**, 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'unités, voir Suffixes de valeur d'élément.
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 temporels sont pris en charge, par ex. 30s, 1m, 2h, 1d. Les macros utilisateur sont prises en charge. Une seule macro doit remplir l'ensemble du champ. Plusieurs macros dans un champ ou des macros mélangées avec du texte ne sont pas prises en charge. *Remarque* : l'intervalle de mise à jour ne peut être défini à « 0 » que si des intervalles personnalisés existent avec une valeur non nulle. S'il est défini à « 0 » et qu'un intervalle personnalisé (flexible ou planifié) existe avec une valeur non nulle, l'élément sera interrogé pendant la durée de l'intervalle personnalisé. *Notez* que la première interrogation de l'élément après son activation ou après une 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 ont un intervalle de mise à jour planifié ou flexible et que *l'intervalle de mise à jour* est défini à 0. Un élément passif existant peut être interrogé immédiatement pour obtenir une valeur en appuyant sur le bouton *Exécuter maintenant*.
Intervalles personnalisés	Vous pouvez créer des règles personnalisées pour vérifier l'élément : **Flexible** - créer une exception à *l'intervalle de mise à jour* (intervalle avec une fréquence différente). **Planification** - créer un calendrier d'interrogation personnalisé. Pour des informations détaillées, voir Intervalles personnalisés. Les suffixes temporels sont pris en charge dans le champ *Intervalle*, par ex. 30s, 1m, 2h, 1d. Les macros utilisateur sont prises en charge. Une seule macro doit remplir l'ensemble du champ. Plusieurs macros dans un champ ou des macros mélangées avec du texte ne sont pas prises en charge.
Délai d'expiration	Définissez le délai d'expiration de la vérification de l'élément (disponible pour les types d'élément pris en charge). 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). **Remplacer** - un délai d'expiration personnalisé est utilisé (défini dans le champ *Délai d'expiration* ; plage autorisée : 1 - 600s). Les suffixes temporels, par ex. 30s, 1m, et les macros utilisateur sont pris en charge. Cliquer sur le lien *Timeouts* vous permet de configurer les délais d'expiration du proxy ou les délais d'expiration globaux (si un proxy n'est pas utilisé). Notez que le lien *Timeouts* n'est visible que pour les utilisateurs de type *Super admin* disposant des autorisations pour les sections de l'interface *Administration* > *General* ou *Administration* > *Proxies*.

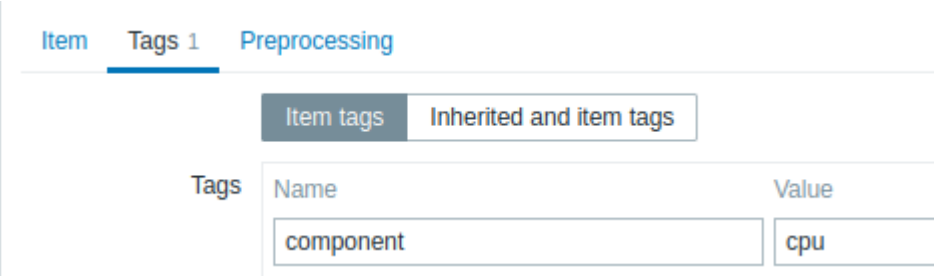
Paramètre	Description
Historique	Sélectionnez l'une des options suivantes : **Ne pas stocker** - l'historique de l'élément n'est pas stocké. Utile pour les éléments maîtres si seuls les éléments dépendants doivent conserver un historique. Ce paramètre ne peut pas être remplacé par les paramètres globaux du housekeeper. **Conserver jusqu'à** - spécifiez 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. Stocké en secondes. Les suffixes temporels sont pris en charge, par ex. 2h, 1d. Les macros utilisateur sont prises en charge. La valeur *Conserver jusqu'à* peut être remplacée globalement dans *Administration > Housekeeping*. Si un paramètre global de remplacement existe, une icône d'information orange  s'affiche. Si vous placez le pointeur de la souris dessus, un message d'avertissement s'affiche, par ex. *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 long historique de valeurs, vous pouvez conserver plus longtemps les données de tendances. Voir aussi 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. **Conserver jusqu'à** - spécifiez la durée de conservation de l'historique agrégé (minimum, maximum, moyenne, nombre par heure) dans la base de données (de 1 jour à 25 ans). Les données plus anciennes seront supprimées par le housekeeper. Stocké en secondes. Les suffixes temporels sont pris en charge, par ex. 24h, 1d. Les macros utilisateur sont prises en charge. La valeur *Conserver jusqu'à* peut être remplacée globalement dans *Administration > Housekeeping*. Si un paramètre global de remplacement existe, une icône d'information orange  s'affiche. Si vous placez le pointeur de la souris dessus, un message d'avertissement s'affiche, par ex. *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 aussi Historique et tendances.
Correspondance de valeurs	Appliquez une correspondance de valeurs à cet élément. La correspondance de valeurs ne modifie pas les valeurs reçues, elle sert uniquement à l'affichage des données. Elle 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 laissé vide, l'horodatage sera défini à 0 en temps Unix, représentant le 1er janvier 1970. Par exemple, considérez 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 n'importe quels caractères sauf "yMdhms".
Renseigne le champ d'inventaire de l'hôte	Vous pouvez sélectionner un champ d'inventaire de l'hôte que la valeur de l'élément renseignera. Cela fonctionnera 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 d'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 n'est disponible que lors de la modification d'un élément déjà existant.

Note:
Les champs spécifiques au type d'élément sont décrits sur les pages correspondantes.

Note:
Lors de la modification, au niveau de l'hôte, d'un élément existant au niveau du modèle, un certain nombre de 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 **Tags** permet de définir des tags au niveau de l'élément.



Prétraitement des valeurs d'élément

L'onglet **Preprocessing** permet de définir des règles de transformation pour les valeurs reçues.

Test d'un élément

Attention:
Pour effectuer le test d'un élément, assurez-vous que l'heure système du serveur et du proxy est synchronisée. Si l'heure du serveur est en retard, le test d'un élément peut renvoyer le message d'erreur « The task has been expired. » En revanche, le fait de définir des fuseaux horaires différents sur le serveur et le proxy n'affectera pas le résultat du test.

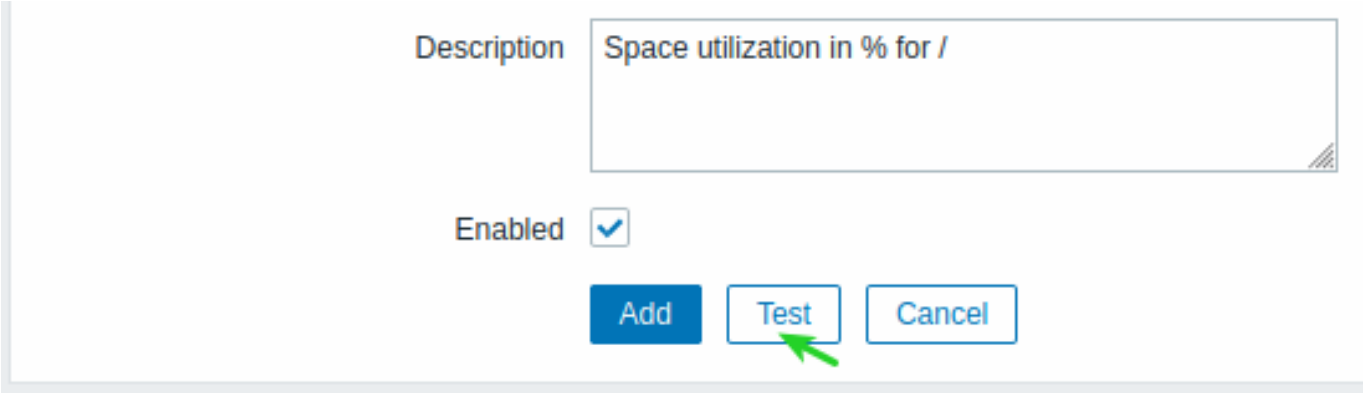
Il est possible de tester un élément et, s'il est correctement configuré, d'obtenir en retour une valeur réelle. Le test peut être effectué même avant l'enregistrement de l'élément.

Le test est disponible pour les éléments d'hôte et de modèle, les prototypes d'élément et les règles de découverte de bas niveau. Le test n'est pas disponible pour les éléments actifs.

Le test d'élément est disponible pour les types d'éléments passifs suivants :

- Agent Zabbix
- Agent SNMP (v1, v2, v3)
- Agent IPMI
- Vérifications SSH
- Vérifications Telnet
- Agent JMX
- Vérifications simples (sauf les éléments `icmpping*`, `vmware.*`)
- Interne Zabbix
- Éléments calculés
- Vérifications externes
- Surveillance de base de données
- Agent HTTP
- Script
- Navigateur

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).

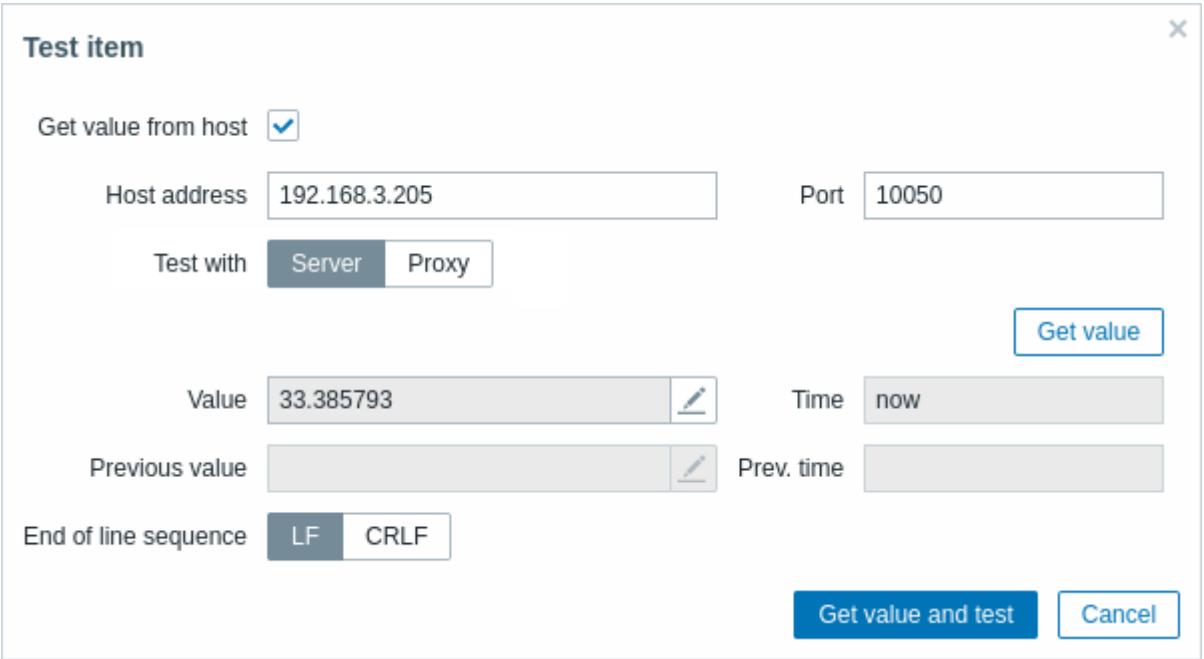


Le formulaire de test d'élément comporte des champs pour les paramètres d'hôte requis (adresse de l'hôte, port, test avec serveur/proxy (nom du proxy)) et des détails spécifiques à l'élément (tels que la communauté SNMPv2 ou les informations d'identification de sécurité SNMPv3). Ces champs sont contextuels :

- Les valeurs sont préremplies lorsque cela est possible, c'est-à-dire, pour les éléments nécessitant un agent, en reprenant les informations de l'interface agent sélectionnée de l'hôte.
- Les valeurs doivent être renseignées manuellement pour les éléments de modèle.
- Les valeurs de macros en texte brut sont résolues.
- Les champs dont la valeur (ou une partie de la valeur) est une macro secrète ou 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)

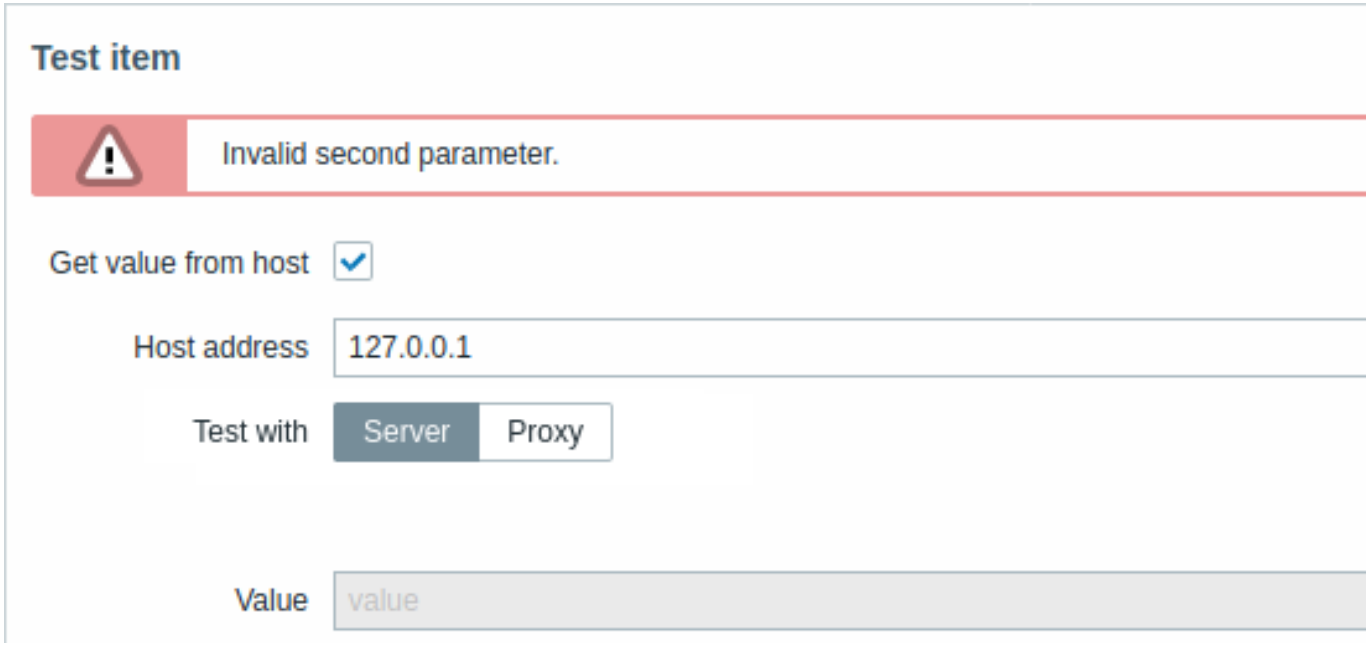
Pour tester l'élément, cliquez sur *Get value*. Si la valeur est récupérée avec succès, elle remplira le champ *Value*, en déplaçant la valeur actuelle (le cas échéant) vers le champ *Previous value* tout en calculant également le champ *Prev. time*, c'est-à-dire la différence de temps entre les deux valeurs (clics), et en essayant de détecter une séquence EOL et de basculer vers 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 512KB lors de l'envoi au frontend. Si un résultat de test est tronqué, une icône d'avertissement s'affiche ; survolez-la pour obtenir des 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 512KB sont tout de même entièrement traitées par le serveur Zabbix.

Si la configuration est incorrecte, un message d'erreur s'affiche en décrivant la cause possible.



Une valeur récupérée avec succès depuis un hôte peut également être utilisée pour tester les étapes de prétraitement.

Boutons du formulaire

Les boutons en bas du formulaire permettent d'effectuer plusieurs opérations.

Add	Ajouter un élément. Ce bouton n'est disponible que 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 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 moteur 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 (à virgule flottante) 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 texte sont tronquées avant d'être stockées afin de correspondre à la limite du type de valeur de la base de données :

Base de données	Type d'information		
	Caractère	**Journal**	**Texte**

Base de données	Type d'information		
MySQL	255 caractères	65536 octets	65536 octets
PostgreSQL	255 caractères	65536 caractères	65536 caractères
SQLite (proxy Zabbix uniquement)	255 caractères	65536 caractères	65536 caractères

Les valeurs binaires sont limitées à 16MiB (16777216 octets). Les valeurs qui dépassent cette limite sont ignorées, et une erreur correspondante est affichée dans l'interface web.

Les valeurs avec le type de données JSON sont limitées à 128MiB (134217728 octets). Les valeurs qui dépassent cette limite sont ignorées, et une erreur correspondante est affichée dans l'interface web. Pour MySQL/MariaDB, envisagez de configurer des partitions quotidiennes pour la table `history_json` (utilisée pour stocker les valeurs JSON), car elle peut rapidement devenir volumineuse et verrouiller le housekeeper pendant une période prolongée. Si votre cas d'utilisation implique de grandes valeurs JSON (1MiB ou plus), consultez comment configurer 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 longues entrées de journal sont tronquées à 64KB.

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 indique combien de temps Zabbix doit attendre avant d'abandonner la vérification comme ayant échoué.

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 correctement collectées pour un seul de plusieurs OID dans une vérification SNMP).

Pour de nombreux types d'éléments, vous pouvez définir des délais d'expiration d'élément **flexibles** :

- par élément individuel
- par type d'élément (au niveau d'un proxy)
- par type d'élément (au niveau global)

Un délai d'expiration personnalisé *par élément individuel* est utile si vous souhaitez spécifier un délai plus long pour un élément particulier, tout en conservant des délais faibles pour les autres éléments.

Priorité des délais d'expiration

1. Le délai d'expiration d'un élément individuel remplace tous les autres délais d'expiration.
2. Les délais d'expiration au niveau du proxy remplacent les délais d'expiration globaux.

Le délai d'expiration défini dans la configuration du server ou du proxy Zabbix ne joue **aucun** rôle pour 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 de données sont pris en charge pour les types d'éléments suivants :

- Agent Zabbix (vérifications passives et actives)
- Agent SNMP (à l'exception des vérifications SNMP héritées[1])
- Vérification simple (à l'exception de `icmpping*` et des éléments VMware[2])
- Agent SSH
- Agent Telnet
- Vérification externe
- Supervision 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 supervision VMware utilisent leur propre paramètre VMwareTimeout 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).

1 Format de clé d'élément

Le format de la clé d'élément, y compris les paramètres clés, doit suivre les règles de syntaxe. Les illustrations suivantes illustrent 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 n'est pas autorisé.

Pour construire une clé d'élément valide, on commence par spécifier le nom de la clé, puis on a le choix d'avoir des paramètres ou non - comme l'illustrent les deux lignes qui peuvent être suivies.

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 :

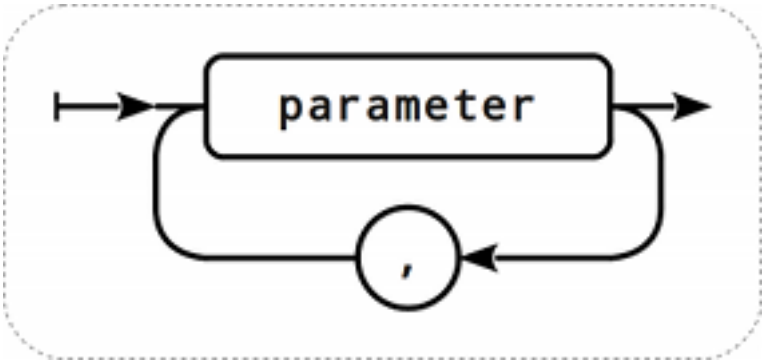
`0-9a-zA-Z_-.`

Ce qui signifie :

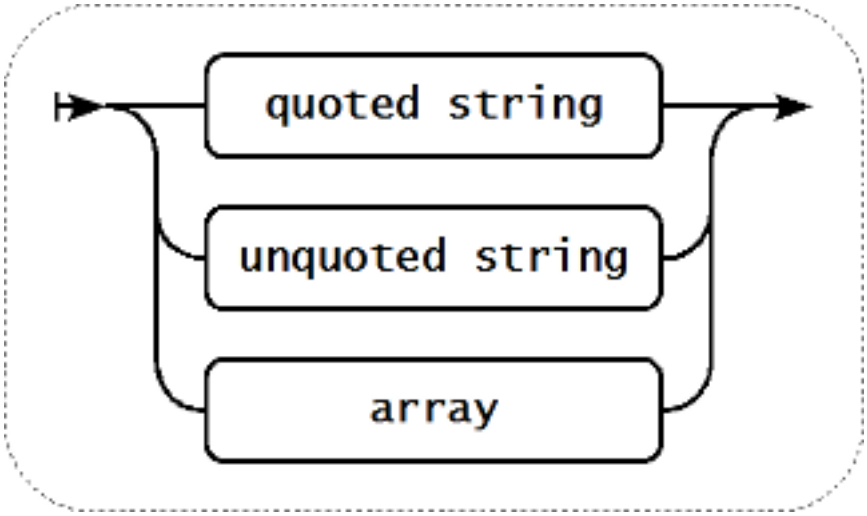
- tous les nombres ;
- toutes les lettres minuscules ;
- toutes les lettres majuscules ;
- underscore ;
- tiret ;
- point.

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.

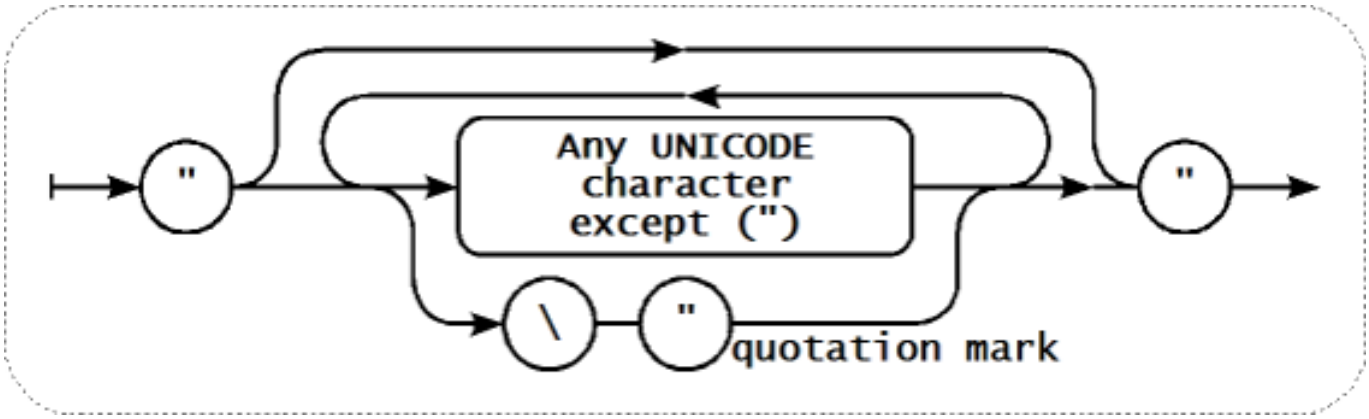


Le paramètre peut également être laissé vide, utilisant ainsi la valeur par défaut. Dans ce cas, le nombre approprié de virgules doit être ajouté si d'autres paramètres sont spécifiés. Par exemple, la clé d'élément **icmpping[„200„500]** indiquerait que l'intervalle entre les pings individuels est de 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.

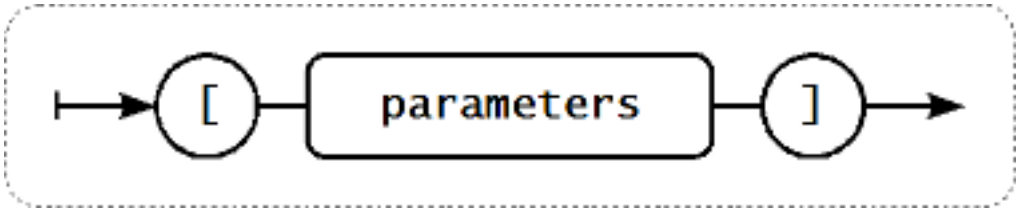
Attention:
Les tableaux de paramètres à plusieurs niveaux, par ex. `[a,[b,[c,d]],e]`, ne sont pas autorisés.

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.



2 Intervalles personnalisés

Aperçu

Il est possible de créer des règles personnalisées concernant les moments où un élément est vérifié. Les deux méthodes pour cela sont les *intervalles flexibles*, qui permettent de redéfinir l'intervalle de mise à jour par défaut, et la *planification*, grâce à laquelle la vérification d'un élément peut être exécutée à une heure précise ou selon une séquence d'heures.

Intervalles flexibles

Les intervalles flexibles permettent de redéfinir l'intervalle de mise à jour par défaut pour des périodes spécifiques. Un intervalle flexible est défini par *Interval* et *Period*, où :

- *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 (voir les périodes de temps pour une description détaillée du format de *Period*)

Si plusieurs intervalles flexibles se chevauchent, la plus petite valeur de *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 l'interrogation selon l'*Update interval* 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 ouvrées.
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 cela était utilisé comme solution de contournement pour les vérifications planifiées et qu'il est recommandé d'utiliser des intervalles de planification pour ce type de 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 (incluses). 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 de progression 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 soit avoir une plage, soit avoir 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, seule l'heure « 0 » correspondra, à condition que les filtres des minutes et des secondes soient également omis ; sinon, un filtre d'heures vide correspondra à toutes les valeurs d'heure.

Les valeurs valides de `<from>` et `<to>` pour leur préfixe de filtre respectif 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 le frontend Zabbix, plusieurs intervalles de planification sont saisis sur des lignes séparées. 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 le frontend*)	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"
```

2 Prétraitement de la valeur de l'élément

Aperçu

Le prétraitement vous permet d'appliquer des transformations aux valeurs d'élément reçues avant leur enregistrement dans la base de données. Ces transformations/étapes de prétraitement sont effectuées par le serveur Zabbix ou le proxy (si les éléments sont surveillés par un proxy).

Cette fonctionnalité prend en charge divers cas d'utilisation, tels que :

- convertir des octets en bits (par exemple, multiplier les valeurs du 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 les 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 *Custom on fail* (disponible pour la plupart des transformations), qui 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, le rendant 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.

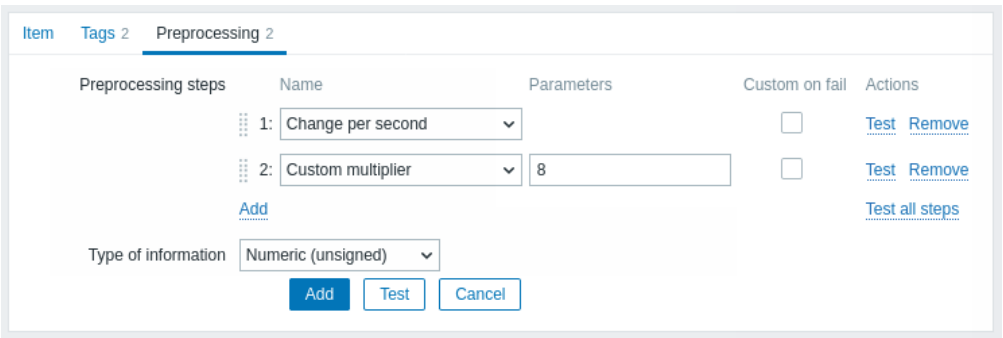
Toutes les valeurs transmises au prétraitement sont initialement traitées comme des chaînes de caractères. La conversion vers le type de valeur souhaité (tel que défini dans la configuration de l'élément) est effectuée à la fin du pipeline de prétraitement. Cependant, certaines étapes de prétraitement 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*. Voir Création d'un élément pour une description détaillée des paramètres.

Transformations prises en charge

Toutes les transformations prises en charge sont listées ci-dessous. Cliquez sur le nom de la transformation pour en voir tous les détails.

Nom	Description	Type
Expression régulière	Faire correspondre la valeur à l'expression régulière et la remplacer par la sortie requise.	Texte
Remplacer	Rechercher la chaîne à trouver et la remplacer par une autre (ou par rien).	
Supprimer les caractères en début et fin	Supprimer les caractères spécifiés au début et à la fin de la valeur.	
Supprimer les caractères à droite	Supprimer les caractères spécifiés à la fin de la valeur.	
Supprimer les caractères à gauche	Supprimer les caractères spécifiés au début de la valeur.	
XML XPath	Extraire une valeur ou un fragment à partir de données XML à l'aide de la fonctionnalité XPath.	Données structurées
JSON Path	Extraire une valeur ou un fragment à partir 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.	
Valeur SNMP walk	Extraire une valeur à l'aide du nom OID/MIB spécifié et appliquer des options de formatage.	SNMP
SNMP walk vers JSON	Convertir les valeurs SNMP en JSON.	
Valeur SNMP get	Appliquer des options de formatage à la valeur SNMP get.	
Multiplicateur personnalisé	Multiplier la valeur par la valeur entière ou à virgule flottante spécifiée.	Arithmétique
Changement simple	Calculer la différence entre la valeur actuelle et la valeur précédente.	Changement
Changement par seconde	Calculer la vitesse de changement par seconde de la valeur (différence entre la valeur actuelle et la valeur précédente).	
Booléen vers décimal	Convertir la valeur du format booléen au format décimal.	Systèmes de numération
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 un intervalle dans lequel une valeur doit se trouver.	Validation
Correspond à l'expression régulière	Spécifier une expression régulière à laquelle une valeur doit correspondre.	
Ne correspond pas à l'expression régulière	Spécifier une expression régulière à laquelle une valeur ne doit pas correspondre.	
Vérifier une erreur dans JSON	Vérifier un message d'erreur au niveau de l'application situé à JSONPath.	
Vérifier une erreur dans XML	Vérifier un message d'erreur au niveau de l'application situé à XPath.	
Vérifier une erreur à l'aide d'une expression régulière	Vérifier un message d'erreur au niveau de l'application à l'aide d'une expression régulière.	
Vérifier une valeur non prise en charge	Vérifier si aucune valeur d'élément n'a pu être récupérée.	
Ignorer si inchangé	Ignorer une valeur si elle n'a pas changé.	Limitation
Ignorer si inchangé avec heartbeat	Ignorer une valeur si elle n'a pas changé pendant la période définie.	
Motif Prometheus	Utiliser la requête suivante pour extraire les données requises à partir des métriques Prometheus.	Prometheus
Prometheus vers JSON	Convertir les métriques Prometheus requises en JSON.	

Notez que pour les étapes de prétraitement *Changement* et *Limitation*, Zabbix doit mémoriser la dernière valeur afin de calculer/comparer la nouvelle valeur selon les besoins. Ces valeurs précédentes sont gérées par le gestionnaire de prétraitement. Si le serveur ou le proxy Zabbix est redémarré, ou si une modification est apportée aux étapes de prétraitement, la dernière valeur de l'élément correspondant est réinitialisée, ce qui entraîne :

- pour les étapes *Changement simple* et *Changement par seconde* - la valeur suivante sera ignorée, car il n'existe aucune valeur précédente à partir de laquelle calculer le changement ;
- pour les étapes *Ignorer si inchangé* et *Ignorer si inchangé avec heartbeat* - la valeur suivante ne sera jamais ignorée, même si elle aurait dû l'être en raison des 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 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.

Commentaires :

- Si la valeur d'entrée ne correspond pas, l'élément deviendra non pris en charge ;
- L'expression régulière prend en charge l'extraction d'un maximum de 10 groupes capturés avec la séquence \N ;
- Si vous cochez la case *Custom on fail*, il est possible de spécifier des options personnalisées de gestion des erreurs : soit ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié. En cas d'échec d'une étape de prétraitement, l'élément ne deviendra pas non pris en charge si l'option permettant d'ignorer la valeur ou de définir une valeur spécifiée est sélectionnée.
- Veuillez consulter la section expressions régulières pour voir quelques exemples existants.

Remplacer

Recherchez la chaîne de recherche et remplacez-la par une autre (ou par rien).

Paramètres :

- **chaîne de recherche** - la chaîne à rechercher et à remplacer, sensible à la casse (obligatoire) ;
- **remplacement** - la chaîne qui remplacera la chaîne de recherche. La chaîne de remplacement peut également être vide, ce qui permet effectivement 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, retours chariot, tabulations et espaces "\n \r \t \s" ; l'antislash peut être échappé sous la forme "\\" et les séquences d'échappement peuvent être échappées sous la forme "\\n" ;
- L'échappement des sauts de ligne, retours chariot et 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 à partir 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;
- Si vous cochez la case *Personnalisé en cas d'échec*, il est possible de spécifier des options personnalisées de gestion des erreurs : soit ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié. En cas d'échec d'une étape de prétraitement, l'élément ne deviendra pas non pris en charge si l'option consistant à ignorer la valeur ou à définir une valeur spécifiée est sélectionnée.

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></doc
```

Chemin JSON

Extraire une valeur ou un fragment à partir de données JSON à l'aide de la fonctionnalité JSONPath.

Si vous cochez la case *Personnalisé en cas d'échec*, il est possible de spécifier des options personnalisées de gestion des erreurs : soit ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié. En cas d'échec d'une étape de prétraitement, l'élément ne deviendra pas non pris en charge si l'option permettant d'ignorer la valeur ou de définir une valeur spécifiée est sélectionnée.

CSV vers JSON

Convertissez les données d'un fichier CSV au format JSON.

Pour plus d'informations, voir : prétraitement CSV vers JSON.

XML en JSON

Convertissez les données au format XML en JSON.

Pour plus d'informations, voir : Règles de sérialisation.

Si vous cochez la case *Custom on fail*, il est possible de spécifier des options personnalisées de gestion des erreurs : soit ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié. En cas d'échec d'une étape de prétraitement, l'élément ne deviendra pas non pris en charge si l'option permettant d'ignorer la valeur ou de définir une valeur spécifiée est sélectionnée.

Valeur SNMP walk

Extraire la valeur par le nom OID/MIB spécifié et appliquer les options de formatage :

- **Inchangée** - renvoyer la chaîne hexadécimale comme une chaîne hexadécimale non échappée (*notez* que les indications d'affichage sont toujours appliquées) ;
- **UTF-8 à partir de hex-STRING** - convertir la chaîne hexadécimale en chaîne UTF-8 ;
- **MAC à partir de hex-STRING** - valider la chaîne hexadécimale en tant qu'adresse MAC et renvoyer une chaîne d'adresse MAC correcte (où les `' '` sont remplacés par `':'`) ;
- **Entier à partir de BITS** - convertir les 8 premiers octets d'une chaîne de bits exprimée comme une 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.

Si vous cochez la case *Personnalisé en cas d'échec*, il est possible de spécifier des options personnalisées de gestion des erreurs : soit ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié. En cas d'échec d'une étape de prétraitement, l'élément ne deviendra pas non pris en charge si l'option d'ignorer la valeur ou de définir une valeur spécifiée est sélectionnée.

Conversion SNMP walk en JSON

Convertissez 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 à partir des 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;
- Si vous cochez la case *Personnalisé en cas d'échec*, il est possible de spécifier des options personnalisées de gestion des erreurs : soit ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié. En cas d'échec d'une étape de prétraitement, l'élément ne deviendra pas non pris en charge si l'option permettant d'ignorer la valeur ou de définir une valeur spécifiée est sélectionnée.

Valeur obtenue par SNMP

Appliquez des options de formatage à la valeur obtenue par SNMP :

- **UTF-8 à partir de Hex-STRING** - convertit une chaîne hexadécimale en chaîne UTF-8 ;
- **MAC à partir de Hex-STRING** - valide une chaîne hexadécimale en tant qu'adresse MAC et renvoie une chaîne d'adresse MAC correcte (où les `' '` sont remplacés par `':'`) ;
- **Entier à partir de BITS** - convertit les 8 premiers octets d'une chaîne de bits exprimée comme une 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.

Si vous cochez la case *Personnalisé en cas d'échec*, il est possible de spécifier des options personnalisées de gestion des erreurs : soit ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié. En cas d'échec d'une étape de prétraitement, l'élément ne deviendra pas non pris en charge si l'option permettant d'ignorer la valeur ou de définir une valeur spécifiée est sélectionnée.

Multiplicateur personnalisé

Multipliez la valeur par la valeur entière ou à virgule flottante spécifiée.

Commentaires :

- Utilisez cette option pour convertir les valeurs reçues en KB, MBps, etc., en B, Bps. 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 comme *Changement 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 en tronquant la partie décimale et en stockant la valeur comme un entier ;
- Pris en charge : notation scientifique, par exemple 1e+70 ; macros utilisateur et macros LLD ; chaînes incluant des macros, par exemple `{#MACRO}e+10`, `{$MACRO1}e+{$MACRO2}`. Les macros doivent être résolues en un entier ou un nombre à virgule flottante.
- Si vous cochez la case *Personnalisé en cas d'échec*, il est possible de spécifier des options personnalisées de gestion des erreurs : soit ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié. En cas d'échec d'une étape de prétraitement, l'élément ne deviendra pas non pris en charge si l'option d'ignorer la valeur ou de définir une valeur spécifiée est sélectionnée.

Changement simple

Calcule la différence entre la valeur actuelle et la valeur précédente.

Commentaires :

- Cette étape peut être utile pour mesurer une valeur en augmentation constante;
- Évalué comme **value**-**prev_value**, où *value* est la valeur actuelle ; *prev_value* est la valeur précédemment reçue;
- Une seule opération de changement par élément (« Changement simple » ou « Changement 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;
- Si vous cochez la case *Custom on fail*, il est possible de spécifier des options personnalisées de gestion des erreurs : soit ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié. En cas d'échec d'une étape de prétraitement, l'élément ne deviendra pas non pris en charge si l'option permettant d'ignorer la valeur ou de définir une valeur spécifiée est sélectionnée.

Variation par seconde

Calcule la vitesse de variation de la valeur (différence entre la valeur actuelle et la valeur précédente) par seconde.

Commentaires :

- Cette étape est utile pour calculer la vitesse par seconde d'une valeur en augmentation constante ;
- 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 entrantes sont des entiers. Cela est particulièrement pertinent pour les petits nombres où 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 peut être perdue, il est en fait recommandé d'utiliser *Numérique (non signé)* et ainsi de ne tronquer que la partie décimale.
- Évalué comme (**value**-**prev_value**)/(**time**-**prev_time**), où *value* est la valeur actuelle ; *prev_value* est la valeur reçue précédemment ; *time* est l'horodatage actuel ; *prev_time* est l'horodatage de la valeur précédente ;
- Une seule opération de variation par élément (« Changement 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 de fonctionner correctement, par exemple, avec un retour à zéro (débordement) des compteurs SNMP 32 bits.
- Si vous cochez la case *Personnalisé en cas d'échec*, il est possible de spécifier des options personnalisées de gestion des erreurs : soit ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié. En cas d'échec d'une étape de prétraitement, l'élément ne deviendra pas non pris en charge si l'option consistant à ignorer la valeur ou à définir une valeur spécifiée est sélectionnée.

Booléen en décimal

Convertit la valeur du format booléen en décimal.

Commentaires :

- La représentation textuelle est convertie en 0 ou 1. Ainsi, « TRUE » est stocké comme 1 et « FALSE » est stocké comme 0. Toutes les valeurs sont mises en correspondance 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 est considéré comme FALSE.

- Si vous cochez la case *Custom on fail*, il est possible de spécifier des options personnalisées de gestion des erreurs : soit ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié. En cas d'échec d'une étape de prétraitement, l'élément ne deviendra pas non pris en charge si l'option permettant d'ignorer la valeur ou de définir une valeur spécifiée est sélectionnée.

Octal en décimal

Convertit la valeur du format octal en décimal.

Si vous cochez la case *Personnalisé en cas d'échec*, il est possible de spécifier des options personnalisées de gestion des erreurs : soit ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié. En cas d'échec d'une étape de prétraitement, l'élément ne deviendra pas non pris en charge si l'option consistant à ignorer la valeur ou à définir une valeur spécifiée est sélectionnée.

Hexadécimal vers décimal

Convertit la valeur du format hexadécimal en décimal.

Si vous cochez la case *Personnalisé en cas d'échec*, il est possible de spécifier des options personnalisées de gestion des erreurs : soit ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié. En cas d'échec d'une étape de prétraitement, l'élément ne deviendra pas non pris en charge si l'option permettant d'ignorer la valeur ou de définir une valeur spécifiée est sélectionnée.

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 indiquant des valeurs minimale/maximale (incluses).

Commentaires :

- Les valeurs numériques sont acceptées (y compris n'importe quel nombre 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 ;
- Si vous cochez la case *Custom on fail*, il est possible de spécifier des options personnalisées de gestion des erreurs : soit ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié. En cas d'échec d'une étape de prétraitement, l'élément ne deviendra pas non pris en charge si l'option permettant d'ignorer la valeur ou de définir une valeur spécifiée est sélectionnée.

Correspond à l'expression régulière

Spécifiez une expression régulière à laquelle une valeur doit correspondre.

Si vous cochez la case *Personnalisé en cas d'échec*, il est possible de spécifier des options personnalisées de gestion des erreurs : soit ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié. En cas d'échec d'une étape de prétraitement, l'élément ne deviendra pas non pris en charge si l'option permettant d'ignorer la valeur ou de définir une valeur spécifiée est sélectionnée.

Ne correspond pas à l'expression régulière

Spécifiez une expression régulière à laquelle une valeur ne doit pas correspondre.

Si vous cochez la case *Personnalisé en cas d'échec*, il est possible de définir des options personnalisées de gestion des erreurs : soit ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié. En cas d'échec d'une étape de prétraitement, l'élément ne deviendra pas non pris en charge si l'option d'ignorer la valeur ou de définir une valeur spécifiée est sélectionnée.

Vérifier une erreur dans JSON

Vérifiez la présence d'un message d'erreur au niveau de l'application situé à l'emplacement JSONPath. Arrêtez le traitement en cas de succès et si le message n'est pas vide ; sinon, poursuivez 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 ajout d'informations sur l'étape de prétraitement ;
- Aucune erreur ne sera signalée en cas d'échec de l'analyse d'un JSON invalide ;
- Si vous cochez la case *Custom on fail*, il est possible de spécifier des options personnalisées de gestion des erreurs : soit ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié. En cas d'échec d'une étape de prétraitement, l'élément ne deviendra pas non pris en charge si l'option consistant à ignorer la valeur ou à définir une valeur spécifiée est sélectionnée.

Vérifier la présence d'une erreur dans le XML

Vérifiez la présence d'un message d'erreur au niveau de l'application situé à l'emplacement XPath. Arrêtez le traitement en cas de succès et si le message n'est pas vide ; sinon, poursuivez le traitement avec la valeur qui précédait cette étape de prétraitement.

Commentaires :

- Ces erreurs de service externes sont signalées à l'utilisateur telles quelles, sans ajout 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*, il est possible de spécifier des options personnalisées de gestion des erreurs : soit ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié. En cas d'échec d'une étape de prétraitement, l'élément ne deviendra pas non pris en charge si l'option permettant d'ignorer la valeur ou de définir une valeur spécifiée est sélectionnée.

Vérifier une erreur à l'aide d'une expression régulière

Vérifiez la présence d'un message d'erreur au niveau de l'application à l'aide d'une expression régulière. Arrêtez le traitement en cas de succès et si le message n'est pas vide ; sinon, poursuivez le traitement avec la valeur qui précédait cette étape de prétraitement.

Paramètres :

- **pattern** - l'expression régulière ;
- **output** - le modèle de formatage de 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.

Commentaires :

- Ces erreurs de service externe sont signalées à l'utilisateur telles quelles, sans ajout d'informations sur l'étape de prétraitement ;
- Si vous cochez la case *Custom on fail*, il est possible de spécifier des options personnalisées de gestion des erreurs : soit ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié. En cas d'échec d'une étape de prétraitement, l'élément ne deviendra pas non pris en charge si l'option permettant d'ignorer la valeur ou de définir une valeur spécifiée est sélectionnée.

Vérifier une valeur non prise en charge

Vérifiez si aucune valeur d'élément n'a pu être récupérée. Indiquez comment l'échec doit être traité, en vous basant sur l'inspection du message d'erreur renvoyé.

Paramètres :

- **scope** - sélectionnez la portée du traitement des erreurs :
toute erreur - toute erreur ;
l'erreur correspond - uniquement l'erreur qui correspond à l'expression régulière spécifiée dans *pattern* ;
l'erreur ne correspond pas - uniquement l'erreur qui ne correspond pas à l'expression régulière spécifiée dans *pattern*
- **pattern** - l'expression régulière à faire correspondre à l'erreur. Si *toute erreur* est sélectionné dans le paramètre scope, ce champ n'est pas affiché. S'il est affiché, ce champ est obligatoire.

Commentaires :

- Normalement, l'absence/l'échec de récupération d'une valeur entraînerait le passage de l'élément à l'état non pris en charge. Cette étape de prétraitement vous permet de modifier ce comportement. Si vous cochez la case *Custom on fail* (toujours cochée et grisée pour cette étape de prétraitement), il est possible de spécifier des options personnalisées de gestion des erreurs : soit ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié. En cas d'échec de l'étape de prétraitement, l'élément ne deviendra pas non pris en charge si l'option d'ignorer la valeur ou de définir une valeur spécifiée est sélectionnée.
- 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 ex. chaîne) correspond au type d'information de l'élément (par ex.

numérique) ; pour plus de détails, voir Exemples de prétraitement. En cas d'incompatibilité de type, l'élément peut tout de même devenir non pris en charge après l'exécution de toutes les étapes de prétraitement. Pour vérifier une incompatibilité de type, vous pouvez utiliser, par exemple, l'étape de prétraitement *Custom multiplier* ; voir Exemples de prétraitement.
- La capture de 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 de l'élément ;
- Plusieurs étapes *Check for not supported value* sont prises en charge, dans l'ordre spécifié. Une étape pour *toute erreur* sera automatiquement placée comme dernière étape de 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 pulsation

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 les secondes (minimum : 1 seconde) ;
- Des suffixes de temps peuvent être utilisés (par ex., 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 aucune connaissance de la réception de cette valeur. Aucune expression de déclencheur ne sera évaluée et, par conséquent, aucun problème pour les déclencheurs associés ne sera créé/résolu. Les fonctions ne fonctionneront qu'à partir des 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 tendance pour cette heure.
- Une seule option de limitation peut être spécifiée par élément.

Modèle Prometheus

Utilisez la requête suivante pour extraire les données requises des métriques Prometheus.

Voir les contrôles Prometheus pour plus de détails.

Prometheus vers JSON

Convertissez les métriques Prometheus requises en JSON.

Voir les contrôles 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.

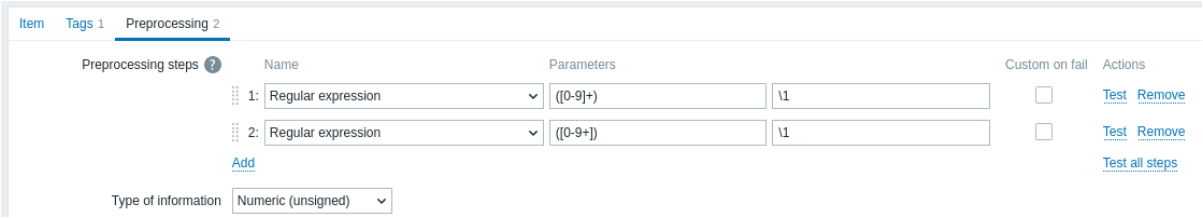
Test

Voir le test du prétraitement.

1 Test du prétraitement

Test

Tester les 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.

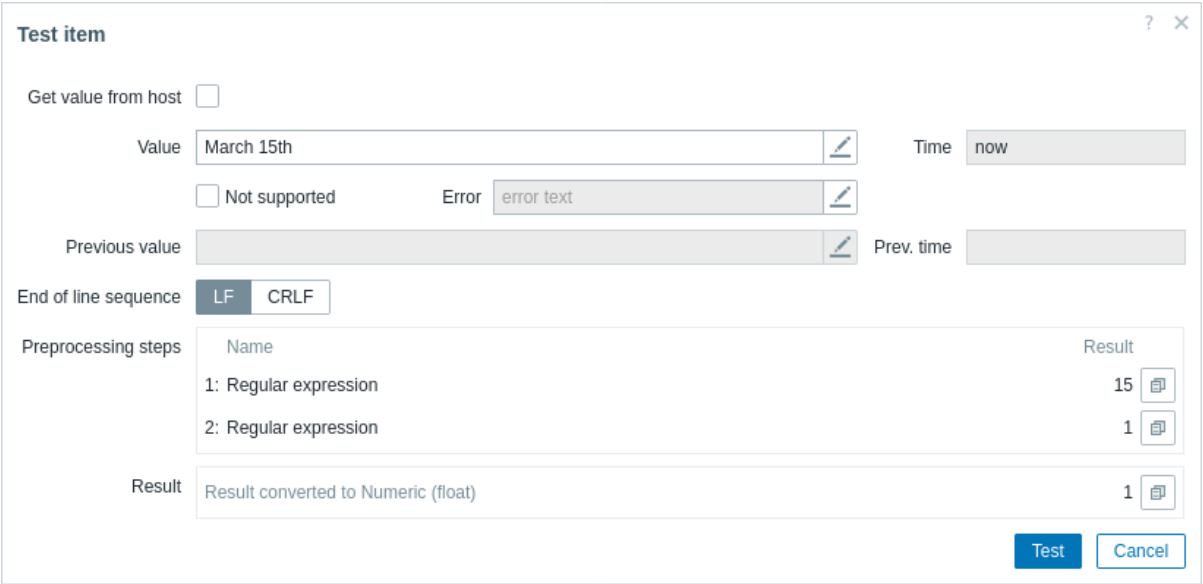


Il est possible de tester :

- avec une valeur hypothétique
- avec une valeur réelle provenant d'un hôte

Chaque étape de prétraitement peut être testée individuellement, et il est également possible de tester toutes les étapes ensemble. Lorsque vous cliquez respectivement sur le bouton *Test* ou *Test all steps* dans le bloc Actions, une fenêtre de test s'ouvre.

Test d'une valeur hypothétique



Paramètre	Description
Obtenir la valeur depuis l'hôte	Si vous souhaitez tester une valeur hypothétique, laissez cette case décochée. Voir aussi : Test d'une valeur réelle.
Valeur	Saisissez la valeur d'entrée à tester. Un clic dans le champ du paramètre ou sur le bouton voir/modifier ouvrira une zone de texte pour saisir la valeur ou le bloc de code.
Non pris en charge	Cochez cette case pour tester une valeur non prise en charge. Cette option est utile pour tester l'étape de prétraitement *Vérifier une valeur non prise en charge*.
Erreur	Saisissez le texte de l'erreur. Ce champ est activé lorsque *Obtenir la valeur depuis l'hôte* est décoché, mais que *Non pris en charge* est coché. Si *Obtenir la valeur depuis l'hôte* est coché, ce champ est rempli avec le message d'erreur réel (lecture seule) provenant de l'hôte.
Heure	L'heure de la valeur d'entrée est affichée : `now` (lecture seule).
Valeur précédente	Saisissez une valeur d'entrée précédente à utiliser pour la comparaison. Uniquement pour les étapes de prétraitement *Change* et *Throttling*.
Heure précédente	Saisissez l'heure de la valeur d'entrée précédente à utiliser pour la comparaison. 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.

Paramètre	Description
Séquence de fin de ligne	Sélectionnez la séquence de fin de ligne pour les valeurs d'entrée multiligne : **LF** - séquence LF (line feed) **CRLF** - séquence CRLF (carriage-return line-feed).
Étapes de prétraitement	Les étapes de prétraitement sont listées ; le résultat du test est affiché pour chaque étape après avoir cliqué sur le bouton *Test*. Les résultats du test sont tronqués à une taille maximale de 512KB lors de l'envoi au frontend. Les résultats du test peuvent être copiés (pas plus que les 512KB tronqués). Si un résultat est tronqué, une icône d'avertissement est affichée. La description de l'avertissement s'affiche au survol de la souris. Notez que les données de plus de 512KB sont tout de même entièrement traitées par le serveur Zabbix. Si l'étape a échoué lors du test, une icône d'erreur est affichée. La description de l'erreur s'affiche au survol de la souris. Si l'option « Custom on fail » est spécifiée 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).
Résultat	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 *Tester toutes les étapes*). Le type de conversion vers le type de valeur de l'élément est également affiché, par exemple `Result converted to Numeric (unsigned)`. Les résultats du test sont tronqués à une taille maximale de 512KB lors de l'envoi au frontend. Le résultat final peut être copié (pas plus que les 512KB tronqués). Si un résultat est tronqué, une icône d'avertissement est affichée. La description de l'avertissement s'affiche au survol de la souris. Notez que les données de plus de 512KB sont tout de même entièrement traitées par le serveur Zabbix.

Cliquez sur *Test* pour voir le résultat après chaque étape de prétraitement.

Les valeurs de test sont conservées entre les sessions de test, que ce soit pour des étapes individuelles ou pour toutes les 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.

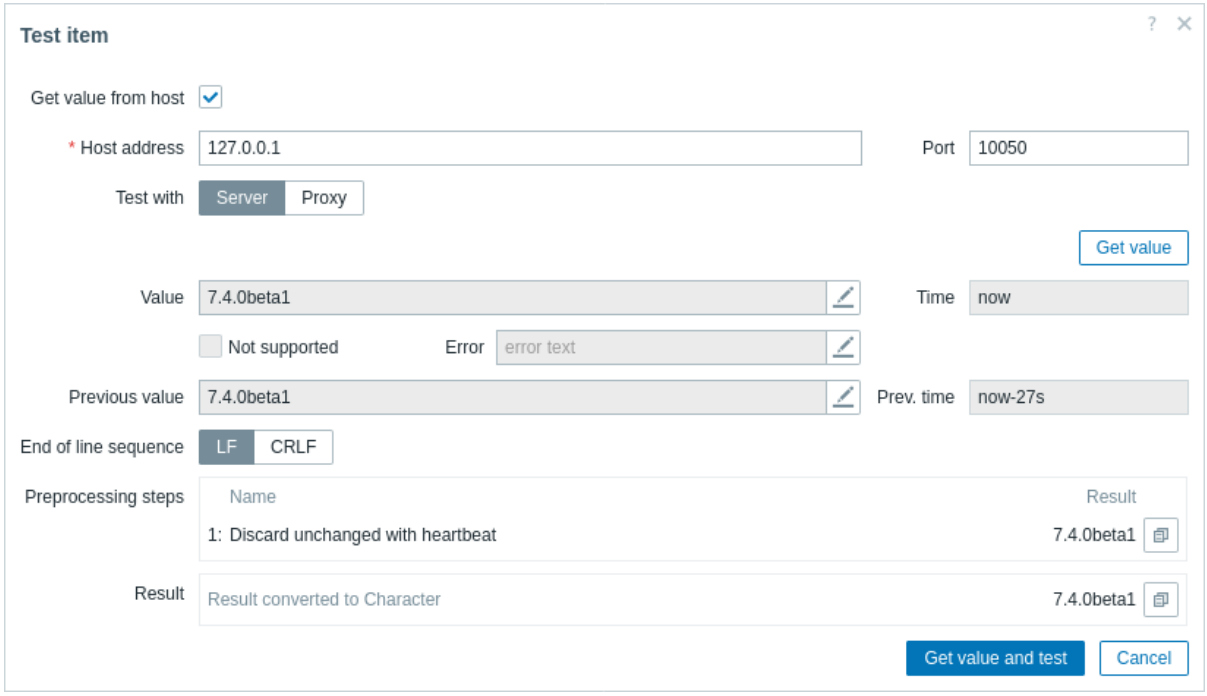
Le test est effectué par le serveur Zabbix. Le frontend envoie une requête correspondante au serveur et attend le résultat. La requête contient la valeur d'entrée et les étapes de prétraitement (avec les macros utilisateur développées). Pour les étapes *Change* et *Throttling*, une valeur et une heure précédentes facultatives peuvent être spécifiées. Le serveur répond avec les résultats pour 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.

Test d'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/absence de proxy) ainsi que les détails spécifiques à l'élément (tels que la communauté SNMPv2 ou les informations d'authentification de sécurité SNMPv3). Ces champs s'adaptent au contexte :
  - Les valeurs sont préremplies lorsque cela est possible, c'est-à-dire, pour les éléments nécessitant un agent, en reprenant les informations de l'interface agent sélectionnée de l'hôte
  - Les valeurs doivent être renseignées manuellement pour les éléments de modèle
  - Les valeurs des macros en texte brut sont résolues
  - Lorsque la valeur du champ (ou une partie de la valeur) est un secret ou une macro Vault, le champ sera vide et devra être rempli manuellement. Si un paramètre de l'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 un mappage de valeurs dans le formulaire de configuration de l'élément (champ « Afficher la valeur »), la boîte de dialogue de test de l'élément affichera une ligne supplémentaire après le résultat final, nommée « Résultat avec mappage de valeurs appliqué ».

Paramètres spécifiques à l'obtention d'une valeur réelle depuis un hôte :

Parameter	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 renseigné avec l'adresse de l'interface hôte de l'élément.
Port	Saisissez le port de l'hôte. Ce champ est automatiquement renseigné avec le port de l'interface hôte de l'élément.
Champs supplémentaires pour les interfaces SNMP (version SNMP, communauté SNMP, nom de contexte, etc.)	Voir Configuration de la supervision SNMP pour plus de détails sur la configuration d'une interface SNMP (v1, v2 et v3). Ces champs sont automatiquement renseignés à partir de l'interface hôte de l'élément.
Proxy	Indiquez le proxy si l'hôte est supervisé par un proxy. Ce champ est automatiquement renseigné avec le proxy de l'hôte (le cas échéant).
Valeur	Valeur récupérée depuis l'hôte. Un clic dans le champ du paramètre ou sur le bouton voir/modifier ouvrira une zone de texte contenant la valeur ou le bloc de code. Les valeurs sont tronquées à une taille maximale de 512KB, et uniquement dans l'interface web. 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 tout de même entièrement traitées par le serveur Zabbix.

Pour le reste des paramètres, voir Test d'une valeur hypothétique ci-dessus.

2 Détails du prétraitement

Aperçu

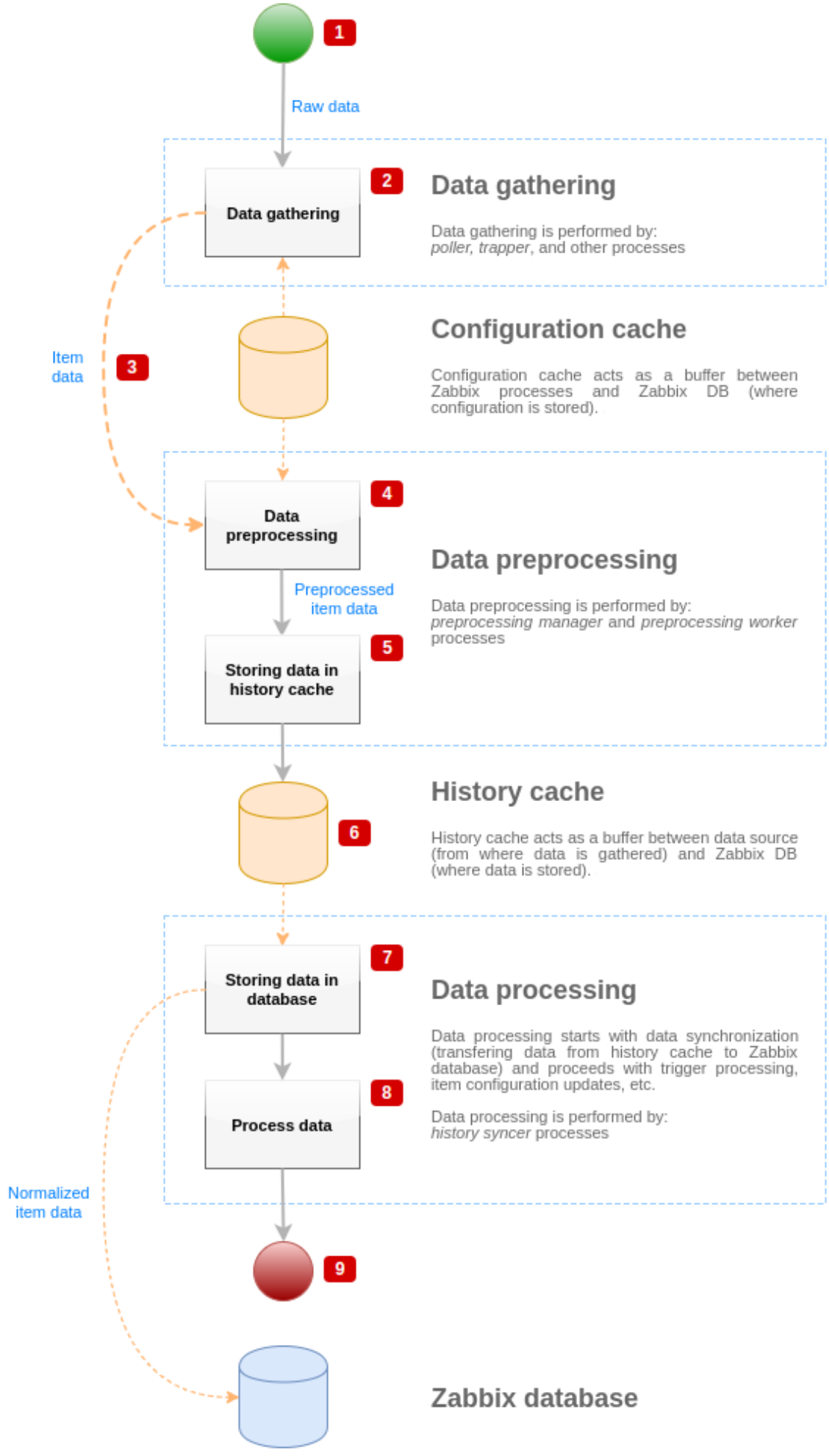
Cette section fournit des détails sur le prétraitement des valeurs d'élément. Le prétraitement des valeurs d'élément permet de définir et d'exécuter des règles de transformation pour les valeurs d'élément reçues.

Le prétraitement est géré par le processus de gestion du prétraitement, ainsi que par des workers de prétraitement 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 (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
6
History cache
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.
Data processing is performed by:
history syncer processes
Process data
8
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.

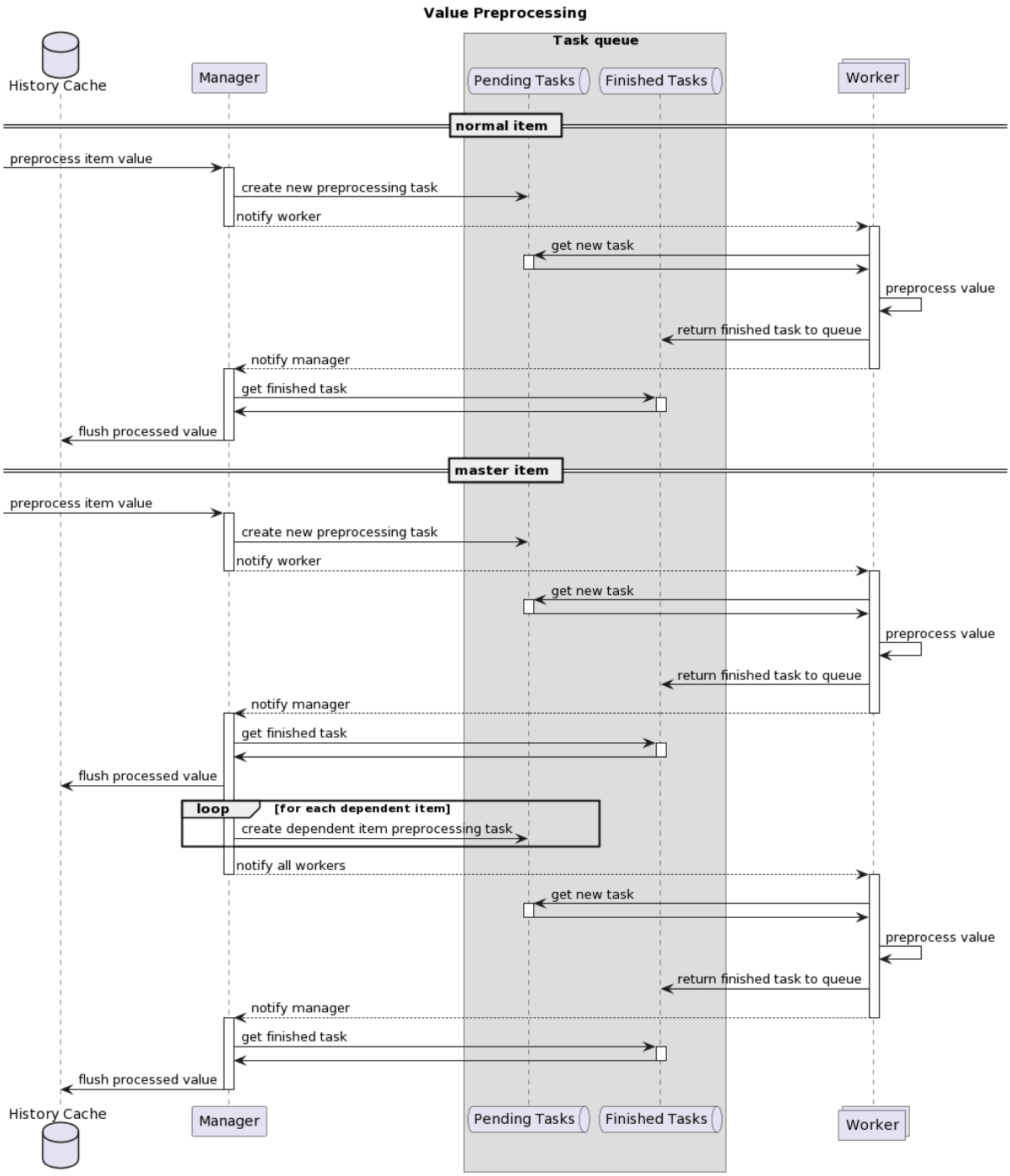
Note:
Un élément peut changer son état en NOT SUPPORTED si la normalisation des données échoue (par exemple, lorsqu'une valeur textuelle ne peut pas être convertie en nombre).

- 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 workers de prétraitement. Le paramètre de configuration StartPreprocessors doit être utilisé pour définir le nombre d'instances de workers de prétraitement démarrées à l'avance, qui doit au minimum 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 pour lesquels il est nécessaire d'exécuter des 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 résoudre les problèmes d'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 manière, il y aura le plus souvent (sauf dans les cas où les données du collecteur arrivent en masse) au moins un worker de prétraitement inoccupé pour les données collectées.

Warning:
Un trop grand nombre de processus de collecte de données (pollers, unreachable pollers, ODBC pollers, HTTP pollers, Java pollers, pingers, trappers, proxypollers), ainsi que IPMI manager, SNMP trapper et les 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 plus tard. Pour éviter ce type de problème, examinez le fichier de configuration du serveur Zabbix afin d'optimiser le nombre de vérifications et de processus simultanés. En outre, si nécessaire, assurez-vous que la limite de descripteurs de fichiers est définie à une valeur suffisamment élevée en vérifiant et en ajustant les limites 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.

3 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.

4 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 infixée.

Opérandes pris en charge :

Operand	Description

`"<text>"`
`'<text>'`
	Constante de texte.
	
	Exemple :
	`'value: \\'1\\''`
	`"value: '1'"`

`<number>`	Constante numérique prenant en charge la notation scientifique. Exemple : 123
`<jsonpath starting with $>`	Valeur référencée par le JSONPath depuis le nœud racine du document d'entrée ; seuls les chemins définis sont pris en charge. Exemple : `$.object.name`
`<jsonpath starting with @>`	Valeur référencée par le JSONPath depuis l'objet/élément actuel ; seuls les chemins définis sont pris en charge. Exemple : `@.name`

Opérateurs pris en charge :

Operator	Type	Description	Result
–	Binary	Soustraction	Nombre
+	Binary	Addition	Nombre
/	Binary	Division	Nombre
*	Binary	Multiplication	Nombre
==	Binary	Égalité	Booléen (1/0)
!=	Binary	Inégalité	Booléen (1/0)
	Binary	Inférieur à	Booléen (1/0)
<=	Binary	Inférieur ou égal à	Booléen (1/0)
>	Binary	Supérieur à	Booléen (1/0)
>=	Binary	Supérieur ou égal à	Booléen (1/0)
=~	Binary	Correspond à l'expression régulière	Booléen (1/0)
!	Unary	NON booléen	Booléen (1/0)

Operator	Type	Description	Result
`||`	Binary	OU booléen	Booléen (1/0)
`&&`	Binary	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
`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": "référence",
      "author": "Nigel Rees",
      "title": "Paroles du siècle",
      "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": "Le Seigneur des anneaux",
      "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": "Réfection de la reliure d'un livre déchiré",
          "price": 99.49
        }
      ]
    }
  },
  "filters": {
    "price": 10,
    "category": "fiction",
    "no filters": "pas de \"filtres\""
```

```
  },
  "closed message": "Le magasin est fermé",
  "tags": [
    "a",
    "b",
    "c",
    "d",
    "e"
  ]
}
```

JSONPath	Type	Résultat
`$.filters.price`	definite	10
`$.filters.category`	definite	fiction
`$.filters['no filters']`	definite	pas de ”filtres”
`$.filters`	definite	{ ”price”: 10, ”category”: ”fiction”, ”no filters”: ”pas de \”filtres\”” }
`$.books[1].title`	definite	Sword of Honour
`$.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
`$.books[0, 2].title`	indefinite	[”Moby Dick”, ”Paroles du siècle”]
`$.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]
`$.books[?(@.id == 4 - 0.4 * 5)].title`	indefinite	[”Sword of Honour”] Remarque : cette requête montre que des opérations arithmétiques peuvent être utilisées dans les requêtes ; elle peut être simplifiée en `$.books[?(@.id == 2)].title`
`$.books[?(@.id == 2 || @.id == 4)].title`	indefinite	[”Sword of Honour”, ”Le Seigneur des anneaux”]
`$.books[?(!(@.id == 2))].title`	indefinite	[”Paroles du siècle”, ”Moby Dick”, ”Le Seigneur des anneaux”]
`$.books[?(@.id != 2)].title`	indefinite	[”Paroles du siècle”, ”Moby Dick”, ”Le Seigneur des anneaux”]
`$.books[?(@.title =~ " of ")].title`	indefinite	[”Paroles du siècle”, ”Sword of Honour”, ”Le Seigneur des anneaux”]
`$.books[?(@.price > 12.99)].title`	indefinite	[”Le Seigneur des anneaux”]
`$.books[?(@.author > "Herman Melville")].title`	indefinite	[”Paroles du siècle”, ”Le Seigneur des anneaux”]
`$.books[?(@.price > $.filters.price)].title`	indefinite	[”Sword of Honour”, ”Le Seigneur des anneaux”]
`$.books[?(@.category == $.filters.category)].title`	indefinite	[”Sword of Honour”,”Moby Dick”,”Le Seigneur des anneaux”]
`$.books[?(@.category == "fiction" && @.price < 10)].title`	indefinite	[”Moby Dick”]

JSONPath	Type	Résultat

`$..[?(@.id)]`	indefinite	[
		{
		”price”: 8.95,
		”id”: 1,
		”category”: ”référence”,
		”author”: ”Nigel Rees”,
		”title”: ”Paroles du siècle”
		},
		{
		”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”: ”Le Seigneur des anneaux”,
		”isbn”: ”0-395-19395-8”
		}
		]

`$.services..[?(@.price > 50)].description`	indefinite	[”Impression et assemblage du livre au format A5”, ”Réfection de la reliure d’un livre déchiré”]
`$..id.length()`	definite	4
`$.books[?(@.id == 2)].title.first()`	definite	Sword of Honour
`$..tags.first().length()`	definite	5 Remarque : `$..tags` est un chemin indéfini, il renvoie donc un tableau des éléments correspondants, c’est-à-dire `[["a", "b", "c", "d", "e" ]]` ; `first()` renvoie le premier élément, c’est-à-dire `["a", "b", "c", "d", "e"]` ; `length()` calcule la longueur de l’élément, c’est-à-dire 5.
`$.books[*].price.min()`	definite	8.95
`$..price.max()`	definite	154.99
`$.books[?(@.category == "fiction")].price.avg()`	definite	14.99
`$.books[?(@.category == $.filters.xyz)].title`	indefinite	Remarque : une requête sans correspondance renvoie NULL pour les chemins définis et indéfinis.
`$.services[?(@.active=="true")].servicegroup`	indefinite	[1001,1000] Remarque : les constantes textuelles doivent être utilisées dans les comparaisons de valeurs booléennes.
`$.services[?(@.active=="false")].servicegroup`	indefinite	[1002] Remarque : les constantes textuelles doivent être utilisées dans les comparaisons de valeurs booléennes.
`$.services[?(@.servicegroup=={{#1}})].description`	definite	[1002]

1 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 suivantes d'échappement des caractères spéciaux s'appliquent :

- seuls les caractères barre oblique inverse (\) et guillemet double (") sont pris en compte pour l'échappement ;
- si la valeur de macro résolue contient ces caractères, chacun d'eux est échappé avec une barre oblique inverse ;
- s'ils sont déjà échappés par une barre oblique inverse, cela n'est pas considéré comme un échappement et la barre oblique inverse ainsi que les caractères spéciaux suivants sont à nouveau échappés.

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 susceptible de 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 susceptible de 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**

5 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 ; le serveur l'encapsulera dans une fonction JavaScript :

```
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 alors renvoyée sous forme de valeur de chaîne), avec quelques exceptions :

- le renvoi de la valeur undefined entraînera une erreur ;
- le renvoi de la valeur null entraînera l'abandon de la valeur d'entrée, de manière similaire au 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 maximal d'exécution de 10 secondes (selon le script, le déclenchement du délai peut prendre plus de temps) ; si ce délai est dépassé, une erreur sera renvoyée. 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 la fois suivante. Toute modification des étapes de prétraitement de l'élément entraînera la réinitialisation du script mis en cache et sa recompilation ultérieure.

Des échecs d'exécution consécutifs (3 d'affilée) entraîneront la réinitialisation du moteur afin de réduire le risque qu'un script perturbe l'environnement d'exécution des scripts suivants (cette action est journalisée avec DebugLevel 4 et supérieur).

Le prétraitement JavaScript est implémenté avec le moteur JavaScript Duktape.

Voir aussi : Objets JavaScript supplémentaires et fonctions globales

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 d'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
    }
}
```

1 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 permettant d'effectuer des requêtes HTTP simples. Les erreurs sont levées sous forme d'exceptions.

Attention:
L'initialisation de plusieurs objets `HttpRequest` est limitée à 10 par exécution de script.

Method	Description
`addHeader(value)`	Ajoute un champ d'en-tête HTTP. Ce champ est utilisé pour toutes les requêtes suivantes jusqu'à ce qu'il soit effacé avec la méthode `clearHeader()`. La longueur totale des champs d'en-tête pouvant être ajoutés à un seul objet `HttpRequest` est limitée à 128 Kio (caractères spéciaux et noms d'en-tête inclus).
`clearHeader()`	Efface l'en-tête HTTP. Si aucun champ d'en-tête n'est défini, `HttpRequest` définira Content-Type sur application/json si les données envoyées sont au format JSON ; sinon text/plain.
`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 de 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 des 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 HTTP reçus seront renvoyées sous forme de tableaux ; cela doit être utilisé pour récupérer les valeurs de champs 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 seront 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'.
	Le drapeau 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').

2 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 la terminant 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)`	Navigue vers l'URL spécifiée. Paramètres : `url` - (string) URL vers laquelle naviguer.
`getUrl()`	Renvoie une chaîne contenant l'URL de la page ouverte.
`getPageSource()`	Renvoie une chaîne contenant le code source de la page ouverte.
`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, sélecteur CSS/texte du lien/texte partiel du lien/nom de balise/Xpath) Stratégie de localisation ; `selector` - (string) Sélecteur d'élément utilisant la stratégie de localisation spécifiée.

Method	Description
`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, sélecteur CSS/texte du lien/texte partiel du lien/nom de balise/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, facultatif) marque de l'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 sur les erreurs, 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 pour le navigateur Chrome.
`firefoxOptions()`	Renvoie un objet `firefoxOptions` contenant des options prédéfinies pour le navigateur Firefox.
`safariOptions()`	Renvoie un objet `safariOptions` contenant des options prédéfinies pour le navigateur Safari.
`edgeOptions()`	Renvoie un objet `edgeOptions` contenant des options prédéfinies pour le navigateur Edge.

Method	Description
`switchFrame(target)`	Bascule vers le cadre spécifié. Paramètres : `target` - (élément du navigateur ou integer, facultatif) 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` ayant 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.

Élément

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 de la page web et fournit des méthodes pour interagir avec celui-ci.

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 est introuvable). Paramètres : `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 est introuvable). Paramètres : `name` - (string) Nom de la propriété.
`getText()`	Renvoie une chaîne de valeur textuelle du texte de l'élément.
`click()`	Clique sur un élément.
`clear()`	Efface le contenu d'un élément modifiable.
`sendKeys(keys)`	Envoie des touches. Paramètres : `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` ne possède 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.
httpOnly	boolean	Indique si le cookie est HTTP-only. Par défaut, `false` si omis lors de l'ajout d'un cookie.

Propriété	Type	Description
expiry	integer	Date 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.

6 Prétraitement CSV vers JSON

Aperçu

Dans cette étape de prétraitement, il est possible de convertir les données du fichier CSV au format JSON. Il est pris en charge dans :

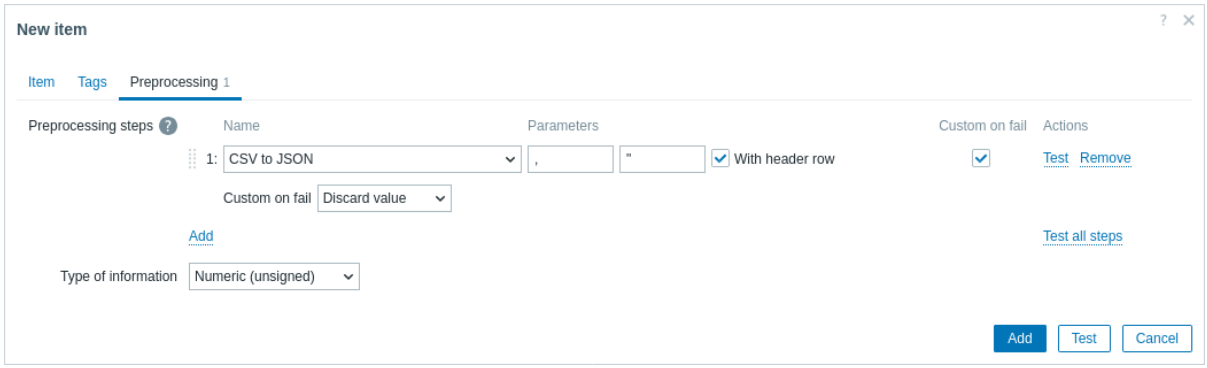
- éléments (prototypes d'éléments)

- règles de découverte de bas niveau

Configuration

Pour configurer une étape de prétraitement CSV vers JSON :

- Accédez à l'onglet Prétraitement dans la configuration de l'élément/la règle de découverte
- Cliquez sur *Ajouter*
- 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'un délimiteur n'est pas extrait de la ligne "Sep=", une virgule est utilisée comme séparateur.

Le deuxième paramètre optionnel permet de définir un symbole de citation.

Si la case *Avec ligne d'en-tête* est cochée, les valeurs de la ligne d'en-tête seront interprétées comme des noms de colonne (voir Traitement des en-têtes 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. De plus, 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"
   }
]
```

3 Types d'éléments

Aperçu

Les types d'éléments couvrent différentes méthodes d'acquisition de données à partir de votre système. Chaque type d'élément est fourni avec son propre ensemble de clés d'élément prises en charge et de paramètres requis.

Les types d'éléments suivants sont actuellement proposés par Zabbix :

- Contrôles par agent Zabbix
- Contrôles par agent SNMP
- Traps SNMP
- Contrôles IPMI
- Contrôles simples
  - Supervision VMware
- Supervision des fichiers journaux
- Éléments calculés
  - Calculs agrégés
- Contrôles internes de Zabbix
- Contrôles SSH
- Contrôles Telnet
- Contrôles externes
- Éléments trapper

- Supervision JMX
- Contrôles ODBC
- Éléments dépendants
- Contrôles HTTP
- Contrôles Prometheus
- Éléments de script
- Éléments de navigateur

Les détails de tous les types d'éléments sont inclus dans les sous-pages de cette section. Même si les types d'éléments offrent de nombreuses options pour la collecte de données, il existe d'autres possibilités via les paramètres utilisateur ou les modules chargeables.

Certains contrôles sont effectués par le seul serveur Zabbix (en tant que supervision sans agent), tandis que d'autres nécessitent l'agent Zabbix, voire la Java gateway de Zabbix (avec la supervision JMX).

> **Attention:**
> Si un type d'élément particulier nécessite une interface particulière (par exemple, un contrôle 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 : agent Zabbix, agent SNMP, JMX et IPMI. Si un élément peut utiliser plus d'une interface, il recherchera parmi les interfaces d'hôte disponibles (dans l'ordre : Agent→SNMP→JMX→IPMI) la première interface appropriée à laquelle être lié.

Tous les éléments qui renvoient du texte (types d'information caractère, journal, texte) peuvent également ne renvoyer que des espaces (le cas échéant), en définissant la valeur de retour sur une chaîne vide (pris en charge depuis la version 2.0).

1 Agent Zabbix

Aperçu

Cette section fournit des détails sur les clés d'élément qui utilisent la communication avec l'agent Zabbix pour la collecte de données.

Il existe des contrôles d'agent passifs et actifs. Lors de la configuration d'un élément, vous pouvez sélectionner le type requis :

- *Agent Zabbix* - pour les contrôles passifs
- *Agent Zabbix (actif)* - pour les contrôles actifs

Notez que toutes les clés d'élément prises en charge par l'agent Zabbix 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.

Clés d'élément supportés

Les clés d'élément que vous pouvez utiliser avec l'agent Zabbix sont répertoriées ci-dessous.

Les clés d'élément sont listées sans paramètres ni informations supplémentaires. Cliquez sur une 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 avec rotation.	
logrt.count	Le nombre de lignes correspondantes dans un fichier journal surveillé avec rotation.	
modbus.get	Lit des 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.	
net.if.collisions	Le nombre de collisions hors fenêtre.	
net.if.discovery	La liste des interfaces réseau.	

Clé d'élément	Description	Groupe d'éléments
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 au port spécifié.	
net.tcp.service	Vérifie si un service est en cours d'exécution et accepte des 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 CPU du 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	Lecture du capteur matériel.	Capteurs
system.boottime	L'heure de démarrage du système.	Système
system.cpu.discovery	La liste des CPU/cœurs CPU détectés.	
system.cpu.intr	Les interruptions de périphériques.	
system.cpu.load	La charge CPU.	
system.cpu.num	Le nombre de CPU.	
system.cpu.switches	Le nombre de changements de contexte.	
system.cpu.util	Le pourcentage d'utilisation CPU.	
system.hostname	Le nom d'hôte du système.	
system.hw.chassis	Les informations sur le châssis.	
system.hw.cpu	Les informations sur le CPU.	
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 système.	
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.).	
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	Le temps de fonctionnement du système en secondes.	
system.users.num	Le nombre d'utilisateurs connectés.	
vfs.dev.discovery	La liste des périphériques de blocs et de leur type.	Systèmes de fichiers virtuels
vfs.dev.read	Les statistiques de lecture disque.	
vfs.dev.write	Les statistiques d'écriture 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 par l'algorithme UNIX cksum.	
vfs.file.contents	Récupération du 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.	
vfs.file.owner	Récupère le propriétaire d'un fichier.	
vfs.file.permissions	Renvoie une chaîne de 4 chiffres contenant le nombre octal des permissions UNIX.	
vfs.file.regexp	Récupère une chaîne dans le fichier.	
vfs.file.regmatch	Trouve une chaîne dans le fichier.	

Clé d'élément	Description	Groupe d'éléments
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, l'espace disque disponible, les statistiques d'inodes et les options de montage.	
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	Obtient le contenu d'une page web.	Surveillance web
web.page.perf	Le temps de chargement d'une page web complète.	
web.page.regexp	Trouve 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 l'agent Zabbix (agent Zabbix ou agent Zabbix 2).	
agent.version	La version de l'agent Zabbix.	
zabbix.stats	Renvoie à distance un ensemble de métriques internes du serveur Zabbix ou du proxy.	
zabbix.stats	Renvoie à distance le nombre d'éléments surveillés dans la file d'attente qui sont retardés sur le serveur Zabbix ou le proxy.	

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 plateformes 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 codes ;
- **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) ou *skip* - ignorer le traitement des anciennes données (affecte uniquement les éléments nouvellement créés) ;
- **output** - un modèle facultatif de formatage de sortie. La séquence d'échappement **\0** est remplacée par la partie du texte qui correspond (du premier caractère où la correspondance commence jusqu'au caractère où elle se termine), tandis qu'une séquence d'échappement **\N** (où N=1...9) est remplacée par le Nième 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'analyser les lignes les plus récentes 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 désormais mtime est 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 également les notes supplémentaires sur les fichiers persistants.

Commentaires :

- L'élément doit être configuré comme une vérification active.
- Si le fichier est manquant ou si les permissions ne permettent pas l'accès, l'élément devient 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 « Result is TRUE », renvoient toujours la ligne entière correspondante et le paramètre `output` est ignoré.
- L'extraction du contenu à l'aide du paramètre `output` a lieu 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 l'envoi 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 plate-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 la page de codes ;
- **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) ou *skip* - ignorer le traitement des anciennes données (affecte uniquement 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 que les lignes les plus récentes soient 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 désormais mtime est 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 également 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 l'accès, l'élément devient non pris en charge.

logrt[file regexp,<regexp>,<encoding>,<maxlines>,<mode>,<output>,<maxdelay>,<options>,<persistent dir>]

 La surveillance d'un fichier journal soumis à une rotation.
 Valeur de retour : *Log*.
 Voir les plateformes prises en charge.

Paramètres :

- **file regexp** - le chemin absolu vers le fichier, avec le nom du 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 à l'intégralité du nom (par exemple, /path/to/agent correspondra à zabbix_agentd.log).
- **regexp** - une expression régulière décrivant le motif de contenu requis.
- **encoding** - l'identifiant de la page de codes.
- **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) ou *skip* - ignorer le traitement des anciennes données (affecte uniquement les éléments nouvellement créés).
- **output** - un modèle facultatif de formatage de sortie. La séquence d'échappement **\0** est remplacée par la partie du texte qui correspond (du premier caractère où la correspondance commence jusqu'au caractère où elle se termine), tandis qu'une séquence d'échappement **\N** (où N=1...9) est remplacée par le Nième 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'analyser les lignes les plus récentes 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 égal à 0 ou ne pas être spécifié ; voir les notes sur 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 également les notes supplémentaires sur les fichiers persistants.

Commentaires :

- L'élément doit être configuré comme une vérification active.
- La rotation des journaux est basée sur l'heure de dernière modification des fichiers.
- Notez que logrt est conçu pour fonctionner avec un seul fichier journal actuellement actif, avec plusieurs autres fichiers correspondants inactifs ayant subi une rotation. 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 élément logrt prend en charge trop de fichiers, cela peut entraîner un épuisement de la mémoire et un arrêt du processus de 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 « Result is TRUE », renvoient toujours la ligne entière correspondante et le paramètre `output` est ignoré.
- L'extraction du contenu à l'aide du paramètre `output` a lieu 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 séparateur de répertoire correct :
  - Sous Windows, le séparateur doit être une barre oblique inverse (\) ; les barres obliques (/) peuvent être tolérées à d'autres positions, sauf à 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}$",,,100] #cet élément correspondra à un fichier tel que "logfi
logrt["/home/user/^logfile_.*_[0-9]{1,3}$","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 l'envoi 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 faisant l'objet d'une rotation.
 Valeur de retour : *Integer*.
 Voir les plateformes prises en charge.

Paramètres :

- **file regexp** - le chemin absolu vers le fichier, avec le nom du 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 la page de codes.
- **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) ou *skip* - ignorer le traitement des anciennes données (affecte uniquement 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'analyser les lignes les plus récentes 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 ne pas être spécifié ; voir les notes sur copytruncate,
mtime-reread - enregistrements non uniques, relire si l'heure de modification ou la taille change (par défaut),
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 également 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.
- La rotation des journaux est basée sur l'heure de dernière modification des fichiers.
- Dans le paramètre `file regexp`, le chemin du répertoire du journal et l'expression régulière du fichier journal doivent être séparés par le séparateur de répertoire correct :
  - Sous Windows, le séparateur doit être une barre oblique inverse (\) ; les barres obliques (/) peuvent être tolérées à d'autres positions, sauf à celle qui sépare le chemin du répertoire du journal 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 des 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 de l'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 plateformes prises en charge.

Paramètres :

- **ip** (ignoré sous Windows sauf en cas d'utilisation de Zabbix agent 2) - l'adresse IP du serveur DNS (laisser vide pour 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 en cas d'utilisation de Zabbix agent 2) - le délai d'expiration de la requête en secondes (la valeur par défaut est de 1 seconde) ;
- **count** (ignoré sous Windows sauf en cas d'utilisation de Zabbix agent 2) - le nombre de tentatives pour la requête (la valeur par défaut est de 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 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 indisponible ; secondes - le nombre de secondes passées à attendre une réponse du service).
 Voir les plateformes prises en charge.

Paramètres :

- **ip** (ignoré sous Windows sauf en cas d'utilisation de Zabbix agent 2) - l'adresse IP du serveur DNS (laisser vide pour 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 en cas d'utilisation de Zabbix agent 2) - le délai d'expiration de la requête en secondes (la valeur par défaut est de 1 seconde) ;
- **count** (ignoré sous Windows sauf en cas d'utilisation de Zabbix agent 2) - le nombre de tentatives pour la requête (la valeur par défaut est de 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 à la fois au format inversé et 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 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 qu'un plus grand nombre de types d'enregistrements et un contrôle accru de la supervision DNS. Valeur de retour : une chaîne de caractères contenant le type d'information requis.
 Voir les plateformes prises en charge.

Paramètres :

- **ip** (ignoré sous Windows sauf en cas d'utilisation de Zabbix agent 2) - l'adresse IP du serveur DNS (laisser 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 en cas d'utilisation de Zabbix agent 2) - le délai d'expiration de la requête en secondes (la valeur par défaut est de 1 seconde) ;
- **count** (ignoré sous Windows sauf en cas d'utilisation de Zabbix agent 2) - le nombre de tentatives pour la requête (la valeur par défaut est de 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 par Zabbix agent sous Windows, ni par 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.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. Utilisée 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.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 du périphérique
multicast - nombre de trames multicast reçues par le pilote du 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 renvoyée : *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 du tampon FIFO
collisions (colls) - nombre de collisions détectées sur l'interface
carrier - nombre de pertes de porteuse détectées par le pilote du périphérique
compressed - nombre de paquets compressés transmis par le pilote du 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 sous Linux et HP-UX.
- Les modes *overruns*, *collision*, *carrier*, *compressed* sont pris en charge uniquement sous Linux.
- Sous HP-UX, cet élément ne fournit pas de détails sur les interfaces loopback (par exemple, lo0).

Exemples :

```
net.if.out[eth0]
net.if.out[eth0,errors]
```

net.if.total[if,<mode>]

 La somme des statistiques de 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 du périphérique

Commentaires :

- Vous pouvez utiliser cette clé avec l'étape de prétraitement *Change per second* afin d'obtenir des statistiques en octets par seconde.
- Le mode *dropped* est pris en charge uniquement sur Linux et HP-UX. Les paquets abandonnés sont pris en charge uniquement si `net.if.in` et `net.if.out` fonctionnent 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 exemple, 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 au port spécifié.
 Valeurs de retour : 0 - connexion impossible ; 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 des démons système (les sessions SMTP et SSH étant généralement journalisées).

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 les connexions TCP.
 Valeurs de retour : 0 - le service est arrêté ; 1 - le service est en cours d'exécution.
 Voir les plateformes 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 (la valeur par défaut est 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 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.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] #cet élément peut être utilisé pour tester la disponibilité du serveur FTP sur le
```

net.tcp.service.perf[service,<ip>,<port>]

 Vérifie les performances d'un service TCP.
 Valeurs de retour : *Float* (0 - le service est indisponible ; secondes - le nombre de secondes passées à attendre une réponse du service).
 Voir les plateformes 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 (la valeur par défaut est 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 vérifications.
- 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 plateformes prises en charge.

Paramètres :

- **service** - *ntp* (voir les détails) ;
- **ip** - l'adresse IP ou le nom DNS (la valeur par défaut est 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 indisponible ; secondes - le nombre de secondes passées à attendre une réponse du service).
 Voir les plateformes prises en charge.

Paramètres :

- **service** - *ntp* (voir les détails) ;
- **ip** - l'adresse IP ou le nom DNS (la valeur par défaut est 127.0.0.1) ;
- **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 (par défaut, *tous les processus*) ;
- **user** - le nom d'utilisateur (par défaut, *tous les utilisateurs*) ;
- **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 renvoyée est basée sur le pourcentage d'utilisation d'un seul cœur CPU. Par exemple, l'utilisation CPU d'un processus utilisant pleinement deux cœurs 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 auxquelles aucun accès n'a eu lieu au cours des dernières 24 heures sont supprimées du collecteur.
- Lorsque le paramètre `zone` est défini sur *current* (ou par défaut), si l'agent a été compilé sur un Solaris sans prise en charge des zones, mais s'exécute sur un Solaris plus récent où les zones sont prises en charge, alors l'agent renverra NOTSUPPORTED (l'agent ne peut pas limiter les résultats à la zone courante 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 d'utilisateur (par défaut *tous les utilisateurs*) ;
- **cmdline** - filtrer par ligne de commande (il s'agit d'une expression régulière). Ce paramètre n'est pas pris en charge sous Windows ; sur les autres plateformes, il n'est pas pris en charge si le mode est défini sur « summary ».
- **mode** - valeurs possibles :
process (par défaut), *thread* (non pris en charge pour NetBSD), *summary*. Consultez la 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 s'est déjà terminé, manque de permissions, 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'utilisat
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'utilisateur
```

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 (la valeur par défaut est *tous les processus*) ;
- **user** - le nom de l'utilisateur (la valeur par défaut est *tous les utilisateurs*) ;
- **mode** - valeurs possibles : *avg*, *max*, *min* ou *sum* (par défaut) ;
- **cmdline** - filtrer 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 sous Linux, FreeBSD, Solaris[6] et AIX.
- Lorsque plusieurs processus utilisent de la mémoire partagée, la somme de la mémoire utilisée par les processus peut entraîner 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 *tous les processus*) ;
- **user** - le nom d'utilisateur (la valeur par défaut est *tous les utilisateurs*) ;
- **state** - valeurs possibles :
all (par défaut),
disk - sommeil non interruptible,
run - en cours d'exécution,
sleep - sommeil interruptible,
trace - arrêté,
zomb - zombie ;
- **cmdline** - filtrer par ligne de commande (il s'agit d'une expression régulière) ;
- **zone** - la zone cible : *current* (par défaut), ou *all*. Ce paramètre n'est pris en charge que sur Solaris.

Commentaires :

- Les paramètres d'état *disk* et *trace* ne sont pris en charge que sur Linux, FreeBSD, OpenBSD et NetBSD.
- Sous Windows, seuls les paramètres `name` et `user` sont pris en charge.
- Lorsque cet élément est invoqué depuis la ligne de commande et contient un paramètre de ligne de commande (par exemple en utilisant le mode test de l'agent : `zabbix_agentd -t proc.num[,,,apache2]`), un processus supplémentaire sera compté, car l'agent se comptera lui-même.
- Lors du réglage 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 un Solaris plus récent où les zones sont prises en charge, alors l'agent renverra NOTSUPPORTED (l'agent ne peut pas limiter les résultats à la zone courante 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 en état de sommeil exécutés sous Oracle ayant
```

sensor[device,sensor,<mode>]

 Valeur du 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 sous Linux 2.4.
- Lit /sys/class/hwmon sous Linux 2.6+. Voir une description plus détaillée de l'élément sensor sous Linux.
- Lit le MIB *hw.sensors* sous OpenBSD.
- Sous Linux 2.4, si vous spécifiez mode, alors device et sensor sont traités comme des expressions régulières.
- Sous 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] #the temperature of one CPU
sensor["cpu[0-2]$",temp,avg] #the average temperature of the first three CPUs
```

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 points), *fqdn* (renvoie le nom de domaine pleinement qualifié) ;
- **transform** - valeurs possibles : *none* (par défaut) ou *lower* (convertit en minuscules).

La valeur est obtenue en prenant `nodename` dans 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>]

 La 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 :

- Répertorie les adresses MAC des interfaces dont le nom correspond à l'expression régulière `interface` donnée (*all* répertorie toutes les interfaces).
- Si `format` est spécifié comme *short*, les noms d'interface et les adresses MAC identiques ne sont pas répertoriés.

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 plateformes prises en charge.

Paramètres :

- **type** - valeurs possibles : *utc* - (par défaut) le temps écoulé depuis l'époque Unix (00:00:00 UTC, 1er janvier 1970), mesuré en secondes, ou *local* - l'heure au format 'yyyy-mm-dd,hh:mm:ss.nnn,+hh:mm'

Doit être utilisé comme une vérification passive pour l'agent Zabbix ; sur l'agent Zabbix 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 *texte* 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) ou *nowait* - ne pas attendre.

Commentaires :

- Cet élément est désactivé par défaut. Découvrez comment les activer.
- La valeur de retour de l'élément correspond à la sortie standard ainsi qu'à la sortie d'erreur standard produites par la commande. La vérification du code de sortie n'est pas effectuée.
- Pour être traitée correctement, la valeur de retour de la commande doit être du type de données `text`. Un résultat vide est également autorisé.
- La valeur de retour est limitée à 16MB (y compris les espaces 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 /] #renvoie une liste détaillée des fichiers du répertoire racine
```

system.stat[resource,<type>]

 Les statistiques du 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 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 fichier par seconde (float)
fo - sorties de pages de fichier 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 page) (float)
sr - pages analysées par l'algorithme de remplacement de page (float)
- **faults,<type>** - taux de traps et d'interruptions :
in - interruptions de périphérique (float)
sy - appels système (float)
cs - changements de contexte des threads du noyau (float)
- **cpu,<type>** - répartition en pourcentage de l'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 une ou plusieurs requêtes d'E/S disque/NFS en attente (float)
pc - nombre de processeurs physiques consommés (float)
ec - pourcentage de capacité allouée 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 émis vers le disque/bande physique (float)

Commentaires :

- Notez 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>]

 Les 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 sont obtenues à partir de (notez que tous les fichiers et toutes les options ne sont pas présents dans toutes les distributions) :

- `/proc/version` (*full*) sous Linux ;
- `/proc/version_signature` (*short*) sous 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` sous 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 : *Texte*.
 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 :

- Répertorie (par ordre alphabétique) les paquets installés dont le nom correspond à l'expression régulière donnée (*all* les répertorie tous).
- Gestionnaires de paquets pris en charge (commande exécutée) :
  - dpkg (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 distincte 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, valeur par défaut sur les plateformes non Linux), *sectors* (secteurs chargés en mémoire via le swap) ou *pages* (pages chargées en mémoire via le swap, valeur par défaut sous 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* n'est pris en charge que sous Linux.

Exemple :

```
system.swap.in[,pages]
```

system.swap.out[<device>,<type>]

 Statistiques de swap-out (de la mémoire vers le périphérique).
 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-outs, valeur par défaut sur les plateformes non Linux), *sectors* (secteurs écrits dans le swap) ou *pages* (pages écrites dans le swap, valeur 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 sous 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) ou *used* (espace d'échange utilisé).

Commentaires :

- Notez que *pfree* et *pused* ne sont pas pris en charge sous Windows si la taille du swap est de 0.
- Si le périphérique n'est pas spécifié, l'agent Zabbix ne prendra en compte que les périphériques (fichiers) d'échange ; la mémoire physique sera ignorée. Par exemple, sur les systèmes Solaris, la commande `swap -s` inclut une partie de la mémoire physique et les 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.read[<device>,<type>,<mode>]

 Les statistiques de lecture 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 que lorsque `type` est dans : sps, ops, bps.

Commentaires :

- Si un intervalle de mise à jour de trois heures ou plus[2] est utilisé, cet élément renverra toujours '0'.
- Les paramètres de type *sectors* et *sps* ne sont pris en charge que sous Linux.
- Le paramètre de type *ops* n'est pris en charge que sous Linux et FreeBSD.
- Le paramètre de type *bps* n'est pris en charge que sous FreeBSD.
- Le paramètre de type *bytes* n'est pris en charge que sous FreeBSD, Solaris, AIX, OpenBSD.
- Le paramètre `mode` n'est pris en charge que sous Linux et 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' pour 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 sous Linux.
- Le paramètre de type *ops* n'est pris en charge que sous Linux et FreeBSD.
- Le paramètre de type *bps* n'est pris en charge que sous FreeBSD.
- Le paramètre de type *bytes* n'est pris en charge que sous FreeBSD, Solaris, AIX, OpenBSD.
- Le paramètre `mode` n'est pris en charge que sous Linux et 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 » pour 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 de retour : *Integer*.
 Voir les plateformes prises en charge.

Paramètres :

- **dir** - le chemin absolu vers le répertoire ;
- **regex incl** - une expression régulière décrivant le motif de nom de l'entité (fichier, répertoire, lien symbolique) à inclure ; inclut tout si vide (valeur par défaut) ;
- **regex excl** - une expression régulière décrivant le motif de nom de l'entité (fichier, répertoire, lien symbolique) à exclure ; n'exclut rien 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 de bloc, *cdev* - périphérique de caractères, *fifo* - FIFO, *dev* - synonyme de "bdev,cdev", *all* - tous les types (par défaut), c.-à-d. "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 pour qu'il soit compté. Les fichiers plus petits ne seront pas comptés. Les suffixes de mémoire peuvent être utilisés.
- **max size** - la taille maximale (en octets) du fichier pour qu'il soit compté. Les fichiers plus grands ne seront pas comptés. Les suffixes de mémoire peuvent être utilisés.
- **min age** - l'âge minimal (en secondes) de l'entrée de répertoire pour qu'elle soit comptée. Les entrées plus récentes ne seront pas comptées. Les suffixes de temps peuvent être utilisés.
- **max age** - l'âge maximal (en secondes) de l'entrée de répertoire pour qu'elle soit comptée. Les entrées de cet âge 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 motif 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 présenter un risque de sécurité.
- Les variables d'environnement, par ex. %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, alors tous les sous-répertoires seront parcourus, mais seuls les fichiers de type zip seront comptés).
- Le temps d'exécution est limité par la valeur de délai d'expiration dans la configuration de l'agent (3 s). 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 comptage 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 (quelques Ko en général). 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 définir <types_incl> sur "*file*" afin d'éviter les surprises.

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 de retour : *chaîne JSON*.
 Voir les plateformes prises en charge.

Paramètres :

- **dir** - le chemin absolu vers le répertoire ;
- **regex incl** - une expression régulière décrivant le motif de nom de l'entité (fichier, répertoire, lien symbolique) à inclure ; inclut tout si vide (valeur par défaut) ;
- **regex excl** - une expression régulière décrivant le motif de nom de l'entité (fichier, répertoire, lien symbolique) à exclure ; n'exclut rien 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 de bloc, *cdev* - périphérique de caractères, *fifo* - FIFO, *dev* - synonyme de "bdev,cdev", *all* - tous les types (par défaut), c.-à-d. "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 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) pour qu'un fichier soit listé. Les fichiers plus petits ne seront pas listés. Les suffixes de mémoire peuvent être utilisés.
- **max size** - la taille maximale (en octets) pour qu'un fichier soit listé. 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 pour être listée. 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 pour être listée. Les entrées aussi anciennes ou plus anciennes ne seront pas listées (heure de modification). Les suffixes de temps peuvent être utilisés.
- **regex excl dir** - une expression régulière décrivant le motif 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 présenter un risque de sécurité.
- Les variables d'environnement, par ex. %APP_HOME%, $HOME et %TEMP%, ne sont pas prises en charge.
- Les pseudo-répertoires "." et ".." ne sont jamais listés.
- Les liens symboliques ne sont jamais suivis lors du parcours des répertoires.
- `regex incl` et `regex excl` sont tous deux appliqués aux fichiers et aux répertoires lors de la génération de la liste des entrées, mais sont ignorés lors de la sélection des sous-répertoires à parcourir (si `regex incl` vaut "(?i)^.+\.zip$" et que `max depth` n'est pas défini, alors 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'expiration dans la configuration de l'agent. Comme le parcours de grands répertoires peut prendre plus de temps, aucune donnée ne sera renvoyée et l'élément deviendra non pris en charge.

Une liste partielle ne sera pas renvoyée.

- 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 (quelques Ko en général). 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 définir `types incl` sur "*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 de nom de l'entité (fichier, répertoire, lien symbolique) à inclure ; inclut tout si vide (valeur par défaut) ;
- **regex excl** - une expression régulière décrivant le motif de nom de l'entité (fichier, répertoire, lien symbolique) à exclure ; n'exclut rien si vide (valeur par défaut) ;
- **mode** - valeurs possibles : *apparent* (par défaut) - obtient les tailles apparentes des fichiers plutôt que l'utilisation du disque (agit comme `du -sb dir`), *disk* - obtient l'utilisation du 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** - la profondeur maximale des sous-répertoires à parcourir : **-1** (par défaut) - illimitée, **0** - ne descend pas dans les sous-répertoires.
- **regex excl dir** - une expression régulière décrivant le motif 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 présenter un risque de sécurité.
- Seuls les répertoires disposant au minimum du droit de lecture pour l'utilisateur *zabbix* sont calculés. Pour les répertoires disposant uniquement du droit de lecture, seule la taille du répertoire lui-même est calculée. Les répertoires disposant des droits de lecture et d'exécution sont calculés avec leur contenu.
- Avec de grands répertoires ou des disques lents, cet élément peut expirer en raison du paramètre Timeout dans les fichiers de configuration de agent et de server/proxy. Augmentez les valeurs de délai d'attente si nécessaire.
- La limite de taille de fichier dépend de la prise en charge des grands fichiers.

Exemples :

```
vfs.dir.size[/tmp,log] #calculates the size of all files in /tmp containing 'log' in their names
vfs.dir.size[/tmp,log,^.+\.old$] #calculates the size of all files in /tmp containing 'log' in their names
```

vfs.file.cksum[file,<mode>]

 La somme de contrôle du fichier, calculée par l'algorithme UNIX cksum.
 Valeur de retour : *Integer* - avec `mode` défini sur *crc32*, *String* - avec `mode` défini sur *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 présenter 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 : *Texte*.
 Voir les plateformes prises en charge.

Paramètres :

- **file** - le chemin complet vers le fichier ;
- **encoding** - l'identifiant de la page de codes.

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 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.
- La marque d'ordre des octets (BOM) est exclue 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 - un fichier du type spécifié existe.
 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 de bloc), *cdev* (périphérique de caractères), *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 présenter un risque de sécurité.
- Plusieurs types doivent être séparés par une virgule et l'ensemble complet 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 fichiers volumineux.

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 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 présenter un risque de sécurité.
- Types de fichiers pris en charge sur les systèmes de type UNIX : fichier ordinaire, répertoire, lien symbolique, socket, périphérique de bloc, périphérique de caractère, FIFO.
- La limite de taille du fichier dépend de la prise en charge des fichiers volumineux.

Exemple :

```
vfs.file.get[/etc/passwd] #renvoie un JSON contenant des informations sur le fichier /etc/passwd (type, ut
```

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 présenter un risque de sécurité.
- La limite de taille du 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 plateformes 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 sous Unix, SID sous Windows.

Commentaires :

- Sous Windows, cet élément suit les chemins UNC, ce qui peut présenter un risque de sécurité.
- La limite de taille de fichier dépend de la prise en charge des fichiers volumineux.

Exemples :

```
vfs.file.owner[/tmp/zabbix_server.log] #return the file owner of /tmp/zabbix_server.log
vfs.file.owner[/tmp/zabbix_server.log,,id] #return the file owner ID of /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 codes ;
- **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 facultatif de formatage de sortie. La séquence d'échappement **\0** est remplacée par la partie du texte correspondante (du premier caractère où la correspondance commence jusqu'au caractère où elle se termine), tandis qu'une séquence d'échappement **\N** (où N=1...9) est remplacée par le Nième groupe correspondant (ou par 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.
- La marque d'ordre des octets (BOM) est exclue de la sortie.
- L'extraction du contenu à l'aide du paramètre `output` a lieu 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'identifiant de l'utilisateur *zabb
```

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 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 codes ;
- **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 fichiers volumineux.
- La marque d'ordre des octets (BOM) est ignorée.

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) ou *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 disposer des autorisations 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>]

 Les informations temporelles du fichier.
 Valeur de retour : *Integer* (horodatage Unix).
 Voir les 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,
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 présenter un risque de sécurité.
- La limite de taille du fichier dépend de la prise en charge des fichiers volumineux.

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

 La liste des systèmes de fichiers montés avec leur type, l'espace disque disponible, les statistiques d'inodes et les options de montage. Peut être utilisée 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.

Commentaires :

- Les systèmes de fichiers dont le nombre d'inodes est égal à zéro, ce qui peut être le cas pour les systèmes de fichiers avec des inodes dynamiques (par exemple btrfs), sont également signalés.
- Voir aussi : Découverte des systèmes de fichiers montés.

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>]

 Récupère le contenu d'une page web.
 Valeur de retour : code 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]. Un schéma manquant sera traité comme *http*. Si une URL est spécifiée, `path` et `port` doivent être vides. La spécification d'un nom d'utilisateur/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)

Commentaires :

- Cet élément devient non pris en charge si la ressource spécifiée dans `host` n'existe pas ou n'est pas disponible.
- `host` peut être un nom d'hôte, un nom de domaine, une adresse IPv4 ou IPv6. Mais pour une adresse IPv6, l'agent Zabbix doit être compilé avec la prise en charge d'IPv6 activée.

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.perf[host,<path>,<port>]

 Le temps de chargement d'une page web complète (en secondes).
 Valeur de retour : *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]. Un schéma manquant sera traité comme *http*. Si une URL est spécifiée, `path` et `port` doivent être vides. La spécification d'un nom d'utilisateur/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)

Commentaires :

- Cet élément devient non pris en charge si la ressource spécifiée dans `host` n'existe pas ou n'est pas disponible.
- `host` peut être un nom d'hôte, un nom de domaine, une adresse IPv4 ou IPv6. Mais pour une adresse IPv6, l'agent Zabbix doit être compilé avec la prise en charge d'IPv6 activée.

Exemples :

```
web.page.perf[www.example.com,index.php,80]
web.page.perf[https://www.example.com]
```

web.page.regexp[host,<path>,<port>,regexp,<length>,<output>]

 Trouve une chaîne sur la page web.
 Valeur de retour : la chaîne correspondante, ou comme spécifié par le paramètre facultatif `output`.
 Voir les plateformes prises en charge.

Paramètres :

- **host** - le nom d'hôte ou l'URL (sous la forme `scheme://host:port/path`, où seul *host* est obligatoire). Schémas d'URL autorisés : *http*, *https*[4]. Un schéma manquant sera traité comme *http*. Si une URL est spécifiée, `path` et `port` doivent être vides. La spécification d'un nom d'utilisateur/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)
- **regexp** - une expression régulière décrivant le motif requis ;

- **length** - le nombre maximal de caractères à renvoyer ;
- **output** - un modèle facultatif de formatage de sortie. La séquence d'échappement **\0** est remplacée par la partie du texte correspondante (du premier caractère où la correspondance commence jusqu'au caractère où elle se termine), tandis qu'une séquence d'échappement **\N** (où N=1...9) est remplacée par le Nième groupe correspondant (ou par une chaîne vide si N dépasse le nombre de groupes capturés).

Commentaires :

- Cet élément devient non pris en charge si la ressource spécifiée dans `host` n'existe pas ou n'est pas disponible.
- `host` peut être un nom d'hôte, un nom de domaine, une adresse IPv4 ou IPv6. Mais 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` a lieu sur l'agent.

Exemples :

```
web.page.regexp[www.example.com,index.php,80,OK,2]
web.page.regexp[https://www.example.com,,,OK,2]
```

agent.hostmetadata

 Les métadonnées d'hôte de l'agent.
 Valeur de retour : *String*.
 Voir les plateformes 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 les plateformes prises en charge.

Renvoie :

- En tant que contrôle passif - le nom du premier hôte indiqué dans le paramètre Hostname du fichier de configuration de l'agent ;
- En tant que contrôle actif - le nom du nom d'hôte actuel.

agent.ping

 La vérification de disponibilité de l'agent.
 Valeur de retour : Nothing - indisponible ; 1 - disponible.
 Voir les plateformes prises en charge.

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 à distance un ensemble de métriques internes du serveur ou du proxy Zabbix.
 Valeur de retour : *chaîne JSON*.
 Voir les plateformes prises en charge.

Paramètres :

- **ip** - la liste d'IP/DNS/masques réseau des serveurs/proxys à interroger à distance (127.0.0.1 par défaut) ;
- **port** - le port du serveur/proxy à interroger à distance (10051 par défaut)

Commentaires :

- Un ensemble sélectionné de métriques internes est renvoyé par cet élément. Pour plus de détails, voir Supervision à distance des statistiques Zabbix.
- Notez que la requête de statistiques ne sera acceptée que depuis les adresses répertoriées dans le paramètre server/proxy 'StatsAllowedIP' sur l'instance cible.

zabbix.stats[<ip>,<port>,queue,<from>,<to>]

 Renvoie le nombre d'éléments surveillés dans la file d'attente qui sont retardés à distance sur le serveur ou le proxy Zabbix.
 Valeur de retour : *chaîne JSON*.
 Voir les plateformes prises en charge.

Paramètres :

- **ip** - la liste IP/DNS/masque 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** - retardé d'au moins (la valeur par défaut est 6 secondes)
- **to** - retardé 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 server/proxy 'StatsAllowedIP' sur 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 de 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, alors la clé renverra des statistiques récapitulatives, incluant tous les périphériques de blocs comme sda, sdb, ainsi que leurs partitions (sda1, sda2, sdb3...) et les périphériques multiples (RAID MD) basés sur ces périphériques de blocs/partitions, ainsi que les volumes logiques (LVM) basés sur ces périphériques de blocs/partitions. Dans de tels cas, les valeurs renvoyées doivent être considérées uniquement comme des valeurs relatives (dynamiques dans le temps) et non comme des valeurs absolues.

[4] SSL (HTTPS) est pris en charge uniquement si l'agent est compilé avec la prise en charge de cURL. Sinon, l'élément deviendra non pris en charge.

[5] Les valeurs `bytes` et `errors` ne sont pas prises en charge pour les interfaces loopback sur les systèmes Solaris jusqu'à Solaris 10 6/06 inclus, car les statistiques d'octets, d'erreurs et d'utilisation ne sont pas stockées et/ou rapportées par le noyau. Cependant, si vous surveillez un système Solaris via net-snmp, des valeurs peuvent être renvoyées, car net-snmp contient du code hérité de cmu-snmp datant d'aussi loin que 1997 qui, lorsqu'il ne parvient pas à lire les valeurs d'octets dans les statistiques de l'interface, renvoie le compteur de paquets (qui existe sur les interfaces loopback) multiplié par une valeur arbitraire de 308. Cela repose sur l'hypothèse que la longueur moyenne d'un paquet est de 308 octets, ce qui est une estimation très approximative, puisque la limite MTU sur les systèmes Solaris pour les interfaces loopback est de 8892 octets. Il ne faut pas supposer que ces valeurs sont correctes, ni même proches de la réalité. Ce sont des estimations approximatives. Zabbix agent ne fait aucune approximation, mais net-snmp renverra une valeur pour ces champs.

[6] La ligne de commande sur Solaris, obtenue depuis /proc/pid/psinfo, est limitée à 80 octets et contient la ligne de commande telle qu'elle était au moment du 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 restreindre l'accès à des fichiers spécifiques contenant des informations sensibles, exécutez Zabbix agent sous un utilisateur ne disposant pas des autorisations nécessaires 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 s'assurer que les données acquises ne sont pas corrompues, vous pouvez spécifier l'encodage correct pour le traitement de la vérification (par ex. `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 marque parfois certaines pages de codes comme « disponibles uniquement pour les applications managées », c'est-à-dire exposées uniquement via l'environnement d'exécution .NET et non disponibles via les API Win32 natives. 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 Windows natives 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 est une chaîne vide), il est supposé être UTF-8 ; les données sont traitées « telles quelles » ;
- Analyse du BOM — applicable aux éléments `vfs.file.contents`, `vfs.file.regexp`, `vfs.file.regmatch`. Une tentative est faite 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 d'agent

Dans le cas des contrôles passifs, afin d'éviter qu'un élément ne reçoive aucune valeur parce que la requête du serveur vers l'agent expire en premier, il convient de noter ce qui suit :

- 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.

1 Agent Zabbix 2

Zabbix agent 2 prend en charge toutes les clés d'élément prises en charge par 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 listées sans paramètres ni informations supplémentaires. Cliquez sur une clé d'élément pour voir tous les détails.

Clé d'élément	Description	Plugin
ceph.df.details	L'utilisation des données du cluster et leur répartition entre les pools.	Ceph
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 statuts des OSD.	
ceph.ping	Teste si une connexion à Ceph peut être établie.	
ceph.pool.discovery	La liste des pools découverts.	
ceph.status	Le statut 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 demandé.	Ember+
memcached.ping	Teste si une connexion est active ou non.	Memcached
memcached.stats	Obtient la sortie de la commande STATS.	
mongodb.collection.stats	Renvoie diverses statistiques de stockage pour une collection donnée.	MongoDB
mongodb.cfg.discovery	Renvoie une liste des serveurs de configuration découverts.	MongoDB
mongodb.collections.discovery	Renvoie une liste des collections découvertes.	
mongodb.collections.usage	Renvoie les statistiques d'utilisation des collections.	
mongodb.connpool.stats	Renvoie des informations concernant les connexions sortantes ouvertes depuis l'instance de base de données actuelle vers d'autres membres du cluster fragmenté ou du jeu 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 des bases de données découvertes.	
mongodb.jumbo_chunks.count	Renvoie le nombre de jumbo chunks.	
mongodb.oplog.stats	Renvoie l'état du jeu de réplicas à l'aide des données interrogées depuis l'oplog.	
mongodb.ping	Teste si une connexion est active ou non.	
mongodb.rs.config	Renvoie la configuration actuelle du jeu de réplicas.	
mongodb.rs.status	Renvoie l'état du jeu 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.	

Clé d'élément	Description	Plugin
mqtt.get	S'abonne à un ou plusieurs sujets spécifiques (avec jokers) du broker fourni et attend les publications.	MQTT
mssql.availability.group.get	Renvoie les groupes de disponibilité.	MSSQL
mssql.custom.query	Renvoie le résultat d'une requête personnalisée.	
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 (primaire ou secondaire), et qui sont situées sur le serveur auquel la connexion a été établie.	
mssql.mirroring.get	Renvoie les informations de mise en miroir.	
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) 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 performance.	
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 de 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 de MySQL.	
net.dns.get	Effectue une requête DNS et renvoie des informations détaillées sur les enregistrements DNS.	Network
nvml.device.count	Le nombre de périphériques GPU.	NVIDIA GPU
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 ECC de mémoire du périphérique GPU.	
nvml.device.errors.register	Informations sur les erreurs ECC de registre 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	Consommation électrique 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.	

Clé d'élément	Description	Plugin
oracle.cdb.info	Les informations sur les Container Databases (CDB).	
oracle.custom.query	Le résultat d'une requête personnalisée.	
oracle.datafiles.stats	Renvoie les statistiques des fichiers de données.	
oracle.db.discovery	Renvoie la liste des bases de données.	
oracle.fra.stats	Renvoie les statistiques de la Fast Recovery Area (FRA).	
oracle.instance.info	Les statistiques de l'instance.	
oracle.pdb.info	Les informations sur les Pluggable Databases (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 du fichier journal issues 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 des tablespaces.	
oracle.user.info	Renvoie les informations sur les utilisateurs Oracle.	
oracle.version	Renvoie la version du serveur de base de données.	
pgsql.autovacuum.count	Le nombre de workers autovacuum.	PostgreSQL
pgsql.archive	Les informations sur les fichiers archivés.	
pgsql.bgwriter	Le nombre combiné de checkpoints pour le cluster de bases de données, ventilé par type de checkpoint.	
pgsql.cache.hit	Le taux de succès du cache de tampons PostgreSQL.	
pgsql.connections	Renvoie les connexions par type.	
pgsql.custom.query	Renvoie le résultat d'une requête personnalisée.	
pgsql.db.age	L'âge du plus ancien FrozenXID 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 résumées 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'âge du plus ancien XID.	
pgsql.ping	Teste si une connexion est active ou non.	
pgsql.queries	Métriques des requêtes par temps d'exécution.	
pgsql.replication.count	Le nombre de serveurs de secours.	
pgsql.replication.process	Le retard de flush, le retard d'écriture et le retard de replay pour chaque processus émetteur.	
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 motif.	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 lent depuis le démarrage de Redis.	
smart.attribute.discovery	Renvoie une liste des attributs de périphérique S.M.A.R.T.	S.M.A.R.T.
smart.disk.discovery	Renvoie une liste des 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.	Web certificates

Voir aussi :

- Plugins intégrés
- Plugins chargeables

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 ou plusieurs sujets spécifiques (avec caractères génériques) du broker fourni et attend les publications.
 Valeur de retour : dépend du contenu du sujet. Si des caractères génériques sont utilisés, renvoie le contenu du sujet 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 est omis, le protocole par défaut (`tcp`) ou le port par défaut (1883) sera utilisé ;
- **topic** - le sujet 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/` sous 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 les enregistrements DNS.
 Cet élément est une version étendue de l'élément d'agent Zabbix `net.dns.record`, avec davantage de types d'enregistrements et la prise en charge d'indicateurs personnalisables.
 Valeurs de retour : *objet JSON*

Paramètres :

- **ip** - l'adresse IP du serveur DNS (laisser 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'expiration de la requête en secondes (la valeur par défaut est de 1 seconde) ;
- **count** - le nombre de tentatives pour la requête (la valeur par défaut est de 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 uniquement en majuscules ; 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 à la fois au format inversé et 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 uniquement en minuscules ; 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 les enregistrements DNS en fonction des paramètres fournis (voir plus de détails).

Exemples :

```
net.dns.get[192.0.2.0,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 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 ;
- **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 ;
  - 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 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 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 ;
- **destination** - le nom de la destination à interroger.

oracle.cdb.info[connString,<user>,<password>,<service>,<database>]

 Les informations sur les bases de données conteneurs (CDB).
 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 ;
    - 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 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 ;
- **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 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.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 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.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 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.

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 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.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 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 ;
  - 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.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 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 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.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 ;
  - 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.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 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.

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 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 ;
    - 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 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.redolog.info[connString,<user>,<password>,<service>]

 Les informations sur les fichiers journaux provenant du fichier de contrôle.
 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. 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 zone SGA (System Global Area).
 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 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 ;
- **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 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 ;
    - 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 ;
- **duration** - l'intervalle de capture (en secondes) des valeurs des 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 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.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 ;
  - 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 ;
- **tablespace** - le 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** - le nom du conteneur pour lequel les informations sont requises.

Si `tablespace`, `type` ou `conname` est omis, l'élément renverra les statistiques du tablespace pour tous les conteneurs correspondants (y compris les PDB et le CDB).

oracle.ts.discovery[connString,<user>,<password>,<service>]

 Renvoie une liste des 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 ;
  - 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 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.user.info[connString,<user>,<password>,<service>,<username>]

 Renvoie les informations sur l'utilisateur Oracle.
 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 ;
- **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 ;
  - 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.

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 (s'il est omis, la valeur de Plugins.PostgreSQL.Default.Database de postgresql.conf est utilisée).

pgsql.archive[uri,<username>,<password>,<database name>]

 Les 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 (s'il est omis, la valeur de Plugins.PostgreSQL.Default.Database de postgresql.conf est utilisée).

pgsql.bgwriter[uri,<username>,<password>,<database name>]

 Le nombre combiné de checkpoints pour le cluster de bases de données, ventilé par type de checkpoint.
 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.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 (s'il est omis, la valeur de Plugins.PostgreSQL.Default.Database de 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 (s'il est omis, la valeur de Plugins.PostgreSQL.Default.Database de postgresql.conf est utilisée).

pgsql.custom.query[uri,<username>,<password>,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 ;
- **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 FrozenXID le plus ancien de la 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 (s'il est 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 présentant un gonflement 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 (s'il est 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 (s'il est omis, la valeur de Plugins.PostgreSQL.Default.Database de 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 (s'il est 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 récapitulatives 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 (s'il est omis, la valeur de Plugins.PostgreSQL.Default.Database de postgresql.conf est utilisée).

pgsql.oldest.xid[uri,<username>,<password>,<database name>]

 L'âge du XID le plus ancien.
 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 interrompue (si une erreur est présente, y compris des problèmes d'AUTH 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 de postgresql.conf est utilisée).

pgsql.queries[uri,<username>,<password>,<database name>,time period]

 Métriques des requêtes par 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 (s'il est 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 (s'il est 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 (s'il est 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 de périphérique S.M.A.R.T.
 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 des périphériques S.M.A.R.T.
 Valeur de retour : *objet JSON*.

Paramètre :

- **type** - spécifie une valeur à analyser 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 suffixe dans le cas d'un RAID, par ex. : {"{#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 lecteurs HDD, SSD et NVME sont pris en charge. Les lecteurs 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 compléments requis pour la découverte RAID, par ex. : {"disk_name": "/dev/sda cciss,2"}
disk_type - contient le type de disque HDD, SSD ou NVME, par ex. : {"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 n'est pris en charge que sur la plateforme Linux ;
- Les valeurs LoadState, ActiveState et UnitFileState pour l'interface Unit sont renvoyées sous forme de texte et d'entier : `"ActiveState":{"state":1,"text":"active"}`.

systemd.unit.info[nom de l'unité,<property>,<interface>]

 Informations sur une unité systemd.
 Valeur renvoyée : *String*.

Paramètres :

- **nom de l'unité** - le 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 ex. ActiveState (par défaut), LoadState, Description) ;
- **interface** - le type d'interface de l'unité (par ex. 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 à partir d'un type d'interface spécifique, comme décrit dans l'API dbus.

Exemples :

```
systemd.unit.info["{#UNIT.NAME}"] #collect active state (active, reloading, inactive, failed, activating,
systemd.unit.info["{#UNIT.NAME}",LoadState] #collect load state info on discovered systemd units
systemd.unit.info[mysqld.service,Id] #retrieve the service technical name (mysqld.service)
systemd.unit.info[mysqld.service,Description] #retrieve the service description (MySQL Server)
systemd.unit.info[mysqld.service,ActiveEnterTimestamp] #retrieve the last time the service entered the act
systemd.unit.info[dbus.socket,NConnections,Socket] #collect the number of connections from this socket uni
```

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 un nom 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, hostname est utilisé uniquement pour le SNI et la vérification du nom d'hôte ;

- **port** - le numéro de port (443 par défaut pour HTTPS) ;
- **address** - peut être une adresse IP ou un nom DNS. S'il est spécifié, il sera utilisé pour la connexion, et hostname (le 1er paramètre) sera utilisé pour le SNI et la vérification de l'hôte. Si le 1er paramètre est une adresse IP et le 3e paramètre est un nom DNS, le 1er paramètre sera utilisé pour la connexion, et le 3e paramètre sera utilisé pour le 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, n'est pas disponible, ou si la négociation TLS échoue avec une 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 exemple, 3).
    - *serial_number* : le numéro de série du certificat.
    - *signature_algorithm* : l'algorithme utilisé pour signer le certificat (par exemple, `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 exemple, `RSA`).
    - *alternative_names* : les noms alternatifs du sujet (s'ils sont présents), sinon `null`.
  - result : contient le résultat de la validation.
    - *value* : l'état de la validation (voir les valeurs possibles ci-dessous).
    - *message* : message de validation détaillé (par exemple, `"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"
}
```

2 Agent Zabbix Windows

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 l'agent Zabbix pris en charge sous Windows et partagés avec l'agent Zabbix UNIX :

- La clé d'élément est un lien vers les détails complets de l'élément de l'agent Zabbix UNIX
- Les commentaires relatifs à 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 avec rotation. 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 surveillé avec rotation. 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 les données Modbus.	Modbus
net.dns	Vérifie si le service DNS est opérationnel. Les paramètres `ip`, `timeout` et `count` sont ignorés sous Windows, sauf lors de l'utilisation de 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 lors de l'utilisation de Zabbix agent 2.	
net.dns.record	Effectue une requête DNS. Les paramètres `ip`, `timeout` et `count` sont ignorés sous Windows, sauf lors de l'utilisation de 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 64 bits de statistiques d'interface 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 multi-octets 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 64 bits de statistiques d'interface 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 multi-octets 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 64 bits de statistiques d'interface 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. 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 au 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 collecteur est démarré sur l'agent Zabbix, 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, son 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 sous Windows à l'aide des fonctions GetComputerName() (pour **netbios**), GetComputerNameExA() (pour **fqdn**) ou gethostname() (pour **host**). Voir aussi une description plus détaillée.	
system.localtime	L'heure du 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 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 correct d'espace d'échange.	
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 du 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 du 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 la clé d'élément entière 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	Trouve 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	Obtient le contenu d'une page web.	Surveillance web
web.page.perf	Le temps de chargement d'une page web complète.	
web.page.regexp	Trouve 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 l'agent Zabbix (Zabbix agent ou Zabbix agent 2).	

Item key	Description	Item group
agent.version	La version de l'agent Zabbix.	
zabbix.stats	Renvoie à distance un ensemble de métriques internes du serveur ou du proxy Zabbix.	
zabbix.stats	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 (*Nom du journal* 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" (sur Windows Vista ou version plus récente).
- **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) ;
- **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) ou *skip* - ignorer le traitement des anciennes données (affecte uniquement les éléments nouvellement créés).

Commentaires :

- L'élément doit être configuré comme une vérification active ;
- L'agent ne peut pas envoyer les événements du journal "Forwarded events" ;
- Windows Eventing 6.0 est pris en charge ;
- La sélection 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 également les 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>]

 Le nombre de lignes dans le journal des événements Windows.
 Valeur de retour : *Entier*.

Paramètres :

- **name** - le nom du canal du journal des événements (*Nom du journal* 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 sévérité (insensible à la casse). Ce paramètre accepte une expression régulière basée sur les valeurs suivantes : "Information", "Warning", "Error", "Critical", "Verbose" (sur Windows Vista ou version plus récente).
- **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** - 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) ou *skip* - ignorer le traitement des anciennes données (affecte uniquement les éléments nouvellement créés).

Commentaires :

- L'élément doit être configuré comme une vérification active ;
- L'agent ne peut pas envoyer les événements du journal "Forwarded events" ;
- Windows Eventing 6.0 est pris en charge ;
- La sélection d'un type d'information autre que Log pour cet élément entraînera la perte de l'horodatage local, ainsi que des informations de sévérité et de source du journal ;
- Voir également les 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 le stockage de la valeur moyenne. La valeur de `interval` doit être comprise entre 1 et 900 secondes (incluses) 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 processeur), de sorte que la vérification renvoie à chaque fois une valeur moyenne pour les « interval » dernières secondes ;
- 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. `interval` doit être compris entre 1 et 900 secondes (incluses) et la valeur par défaut est 1.

Commentaires :

- Cet élément suit les chemins UNC, ce qui peut présenter un risque de sécurité ;
- `interval` est utilisé pour les compteurs qui nécessitent plus d'un échantillon (comme l'utilisation du processeur), de sorte que la vérification renvoie à chaque fois une valeur moyenne pour les « interval » dernières secondes ;
- 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 consultant la clé de registre suivante : `HKEY_LOCAL_MACHINE\SOFTWARE\Micro 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 de processus demandé ;
- **type** - le type de représentation (pertinent lorsque plusieurs processus portant le même nom existent)

Commentaires :

- Les `attributes` suivants sont pris en charge :
vmsize (par défaut) - taille de la mémoire virtuelle du processus en Ko
wkset - taille de l'ensemble de travail du processus (quantité de mémoire physique utilisée par le processus) en Ko
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 renvoyée : *Integer*, *string* ou *text* (selon le type de valeur)

Paramètres :

- **key** - la clé de registre, y compris la clé racine ; les abréviations de racine (par ex. HKLM) sont autorisées ;
- **value name** - le nom de la valeur de 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 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"] #return the data of
registry.data["HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting","EnableZip"] #return the data of t
```

registry.get[key,<mode>,<name regexp>]

 La liste des valeurs ou des clés du Registre Windows situées dans la clé donnée.
 Valeur de retour : *objet JSON*.

Paramètres :

- **key** - la clé de registre, y compris la clé racine ; les abréviations de racine (par ex. HKLM) sont autorisées (voir les commentaires de registry.data[] pour consulter la liste complète des abréviations) ;
- **mode** - valeurs possibles :
values (par défaut) ou *keys* ;
- **name regexp** - découvre uniquement les valeurs dont les noms correspondent à l'expression régulière (par défaut : découvre 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] #renvoie les données de tout
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] #renvoie 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* ; *String* - avec param comme *displayname*, *path*, *user* ; *Text* - avec param comme *description*
Plus précisément pour *state* : 0 - en cours d'exécution, 1 - en pause, 2 - démarrage en attente, 3 - mise en pause en attente, 4 - reprise en attente, 5 - arrêt en attente, 6 - arrêté, 7 - inconnu, 255 - service inexistant
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

Paramètres :

- **service** - un nom réel de service ou son nom d'affichage tel qu'il apparaît dans le composant logiciel enfichable Services de MMC ;

- **param** - *state* (par défaut), *displayname*, *path*, *user*, *startup* ou *description*.

Commentaires :

- Les éléments tels que `service.info[service,state]` et `service.info[service]` renverront les mêmes informations ;
- 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] - le 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%'] #returns status inf
```

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 MMC et en affichant les propriétés du service.
Dans l'onglet *Général*, vous devriez voir un champ appelé « Nom du service ».
La valeur indiquée est le nom que vous utiliserez lors de la configuration d'un élément de supervision.
Par exemple, si vous souhaitez superviser le service « workstation », le nom du 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.

2 Agent SNMP

Aperçu

Vous pouvez souhaiter utiliser la supervision SNMP sur des équipements 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 équipements, le serveur Zabbix doit être initialement configuré avec la prise en charge de 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 soient affichées dans le bon format. Sans les fichiers MIB, des problèmes de formatage peuvent survenir, par exemple l'affichage des valeurs en HEX au lieu de UTF-8 ou inversement.

Les vérifications SNMP sont effectuées uniquement via le protocole UDP.

Les démons serveur et proxy Zabbix 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 équipements 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 OID unique ou chaîne), 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 supervisez des équipements SNMPv3, assurez-vous que msgAuthoritativeEngineID (également connu sous le nom de snmpEngineID ou « Engine ID ») n'est jamais partagé par deux équipements. Selon la RFC 2571 (section 3.1.1.1), il doit être unique pour chaque équipement.

Warning:
La RFC3414 exige que les équipements SNMPv3 conservent leur engineBoots. Certains équipements ne le font pas, ce qui entraîne le rejet de leurs messages SNMP comme obsolètes après leur 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 afin de 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 OID) de l'élément que vous souhaitez superviser.

Pour obtenir une liste des chaînes SNMP, utilisez la commande **snmpwalk** (qui fait 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 » représente ici la version SNMP, vous pouvez également le remplacer par « 1 », afin d'indiquer SNMP version 1 sur l'appareil.

Cela devrait vous fournir une liste des chaînes SNMP et de leur dernière valeur. Si ce n'est pas le cas, il est possible que la « communauté » SNMP soit différente de la valeur standard « public », auquel cas vous devrez déterminer laquelle est utilisée.

Vous pouvez ensuite parcourir la liste jusqu'à trouver la chaîne que vous souhaitez superviser, par exemple : si vous vouliez superviser les octets entrants sur le port 3 de votre commutateur, vous utiliseriez la chaîne `IF-MIB::ifHCInOctets.3` 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 de la chaîne est le numéro du port que vous cherchez à superviser. Voir aussi : Index dynamiques.

Cela devrait vous donner quelque chose comme ceci :

```
.1.3.6.1.2.1.31.1.1.1.6.3 = Counter64: 3472126941
```

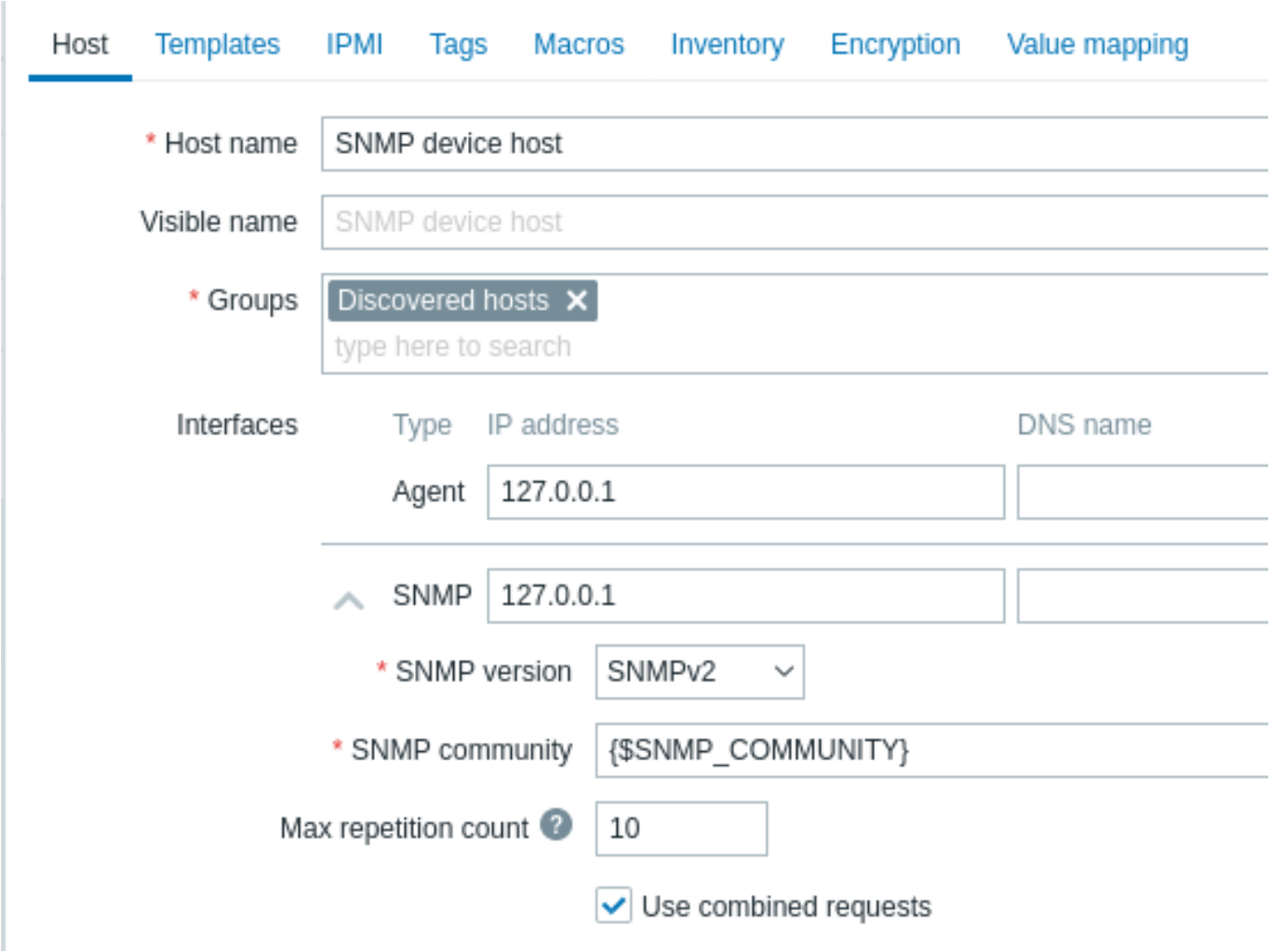
Là encore, le dernier nombre de l'OID est le numéro du port.

Note:
Certains des OID SNMP les plus utilisés sont traduits automatiquement en représentation numérique par Zabbix.

Dans le dernier exemple ci-dessus, 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'une indication d'affichage.

Étape 2

Créez un hôte correspondant à un appareil.



Ajoutez une interface SNMP pour l'hôte :

- Saisissez l'adresse IP/le nom DNS et le numéro de port.
- Sélectionnez la *version SNMP* dans la liste déroulante.
- Ajoutez les identifiants de l'interface en fonction de la version SNMP sélectionnée :
  - SNMPv1, v2 nécessitent uniquement la communauté (généralement « public »).
  - SNMPv3 nécessite des options plus spécifiques (voir ci-dessous).
- Spécifiez la valeur de répétition maximale (par défaut : 10) pour les requêtes SNMP bulk natives (GetBulkRequest-PDUs) ; uniquement pour les éléments `discovery[]` et `walk[]` en SNMPv2 et v3. Notez qu'une valeur trop élevée peut provoquer un dépassement du délai d'attente de la vérification de l'agent SNMP.
- 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 »).

Paramètre SNMPv3	Description
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.

Paramètre SNMPv3	Description
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). Voir les remarques 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 ERROR de net-snmp, sauf en cas de *phrase secrète de confidentialité* incorrecte, où Zabbix reçoit une erreur TIMEOUT de net-snmp.
- La disponibilité de l'interface SNMP passera au rouge (indisponible).

Note:
Les modifications du *protocole d'authentification*, de la *phrase secrète d'authentification*, du *protocole de confidentialité* ou de la *phrase secrète de confidentialité*, effectuées sans modifier le *nom de sécurité*, sont normalement appliquées automatiquement lorsque l'interface SNMPv3 correspondante est mise à jour dans Zabbix. Dans les cas où le *nom de sécurité* est également modifié, tous les paramètres seront mis à jour immédiatement.

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.

Prise en charge des protocoles 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 a été abandonnée dans le paquet net-snmp.
- Les protocoles de chiffrement AES192 et supérieurs ne sont pas pris en charge prêts à l'emploi 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 paquet 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.

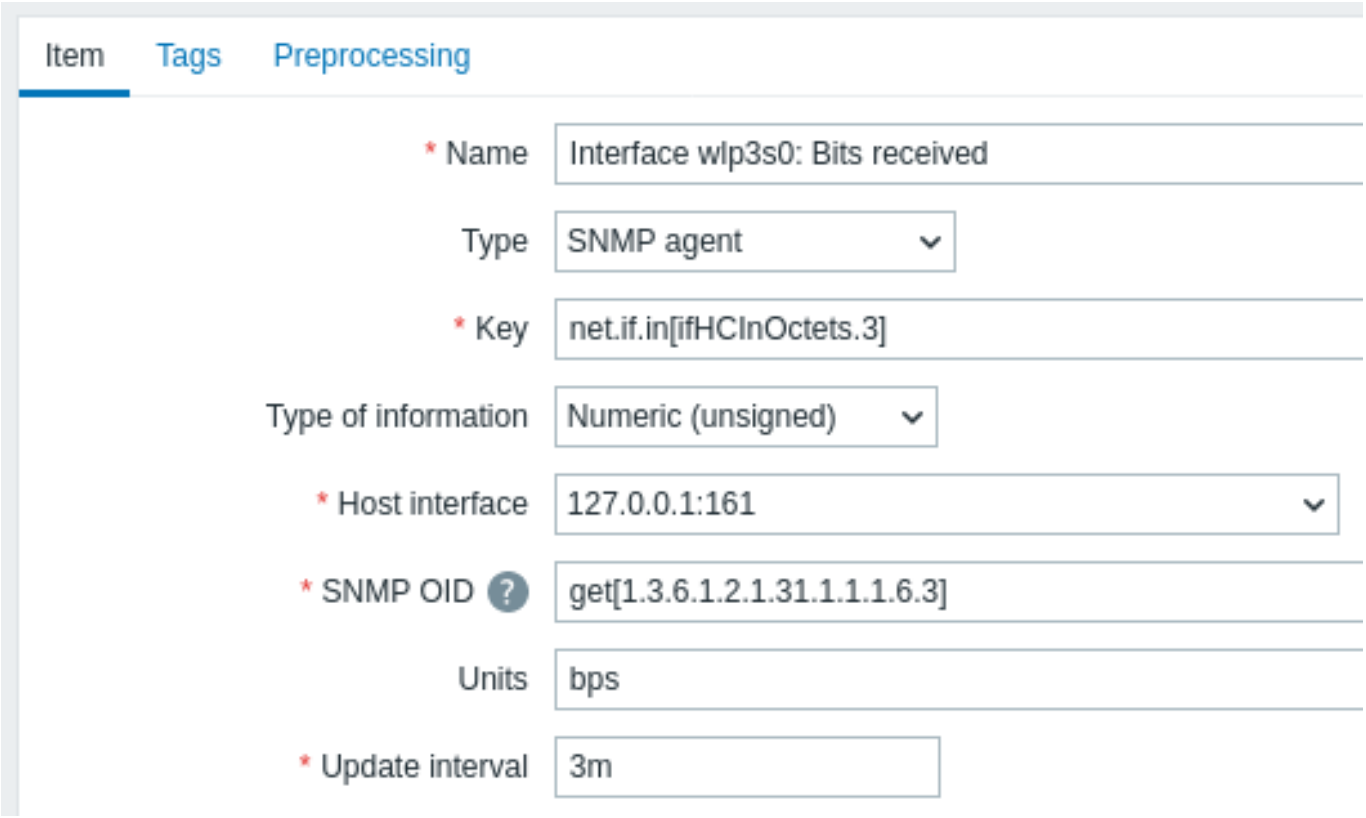
Si votre bibliothèque net-snmp ne prend pas en charge AES192 et les protocoles supérieurs, recompilez net-snmp avec l'option `--enable-blumenthal-aes`, puis recompilez le serveur Zabbix en spécifiant l'option `--with-net-snmp=/home/user/yourcustomnets`

Étape 3

Créez un élément pour la surveillance.

Retournez maintenant dans Zabbix et cliquez sur *Éléments* pour l'hôte SNMP que vous avez créé précédemment. Selon que vous avez utilisé ou non un modèle lors de la création de votre hôte, vous verrez soit une liste d'éléments SNMP associés à votre hôte, soit simplement une liste vide. Nous partirons du principe que vous allez créer l'élément vous-même à l'aide des informations que vous venez de recueillir avec snmpwalk et snmpget ; cliquez donc sur *Créer un élément*.

Renseignez les paramètres requis dans le formulaire du nouvel élément :



Paramètre	Description
Nom	Saisissez le nom de l'élément.
Type	Sélectionnez ici **Agent SNMP**.
Clé	Saisissez une clé explicite.
Interface de l'hôte	Assurez-vous de 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-PDUs) de manière asynchrone. Les paramètres de délai d'expiration pour cet élément peuvent être définis dans le formulaire de configuration de l'élément. Envisagez de définir une valeur de délai faible afin d'éviter de longs retards si l'appareil est inaccessible, car jusqu'à 5 nouvelles tentatives seront effectuées si les précédentes expirent ou échouent (par exemple, un délai de 3 secondes peut entraîner un temps d'attente de 15 secondes). Vous pouvez l'utiliser comme élément maître, avec des éléments dépendants qui extraient les données du maître à l'aide du prétraitement. Il est possible de spécifier plusieurs OID dans un seul snmp walk, par exemple `walk[OID1,OID2,...]`, afin de traiter un OID à la fois de manière asynchrone. Si la requête bulk ne renvoie aucun résultat, une tentative est effectuée pour récupérer un enregistrement unique sans requête bulk. 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 la même sortie. Si plusieurs OID/MIB sont spécifiés, c'est-à-dire `walk[ifDescr,ifType,ifPhysAddress]`, alors la sortie 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 affecte 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, réduisant le nombre de transmissions nécessaires. Cependant, tous les appareils ne prennent pas forcément en charge des valeurs très élevées, ce qui peut poser 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 pour cet élément peuvent être définis dans le formulaire de configuration de l'élément. Envisagez de définir une valeur de délai faible afin d'éviter de longs retards si l'appareil est inaccessible, car jusqu'à 5 nouvelles tentatives seront effectuées si les précédentes expirent ou échouent (par exemple, un délai de 3 secondes peut entraîner un temps d'attente de 15 secondes). **OID** - (hérité) saisissez un seul OID textuel ou numérique pour récupérer une valeur unique de manière synchrone, éventuellement combinée avec d'autres valeurs. Par exemple : `1.3.6.1.2.1.31.1.1.1.6.3`. Pour 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 concurrence 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 ou l'autre des méthodes, une étape *Variation par seconde* doit être ajoutée dans l'onglet *Prétraitement* ; sinon, vous obtiendrez la valeur cumulée de l'appareil SNMP au lieu de la dernière variation.

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

Enregistrez maintenant l'élément et allez dans *Surveillance* > *Dernières données* pour 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 groupées natives

L'élément **walk[OID1,OID2,...]** permet d'utiliser la fonctionnalité SNMP native pour les requêtes groupées (GetBulkRequest-PDUs), disponible dans les versions 2/3 de SNMP.

Une requête GetBulk en SNMP exécute plusieurs requêtes GetNext et renvoie le résultat dans une seule réponse. Cela peut être utilisé pour les éléments SNMP classiques ainsi 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 qui collecte 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 groupées natives n'est pas liée à l'option de combinaison des requêtes SNMP, qui est 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 groupés afin d'éviter un échec si l'un des paquets est perdu. Le délai d'expiration des éléments SNMP avec `get` et `walk` (défini dans le formulaire de configuration de l'élément) s'applique à l'ensemble d'une session. Le délai d'expiration s'applique indépendamment du fait que les données soient récupérées complètement ; si les données ne sont reçues que partiellement (par exemple, si les données ne sont collectées avec succès que pour un seul des multiples OID), alors l'élément devient non pris en charge avec le message « Only partial data received ». Si le délai d'expiration est atteint, une nouvelle tentative aura lieu, le délai sera réinitialisé et la dernière requête sera renvoyée, ce qui permettra de poursuivre la session 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 retards si l'appareil 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 et le proxy Zabbix peuvent interroger des périphériques SNMP pour plusieurs valeurs dans une seule requête. Cela affecte plusieurs types d'éléments SNMP :

- éléments SNMP classiques
- éléments SNMP avec index dynamiques
- règles de découverte de bas niveau SNMP

Tous les éléments SNMP sur une même interface avec des paramètres identiques sont planifiés pour être interrogés en même temps. 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, il existe deux types d'opérations effectuées pour interroger les valeurs : l'obtention de plusieurs objets spécifiés et le parcours d'un arbre OID.

Pour « l'obtention », un GetRequest-PDU est utilisé avec au maximum 128 liaisons de variables. Pour le « parcours », un GetNextRequest-PDU est utilisé pour SNMPv1, et GetBulkRequest avec un champ « max-repetitions » de 128 au maximum est utilisé pour SNMPv2 et SNMPv3.

Ainsi, les avantages du traitement combiné pour chaque type d'élément SNMP sont décrits ci-dessous :

- les éléments SNMP classiques bénéficient des améliorations de « l'obtention » ;
- les éléments SNMP avec index dynamiques bénéficient à la fois des améliorations de « l'obtention » et du « parcours » : « l'obtention » est utilisée pour la vérification des index et le « parcours » pour construire le cache ;
- les règles de découverte de bas niveau SNMP bénéficient des améliorations du « parcours ».

Cependant, il existe un problème technique : tous les périphériques ne sont pas capables de renvoyer 128 valeurs par requête. Certains renvoient toujours une réponse correcte, mais d'autres répondent soit avec une erreur « tooBig(1) », soit ne répondent pas du tout dès que la réponse potentielle dépasse une certaine limite.

Afin de trouver un nombre optimal d'objets à interroger pour un périphérique donné, Zabbix utilise la stratégie suivante. Il commence prudemment en interrogeant 1 valeur dans une requête. Si cela réussit, il interroge 2 valeurs dans une requête. Si cela réussit à nouveau, il interroge 3 valeurs dans une requête et continue de manière similaire en multipliant le nombre d'objets interrogés par 1,5, ce qui donne la séquence suivante de tailles de requête : 1, 2, 3, 4, 6, 9, 13, 19, 28, 42, 63, 94, 128.

Cependant, dès qu'un périphérique refuse de fournir une réponse correcte (par exemple, pour 42 variables), Zabbix fait deux choses.

Premièrement, pour le lot d'éléments en cours, il divise par deux le nombre d'objets dans une seule requête et interroge 21 variables. Si le périphérique est joignable, alors la requête devrait fonctionner dans la grande majorité des cas, car 28 variables étaient connues pour fonctionner et 21 est nettement inférieur à cette valeur. Cependant, si cela échoue encore, Zabbix revient à l'interrogation des valeurs une par une. Si cela échoue encore à ce stade, alors le périphérique ne répond définitivement pas et la taille de la requête n'est pas en cause.

La deuxième chose que fait Zabbix pour les lots d'éléments suivants est de commencer avec le dernier nombre de variables ayant réussi (28 dans notre exemple) et de continuer à incrémenter les tailles de requête de 1 jusqu'à atteindre la limite. Par exemple, en supposant que la plus grande taille de réponse soit de 32 variables, les requêtes suivantes auront des tailles de 29, 30, 31, 32 et 33. La dernière requête échouera et Zabbix n'émettra plus jamais de requête de taille 33. À partir de ce moment, Zabbix interrogera au maximum 32 variables pour ce périphérique.

Si de grandes requêtes échouent avec ce nombre de variables, cela peut signifier l'une des deux choses suivantes. Les critères exacts qu'un périphérique utilise pour limiter la taille de réponse 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, parfois elle la dépasse. La deuxième possibilité est qu'un paquet UDP dans un sens ou dans l'autre ait simplement été perdu. Pour ces raisons, si Zabbix obtient une requête échouée, il réduit le nombre maximal de variables afin d'essayer de revenir plus profondément dans la plage de fonctionnement confortable du périphérique, mais seulement jusqu'à deux fois.

Dans l'exemple ci-dessus, si une requête avec 32 variables échoue, Zabbix réduira le nombre à 31. Si cela échoue également, Zabbix réduira le nombre à 30. Cependant, Zabbix ne réduira pas le nombre en dessous de 30, car il supposera que les échecs supplémentaires sont dus à des paquets UDP perdus, 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, il existe un paramètre « Use combined requests » pour chaque interface qui permet de désactiver les requêtes combinées pour ce périphérique.

Si les requêtes combinées provoquent des réponses partielles ou mal formées 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 séparées par élément ; cela évite souvent les faux pics. Vous pouvez également envisager d'utiliser des éléments asynchrones `get[]` ou `walk[]`, qui sont exécutés de manière asynchrone et ne sont pas soumis au regroupement par interface de *Use combined requests* — 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 affichée dans la section Vue d'ensemble pour aider à identifier les périphériques concernés.

En outre, 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 OID.

1 Index dynamiques

Aperçu

Bien que vous puissiez trouver le numéro d'index nécessaire (par exemple, d'une interface réseau) parmi les OID SNMP, il se peut que vous ne puissiez pas toujours compter sur le numéro d'index qui ne reste pas toujours identique.

Les numéros d'index peuvent être dynamiques - ils peuvent changer au fil du temps et votre élément peut, en conséquence, cesser de fonctionner.

Pour éviter ce scénario, il est possible de définir un OID qui prend en compte la possibilité de changement du numéro d'index.

Par exemple, si vous devez extraire la valeur d'index pour l'ajouter à **ifInOctets** correspondant à l'interface **GigabitEthernet0/1** sur un périphérique Cisco, utilisez l'OID :

```
ifInOctets["index","ifDescr","GigabitEthernet0/1"]
```

Syntaxe

Une syntaxe particulière pour les OID est utlisée :

<OID of data>["index","<base OID of index>","<string to search for>"]

Paramètre	Description
OID of data	OID principal à utiliser pour la récupération de données sur l'élément.
index	Méthode de traitement. Actuellement, une méthode est supportée : **index** – recherche index et ajoute à l'OID
base OID of index	Cet OID sera recherché pour obtenir la valeur d'index correspondant à la chaîne.
string to search for	Chaîne à utiliser pour une correspondance exacte avec une valeur lors de la recherche. Sensible aux majuscules et minuscules.

Exemple

Obtenir l'utilisation de la mémoire du processus *apache*.

Si vous utilisez cette syntaxe d'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"
...
```

Maintenant que nous avons 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 toute la table SNMP sous l'OID de base pour l'indexation, même si une correspondance a été trouvée plus tôt. Ceci est fait dans le cas où un autre élément se rapporterait au même OID de base plus tard - Zabbix rechercherait l'index dans le cache, au lieu d'interroger à nouveau l'hôte surveillé. Notez que chaque processus poller utilise un cache distinct.

Dans toutes les opérations de récupération de valeur ultérieures, seul l'index trouvé est vérifié. Si cela n'a pas changé, la valeur est demandée. Si elle a été modifiée, le cache est reconstruit. Chaque poller qui rencontre un index modifié parcourt à nouveau la table d'index SNMP.

2 OID spéciaux

Certains des OID SNMP les plus utilisés sont traduits automatiquement 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 suivant contient la liste des OID spéciaux.

OID Spéciaux	Identificateur	Description
ifIndex	1.3.6.1.2.1.2.2.1.1	Une valeur unique pour chaque interface.
ifDescr	1.3.6.1.2.1.2.2.1.2	Une chaîne textuelle 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 en fonction du (des) protocole(s) physique(s) / de liaison immédiatement "en dessous" de la couche réseau dans la pile de protocole.
ifMtu	1.3.6.1.2.1.2.2.1.4	La taille du plus grand datagramme pouvant être envoyé/reçu sur l'interface, spécifié 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 est immédiatement inférieure à 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	Nombre total d'octets reçus sur l'interface, y compris les caractères de cadre.
ifInUcastPkts	1.3.6.1.2.1.2.2.1.11	Le nombre de paquets sub-network-unicast livrés à un protocole de couche supérieure.
ifInNUcastPkts	1.3.6.1.2.1.2.2.1.12	Nombre de paquets non monodiffusion (c.-à-d., Sous-réseau diffusé ou sous-réseau-multidiffusion) transmis 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 rejetés même si aucune erreur n'a été détectée pour empêcher leur transmission à un protocole de couche supérieure. Une raison possible pour éliminer un tel paquet pourrait être de libérer de l'espace buffer.
ifInErrors	1.3.6.1.2.1.2.2.1.14	Nombre de paquets entrants contenant des erreurs empêchant leur transmission à un protocole de couche supérieure.
ifInUnknownProtos	1.3.6.1.2.1.2.2.1.15	Nombre de paquets reçus via l'interface qui ont été rejetés en raison d'un protocole inconnu ou non supporté.
ifOutOctets	1.3.6.1.2.1.2.2.1.16	Nombre total d'octets transmis hors de l'interface, y compris les caractères de cadre.
ifOutUcastPkts	1.3.6.1.2.1.2.2.1.17	Nombre total de paquets que les protocoles de niveau supérieur requis ont transmis et qui n'ont pas été adressés à une adresse de multidiffusion ou de diffusion de cette sous-couche, y compris ceux qui ont été rejetés ou non envoyés.

OID Spéciaux	Identificateur	Description
ifOutNUcastPkts	1.3.6.1.2.1.2.2.1.18	Nombre total de paquets que les protocoles de niveau supérieur requis ont transmis et qui ont été adressés à une adresse de multidiffusion ou de diffusion de cette sous-couche, y compris ceux qui ont été rejetés 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 rejetés même si aucune erreur n'a été détectée pour empêcher leur transmission. Une raison possible pour éliminer un tel paquet pourrait être de libérer de l'espace buffer.
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 de paquets de sortie (en paquets).

3 Fichiers MIB

Introduction

MIB signifie Management Information Base. Les fichiers MIB permettent d'utiliser une représentation textuelle d'un OID (Object Identifier). Il est possible d'utiliser des OID bruts lors de la supervision d'appareils 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.

4 Vérifications IPMI

Aperçu

Vous pouvez surveiller l'état de santé et la disponibilité des périphériques Intelligent Platform Management Interface (IPMI) dans Zabbix. Pour effectuer des vérifications IPMI, le serveur Zabbix doit d'abord être configuré avec la prise en charge d'IPMI.

IPMI est une interface normalisée pour la gestion à distance « lights-out » ou « out-of-band » des systèmes informatiques. Elle permet de surveiller directement l'état du matériel à partir des cartes de gestion dites « out-of-band », indépendamment du système d'exploitation et même si la machine est hors tension.

La supervision IPMI de Zabbix fonctionne uniquement pour les périphériques prenant en charge IPMI (HP iLO, DELL DRAC, IBM RSA, Sun SSP, etc.).

Un processus gestionnaire IPMI planifie les vérifications IPMI effectuées par les collecteurs IPMI. Un hôte est toujours interrogé par un seul collecteur IPMI à la fois, ce qui réduit le nombre de connexions ouvertes aux contrôleurs BMC. Il est donc possible d'augmenter en toute sécurité le nombre de collecteurs IPMI sans craindre de surcharger les contrôleurs BMC. Le processus gestionnaire IPMI est démarré automatiquement lorsqu'au moins un collecteur IPMI est démarré.

Voir aussi 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 les pollers IPMI. Par conséquent, les éléments IPMI ajoutés ne fonctionneront pas. Pour changer cela, ouvrez le fichier de configuration du serveur Zabbix (zabbix_server.conf) en tant qu'utilisateur root et recherchez la ligne suivante :

```
# StartIPMIPollers=0
```

Décommentez-la et définissez le nombre d'interrogation sur, disons, 3, afin qu'il indique :

```
StartIPMIPollers=3
```

Enregistrez le fichier et redémarrez ensuite 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.

5 Vérifications simples

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 répertoriées sans les paramètres optionnels ni les informations supplémentaires. Cliquez sur la clé d'élément pour voir tous les détails.

Voir aussi les clés d'élément de supervision VMware.

Clé d'élément	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 nouvelles tentatives.
icmppingsec	Le temps de réponse au 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* - échec du ping ICMP ; *1* - ping ICMP réussi.

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.

Exemple :

```
icmpping[,4] #Si au moins un paquet sur les quatre est renvoyé, l'élément retournera 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 simples LDAP (par ex. `net.tcp.service[ldap]`), OpenLDAP version 2.6.1 ou supérieure 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 pouvoir être exécuté 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 des versions de fping antérieures à 3.10.

Valeurs par défaut

Valeurs par défaut, limites et description des valeurs pour les paramètres de vérification ICMP :

Paramètre	Unité	Description	Option 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 des paquets successifs vers une cible individuelle	-p	1000		20	illimité
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'option *-C* **fping v4.x** - délai d'attente individuel pour chaque paquet	-t	**fping v3.x** - 500 **fping v4.x** et versions plus récentes - hérité de l'option *-p*, mais pas supérieur à 2000		50	illimité
retries	nombre	Nombre de tentatives de ping d'une cible, sans compter le premier essai	-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 varier 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 adresse IP source* (ou *-I* dans les anciennes versions de fping). Ces options sont détectées automatiquement en exécutant des vérifications avec différentes combinaisons d'options. Zabbix essaie de détecter la valeur minimale en millisecondes que fping autorise avec *-i* en testant 3 valeurs : 0, 1 et 10. La première valeur qui réussit 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 prochaine tentative d'exécution d'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 par l'une des clés *icmpping** dans un fichier temporaire, qui est ensuite transmis à fping. Si des éléments ont 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 un même fichier seront vérifiées par fping en parallèle, ainsi le processus ICMP pinger de Zabbix passera un temps fixe, indépendamment du nombre d'adresses IP dans le fichier.

1 Clés d'élément de supervision VMware

La liste des clés d'élément de la surveillance VMware a été déplacée vers la section Surveillance VMware.

6 Surveillance des fichiers journaux

Aperçu

Zabbix peut être utilisé pour la surveillance centralisée et l'analyse des 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 :

- l'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 fichiers volumineux.

Configuration

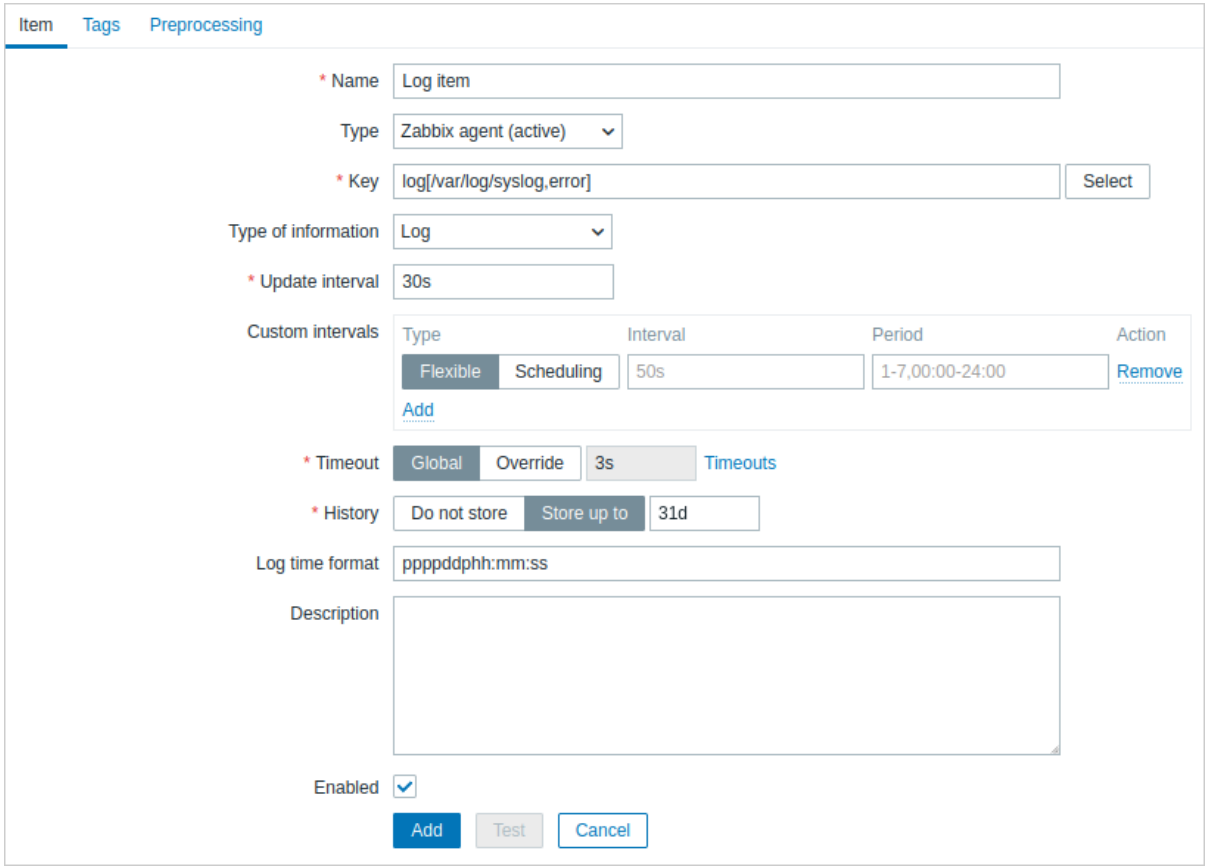
Vérifier les paramètres de l'agent

Assurez-vous que dans le fichier de configuration de l'agent :

- le paramètre `Hostname` correspond au nom d'hôte dans le frontend.
- les serveurs du paramètre `ServerActive` sont spécifiés pour le traitement des contrôles actifs.

Configuration de l'élément

Configurez un item de surveillance des journaux.



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

Plus précisément, pour les éléments de surveillance des journaux, vous devez saisir :

Type	Sélectionnez **Zabbix agent (active)** ici.
Key	Utilisez l'une des clés d'élément suivantes : **log[]** ou **logrt[]** : Ces deux clés d'élément permettent de surveiller les journaux et de filtrer les entrées de journal par le contenu regexp, s'il est présent. Par exemple : `log[/var/log/syslog,error]`. Assurez-vous que le fichier dispose des autorisations de lecture pour l'utilisateur « zabbix », sinon le statut de l'élément sera défini sur « unsupported ». **log.count[]** ou **logrt.count[]** : Ces deux clés d'élément permettent de renvoyer uniquement le nombre de lignes correspondantes. Consultez la section des clés Zabbix agent item prises en charge pour plus de détails 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 le 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 vérifiera les éventuelles modifications du fichier journal. Le définir à 1 seconde garantit que vous recevrez 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 à 0 en temps Unix, représentant le 1er janvier 1970. Par exemple, considérez 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 d'heure du journal pour cette ligne serait "pppppp:yyyyMMdd:hhmmss". Notez que les caractères « p » et « : » sont des espaces réservés et peuvent être n'importe quels caractères sauf « yMdhms ».

Remarques 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. En outre :
  - L'agent utilise également en interne les numéros d'inode (sur UNIX/GNU/Linux), les index de fichier (sur Microsoft Windows) et les 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, on suppose que les systèmes de fichiers où sont stockés les fichiers journaux rapportent 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 résident les fichiers journaux et utilise :
    * Sur les systèmes de fichiers NTFS, des index de fichier sur 64 bits.
    * Sur les systèmes de fichiers ReFS (uniquement à partir de Microsoft Windows Server 2012), des ID de fichier sur 128 bits.
    * Sur les systèmes de fichiers où les index de fichier changent (par ex. FAT32, exFAT), un algorithme de repli est utilisé afin d'adopter une approche raisonnable dans des conditions incertaines lorsque la rotation des fichiers journaux produit plusieurs fichiers journaux avec la même heure de dernière modification.
  - Les numéros d'inode, les index de fichier 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 recopiant un fichier sous son nom d'origine (cela crée un nouvel inode). Dans l'un ou

l'autre cas, Zabbix peut traiter 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 correspondants pour l'élément `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 inférieure à l'heure de modification la plus récente vue 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 l'heure de dernière modification (le compteur de temps) sont stockés dans la base de données Zabbix et envoyés à l'agent pour s'assurer que l'agent commence à lire le fichier journal à partir de ce point lorsque l'agent vient juste de démarrer 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 des 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 par 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 dans le répertoire plusieurs fichiers correspondants ayant la même heure de dernière modification, l'agent essaie 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 confronté à 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 le signalement des enregistrements de fichiers journaux correspondants dans un ordre modifié (ce 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** d'un fichier journal par seconde. Cette limite empêche la surcharge des 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 n'analysera pas plus de 200 enregistrements de fichier journal et n'enverra pas plus de 20 enregistrements correspondants au serveur Zabbix lors d'une 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 lors d'une vérification. Si *Update interval* est défini sur 2 secondes, les limites pour une vérification seront 2 fois plus élevées qu'avec un *Update interval* de 1 seconde.
- En outre, 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 pas de valeurs non-log dedans. Ainsi, pour que les valeurs **maxlines** soient envoyées en une seule connexion (et non en plusieurs connexions), 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 ainsi libérer le tampon, tandis que l'agent Zabbix 2 arrêtera 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-log. Lorsque des valeurs de journal arrivent, elles remplacent les anciennes valeurs non-log selon les besoins, jusqu'aux 50 % désignés.
- Pour les enregistrements de fichier journal de plus de 256kB, seuls les 256kB premiers 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 spéciale concernant 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` ne sont prises en charge que 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 devient pas NOTSUPPORTED (par exemple, si le nom du 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 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 lorsque DebugLevel=4 ou DebugLevel=5.
- La recherche d'un point d'interrogation à l'aide d'une expression régulière, par ex. \?, peut produire 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 utile d'extraire uniquement la valeur intéressante d'un fichier cible au lieu de renvoyer toute la ligne lorsqu'une correspondance avec une expression régulière est trouvée.

Les éléments de journal ont la capacité d'extraire les valeurs souhaitées à partir des lignes correspondantes. Cela est réalisé grâce au paramètre supplémentaire **output** dans les éléments `log` et `logrt`.

L'utilisation du paramètre « output » permet d'indiquer le « groupe de capture » de la correspondance qui nous intéresse.

Ainsi, par exemple

```
log[/path/to/the/file,"large result buffer allocation.*Entries: ([0-9]+)",,,,\1]
```

doit permettre de renvoyer le nombre d'entrées tel qu'il apparaît 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 de capture : **([0-9]+)**.

Et, 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 d'analyser les lignes les plus récentes dans le délai de `maxdelay` secondes.

> **Warning:**
> La spécification de `maxdelay` > 0 peut conduire à **ignorer des enregistrements importants du fichier journal et à manquer des alertes**. Utilisez-le avec précaution, à vos propres risques, 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 très grand nombre de messages dans leurs fichiers journaux. Par exemple, si une base de données ou un serveur DNS est indisponible, de telles applications inondent les fichiers journaux de milliers de messages d'erreur presque identiques jusqu'au rétablissement du fonctionnement normal. Par défaut, tous ces messages seront consciencieusement analysés 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 `maxlines` configurable (protège le serveur contre un trop grand nombre de lignes de journal correspondantes entrantes) et une limite de 10*`maxlines` (protège le CPU et les E/S de l'hôte contre une surcharge causée par l'agent lors d'une vérification). Il subsiste néanmoins 2 problèmes avec la protection intégrée. Premièrement, un grand nombre de messages potentiellement peu informatifs sont signalés au serveur et consomment de l'espace dans la base de données. Deuxièmement, en raison du nombre limité de lignes analysées par seconde, l'agent peut prendre des heures de retard par rapport aux enregistrements les plus récents du journal. 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, à 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 retard estimé : le nombre de secondes qu'il faudrait pour analyser tous les enregistrements restants dans un fichier journal.

Si le retard ne dépasse pas `maxdelay`, l'agent continue alors à analyser le fichier journal comme d'habitude.

Si le retard est supérieur à `maxdelay`, alors l'agent **ignore une partie d'un fichier journal en la « sautant »** vers une nouvelle position estimée afin que les lignes restantes puissent être analysées dans le délai de `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 le 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
```

Le nombre « to byte » est approximatif, 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 comparaison entre la vitesse de croissance et la vitesse d'analyse du fichier journal, vous pouvez ne voir aucun « saut », des « sauts » rares ou fréquents, des « sauts » grands ou petits, voire un petit « saut » à chaque vérification. Les fluctuations de la charge du système et de la latence réseau affectent également le calcul du retard et, par conséquent, le « saut » en avant pour respecter le paramètre `maxdelay`.

Il n'est pas recommandé de définir `maxdelay < update interval` (cela peut entraîner de fréquents petits « sauts »).

Remarques 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 leurs horodatages sont différents), par conséquent les sommes MD5 des blocs initiaux (jusqu'aux 512 premiers octets) seront différentes. Deux fichiers ayant les mêmes sommes MD5 des 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, il n'est pas recommandé de produire plusieurs copies de fichiers journaux avec le même horodatage, d'effectuer une rotation des fichiers journaux plus fréquemment que l'intervalle de mise à jour de l'élément logrt[], ni de redémarrer fréquemment l'agent. L'agent essaie de gérer toutes ces situations de manière raisonnablement correcte, mais de bons résultats ne peuvent pas être garantis dans toutes les circonstances.

Remarques sur les fichiers persistants pour les éléments log*[]

Objectif des fichiers persistants

Lorsque l'agent Zabbix est démarré, il reçoit une liste de contrôles actifs du serveur Zabbix ou du proxy. Pour les métriques log*[], il reçoit la taille du journal traitée et l'heure de modification afin de déterminer à partir de quel point commencer la surveillance du fichier journal. En fonction de la taille réelle du fichier journal et de 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 traitée, soit de réanalyser le fichier journal depuis le début.

Un agent en cours d'exécution conserve un ensemble plus important d'attributs pour suivre tous les fichiers journaux surveillés entre les contrôles. Cet état en mémoire est perdu lorsque l'agent est arrêté.

Le nouveau paramètre optionnel **persistent_dir** spécifie un répertoire pour 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 de recevoir de nouveaux enregistrements. L'agent Zabbix l'analyse et envoie au serveur la taille des journaux traités et l'heure de modification. 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 traitée et l'heure de modification que l'agent Zabbix reçoit du serveur peuvent ne pas être valides dans la situation de la copie passive. Pour poursuivre la surveillance du fichier journal à partir de 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 contrôles actifs du serveur Zabbix (proxy) que l'agent voit que certains éléments de journal doivent être sauvegardés par des fichiers persistants dans les 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. La probabilité de perdre les données du fichier persistant si l'écrasement et la séparation du miroir du système de fichiers se produisent au même moment est très faible, aucune gestion particulière n'est prévue pour cela. L'écriture dans le fichier persistant n'est PAS suivie d'une synchronisation forcée vers le support de stockage (fsync() n'est pas appelé).

L'écrasement avec les données les plus récentes est effectué après le signalement réussi de l'enregistrement de fichier journal correspondant ou des métadonnées (taille du journal traitée et heure de modification) au serveur Zabbix. Cela peut se produire aussi souvent qu'à chaque vérification d'élément si le fichier journal continue de changer.

Aucune action particulière lors de l'arrêt de l'agent.

Après avoir reçu une liste de contrôles actifs, l'agent marque les fichiers persistants obsolètes en vue de leur 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 utilisé auparavant.

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 contrôles actifs, mais ils peuvent redevenir SUPPORTED plus tard et leurs fichiers persistants seront alors 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 de la part du serveur Zabbix.

Warning:
Reconfigurer **persistent_dir** d'un élément de journal vers l'ancien emplacement **persistent_dir** pendant que l'agent est arrêté, sans que l'utilisateur supprime l'ancien fichier persistant, 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/test.log,] sont des éléments différents. La modification de l'élément logrt[/home/zabbix/test.log„,10] dans le frontend en logrt[/home/zabbix/test.log„,20] entraînera 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 le frontend/serveur, mais pas dans l'agent).

Le nom du fichier est composé de la somme MD5 de la clé de l'élément, avec la longueur de la clé ajoutée 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 des dispositions spécifiques du système 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 les droits d'accès de l'agent Zabbix ne permettent pas de créer/écrire/lire/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 alors consignées dans le fichier journal de l'agent, mais l'élément 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 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 2e lot, et le fichier persistant sera de nouveau mis à jour.

Actions en cas d'échec de la communication entre l'agent et le serveur

Chaque ligne correspondante d'un élément `log[]` et `logrt[]`, ainsi que chaque résultat d'une vérification d'élément `log.count[]` et `logrt.count[]`, nécessite un emplacement libre dans la zone réservée 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 de nouveau libérés.

Tant qu'il reste des emplacements libres dans la zone réservée aux journaux du tampon d'envoi de l'agent et que la communication entre l'agent et le serveur (ou le proxy) échoue, les résultats de surveillance des journaux sont accumulés dans le tampon d'envoi. Cela permet d'atténuer les courtes interruptions de communication.

En cas d'interruptions de communication plus longues, tous les emplacements réservés aux 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, elles sont simplement signalées 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[]` décrit 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 il n'y a pas d'emplacements libres 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 la croissance d'un fichier journal même en cas d'échec de la communication.

Gestion de la compilation des expressions régulières et des erreurs d'exécution

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, alors l'élément passe à l'état NOTSUPPORTED 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 à l'exécution (sur certains ou sur tous les enregistrements du journal), alors l'élément de journal reste pris en charge et la surveillance continue. L'erreur d'exécution est consignée dans le fichier journal de l'agent Zabbix (sans l'enregistrement du fichier journal).

Le taux de journalisation est limité 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 avec une erreur d'exécution d'expression régulière, un enregistrement est produit dans le journal de l'agent.

Exception : si `MaxLinesPerSecond=1` et intervalle de mise à jour=1 (un seul enregistrement est autorisé à être analysé par vérification), alors les erreurs d'exécution d'expression régulière ne sont pas consignées.

zabbix_agentd consigne la clé de l'élément en cas d'erreur d'exécution, zabbix_agent2 consigne l'ID de l'élément afin d'aider à identifier quel élément de journal présente des erreurs d'exécution. Il est recommandé de repenser l'expression régulière en cas d'erreurs d'exécution.

7 Éléments calculés

Aperçu

Un élément calculé permet de créer un calcul basé sur les valeurs de certains éléments existants. Par exemple, vous pouvez vouloir calculer la moyenne horaire de la valeur d'un élément ou calculer la valeur totale pour un groupe d'éléments. C'est précisément à cela que sert un élément calculé.

Les calculs peuvent utiliser à la fois :

- des valeurs uniques d'éléments individuels
- un filtre complexe pour sélectionner plusieurs éléments à agréger (voir calculs agrégés pour plus de détails)

Les éléments calculés constituent un moyen de créer des sources de données virtuelles. Tous les calculs sont effectués uniquement par le serveur Zabbix. Les valeurs sont calculées périodiquement sur la base de l'expression arithmétique utilisée.

Les données résultantes sont stockées dans la base de données Zabbix comme pour tout autre élément ; les valeurs d'historique et de tendance sont toutes deux stockées et des graphiques peuvent être générés.

Note:
Si le résultat du calcul est une valeur flottante, il sera tronqué en entier si le type d'information de l'élément calculé est *Numérique (non signé)*.
De plus, 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.

Les éléments calculés partagent leur syntaxe avec les expressions de déclencheur. Si vous connaissez les expressions de déclencheur, vous pouvez considérer les éléments calculés comme des fonctions analysant certaines valeurs, mais sans comparaison à un seuil.

La comparaison avec des chaînes est autorisée dans les éléments calculés. Les éléments calculés peuvent être référencés par des macros ou d'autres entités comme n'importe quel autre type d'élément.

Pour utiliser des éléments calculés, choisissez le type d'élément **Calculated**.

Champs configurables

La **clé** est un identifiant unique d'élément (par hôte). Vous pouvez créer n'importe quel nom de clé en utilisant les symboles pris en charge.

La définition du calcul doit être saisie dans le champ **Formule**. Il n'y a aucun lien entre la formule et la clé. Les paramètres de la clé ne sont en aucun cas utilisés dans la formule.

La syntaxe d'une formule simple est la suivante :

```
function(/host/key,<parameter1>,<parameter2>,...)
```

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.
`parameter(s)`	Paramètres de la fonction, si nécessaire. Les suffixes de temps et les suffixes de taille mémoire sont pris en charge.

Attention:
Les macros utilisateur dans la formule seront développées si elles sont utilisées pour référencer un paramètre de fonction, un paramètre de filtre d'élément ou une constante. Les macros utilisateur ne seront PAS développées si elles référencent une fonction, un nom d'hôte, une clé d'élément, un paramètre de clé d'élément ou un opérateur.

Une formule plus complexe peut utiliser une combinaison de fonctions, d'opérateurs et de parenthèses. Vous pouvez utiliser toutes les fonctions et tous les opérateurs pris en charge dans les expressions de déclencheur. La logique et la priorité des opérateurs sont exactement les mêmes.

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.

Tous les éléments référencés par des fonctions d'historique dans la formule de l'élément calculé doivent exister et collecter des données. De plus, si vous modifiez la clé d'élément d'un élément référencé, vous devez mettre à jour manuellement toutes les formules utilisant cette clé.

Un élément calculé peut devenir non pris en charge dans plusieurs cas :

- élément(s) référencé(s)
  - introuvable(s)
  - désactivé(s)
  - 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

Exemples d'utilisation

Exemple 1

Calcul du pourcentage d'espace disque libre sur '/'.

Utilisation de la fonction **last**:

```
100*last(//vfs.fs.size[/,free])/last(//vfs.fs.size[/,total])
```

Zabbix prendra les dernières valeurs pour les espaces disque libres et totaux et calculera le pourcentage selon la formule donnée.

Exemple 2

Calcul d'une moyenne sur 10 minutes du nombre de valeurs traitées par Zabbix.

Utilisation de la fonction **avg**:

```
avg(/Zabbix Server/zabbix[wcache,values],10m)
```

Notez qu'une utilisation intensive d'éléments calculés avec de longues périodes peut affecter les performances du serveur Zabbix.

Exemple 3

Calcul de la bande passante totale sur eth0.

Somme de deux fonctions :

```
last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes])
```

Exemple 4

Calcul du pourcentage du trafic entrant.

Expression plus complexe :

```
100*last(//net.if.in[eth0,bytes])/(last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes]))
```

Voir également : Exemples de calculs agrégés

1 Calculs agrégés

Vue d'ensemble

Les calculs agrégés sont un type d'élément calculé permettant de collecter des informations à partir de plusieurs éléments par le serveur Zabbix, puis de calculer une valeur agrégée, selon la fonction d'agrégation utilisée.

Les calculs agrégés ne nécessitent aucun agent en cours d'exécution sur l'hôte surveillé.

Syntaxe

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 afin de 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 les 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.

Note:
Si l'agrégat produit une valeur flottante, elle sera tronquée en entier si le type d'information de l'élément agrégé est *Numérique (non signé)*.

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 d'agrégat peut devenir non pris en charge si :

- aucun des éléments référencés n'est trouvé (ce qui peut arriver 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'utilisation

Exemples de clés pour les calculs agrégés.

Exemple 1

Espace disque total du groupe d'hôtes « MySQL Servers ».

```
sum(last_foreach(/*/vfs.fs.size[/,total]?[group="MySQL Servers"]))
```

Exemple 2

Somme des dernières valeurs de tous les éléments correspondant à net.if.in[*] sur l'hôte.

```
sum(last_foreach(/host/net.if.in[*]))
```

Exemple 3

Charge processeur moyenne du groupe d'hôtes « MySQL Servers ».

```
avg(last_foreach(/*/system.cpu.load[,avg1]?[group="MySQL Servers"]))
```

Exemple 4

Moyenne sur 5 minutes du nombre de requêtes par seconde pour le groupe d'hôtes « MySQL Servers ».

```
avg(avg_foreach(/*/mysql.qps?[group="MySQL Servers"],5m))
```

Exemple 5

Charge CPU moyenne sur tous les hôtes de plusieurs groupes d'hôtes ayant les balises spécifiques.

```
avg(last_foreach(/*/system.cpu.load?[(group="Servers A" or group="Servers B" or group="Servers C") and (ta
```

Exemple 6

Calcul utilisé sur les sommes des dernières valeurs d'élément d'un groupe d'hôtes entier.

```
sum(last_foreach(/*/net.if.out[eth0,bytes]?[group="video"])) / sum(last_foreach(/*/nginx_stat.sh[active]?[
```

Exemple 7

Le nombre total d'éléments non pris en charge dans le groupe d'hôtes « Zabbix servers ».

```
sum(last_foreach(/*/zabbix[host,,items_unsupported]?[group="Zabbix servers"]))
```

Exemple 8

Somme des derniers résultats numériques des vérifications DNS sur tous les hôtes. La forme d'élément affichée est `net.dns[192.0.2.0,example.com,A]`, à titre d'exemple d'une clé possible.

```
sum(last_foreach(/*/net.dns[*,*,*]))
```

Notez que les caractères génériques doivent correspondre au nombre de paramètres dans la clé (ici `net.dns` comporte trois paramètres : ip, name, type).

Exemples de syntaxe correcte/incorrecte

Les expressions (y compris les appels de fonction) ne peuvent pas être utilisées comme paramètres des fonctions d'historique, de tendance ou foreach. Cependant, ces fonctions elles-mêmes peuvent être utilisées dans d'autres paramètres de fonction (non historiques).

Expression	Example
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))
```

il n'est pas possible de garantir que les deux parties de l'équation auront toujours le même ensemble de valeurs. Pendant l'évaluation d'une partie de l'expression, une nouvelle valeur pour la période demandée peut arriver, et l'autre partie de l'expression aura alors un ensemble de valeurs différent.

8 Vérifications internes

Aperçu

Les contrôles internes permettent de surveiller les processus internes de Zabbix. En d'autres termes, vous pouvez surveiller ce qui se passe avec Zabbix server ou Zabbix proxy.

Les contrôles internes sont calculés :

- sur Zabbix server - si l'hôte est surveillé par le serveur
- sur Zabbix proxy - si l'hôte est surveillé par le proxy

Les contrôles internes sont traités par le serveur ou le proxy независимо du statut de maintenance de l'hôte.

Pour utiliser cet élément, choisissez 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'élément sont répertoriées sans paramètres personnalisables ni informations supplémentaires. Cliquez sur la clé d'élément pour voir tous les détails.

Item key	Description
zabbix[boottime]	L'heure de démarrage du processus Zabbix server ou Zabbix proxy, en secondes.
zabbix[cluster,discovery,nodes]	Découvre les nœuds du cluster haute disponibilité.
zabbix[connector_queue]	Le nombre de valeurs mises en file d'attente dans la file d'attente du connecteur.
zabbix[discovery_queue]	Le nombre de vérifications réseau mises en file d'attente dans la file d'attente de découverte.
zabbix[host,,items]	Le nombre d'éléments activés (pris en charge et non pris en charge) sur l'hôte.
zabbix[host,,items_unsupported]	Le nombre d'éléments non pris en charge activés sur l'hôte.
zabbix[host,,maintenance]	L'état de maintenance actuel de l'hôte.
zabbix[host,active_agent,available]	La disponibilité des vérifications d'agent actives sur l'hôte.
zabbix[host,discovery,interfaces]	Les détails de toutes les interfaces configurées de l'hôte dans le frontend Zabbix.
zabbix[host,,available]	La disponibilité de l'interface principale d'un type particulier de vérifications sur l'hôte.
zabbix[hosts]	Le nombre d'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 Zabbix Java gateway.
zabbix[lld_queue]	Le nombre de valeurs mises en file d'attente dans la file de traitement de la découverte de bas niveau.
zabbix[preprocessing]	Statistiques des valeurs reçues par le gestionnaire de prétraitement.
zabbix[preprocessing_queue]	Le nombre de valeurs mises en file d'attente dans la file d'attente de prétraitement.
zabbix[process,,]	Le pourcentage de temps qu'un processus Zabbix particulier ou un groupe de processus (identifiés par <type> et <mode>) a passé dans l'état <state>.
zabbix[proxy,,]	Les informations sur Zabbix proxy.
zabbix[proxy,discovery]	La liste des Zabbix proxies.
zabbix[proxy group,,available]	Le nombre de proxies en ligne dans un groupe de proxies.
zabbix[proxy group,,pavailable]	Le pourcentage de proxies en ligne dans un groupe de proxies.
zabbix[proxy group,,proxies]	La liste des Zabbix proxies dans un groupe de proxies.
zabbix[proxy group,,state]	L'état d'un groupe de proxies.
zabbix[proxy group,discovery]	Renvoie une liste de groupes de proxies avec des données de configuration et des données en temps réel.
zabbix[proxy_buffer,buffer,]	Renvoie les statistiques d'utilisation du tampon mémoire du proxy.

Item key	Description
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'envoi au serveur.
zabbix[queue,,]	Le nombre d'éléments surveillés dans la file d'attente dont le traitement est retardé 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 de Zabbix server ou Zabbix proxy, en nouvelles valeurs par seconde attendues.
zabbix[stats,,]	Les métriques internes d'un Zabbix server ou proxy distant.
zabbix[stats,,,queue,,]	Les métriques internes de file d'attente d'un Zabbix server ou proxy distant.
zabbix[tcache,,]	Les statistiques d'efficacité du cache des fonctions de tendance Zabbix.
zabbix[triggers]	Le nombre de déclencheurs activés dans la base de données Zabbix, avec tous les éléments activés sur des hôtes activés.
zabbix[uptime]	Le temps de fonctionnement du processus Zabbix server ou 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 de Zabbix server ou proxy.
zabbix[vmware,buffer,]	Les statistiques de disponibilité du cache vmware de 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[boottime]

 L'heure de démarrage du processus Zabbix server ou Zabbix proxy, en secondes.
 Valeur de retour : *Integer*.

zabbix[cluster,discovery,nodes]

 Découvre les nœuds du cluster haute disponibilité.
 Valeur de retour : *objet JSON*.

Commentaires :

- Cet élément peut être utilisé dans la découverte de bas niveau.
- Les champs renvoyés correspondent pour la plupart 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 heartbeat du nœud (db_timestamp - lastaccess)
  - `status` — état du nœud :
    - 0 — en attente
    - 1 — arrêté manuellement
    - 2 — indisponible
    - 3 — actif

Exemple de valeurs de retour :

```
[
  {
    "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 particulier de contrôles 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'inaccessibilité/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>]

 Les informations concernant 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 une utilisation future.

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 pendant lequel un processus Zabbix particulier ou un groupe de processus (identifiés par <type> et <mode>) a passé dans l'état <state>. Il est calculé uniquement pour la dernière minute.
 Valeur de retour : *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*, *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 (entre 1 et le nombre d'instances pré-forkées ; par exemple, si 4 processus trapper sont en cours d'exécution, la valeur est comprise entre 1 et 4) ;
- **state** - *busy* - le processus est dans un état occupé, par exemple lors du traitement d'une requête (par défaut) ;
idle - le processus est dans un état inactif et ne fait rien.

Commentaires :

- Si <mode> est un numéro de processus Zabbix qui n'est pas en cours d'exécution (par exemple, avec 5 pollers en cours d'exécution, <mode> est défini sur 6), un tel élément deviendra non pris en charge.
- Les valeurs minimale et maximale se réfèrent au pourcentage d'utilisation d'un seul processus. Ainsi, si dans un groupe de 3 pollers les pourcentages d'utilisation par processus sont 2, 18 et 66, *min* renverra 2 et *max* renverra 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 (busy/idle) sont enregistrés au moment du changement ; ainsi, un processus qui devient occupé est enregistré comme tel et ne change ni ne met à jour son état jusqu'à ce qu'il devienne inactif. Cela garantit que même les processus complètement bloqués seront correctement enregistrés comme étant occupés à 100 %.
- Actuellement, "busy" signifie "non endormi", mais à l'avenir des états supplémentaires pourraient être introduits : attente de verrous, exécution de requêtes de base de données, etc. Notez que les pollers asynchrones sont considérés comme occupés s'ils ont atteint la limite définie par le paramètre de configuration MaxConcurrentChecksPerPoller du server/proxy.
- Sous Linux et sur la plupart des autres systèmes, la résolution est de 1/100 de seconde.

Exemples :

```
zabbix[process,poller,avg,busy] #le temps moyen passé par les processus poller à faire quelque chose penda
zabbix[process,"icmp pinger",max,busy] #le temps maximal passé à faire quelque chose par n'importe quel pr
zabbix[process,"history syncer",2,busy] #le temps passé à faire quelque chose par le processus history syn
zabbix[process,trapper,count] #le nombre de processus trapper actuellement en cours d'exécution
```

zabbix[proxy,<name>,<param>]

 Les informations concernant le proxy Zabbix.
 Valeur de retour : *Integer*.

Paramètres :

- **name** - le nom du proxy ;
- **param** - *lastaccess* - l'horodatage du dernier message heartbeat reçu du proxy ;
delay - depuis combien de temps les valeurs collectées n'ont pas été envoyées ; calculé comme « proxy delay » + (« current server time » - « proxy lastaccess »), où « proxy delay » est la différence entre l'heure actuelle du proxy et l'horodatage de la plus ancienne valeur non envoyée sur le proxy.

Commentaires :

- Cet élément est toujours traité par le serveur Zabbix, quel que soit l'emplacement de l'hôte (sur le serveur ou le proxy).
- La fonction `fuzzytime()` peut être utilisée pour vérifier la disponibilité du proxy.

Exemple :

```
zabbix[proxy,"Germany",lastaccess] #l'horodatage du dernier message heartbeat reçu du proxy "Germany"
```

zabbix[proxy,discovery]

 La liste des proxys Zabbix avec le nom, le mode, le chiffrement, la compression, la version, la date de dernière activité, le nombre d'hôtes, le nombre d'éléments, les valeurs requises par seconde (vps), le statut de version (actuelle/obsolète/non prise en charge), les délais d'expiration par type d'élément, le nom du groupe de proxys (si le proxy appartient à un groupe), l'état (inconnu/hors ligne/en ligne).
 Valeur de retour : *objet JSON*.

zabbix[proxy group,<name>,available]

 Le nombre de proxys en ligne dans un groupe de proxys.
 Valeur de retour : *Integer*.

Paramètres :

- **name** - le nom du groupe de proxys.

zabbix[proxy group,<name>,pavailable]

 Le pourcentage de proxys en ligne dans un groupe de proxys.
 Valeur de retour : *Float*.

Paramètres :

- **name** - le nom du groupe de proxys.

zabbix[proxy group,<name>,proxies]

 La liste des proxys Zabbix dans un groupe de proxys avec le nom, le mode, le chiffrement, la compression, la version, la date de dernière activité, le nombre d'hôtes, le nombre d'éléments, les valeurs requises par seconde (vps), le statut de version (actuelle/obsolète/non prise en charge), les délais d'expiration, le nom du groupe de proxys, l'état (inconnu/hors ligne/en ligne).
 Valeur de retour : *JSON*.

Paramètres :

- **name** - le nom du groupe de proxys.

zabbix[proxy group,<name>,state]

 L'état d'un groupe de proxys.
 Valeur de retour : *0* - inconnu ; *1* - hors ligne ; *2* - en cours de récupération ; *3* - en ligne ; *4* - en dégradation.

Paramètres :

- **name** - le nom du groupe de proxys.

zabbix[proxy group,discovery]

 Renvoie une liste de groupes de proxys avec des données de configuration et des données en temps réel. Les données de configuration incluent le nom du groupe de proxys, le délai de basculement et le nombre minimal de proxys en ligne requis. Les données en temps réel incluent l'état du groupe de proxys (voir les commentaires pour plus de détails), le nombre de proxys en ligne et le pourcentage de proxys en ligne.
 Valeur de retour : *JSON*.

Commentaires :

- Cet élément ne renvoie pas les proxys sans groupe.
- S'il existe une valeur invalide pour "failover_delay" ou "min_online", une valeur spéciale *-1* est renvoyée pour l'indiquer. Des valeurs invalides peuvent apparaître si des macros sont utilisées pour la configuration et que ces macros ne peuvent pas être développées en une valeur valide.
- L'état du groupe de proxys est renvoyé sous forme d'entier : *0* - inconnu ; *1* - hors ligne ; *2* - en récupération ; *3* - en ligne ; *4* - en dégradation.

Exemple de valeurs de retour :

```
{
    "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>]

 Les statistiques d'utilisation du tampon mémoire du proxy.
 Valeurs de retour : *Integer* (pour la taille) ; *Float* (pour le pourcentage).

Paramètres :

- **mode** : *total* - la taille totale du tampon (peut être utilisée pour vérifier si le tampon mémoire est activé) ;
free - la taille du tampon libre ;
pfree - le pourcentage de tampon libre ;
used - la taille du tampon utilisé ;
pused - le pourcentage de tampon utilisé.

Commentaires :

- Renvoie 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 où les nouvelles données sont stockées.
 Valeurs de retour : *0* - disque ; *1* - mémoire.

Commentaires :

- « 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 de retour : *Integer*.

Commentaires :

- 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 dont le retard est d'au moins <from> secondes, mais inférieur à <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'a pas connaissance des périodes de maintenance ; voir Calcul des files d'attente pendant la maintenance pour plus de détails.

zabbix[rcache,<cache>,<mode>]

 The availability statistics of the Zabbix configuration cache.
 Return values: *Integer* (for size); *Float* (for percentage).

Parameters:

- **cache** - *buffer*;
- **mode** - *total* - the total size of buffer
free - the size of free buffer
pfree - the percentage of free buffer
used - the size of used buffer
pused - the percentage of used buffer

zabbix[requiredperformance]

 Les performances requises du serveur Zabbix ou du proxy Zabbix, exprimées en nouvelles valeurs par seconde attendues.
 Valeur de retour : *Float*.

Commentaires :

- Correspond approximativement à « Required server performance, new values per second » dans *Reports* > *System information*.

zabbix[stats,<ip>,<port>]

 Les métriques internes d'un serveur ou proxy Zabbix distant.
 Valeurs de retour : *objet JSON*.

Paramètres :

- **ip** - la liste d'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 :

- La requête de statistiques ne sera acceptée que depuis les adresses répertoriées dans le paramètre 'StatsAllowedIP' du server/proxy sur l'instance cible.
- Cet élément renvoie un ensemble sélectionné de métriques internes. Pour plus de détails, voir Surveillance à distance des statistiques Zabbix.

zabbix[stats,<ip>,<port>,queue,<from>,<to>]

 Les métriques de file d'attente internes (voir `zabbix[queue,<from>,<to>]`) d'un serveur ou proxy Zabbix distant.
 Valeurs de retour : *objet JSON*.

Paramètres :

- **ip** - la liste d'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) ;
- **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 :

- La requête de statistiques ne sera acceptée que depuis les adresses répertoriées dans le paramètre 'StatsAllowedIP' du server/proxy sur l'instance cible.
- Un ensemble sélectionné de métriques internes est renvoyé par cet élément. Pour plus de détails, voir Remote monitoring of Zabbix stats.

> **Attention:**
> Le proxy Zabbix n'a pas connaissance des périodes de maintenance ; voir Calculation of queues during 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 :

- **cache** - *buffer* ;
- **parameter** - *all* - nombre total de requêtes du cache (par défaut) ;
hits - accès réussis au cache ;
phits - pourcentage d'accès réussis au cache ;
misses - échecs d'accès au cache ;
pmisses - pourcentage d'échecs d'accès au cache ;
items - nombre d'éléments mis en cache ;
requests - nombre de requêtes mises en cache ;
pitems - pourcentage d'éléments mis en cache par rapport au total des éléments mis en cache + requêtes. Un faible pourcentage signifie très probablement que la taille du cache peut être réduite.

Commentaires :

- Cet élément n'est pas pris en charge sur Zabbix proxy.

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 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.

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 Zabbix.
 Valeur de retour : *Integer*. Si <parameter> est défini sur *mode*, la valeur de retour est : *0* - mode normal ; *1* - mode mémoire faible.

Paramètres :

- **parameter** - *requests* - le nombre total de requêtes ;
hits - le nombre de succès du cache (valeurs d'historique prises depuis le cache) ;
misses - le nombre d'échecs 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 mémoire faible activé, le cache de valeurs restera dans cet état pendant 24 heures, même si le problème ayant 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 en valeurs par seconde.
- Cet élément n'est pas pris en charge sur Zabbix proxy.

zabbixversion

 La version du serveur ou du proxy Zabbix.
 Valeur de retour : *String*. Par exemple : `8.0.0`.

zabbix[vmware,buffer,<mode>]

 The availability statistics of the Zabbix vmware cache.
 Return values: *Integer* (for size); *Float* (for percentage).

Parameters:

- **mode** - *total* - the total size of buffer
free - the size of free buffer
pfree - the percentage of free buffer
used - the size of used buffer
pused - the percentage of used buffer

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 *values*) *all* (par défaut) - le nombre total de valeurs traitées par le serveur/proxy Zabbix, à l'exception des éléments non pris en charge (compteur) ;
float - le nombre de valeurs à virgule flottante traitées (compteur) ;
uint - le nombre de valeurs entières non signées traitées (compteur) ;
str - le nombre de valeurs caractère/chaîne traitées (compteur) ;
log - le nombre de valeurs de journal traitées (compteur) ;
text - le nombre de valeurs texte 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 le cache *history*, *index*, *trend*) *pfree* (par défaut) - le pourcentage de tampon libre ;
total - la taille totale du tampon ;
free - la taille du tampon libre ;
used - la taille du tampon utilisé ;
pused - le pourcentage de tampon utilisé.

Commentaires :

- Le paramètre de cache `trend` n'est pas pris en charge avec Zabbix proxy.
- Le cache d'historique est utilisé pour stocker les valeurs des éléments. Un nombre faible indique des problèmes de performances du côté de la base de données.
- Le cache d'index d'historique est utilisé pour indexer les valeurs stockées dans le cache d'historique.
- Une fois le cache d'historique rempli puis vidé, le cache d'index d'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.

9 Vérifications SSH

Aperçu

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 à la place de 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 une authentification basée sur des clés 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 seront stockées 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.

Habituellement, les paquets d'installation du serveur Zabbix provenant de différentes distributions de systèmes d'exploitation 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 réaffecter le répertoire personnel à */home/zabbix*, afin qu'il corresponde au paramètre de configuration du serveur Zabbix `SSHKeyLocation` 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 en cours qui l'utilisent doivent être arrêtés :

```
systemctl stop zabbix-agent
systemctl stop zabbix-server
```

Pour modifier l'emplacement du répertoire personnel en essayant de le déplacer (s'il existe), la commande suivante doit être exécutée :

```
usermod -m -d /home/zabbix zabbix
```

Il est également possible qu'aucun répertoire personnel n'existe à l'ancien emplacement ; il doit alors être créé au nouvel emplacement. Une tentative sûre de le faire est :

```
test -d /home/zabbix || mkdir /home/zabbix
```

Pour s'assurer que tout est sécurisé, des commandes supplémentaires peuvent être exécutées afin de définir les permissions du répertoire personnel :

```
chown zabbix:zabbix /home/zabbix
chmod 700 /home/zabbix
```

Les processus précédemment arrêtés peuvent maintenant être redémarrés :

```
systemctl start zabbix-agent
systemctl start zabbix-server
```

À présent, les étapes de génération des clés publique et privée peuvent être effectuées à l'aide des commandes suivantes (pour une meilleure lisibilité, les invites de commande sont commentées) :

```
sudo -u zabbix ssh-keygen -t rsa
##### Génération de la paire de clés rsa publique/privée.
##### Entrez le fichier dans lequel enregistrer la clé (/home/zabbix/.ssh/id_rsa) :
/home/zabbix/.ssh/id_rsa
##### Entrez la phrase secrète (laisser vide pour ne pas utiliser de phrase secrète) :
<Leave empty>
##### Entrez à nouveau la même phrase secrète :
<Leave empty>
##### Votre identification a été enregistrée dans /home/zabbix/.ssh/id_rsa.
##### Votre clé publique a été enregistrée dans /home/zabbix/.ssh/id_rsa.pub.
##### L'empreinte de la clé est :
##### 90:af:e4:c7:e3:f0:2e:5a:8d:ab:48:a2:0c:92:30:b9 zabbix@it0
##### L'image randomart de la clé est :
##### +--[ 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*, qui correspond au paramètre de configuration du serveur Zabbix `SSHKeyLocation`.

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 ne doit être effectuée qu'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 mises en commentaire) :

```
sudo -u zabbix ssh-copy-id root@10.10.10.10
##### L'authenticité de l'hôte '10.10.10.10 (10.10.10.10)' ne peut pas être établie.
##### L'empreinte de la clé RSA est 38:ba:f2:a4:b5:d9:8f:52:00:09:f7:1f:75:cc:0b:46.
##### Êtes-vous sûr de vouloir continuer la connexion (yes/no) ?
yes
##### Avertissement : '10.10.10.10' (RSA) a été ajouté de façon permanente à la liste des hôtes connus.
##### Mot de passe de root@10.10.10.10 :
<Entrez le mot de passe root>
##### Essayez maintenant de vous connecter à la machine avec "ssh 'root@10.10.10.10'",
##### et vérifiez que seules les clés que vous vouliez ont été ajoutées.
```

Il est maintenant possible de vérifier la connexion SSH en utilisant la clé privée par défaut (*/home/zabbix/.ssh/id_rsa*) pour le compte utilisateur *zabbix* :

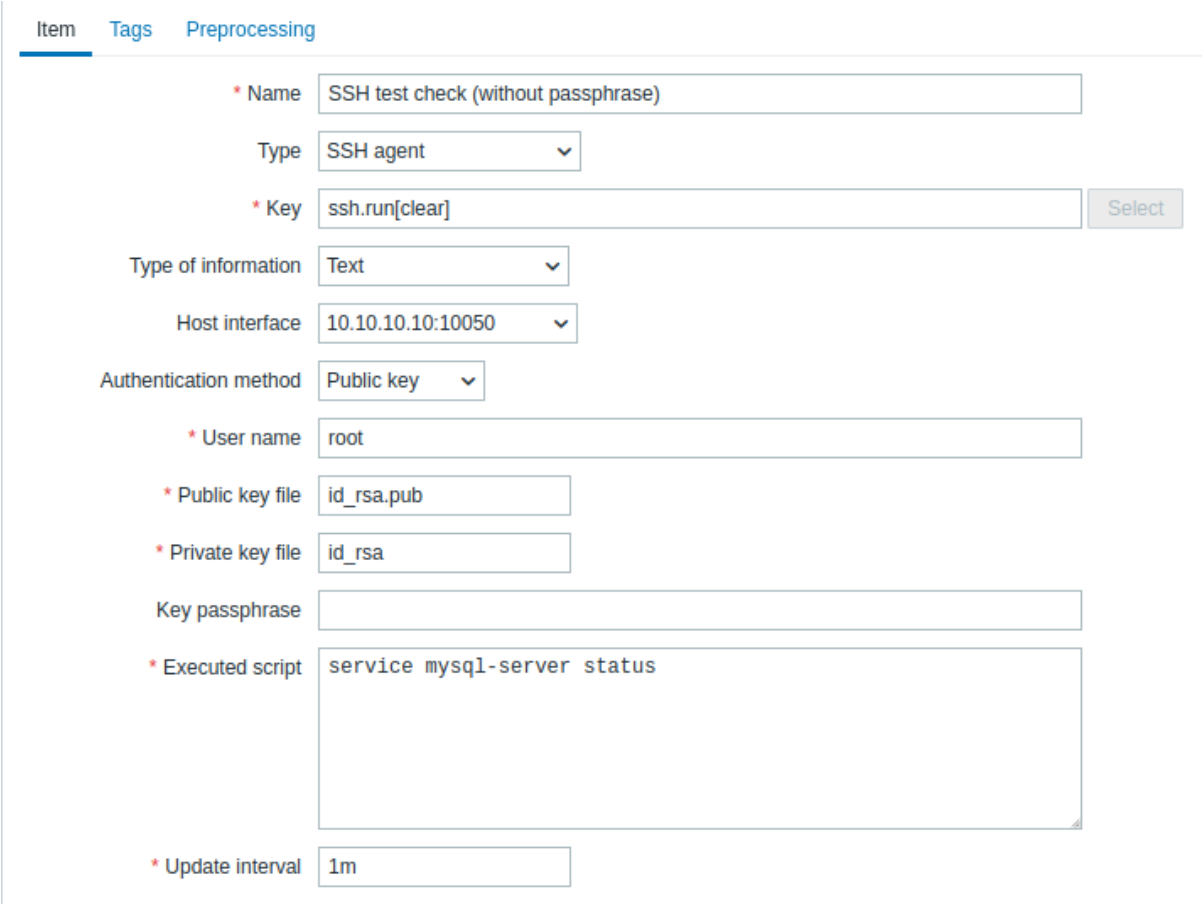
```
sudo -u zabbix ssh root@10.10.10.10
```

Si la connexion réussit, alors la partie configuration dans le shell est terminée et la session SSH distante peut être fermée.

Configuration de l'élément

La ou les commande(s) réelles 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 de saisie obligatoires sont marqués d'un astérisque rouge.

Les champs qui nécessitent des informations spécifiques pour les éléments SSH sont :

Parameter	Description	Comments
Type	Sélectionnez **SSH agent** ici.	

Parameter	Description	Comments

Key	Clé d'élément unique (par hôte) au format **ssh.run[courte description unique,<ip>,<port>,<encoding>,<ssh options>,<subsystem>]**	La **courte description unique** 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.
		
		Les **ssh options** permettent 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 une virgule (dans ce cas, le paramètre doit être entre guillemets) ; plusieurs clés d'option peuvent être transmises en les séparant par un point-virgule.
		
		Les clés d'option suivantes sont prises en charge : `KexAlgorithms`, `HostkeyAlgorithms`, `Ciphers`, `MACs`, `PubkeyAcceptedKeyTypes`. La prise en charge des clés et valeurs d'option dépend de la bibliothèque SSH (par exemple, `PubkeyAcceptedKeyTypes` n'est pris en charge qu'avec libssh) ; si une option n'est pas prise en charge, une erreur sera renvoyée et l'élément deviendra non pris en charge.
		
		Notez que le signe "+" pour ajouter des paramètres de chiffrement et "!" 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 à des opérations spécifiques autorisées par le sous-système (par exemple, les transferts de fichiers via SFTP ou la gestion d'équipements réseau via NETCONF). Notez que l'utilisation d'un sous-système peut également nécessiter l'utilisation d'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]`

Authentication method	L'une des méthodes « Password » ou « Public key ».	
User name	Nom d'utilisateur (jusqu'à 255 caractères) pour l'authentification sur l'hôte distant. Obligatoire.	
Public key file	Nom du fichier de clé publique si *Authentication method* est « Public key ». Obligatoire.	Exemple : *id_rsa.pub* - nom de fichier de clé publique par défaut généré par une commande ssh-keygen.

Parameter	Description	Comments
Private key file	Nom du fichier de clé privée si *Authentication method* est « Public key ». Obligatoire.	Exemple : *id_rsa* - nom de fichier de clé privée par défaut.
Password or *Key passphrase*	Mot de passe (jusqu'à 255 caractères) pour l'authentification ou Phrase secrète **si** elle a été utilisée pour la clé privée.	Laissez le champ *Key passphrase* vide si aucune phrase secrète n'a été utilisée. Voir aussi les problèmes connus concernant l'utilisation de la phrase secrète.

Executed script	Commande(s) shell exécutée(s) à l'aide d'une session distante SSH.	La valeur de retour de la ou des commande(s) shell exécutée(s) est limitée à 16 Mo (y compris les espaces de fin qui sont tronqués) ; les limites de la base de données s'appliquent également.
		
		Notez que la bibliothèque libssh2 peut tronquer les scripts exécutables à ~32kB.
		
		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>`
		`]]>]]>`

10 Vérifications 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

Les commandes réelles à exécuter doivent être placées dans le champ **Script exécuté** dans 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 retournée sera également formulée en multi-lignes.

Caractères pris en charge avec lesquels l'invite du shell peut se terminer :

- $
- #
-
- %

Note:
Une ligne d'invite telnet qui se termine 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 uniquement au début de la session Telnet.

Clé	Description	Commentaires
telnet.run[<unique short description>,<ip>,<port>,<encoding>]	Exécute une commande sur un périphérique distant à l'aide de la connexion Telnet	

Attention:
Si une vérification telnet renvoie une valeur avec des caractères non ASCII et un codage non-UTF8, le paramètre <encoding> de la clé doit être correctement spécifié. Voir la page d'encodage des valeurs renvoyées pour plus de détails.

11 Vérifications externes

Vue d'ensemble

Une vérification externe est une vérification exécutée par le serveur Zabbix en lançant un script shell ou un binaire. Toutefois, lorsque des hôtes sont supervisés par un proxy Zabbix, les vérifications externes sont exécutées par le proxy.

Les vérifications externes ne nécessitent pas qu'un agent soit exécuté sur l'hôte supervisé.

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[] or
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 server/proxy Zabbix). La commande sera exécutée sous le même utilisateur que le serveur/proxy Zabbix ; ainsi, les autorisations d'accès ou les variables d'environnement doivent être gérées dans un script wrapper, si nécessaire. Les autorisations sur la commande doivent également permettre à cet utilisateur de l'exécuter. Seules les commandes présentes dans le répertoire spécifié peuvent être exécutées.

Warning:
N'abusez pas des vérifications externes, car chaque script nécessite le démarrage 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 soit configuré pour utiliser une 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.

12 Elément trapper

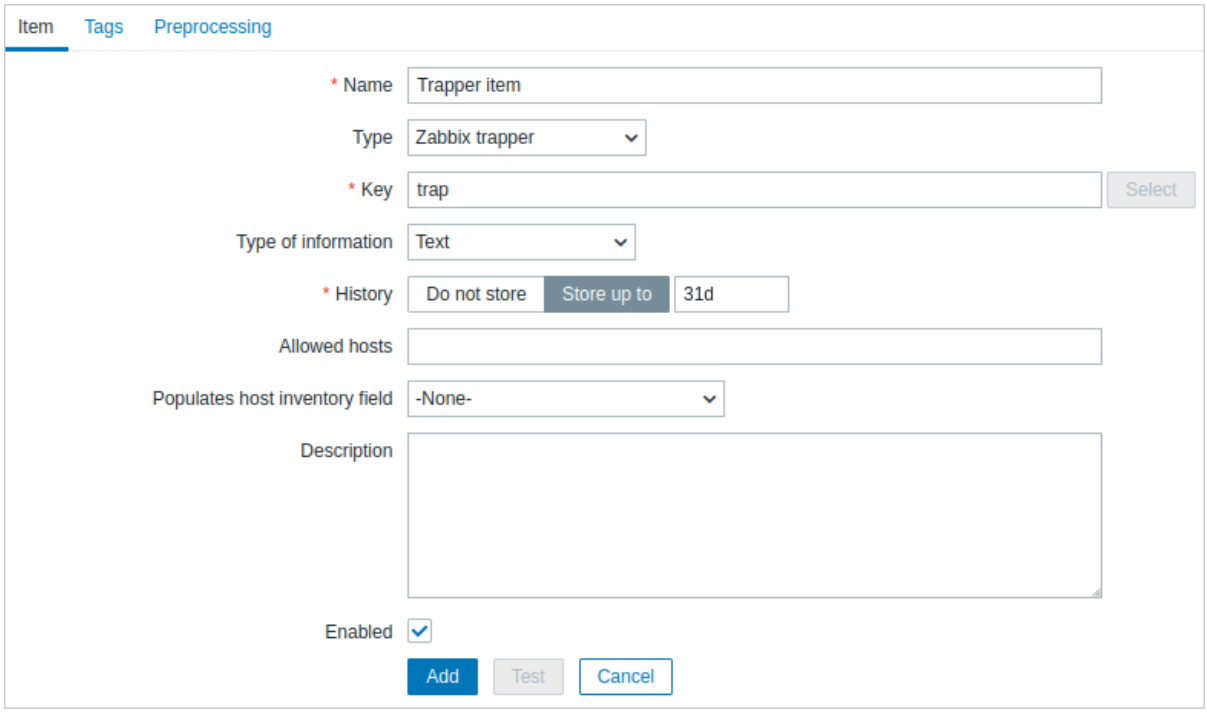
Vue d'ensemble

Les éléments trapper acceptent des données entrantes au lieu d'aller les interroger. Cela est utile pour toutes les données que vous souhaitez envoyer à Zabbix.

Configuration

Pour configurer un élément trapper :

1. Accédez à *Collecte de données* → *Hôtes*.
2. Cliquez sur *Éléments* sur 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 de saisie obligatoires sont marqués d'un astérisque rouge.

Les champs qui nécessitent des informations spécifiques pour les éléments trapper sont :

Type	Sélectionnez "Zabbix trapper".
Clé	Saisissez une clé qui sera utilisée pour reconnaître l'élément lors de l'envoi des données au serveur Zabbix.
Type d'information	Sélectionnez le type d'information qui correspondra au format des données qui seront envoyées.
Hôtes autorisés	Liste d'adresses IP séparées par des virgules (éventuellement en notation CIDR) ou de noms DNS. Si ce champ est renseigné, 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 manière identique 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 déconseillées par 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 pour que le serveur Zabbix prenne en compte les modifications lors d'une 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 d'API `history.push`.

Zabbix sender

Pour envoyer des données au serveur ou au proxy Zabbix 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 (veillez à utiliser le nom d'hôte technique plutôt que le nom visible)
- `-k` pour spécifier la clé de l'élément configurée 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 ou le proxy Zabbix, consultez *protocole Zabbix sender*.

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 ;

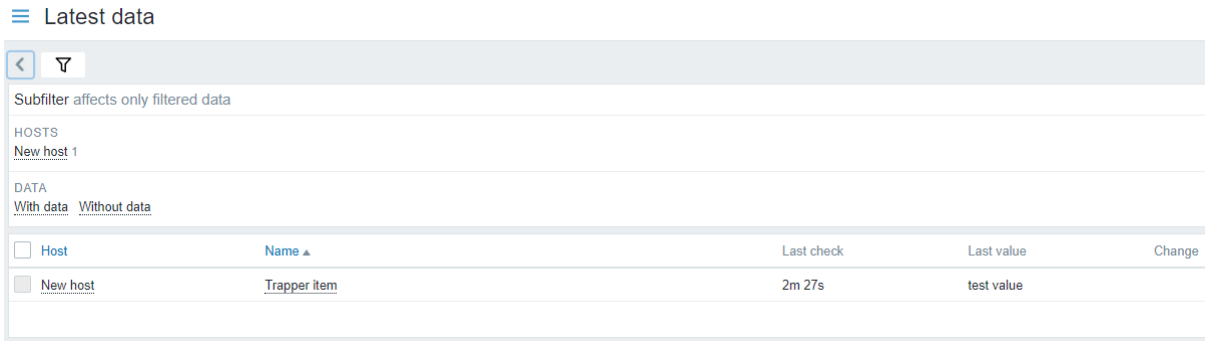
- l'hôte est désactivé ou en maintenance sans collecte de données ;
- l'élément n'existe pas ou n'est pas encore inclus dans le cache de configuration du serveur ;
- l'élément est désactivé ou son type est autre que Zabbix trapper ou HTTP agent (avec le trapping activé) ;
- l'adresse IP ou le DNS de 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 *Surveillance* → *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.

13 Surveillance 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.

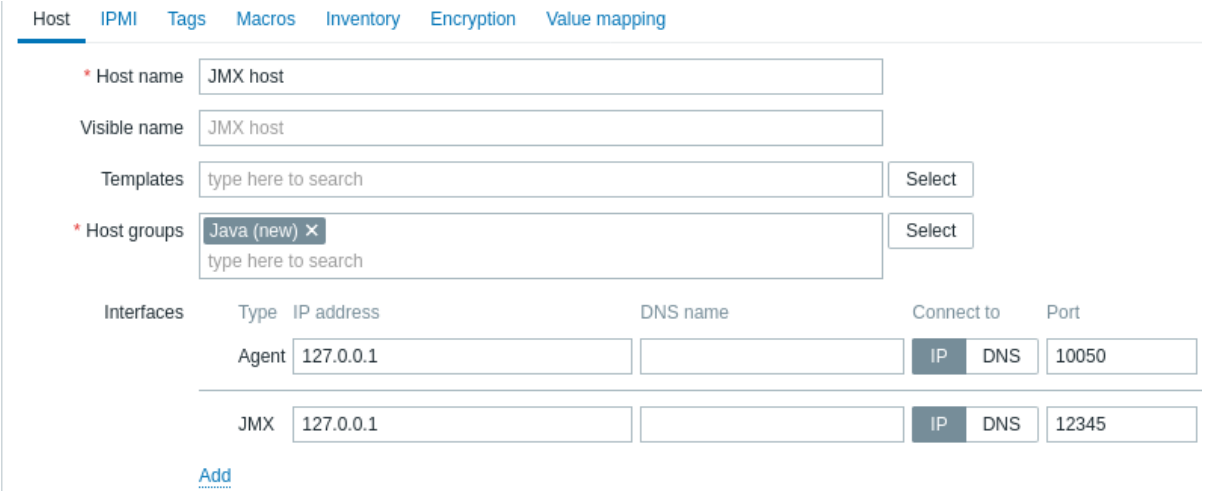
Consultez Monitoring and Management Using JMX pour une description détaillée.

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 :

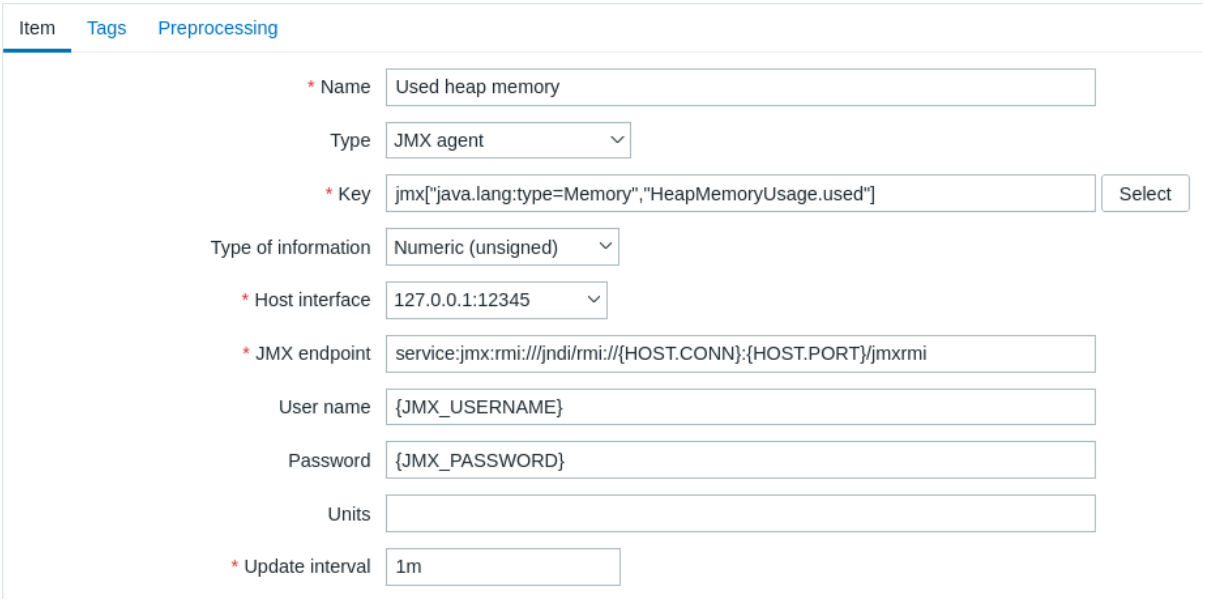


Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge.

Ajout d'un élément d'agent JMX

Pour chaque compteur JMX qui vous intéresse, 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 : **nom d'objet** - le nom d'objet d'un MBean **nom d'attribut** - un nom d'attribut MBean avec, éventuellement, des noms de champs de données composites séparés par des points **description courte unique** - une description unique permettant d'avoir 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[]` de découverte de bas niveau.
JMX endpoint	Vous pouvez spécifier un JMX endpoint personnalisé. Assurez-vous que les paramètres de connexion du JMX endpoint correspondent à l'interface JMX. Cela peut être réalisé en utilisant les macros {HOST.*}, comme dans le JMX endpoint par défaut. Les macros {HOST.*} et les macros utilisateur sont prises en charge.
User name	Indiquez 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	Indiquez 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 superviser un compteur booléen qui vaut soit "true", soit "false", vous devez alors spécifier le type d'information comme "Numeric (unsigned)" et sélectionner l'étape de prétraitement "Boolean to decimal" dans l'onglet Preprocessing. Le serveur stockera alors les valeurs booléennes sous forme de 1 ou 0, respectivement.

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 gérer tout autre caractère spécial dans une clé d'élément JMX, veuillez consulter la section sur le format de clé d'élément.

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 protocole RMI par défaut.

Pour illustrer cette possibilité, essayons de configurer la supervision de JBoss EAP 6.4 comme exemple. Commençons par poser quelques hypothèses :

- Vous avez déjà installé Zabbix Java gateway. Sinon, vous pouvez le faire conformément à la documentation.
- Zabbix server et Java gateway sont installés avec le préfixe `/usr/local/`
- JBoss est déjà installé dans `/opt/jboss-eap-6.4/` et fonctionne 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)

Appliquons quelques réglages simples dans zabbix_server.conf :

```
JavaGateway=127.0.0.1
StartJavaPollers=5
```

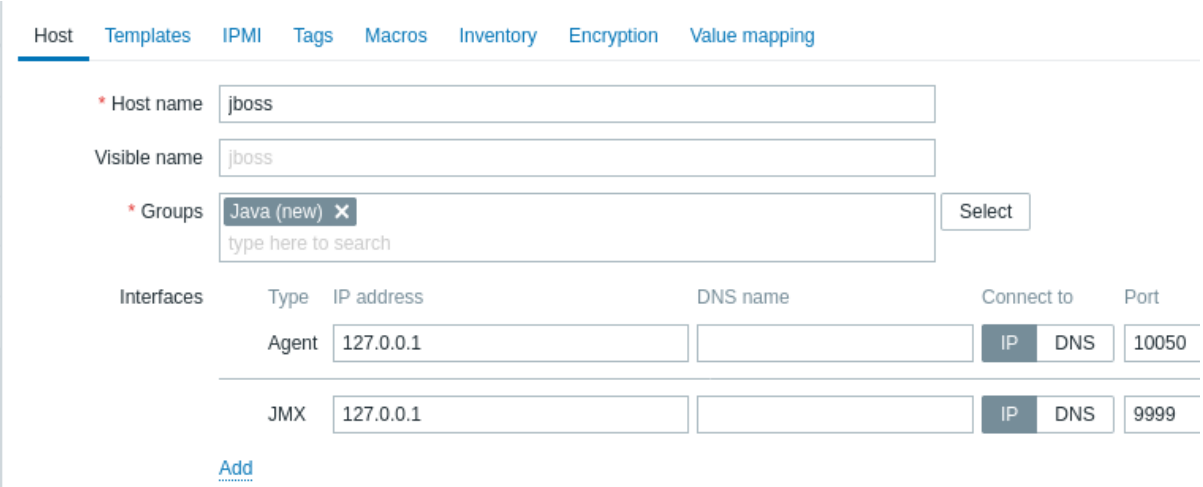
Et dans le fichier de configuration `zabbix_java/settings.sh` (ou `zabbix_java_gateway.conf`) :

```
START_POLLERS=5
```

Vérifiez que JBoss écoute sur son port de gestion standard :

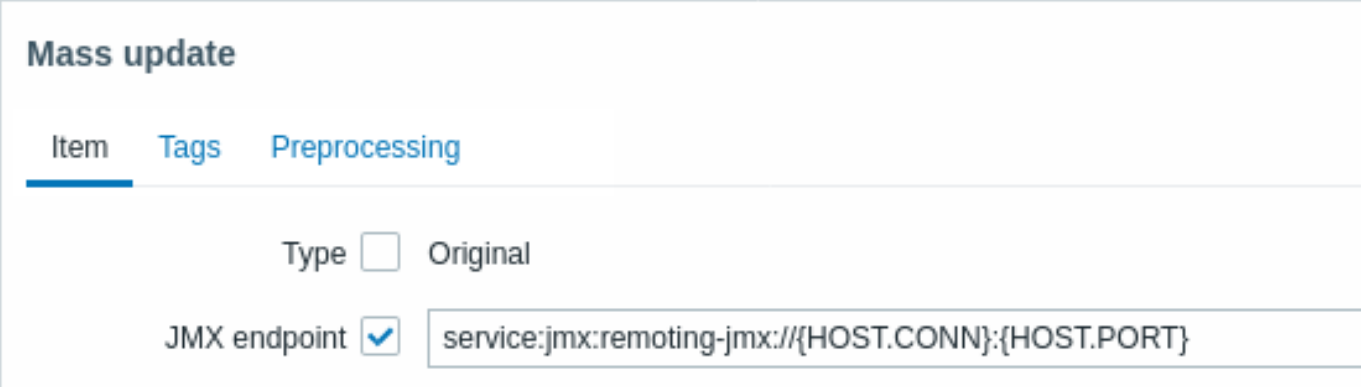
```
$ netstat -natp | grep 9999
tcp        0      0 127.0.0.1:9999          0.0.0.0:*               LISTEN      10148/java
```

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 :

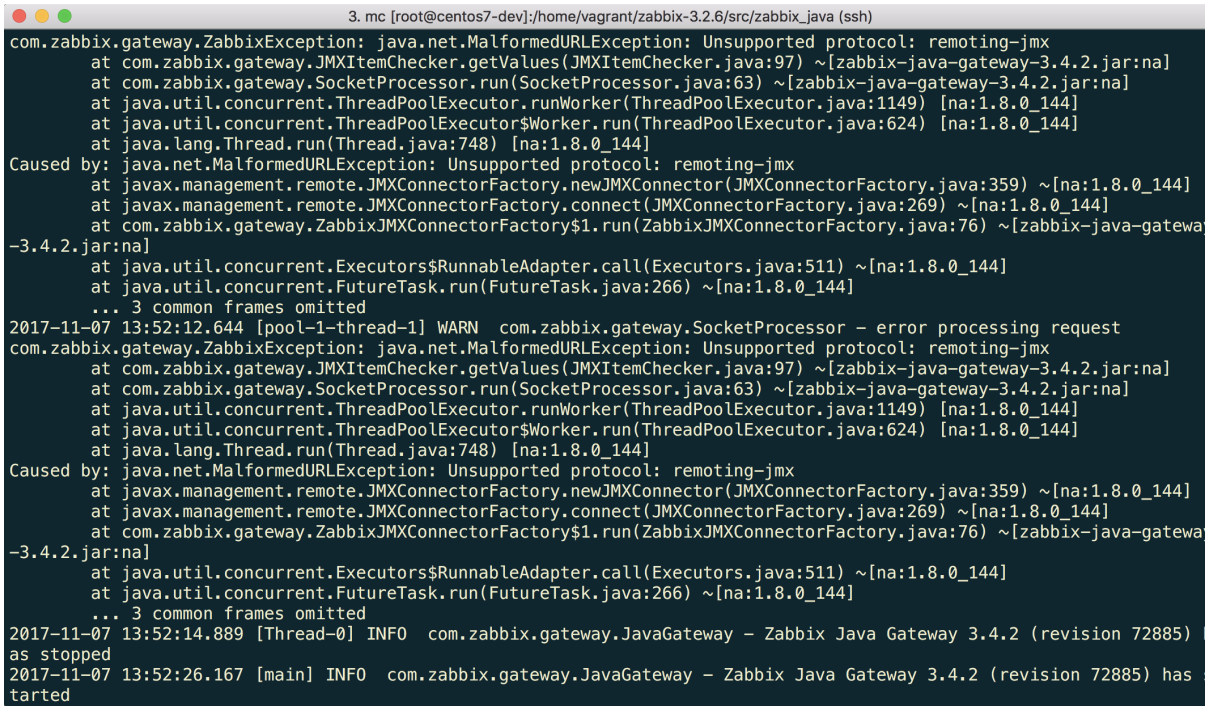
```
service:jmx:remoting-jmx://{HOST.CONN}:{HOST.PORT}
```



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 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, Java gateway disposera de tous les modules nécessaires pour fonctionner avec jmx-remoting. Il ne reste plus qu'à redémarrer Java gateway, attendre un peu et, si vous avez tout fait correctement, constater que les données de supervision JMX commencent à arriver dans Zabbix (voir aussi : Données récentes).

14 Surveillance ODBC

Aperçu

La surveillance ODBC correspond au type d'élément **Database monitor** dans l'interface web Zabbix.

ODBC est une API middleware du langage de programmation C permettant d'accéder aux systèmes de gestion de bases de données (SGBD). Le concept ODBC a été développé par Microsoft, puis porté sur d'autres plateformes.

Zabbix peut interroger toute base de données prise en charge par ODBC. Pour ce faire, Zabbix ne se connecte pas directement aux bases de données, mais utilise l'interface ODBC et les pilotes configurés dans ODBC. Cela permet une surveillance plus efficace de différentes bases de données à des fins multiples (par exemple, vérifier des files d'attente spécifiques de base de données, des 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 d'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.

Parameter	Description
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

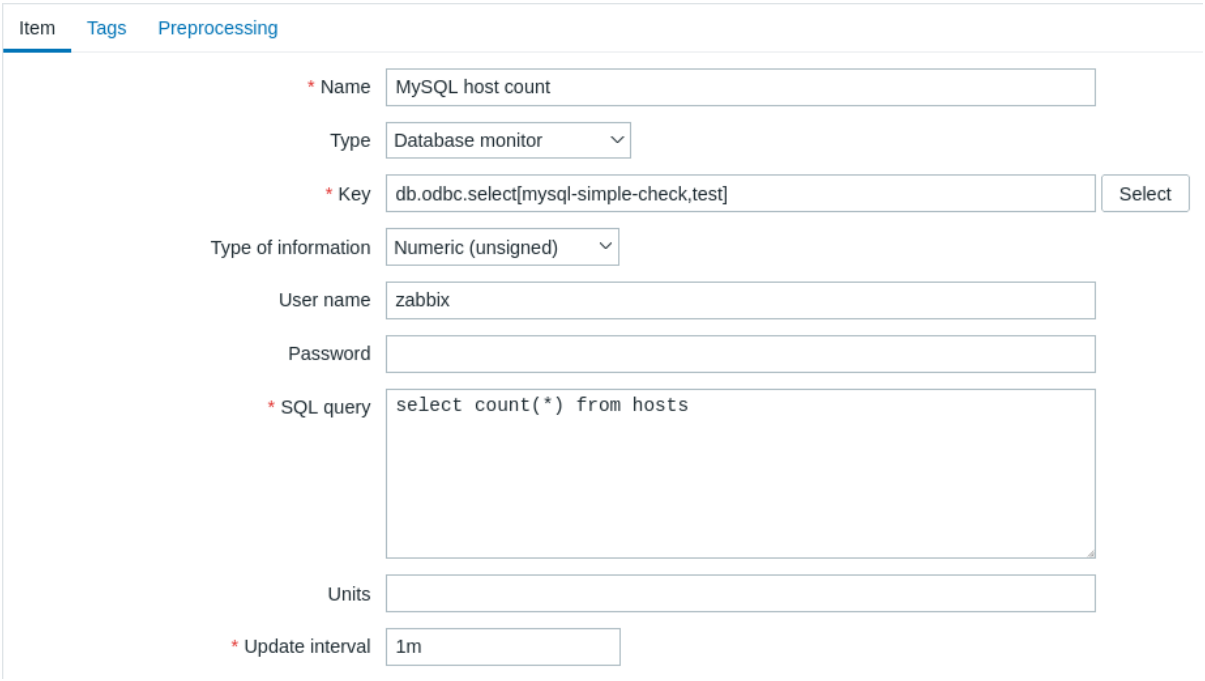
Pour vérifier si la connexion ODBC fonctionne correctement, vous pouvez utiliser l'utilitaire `isql` (inclus dans le paquet `unixODBC`) :

```
isql test
+---------------------------------------+
| Connected!                            |
```

```
|                                      |
| sql-statement                        |
| help [tablename]                     |
| quit                                 |
|                                      |
+--------------------------------------+
```

Configuration d'un élément dans l'interface web Zabbix

Configurez un item 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 « Database monitor » ici.
Key	Saisissez l'une des clés d'élément prises en charge : **db.odbc.select[]** - cet élément renvoie une 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. 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 résultante, 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 ne doit renvoyer qu'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 de manière incorrecte, l'élément deviendra non pris en charge.

Remarques 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 pollers ODBC, définissez le paramètre `StartODBCPollers` dans le fichier de configuration du server 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'expiration de connexion ODBC et comme délai d'expiration d'exécution de la requête. Notez que ces paramètres de délai d'expiration peuvent être ignorés si le pilote ODBC installé ne les prend pas en charge.
- La commande SQL doit renvoyer un ensemble de résultats comme toute requête utilisant l'instruction `select`. La syntaxe de la requête dépend du SGBDR qui la traitera. La syntaxe d'une requête vers une procédure stockée 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[<description courte unique>,<dsn>,<chaîne de connexion>]

 Transforme le résultat de la requête SQL en un tableau JSON.
 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.
- Plusieurs lignes/colonnes au format JSON peuvent être renvoyées. Cet élément peut être utilisé comme élément maître collectant 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, consultez un exemple du format renvoyé, utilisé dans la découverte de bas niveau.

Exemple :

```
# Connexion pour le pilote ODBC MySQL 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.

15 Éléments dépendants

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.

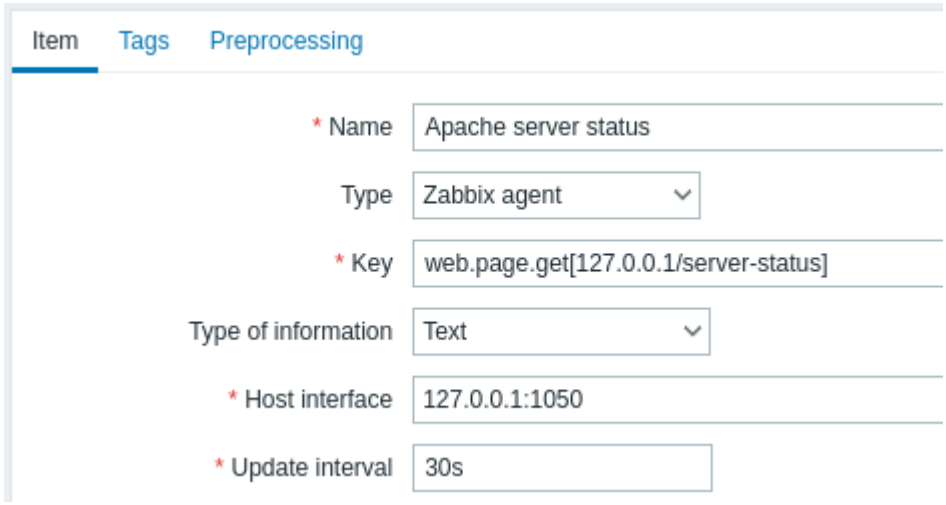
Limitations

- 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 de l'élément

Un élément dépendant dépend de son élément maître pour les données. C'est pourquoi l'**élément maître** doit d'abord être configuré (ou exister) :

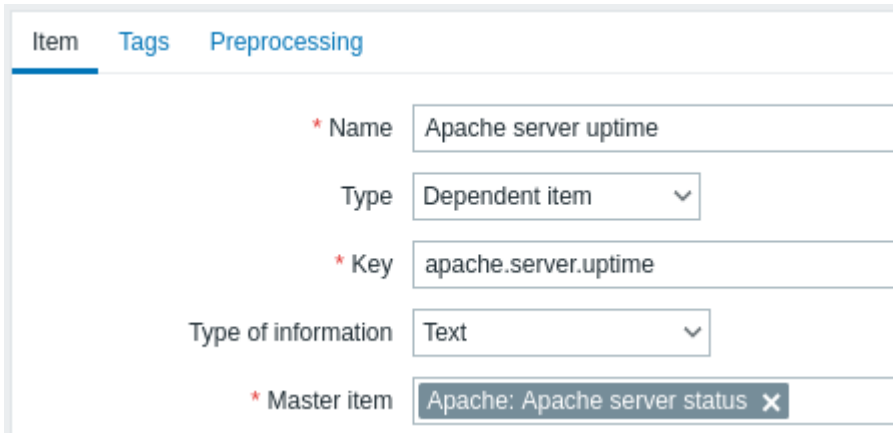
- Allez à : *Collecte de données* → *Hôtes*
- Cliquez sur *Éléments* sur la ligne de l'hôte
- Cliquez sur *Créer un élément*
- Saisissez les paramètres de l'élément dans le formulaire



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**.

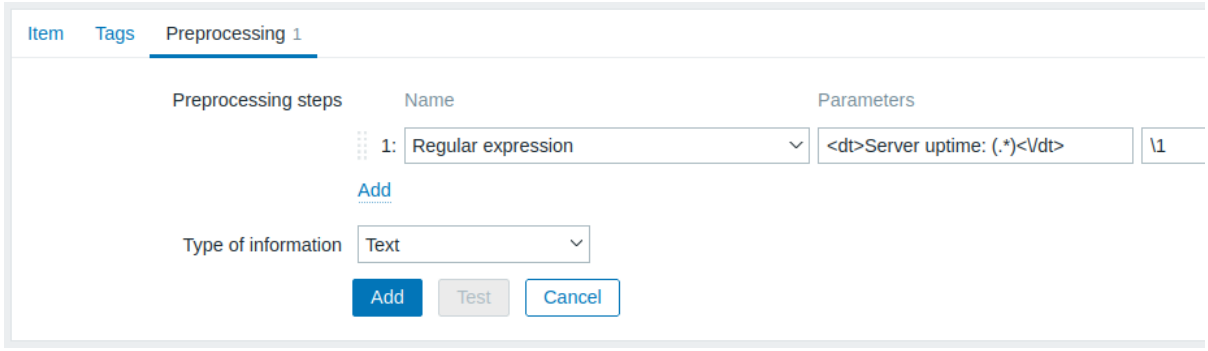


Tous les champs obligatoires sont marqués d'un astérisque rouge.

Les champs qui nécessitent des informations spécifiques pour les éléments dépendants sont :

Type	Sélectionnez **Élément dépendant** ici.
Clé	Saisissez une clé qui sera utilisée pour reconnaître l'élément.
Élément maître	Sélectionnez l'élément maître. La valeur de l'élément maître sera utilisée pour renseigner la valeur de l'élément dépendant.
Type d'information	Sélectionnez le type d'information qui correspondra au format des données qui seront stockées.

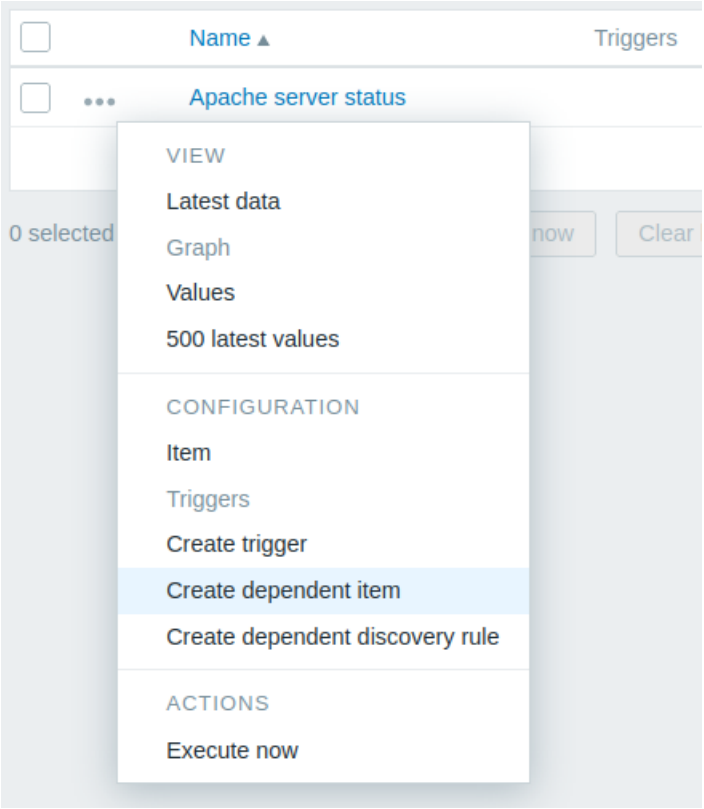
Vous pouvez utiliser le prétraitement de la valeur de l'élément pour extraire la partie requise de la valeur de l'élément maître.



Sans prétraitement, la valeur de l'élément dépendant sera exactement la même que la valeur de l'élément maître.

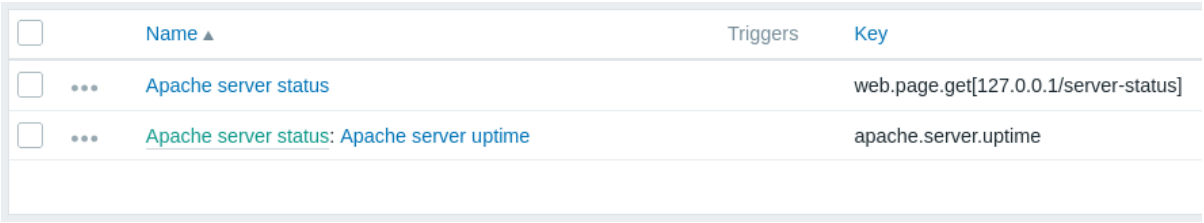
Cliquez sur *Ajouter* pour enregistrer l'élément dépendant.

Un raccourci pour créer plus rapidement un élément dépendant est accessible en cliquant sur le bouton ••• dans la liste des éléments et 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.

16 Agent HTTP

Vue d'ensemble

Ce type d'élément permet l'interrogation des données à l'aide du protocole HTTP/HTTPS. Le trapping est également possible à l'aide de l'utilitaire Zabbix sender ou du protocole Zabbix sender (pour l'envoi de données au serveur ou au proxy Zabbix), ou en utilisant la méthode d'API `history.push` (pour l'envoi de données au serveur Zabbix).

Les vérifications d'éléments HTTP sont exécutées par le serveur Zabbix. Toutefois, lorsque des hôtes sont surveillés par un proxy Zabbix, les vérifications d'éléments HTTP sont exécutées par le proxy.

Les vérifications d'éléments HTTP ne nécessitent aucun agent en cours d'exécution sur un hôte surveillé.

L'agent HTTP prend en charge HTTP et HTTPS. Zabbix peut éventuellement suivre les redirections (voir l'option *Follow redirects* ci-dessous). Le nombre maximal de redirections est codé en dur à 10 (à l'aide de l'option cURL CURLOPT_MAXREDIRS).

> **Attention:**
> Le serveur/proxy Zabbix 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.

Le niveau maximal de concurrence des vérifications asynchrones est de 1000 (défini par MaxConcurrentChecksPerPoller).

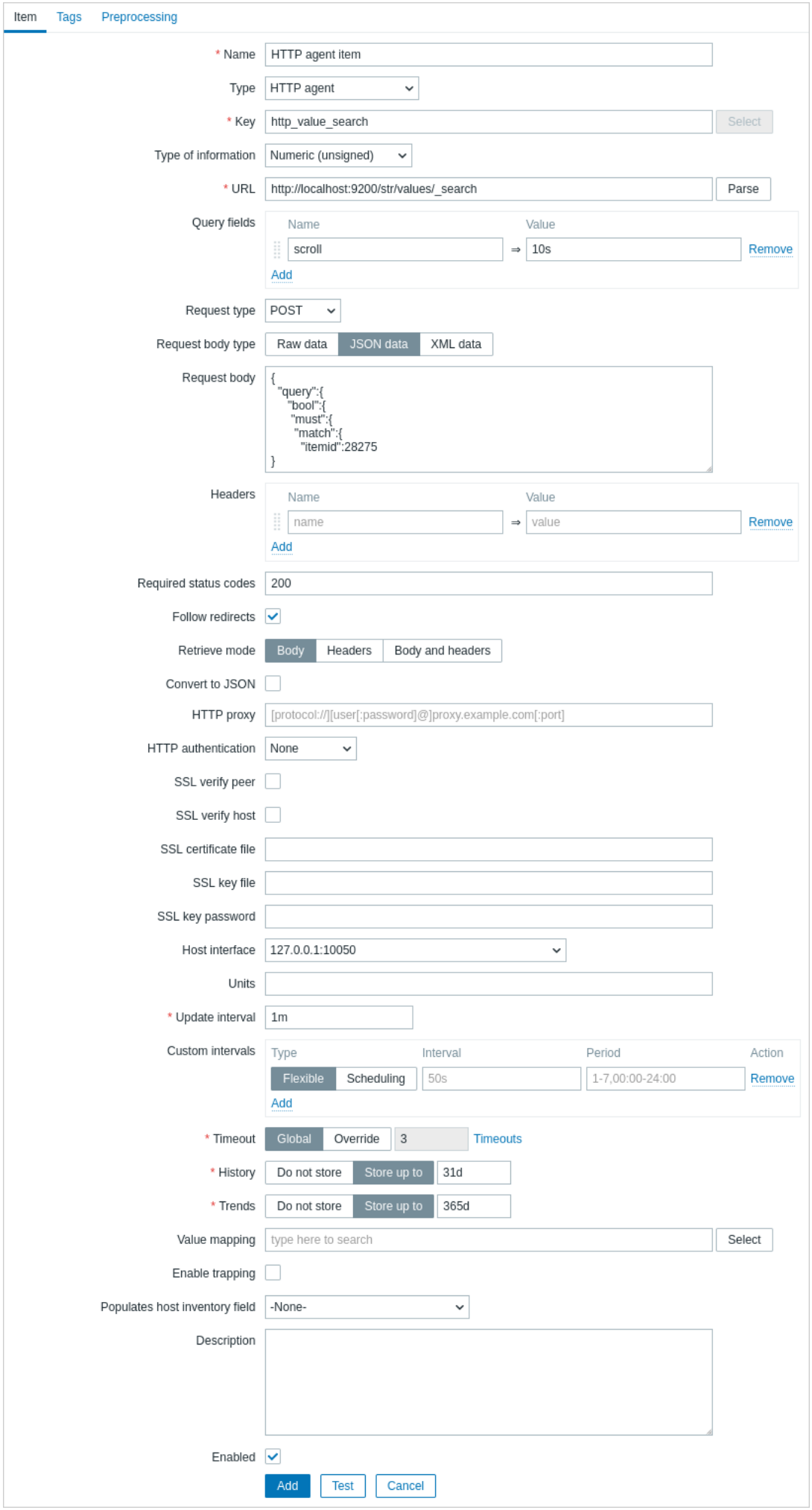
Le nombre de pollers asynchrones de l'agent HTTP est défini par le paramètre StartHTTPAgentPollers.

La fonctionnalité de connexions persistantes de cURL a été ajoutée aux vérifications de l'agent HTTP depuis Zabbix 7.0.

Configuration

Pour configurer un élément HTTP :

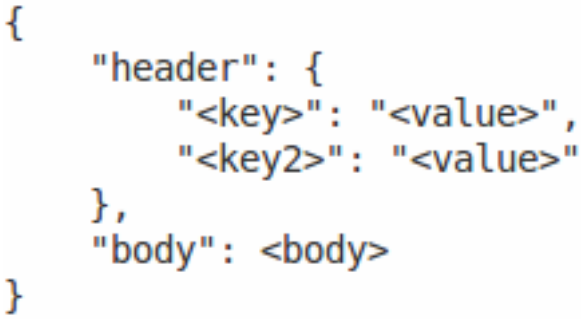
- Allez dans : *Collecte de données* → *Hôtes*
- Cliquez sur *Éléments* sur la ligne de l’hôte
- Cliquez sur *Créer un élément*
- Saisissez les paramètres de l’élément dans le formulaire


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
Action
Flexible
Scheduling
50s
1-7,00:00-24:00
Remove
Add
* Timeout
Global
Override
3
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
Populates host inventory field
-None-
Description
Enabled
Add
Test
Cancel

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

Les champs qui nécessitent des informations spécifiques pour les éléments HTTP sont :

Parameter	Description
Type	Sélectionnez **Agent HTTP** ici.
Key	Saisissez une clé d'élément unique.
URL	URL à laquelle se connecter pour récupérer des 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 vers ASCII lors de l'exécution de la vérification HTTP. Le bouton *Analyser* peut être utilisé pour séparer les champs de requête facultatifs (comme ?name=Admin&password=mypassword) de l'URL, en déplaçant les attributs et les valeurs dans *Champs de requête* pour un encodage URL automatique. 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. Définit l'option cURL CURLOPT_URL.
Query fields	Variables pour l'URL (voir ci-dessus). Spécifiées sous forme de paires attribut/valeur. Les valeurs sont automatiquement encodées en URL. Les valeurs provenant des macros sont résolues puis automatiquement encodées en 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. Définit l'option cURL CURLOPT_URL.
Request type	Sélectionnez le type de méthode de requête : *GET*, *POST*, *PUT* ou *HEAD*
Request body type	Sélectionnez le type de corps de requête : **Données brutes** - corps de requête HTTP personnalisé, les macros sont substituées mais aucun encodage n'est effectué **Données JSON** - corps de 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 des 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 requête HTTP au format XML. Les macros peuvent être utilisées comme nœud de texte, attribut ou section CDATA. Les valeurs provenant des 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"
Request body	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.
Headers	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. Définit l'option cURL CURLOPT_HTTPHEADER.
Required status codes	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 ce champ 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. Utilise l'option cURL CURLINFO_RESPONSE_CODE.
Follow redirects	Cochez cette case pour suivre les redirections HTTP. Définit l'option cURL CURLOPT_FOLLOWLOCATION.
Retrieve mode	Sélectionnez la partie de la réponse à récupérer : **Corps** - corps uniquement **En-têtes** - en-têtes uniquement **Corps et en-têtes** - corps et en-têtes

Parameter	Description
Convert to JSON	Les en-têtes sont enregistrés sous forme de paires attribut/valeur sous la clé "header" si *Mode de récupération* n'est pas défini sur *Corps*. Si 'Content-Type: application/json' est rencontré, le corps est enregistré comme objet ; sinon, il est stocké comme chaîne, par exemple : 
HTTP proxy	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 ex. 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. Si spécifié, le proxy remplacera les variables d'environnement liées au proxy telles que http_proxy, 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 aucun contrôle de validité. *Notez* que seule l'authentification simple est prise en charge avec le 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. Définit l'option cURL CURLOPT_PROXY.
HTTP authentication	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 aussi : Configuration de Kerberos avec Zabbix) ; **Digest** - l'authentification Digest est utilisée. Définit l'option cURL CURLOPT_HTTPAUTH.
User name	Saisissez le nom d'utilisateur (jusqu'à 255 caractères). Ce champ est disponible si *Authentification HTTP* est défini sur Basic, NTLM, Kerberos ou Digest. Les macros utilisateur et les macros de découverte de bas niveau sont prises en charge.
Password	Saisissez le mot de passe utilisateur (jusqu'à 255 caractères). Ce champ est disponible si *Authentification HTTP* est défini sur Basic, NTLM, Kerberos ou Digest. Les macros utilisateur et les macros de découverte de bas niveau sont prises en charge.
SSL verify peer	Cochez cette case pour vérifier le certificat SSL du serveur web. Le certificat du serveur sera automatiquement pris à partir de l'emplacement système global de l'autorité de certification (CA). Vous pouvez remplacer l'emplacement des fichiers CA à l'aide du paramètre de configuration du serveur Zabbix ou du proxy SSLCALocation. Définit l'option cURL CURLOPT_SSL_VERIFYPEER.
SSL verify host	Cochez cette case pour vérifier que le champ Common Name ou le champ Subject Alternate Name du certificat du serveur web correspond. Définit l'option cURL CURLOPT_SSL_VERIFYHOST.

Parameter	Description
SSL certificate file	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 du fichier de clé SSL vide. Si la clé est chiffrée, indiquez le mot de passe dans le champ du 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. Définit l'option cURL CURLOPT_SSLCERT.
SSL key file	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. Définit l'option cURL CURLOPT_SSLKEY.
SSL key password	Mot de passe du fichier de clé privée SSL. Macros prises en charge : macros utilisateur, macros de découverte de bas niveau. Définit l'option cURL CURLOPT_KEYPASSWD.
Timeout	Zabbix ne passera pas plus que le temps défini à traiter l'URL (1 à 600 secondes). En réalité, ce paramètre définit le temps maximal pour établir une connexion à l'URL et le temps maximal pour exécuter une requête HTTP. Par conséquent, Zabbix ne passera pas plus de 2 x *Timeout* secondes sur une vérification. Définit l'option cURL CURLOPT_TIMEOUT. Pour plus d'informations sur le paramètre *Timeout*, voir attributs généraux des éléments.
Enable trapping	Si cette case est cochée, l'élément fonctionnera également comme un élément trapper et acceptera les données envoyées au serveur Zabbix ou au proxy à l'aide de l'utilitaire Zabbix sender ou du protocole Zabbix sender, ou acceptera les données envoyées au serveur Zabbix à l'aide de la méthode API `history.push`. Pour plus d'informations sur l'envoi de données, voir : Éléments trapper.
Allowed hosts	Visible uniquement si la case *Activer le trapping* est cochée. Liste d'adresses IP séparées par des virgules, éventuellement en notation CIDR, ou de noms DNS. Si spécifié, les connexions entrantes seront acceptées uniquement 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 traitées de manière identique 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 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. Les macros d'hôte : {HOST.HOST}, {HOST.NAME}, {HOST.IP}, {HOST.DNS}, {HOST.CONN} sont autorisées dans ce champ.

Note:
Si le champ *HTTP proxy* est laissé vide, une autre façon d'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`. Plus de détails sont disponibles en exécutant la commande shell : *# man curl*.

Attention:
[1] Zabbix prend en charge uniquement les fichiers de certificat et de clé privée au format PEM. Si vos données de certificat et de clé privée sont dans un fichier au format PKCS #12 (généralement avec l'extension *.p12 ou *.pfx), vous pouvez générer le fichier PEM à partir de celui-ci 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 des requêtes GET simples pour récupérer des données à partir de services tels que Elasticsearch :

- Créez un élément GET avec l'URL : `localhost:9200/?pretty`
- Notez 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 des requêtes POST simples pour récupérer des données à partir de 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 minute 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,
```

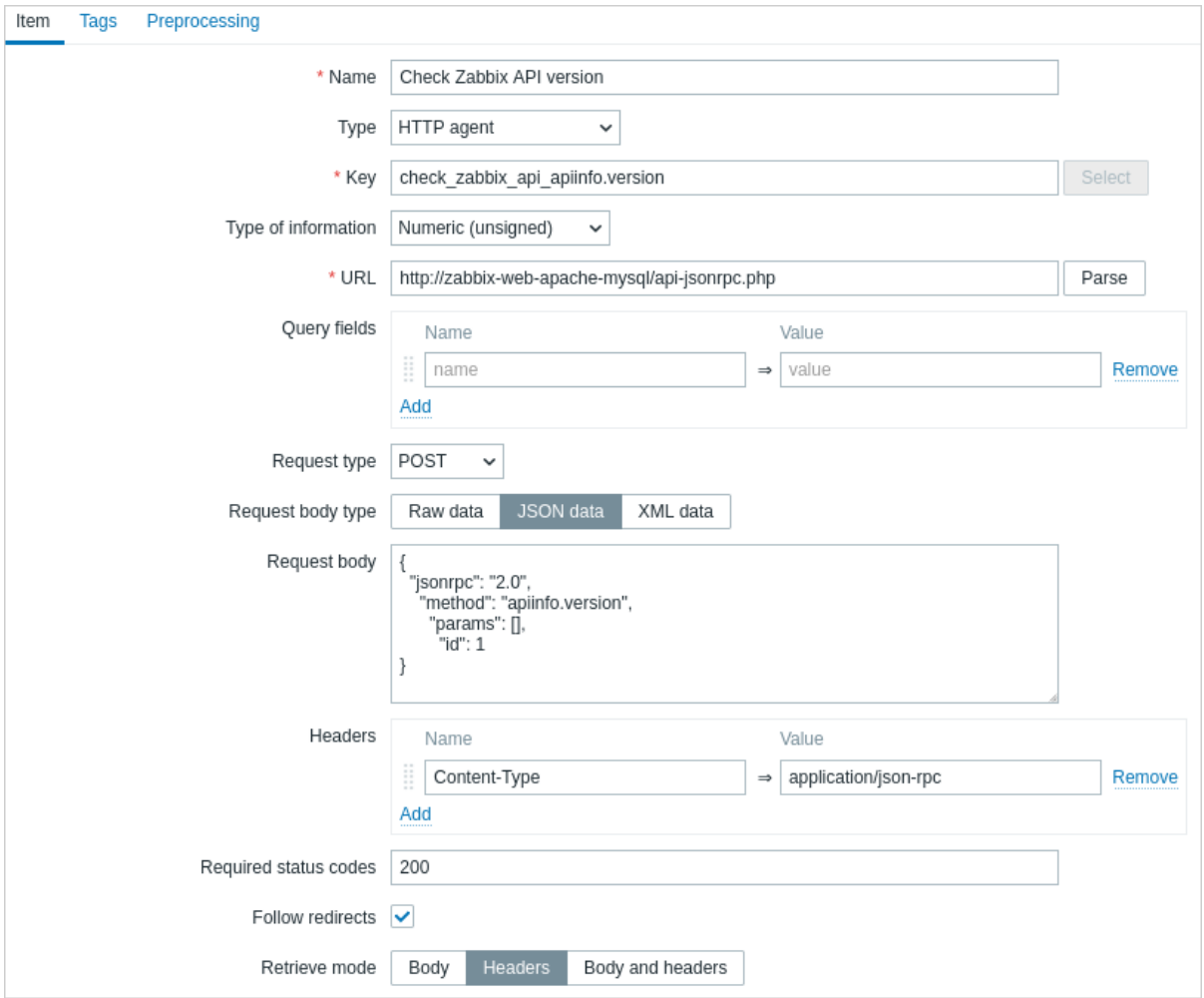
```
            "max_score": 1.0,
            "hits": [{
                "_index": "dbl",
                "_type": "values",
                "_id": "dqX9VWEBV6sEKSMyk6sw",
                "_score": 1.0,
                "_source": {
                    "itemid": 28275,
                    "value": "0.138750",
                    "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

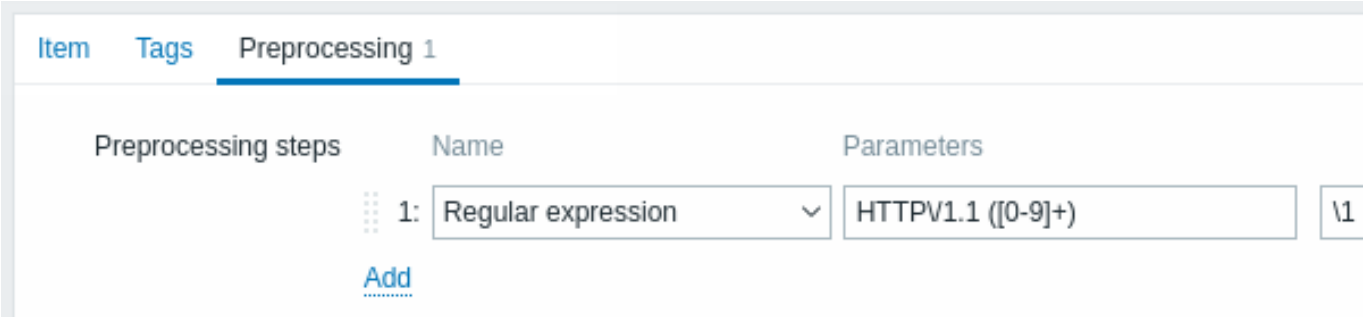
Vérification que l'API Zabbix est active, à l'aide de apiinfo.version.

- Configuration de l'élément :

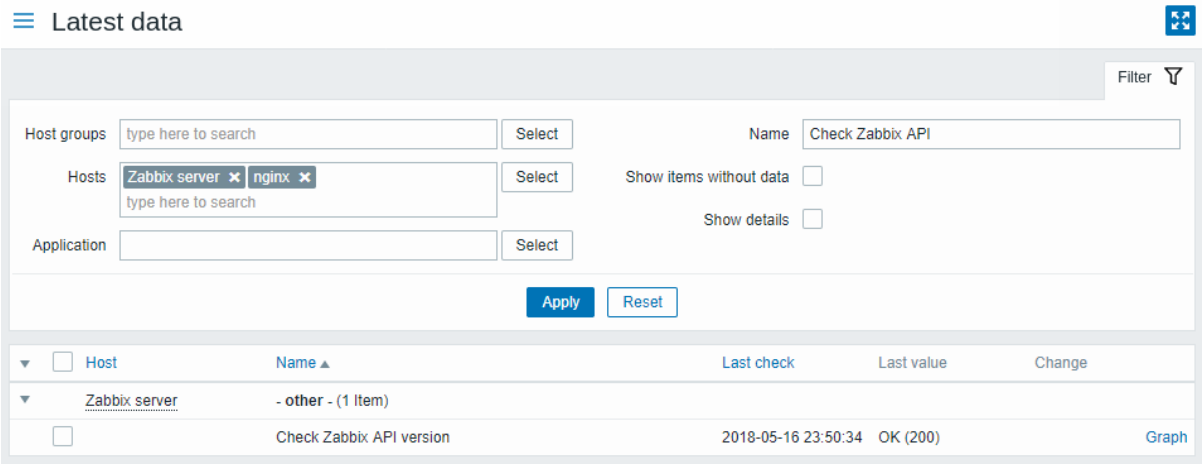


Notez l'utilisation de la méthode POST avec des données JSON, la définition des en-têtes de requête et la demande de ne renvoyer que les en-têtes :

- Prétraitement de la valeur de l'élément avec une expression régulière pour obtenir le code HTTP :



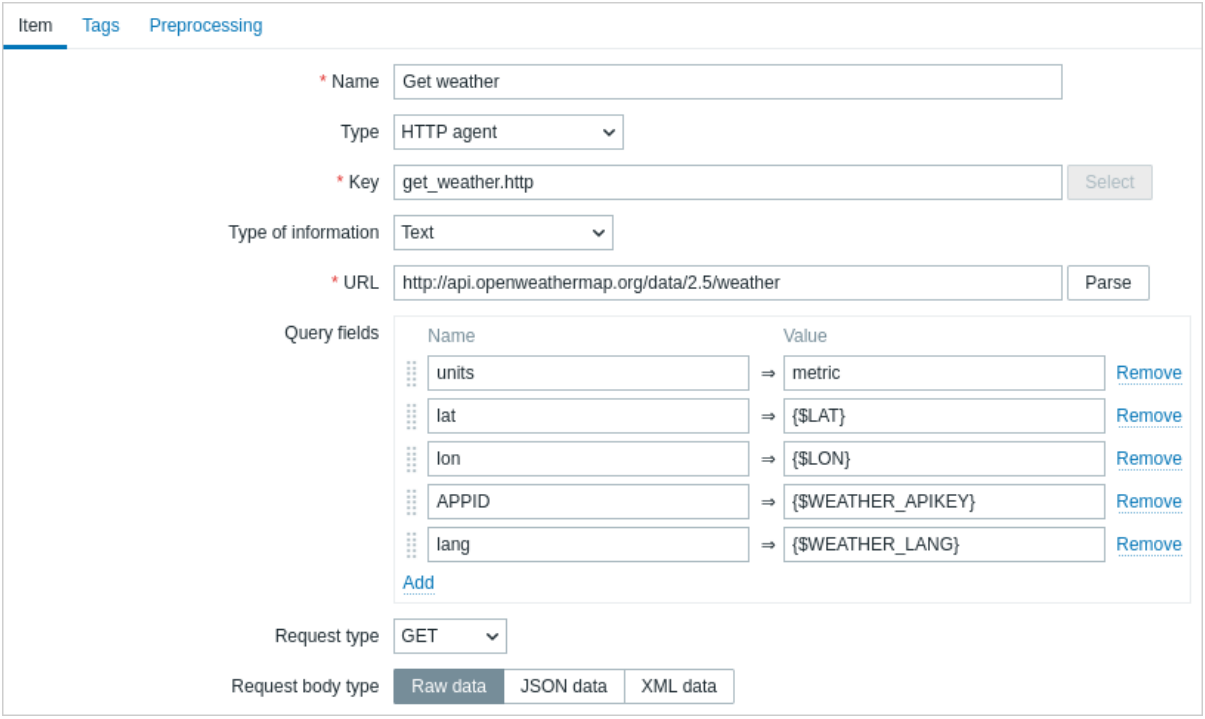
- Vérification du résultat dans *Dernières données* :



Exemple 4

Récupération des informations météorologiques en se connectant au service public Openweathermap.

- Configurez un élément maître pour la collecte groupée des données dans un seul JSON :



Notez l'utilisation des macros dans les champs de requête. Consultez l'API Openweathermap pour savoir comment les renseigner.

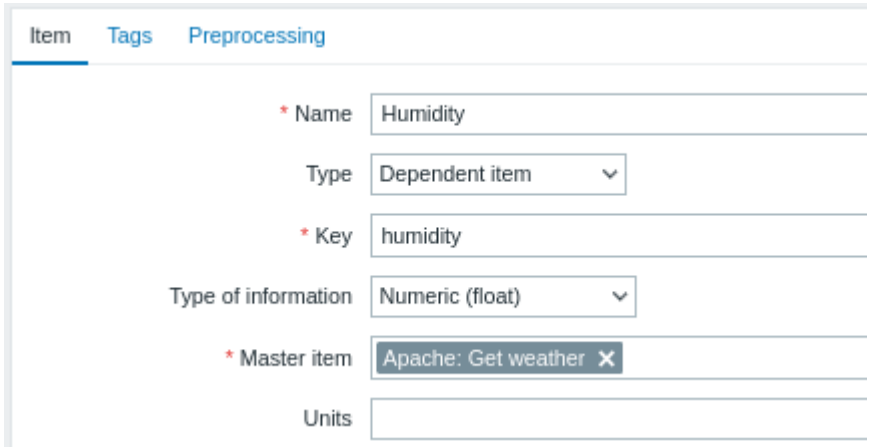
Exemple de JSON renvoyé en réponse à l'agent HTTP :

```
{
    "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
    }
}
```

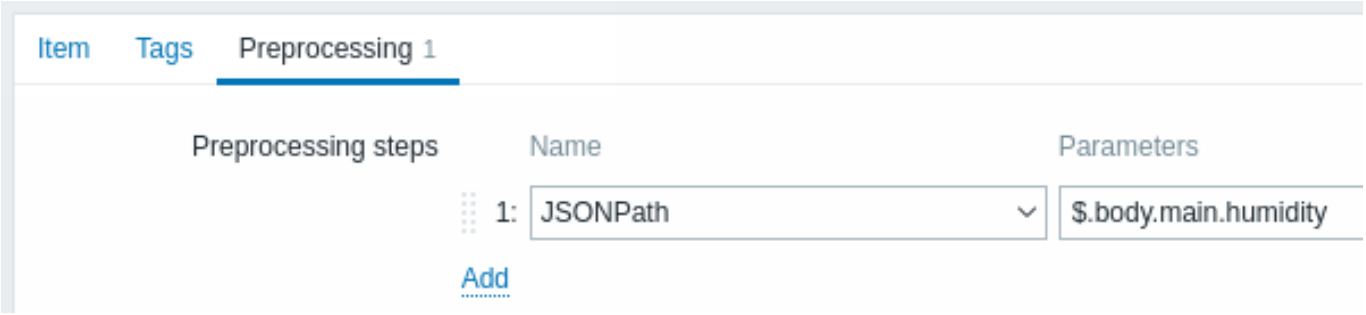

L'étape suivante consiste à configurer des éléments dépendants qui extraient les données du JSON.

- Configurez un exemple d'élément dépendant pour l'humidité :

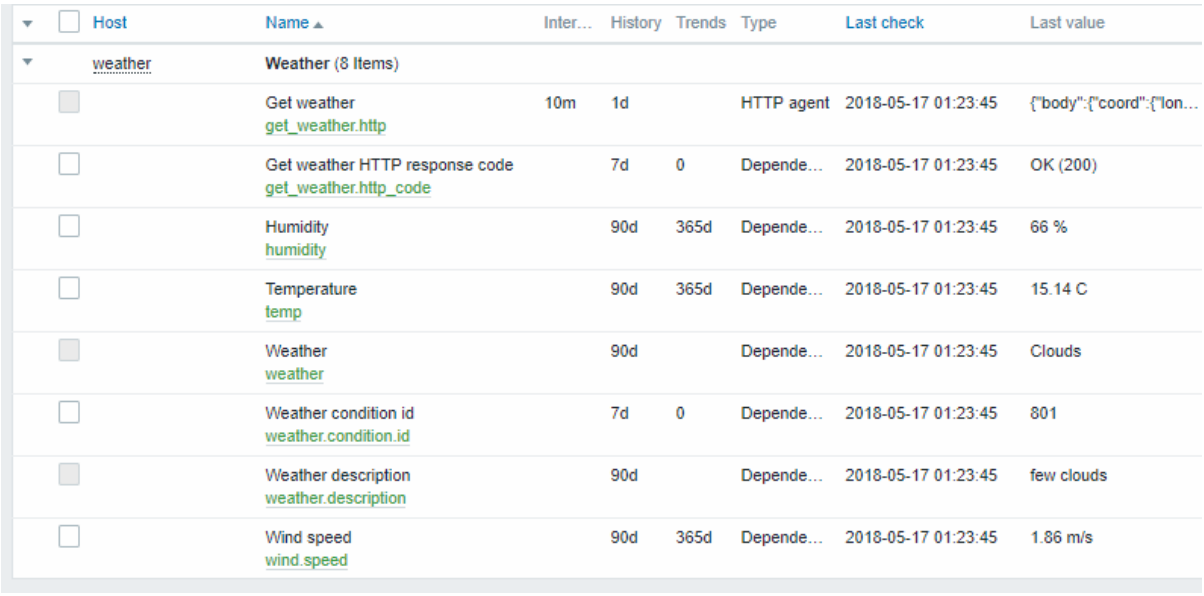


D'autres métriques météorologiques telles que « Température » sont ajoutées de la même manière.

- Exemple de prétraitement de la valeur d'un élément dépendant avec JSONPath :



- Vérifiez le résultat des données météorologiques dans *Dernières données* :

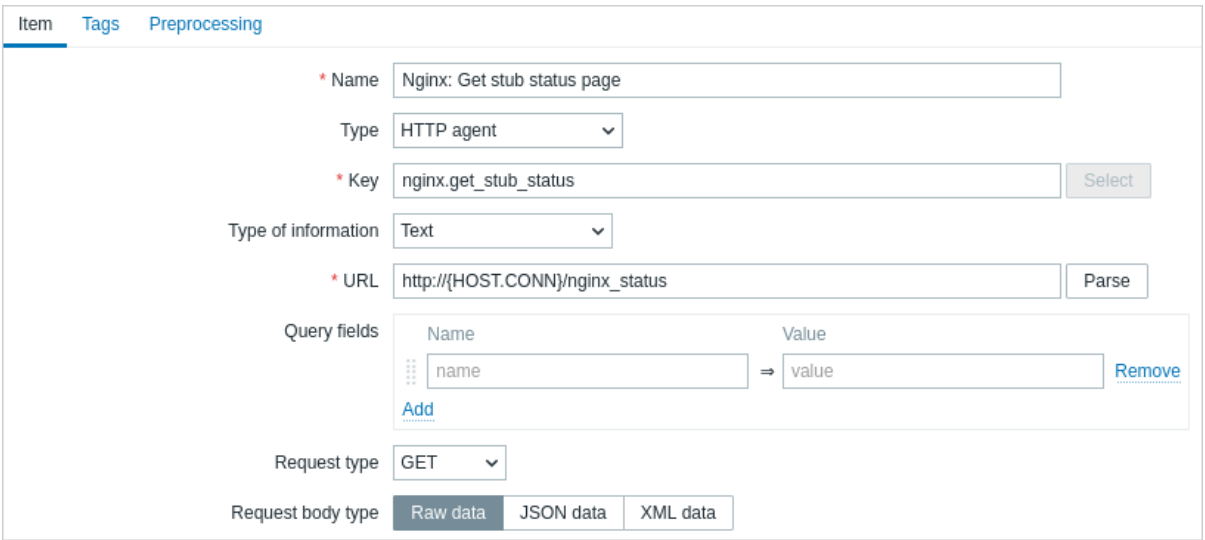


Host	Name	Inter...	History	Trends	Type	Last check	Last value
weather	**Weather** (8 Items)						
	Get weather get_weather.http	10m	1d		HTTP agent	2018-05-17 01:23:45	{"body":{"coord":{"lon...
	Get weather HTTP response code get_weather.http_code		7d	0	Depende...	2018-05-17 01:23:45	OK (200)
	Humidity 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 de 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 :

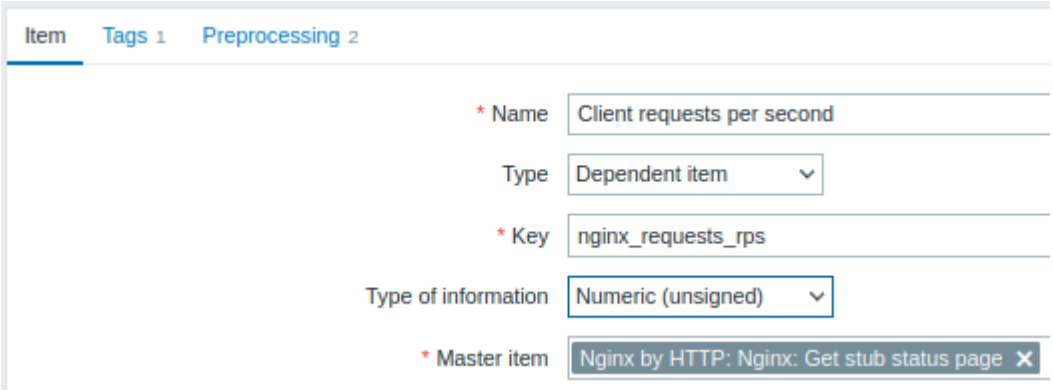


Exemple de sortie de l'état stub de Nginx :

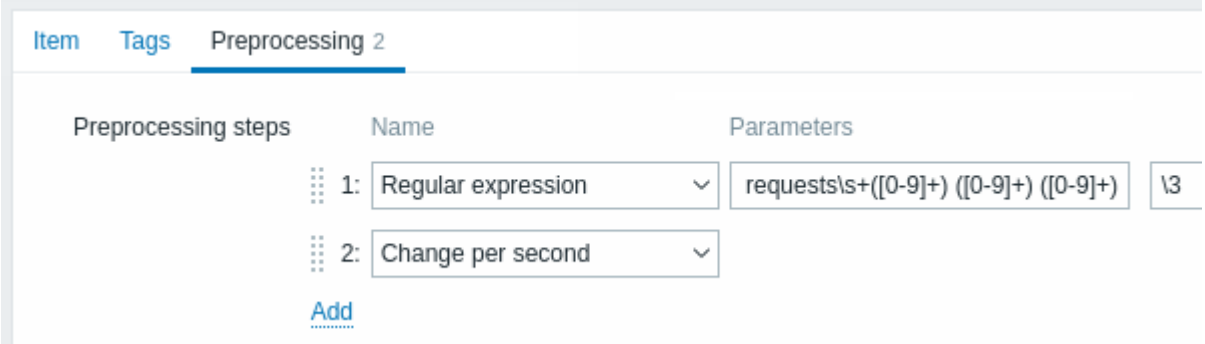
```
Active connections: 1 Active connections:
server accepts handled requests
 52 52 52
Reading: 0 Writing: 1 Waiting: 0
```

La tâche suivante consiste à configurer des éléments dépendants qui extraient les données.

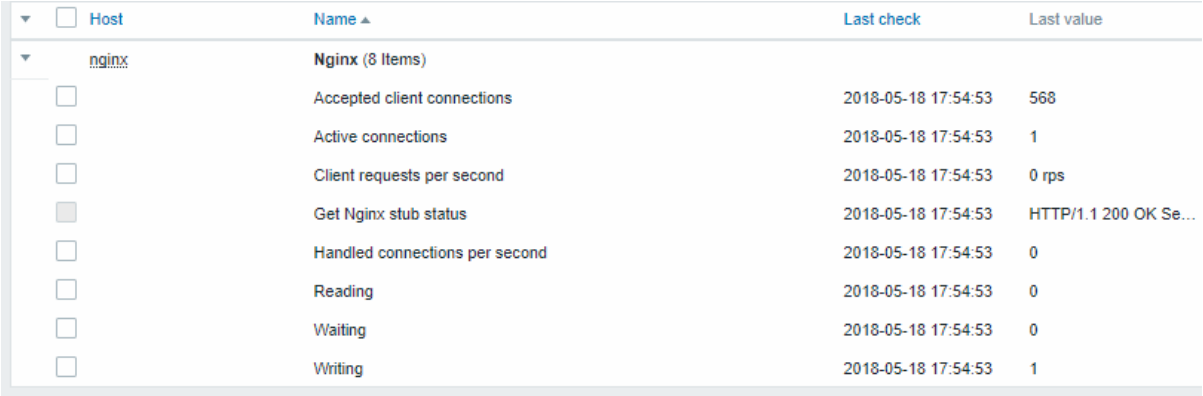
- Configurez un exemple d'élément dépendant pour les requêtes par seconde :



- 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 *Données récentes* :



17 Vérifications Prometheus

Aperçu

Zabbix peut interroger les 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 de masse

Le traitement de masse est pris en charge pour les éléments dépendants. Pour activer la mise en cache et l'indexation, le prétraitement *Motif Prometheus* doit être la **première** étape de prétraitement. Lorsque *Motif Prometheus* est la première étape de prétraitement, les données Prometheus analysées sont mises en cache et indexées par la première condition `<label>==<value>`

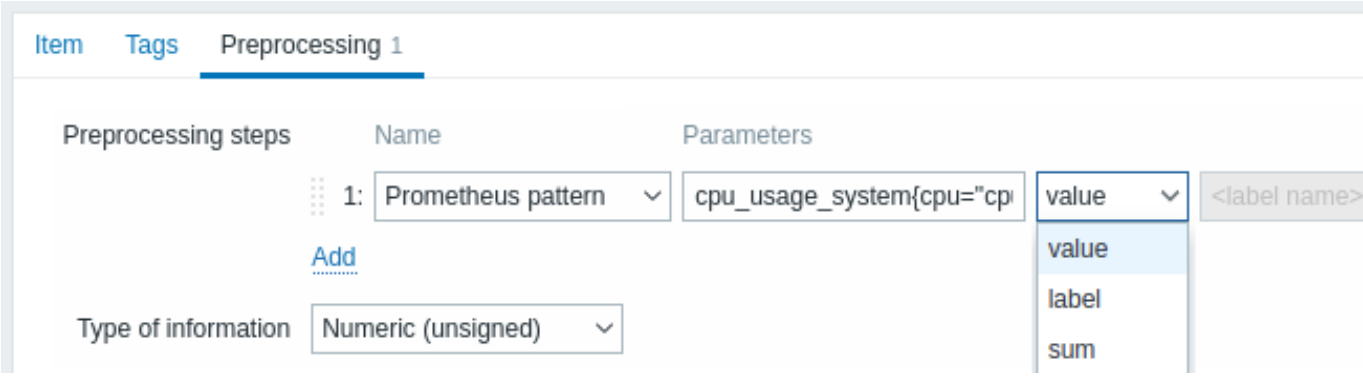
dans l'étape de prétraitement *Motif Prometheus*. Ce cache est réutilisé lors du traitement d'autres éléments dépendants dans ce lot. Pour des performances optimales, la première étiquette doit être celle avec les valeurs les plus différentes.

S'il y a un autre prétraitement à effectuer avant la première étape, il doit être déplacé soit vers l'élément principal, soit vers un nouvel élément dépendant qui serait utilisé comme élément principal pour les éléments dépendants.

Configuration

Si l'élément maître HTTP est configuré, vous devez créer un élément dépendant qui utilise une étape de prétraitement Prometheus :

- Saisissez les paramètres généraux de l'élément dépendant dans le formulaire de configuration
- Accédez à l'onglet Prétraitement
- Sélectionnez une option de prétraitement Prometheus (*Motif Prometheus* ou *Prometheus vers JSON*)



Les paramètres suivants sont spécifiques à l'option de prétraitement *Motif Prometheus* :

Paramètre	Description	Exemples
Motif	Pour définir le modèle de données requis, vous pouvez utiliser un langage de requête similaire au langage de requête Prometheus (voir la table de comparaison), par exemple : <nom métrique> - sélectionner par nom de métrique {__name__="<nom métrique>"} - sélectionner par nom de métrique {__name__=~"<regex>"} - sélectionner par nom de métrique correspondant à une expression régulière {<nom étiquette>="<\valeur étiquette>",...} - sélectionner par nom d'étiquette {<nom étiquette>=~"<regex>",...} - sélectionner par nom d'étiquette correspondant à une expression régulière {__name__=~".*"}==<valeur> - sélectionner par valeur de métrique Ou une combinaison des éléments ci-dessus : <nom métrique>{<nom étiquette1>="<valeur étiquette1>",<nom étiquette2>=~"<regex>",...}==<valeur> La valeur de l'étiquette peut être n'importe quelle séquence de caractères UTF-8, mais les caractères de barre oblique inverse, de guillemet double et de saut de ligne doivent être échappés sous la forme \\, \" et \n respectivement ; 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
Traitement du résultat	Indiquez s'il faut renvoyer la valeur, l'étiquette ou appliquer la fonction appropriée (si le modèle correspond à plusieurs lignes et que le résultat doit être agrégé) : **valeur** - renvoie la valeur de la métrique (erreur si plusieurs lignes correspondent) **étiquette** - renvoie la valeur du libellé spécifié dans le champ *Étiquette* (erreur si plusieurs métriques correspondent) **sum** - renvoie la somme des valeurs **min** - renvoie la valeur minimale **max** - renvoie la valeur maximale **avg** - renvoie la valeur moyenne **count** - renvoie le nombre de valeurs Ce champ n'est disponible que pour l'option *Motif Prometheus*.	Voir aussi des exemples d'utilisation de paramètres ci-dessous.
Sortie	Définir le nom de l'étiquette (optionnel). Dans ce cas, la valeur correspondant au nom de l'étiquette est renvoyée. Ce champ n'est disponible que pour l'option *Motif Prometheus*, si 'Étiquette' est sélectionné dans le champ *Traitement du résultat*.	

Exemples d'utilisation de paramètres

1. Le cas d'usage le plus courant consiste à renvoyer la **valeur**. Pour renvoyer la valeur de `/var/db` à partir de :

   `node_disk_usage_bytes{path="/var/cache"} 2.1766144e+09`
`node_disk_usage_bytes{path="/var/db"} 20480`
`node_disk_usage_bytes{path="/var/dpkg"} 8192`
`node_disk_usage_bytes{path="/var/empty"} 4096`

utilisez les paramètres suivants :

- *Motif* - `node_disk_usage_bytes{path="/var/db"}`
- *Traitement du résultat* - sélectionnez 'valeur'

2. Vous pouvez également être intéressé par la valeur **moyenne** de tous les paramètres `node_disk_usage_bytes` :

- *Motif* - `node_disk_usage_bytes`
- *Traitement du résultat* - sélectionnez 'avg'

3. Alors que Prometheus ne prend en charge que les données numériques, il est courant d'utiliser une solution de contournement qui permet également de renvoyer la description textuelle pertinente. Cela peut être accompli avec un filtre et en spécifiant l'étiquette. Ainsi, pour renvoyer la valeur de l'étiquette 'color' de

   `elasticsearch_cluster_health_status{cluster="elasticsearch",color="green"} 1`
`elasticsearch_cluster 0`
`elasticsearch_cluster_health_status{cluster="elasticsearch",color="yellow"} 0`

utilisez les paramètres suivants :

- *Motif* - `elasticsearch_cluster_health_status {cluster="elasticsearch"} == 1`
- *Traitement du résultat* - sélectionnez 'étiquette'
- *Étiquette* - spécifiez 'color'

Le filtre (basé sur la valeur numérique '1') correspondra à la ligne appropriée, tandis que l'étiquette renverra 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 vers JSON* renverra exactement cela.

Pour plus de détails, voir Découverte à l'aide des données Prometheus.

Comparaison des langages de requête

Le tableau suivant répertorie les différences et les similitudes entre le langage de requête de prétraitement PromQL et Zabbix Prometheus.

PromQL instant vector selector		Prétraitement Zabbix Prometheus
Différences		
Cible de la requête	Serveur Prometheus	Texte brut au format d'exposition Prometheus
Renvois	Vecteur instantané	Valeur de métrique ou d'étiquette (modèle Prometheus) Tableau de métriques pour une valeur unique dans JSON (Prometheus vers JSON)
Opérateurs de correspondance d'étiquettes	**=**, **!=**, **=~**, **!~**	**=**, **!=**, **=~**, **!~**
Expression régulière utilisée dans la correspondance de libellé ou de nom de métrique	RE2	PCRE
Opérateurs de comparaison	Voir la liste	Seulement **==** (égal) est pris en charge pour le filtrage de valeur
Similitudes		
Sélection par nom de métrique égal à chaîne	<nom de métrique> or {__nom__="<nom de métrique>"}	<nom de métrique> or {__nom__="<nom de métrique>"}
Sélection par nom de métrique qui correspond à l'expression régulière	{__nom__=~"<regex>"}	{__nom__=~"<regex>"}
Sélection par <nom d'étiquette> valeur égale à string	{<nom d'étiquette>="<valeur d'étiquette>",...}	{<nom d'étiquette>="<valeur d'étiquette>",...}
Sélection par la valeur <nom d'étiquette> qui correspond à l'expression régulière	{<nom d'étiquette>=~"<regex>",...}	{<nom d'étiquette>=~"<regex>",...}
Sélection par valeur égale à chaîne	{__nom__=~".*"} == <valeur>	{__nom__=~".*"} == <valeur>

18 Éléments de script

Aperçu

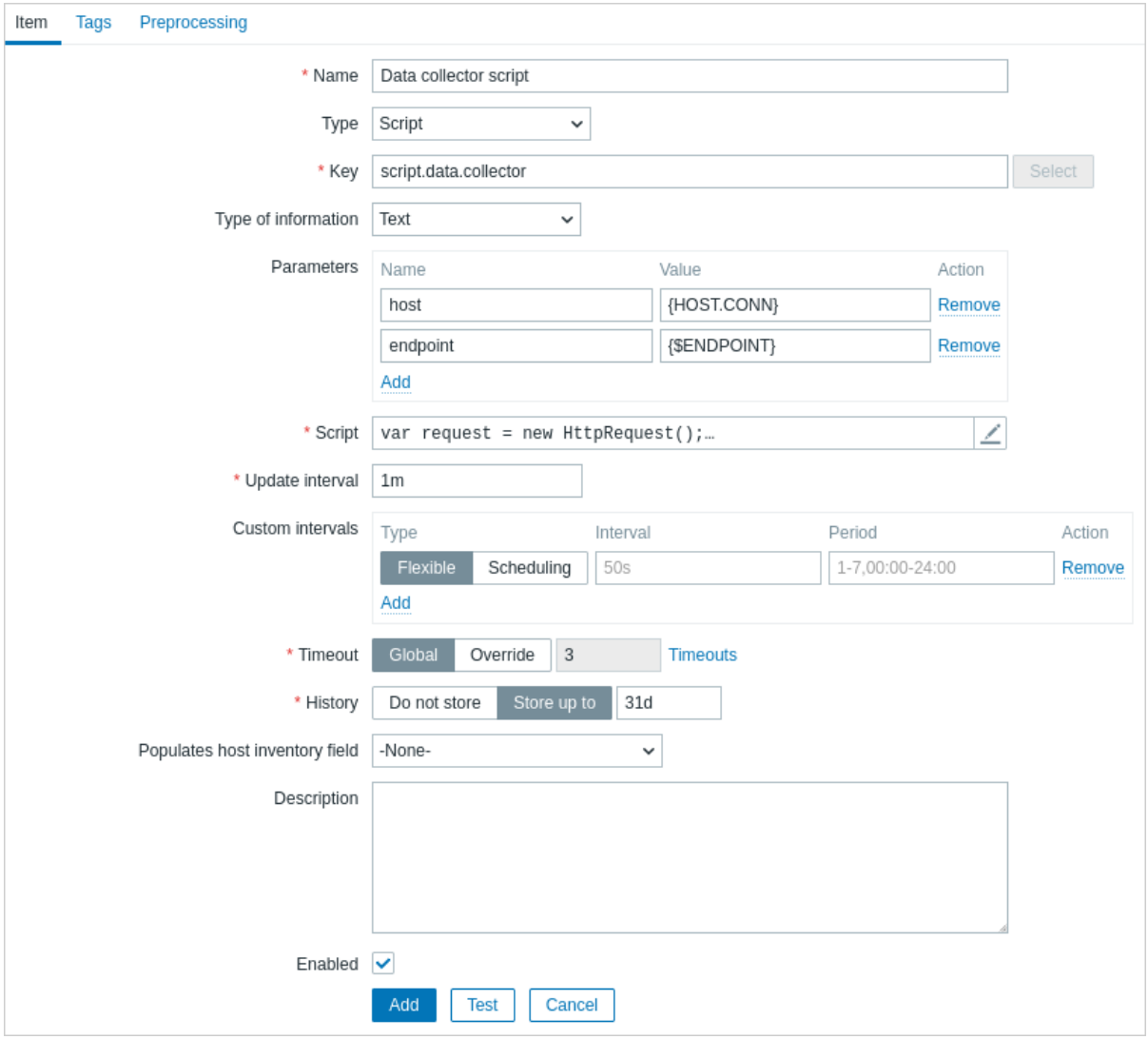
Les éléments de type script peuvent être utilisés pour collecter des données en exécutant un code JavaScript défini par l'utilisateur, avec la possibilité de récupérer des données via HTTP/HTTPS. En plus du script, une liste facultative de paramètres (paires de nom et de valeur) ainsi qu'un délai d'expiration 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*:

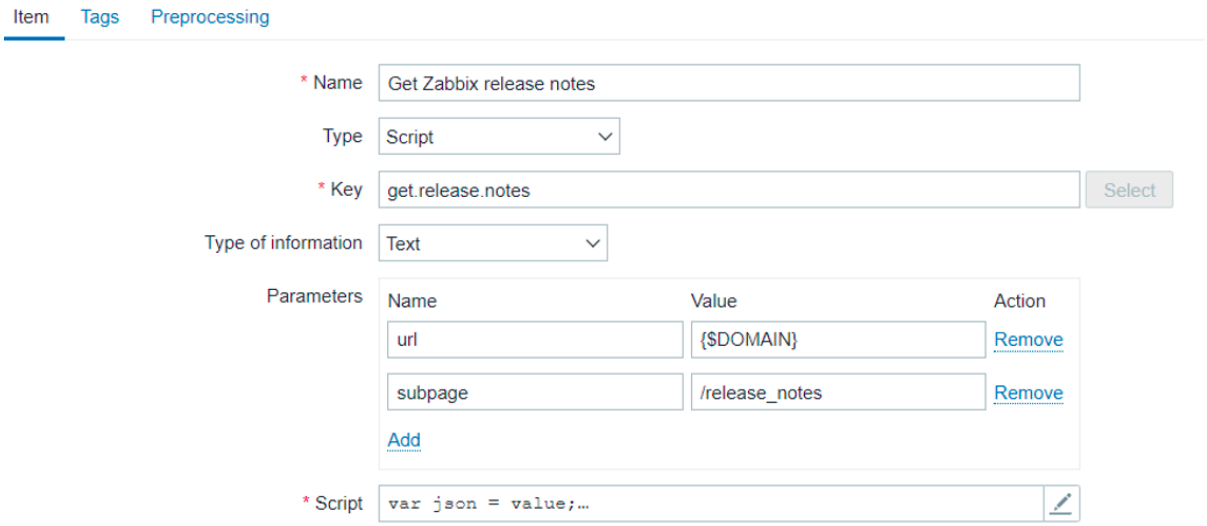
- Créer un élément de type "Script".
- Dans le champ *Script*, saisir :

```
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;
```

19 Éléments de navigateur

Aperçu

Les éléments de navigateur permettent de surveiller des sites web complexes et des applications web 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 un 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 entre des pages web et d'autres interactions utilisateur avec des sites web ou des applications web.

En plus du script, il est possible de spécifier une liste facultative de paramètres (paires de nom et de valeur) ainsi qu'un délai d'expiration.

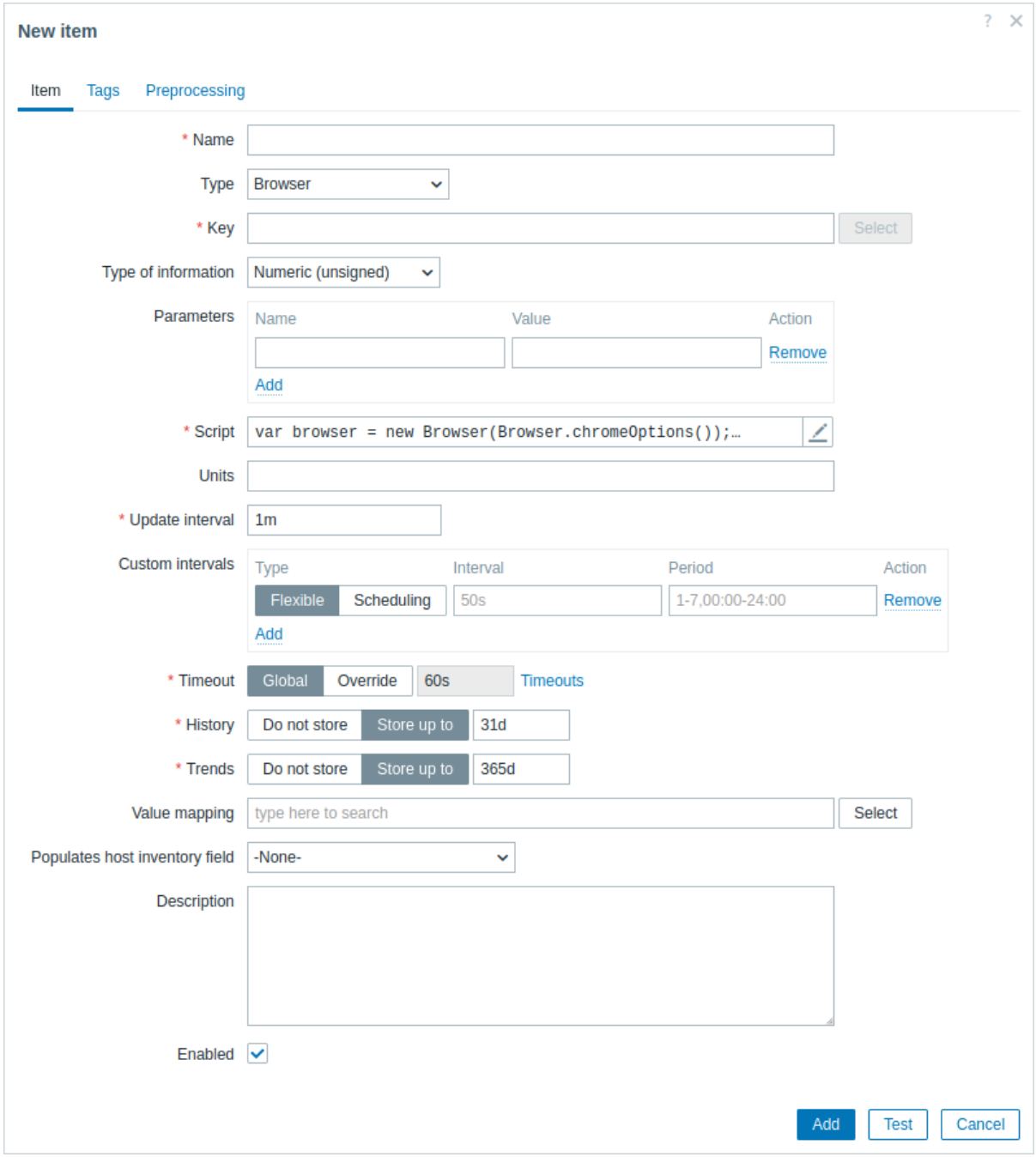
> **Attention:**
> L'élément implémente partiellement la norme W3C WebDriver avec soit Selenium Server, soit un WebDriver simple (par exemple, ChromeDriver) comme point de terminaison de test web. Pour que l'élément fonctionne, définissez le point de terminaison dans le paramètre `WebDriverURL` du fichier de configuration du server/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 processus browser poller du serveur ou du proxy Zabbix. Si nécessaire, vous pouvez ajuster le nombre d'instances pré-forkées des browser pollers dans le paramètre `StartBrowserPollers` du fichier de configuration du server/proxy Zabbix.

Pour surveiller des sites web complexes et des applications web, 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 afin de surveiller des 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);
}
```

manual/config/items/itemtypes/snmptrap

Vue d'ensemble

La réception des traps SNMP est l'opposé de l'interrogation des périphériques compatibles SNMP.

Dans ce cas, les informations sont envoyées depuis un périphérique compatible SNMP vers snmptrapd et sont collectées ou « piégées » par Zabbix server ou Zabbix proxy à partir d'un fichier.

En général, les traps sont envoyés lors d'un changement de condition et l'agent se connecte au serveur sur le port 162 (par opposition au port 161 du côté de l'agent, qui est utilisé pour les requêtes). L'utilisation des traps peut permettre de détecter certains problèmes de courte durée qui surviennent entre deux intervalles d'interrogation 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 à 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 est d'utiliser la solution basée sur un 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 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 processus SNMP trapper de Zabbix lit et analyse le fichier de traps
5. Pour chaque trap, Zabbix recherche tous les éléments « SNMP trapper » dont les interfaces hôte correspondent à l'adresse du trap reçu. Notez que seule la valeur « IP » ou « DNS » sélectionnée dans l'interface hôte est utilisée 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 valeur de celui-ci.
7. Si le trap n'a été défini comme valeur d'aucun élément, Zabbix enregistre par défaut le trap non correspondant dans le journal. (Cela se configure avec « Log unmatched SNMP traps » dans Administration > General > Other.)

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 du frontend suivante.

1. Créez une interface SNMP pour votre hôte

   - Dans *Collecte de données* > *Hôtes*, créez/modifiez l'hôte et, dans le champ *Interfaces*, ajoutez le type d'interface « SNMP » en spécifiant l'adresse IP ou DNS.

L'adresse de chaque trap reçu sera comparée aux adresses IP et DNS de toutes les interfaces SNMP afin de trouver les hôtes correspondants.

2. Configurez l'élément

   - Dans *Collecte de données* > *Hôtes*, créez/modifiez l'élément nécessaire.
   - Dans le champ *Clé*, utilisez l'une des clés de trap SNMP :

Clé		
Description	Valeur de retour	Commentaires
snmptrap[regexp] Capture tous les traps SNMP correspondant à l'expression régulière spécifiée dans **regexp**. Si regexp n'est pas spécifié, capture n'importe quel trap.	Trap SNMP	Cet élément ne peut être défini que pour les interfaces SNMP. Les macros utilisateur et les expressions régulières globales sont prises en charge dans le paramètre de cette clé d'élément.
snmptrap.fallback Capture tous les traps SNMP qui n'ont été capturés par aucun des éléments snmptrap[] pour cette interface.	Trap SNMP	Cet élément ne peut être défini que pour les interfaces SNMP.

Note:
La correspondance d'expressions régulières multilignes n'est actuellement pas prise en charge.

   - Définissez le *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 traps 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 dans lequel SNMPTT ou un récepteur de traps Bash/Perl écrit. Pour cela, 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

Prérequis : uniquement snmptrapd.

Un script de récepteur de traps Bash peut être utilisé pour transmettre les 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 directement au serveur Zabbix 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:
Il peut être nécessaire de redémarrer snmptrapd pour que les modifications de sa configuration soient prises en compte.

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

Tout d'abord, snmptrapd doit être configuré pour utiliser SNMPTT.

Note:
Pour obtenir les meilleures performances, SNMPTT doit être configuré comme un démon en utilisant **snmptthandler-embedded** pour lui transmettre les traps. Voir les instructions de configuration de SNMPTT.

Lorsque SNMPTT est configuré pour recevoir les traps, configurez `snmptt.ini` :

1. activez l'utilisation du module Perl du paquet NET-SNMP :

```
net_snmp_perl_enable = 1
```

2. journalisez 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 paquet "net-snmp-perl" a été supprimé dans RHEL 8.0-8.2 ; réintroduit 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 dans 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 d'informations sur le format des traps SNMP ci-dessous.

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 des traps SNMP

Tous les récepteurs de traps Perl personnalisés et la configuration des traps SNMPTT doivent formater le trap de la manière suivante :

```
[timestamp] [le trap, partie 1] ZBXTRAP [address] [le trap, partie 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 correspondant à ce trap

Notez que "ZBXTRAP" et "[address]" seront supprimés du message pendant le traitement. Si le trap est formaté différemment, 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 fichiers volumineux

Zabbix prend en charge les fichiers volumineux pour les fichiers de trapper SNMP. La taille limite maximum que Zabbix peut lire est 2^63 (8 EiB). Notez que le système de fichier peut imposer une limite inférieure sur 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 le supprimer seulement ensuite, afin qu'aucun trap ne soit perdu :

1. Zabbix ouvre le fichier de traps au dernier emplacement connu et passe à l'étape 3.
2. Zabbix vérifie si le fichier actuellement ouvert a été pivoté en comparant le numéro d'inode au numéro d'inode du fichier de traps défini. S'il n'y a aucun fichier ouvert, Zabbix réinitialise le dernier emplacement et passe à l'étape 1.
3. Zabbix lit les données du fichier actuellement ouvert et définit le nouvel emplacement.
4. Les nouvelles données sont analysées. S'il s'agissait du fichier pivoté, le fichier est fermé et le processus revient à l'étape 2.
5. S'il n'y a pas de nouvelles données, Zabbix attend 1 seconde et revient à l'étape 2.

Système de fichiers

En raison de l'implémentation du fichier de trap, Zabbix a besoin du système de fichiers pour prendre en charge les inodes afin de différencier les différents fichiers (les informations sont acquises 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 SNMP trapper et définir le fichier de traps. Ajoutez à `zabbix_server.conf` :

```
StartSNMPTrapper=1
SNMPTrapperFile=/var/lib/zabbix/snmptraps/snmptraps.log
```

2. Téléchargez le script Bash dans `/usr/sbin/zabbix_trap_handler.sh` :

```
curl -o /usr/sbin/zabbix_trap_handler.sh https://raw.githubusercontent.com/zabbix/zabbix-docker/trunk/Dock
```

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 ce qui suit à `snmtrapd.conf` (voir l'exemple fonctionnel)

```
traphandle default /bin/bash /usr/sbin/zabbix_trap_handler.sh
```

> **Note:**
> Il peut être nécessaire de redémarrer snmptrapd pour prendre en compte les modifications de sa configuration.

4. Créez un élément SNMP TEST (gardez à l'esprit les exigences de configuration initiales):

   Type : SNMP trap
   Type d'information : Log Interface 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. Ensuite, nous allons 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 reposent sur une authentification par « chaîne de communauté ». Dans l'exemple ci-dessous, nous utiliserons « secret » comme chaîne de communauté. Elle doit être définie à la même valeur sur les expéditeurs de traps SNMP.

Veuillez noter que, bien qu'encore largement utilisé dans les environnements de production, SNMPv2 n'offre ni chiffrement ni authentification réelle de l'expéditeur. Les données sont envoyées en texte brut et, par conséquent, ces versions du protocole ne doivent être utilisées que 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 pratiquement plus utilisée de nos jours, car elle ne prend pas en charge les compteurs 64 bits et est considérée comme un protocole hérité.

Pour autoriser la réception des traps SNMPv1 ou SNMPv2, vous devez ajouter la ligne suivante à `snmptrapd.conf`. Remplacez « secret » par la chaîne de communauté SNMP configurée sur les expéditeurs de traps SNMP :

```
authCommunity log,execute,net secret
```

Ensuite, nous pouvons envoyer un trap de test à l'aide de `snmptrap`. Nous utiliserons dans cet exemple l'OID commun « link up » :

```
snmptrap -v 2c -c secret localhost '' linkUp.0
```

SNMPv3

SNMPv3 corrige les problèmes de sécurité de SNMPv1/v2 et fournit authentification et chiffrement. Vous pouvez utiliser MD5 ou plusieurs méthodes d'authentification SHA, ainsi que DES ou plusieurs variantes d'AES comme chiffrement.

Pour autoriser la réception de SNMPv3, ajoutez les lignes suivantes à `snmptrapd.conf` :

```
createUser -e 0x8000000001020304 traptest SHA mypassword AES
authuser log,execute traptest
```

> **Attention:**
> Veuillez noter le mot-clé « execute », qui permet d'exécuter des scripts pour ce modèle de sécurité utilisateur.

```
snmptrap -v 3 -n "" -a SHA -A mypassword -x AES -X mypassword -l authPriv -u traptest -e 0x800000000102030
```

> **Warning:**
> Si vous souhaitez utiliser des méthodes de chiffrement fortes telles que AES192 ou AES256, veuillez utiliser net-snmp à partir de la version 5.8. Il se peut que vous deviez le recompiler avec l'option `configure` suivante : `--enable-blumenthal-aes`. Les anciennes versions de net-snmp ne prennent pas en charge AES192/AES256. Voir aussi : Strong Authentication or Encryption.

Vérification

Dans les deux exemples, vous verrez des lignes similaires dans votre `/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)

4 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 du processus de nettoyage

Toutes les données plus anciennes seront supprimées par le processus de nettoyage.

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 les données de tendances plus longtemps. 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 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.

5 Paramètres utilisateurs

Aperçu

Il peut arriver que vous souhaitiez exécuter une vérification d'agent qui n'est pas prédéfinie dans Zabbix. C'est là que les paramètres utilisateur deviennent utiles.

Vous pouvez écrire une commande qui récupère les données dont vous avez besoin et l'inclure dans le paramètre utilisateur du fichier de configuration de l'agent (paramètre de configuration « UserParameter »).

Un paramètre utilisateur a la syntaxe suivante :

```
UserParameter=<key>,<command>
```

Comme vous pouvez le voir, un paramètre utilisateur contient également une clé. La clé sera nécessaire lors de la configuration d'un élément. Saisissez une clé de votre choix, facile à référencer (elle doit être unique au sein d'un hôte).

Redémarrez l'agent ou utilisez l'option de contrôle d'exécution de l'agent pour prendre en compte le nouveau paramètre, par exemple :

```
zabbix_agentd -R userparameter_reload
```

Ensuite, lors de la configuration d'un élément, saisissez la clé pour référencer la commande du paramètre utilisateur que vous souhaitez exécuter.

Les paramètres utilisateur sont des commandes exécutées par l'agent Zabbix. Notez qu'il est possible de renvoyer jusqu'à 16 Mo de données avant les étapes de prétraitement des valeurs d'élément.

/bin/sh est utilisé comme interpréteur de ligne de commande sous les systèmes d'exploitation UNIX. Les paramètres utilisateur respectent le délai d'expiration des vérifications d'agent ; si ce délai est atteint, le processus de paramètre utilisateur créé est arrêté.

Voir aussi :

- Tutoriel étape par étape sur l'utilisation des paramètres utilisateur
- Exécution de commandes

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.

Paramètres utilisateur flexibles

Les paramètres utilisateur flexibles acceptent des paramètres avec la clé. De cette manière, un paramètre utilisateur flexible peut servir de base à la création de plusieurs éléments.

Les paramètres utilisateur flexibles ont la syntaxe suivante :

```
UserParameter=key[*],command
```

Paramètre	Description
Clé	Clé d'élément unique. Le [*] définit que cette clé accepte des paramètres entre crochets. Les paramètres sont fournis lors de la configuration de l'élément.
Commande	Commande à exécuter pour évaluer la valeur de la clé. *Pour les paramètres utilisateur flexibles uniquement* : Vous pouvez utiliser des références positionnelles $1…$9 dans la commande pour faire référence au paramètre correspondant dans la clé de l'élément. Zabbix analyse les paramètres placés entre [ ] dans la clé de l'élément et remplace $1,…,$9 dans la commande en conséquence. $0 sera remplacé par la commande d'origine (avant le développement de $0,…,$9) à exécuter. Les références positionnelles sont interprétées qu'elles soient ou non placées entre guillemets doubles (") ou simples ('). Pour utiliser les références positionnelles sans modification, indiquez un double signe dollar ; par exemple, awk '{print $$2}'. Dans ce cas, $$2 deviendra effectivement $2 lors de l'exécution de la commande.

Attention:
Les références positionnelles avec le signe $ sont recherchées et remplacées par l'agent Zabbix uniquement pour les paramètres utilisateur flexibles. Pour les paramètres utilisateur simples, ce traitement des références est ignoré et, par conséquent, aucun échappement du signe $ n'est nécessaire.

Attention:
Certains caractères ne sont pas autorisés par défaut dans les paramètres utilisateur. Pour la liste complète des caractères, voir UnsafeUserParameters.

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]
```

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.

1 Extension des agents Zabbix

Ce tutoriel fournit des instructions pas à pas sur l'extension des fonctionnalités de l'agent Zabbix à l'aide d'un paramètre utilisateur.

Étape 1

Ecrivez un script ou une 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 à 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 des *requêtes*.

Testez ce paramètre en utilisant l'agent Zabbix avec l'option "-t" (s'il est exécuté sous root, notez toutefois que l'agent peut avoir des autorisations différentes lorsqu'il est lancé en tant que démon) :

zabbix_agentd -t mysql.questions

Étape 3

Rechargez les paramètres utilisateur à partir du fichier de configuration en exécutant :

zabbix_agentd -R userparameter_reload

Vous pouvez également redémarrer l'agent au lieu de la commande de contrôle d'exécution.

Testez le paramètre en utilisant l'utilitaire zabbix_get.

Étape 4

Ajouter un nouvel élément avec Key=mysql.questions à l'hôte supervisé. Le type de l'élément doit être agent Zabbix ou agent Zabbix (actif).

Sachez que le type de valeurs renvoyées doit être correctement défini sur le serveur Zabbix. Sinon, Zabbix ne les acceptera pas.

6 Compteurs de performance Windows

Aperçu

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.

Afin d'obtenir une 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'objet de 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 d'objets et de compteurs de performance. Zabbix prend en charge ces représentations numériques comme paramètres des clés d'élément `perf_counter`, `perf_counter_en` et dans les 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 qui fonctionnent sur différents hôtes avec diverses versions localisées de Windows, vous pouvez utiliser la clé `perf_counter_en` ou le paramètre de configuration `PerfCounterEn`, qui permettent d'utiliser des noms anglais indépendamment de la langue du système.

Pour connaître les équivalents numériques, exécutez **regedit**, puis localisez *Counter* dans *HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windo NT\CurrentVersion\Perflib\009*.

L'entrée du registre contient des informations comme 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 pouvez y trouver les numéros correspondants pour chaque partie de chaîne du compteur de performance, comme dans '\System\% Processor Time' :

```
System → 2
% Processor Time → 6
```

Vous pouvez ensuite utiliser ces numéros pour représenter le chemin sous forme numérique :

```
\2\6
```

Paramètres du compteur de performance

Vous pouvez déployer certains paramètres PerfCounter pour la surveillance des 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
```

Avec de tels paramètres en place, vous pouvez alors simplement utiliser *UserPerfCounter1* ou *UserPerfCounter2* comme clés pour créer le éléments respectifs.

N'oubliez pas de redémarrer l'agent Zabbix après avoir apporté des modifications au fichier de configuration.

7 Mise à jour en masse

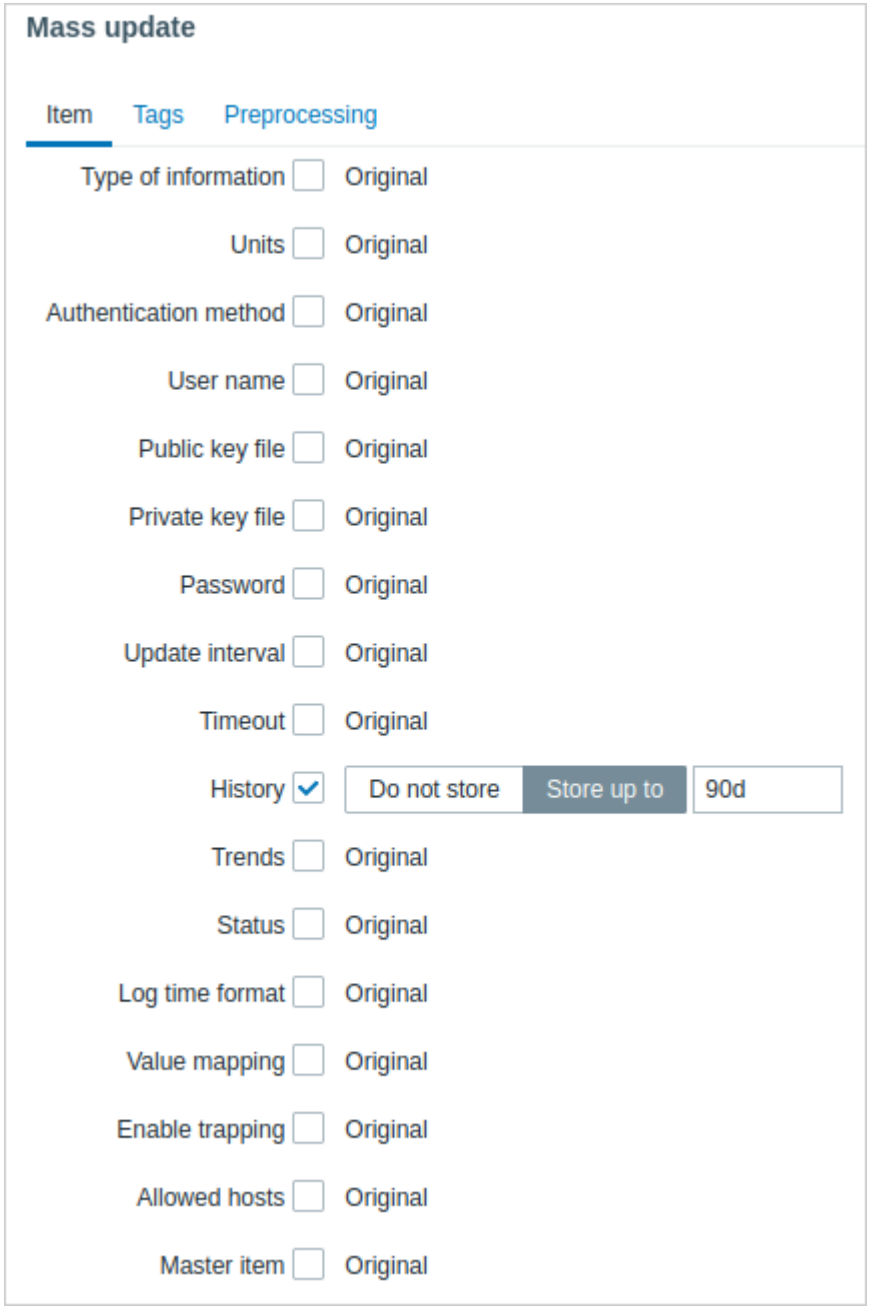
Aperçu

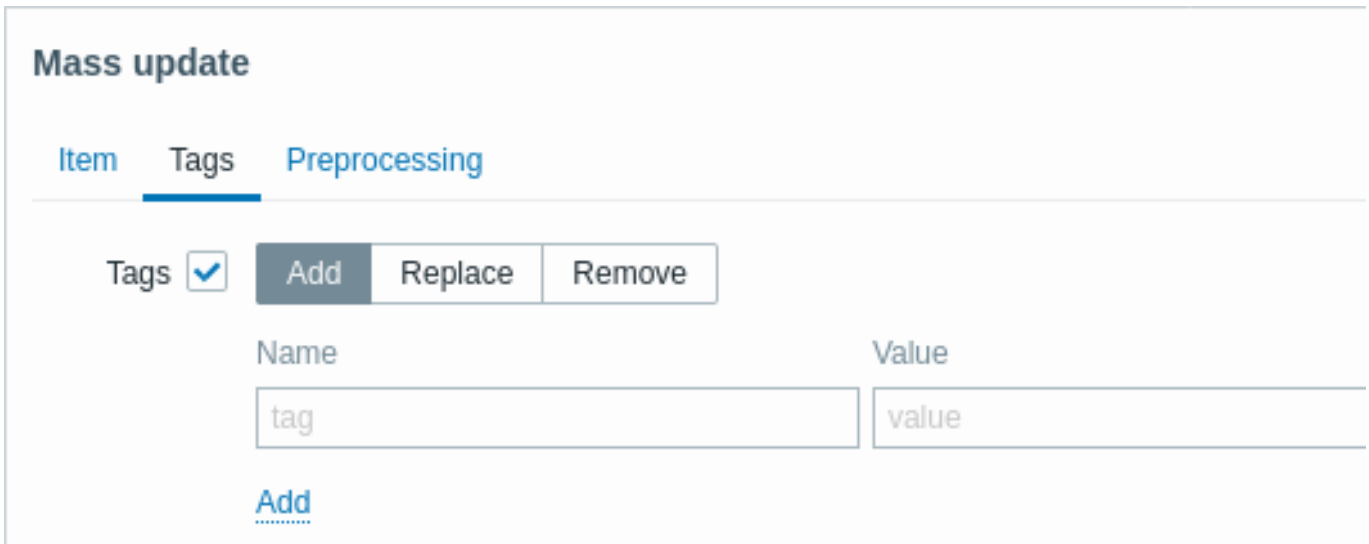
Parfois, vous voudrez peut-être modifier un attribut pour un certain nombre d'éléments à la fois. Au lieu d'ouvrir chaque élément individuel pour le modifier, vous pouvez utiliser la fonction de mise à jour en masse pour cela.

Utilisation de la mise à jour en masse

Pour mettre à jour en masse certains éléments, 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*, *Tags* ou *Prétraitement*)
- Cochez les cases des attributs à mettre à jour
- Saisissez de nouvelles valeurs pour les attributs



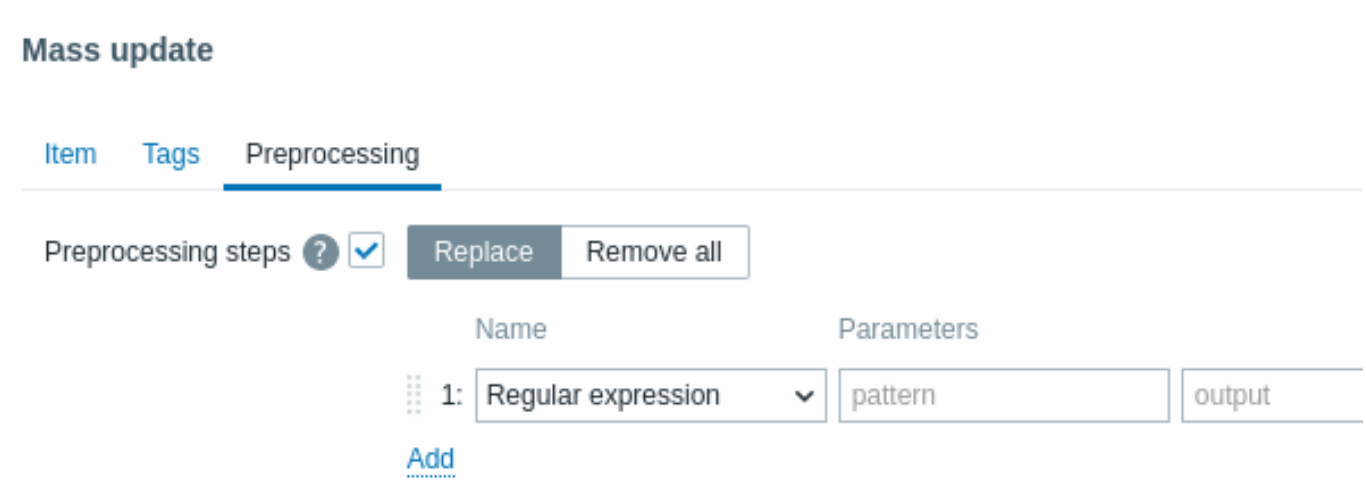


L'option *Tags* permet de :

- *Ajouter* - ajouter les tags spécifiés aux éléments (les tags portant le même nom, mais des valeurs différentes, ne sont pas considérés comme des « doublons » et peuvent être ajoutés au même élément).
- *Remplacer* - supprimer les tags spécifiés et ajouter des tags avec de nouvelles valeurs

- *Supprimer* - supprimer les tags spécifiés 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 tags.



Cochez la case 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* permet de :

- *Remplacer* - remplacer les étapes de prétraitement existantes des éléments par les étapes de prétraitement spécifiées ci-dessous
- *Tout supprimer* - supprimer toutes les étapes de prétraitement des éléments

Une fois terminé, cliquez sur *Mettre à jour*.

8 Correspondance des 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*.

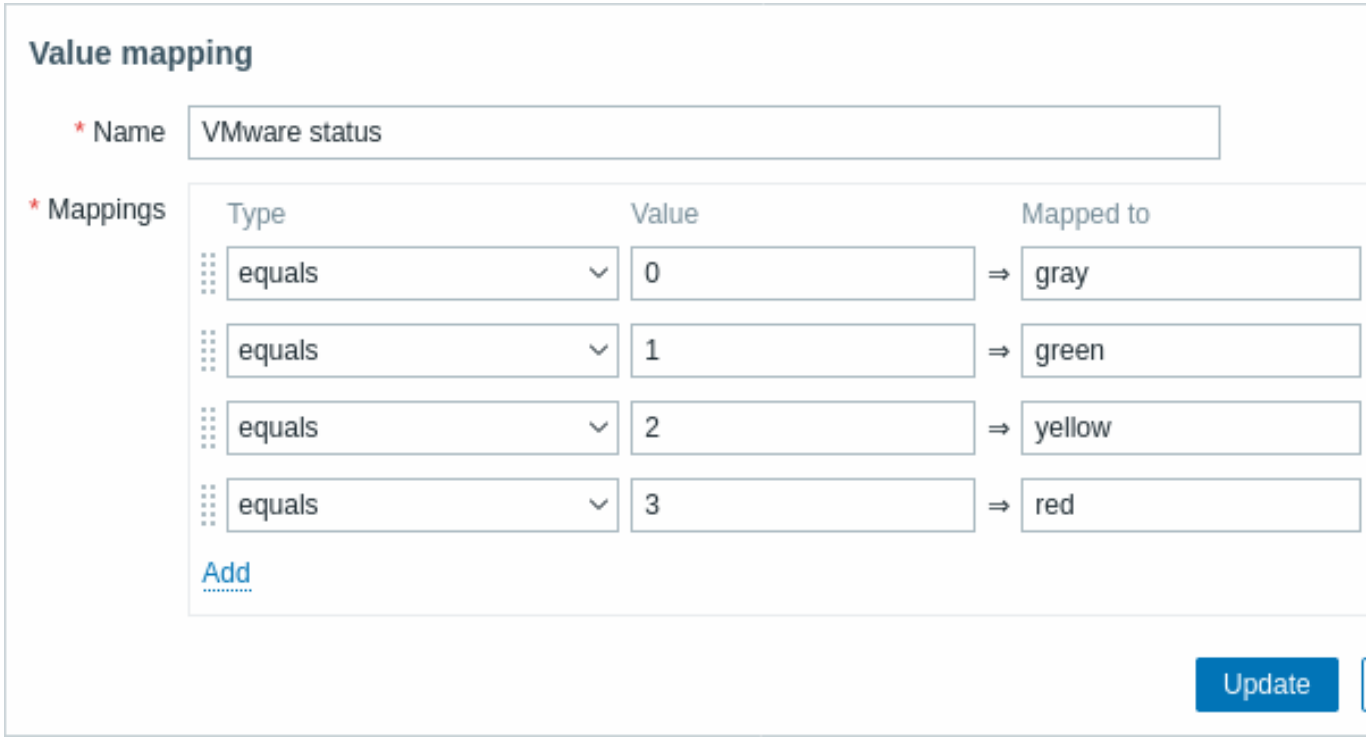
Les mappages de valeurs sont exportés/importés avec le modèle ou l'hôte respectif. Ils peuvent également être mis à jour en masse à l'aide des formulaires de mise à jour en masse hôte et modèle.

Configuration

Pour configurer un mappage de valeurs, suivez ces étapes :

1. Ouvrez le formulaire de configuration de l'hôte ou du modèle.

2. Dans l'onglet *Mappage de valeurs*, cliquez sur *Ajouter* pour ajouter un nouveau mappage de valeurs, ou cliquez sur le nom d'un mappage existant pour le modifier.



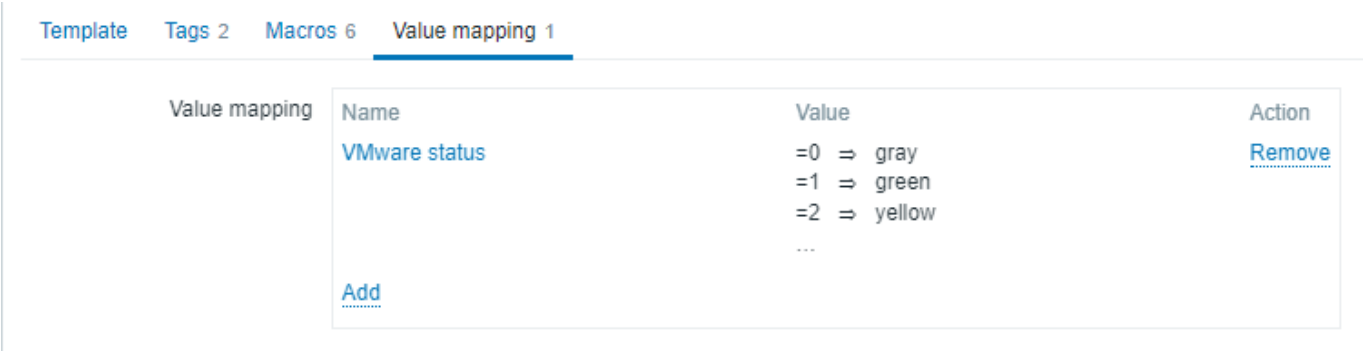
Paramètres d'un mappage de valeurs :

Paramètre	Description
Nom	Nom unique pour l'ensemble des mappages de valeurs.
Mappages	Règles individuelles pour mapper des valeurs numériques/chaînes vers des représentations sous forme de chaîne. Le mappage est appliqué dans l'ordre des règles, qui peuvent être réorganisées par glisser-déposer.

Paramètre	Description
Type	Type de mappage : **égal à** - les valeurs égales seront mappées ; **supérieur ou égal à** - les valeurs égales ou supérieures seront mappées ; **inférieur ou égal à** - les valeurs égales ou inférieures seront mappées ; **dans l'intervalle** - les valeurs comprises dans l'intervalle seront mappées ; l'intervalle s'exprime sous la forme <number1>-<number2> ou <number> ; plusieurs intervalles sont pris en charge (par exemple, 1-10,101-110,201) ; **regexp** - les valeurs correspondant à l'expression régulière seront mappées (les expressions régulières globales ne sont pas prises en charge) ; **par défaut** - toutes les autres valeurs seront mappées, à l'exception de celles disposant d'un mappage spécifique. Pour les intervalles de mappage, seuls les types de valeurs numériques (*supérieur ou égal à*, *inférieur ou égal à*, *dans l'intervalle*) sont pris en charge.
Valeur	Valeur entrante (peut contenir un intervalle ou une expression régulière, selon le type de mappage).
Mappé vers	Représentation sous forme de chaîne (jusqu'à 64 caractères) pour la valeur entrante.

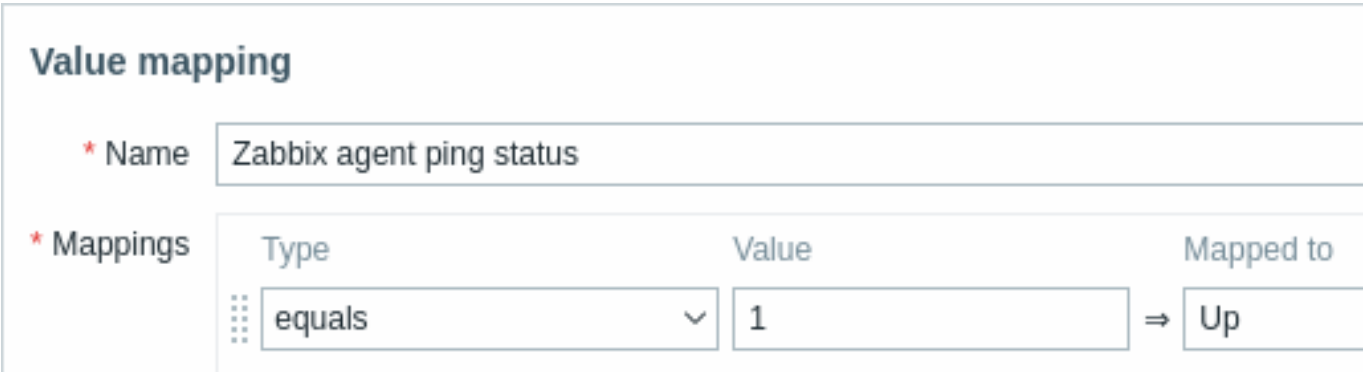
Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge.

Lors de l'affichage du mappage de valeurs dans la liste, seuls les trois premiers mappages sont visibles, trois points indiquant que d'autres mappages existent.



Exemple de correspondance de valeurs

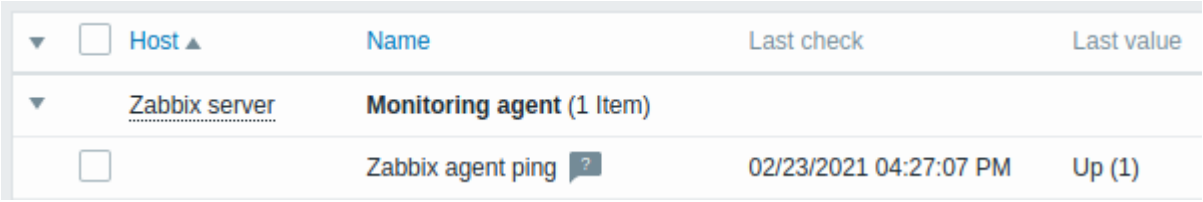
L'un des éléments d'agent prédéfinis, *Zabbix agent ping*, utilise une correspondance de valeurs au niveau du modèle, « Zabbix agent ping status », pour afficher ses valeurs.



Dans le formulaire de configuration de l'élément, vous pouvez trouver une référence à cette correspondance de valeurs dans le champ *Value mapping* :

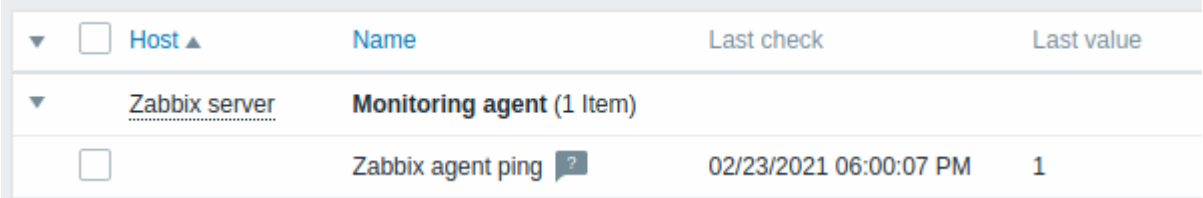


Cette correspondance est utilisée dans la section *Monitoring* → *Latest data* pour afficher « Up » (avec la valeur brute entre parenthèses).



Note:
Dans la section *Latest data*, les valeurs affichées sont raccourcies à 20 caractères. Si une correspondance de valeurs est utilisée, ce raccourcissement 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.



9 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

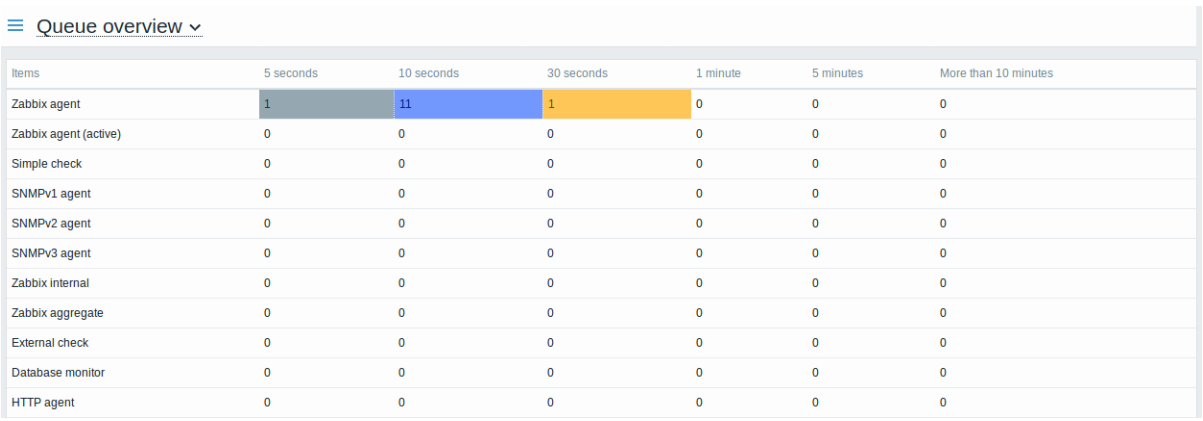
Les statistiques affichées par la file d'attente constituent un bon indicateur des performances du serveur Zabbix.

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

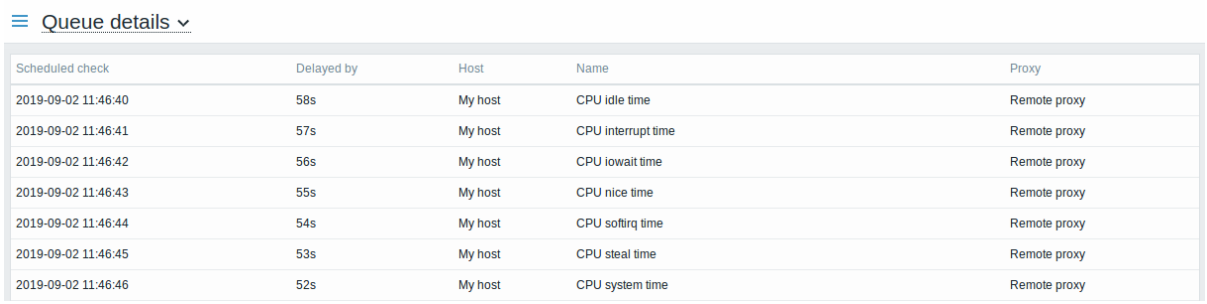
Pour lire la file d'attente, accédez à *Administration* > *Queue* > *Queue overview*.



L'image ici est généralement « ok », nous pouvons donc supposer que le serveur fonctionne correctement.

La file d'attente montre certains éléments en attente depuis jusqu'à 30 secondes. Il serait utile de savoir de quels éléments il s'agit.

Pour cela, sélectionnez *Queue details* dans la liste déroulante du titre. Vous pouvez maintenant voir une liste de ces éléments retardés.



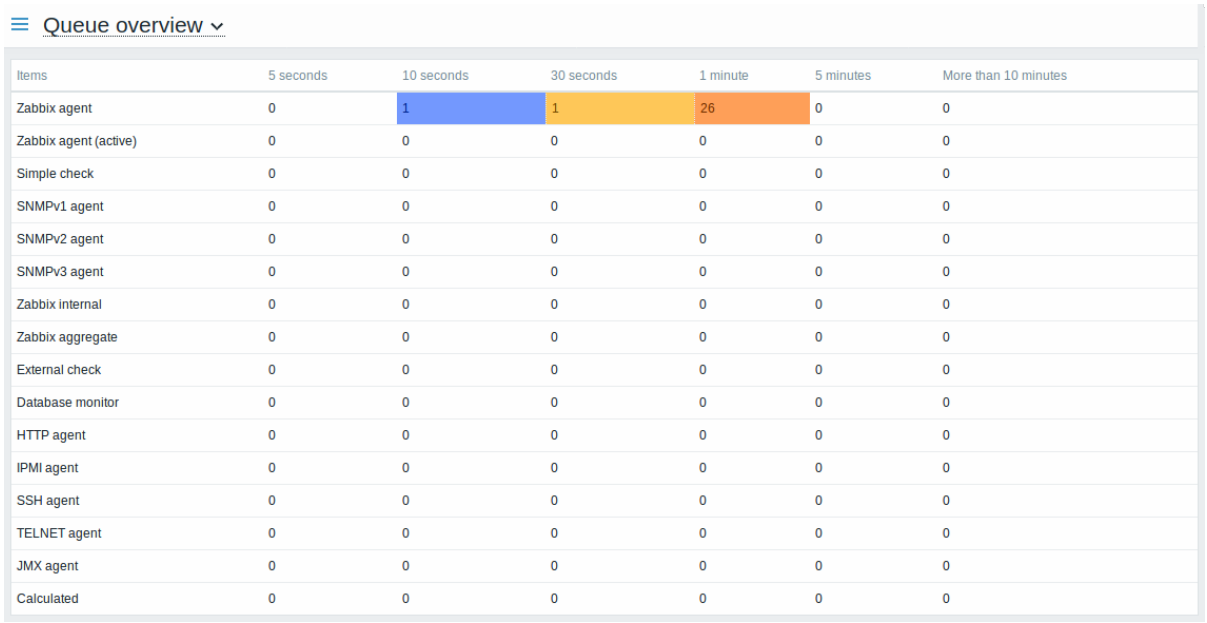
Queue details

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 détails, il peut être possible de déterminer pourquoi ces éléments peuvent être retardés.

Avec un ou deux éléments retardés, il n'y a peut-être pas lieu de s'inquiéter. Ils pourraient être mis à jour dans une seconde. Cependant, si vous voyez un grand nombre d'éléments retardés pendant trop longtemps, il pourrait y avoir un problème plus sérieux.

Voir aussi : 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 spécifique à **zabbix[queue,<from>,<to>]** peut être utilisé pour superviser l'intégrité de la file d'attente dans Zabbix. Il renverra le nombre d'éléments retardés par rapport à la durée définie. Pour plus d'informations, voir Éléments internes.

10 Cache des 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, un paramètre facultatif **ValueCacheSize** est pris en charge dans le fichier de 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>]**. Voir plus de détails dans éléments internes.

11 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

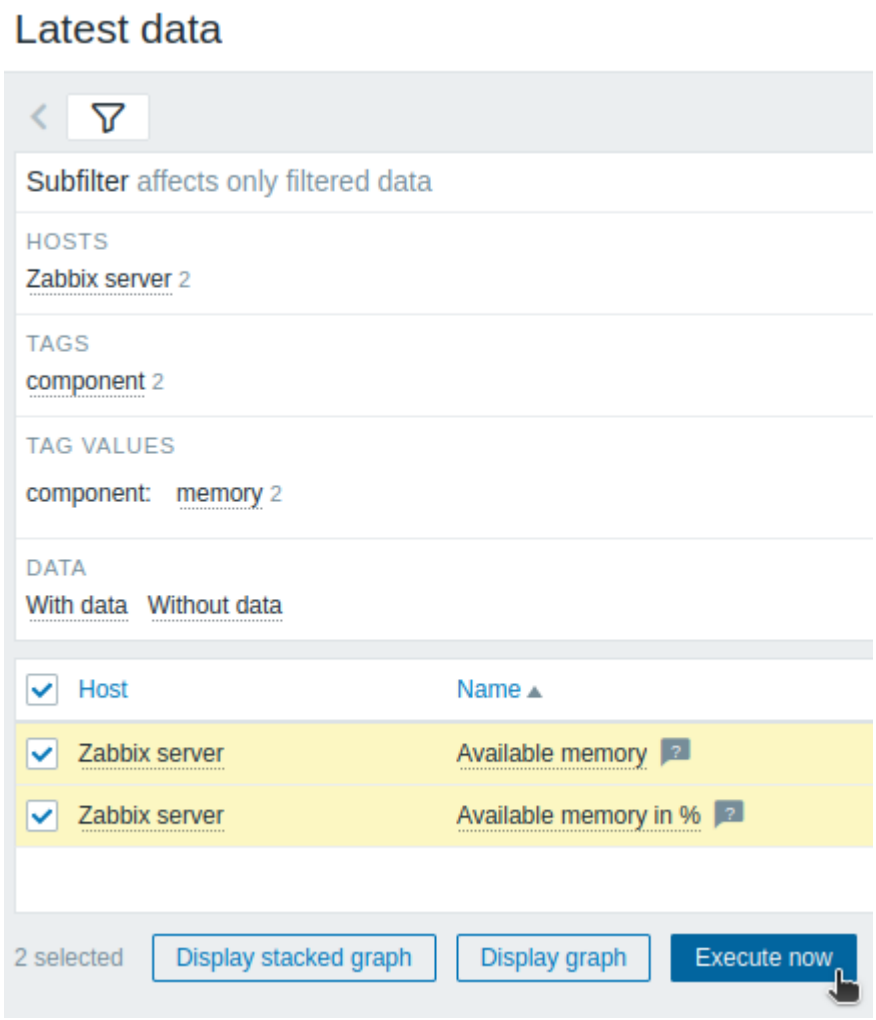
Vous pouvez exécuter des contrôles passifs depuis :

- *Surveillance* > *Dernières données*
- *Collecte de données* > *Hôtes* > *Éléments* ou *Découverte*
- Le menu des éléments
- Lors de la modification des éléments ou des règles de découverte de bas niveau (dans leurs formulaires de configuration)

Attention:
Le contrôle doit exister dans le cache de configuration pour pouvoir être exécuté ; voir aussi CacheUpdateFrequency. Le cache n'est **pas mis à jour** avant l'exécution du contrôle ; ainsi, 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 un contrôle passif dans *Surveillance* > *Dernières données* :

1. Sélectionnez les éléments ; vous pouvez exécuter le contrôle pour plusieurs éléments à la fois.
2. Cliquez sur *Exécuter maintenant*.



Note:
Dans *Dernières données*, les utilisateurs peuvent *Exécuter maintenant* des éléments sur des hôtes pour lesquels ils disposent d'une autorisation *lecture-écriture*, ou d'une autorisation *lecture* avec l'action *Invoquer "Exécuter maintenant" sur des hôtes en lecture seule* activée pour leur rôle.

12 Restriction des contrôles de l'agent

Aperçu

Vous pouvez contrôler quelles clés d'élément l'agent Zabbix ou l'agent 2 est autorisé à utiliser ou à refuser 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/refus :

- `AllowKey=<pattern>`
- `DenyKey=<pattern>`

Le `<pattern>` doit contenir une seule clé d'élément et 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.*[*]`).

Attention:
Pour améliorer la sécurité, il est recommandé d'utiliser des clés d'élément exactes au lieu de caractères génériques. Pour plus de détails, voir Sécurisation des règles d'autorisation/refus.

Contrairement aux autres paramètres de configuration de l'agent, vous pouvez spécifier un nombre illimité de paramètres `AllowKey` ou `DenyKey`.

Remarques importantes

- Tous les éléments `system.run` sont désactivés par défaut (même lorsque DenyKey est vide), comme si `DenyKey=system.run[*]` était défini comme la dernière règle. De ce fait, vous pouvez autoriser des éléments `system.run` spécifiques sans refuser explicitement les autres éléments `system.run`.
- Un élément spécifié dans `AllowKey` doit également être spécifié dans DenyKey (à l'exception des éléments `system.run`) ; sinon, l'agent Zabbix **ne démarrera pas**.
- Si possible, utilisez `AllowKey` pour n'autoriser que 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/pass` et les nouvelles versions de l'agent Zabbix peuvent introduire des clés non couvertes par vos règles DenyKey.
- La configuration de `AllowKey` et DenyKey n'affecte pas les paramètres d'agent `HostnameItem`, `HostMetadataItem` ou `HostInterfaceItem`.
- Les éléments refusés deviennent non pris en charge sans aucun indice ni message d'erreur ; par exemple :
  - Le paramètre de ligne de commande `--print (-p)` de l'agent Zabbix n'affichera pas les clés d'élément refusées.
  - Le paramètre de ligne de commande `--test (-t)` de l'agent Zabbix renverra « Unsupported item key. » pour les clés d'élément refusées.
  - Le fichier journal de l'agent Zabbix n'enregistrera pas les commandes distantes refusées si `LogRemoteCommands=1`.

Ordre des règles d'autorisation/refus

Vous pouvez spécifier un nombre illimité de règles `AllowKey` ou DenyKey, 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 motif de clé d'élément ne correspond pas, l'agent passe à
AllowKey=vfs.file.contents[/etc/passwd]     # Le motif de clé d'élément correspond ; l'agent autorise la vé
DenyKey=vfs.file.*[*]                       # L'agent ignore la règle, car l'évaluation est déjà arrêtée.
```

L'ordre de règles suivant refusera la vérification de l'élément :

```
DenyKey=vfs.file.*[*]                       # Le motif de clé d'élément correspond ; l'agent refuse la véri
AllowKey=vfs.file.contents[/etc/passwd]     # L'agent ignore la règle, car l'évaluation est déjà arrêtée.
AllowKey=vfs.file.contents[/tmp/app.log]    # L'agent ignore la règle, car l'évaluation est déjà arrêtée.
```

Exemples

Les exemples suivants montrent des modèles de configuration courants pour `AllowKey` et DenyKey.

Autoriser des vérifications et des commandes spécifiques

Autorisez uniquement deux vérifications d'élément `vfs.file` et deux commandes `system.run` :

```
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.*[*]
```

Note:
Le paramètre `DenyKey=system.run[*]` n'est pas nécessaire, 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 :

- Les scripts globaux pouvant être exécutés dans l'interface web ou via l'API (cette méthode utilise toujours la clé `system.run[myscript.sh]`)
- Les commandes distantes depuis les opérations d'action (cette méthode utilise toujours la clé `system.run[myscript.sh,nowait]`)
- Les éléments d'agent Zabbix `system.run` avec le script, par exemple :
  - `system.run[myscript.sh]`
  - `system.run[myscript.sh,wait]`
  - `system.run[myscript.sh,nowait]`

```
AllowKey=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]
```

Sécurisation des règles d'autorisation/refus

Cet exemple montre comment sécuriser des règles AllowKey ou DenyKey trop permissives.

Considérez les règles suivantes :

```
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat*"]
DenyKey=vfs.file.*
DenyKey=system.cpu.load[*]
```

Note:
Sous Windows, vous devez échapper les espaces dans le chemin à l'aide d'un accent circonflexe (^).

Ces règles contiennent un caractère générique (*), qui peut être utilisé de manière abusive :

- Le script test.bat peut être exécuté avec n'importe quels arguments, y compris des arguments non prévus.
- Le motif vfs.file.* ne correspond qu'aux clés d'élément sans paramètres ; cependant, tous les éléments vfs.file nécessitent des paramètres.
- Le motif system.cpu.load[*] ne correspond qu'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 corrects de clés d'élément ; par exemple :

```
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[*]
```

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

Le tableau suivant montre comment les motifs de clés d'élément sont mis en correspondance :

- Une clé correspond au motif uniquement si elle remplit **toutes** les conditions de la colonne *Correspondances*.
- Les paramètres doivent être entièrement entourés de crochets (par exemple, vfs.file.contents[* et vfs.file.contents*utf8] sont des motifs non valides).

Motif	Correspondances	Exemples
*	Toute clé avec ou sans paramètres	
vfs.file.*	La clé commence par vfs.file. Aucun paramètre	Correspond à : vfs.file.size vfs.file.contents Ne correspond pas à : 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[]

Motif	Correspondances	Exemples

`vfs.file.*[*]`	La clé commence par `vfs.file.`
	N'importe quels paramètres ou paramètres 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`

`vfs.file.contents[*]`	La clé est `vfs.file.contents`
	N'importe quels paramètres ou paramètres 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`
	N'importe quel deuxième paramètre ou deuxième paramètre 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`
	Le premier paramètre inclut `passwd`
	Pas de deuxième paramètre
		Correspond à :
		`vfs.file.contents[/etc/passwd]`
		
		Ne correspond pas à :
		`vfs.file.contents[/etc/passwd,]`
		`vfs.file.contents[/etc/passwd,utf8]`

`vfs.file.contents[*passwd*,*]`	La clé est `vfs.file.contents`
	Le premier paramètre inclut `passwd`
	N'importe quel deuxième paramètre ou deuxième paramètre 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]`

3 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 survient.

Le flux de travail complet de détection des problèmes est le suivant :

1. Assurez-vous d'avoir 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 Zabbix officiels contiennent des déclencheurs prédéfinis. Consultez le guide sur l'application d'un modèle pour surveiller 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.

1 Exemples de déclencheurs

Cette page est une collection d'exemples de déclencheurs.

Les exemples sont classés par cas d'utilisation :

- Hôte indisponible

- Proxy indisponible
- Cluster SMTP indisponible
- Hôte inaccessible
- Redémarrage inattendu
- Modifications dans des fichiers importants
- Modifications dans une requête DNS
- Logiciels non concordants sur différents hôtes
- Horloges non synchronisées
- 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 la requête DNS

```
last(/Zabbix server/net.dns.record[192.0.2.0,{$WEBSITE_NAME},{$DNS_RESOURCE_RECORD_TYPE},2,1])<>"{$WEBSITE
```

Notez les guillemets autour du second opérande.

Ce déclencheur se déclenche si le résultat de la requête n'est pas égal à ce qu'il renvoie normalement :

```
example.com             MX       0 mail.example.com
```

Fonction : last
 Élément de données : net.dns.record[192.0.2.0,{$WEBSITE_NAME},{$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 des horloges non synchronisées

```
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 l'agent Zabbix 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`

2 Configuration d'un déclencheur

Cette page décrit comment configurer un déclencheur dans l'interface web de Zabbix.

Notez que les modèles Zabbix officiels contiennent des déclencheurs prédéfinis. Consultez le guide pour surveiller Linux en appliquant un modèle. Il est également possible de mettre à jour en masse les déclencheurs configurés.

Autres ressources :

- Détection des problèmes avec les déclencheurs (introduction générale)
- Exemples de déclencheurs (par cas d'utilisation)
- Expression de déclencheur (détails de la syntaxe)

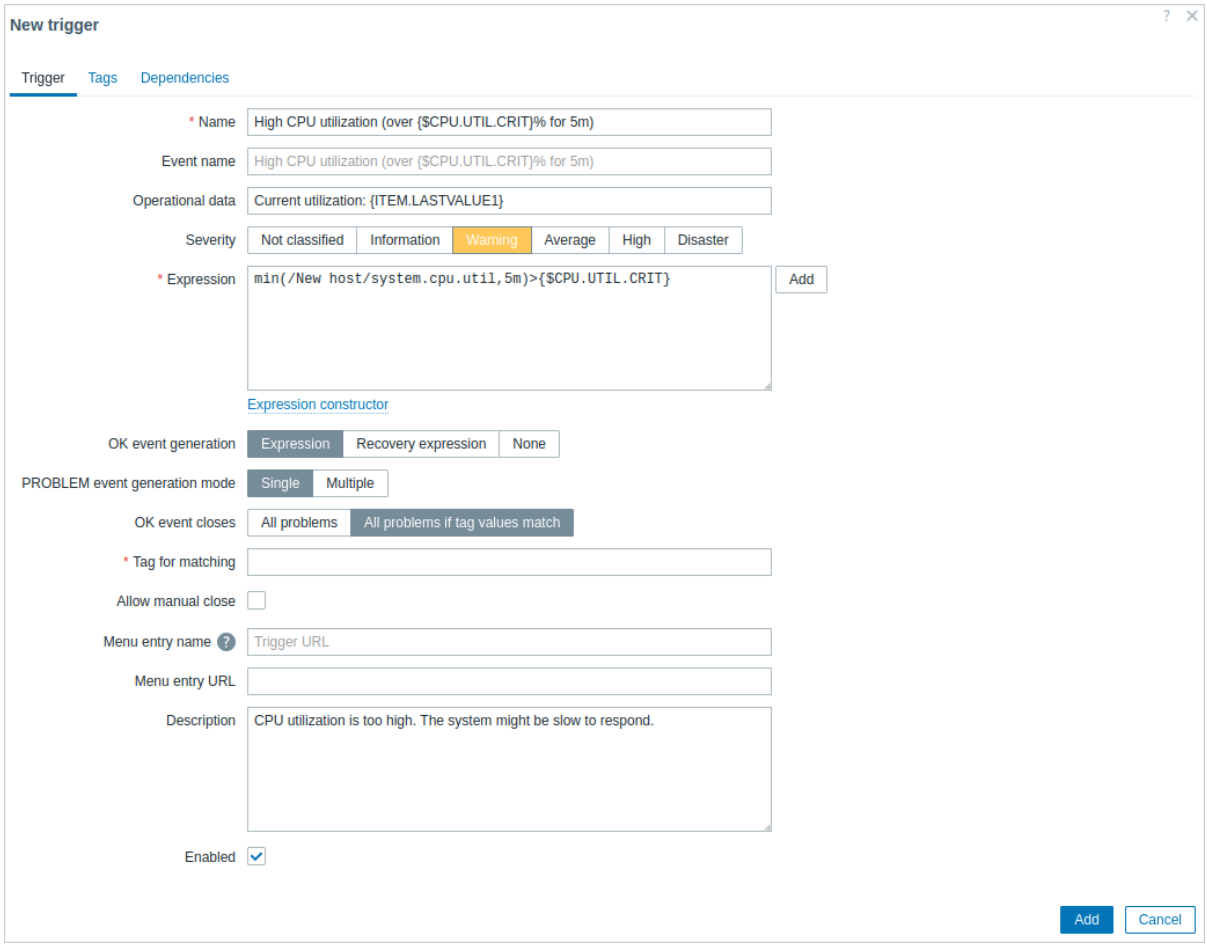
- Bonnes pratiques

Configuration

Pour configurer un déclencheur, procédez comme suit :

- Allez dans : *Collecte de données* > *Hôtes*
- Cliquez sur *Déclencheurs* sur la ligne de l'hôte
- Cliquez sur *Créer un déclencheur* à droite (ou sur le nom du déclencheur pour modifier un déclencheur existant)
- Saisissez les paramètres du déclencheur dans le formulaire

L'onglet **Déclencheur** contient tous les attributs essentiels du déclencheur.



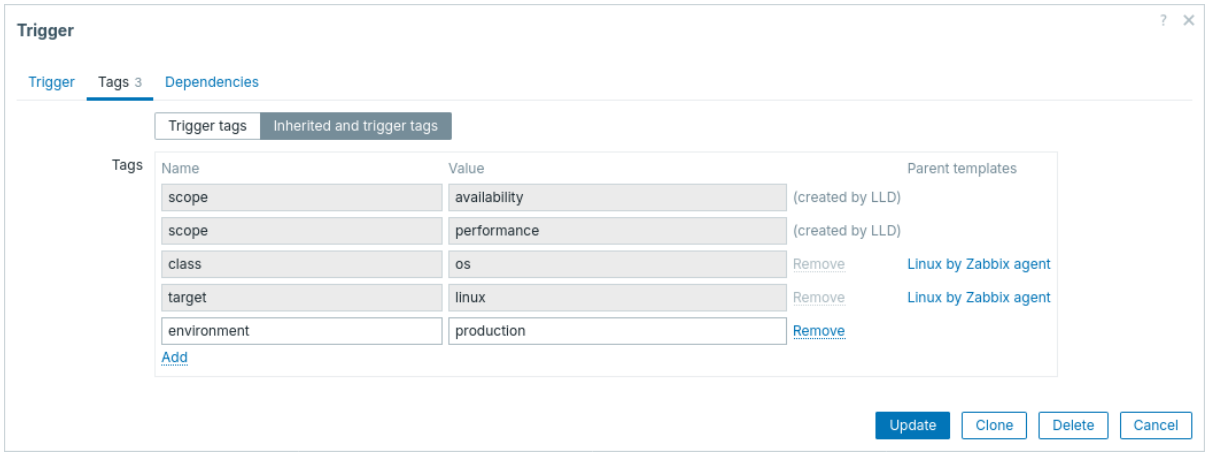
Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge.

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.*}, ainsi que 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 correctement résolues si elles font référence à des constantes dans des expressions relativement simples et directes. Par exemple, le nom « Charge du processeur supérieure à $1 sur {HOST.NAME} » deviendra automatiquement « Charge du processeur supérieure à 5 sur 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 créer des alertes explicites contenant des données de problème (voir exemple). Le même ensemble de macros que pour le nom du déclencheur est pris en charge, avec en plus {TIME}, {TIMESTAMP} et les macros d'expression {?EXPRESSION}.

Parameter	Description
Operational data	Les données opérationnelles permettent de définir des chaînes arbitraires avec des macros. Les macros seront résolues dynamiquement en données en temps réel dans *Surveillance* > *Problèmes*. Alors que les macros dans le nom du déclencheur (voir ci-dessus) seront résolues à leurs valeurs au moment où un problème survient et constitueront la base d'un nom de problème statique, les macros dans les données opérationnelles conservent la capacité d'afficher dynamiquement les informations les plus récentes. Si aucune donnée opérationnelle n'est configurée au niveau du déclencheur, les dernières valeurs de tous les éléments de l'expression seront affichées. Le même ensemble de macros que pour le nom du déclencheur est pris en charge.
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 mémoire sont pris en charge. Un problème est créé après 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 récupération supplémentaires sont spécifiées dans *Expression de récupération*.
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 du problème est évaluée à FALSE et l'expression de récupération 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 soit résolu, après que l'expression de problème d'origine a déjà été évaluée à FALSE. L'expression de récupération est utile pour l'hystérésis des déclencheurs. Il n'est **pas** possible de résoudre un problème à l'aide de la seule expression de récupération si l'expression de problème est toujours TRUE. Ce champ n'est disponible que 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 à l'état « Problem » pour la première fois ; **Multiple** - un événement est généré à *chaque* évaluation « Problem » du déclencheur.
OK event closes	Sélectionnez ce que l'événement OK ferme : **All problems** - tous les problèmes de ce déclencheur ; **All problems if tag values match** - uniquement les problèmes de ce déclencheur dont les valeurs des tags d'événement correspondent.
Tag for matching	Saisissez le nom du tag d'événement à utiliser pour la corrélation d'événements. Ce champ s'affiche si « All problems if tag values match » est sélectionné pour la propriété *OK event closes* et il est obligatoire dans ce cas.
Allow manual close	Cochez cette option 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	S'il n'est pas vide, le nom saisi ici (jusqu'à 64 caractères) est utilisé à plusieurs endroits 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 que pour le nom du déclencheur est pris en charge, avec en plus {EVENT.ID}, {HOST.ID} et {TRIGGER.ID}.
Menu entry URL	S'il n'est pas vide, l'URL saisie ici (jusqu'à 2048 caractères) est disponible sous forme de lien dans le menu d'événement à plusieurs endroits de l'interface, par exemple en cliquant sur le nom du problème dans *Surveillance* > *Problèmes* ou dans le widget de tableau de bord *Problèmes*. Le même ensemble de macros que pour le nom du déclencheur est pris en charge, avec en plus {EVENT.ID}, {HOST.ID} et {TRIGGER.ID}. Remarque : les macros utilisateur avec des valeurs secrètes ne seront pas résolues dans l'URL.
Description	Champ de texte utilisé pour fournir plus d'informations sur ce déclencheur. Peut contenir des instructions pour corriger un problème spécifique, les coordonnées du personnel responsable, etc. Le même ensemble de macros que pour le nom du déclencheur est pris en charge.

Parameter	Description
Enabled	Décocher cette case désactivera 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 tags au niveau du déclencheur. Tous les problèmes de ce déclencheur seront marqués avec les valeurs saisies ici.



En outre, l'option *Inherited and trigger tags* vous permet d'afficher les tags définis au niveau du modèle si le déclencheur provient de ce modèle. S'il existe plusieurs modèles avec le même tag, ces tags sont affichés une seule fois et les noms des modèles sont séparés par des virgules. Un déclencheur n'« hérite » pas des tags au niveau de l'hôte et ne les affiche pas.

Parameter	Description
Name/Value	Définissez des tags personnalisés pour marquer les événements du déclencheur. Les tags sont une paire composée d'un nom de tag et d'une valeur. Vous pouvez utiliser uniquement le nom ou l'associer à une valeur. Un déclencheur peut avoir plusieurs tags 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 macro avec `{{ITEM.VALUE}}`, `{{ITEM.LASTVALUE}}` sont prises en charge dans les tags 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 tags du déclencheur. Elle peut être utile pour identifier les déclencheurs créés à partir de prototypes de déclencheur et, par exemple, 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 tags d'événement. Les tags d'événement peuvent être utilisés pour la corrélation d'événements, dans les conditions d'action, et seront également visibles dans *Surveillance* > *Problèmes* ou dans le widget *Problèmes*.

L'onglet **Dépendances** contient toutes les dépendances du déclencheur.

Cliquez sur *Ajouter* pour ajouter une nouvelle dépendance.

> **Note:**
> Vous pouvez également configurer un déclencheur en ouvrant un déclencheur existant, en appuyant sur le bouton *Cloner*, puis en l'enregistrant sous un autre nom.

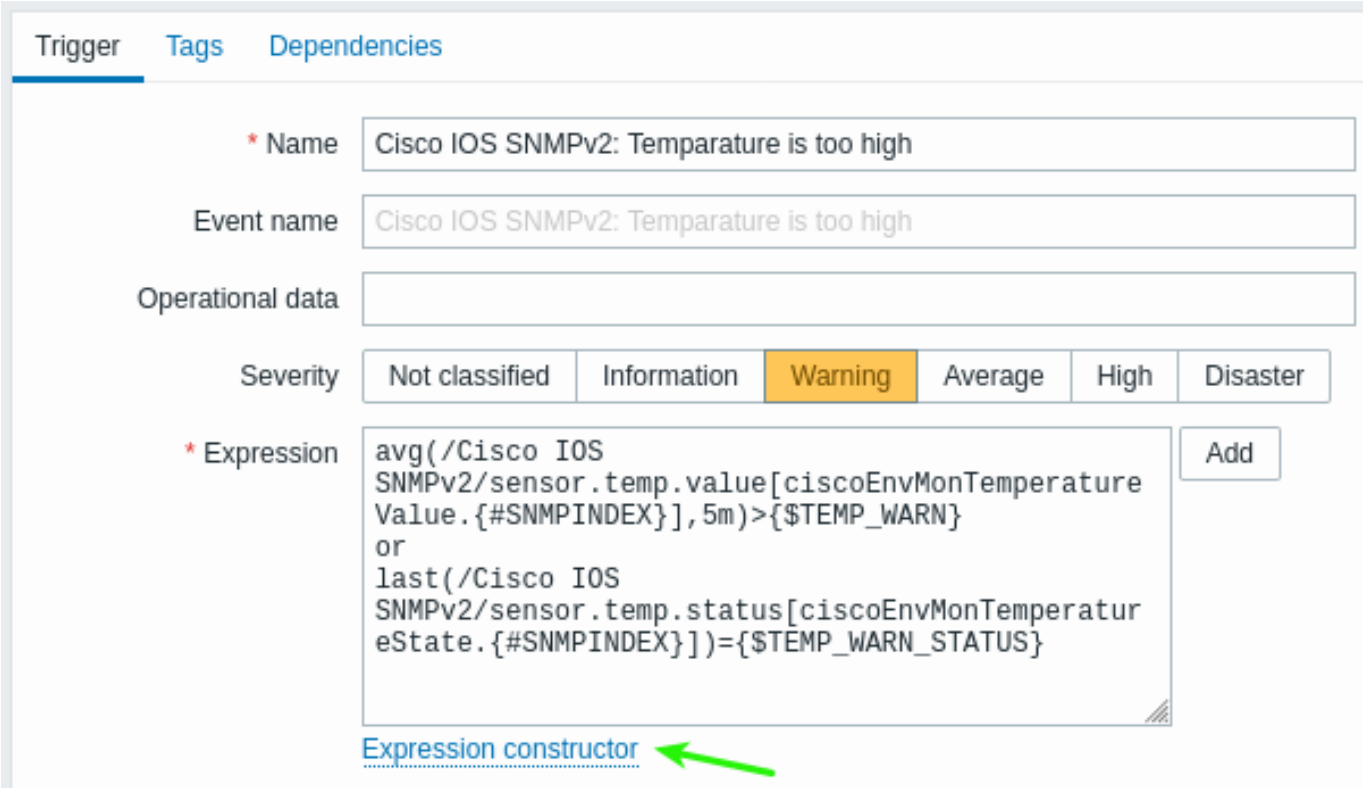
Test d'expression de déclencheur

Il est possible de tester quel serait le résultat de l'expression en fonction de la valeur reçue.

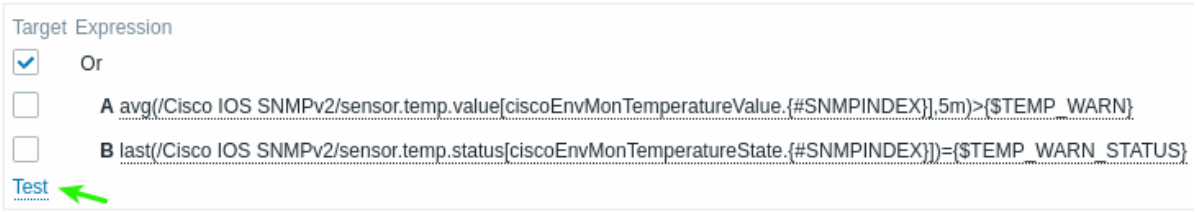
L'expression suivante provenant d'un modèle officiel est prise comme exemple :

```
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}
```

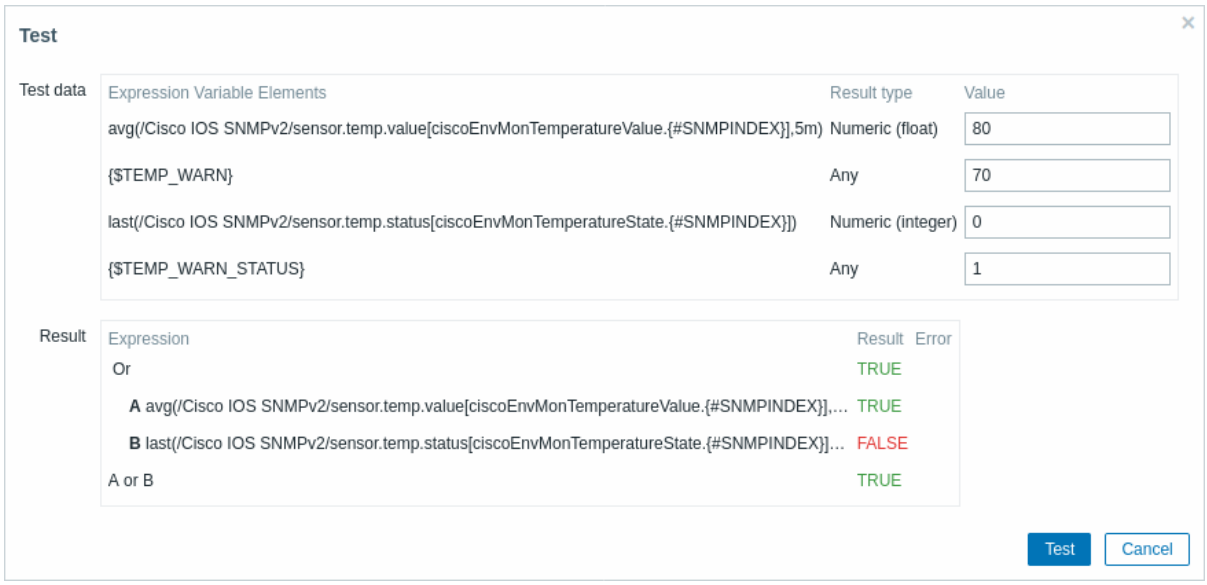
Pour tester l'expression, cliquez sur *Constructeur d'expression* sous le champ d'expression.



Dans le constructeur d'expression, toutes les expressions individuelles sont répertoriées. 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 affiché.

« TRUE » signifie que l'expression spécifiée est correcte. Dans ce cas particulier A, « 80 » est supérieur à la valeur spécifiée {$TEMP_WARN}, soit « 70 » dans cet exemple. Comme prévu, un résultat « TRUE » apparaît.

« FALSE » signifie que l'expression spécifiée est incorrecte. Dans ce cas particulier B, {$TEMP_WARN_STATUS} « 1 » doit être égal à la valeur spécifiée, soit « 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.

3 Expression de déclencheur

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 de calculer la moyenne ou de trouver une chaîne spécifique.

Cliquez sur le groupe de fonctions correspondant pour voir 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, last, lastclock, logeventid, logseverity, logsource, logtimestamp, monodec, monoinc, nodata, percentile, rate
Fonctions de tendances	baselinedev, baselinewma, trendavg, trendcount, trendmax, trendmin, trendstl, trendsum

Groupe de fonctions	Fonctions
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, insert, jsonpath, left, length, ltrim, mid, repeat, replace, right, rtrim, 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 agrégés.

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 des logiciels non concordants sur différents hôtes).

Paramètres de fonction

Les paramètres de fonction permettent de spécifier :

- la clé d'élément (sous la forme `/host/key`) pour les fonctions faisant référence à l'historique d'un élément d'hôte
- la période de temps (et 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 de temps

Les paramètres spécifiques à une fonction sont placés après la clé d'élément et sont séparés de la clé d'élément par une virgule.

La plupart des fonctions numériques acceptent une **période de temps** comme paramètre. Elle permet de spécifier l'intervalle qui nous intéresse. Elle peut être indiquée comme une période de temps (30s, 10m, 1h) ou comme une plage de valeurs (#5 - pour les cinq dernières valeurs).

Vous pouvez utiliser des secondes ou des suffixes de temps pour indiquer la période de temps. Précédé d'un 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 dièse ont une signification différente avec la fonction **last** : ils désignent la Nième valeur précédente ; ainsi, étant donné les valeurs 30, 70, 20, 60, 50 (de la plus récente à la plus ancienne) :

- `last(/host/key,#2)` renverrait '70'
- `last(/host/key,#5)` renverrait '50'

La période de temps est mesurée jusqu'à « maintenant » — où « maintenant » correspond à la dernière heure de recalcul du déclencheur (voir Fréquence de calcul) ; « maintenant » n'est pas l'heure « actuelle » du serveur.

La période de temps spécifie soit :

- de prendre en compte toutes les valeurs entre « maintenant - période de temps » et « maintenant » (ou, avec décalage temporel, entre « maintenant - décalage temporel - période de temps » et « maintenant - time_shift »)
- de ne pas prendre en compte plus que le nombre num de valeurs passées, jusqu'à « maintenant »
  - S'il y a 0 valeur disponible pour la période de temps ou le nombre num spécifié, alors le déclencheur ou l'élément calculé qui utilise 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 de temps sera considérée jusqu'à cette 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 and time et nodata(). Ainsi, le déclencheur sera calculé même si la période de temps 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 plage temporelle a été définie, par exemple, sur les dix dernières valeurs.

Décalage temporel

Un décalage temporel facultatif est pris en charge avec le temps ou le nombre de valeurs comme paramètre de fonction. Ce paramètre permet de référencer des données provenant d'une période passée.

Le décalage temporel commence par `now` — qui indique l'heure actuelle — et est 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)` renverra la valeur moyenne sur une heure, un jour auparavant.

Attention:
Le décalage temporel spécifié en mois (M) et en années (y) n'est pris en charge que pour les fonctions de tendances. 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 une journée, de lundi à 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 — et est 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 une journée, `+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 1.

Paramètre	Description
1d:now/d	Hier
1d:now/d+1d	Aujourd'hui
2d:now/d+1d	Les 2 derniers jours
1w:now/w	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
	>=	Supérieur ou égal 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]
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, entraîne une erreur

Dans ce cas, l'expression de déclencheur est généralement évaluée à `Unknown` (car elle ne peut pas être évaluée)

Il est possible de recevoir une notification concernant des déclencheurs inconnus.

Exceptions

Malgré la présence d'un opérande inconnu, les expressions de déclencheur peuvent, dans certains cas, être évaluées à un résultat connu (Problem/OK) :

- La fonction `nodata()` est évaluée независимо 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 entraîne une erreur, la valeur de la fonction devient `Unknown` et elle participe comme opérande inconnu à l'évaluation ultérieure de l'expression.

Les opérandes inconnus ne peuvent "disparaître" que dans les expressions logiques comme décrit ci-dessus. Dans les expressions arithmétiques, les opérandes inconnus conduisent toujours à `Unknown` (sauf en cas de 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 à l'état de problème même si la partie connue est résolue ('1' devient '0'), car l'expression est maintenant 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 le frontend 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.

1 Fonctions d'agrégation

Sauf indication contraire, toutes les fonctions répertoriées ici sont prises en charge dans :

- Expressions de déclencheur
- Éléments calculés
- Macros d'expression

Les fonctions d'agrégation peuvent fonctionner avec :

- l'historique des éléments, par exemple `min(/host/key,1h)`
- les fonctions foreach comme seul paramètre, par exemple `min(last_foreach(/*/key))` (uniquement dans les éléments calculés ; ne peut pas être utilisé dans les déclencheurs)

Les fonctions sont répertoriées sans informations supplémentaires. Cliquez sur la fonction pour voir tous les détails.

Function	Description
avg	La valeur moyenne d'un élément pendant la période d'évaluation définie.
bucket_percentile	Calcule le percentile à partir des compartiments d'un histogramme.
count	Le nombre de valeurs dans un tableau renvoyé par une fonction foreach.
histogram_quantile	Calcule le φ-quantile à partir des compartiments 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 des valeurs collectées pendant la période d'évaluation définie.
mad	L'écart absolu médian des valeurs collectées pendant la période d'évaluation définie.
max	La valeur la plus élevée d'un élément pendant la période d'évaluation définie.
min	La valeur la plus basse d'un élément pendant la période d'évaluation définie.

Function	Description
skewness	L'asymétrie de la distribution de probabilité des valeurs collectées pendant la période d'évaluation définie.
stddevpop	L'écart type de population des valeurs collectées pendant la période d'évaluation définie.
stddevsamp	L'écart type d'échantillon des valeurs collectées pendant la période d'évaluation définie.
sum	La somme des valeurs collectées pendant la période d'évaluation définie.
sumofsquares	La somme des carrés des valeurs collectées pendant la période d'évaluation définie.
varpop	La variance de population des valeurs collectées pendant la période d'évaluation définie.
varsamp	La variance d'échantillon des valeurs collectées pendant la période d'évaluation définie.

Paramètres communs

- `/host/key` est un premier paramètre commun obligatoire pour les fonctions faisant référence à l'historique de l'élément d'hôte
- `(sec|#num)<:time shift>` est un deuxième paramètre commun pour les fonctions faisant référence à l'historique de l'élément d'hôte, où :
  - **sec** - période d'évaluation maximale en secondes (les suffixes de temps peuvent être utilisés), ou
  - **#num** - plage d'évaluation maximale dans les dernières valeurs collectées (si elle est précédée d'un dièse)
  - **time shift** (facultatif) permet de déplacer le point d'évaluation dans le temps vers le passé. Voir 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)<:décalage temporel>)

La valeur moyenne 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 les 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) #la valeur moyenne de la dernière heure jusqu'à maintenant
avg(/host/key,1h:now-1d) #la valeur moyenne sur une heure, d'il y a 25 heures à il y a 24 heures à partir
avg(/host/key,#5) #la valeur moyenne des cinq dernières valeurs
avg(/host/key,#5:now-1d) #la valeur moyenne des cinq dernières valeurs, à l'exclusion des valeurs reçues a
```

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))`.

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 les fonctions foreach pour plus de détails. Notez que count_foreach et bucket_rate_foreach prennent en charge des paramètres supplémentaires.
- **item filter** - voir item filter;
- **time period** - voir time period;
- **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 binaire
regexp - correspondance sensible à la casse de l'expression régulière donnée dans `pattern`
iregexp - correspondance insensible à la casse de l'expression régulière donnée dans `pattern`
- **pattern** - le motif requis (les arguments de 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, tandis que l'utilisation de **exists_foreach()**, qui fonctionne uniquement avec les données de configuration, n'aura pas cet effet.
- Les paramètres optionnels *operator* ou *pattern* ne peuvent pas être laissés vides après une virgule, ils doivent soit 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 un « ET binaire » à partir de la valeur et du *mask* et compare le résultat à *number_to_compare_with*. Si le résultat du « ET binaire » 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. Aux fins de 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)) #le nombre d'éléments net.if.in ayant reçu des données au cours de
count(last_foreach(/*/vfs.fs.size[*,pused]),"gt",95) #le nombre de systèmes de fichiers avec plus de 95 %
```

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 de paramètres saisies manuellement (>=2) 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, permettant la référence 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"]) #le nombre d'hôtes avec l'élément *agent.ping* dans le "H
```

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 les paramètres communs.

Exemple :

```
kurtosis(/host/key,1h) #kurtosis pour la dernière heure jusqu'à maintenant
```

mad(/host/key,(sec|#num)<:time shift>)

L'écart absolu médian des valeurs collectées dans 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 les paramètres communs.

Exemple :

```
mad(/host/key,1h) #écart absolu médian pour la dernière heure jusqu'à maintenant
```

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 les paramètres communs.

Exemple :

```
max(/host/key,1h) - min(/host/key,1h) #calcule la différence entre les valeurs maximale et minimale au cou
```

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 les paramètres communs.

Exemple :

```
max(/host/key,1h) - min(/host/key,1h) #calcule la différence entre les valeurs maximale et minimale au cou
```

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.

Exemple :

```
skewness(/host/key,1h) #l'asymétrie pour la dernière heure jusqu'à maintenant
```

stddevpop(/host/key,(sec|#num)<:time shift>)

L'écart type de la population 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.

Exemple :

```
stddevpop(/host/key,1h) #l'écart type de la population pour la dernière heure jusqu'à maintenant
```

stddevsamp(/host/key,(sec|#num)<:time shift>)

L'écart-type d'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 les paramètres communs.

Au moins deux valeurs de données sont requises pour que cette fonction fonctionne.

Exemple :

```
stddevsamp(/host/key,1h) #l'écart-type d'échantillon pour la dernière heure jusqu'à maintenant
```

sum(/host/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 les paramètres communs.

Exemple :

```
sum(/host/key,1h) #la somme des valeurs de la dernière heure jusqu'à maintenant
```

sumofsquares(/host/key,(sec|#num)<:time shift>)

La somme des carrés des valeurs collectées pendant la période d'évaluation définie.
 Types de valeurs pris en charge : *Float*, *Integer*.
 Fonction foreach prise en charge : *last_foreach*.

Paramètres : voir les paramètres communs.

Exemple :

```
sumofsquares(/host/key,1h) #la somme des carrés pour la dernière heure jusqu'à maintenant
```

varpop(/host/key,(sec|#num)<:décalage temporel>)

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.

Exemple :

```
varpop(/host/key,1h) #la variance de population de la dernière heure jusqu'à maintenant
```

varsamp(/host/key,(sec|#num)<:time shift>)

La variance d'échantillon des valeurs collectées pendant 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.

Exemple :

```
varsamp(/host/key,1h) #la variance d'échantillon pour la dernière heure jusqu'à maintenant
```

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.

> **Attention:**
> Les fonctions foreach sont prises en charge uniquement dans les éléments calculés, dans le cadre des formules de calcul agrégé. Elles ne peuvent pas être appelées dans des clés d'élément simples ni dans des expressions de déclencheur en dehors de ce contexte.

Par exemple, la fonction *avg_foreach* renverra un tableau de valeurs, où chaque valeur correspond à la valeur d'historique *moyenne* de l'élément sélectionné, pendant l'intervalle de temps spécifié.

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

Fonction	Description
avg_foreach	Renvoie la valeur moyenne pour chaque élément.

Fonction	Description
bucket_rate_foreach	Renvoie des paires (limite supérieure du bucket, valeur du taux) pouvant être utilisées dans la fonction histogram_quantile(), où la « limite 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 de fonction

Les fonctions foreach prennent en charge deux paramètres communs : `item filter` (voir les détails ci-dessous) et `time period` :

```
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 parameters 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 de temps

Le **deuxième** paramètre permet de spécifier la période de temps pour l'agrégation. La période de temps ne peut être exprimée qu'en durée ; le nombre de valeurs (préfixé par **#**) n'est pas pris en charge.

Les symboles d'unité 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 de temps est un paramètre facultatif (pris en charge depuis Zabbix 7.0), qui peut être omis :

`last_foreach(/*/key?[group="host group"])`

La période de temps 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> correspond à la position de la valeur « bucket » dans la clé d'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 d'élément (doit être placé 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 binaire
regexp - correspondance sensible à la casse de l'expression régulière donnée dans `pattern`
iregexp - correspondance insensible à la casse de l'expression régulière donnée dans `pattern`
- **pattern** est le motif requis (les arguments de type chaîne doivent être placés entre guillemets doubles) ; pris en charge si *operator* est spécifié dans le troisième paramètre.

Remarques :

- Les paramètres facultatifs *operator* ou *pattern* ne peuvent pas être laissés vides après une virgule ; ils doivent être soit entièrement omis.
- Avec *bitand* comme troisième paramètre, le quatrième paramètre `pattern` peut être spécifié sous forme de deux nombres séparés par « / » : **number_to_compare_with/mask**. count_foreach() calcule le « ET binaire » entre la valeur et le *mask* et compare le résultat à *number_to_compare_with*. Si le résultat du « ET binaire » 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 des expressions régulières globales. Aux fins de la correspondance regexp, les valeurs flottantes seront toujours représentées avec 4 chiffres décimaux après le « . ». 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.

Voir calculs agrégés pour plus de détails et d'exemples sur l'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.

2 Fonctions bit à bit

Toutes les fonctions répertoriées ici sont prises en charge dans :

- Expressions de déclencheur
- Éléments calculés
- Macros d'expression

Les fonctions sont répertoriées sans informations supplémentaires. Cliquez sur la fonction pour voir tous les détails.

Function	Description
bitand	La valeur du « ET bit à bit » d'une valeur d'élément et d'un masque.
bitlshift	Le décalage à gauche bit à bit 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 à droite bit à bit 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.

3 Fonctions de date et d'heure

Toutes les fonctions répertoriées ici sont prises en charge dans :

- Expressions de déclencheur
- Éléments calculés
- 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 Moment du 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'époque Unix (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.

4 Fonctions d'historique

Toutes les fonctions répertoriées ici sont prises en charge dans :

- Expressions de déclencheur
- Éléments calculés
- Macros d'expression

Les fonctions sont répertoriées sans informations supplémentaires. Cliquez sur la fonction pour voir tous les détails.

Function	Description
change	La différence entre la valeur précédente et la dernière valeur.
changecount	Le nombre de changements entre des 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	Trouver 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 dans quelle mesure l'heure de l'agent passif diffère de celle du serveur/proxy Zabbix.
last	La valeur la plus récente.
lastclock	L'horodatage de la Nième 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 sévérité 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 Nième valeur la plus récente de l'élément journal.
monodec	Vérifier s'il y a eu une diminution monotone des valeurs.
monoinc	Vérifier s'il y a eu une augmentation monotone des valeurs.
nodata	Vérifier qu'aucune donnée n'a été reçue.
percentile	Le percentile P d'une période, où P (pourcentage) est spécifié par le troisième paramètre.
rate	Le taux moyen par seconde de l'augmentation d'un compteur monotone croissant au cours de la période définie.

Paramètres communs

- `/host/key` est un premier paramètre commun obligatoire pour les fonctions qui font référence à l'historique des éléments de l'hôte
- `(sec|#num)<:time shift>` est un deuxième paramètre commun pour les fonctions qui font référence à l'historique des éléments de l'hôte, où :
  - **sec** - période d'évaluation maximale en secondes (les suffixes de temps peuvent être utilisés), ou
  - **#num** - plage d'évaluation maximale dans les dernières valeurs collectées (si elle est précédée d'un dièse)
  - **time shift** (facultatif) permet de déplacer le point d'évaluation dans le passé. Voir plus de détails sur la manière de spécifier le 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
ge - supérieur ou égal
lt - inférieur
le - inférieur ou égal
like (par défaut pour string, text, log) - correspond si contient le motif (sensible à la casse)
bitand - ET binaire
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 un « ET binaire » à partir de la valeur et du *mask* et compare le résultat à *number_to_compare_with*. Si le résultat du « ET binaire » 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 des expressions régulières globales. 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 au cours des 10 dernières minutes jusqu'à mainter
```

```
count(/host/key,10m,,12) #le nombre de valeurs au cours des 10 dernières minutes jusqu'à maintenant égales
count(/host/key,10m,"gt",12) #le nombre de valeurs au cours des 10 dernières minutes jusqu'à maintenant su
count(/host/key,#10,"gt",12) #le nombre de valeurs parmi les 10 dernières valeurs jusqu'à maintenant supéri
count(/host/key,10m:now-1d,"gt",12) #le nombre de valeurs entre il y a 24 heures moins 10 minutes et il y
count(/host/key,10m,"bitand","6/7") #le nombre de valeurs au cours des 10 dernières minutes jusqu'à mainter
count(/host/key,10m:now-1d) #le nombre de valeurs entre il y a 24 heures moins 10 minutes et il y a 24 heu
```

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). `operators` 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 contient le motif (sensible à la casse)
bitand - ET binaire
regexp - correspondance sensible à la casse de l'expression régulière donnée dans `pattern`
iregexp - correspondance insensible à la casse de 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 la forme de deux nombres, séparés par '/': **number_to_compare_with/mask**. countunique() calcule le "ET binaire" à partir de la valeur et du *mask* et compare le résultat à *number_to_compare_with*. Si le résultat du "ET binaire" 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 :

```
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 maintena
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 il y a 24 heures et il y a 2
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 il y a 24 heures et il y a 24 heures
```

find(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

Trouve une correspondance de valeur dans la période d'évaluation définie.
 Types de valeurs pris en charge : *Float*, *Integer*, *String*, *Text*, *Log*.
 Renvoie : 1 - trouvé ; 0 - sinon.

Paramètres :

- Voir les paramètres communs
- **sec** ou **#num** (optionnel) - utilise par défaut la dernière valeur si non spécifié
- **operator** (doit être entre guillemets doubles) `operators` 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 la chaîne fournie dans `pattern` est contenue (sensible à la casse)
bitand - ET binaire
regexp - correspondance sensible à la casse de l'expression régulière fournie dans `pattern`
iregexp - correspondance insensible à la casse de l'expression régulière fournie 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 entières.
- *like* et *bitand* ne sont pas pris en charge comme opérateurs pour les valeurs flottantes.
- 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") #find a value that contains 'error' within the last 10 minutes until no
find(/host/agent.version,,"like","beta")=1 #Zabbix agent has beta version, must be upgraded
find(/host/log[/var/log/nginx/access.log],,"regexp"," 500 ")=1 #internal web server error has been found
```

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 dans quelle mesure l'heure de l'agent passif diffère de celle du serveur/proxy Zabbix.
 Types de valeurs pris en charge : *Float*, *Integer*.
 Renvoie : 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 les paramètres communs.

Commentaires :

- Généralement utilisé avec l'élément 'system.localtime' pour vérifier que l'heure locale est synchronisée avec l'heure locale du serveur Zabbix. *Notez* que 'system.localtime' doit être configuré comme une vérification passive pour l'agent Zabbix ; sur l'agent Zabbix 2, il peut être configuré comme une vérification active.
- Peut également être utilisé 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 dans les expressions de déclencheur complexes (impliquant plusieurs éléments), car elle peut provoquer des résultats inattendus (la différence de temps sera mesurée à l'aide de 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 celle du serveur/proxy Zabbix
fuzzytime(/host/system.localtime,5m)=0 and nodata(/host/system.localtime,10m)=0 #l'heure locale du client
```

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*.
 Renvoie : 0 - ne correspond pas ; 1 - correspond.

Paramètres :

- Voir les 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 type chaîne doivent être placés entre guillemets doubles)

Exemples :

```
logeventid(/host/eventlog[Security],,"4625")=1 #une entrée de journal avec un ID correspondant à "4625" (é
logeventid(/host/eventlog[System],,"6008|41")=1 #une entrée de journal avec un ID correspondant à "6008" c
```

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<:décalage temporel»,<pattern>)

Vérifie si la source du dernier enregistrement de journal correspond à une expression régulière.
 Types de valeurs pris en charge : *Log*.
 Renvoie : 0 - ne correspond pas ; 1 - correspond.

Paramètres :

- Voir les paramètres communs
- **#num** (facultatif) - la Nième valeur la plus récente
- **pattern** (facultatif) - l'expression régulière décrivant le motif requis, de style Perl Compatible Regular Expression (PCRE) (les arguments de type chaîne doivent être placés entre guillemets doubles)

Généralement utilisé pour les journaux d'événements Windows.

Exemples :

```
logsource(/host/eventlog[Application],,"MSSQLSERVER")=1 #a log entry with source matching "MSSQLSERVER" fo
logsource(/host/eventlog[System],,"Service Control Manager")=1 #a log entry with source matching "Service
logsource(/host/eventlog[System],,"Service Control Manager")=1 and logeventid(/host/eventlog[System],,"703
```

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 - sinon.

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)` n'est pas autorisé
- **mode** - si défini sur *strict* (entre guillemets doubles), cette fonction sera insensible à la disponibilité du proxy (voir les commentaires pour plus de détails)

Commentaires :

- les déclencheurs 'nodata' surveillés par proxy sont, par défaut, sensibles à 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 pour la période de retard. Notez que pour les proxys passifs, la suppression est activée si la connexion est rétablie plus de 15 secondes et au moins 2 secondes plus tard. Pour les proxys actifs, la suppression est activée si la connexion est rétablie plus de 15 secondes plus tard. Pour désactiver la sensibilité à 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, pendant la période du 1er paramètre :
- il n'y a pas de données et le serveur Zabbix a été redémarré
- il n'y a pas de données et la maintenance a été terminée
- il n'y a pas de données et l'élément a été ajouté ou réactivé.
- Les erreurs sont affichées dans la colonne *Info* de la configuration des déclencheurs.
- Cette fonction peut ne pas fonctionner correctement s'il existe des différences de temps entre le serveur Zabbix, le proxy et l'agent. Voir aussi : Exigence de synchronisation de l'heure.

Exemple :

```
nodata(/host/agent.ping,5m)=1 #déclencheur si aucune donnée n'a été reçue de l'agent Zabbix pendant 5 minu
```

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 secc
```

Voir toutes les fonctions prises en charge.

5 Fonctions de tendance

Les fonctions de tendance, contrairement aux fonctions d'historique, utilisent les 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 depuis la base de données qu'une seule fois. Le cache des fonctions de tendance est contrôlé par le paramètre serveur TrendFunctionCacheSize.

Les déclencheurs qui référencent **uniquement** des fonctions de tendance sont évalués une fois selon 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 des fonctions de date et heure et/ou nodata()), il est calculé conformément aux principes habituels.

Toutes les fonctions listées ici sont prises en charge dans :

- les expressions de déclencheur
- les éléments calculés
- les macros d'expression

Les fonctions sont listées sans informations supplémentaires. Cliquez sur la fonction pour voir tous les détails.

Function	Description
baselinedev	Renvoie le nombre d'écarts (selon l'algorithme stddevpop) entre la dernière période de données et les mêmes périodes de données des saisons précédentes.
baselinewma	Calcule la ligne de base en faisant la moyenne des 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 dans la période définie.
trendcount	Le nombre de valeurs d'historique récupérées avec succès utilisées pour calculer la valeur de tendance dans la période définie.
trendmax	Le maximum des valeurs de tendance dans la période définie.
trendmin	Le minimum des valeurs de tendance dans la période définie.
trendstl	Renvoie le taux d'anomalies pendant la période de détection, une valeur décimale comprise entre 0 et 1, soit `((le nombre de valeurs anormales)/(nombre total de valeurs))`.
trendsum	La somme des valeurs de tendance dans la période 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.

6 Fonctions mathématiques

Toutes les fonctions répertoriées ici sont prises en charge dans :

- Expressions de déclencheur
- Éléments calculés
- Macros d'expression

Les fonctions mathématiques sont prises en charge avec les types de valeurs float et integer, sauf indication contraire.

Les fonctions sont répertoriées sans informations supplémentaires. Cliquez sur la fonction pour voir tous les détails.

Function	Description
abs	La valeur absolue d'une valeur.
acos	L'arccosinus d'une valeur sous forme d'angle, exprimé en radians.
asin	L'arcsinus d'une valeur sous forme d'angle, exprimé en radians.
atan	L'arctangente d'une valeur sous forme d'angle, exprimé en radians.
atan2	L'arctangente de l'ordonnée (valeur) et de l'abscisse spécifiées sous forme d'angle, exprimé en radians.
avg	La valeur moyenne des valeurs d'élément référencées.
cbrt	La racine cubique d'une valeur.
ceil	Arrondit la valeur à l'entier supérieur ou égal le plus proche.
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 à la puissance d'une valeur.
expm1	Le nombre d'Euler à la puissance d'une valeur moins 1.
floor	Arrondit la valeur à l'entier inférieur ou égal le plus proche.
log	Le logarithme naturel.
log10	Le logarithme décimal.
max	La valeur la plus élevée parmi les valeurs d'élément référencées.
min	La valeur la plus faible parmi les valeurs d'élément référencées.
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	Renvoie une valeur entière aléatoire.
round	Arrondit la valeur à un certain nombre de décimales.
signum	Renvoie '-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 d'élément référencées.
tan	La tangente d'une valeur.
truncate	Tronque la valeur à un certain 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

La valeur doit être comprise entre -1 et 1. Par exemple, l'arc tangente de la valeur « 1 » sera « 0.785398163 ».

Exemple :

```
atan(last(/host/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 travaille avec l'historique des éléments.

Exemple :

```
avg(avg(/host/key),avg(/host2/key2))
```

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 travaille avec l'historique des éléments.

Exemple :

```
max(avg(/host/key),avg(/host2/key2))
```

min(<value1>,<value2>,...)

La valeur la plus basse 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),avg(/host2/key2))
```

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, où la valeur est un angle exprimé en radians.

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 travaille avec l’historique des éléments.

Exemple :

```
sum(avg(/host/key),avg(/host2/key2))
```

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.

7 Fonctions d'opérateur

Toutes les fonctions listées 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 tous les détails.

Function	Description
between	Vérifie si la valeur appartient à la plage donnée.
in	Vérifie si la valeur est égale à au moins une des valeurs listées.

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.

8 Fonctions prédictives

Toutes les fonctions listées 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 tous les détails.

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 commun obligatoire pour les fonctions faisant référence à l'historique de l'élément d'hôte
- `(sec|#num)<:time shift>` est un deuxième paramètre commun pour les fonctions faisant référence à l'historique de l'élément d'hôte, où :
  - **sec** - période d'évaluation maximale en secondes (les suffixes de temps peuvent être utilisés), ou
  - **#num** - plage d'évaluation maximale dans les dernières valeurs collectées (si elle est précédée d'un dièse)
  - **time shift** (facultatif) permet de déplacer le point d'évaluation dans le temps vers le passé. Voir 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, delta ou moyenne de l'élément.
 Types de valeurs pris en charge : *Float*, *Integer*.

Paramètres :

- Voir paramètres communs;
- **time** - l'horizon de prévision en secondes (les suffixes temporels peuvent être utilisés) ; les valeurs négatives sont prises en charge ;
- **fit** (facultatif ; doit être 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 entre guillemets doubles) - la sortie demandée. 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 à l'instant `now + time` ; *max*, *min*, *delta* et *avg* examinent l'estimation de la valeur de l'élément sur l'intervalle entre `now` et `now + time`.

Commentaires :

- Si la valeur à renvoyer est supérieure à 1.7976931348623158E+308 ou inférieure à -1.7976931348623158E+308, la valeur de retour est ramenée respectivement à 1.7976931348623158E+308 ou -1.7976931348623158E+308 ;
- Devient non pris en charge uniquement s'il est mal utilisé dans l'expression (mauvais type d'élément, paramètres invalides), sinon renvoie -1 en cas d'erreurs ;
- Voir également des informations supplémentaires sur les fonctions prédictives de déclencheur.

Exemples :

```
forecast(/host/key,#10,1h) #prévoit la valeur de l'élément dans une heure sur la base des 10 dernières val
forecast(/host/key,1h,30m) #prévoit la valeur de l'élément dans 30 minutes sur la base des données de la d
forecast(/host/key,1h:now-1d,12h) #prévoit la valeur de l'élément dans 12 heures sur la base d'une heure d
forecast(/host/key,1h,10m,"exponential") #prévoit la valeur de l'élément dans 10 minutes sur la base des d
forecast(/host/key,1h,2h,"polynomial3","max") #prévoit la valeur maximale que l'élément peut atteindre dan
forecast(/host/key,#2,-20m) #estime la valeur de l'élément il y a 20 minutes sur la base des deux dernière
```

timeleft(/host/key,(sec|#num)<:time shift>,threshold,<fit>)

Le temps en secondes nécessaire pour qu'un élément de données atteigne le seuil spécifié.
 Types de valeurs pris en charge : *Float*, *Integer*.

Paramètres :

- Voir les paramètres communs ;
- **threshold** - la valeur à atteindre (les suffixes d'unité peuvent être utilisés) ;
- **fit** (facultatif ; doit être entre guillemets doubles) - voir forecast().

Commentaires :

- Si la valeur à renvoyer est supérieure à 1.7976931348623158E+308, la valeur renvoyée est ramenée à 1.7976931348623158E+308 ;
- Renvoie 1.7976931348623158E+308 si le seuil ne peut pas être atteint ;
- Devient non pris en charge uniquement en cas d'utilisation incorrecte dans l'expression (type d'élément de données incorrect, paramètres invalides), sinon renvoie -1 en cas d'erreurs ;
- Voir également les informations supplémentaires sur les fonctions de déclencheur prédictives.

Exemples :

```
timeleft(/host/key,#10,0) #le temps jusqu'à ce que la valeur de l'élément de données atteigne zéro sur la
timeleft(/host/key,1h,100) #le temps jusqu'à ce que la valeur de l'élément de données atteigne 100 sur la
timeleft(/host/key,1h:now-1d,100) #le temps jusqu'à ce que la valeur de l'élément de données atteigne 100
timeleft(/host/key,1h,200,"polynomial2") #le temps jusqu'à ce que la valeur de l'élément de données atteig
```

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'à quel point 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, qu'il l'atteigne par le haut ou par le bas. Une

fois que nous avons déterminé f(t) (voir ci-dessous), nous devons résoudre l'équation f(t) = `threshold` et renvoyer la racine la plus proche du moment présent 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. La bonne pratique consiste à utiliser les prédictions en complément du diagnostic ordinaire 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)
À chaque fois qu'une fonction de déclencheur est évaluée, elle récupère 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 donné 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)`

9 Fonctions de chaîne

Toutes les fonctions répertoriées ici sont prises en charge dans :

- Expressions de déclencheur
- Éléments calculés
- Macros d'expression

Les fonctions sont répertoriées sans informations supplémentaires. Cliquez sur la fonction pour voir tous les détails.

Function	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.
concat	La chaîne résultant de la concaténation des valeurs d'élément référencées ou de valeurs constantes.
insert	Insère les caractères ou espaces spécifiés dans la chaîne de caractères à partir de la position indiquée 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 du début de la chaîne.
mid	Renvoie une sous-chaîne de N caractères à partir de la position de caractère indiquée par « start ».
repeat	Répète une chaîne.

Function	Description
replace	Recherche le motif dans la valeur et le remplace par la valeur de remplacement.
right	Renvoie les caractères les plus à droite de la valeur.
rtrim	Supprime les caractères spécifiés de la fin de la chaîne.
trim	Supprime les caractères spécifiés du début et de 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])
```

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 à partir de la position de caractère spécifiée par « start ».
 Types de valeurs pris en charge : *String*, *Text*, *Log*.

Paramètres :

- **value** - la valeur à vérifier ;
- **start** - position de début de la sous-chaîne ;
- **length** - 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 départ 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
```

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.

4 Bonnes pratiques pour les déclencheurs

Les déclencheurs sont un outil puissant, mais ils peuvent aussi générer du bruit d'alerte indésirable. Pour voir davantage de vrais signaux et moins de bruit, suivez ces conseils :

1. Rendez les déclencheurs moins sensibles. Au lieu d'alerter 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 battement des déclencheurs — des changements fréquents de l'état du déclencheur (OK ↔ Problème). Une continuité de 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 graves.
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.

Aperçu

Il arrive parfois que la disponibilité d'un hôte dépende d'un autre. Un serveur situé derrière un routeur deviendra injoignable si le routeur tombe en panne. Avec des déclencheurs configurés pour les deux, vous pourriez recevoir des notifications concernant deux hôtes indisponibles, alors qu'en réalité seul le routeur est à l'origine du problème.

C'est dans ce cas qu'une dépendance entre hôtes peut être utile. Lorsqu'une dépendance est définie, les notifications des éléments dépendants peuvent être retenues et seule la notification du problème racine est envoyée.

Bien que Zabbix ne prenne pas directement en charge les dépendances entre hôtes, elles peuvent être définies par une autre méthode, plus flexible : les dépendances de déclencheurs. Un déclencheur peut dépendre d'un ou de plusieurs autres déclencheurs.

Dans notre exemple simple, nous ouvrons donc le formulaire de configuration du déclencheur du serveur et définissons qu'il dépend du déclencheur correspondant du routeur. Avec une telle dépendance, le déclencheur du serveur ne changera pas d'état tant que le déclencheur dont il dépend est dans l'état « PROBLEM » ; ainsi, aucune action dépendante ne sera exécutée et aucune notification ne sera envoyée.

Si à la fois le serveur et le routeur sont indisponibles et qu'une dépendance est définie, Zabbix n'exécutera pas d'actions pour le déclencheur dépendant.

Tant que le déclencheur parent est dans l'état PROBLEM, ses dépendants peuvent signaler des valeurs non 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 change pas d'état de « PROBLEM » à « UNKNOWN » ;
- n'est pas fermé manuellement, par corrélation ou à l'aide des fonctions date and time et/ou nodata() ;
- n'est pas résolu par une valeur d'un élément non impliqué dans le déclencheur dépendant ;
- n'est pas désactivé, n'a pas un élément désactivé ou un hôte d'élément désactivé

Dans tous les cas mentionnés ci-dessus, le déclencheur dépendant (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.

Également :

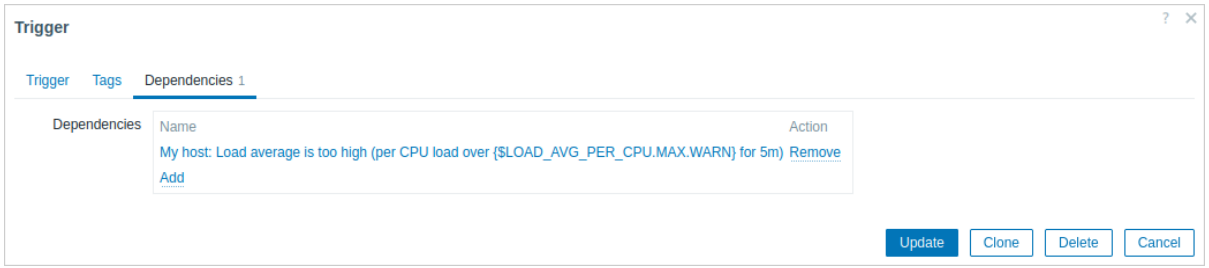
- Une dépendance de déclencheur peut être ajoutée depuis le déclencheur de n'importe quel hôte vers le déclencheur de n'importe quel autre hôte, tant que cela ne crée pas de dépendance circulaire.
- Une dépendance de déclencheur peut être ajoutée d'un modèle à un autre. Si un déclencheur du modèle A 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, mais le modèle B peut être lié seul à un hôte (ou à un autre modèle).
- Une dépendance de déclencheur peut être ajoutée d'un déclencheur de modèle à un déclencheur d'hôte. Dans ce cas, l'association d'un tel modèle à un hôte créera un déclencheur d'hôte qui dépend du même modèle de déclencheur que celui

dont le déclencheur dépendait. Cela permet, par exemple, d'avoir un modèle dans lequel certains déclencheurs dépendent des déclencheurs du routeur (hôte). Tous les hôtes liés à ce modèle dépendront de ce routeur spécifique.

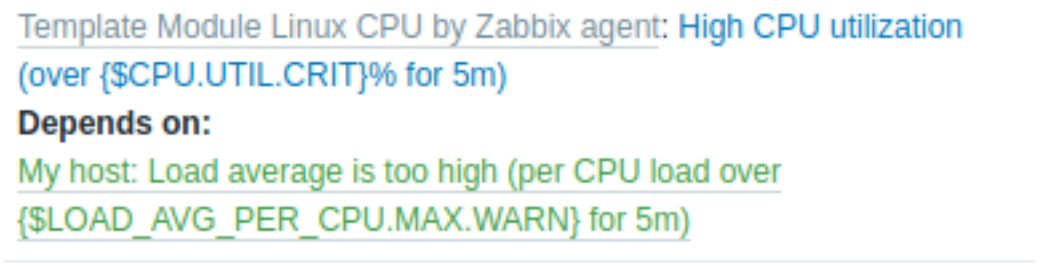
- Une dépendance de déclencheur ne peut pas être ajoutée d'un déclencheur d'hôte à un déclencheur de modèle.
- Une dépendance de déclencheur peut être ajoutée d'un prototype de déclencheur à un autre prototype de déclencheur (au sein de la même règle de découverte de bas niveau) ou à un déclencheur réel. Un prototype de déclencheur ne peut pas dépendre d'un prototype de déclencheur provenant d'une règle LLD différente 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, ouvrez l'onglet Dépendances dans le formulaire de configuration du déclencheur. Cliquez sur *Ajouter* dans le bloc « Dépendances » et sélectionnez un ou plusieurs déclencheurs dont le déclencheur dépendra.



Cliquez sur *Mettre à jour*. Le déclencheur affiche désormais l'indication de sa dépendance dans la liste.



Exemple de plusieurs dépendances

Par exemple, l'hôte se trouve derrière le Routeur2 et le Routeur2 se trouve derrière le Routeur1.

```
Zabbix - Routeur1 - Routeur2 - Hôte
```

Si le Routeur1 est indisponible, alors il est évident que l'hôte et le Routeur2 sont également injoignables, mais recevoir trois notifications indiquant que l'hôte, le Routeur1 et le Routeur2 sont tous indisponibles est excessif.

Dans ce cas, nous définissons donc deux dépendances :

```
le déclencheur « l'hôte est indisponible » dépend du déclencheur « le Routeur2 est indisponible »
le déclencheur « le Routeur2 est indisponible » dépend du déclencheur « le Routeur1 est indisponible »
```

Avant de modifier l'état du déclencheur « l'hôte est indisponible », Zabbix va vérifier les dépendances de déclencheur correspondantes. Si de telles dépendances sont trouvées et que l'un de ces déclencheurs est à 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 le Routeur1 ou le Routeur2 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.

4 Événements

Aperçu

Il existe plusieurs types d'événements générés dans Zabbix :

- événements de déclencheur - chaque fois qu'un déclencheur change d'état (*OK→PROBLEM→OK*)
- événements de service - chaque fois qu'un service change d'état (*OK→PROBLEM→OK*)
- événements de découverte - lorsque des hôtes ou des services sont détectés
- événements d'auto-enregistrement - 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'un e-mail de notification, etc.

Pour afficher les détails des événements dans l'interface web, allez dans *Monitoring* → *Problems*. Vous pouvez y cliquer sur la date et l'heure de l'événement pour afficher les détails d'un événement.

Plus d'informations sont disponibles sur :

- événements de déclencheur
- autres sources d'événements

1 Génération d'événement déclencheur

Vue d'ensemble

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 : le moment où il s'est produit et le nouvel état.

Deux types d'événements sont créés par les déclencheurs : `Problem` et `OK`.

Événements problèmes

Un événement problème est créé :

- lorsqu'une expression de déclencheur est évaluée à VRAI si le déclencheur est dans l'état OK ;
- chaque fois qu'une expression de déclencheur est évaluée à VRAI si la génération multiples d'événements problèmes est activée pour le déclencheur.

Événements OK

Un événement OK ferme le ou les événement(s) problème lié(s) et peut être créé par 3 composants :

- les déclencheurs - basés sur les paramètres 'Génération d'événement OK' et 'Un événement OK ferme' ;
- la corrélation d'événements
- le gestionnaire de tâches - lorsqu'un événement est fermé manuellement

Déclencheurs

Les déclencheurs ont un paramètre 'Génération d'événement OK' qui contrôle la manière dont les événements OK sont générés :

- *Expression* - un événement OK est généré pour un déclencheur en état 'problème' lorsque son expression est évaluée à FAUX. C'est le paramètre le plus simple, activé par défaut.
- *Expression de récupération* : un événement OK est généré pour un déclencheur en état 'problème' lorsque son expression est évaluée à FAUX et que l'expression de récupération est VRAIE. Cela peut être utilisé si les critères de récupération du déclencheur sont différents des critères du problème.
- *Aucun* - un événement OK n'est jamais généré. Cela peut être utilisé conjointement avec la génération multiple d'événements 'problèmes' pour simplement envoyer une notification lorsque quelque chose se produit.

De plus, les déclencheurs ont un paramètre 'Un événement OK ferme' qui contrôle quels événements sont fermés :

- *Tous les problèmes* - un événement OK fermera tous les problèmes ouverts créés par le déclencheur
- *Tous les problèmes si les valeurs de tag correspondent* - 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 déclenchement 'Tag pour la correspondance'. S'il n'y a aucun problème à fermer, l'événement OK n'est pas généré. Ceci est souvent appelé corrélation d'événement au niveau du déclenchement.

Corrélation d'événements

La corrélation d'événements (également appelée corrélation d'événements globale) permet de configurer des règles personnalisées de fermeture d'événement (entraînant la création d'événements OK).

Les règles définissent la manière dont les nouveaux événements problèmes sont associés à des événements problèmes existants et permettent de fermer le nouvel événement ou les événements correspondants en générant des événements OK correspondants.

Cependant, la corrélation d'événements doit être configurée très soigneusement, car elle peut affecter négativement les performances de traitement des événements ou, si elle est mal configurée, fermer plus d'événements que prévu (dans le pire des cas, tous les problèmes peuvent être fermés). Quelques conseils de configuration :

1. toujours réduire la portée de la corrélation en définissant un tag unique pour la vérification de l'événement (l'événement associé à des événements anciens) et en utilisant la condition de corrélation 'nouveau tag d'événement'

2. ne pas oublier d'ajouter une condition basée sur l'ancien événement lors de l'utilisation de l'opération 'fermer l'ancien événement', ou tous les problèmes existants pourraient être fermés
3. évitez d'utiliser des noms de tag identiques utilisés 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 le frontend lors de la mise à jour d'un problème. L'événement n'est pas fermé directement : à la place, une tâche de « close event » est créée, puis traitée peu après par le gestionnaire de tâches. Le gestionnaire de tâches générera un événement OK correspondant et l'événement de problème sera fermé.

2 Autres sources d'événements

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 IP définies dans les règles de découverte du réseau. La fréquence de la vérification est configurable pour chaque règle individuellement. Une fois qu'un hôte ou un service est découvert, un événement de découverte (ou plusieurs événements) est généré.

Zabbix génère les événements suivants :

Événement	Généré quand
Service Up	Chaque fois que Zabbix détecte un service actif.
Service Down	Chaque fois que Zabbix ne peut pas détecter le service.
Hôte Up	Si au moins un des services est UP pour l'IP.
Hôte Down	Si tous les services ne répondent pas.
Service découvert	Si le service est de retour après un temps d'arrêt ou découvert pour la première fois.
Service perdu	Si le service est perdu après avoir été activé.
Hôte découvert	Si l'hôte est de retour après un temps d'arrêt ou découvert pour la première fois.
Hôte perdu	Si l'hôte est perdu après avoir été activé.

Événements d'autoréenregistrement de l'agent actif

L'autoréenregistrement de l'agent actif crée des événements dans Zabbix.

S'il est configuré, un événement d'autoréenregistrement de l'agent actif est créé lorsqu'un agent actif auparavant inconnu demande des contrôles ou si les métadonnées de l'hôte ont changé. Le serveur ajoute un nouvel hôte autoréenregistré en utilisant l'adresse IP et le port de l'agent reçus.

Pour plus d'informations, consultez la page autoréenregistrement de l'agent actif.

Événements internes

Les événements internes se produisent lorsque :

- un élément change d'état de « normal » à « non pris en charge »
- un élément change d'état de « non pris en charge » à « normal »
- une règle de découverte de bas niveau change d'état de « normal » à « non pris en charge »
- une règle de découverte de bas niveau change d'état de « non pris en charge » à « normal »
- un déclencheur change d'état de « normal » à « inconnu »
- un déclencheur change d'état de « inconnu » à « normal »

L'objectif de l'introduction des événements internes est de permettre aux utilisateurs d'être notifiés 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 des événements internes (par exemple, pour les éléments devenant non pris en charge), désactivez toutes les actions pour les événements internes dans Alertes → Actions → Actions internes.

Note:
Si les actions internes sont désactivées, alors qu'un objet est dans l'état « non pris en charge », l'événement de récupération pour cet objet sera tout de même créé.

Si les actions internes sont activées, alors qu'un objet est dans l'état « non pris en charge », l'événement de récupération pour cet objet sera créé, même si l'« événement de problème » n'a pas été créé pour l'objet.

Voir aussi : Réception de notifications pour les éléments non pris en charge

3 Fermeture manuelle des problèmes

Aperçu

Bien que, de manière générale, 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 signaler que certains paramètres du noyau doivent être ajustés pour des performances optimales. Dans ce cas, le problème est signalé aux administrateurs Linux, ils 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. Le processus *task manager* exécute ensuite cette tâche et génère un événement OK, fermant ainsi l'événement de problème.

L'événement *OK* généré inclut l'ensemble complet des tags d'événement résolus pour cet événement (y compris les tags hérités des modèles, des hôtes et des déclencheurs). Ces tags sont disponibles dans les notifications et dans des macros telles que `{EVENT.RECOVERY.TAGS}` et `{EVENT.RECOVERY.TAGSJSON}`.

Un problème 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 de nouveau entraîner un problème :

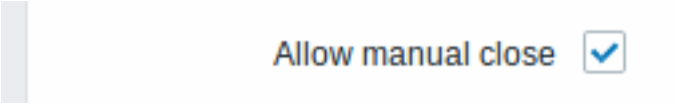
- Lorsque de nouvelles données arrivent pour n'importe quel élément inclus dans l'expression du déclencheur (notez que les valeurs ignorées par une étape de prétraitement de limitation de débit ne sont pas considérées comme reçues et ne provoqueront pas la réévaluation de l'expression du déclencheur) ;
- Lorsque les fonctions date and time et/ou nodata() sont utilisées dans l'expression.

Configuration

Deux étapes sont nécessaires pour résoudre un problème manuellement.

Configuration du déclencheur

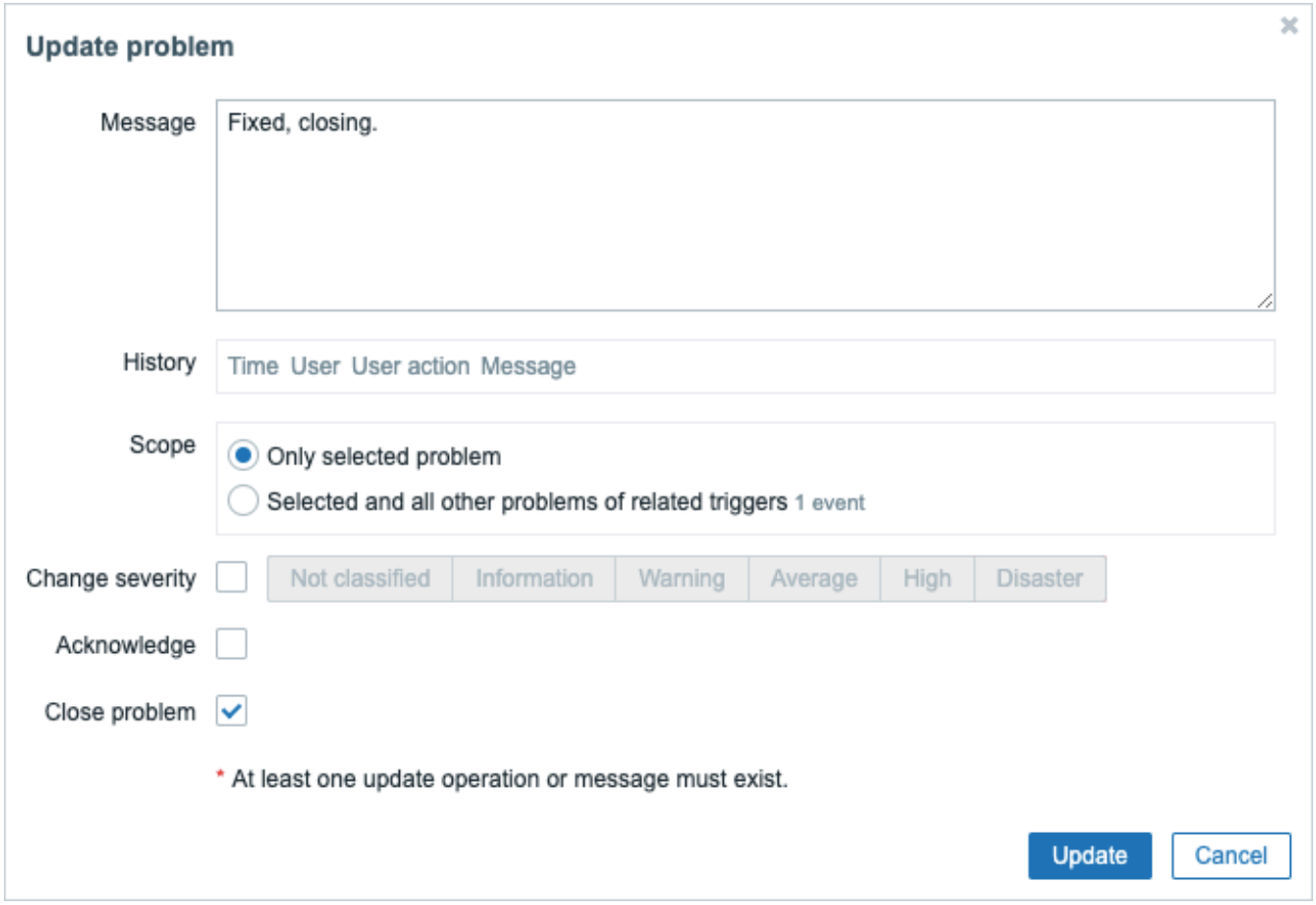
Dans la configuration du déclencheur, activez l'option *Autoriser la fermeture manuelle*.



Fenêtre de mise à jour du problème

Si un problème survient pour un déclencheur avec l'indicateur *Manual close*, vous pouvez ouvrir la fenêtre contextuelle mise à jour du problème de ce problème et fermer le problème manuellement.

Pour fermer le problème, cochez l'option *Close problem* dans le formulaire, puis cliquez sur *Update*.



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

La requête est traitée par le serveur Zabbix. Normalement, quelques secondes suffisent 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.

5 Corrélation d'événements

Aperçu

La corrélation d'événements permet de corréler les événements problématiques 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 relier des problèmes distincts à leur solution
- Globalement - les problèmes peuvent être corrélés à leur solution à partir d'une autre méthode de déclenchement ou d'interrogation à l'aide de règles de corrélation globales

1 Corrélation d'événements basée sur les déclencheurs

Vue d'ensemble

La corrélation d'événements basée sur les déclencheurs permet de corréler des problèmes distincts signalés par un même déclencheur.

Bien qu'en général un événement OK puisse fermer tous les événements de problème créés par un même déclencheur, il existe des cas où une approche plus détaillée est nécessaire. Par exemple, lors de la surveillance de fichiers journaux, vous pouvez vouloir détecter certains problèmes dans un fichier journal et les fermer 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 normalement utilisés pour la surveillance des journaux, le traitement des traps, etc.

Dans Zabbix, il est possible d'associer des événements de problème sur la base des tags. Les tags sont utilisés pour extraire des valeurs et créer une identification pour les événements de problème. En tirant parti de cela, les problèmes peuvent également être fermés individuellement sur la base d'un tag correspondant.

En d'autres termes, un même déclencheur peut créer des événements distincts identifiés par le tag de l'événement. Ainsi, les événements de problème peuvent être identifiés un par un et fermés séparément sur la base de l'identification par le tag de l'événement.

Fonctionnement

Lors de la surveillance des journaux, vous pouvez rencontrer des lignes similaires à celles-ci :

```
Ligne1 : Service 1 arrêté
Ligne2 : Service 2 arrêté
Ligne3 : Service 1 a été redémarré
Ligne4 : Service 2 a été redémarré
```

L'idée de la corrélation d'événements est de pouvoir faire correspondre l'événement de problème de la Ligne1 avec la résolution de la Ligne3, et l'événement de problème de la Ligne2 avec la résolution de la Ligne4, puis de fermer ces problèmes un par un :

```
Ligne1 : Service 1 arrêté
Ligne3 : Service 1 a été redémarré #problème de la Ligne 1 fermé

Ligne2 : Service 2 arrêté
Ligne4 : Service 2 a été redémarré #problème de la Ligne 2 fermé
```

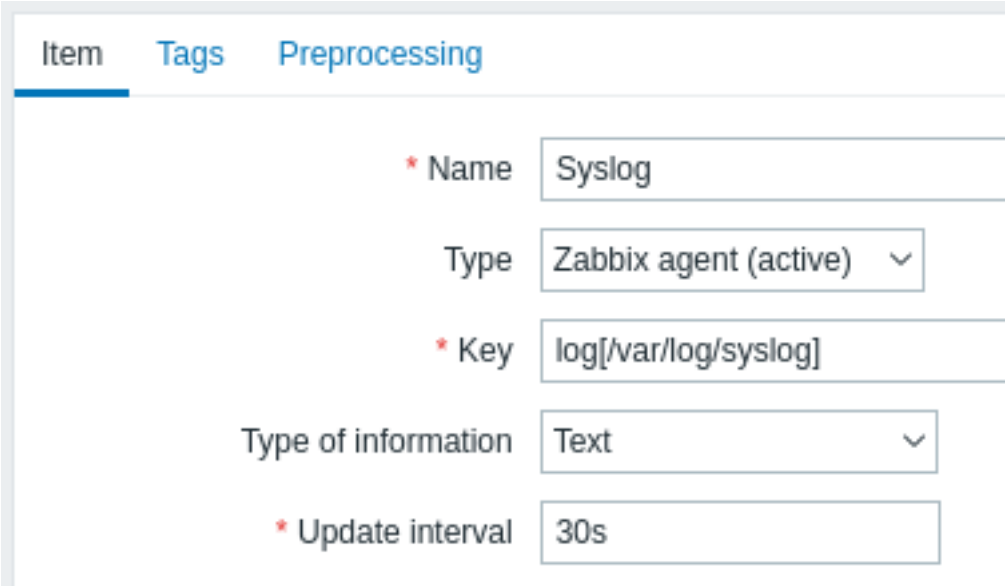
Pour ce faire, vous devez étiqueter ces événements liés, par exemple, avec « Service 1 » et « Service 2 ». Cela peut être fait en appliquant une expression régulière à la ligne du journal afin d'extraire la valeur de l'étiquette. Ensuite, lorsque les événements sont créés, ils sont étiquetés respectivement « Service 1 » et « Service 2 » et le problème peut être mis en correspondance avec la résolution.

Configuration

Elément

Pour commencer, vous souhaiterez peut-être 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 récupérées, puis accédez aux Dernières données pour vous assurer que l'élément a commencé à collecter des données.

Déclencheur

Une fois l'élément opérationnel, 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 telle que 'Stopping' afin de 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’.

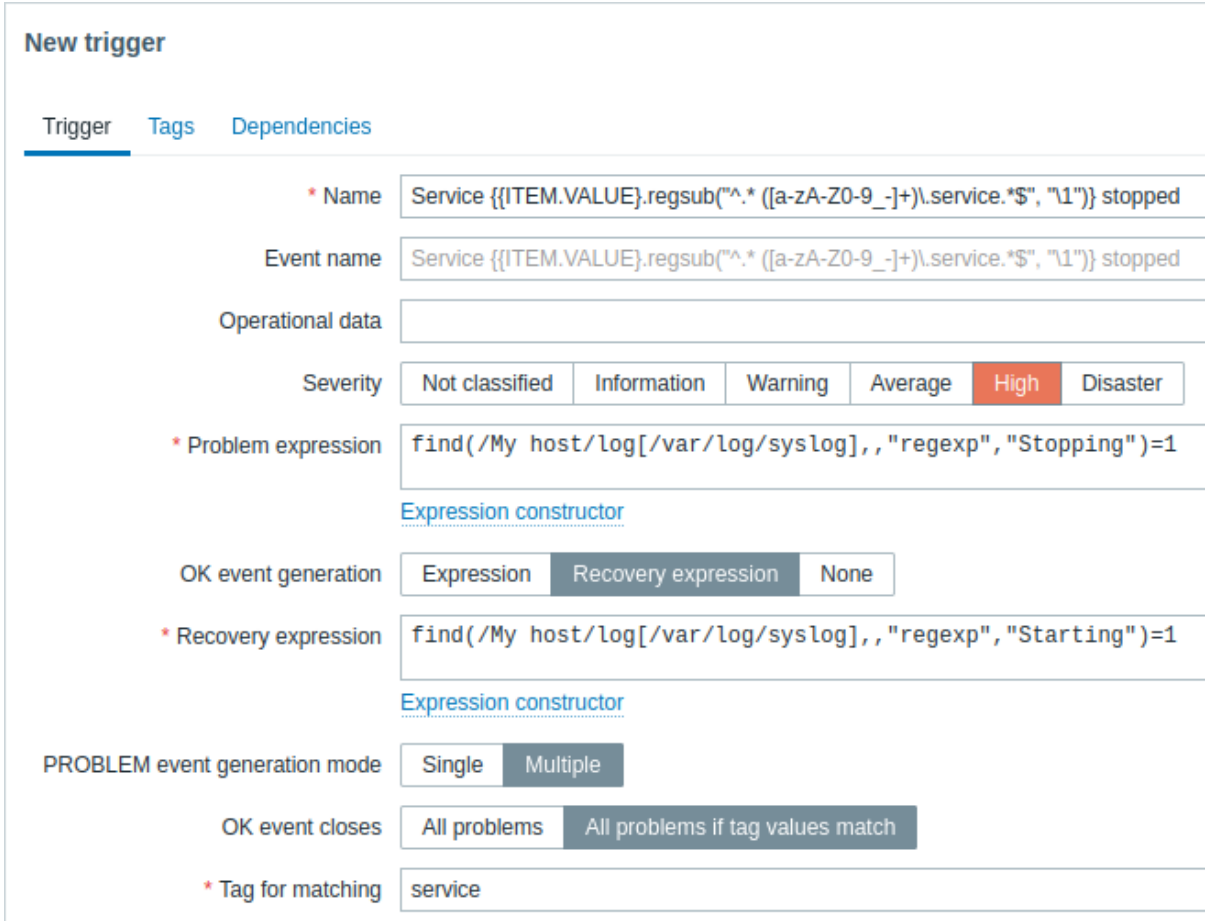
Définissez ensuite une expression de récupération. L’expression de récupération suivante résoudra tous les problèmes si une ligne du journal contenant la chaîne ”Starting” est trouvée :

```
find(/My host/log[/var/log/syslog],,"regexp","Starting")=1
```

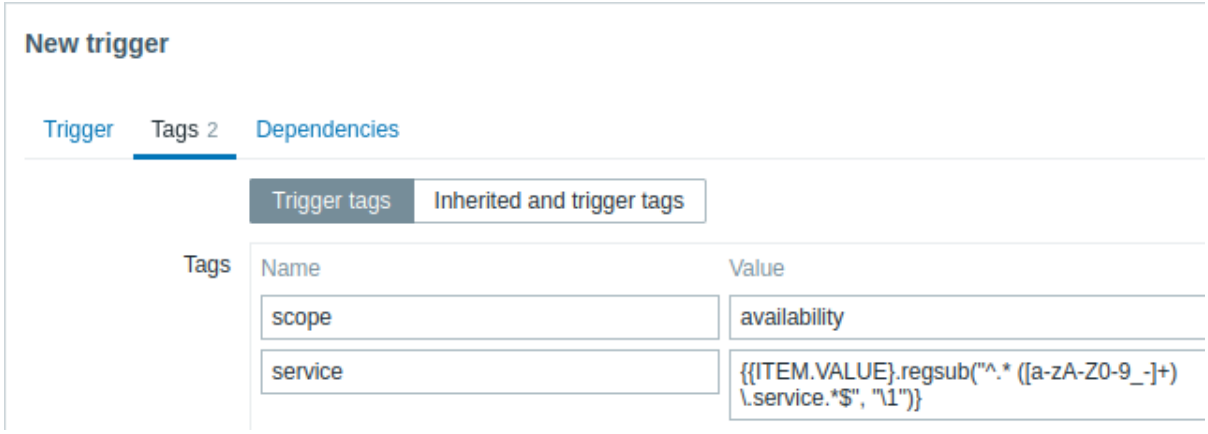
Comme ce n’est pas ce que nous voulons, il est important de s’assurer d’une manière ou d’une autre que les problèmes racine correspondants soient fermés, et pas simplement tous les problèmes. C’est là que le marquage par tags peut aider.

Les problèmes et leurs résolutions peuvent être associés en spécifiant un tag dans la configuration du déclencheur. Les paramètres suivants doivent être définis :

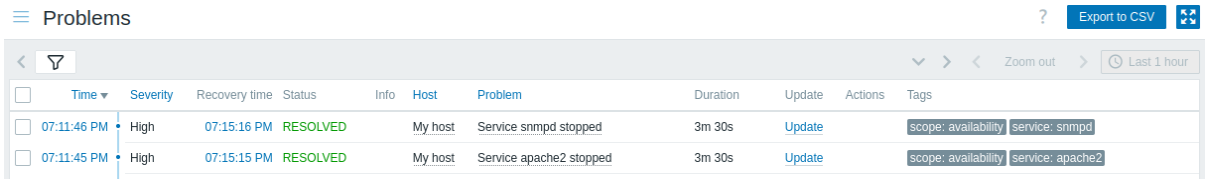
- *Mode de génération des événements de problème* : Multiple
- *Les événements OK ferment* : Tous les problèmes si les valeurs des tags correspondent
- Saisissez le nom du tag pour la correspondance des événements



- configurez les tags pour extraire les valeurs de tag à partir des lignes du journal



Si la configuration est correcte, vous pourrez voir les événements de problème marqués par application et associés à leur résolution dans *Surveillance → Problèmes*.



Warning:
Comme une mauvaise configuration est possible, lorsque des tags d'événement similaires peuvent être créés pour des problèmes **sans rapport**, veuillez examiner les cas décrits ci-dessous !

- Les macros indexées se réfèrent toujours au champ *Expression* de la configuration du déclencheur, et non à l'*Expression de récupération*. Par exemple, dans un événement de récupération, **{ITEM.VALUE1}** correspondra à la dernière valeur du premier élément **dans l'expression de problème** au moment de la récupération. Si l'expression de récupération est basée sur un élément différent et que la valeur de l'élément de l'expression de problème change au moment de la récupération, les événements recevront des tags différents et ne seront pas corrélés.

- Avec deux applications écrivant des messages d'erreur et de récupération 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 de tag pour extraire les noms, par exemple, du service A et du service B à partir de la macro {ITEM.VALUE} (par ex. lorsque les formats de message 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 à les corréler. Ainsi, un message de récupération du service A peut accidentellement fermer un message d'erreur du service B.
- Les tags réels et leurs valeurs 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, des événements OK ultérieurs avec la même valeur de tag *UNKNOWN* peuvent apparaître et 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 limitation à 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'événement similaires pour des problèmes sans rapport.

2 Corrélation globale des événements

Aperçu

La corrélation globale des événements permet de prendre en compte l'ensemble des 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 en effet possible de vous épargner des milliers de notifications répétitives et de vous concentrer sur les causes profondes d'un problème !

La corrélation globale des événements est un mécanisme puissant, qui vous permet de vous affranchir de la logique de problème et de résolution basée sur un seul déclencheur. Jusqu'à présent, un seul événement de problème était créé par un seul déclencheur et nous dépendions de ce même déclencheur pour la résolution du problème. Nous ne pouvions pas résoudre un problème créé par un déclencheur à l'aide d'un autre déclencheur. Mais avec la corrélation des événements basée sur le balisage des événements, c'est possible.

Par exemple, un déclencheur de journal peut signaler des problèmes d'application, tandis qu'un déclencheur d'interrogation peut signaler que l'application est active et fonctionne. En tirant parti des balises d'événement, vous pouvez attribuer au déclencheur de journal la balise *status:down* et au déclencheur d'interrogation la balise *status:up*. Ensuite, dans une règle de corrélation globale, vous pouvez mettre ces déclencheurs en relation et attribuer à cette corrélation une opération appropriée, 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 si nous pouvions ne recevoir qu'un seul rapport de problème par problème de port réseau ? Il n'est pas nécessaire de tous les signaler. Cela aussi est possible avec la corrélation globale des événements.

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 ce qu'il faut faire 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 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 Admin.

> **Attention:**
> La corrélation des événements doit être configurée avec beaucoup de précaution, car elle peut affecter négativement les performances du traitement des événements ou, en cas de mauvaise configuration, fermer plus d'événements que prévu (dans le pire des cas, même tous les événements de problème pourraient être fermés).

Pour configurer la corrélation globale **en toute sécurité**, respectez les conseils importants suivants :

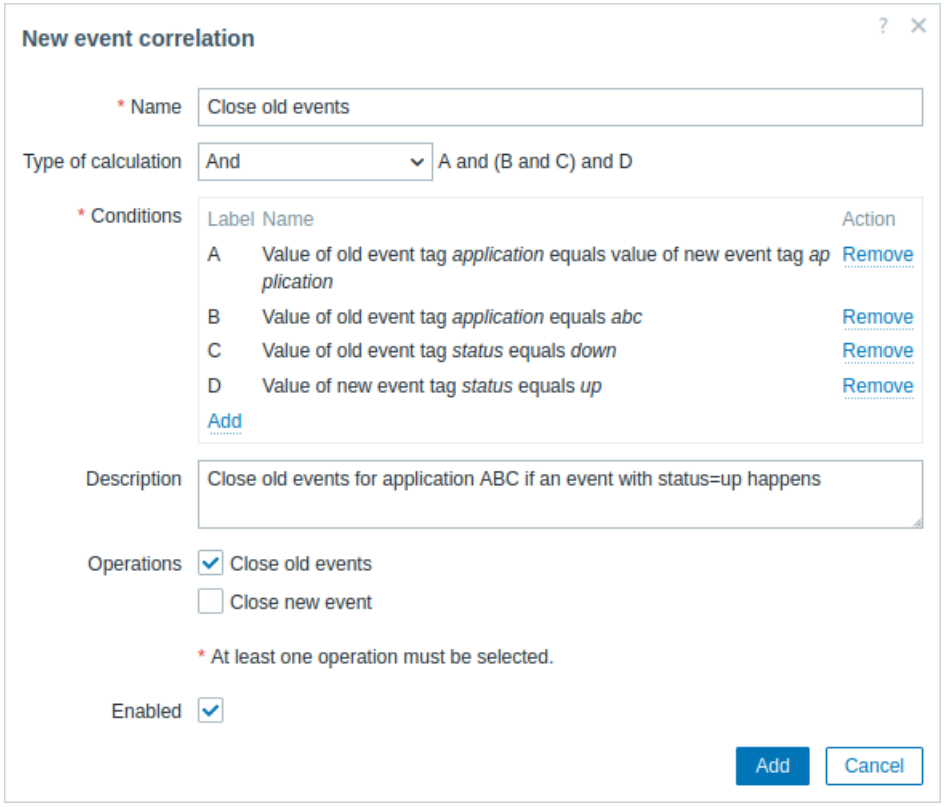
- Réduisez la portée de la corrélation. Définissez toujours une balise unique pour le nouvel événement associé aux anciens événements et utilisez la condition de corrélation *New event tag name* ;
- Ajoutez une condition basée sur l'ancien événement lors de l'utilisation de l'opération *Close old event* (sinon, tous les problèmes existants pourraient être fermés) ;
- Évitez d'utiliser des noms de balise courants qui pourraient finir par être utilisés par différentes configurations de corrélation ;
- Limitez le nombre de règles de corrélation à celles dont vous avez réellement besoin.

Voir aussi : problèmes connus.

Configuration

Pour configurer globalement les règles de corrélation d'événements :

- Allez dans *Data collection* > *Event correlation*
- Cliquez sur *Create event correlation* à droite (ou sur le nom de la corrélation pour modifier une règle existante)
- Saisissez les paramètres de la règle de corrélation dans le formulaire

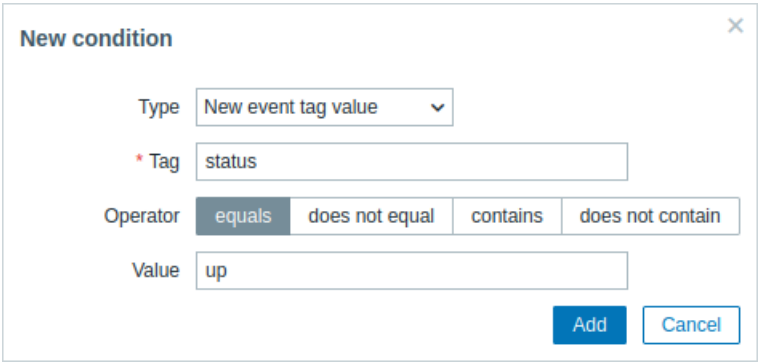


Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge.

Parameter	Description
Name	Nom unique de la règle de corrélation.

Parameter	Description
Type of calculation	Les options suivantes de calcul des conditions sont disponibles : **And** - toutes les conditions doivent être remplies **Or** - il suffit qu'une seule condition soit remplie **And/Or** - AND avec différents types de conditions 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.
Operations	Cochez la case de l'opération à exécuter 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.
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.



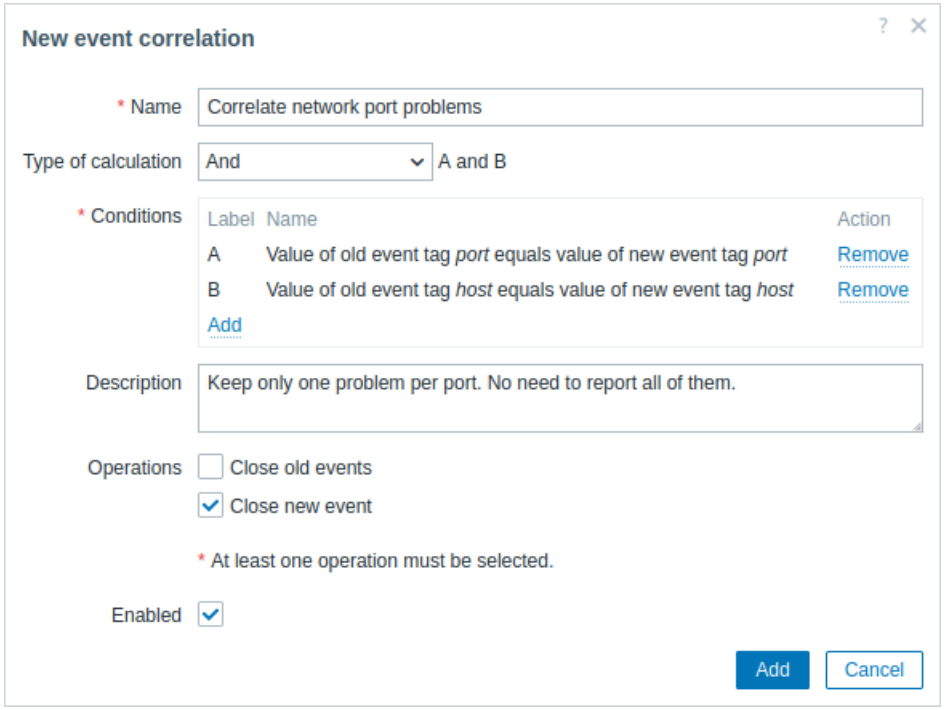
Parameter	Description
New condition	Sélectionnez une condition pour corréler les événements. *Notez* que si aucune condition d'ancien événement n'est spécifiée, tous les anciens événements peuvent être mis en correspondance et fermés. De même, si aucune condition de nouvel événement n'est spécifiée, tous les nouveaux événements peuvent être mis en correspondance et fermés. Les conditions suivantes sont disponibles : **Old event tag name** - spécifiez le nom du tag de l'ancien événement pour la mise en correspondance. **New event tag name** - spécifiez le nom du tag du nouvel événement pour la mise en correspondance. **New event host group** - spécifiez le groupe d'hôtes du nouvel événement pour la mise en correspondance. **Event tag pair** - spécifiez le nom du tag du nouvel événement et le nom du tag de l'ancien événement pour la mise en correspondance. Dans ce cas, il y aura correspondance si les **valeurs** des tags dans les deux événements correspondent. Les *noms* des tags 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 du tag de l'ancien événement pour la mise en correspondance, en utilisant les opérateurs suivants : *equals* - possède la valeur du tag de l'ancien événement *does not equal* - ne possède pas la valeur du tag de l'ancien événement *contains* - contient la chaîne dans la valeur du tag de l'ancien événement *does not contain* - ne contient pas la chaîne dans la valeur du tag de l'ancien événement **New event tag value** - spécifiez le nom et la valeur du tag du nouvel événement pour la mise en correspondance, en utilisant les opérateurs suivants : *equals* - possède la valeur du tag du nouvel événement *does not equal* - ne possède pas la valeur du tag du nouvel événement *contains* - contient la chaîne dans la valeur du tag du nouvel événement *does not contain* - ne contient pas la chaîne dans la valeur du tag du nouvel événement

Warning:
Comme une mauvaise configuration est possible, lorsque des tags d'événement similaires peuvent être créés pour des problèmes **sans rapport**, veuillez examiner les cas décrits ci-dessous !

- Les tags réels et les valeurs de 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, des événements OK ultérieurs avec la même valeur de tag *UNKNOWN* peuvent apparaître et 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 de journal sont longs et que les 255 premiers caractères ne sont pas spécifiques, cela peut également entraîner des tags d'événement similaires pour des problèmes sans rapport.

Exemple

Empêchez les événements de problème répétitifs provenant du même port réseau.



Cette règle globale de corrélation mettra en corrélation les problèmes si les valeurs des tags *host* et *port* existent sur le déclencheur et qu'elles sont identiques dans l'événement d'origine et le nouvel événement.

L'opération fermera les nouveaux événements de problème sur le même port réseau, en ne laissant ouvert que le problème d'origine.

6 Balisage

Aperçu

Les tags se composent d'un nom de tag et d'une valeur de tag. Lors du marquage 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 les tags hérités. Lorsque vous survolez ou cliquez sur un tag *hérité*, une infobulle JavaScript apparaît avec le texte « Inherited tag ». Si un tag hérité existe également comme tag normal, un texte d'infobulle différent est affiché selon la liste où il apparaît (par exemple, « Inherited and template tag » dans la liste des modèles ou « Inherited and host tag » dans la liste des 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'infobulle supplémentaire.

Note:
Consultez les directives officielles de Zabbix pour les recommandations générales sur la définition des tags, ainsi que les recommandations spécifiques pour les modèles, les éléments, les déclencheurs et les règles de découverte de bas niveau.

Les tags ont plusieurs usages, notamment celui de marquer les 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 rétablissement 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 sur la base des tags d'événement.
- les conditions d'action peuvent être configurées sur la base des tags d'événement.
- les problèmes d'élément peuvent être regroupés sur la base 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`). Ces tags ne sont pas considérés comme des doublons.

Cas d'utilisation

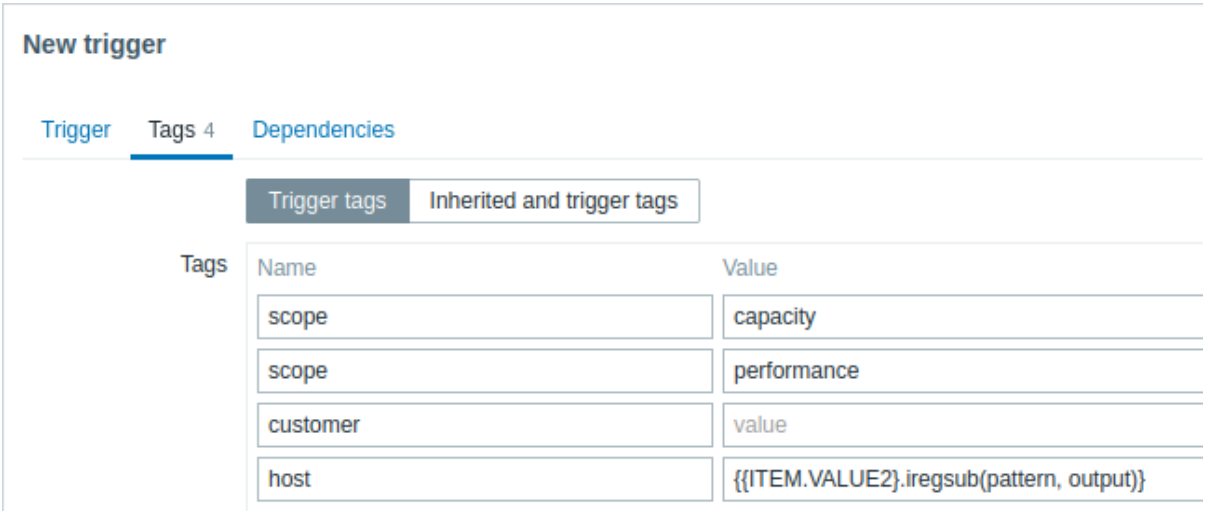
Voici quelques cas d'utilisation courants du marquage par tags :

1. Marquer les événements de déclencheur :
   - Définissez un tag de déclencheur (par exemple, `scope:performance`).
   - Les problèmes créés par ce déclencheur auront le tag de déclencheur.
2. Marquer les problèmes hérités d'un modèle :
   - Définissez un tag de modèle (par exemple, `target:mysql`).
   - Les problèmes créés par les déclencheurs de ce modèle auront le tag de modèle.
3. Marquer les problèmes d'hôte :
   - Définissez un tag d'hôte (par exemple, `service:jira`).
   - Les problèmes créés par les déclencheurs de cet hôte auront le tag d'hôte.
4. Filtrer les éléments de données associés :
   - Définissez un tag d'élément de données (par exemple, `component:cpu`).
   - Dans *Monitoring* > *Latest data*, les éléments de données peuvent être filtrés par le tag `component:cpu`.
5. Utiliser les informations extraites de la valeur de l'élément de données comme valeur du tag :
   - Définissez un tag avec une macro comme valeur du tag (par exemple, `tag-name:{{ITEM.VALUE<N>}.regsub()}`).
   - Dans *Monitoring* > *Problems*, les problèmes auront la valeur du tag résolue en fonction des données extraites de la valeur de l'élément de données.
6. Identifier les problèmes dans un fichier journal et les fermer séparément :
   - Définissez un tag de déclencheur pour le déclencheur de l'élément de surveillance de journal qui extraira des valeurs de la valeur de l'élément de données à l'aide d'une macro (par exemple, `service:{{ITEM.VALUE<N>}.regsub()}`).
   - Dans la configuration du déclencheur, configurez la corrélation d'événements :
     - définissez *PROBLEM event generation mode* sur "Multiple" ;
     - définissez *OK event closes* sur "All problems if tag values match" ;
     - définissez le tag pour la correspondance.
   - Les problèmes créés par le déclencheur de l'élément journal auront le tag de déclencheur et seront fermés individuellement.
7. Filtrer les notifications :
   - Définissez des tags de déclencheur (par exemple, `scope:security` pour trigger1 et `scope:availability` pour trigger2).
   - Utilisez le filtrage par tag dans les conditions d'action pour recevoir des notifications uniquement pour les événements correspondant aux données de tag.
8. Identifier les problèmes dans les notifications :
   - Définissez des tags de déclencheur.
   - Utilisez la macro {EVENT.TAGS} dans la notification de problème.
   - La notification de problème contiendra les tags de déclencheur, ce qui facilitera l'identification de l'application/du service auquel la notification appartient.
9. Simplifier les tâches de configuration à l'aide des tags de modèle :
   - Définissez un tag de déclencheur de modèle.
   - Les déclencheurs créés à partir de ce déclencheur de modèle auront son tag.

10. Créer des déclencheurs avec des tags issus de la découverte de bas niveau (LLD) :
    - Définissez un tag de prototype de déclencheur avec une macro LLD dans le nom ou la valeur du tag (par exemple, `scope:{#FSNAME}`).
    - Les déclencheurs créés à partir du prototype de déclencheur auront son tag.
11. Faire correspondre des services à l'aide de tags de service :
    - Définissez des tags de service.
    - Configurez des actions de service pour les services avec des tags correspondants.
    - En outre, utilisez les tags de service pour lier un service à un SLA pour les calculs de SLA.
12. Lier des services à des problèmes à l'aide de tags de problème de service :
    - Définissez un tag de problème dans la configuration du service (par exemple, `target:mysql`).
    - Les problèmes avec un tag correspondant seront automatiquement corrélés au service, et l'état du service changera en fonction des règles configurées de calcul de l'état du service.
13. Supprimer les problèmes lorsqu'un hôte est en mode maintenance :
    - Définissez des tags dans la configuration de la période de maintenance.
    - Les problèmes avec les tags définis seront supprimés.
14. Accorder l'accès à des groupes d'utilisateurs :
    - Définissez des tags dans la configuration du groupe d'utilisateurs.
    - Les utilisateurs du groupe d'utilisateurs pourront voir uniquement les problèmes avec les tags définis.

Configuration

Les tags peuvent être définis dans un onglet dédié, par exemple, dans la configuration de déclencheur :



Prise en charge des macros

Les macros intégrées et les macros utilisateur dans les tags sont résolues au moment de l'événement. Jusqu'à ce que l'événement se produise, ces macros seront affichées non résolues dans le frontend Zabbix.

Les macros de découverte de bas niveau sont résolues pendant le processus de découverte.

Les macros suivantes peuvent être utilisées dans les noms et les valeurs des tags de déclencheur :

- macros intégrées {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.*} (pour référencer les valeurs d'inventaire d'hôte d'un ou de 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 prototypes de déclencheur)

Les macros suivantes peuvent être utilisées dans les noms et les valeurs des tags de modèle, d'hôte et d'élément/scénario web :

- macros intégrées {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} et {HOST.ID}
- macros intégrées {INVENTORY.*}
- macros utilisateur
- macros de découverte de bas niveau (uniquement dans les tags de prototypes d'hôte et d'élément)

Les macros suivantes peuvent être utilisées dans les notifications basées sur des déclencheurs :

- macros intégrées {EVENT.TAGS} et {EVENT.RECOVERY.TAGS} (ces macros seront résolues en une liste de tags d'événement ou de tags d'événement de récupération séparés par des virgules)

- macros intégrées {EVENT.TAGSJSON} et {EVENT.RECOVERY.TAGSJSON} (ces macros seront résolues en un tableau JSON contenant des objets de tag d'événement 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 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 la résolution de la macro, il sera tronqué à 255 caractères.

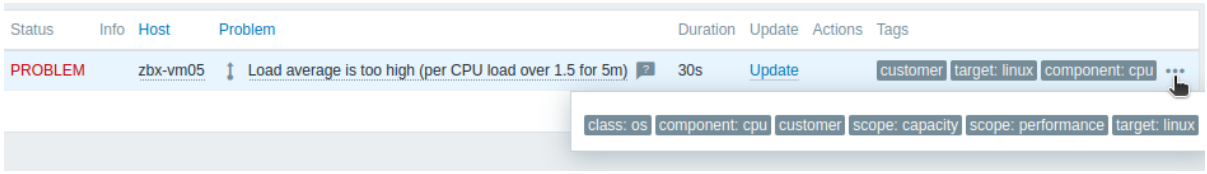
Voir aussi : utilisation des fonctions de macro dans les macros de découverte de bas niveau pour le marquage des événements.

Affichage des tags d'événement

Les tags, s'ils sont définis, peuvent être consultés avec les nouveaux événements dans :

- *Monitoring* > *Problems*
- *Monitoring* > *Problems* > *Event details*
- *Dashboards* > *Problems* widget

L'ordre et le nombre de tags affichés sont déterminés par les options de filtrage *Tag display priority* et *Show tags* dans *Monitoring* > *Problems* ou dans le widget de tableau de bord *Problems*. Notez qu'un maximum de trois tags peut être affiché ; s'il y a davantage de tags, le survol des trois points permet d'afficher tous les tags dans une fenêtre contextuelle.



7 Visualisation

Veuillez utiliser la barre latérale pour accéder au contenu de la section Visualisation.

1 Graphiques

Aperçu

Avec une grande quantité de données entrant dans Zabbix, il devient beaucoup plus facile pour les utilisateurs de consulter une représentation visuelle de ce qui se passe plutôt que de se limiter à des chiffres.

C'est là qu'interviennent les graphiques. Les graphiques permettent de comprendre le flux de données d'un seul coup d'œil, de corréler les problèmes, de découvrir quand quelque chose a commencé ou de présenter à quel moment quelque chose pourrait devenir un problème.

Zabbix fournit aux utilisateurs :

- des graphiques simples intégrés pour les données d'un seul 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 et des camemberts

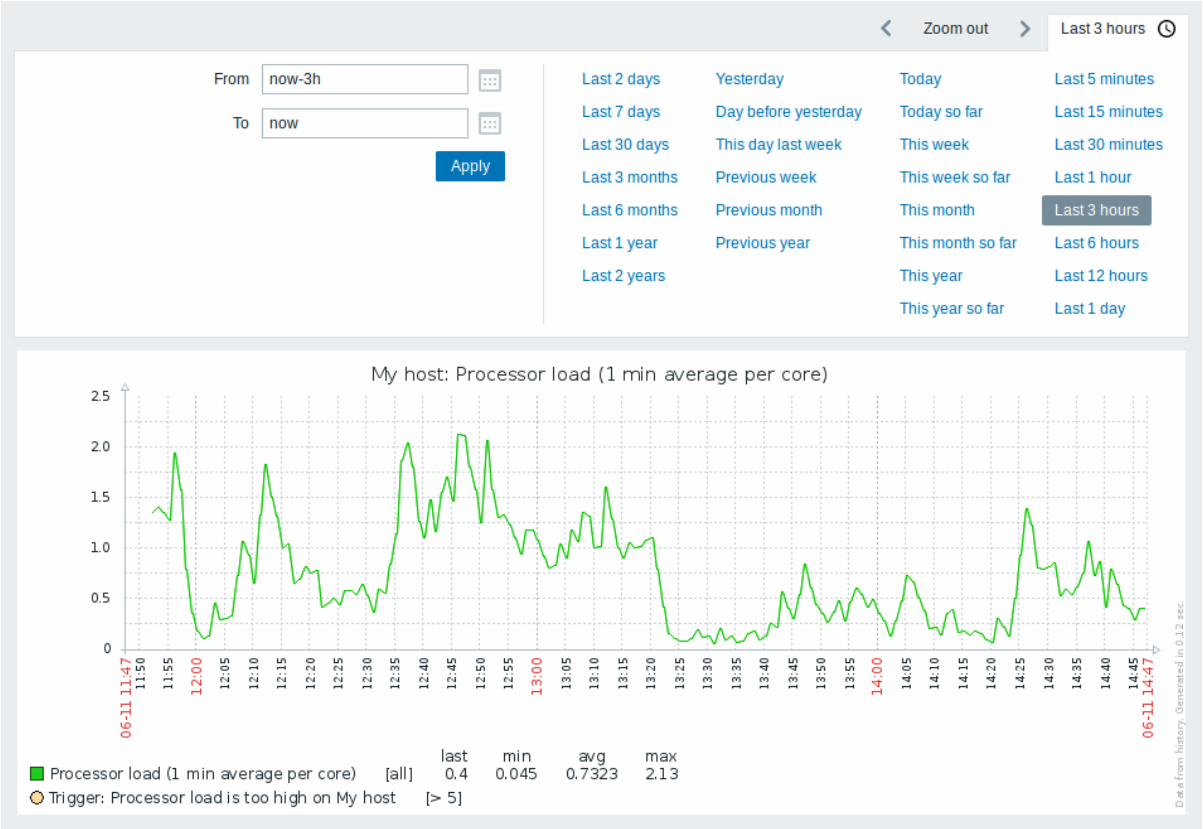
1 Graphiques simples

Aperçu

Des graphiques simples sont fournis pour visualiser les données collectées par les éléments.

Aucune configuration n'est requise de la part de l'utilisateur pour afficher les graphiques simples. Ils sont mis à disposition automatiquement par Zabbix.

Il suffit d'aller dans *Monitoring* > *Latest data* et de cliquer sur le lien Graph de l'élément concerné pour qu'un graphique s'affiche.



Note:
Des graphiques simples sont fournis pour tous les éléments numériques. Pour les éléments textuels, un lien vers l'historique est disponible dans *Monitoring* > *Latest data*.

Sélecteur de période

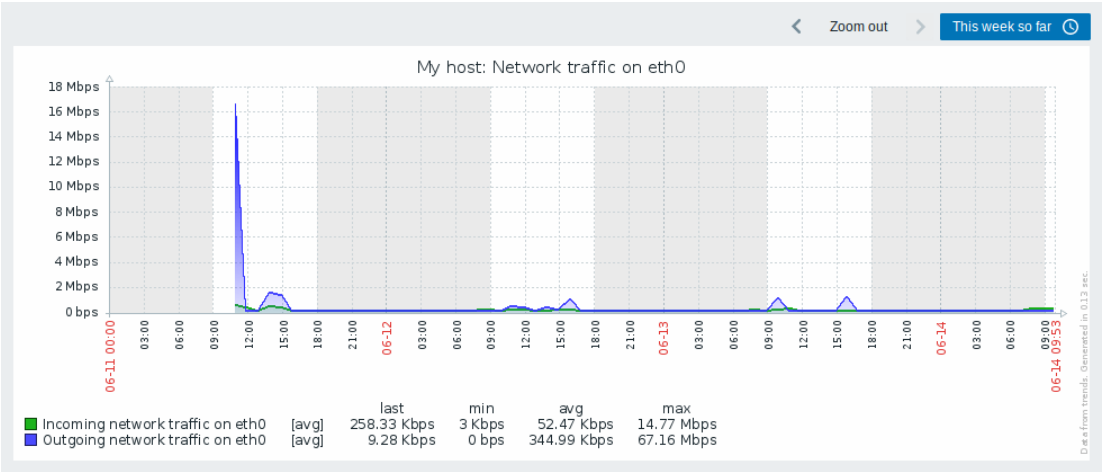
Le sélecteur de période au-dessus du graphique permet de sélectionner les périodes fréquemment requises en un seul clic de souris. Pour plus d'informations, voir Sélecteurs de période et d'hôte.

Données récentes vs périodes plus longues

Pour les données très récentes, une **seule** ligne est tracée, 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 maximales et minimales 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 travaillé est affiché en gris (avec le thème frontal par défaut *Original blue*).

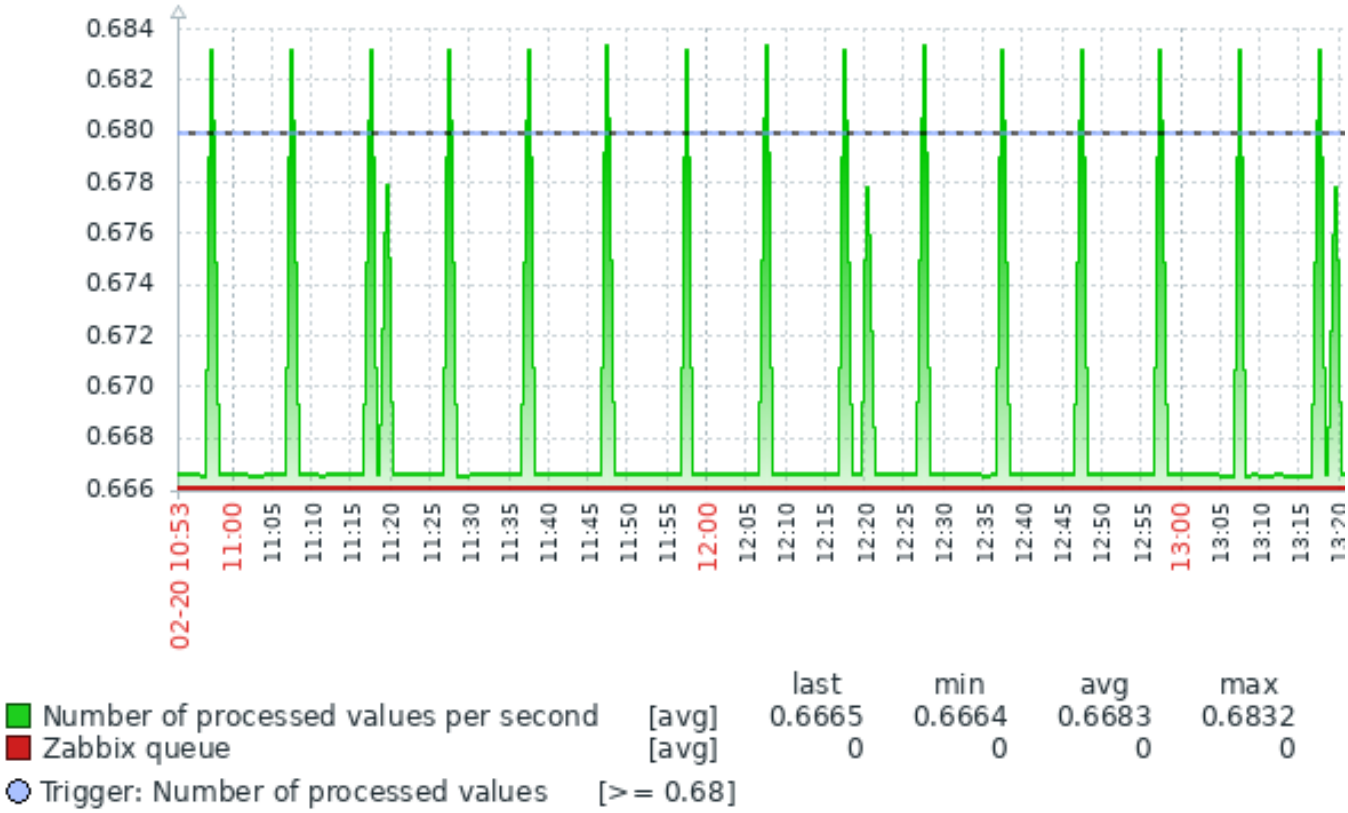


Le temps de travail est toujours affiché dans les graphiques simples, tandis que son affichage dans les graphiques personnalisés dépend des préférences de l'utilisateur.

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 avec des tirets 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 ayant activé le mode de débogage de l'interface web, une légende verticale grise s'affiche en bas à droite d'un graphique pour indiquer d'où proviennent les données.

Plusieurs facteurs influencent l'utilisation de l'historique ou des tendances :

- 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 antérieures à ces quatorze jours proviendront 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 encore disponible pour la même période).
- si les tendances sont désactivées, l'historique de l'élément est utilisé pour construire 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 en haut à droite 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 représentées ici sont brutes, c'est-à-dire qu'aucune unité ni aucun post-traitement des valeurs n'est utilisé. Le mappage des valeurs, en revanche, est appliqué.

Problèmes connus

Voir les problèmes connus pour les graphiques.

2 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.

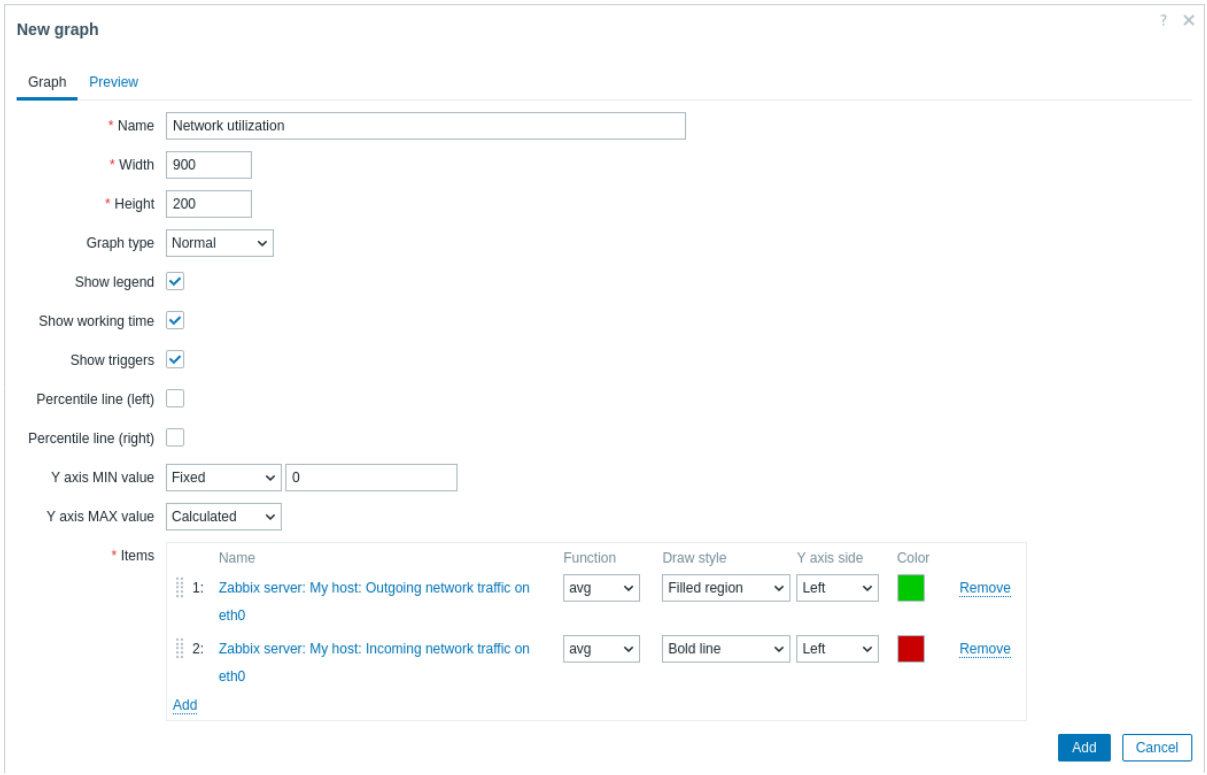
Les graphiques personnalisés sont configurés manuellement.

Ils peuvent être créés pour un hôte, plusieurs hôtes ou un seul modèle.

Configuration de graphiques personnalisés

Pour créer un graphique personnalisé, procédez comme suit :

- Allez dans *Collecte de données > Hôtes (ou Modèles)*
- Cliquez sur *Graphiques* sur la ligne correspondant à l'hôte ou au modèle souhaité
- Dans l'écran Graphiques, cliquez sur *Créer un graphique*
- Modifiez les attributs du graphique



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

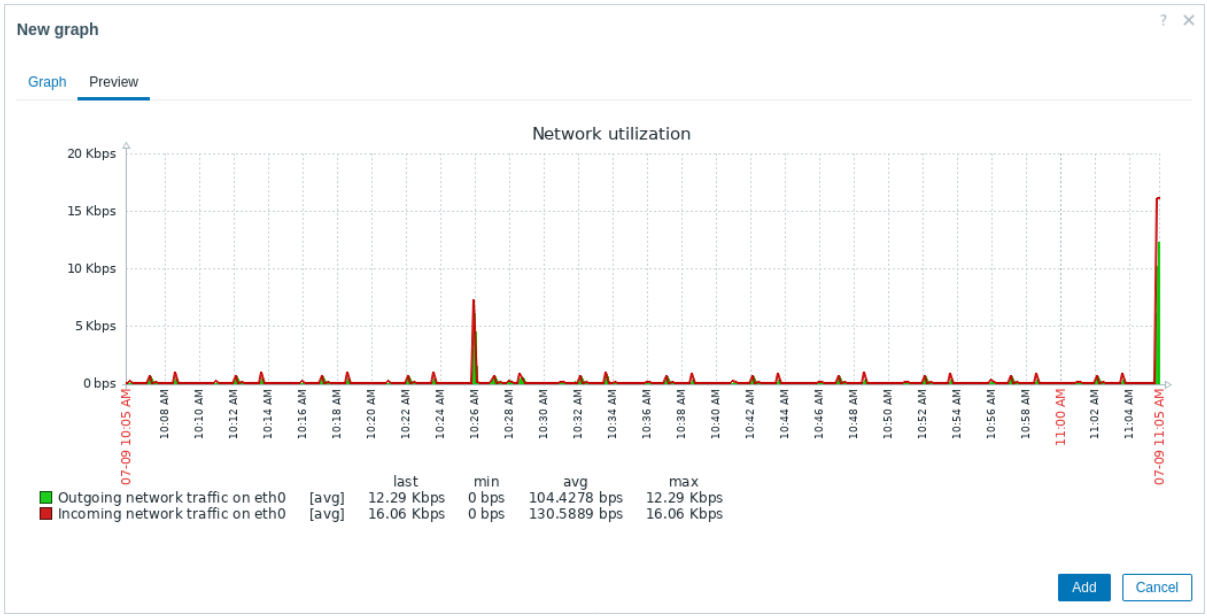
Attributs du graphique :

Paramètre	Description
Nom	Nom unique du graphique. 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)}`). 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 graphique, par exemple `{?avg(/{HOST.HOST2}/key,1h)}`. Notez que faire référence au premier hôte avec cette macro est redondant, car le premier hôte peut être référencé implicitement, par exemple `{?avg(//key,1h)}`.
Largeur	Largeur du graphique en pixels (pour l'aperçu et les graphiques en secteurs/secteurs éclatés uniquement).
Hauteur	Hauteur du graphique en pixels.
Type de graphique	Type de graphique : **Normal** - graphique normal, les valeurs sont affichées sous forme de lignes **Empilé** - graphique empilé, les zones remplies sont affichées **Secteurs** - graphique en secteurs **Secteurs éclatés** - graphique en secteurs « éclatés », les portions sont affichées comme « détachées » du secteur
Afficher la légende	En cochant cette case, la légende du graphique sera affichée.
Afficher le temps de travail	Si cette option est sélectionnée, les heures non ouvrées seront affichées avec un arrière-plan gris. Ce paramètre n'est pas disponible pour les graphiques en secteurs et en secteurs éclatés.
Afficher les déclencheurs	Si cette option est sélectionnée, les déclencheurs simples seront affichés sous forme de lignes en pointillés noirs sur la couleur de gravité du déclencheur. Ce paramètre n'est pas disponible pour les graphiques en secteurs et en secteurs éclatés.
Ligne de percentile (gauche)	Affiche le percentile pour l'axe Y gauche. Si, par exemple, le 95e percentile est défini, la ligne de percentile se situera au niveau sous lequel tombent 95 pour cent des valeurs. Elle est affichée sous forme d'une ligne vert clair. Disponible uniquement pour les graphiques normaux.
Ligne de percentile (droite)	Affiche le percentile pour l'axe Y droit. Si, par exemple, le 95e percentile est défini, la ligne de percentile se situera au niveau sous lequel tombent 95 pour cent des valeurs. Elle est affichée sous forme d'une ligne rouge vif. Disponible uniquement pour les graphiques 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 graphiques en secteurs et en secteurs éclatés.
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 graphiques en secteurs et en secteurs éclatés.
Vue 3D	Active le style 3D. Pour les graphiques en secteurs et en secteurs éclatés uniquement.
Éléments	Éléments dont les données doivent être affichées dans ce graphique. 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).

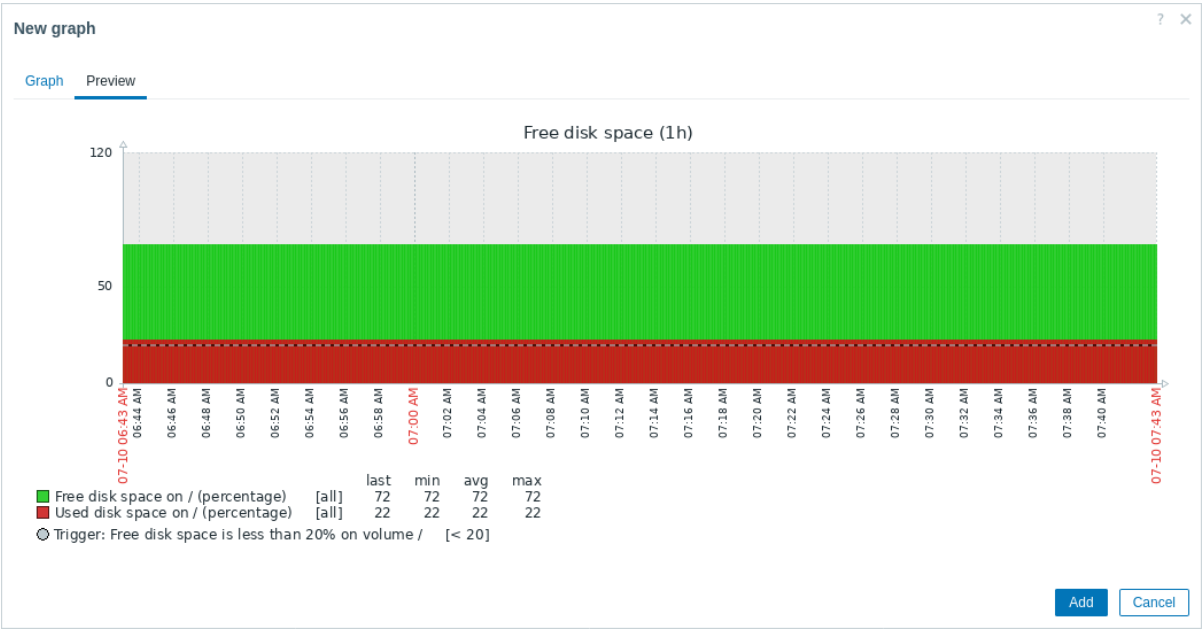
Paramètre	Description
Ordre de tri (0→100)	Ordre de tracé. 0 sera traité en premier. Peut être utilisé pour dessiner des lignes ou des zones derrière (ou devant) une autre. Vous pouvez faire glisser et déposer des éléments à l'aide de l'icône au début d'une ligne pour définir l'ordre de tri ou déterminer quel élément est 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 graphiques en secteurs et en secteurs éclatés) : **Simple** - la valeur de l'élément est représentée proportionnellement dans le secteur **Somme du graphique** - la valeur de l'élément représente l'ensemble du secteur Notez que la couleur de l'élément « somme du graphique » ne sera visible que dans la mesure où elle n'est pas occupée par des éléments « proportionnels ».
Fonction	Sélectionnez quelles valeurs seront affichées lorsqu'il existe plus d'une valeur par pixel vertical du graphique pour un élément : **all** - affiche toutes les valeurs possibles (minimum, maximum, moyenne) dans le graphique. Notez que pour des périodes plus courtes, ce paramètre n'a aucun effet ; ce n'est que pour des périodes plus longues, lorsque la densité des données dans un pixel vertical du graphique augmente, que « all » commence à afficher les valeurs minimale, maximale et moyenne. Cette fonction est disponible uniquement pour le type de graphique *Normal*. Voir aussi : Génération de graphiques à partir de l'historique/des tendances. **avg** - affiche les valeurs moyennes **last** - affiche les valeurs les plus récentes. Cette fonction est disponible uniquement si *Secteurs/Secteurs éclatés* est sélectionné comme type de graphique. **max** - affiche les valeurs maximales **min** - affiche les valeurs minimales
Style de tracé	Sélectionnez le style de tracé (disponible uniquement pour les graphiques normaux ; pour les graphiques empilés, une zone remplie est toujours utilisée) à appliquer aux données de l'élément - *Ligne*, *Ligne épaisse*, *Zone remplie*, *Point*, *Ligne en pointillés*, *Ligne en dégradé*.
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.

3 Graphiques ad-hoc

Vue d'ensemble

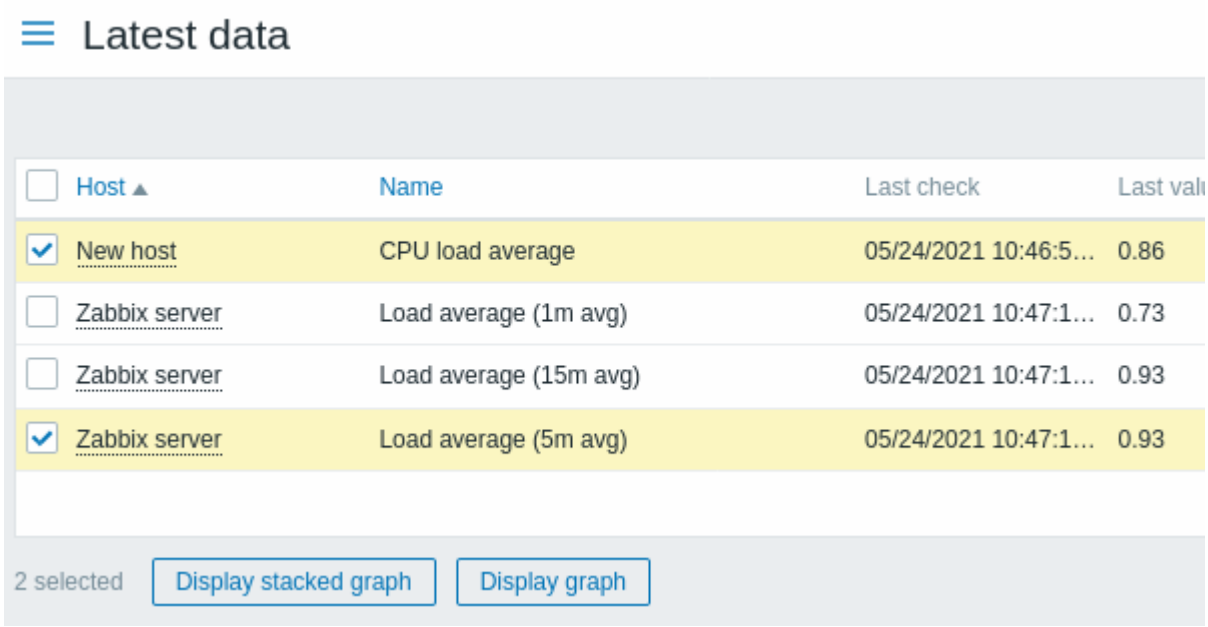
Bien qu'un graphique simple soit idéal pour accéder aux données d'un élément et que les graphiques personnalisés offrent des options de personnalisation, aucun des deux ne permet de créer rapidement un graphique de comparaison pour plusieurs éléments avec peu d'effort et sans maintenance.

Pour résoudre ce problème, il est possible de créer des graphiques ad hoc pour plusieurs éléments de manière très rapide.

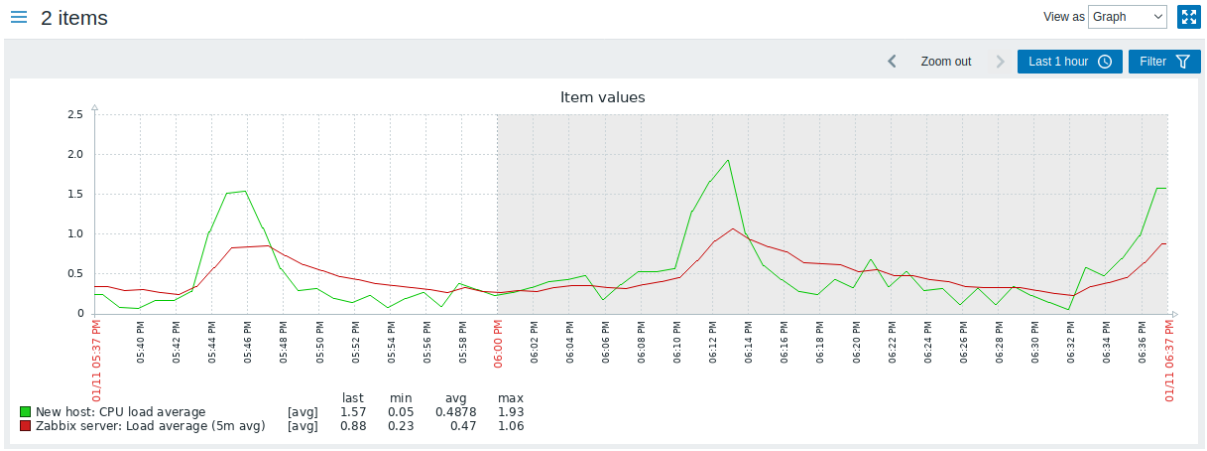
Configuration

Pour créer un graphique ad hoc, procédez comme suit :

- Allez dans *Surveillance → Dernières données*
- Utilisez le filtre pour afficher les éléments souhaités
- Cochez les cases des éléments que vous souhaitez représenter dans le graphique
- Cliquez sur les boutons *Afficher le graphique empilé* ou *Afficher le graphique*

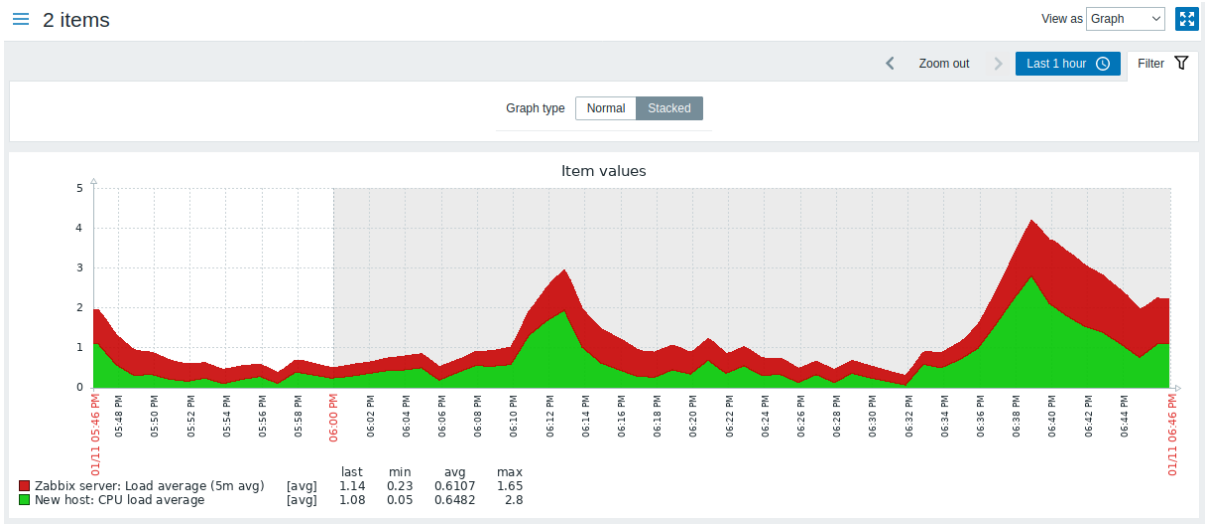


Votre graphique est créé instantanément :



Notez que, pour éviter d'afficher trop de lignes dans le graphique, seule la valeur moyenne de chaque élément est affichée (les lignes des valeurs min/max ne sont pas affichées). Les déclencheurs et les informations sur les déclencheurs ne sont pas affichés dans le graphique.

Dans la fenêtre du graphique créé, le sélecteur de période est disponible, ainsi que la possibilité de passer du graphique en lignes « normal » à un graphique empilé (et inversement).



4 Agrégation dans les graphiques

Aperçu

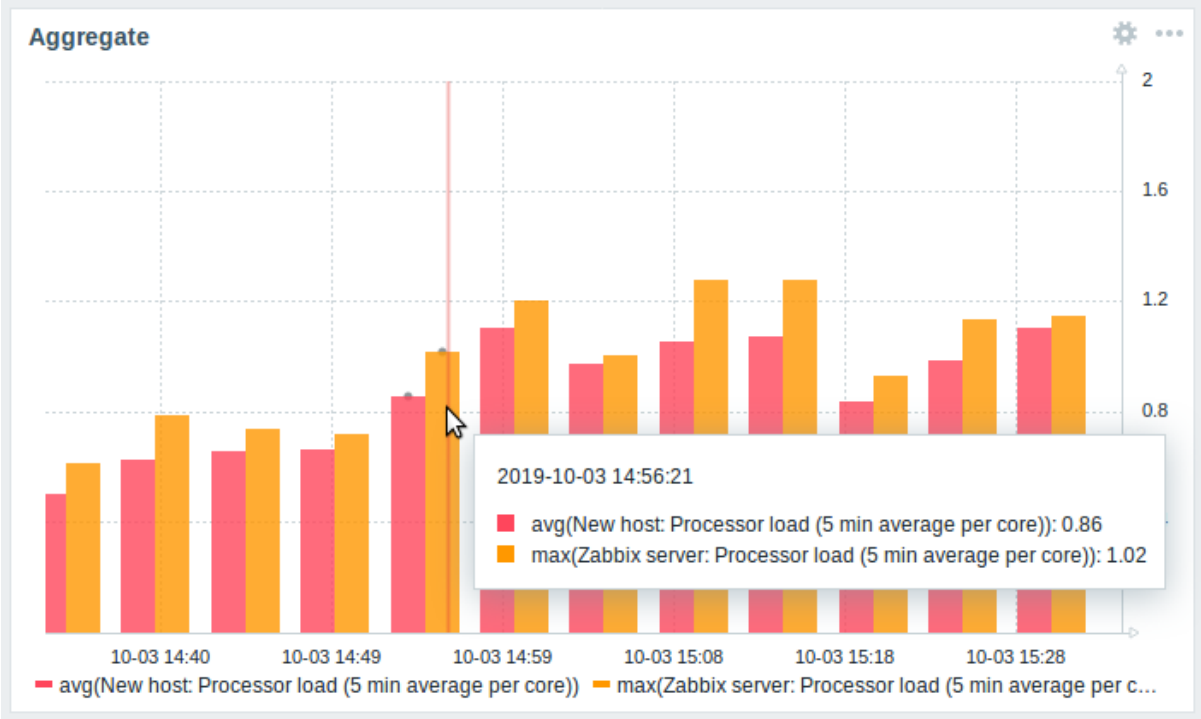
Les fonctions d'agrégation, disponibles dans les widgets de graphique et de diagramme circulaire du tableau de bord, permettent d'afficher une valeur agrégée pour l'intervalle choisi (5 minutes, une heure, une journée), au lieu de toutes les valeurs.

Cette section fournit plus de détails sur les options d'agrégation dans le widget de graphique.

Les options d'agrégation sont les suivantes :

- min
- max
- avg
- count
- sum
- first (première valeur affichée)
- last (dernière valeur affichée)

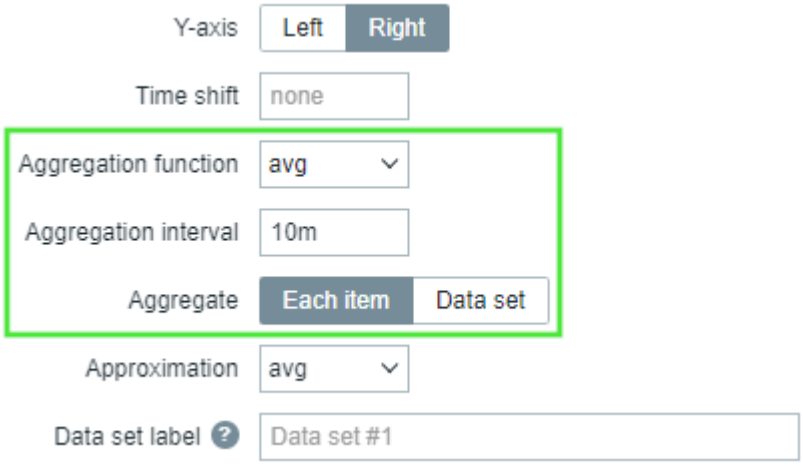
L'utilisation la plus intéressante de l'agrégation des données est la possibilité de créer de belles comparaisons côte à côte des données sur une certaine période :



Lorsque vous survolez un point dans le temps sur le graphique, la date et l'heure s'affichent en plus des éléments et de leurs valeurs agrégées. Les éléments sont affichés entre parenthèses, précédés de la fonction d'agrégation utilisée. Si le widget de graphique a un *Libellé de jeu de données* configuré, le libellé est affiché entre parenthèses, précédé de la fonction d'agrégation utilisée. Notez qu'il s'agit de la date et de l'heure du point sur le graphique, et non de celles des valeurs réelles.

Configuration

Les options d'agrégation sont disponibles dans les paramètres de l'ensemble de données lors de la configuration d'un widget de graphique.



Vous pouvez choisir la fonction d'agrégation et l'intervalle de temps. Comme l'ensemble de données peut comprendre plusieurs éléments, une autre option permet également d'afficher les données agrégées pour chaque élément séparément ou pour tous les éléments de l'ensemble de données sous la forme d'une seule valeur agrégée.

Cas d'usage

Nombre moyen de requêtes vers le serveur Nginx

Afficher le nombre moyen de requêtes par seconde et par jour sur le serveur Nginx :

- ajouter le nombre de requêtes par seconde à l'ensemble de données
- sélectionner la fonction d'agrégation `avg` et spécifier l'intervalle `1d`
- un graphique à barres s'affiche, où chaque barre représente le nombre moyen de requêtes par seconde et par jour

Espace disque hebdomadaire minimal entre les clusters

Affichez l'espace disque le plus faible parmi les clusters sur une semaine.

- ajoutez à l'ensemble de données : hôtes `cluster*`, clé "Espace disque libre sur /data"
- sélectionnez la fonction d'agrégation `min` et spécifiez l'intervalle `1w`
- un graphique à barres s'affiche, où chaque barre représente l'espace disque minimum par semaine pour chaque volume /data du cluster

2 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.

1 Configuration d'une carte réseau

Aperçu

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 proprement dite avec des éléments et leurs liens.

Vous pouvez remplir la carte avec des éléments tels qu'un hôte, un groupe d'hôtes, un déclencheur, une image ou une autre carte.

Des icônes sont utilisées pour représenter les éléments de la carte. Vous pouvez définir les informations qui seront affichées avec les icônes et indiquer que les problèmes récents doivent être affichés d'une manière particulière. 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 dans 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 tirer parti 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.

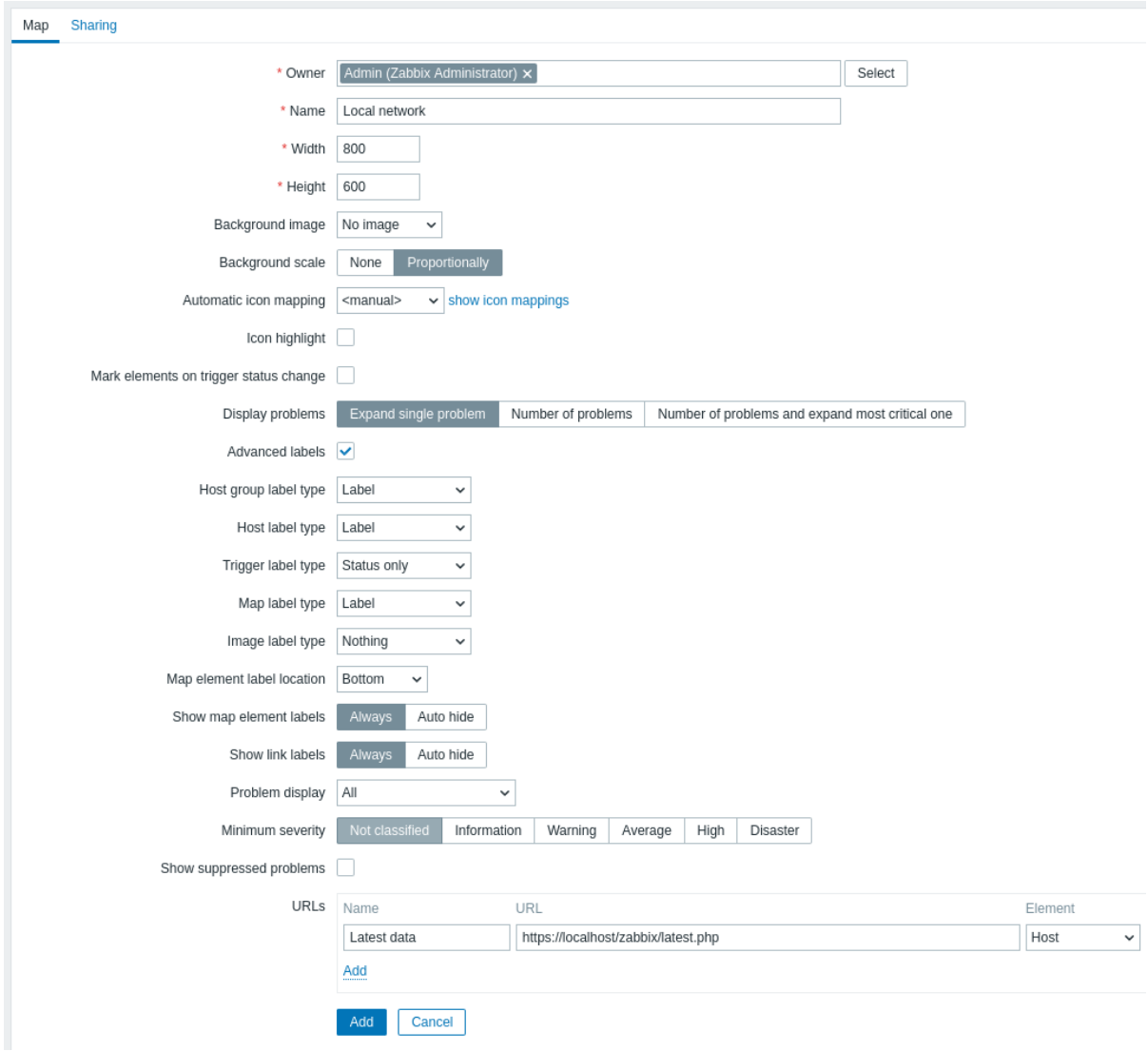
Création d'une carte

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*.

Vous pouvez également utiliser le bouton *Clone* dans le formulaire de configuration d'une carte existante pour créer une nouvelle carte. Cette carte aura toutes les propriétés de la carte existante, y compris 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 :

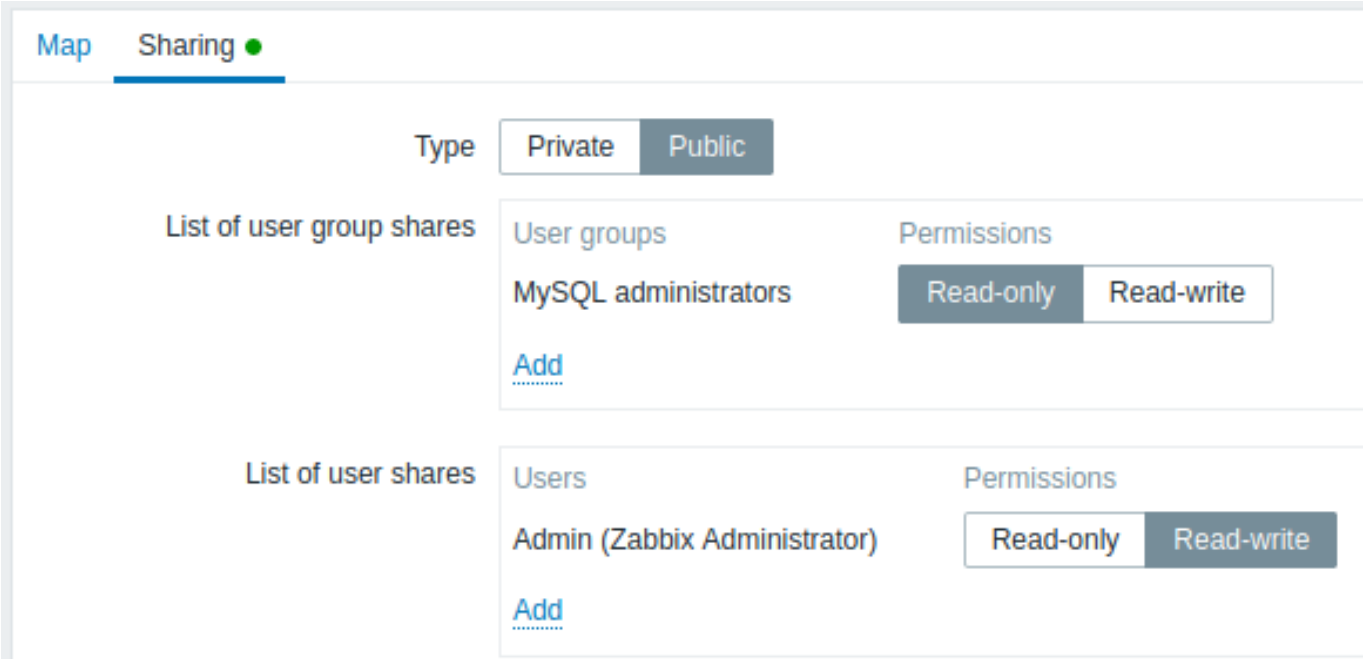
Parameter	Description
Owner	Nom du propriétaire de la carte.
Name	Nom unique de la carte.
Width	Largeur de la carte en pixels.
Height	Hauteur de la carte en pixels.
Background image	Utiliser une image d'arrière-plan : **No image** - aucune image d'arrière-plan (fond blanc) **Image** - l'image sélectionnée sera utilisée 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** - couvre tout l'arrière-plan de la carte, tout en conservant les proportions de l'image (par défaut).
Automatic icon mapping	Vous pouvez définir l'utilisation d'un mappage automatique des icônes, configuré dans *Administration > General > Icon mapping*. Le mappage 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 surbrillance. Les éléments avec un trigger actif recevront un arrière-plan rond, de la même couleur que le trigger de gravité la plus élevée. De plus, une ligne verte épaisse sera affichée autour du cercle si tous les problèmes sont acquittés. Les éléments ayant le statut « disabled » ou « in maintenance » auront respectivement un arrière-plan carré gris et orange. Voir aussi : Viewing maps

Parameter	Description
Mark elements on trigger status change	Une modification récente de l'état du trigger (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 sont libres de l'étiquette. Les marqueurs sont affichés pendant 30 minutes.
Display problems	Sélectionnez comment les problèmes sont affichés avec un élément de carte : **Expand single problem** - s'il n'y a qu'un seul problème, le nom du problème 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 « plus critique » est déterminé en fonction de la gravité du problème et, en cas d'égalité, de l'ID d'événement du problème (l'ID le plus élevé ou le problème le plus récent est affiché en premier). Pour un *trigger map element*, cela est basé sur la gravité du problème et, en cas d'égalité, sur la position du trigger dans la liste des triggers. En cas de plusieurs problèmes pour le même trigger, le plus récent sera affiché.
Advanced labels	Si vous cochez cette case, vous pourrez définir des types d'étiquettes distincts pour différents types d'éléments.
Host group label type	Type d'étiquette utilisé pour l'élément de carte : **Label** - étiquette de l'élément de carte **IP address** - adresse IP **Element name** - nom de l'élément (par exemple, nom de l'hôte) **Status only** - statut uniquement (OK ou PROBLEM) **Nothing** - aucune étiquette n'est affichée
Host label type *Trigger label type* *Map label type* *Image label type*	
Map element label location	Emplacement de l'étiquette 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 comment afficher les étiquettes des éléments de carte : **Always** - toujours affichées (par défaut) ; **Auto hide** - masquer l'étiquette si elle n'est pas survolée ou sélectionnée.
Show link labels	Sélectionnez comment afficher les étiquettes des liens : **Always** - toujours affichées (par défaut) ; **Auto hide** - masquer l'étiquette si elle n'est pas survolée ou sélectionnée.
Problem display	Afficher le nombre de problèmes comme suit : **All** - le nombre total de problèmes sera affiché **Separated** - le nombre de problèmes non acquittés sera affiché séparément du nombre total de problèmes **Unacknowledged only** - seul le nombre de problèmes non acquittés sera affiché
Minimum severity	Les problèmes inférieurs au niveau de gravité minimale sélectionné ne seront pas affichés sur la carte. Par exemple, si *Warning* est sélectionné, les changements avec des triggers de niveau *Information* et *Not classified* ne seront pas reflétés 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	Des URL pour chaque type d'élément peuvent être définies ici, où : **Name** - une étiquette pour l'URL ; **URL** - une URL (jusqu'à 2048 caractères) ; **Element** - le type d'élément (par ex., *Host*, *Host group*, etc.). Celles-ci seront affichées sous forme de liens lorsqu'un utilisateur clique 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 une liste complète, consultez supported macros 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 :



Paramètre	Description
Type	Sélectionnez le type de carte : **Privée** - la carte est visible uniquement par les groupes d'utilisateurs et les utilisateurs sélectionnés **Publique** - la carte est visible par tous
Liste des partages de groupes d'utilisateurs	Sélectionnez les groupes d'utilisateurs auxquels la carte est accessible. Vous pouvez autoriser un accès en lecture seule ou en lecture-écriture.
Liste des partages d'utilisateurs	Sélectionnez les utilisateurs auxquels la carte est accessible. Vous pouvez autoriser un accès en lecture seule ou en lecture-écriture.

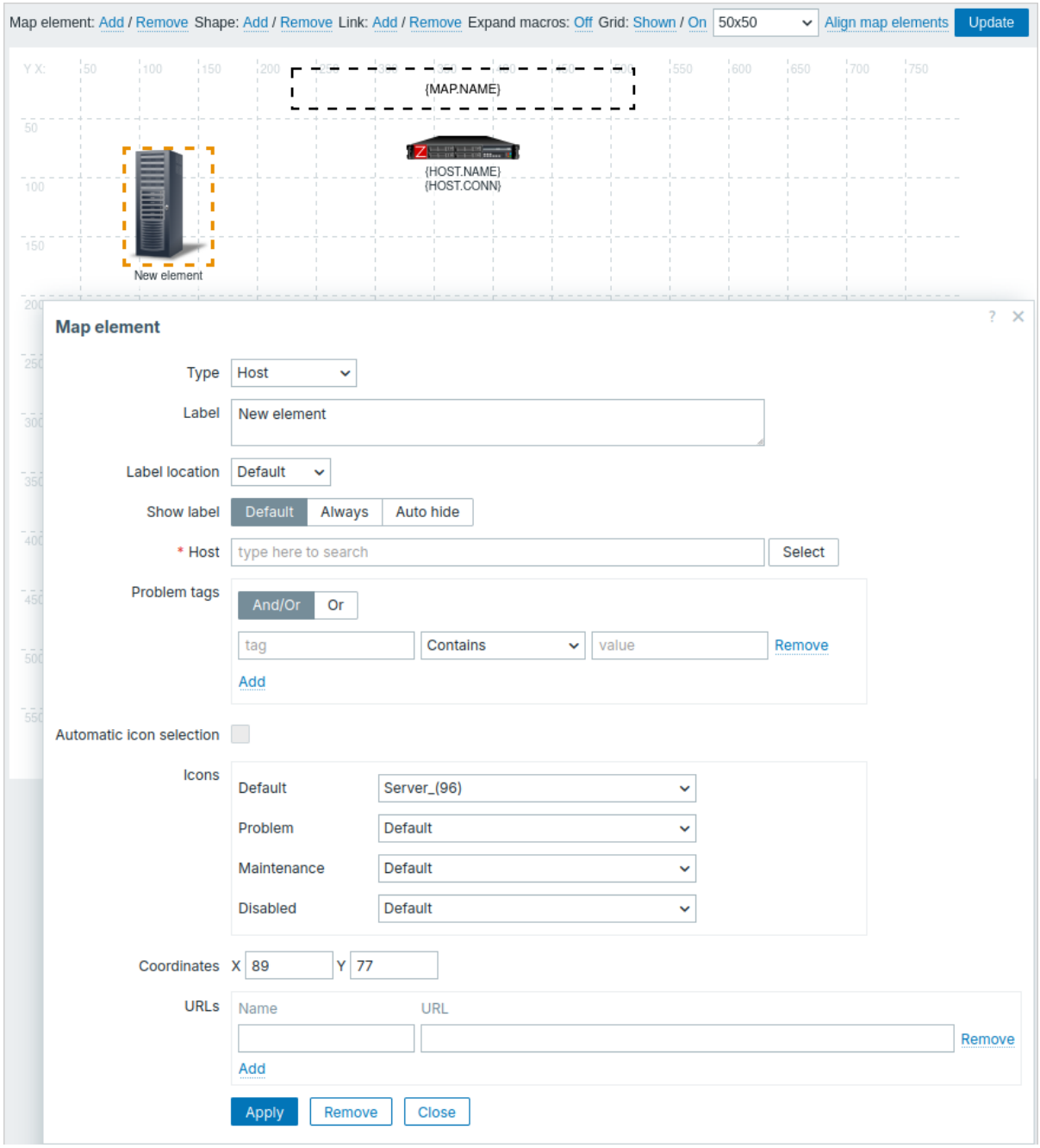
Lorsque vous cliquez sur *Ajouter* pour enregistrer cette carte, vous créez une carte vide avec un nom, des dimensions et certaines préférences. Vous devez maintenant ajouter des éléments. Pour cela, cliquez sur *Modifier* dans la liste des cartes pour ouvrir la zone modifiable.

Ajout d'éléments

Pour ajouter un élément, cliquez sur *Ajouter* à côté de `Map element`. Le nouvel élément apparaîtra dans le coin supérieur gauche de la carte. Faites-le glisser-déposer où vous le souhaitez.

Notez qu'avec l'option Grille définie sur "On", les éléments s'aligneront toujours sur la grille (vous pouvez choisir différentes tailles de grille dans la liste déroulante, ainsi que masquer/afficher la grille). Si vous souhaitez placer les éléments n'importe où sans alignement, définissez l'option sur "Off". (Vous pourrez aligner ultérieurement des éléments aléatoires 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 autre icône, 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, chaîne quelconque. 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 ».

Parameter	Description
Label location	Emplacement du libellé par rapport à l'icône : **Default** - emplacement de libellé par défaut de la carte ; **Bottom** - sous l'icône ; **Left** - à gauche ; **Right** - à droite ; **Top** - au-dessus de l'icône.
Show label	Sélectionnez comment afficher le libellé de l'élément : **Default** - utiliser les paramètres *Show map element labels* de la configuration de la carte (par défaut) ; **Always** - toujours affiché (par défaut) ; **Auto hide** - masquer le libellé s'il n'est pas survolé ou sélectionné.
Host	Sélectionnez un hôte ; vous pouvez également saisir le nom de l'hôte. Ce champ prend en charge l'auto-complétion ; ainsi, lorsque vous commencez à saisir le nom d'un hôte, une liste déroulante des hôtes correspondants s'affiche. Faites défiler vers le bas pour en sélectionner un. Cliquez sur l'icône de suppression (x) pour supprimer la sélection. Ce champ n'est disponible que pour les éléments de type *Host*.
Map	Sélectionnez une carte ; vous pouvez également saisir le nom de la carte. Ce champ prend en charge l'auto-complétion ; ainsi, lorsque vous commencez à saisir le nom d'une carte, une liste déroulante des cartes correspondantes s'affiche. Faites défiler vers le bas pour en sélectionner une. Cliquez sur l'icône de suppression (x) pour supprimer la sélection. Ce champ n'est disponible que pour les éléments de type *Map*.
Triggers	Sélectionnez un ou plusieurs déclencheurs dans le champ *New triggers* ci-dessous et cliquez sur *Add*. Vous pouvez également 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 supprimer la sélection. Ce champ n'est disponible que pour les éléments de type *Trigger*. L'ordre des déclencheurs sélectionnés peut être modifié, mais uniquement au sein du même niveau de sévérité des déclencheurs. La sélection de plusieurs déclencheurs affecte également la résolution de la macro {HOST.*} en mode édition comme en mode affichage. En mode édition, les premières macros {HOST.*} affichées seront résolues en fonction du premier déclencheur de la liste (sur la base de la sévérité 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 (sans tenir compte de la sévérité) 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 sévérité du déclencheur.
Host group	Sélectionnez un groupe d'hôtes ; vous pouvez également saisir le groupe d'hôtes. Ce champ prend en charge l'auto-complétion ; ainsi, lorsque vous commencez à saisir le nom d'un groupe d'hôtes, une liste déroulante des groupes d'hôtes correspondants s'affiche. Faites défiler vers le bas pour en sélectionner un. Cliquez sur l'icône de suppression (x) pour supprimer la sélection. Ce champ n'est disponible que pour les éléments de type *Host group*.
Problem tags	Spécifiez des tags pour limiter le nombre de problèmes 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 : **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 Ce champ n'est disponible que pour les éléments de type *Host* et *Host group*.

Parameter	Description
Automatic icon selection	Lorsque la case est cochée, le mapping d'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 n'est disponible que 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 clique sur l'élément en mode d'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. Des 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 ».

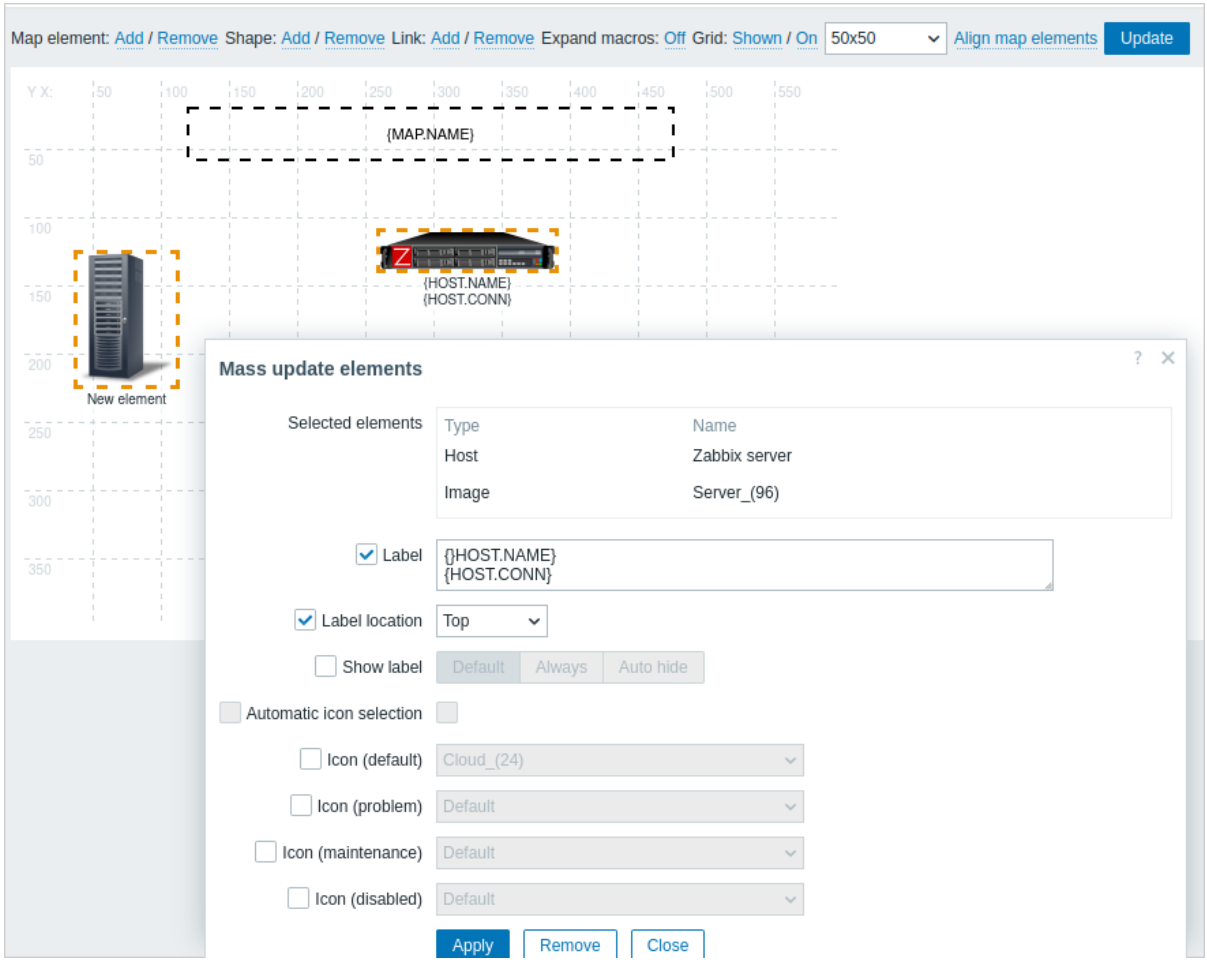
Attention:
Les éléments ajoutés ne sont pas enregistrés automatiquement. Si vous quittez la page, toutes les modifications risquent d'être perdues.
Il est donc recommandé de cliquer sur le bouton **Update** dans le coin supérieur droit. Une fois cliqué, les modifications sont enregistrées quel que soit votre choix 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.

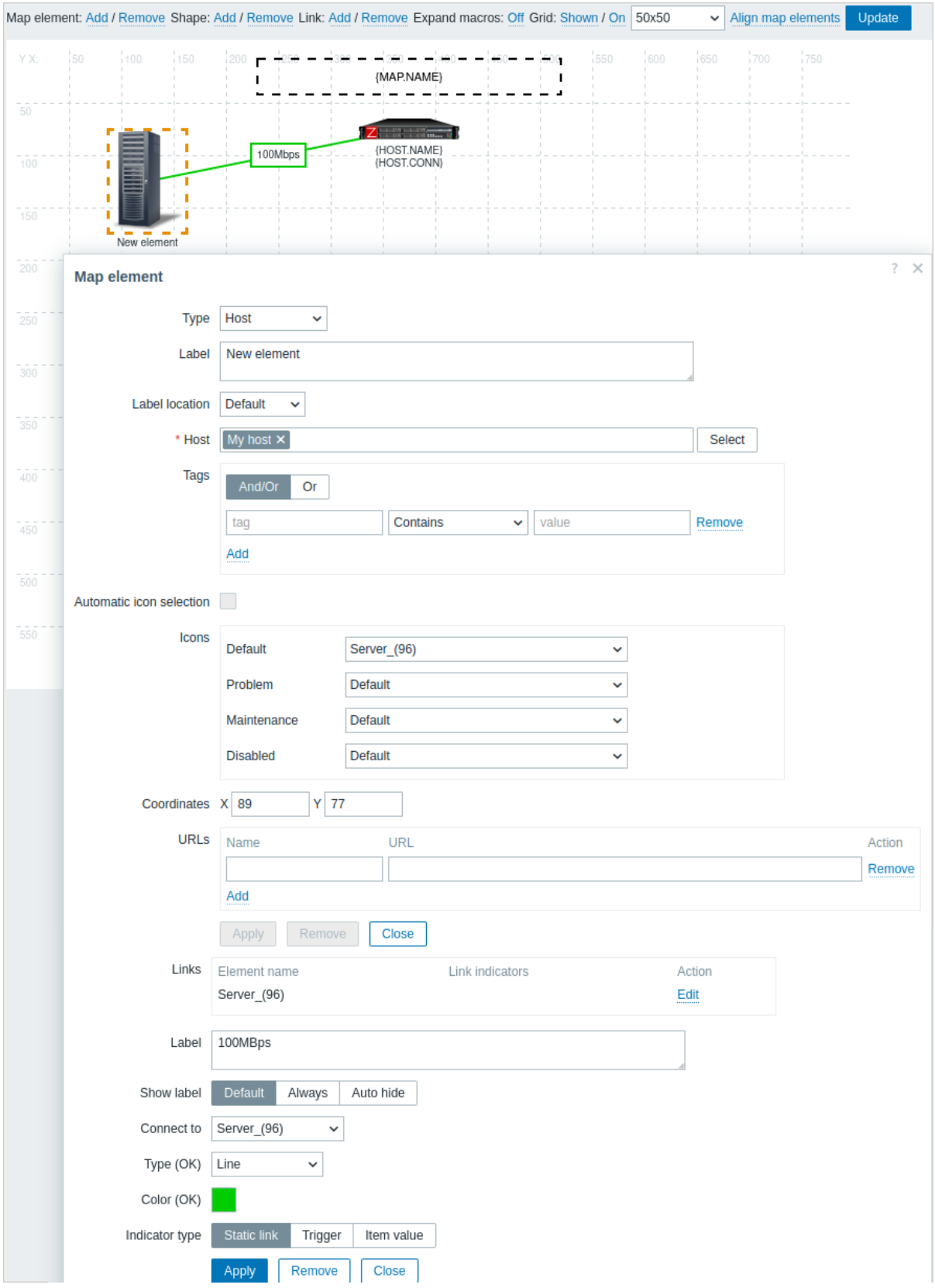
Une fois que vous avez sélectionné plusieurs éléments, le formulaire de propriété de l'élément passe en mode de mise à jour en masse afin que vous puissiez modifier les attributs des éléments sélectionnés en une seule fois. Pour ce faire, marquez l'attribut à l'aide de la case à cocher et entrez une 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 relier. Pour relier 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 de l'élément unique contient désormais une section supplémentaire, *Liens*. Cliquez sur *Modifier* pour modifier les attributs du lien.



Attributs du lien :

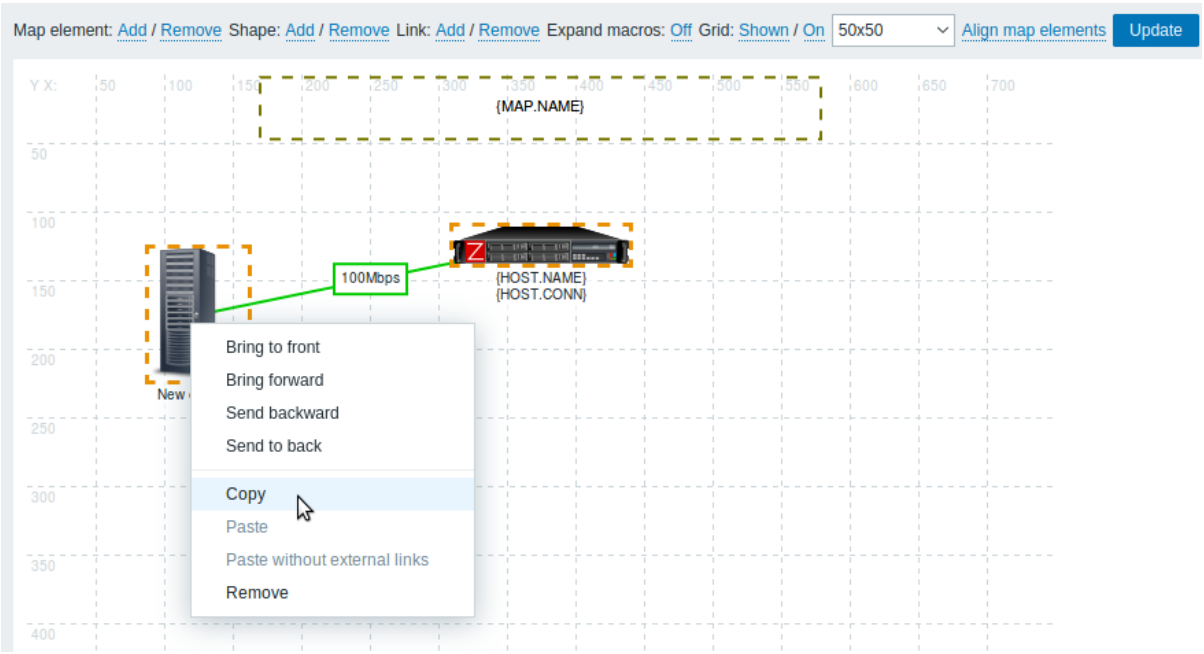
Paramètre	Description
Libellé	Libellé qui sera affiché au-dessus du lien. Les macros d'expression sont prises en charge dans ce champ, mais uniquement avec les fonctions `avg`, `last`, `min` et `max`, avec le temps comme paramètre (par exemple, `{?avg(/host/key,1h)}`).

Paramètre	Description
Afficher le libellé	Sélectionnez comment afficher le libellé du lien : **Par défaut** - utiliser les paramètres *Afficher les libellés des éléments de la carte* de la configuration de la carte (par défaut) ; **Toujours** - toujours affiché (par défaut) ; **Masquage automatique** - masquer le libellé s'il n'est ni survolé ni sélectionné.
Connecter à	L'élément auquel le lien se connecte.
Type (OK)	Style de lien par défaut. Sélectionnez le style : *Ligne*, *Ligne épaisse*, *Point*, ou *Ligne en pointillés*.
Couleur (OK)	Couleur de lien par défaut. Cliquez sur l'échantillon de couleur pour sélectionner une autre couleur.
Type d'indicateur	Sélectionnez le type d'indicateur du lien : **Lien statique** - aucun indicateur ; **Déclencheur** - autoriser les indicateurs de lien basés sur des déclencheurs ; **Valeur d'élément** - autoriser les indicateurs de lien basés sur des valeurs d'élément.
Élément	Sélectionnez l'élément. La valeur de cet élément affectera le style du lien. Ce champ est disponible lorsque *Valeur d'élément* est sélectionné pour *Type d'indicateur*.
Indicateurs	La liste des indicateurs de lien. Ce champ est disponible lorsque *Type d'indicateur* est défini sur *Déclencheur* ou *Valeur d'élément*. S'il est défini sur *Déclencheur*, des déclencheurs peuvent être attribués comme indicateurs de lien. Lorsqu'un déclencheur est à l'état *Problème*, son style (couleur et type de ligne sélectionnés) sera appliqué au lien. S'il est défini sur *Valeur d'élément* et qu'un élément est sélectionné dans le champ *Élément*, des 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 motif (pour un élément texte), son style (couleur et type de ligne sélectionnés) est appliqué au lien.

Déplacer et copier-coller des éléments

Plusieurs éléments sélectionnés peuvent être **déplacés** vers un autre endroit de la carte en cliquant sur l'un des éléments sélectionnés, en maintenant le bouton de la souris enfoncé et en déplaçant le curseur à l'emplacement souhaité.

Un ou plusieurs éléments peuvent être **copiés** en sélectionnant les éléments, puis en cliquant sur un élément sélectionné avec le bouton droit de la souris et en sélectionnant *Copier* dans le menu.

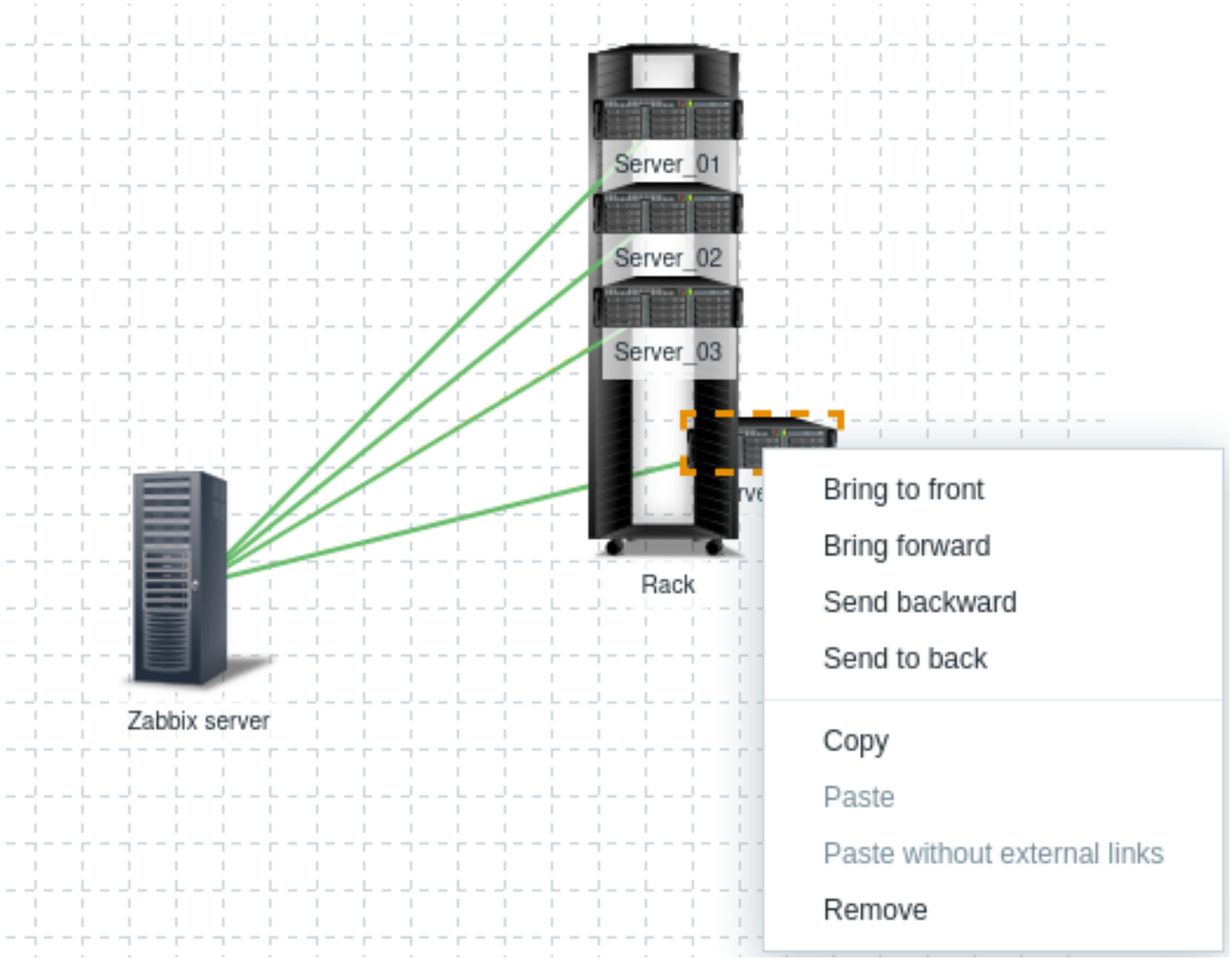


Pour coller les éléments, cliquez sur une zone de la carte avec le bouton droit de la souris et sélectionnez *Coller* dans le menu. L'option *Coller sans liens externes* collera les éléments en ne conservant que les liens qui se trouvent entre les éléments sélectionnés.

Le copier-coller fonctionne dans la même fenêtre de navigateur. Les raccourcis clavier ne sont pas pris en charge.

Ordonnancement des éléments

Pour placer un élément devant un autre (ou inversement), cliquez sur l'élément avec le bouton droit de la souris et sélectionnez *Avancer/Mettre au premier plan* ou *Reculer/Mettre à l'arrière-plan*.

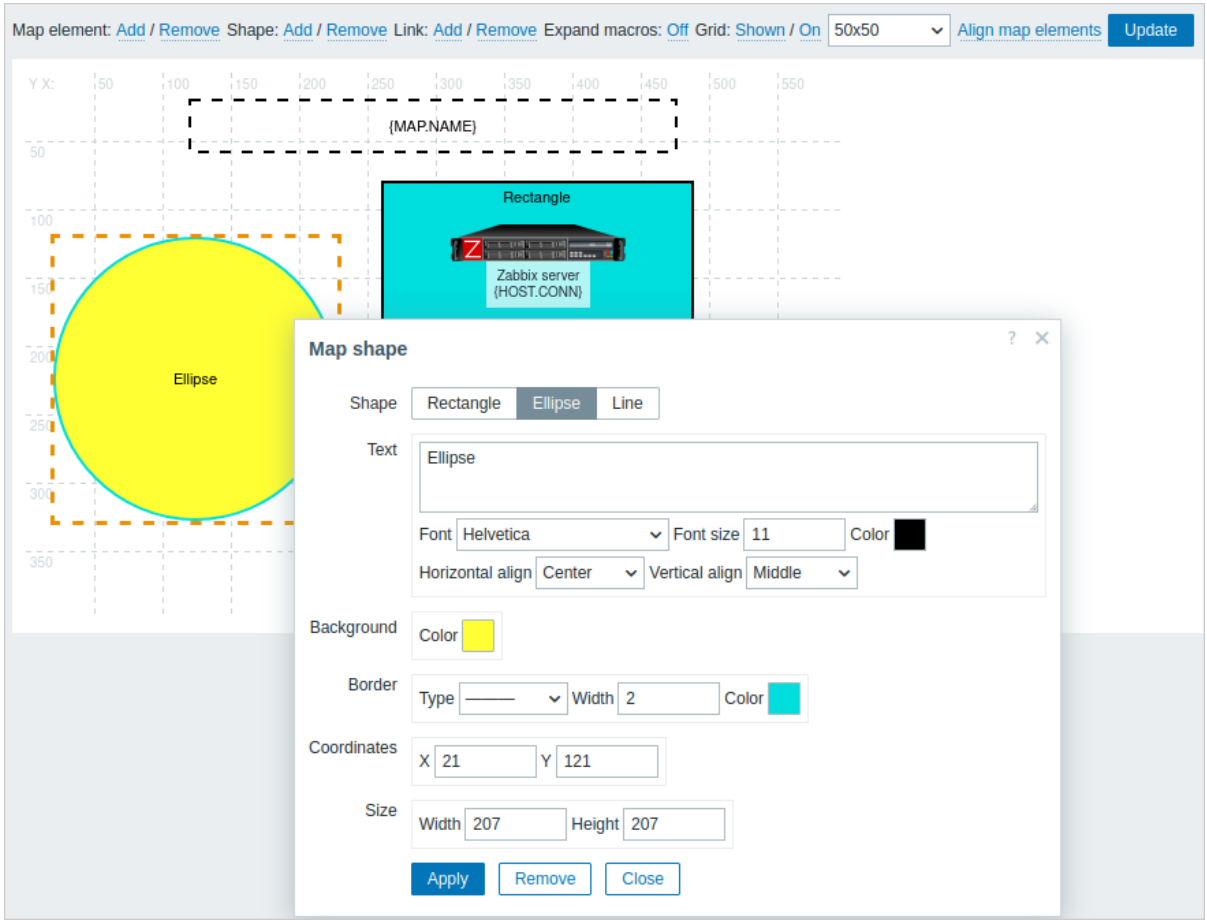


Ajout de formes

En plus des éléments de carte, il est également possible d'ajouter des formes. Les formes ne sont pas des éléments de carte ; elles constituent simplement 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 Shape. La nouvelle forme apparaîtra dans le coin supérieur gauche de la carte. Faites-la glisser-déposer où vous le souhaitez.

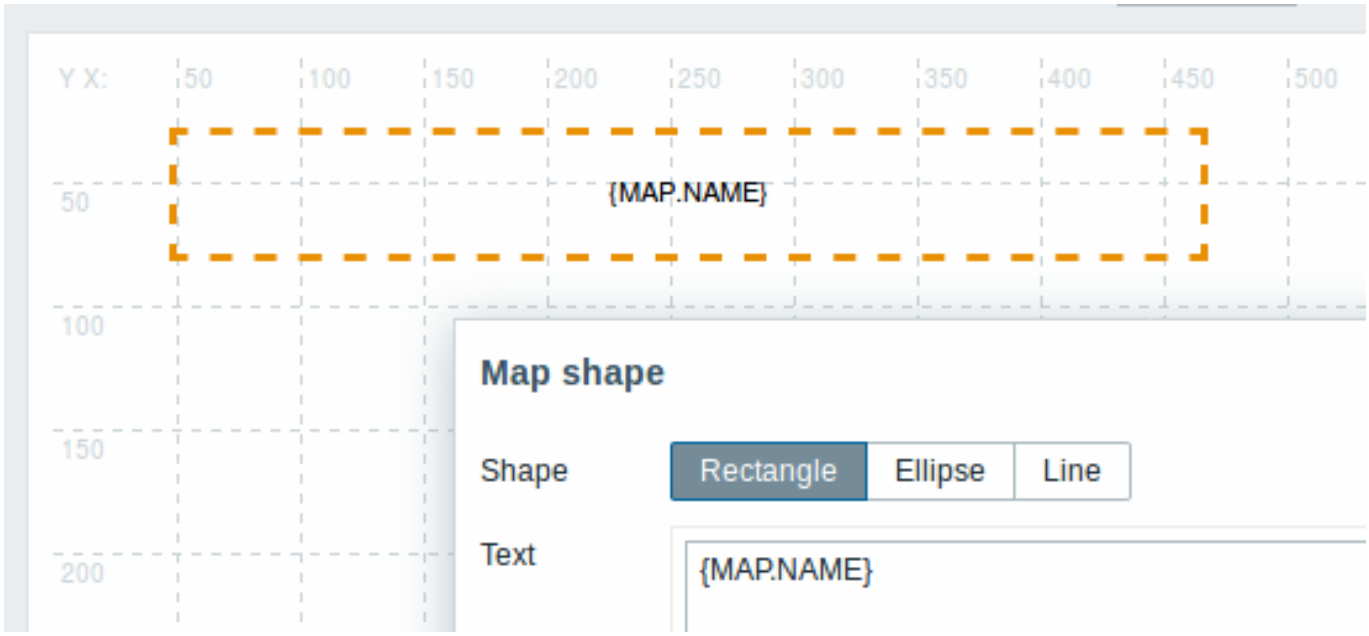
Une nouvelle forme est ajoutée avec des couleurs par défaut. En cliquant sur la forme, un formulaire s'affiche et vous pouvez personnaliser son apparence, ajouter du texte, etc.



Pour sélectionner des formes, sélectionnez-en une, puis maintenez la touche *Ctrl* enfoncée pour sélectionner les autres. Lorsque plusieurs formes sont sélectionnées, les propriétés communes peuvent être mises à jour en masse, de la même manière qu'avec 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 le 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 qui peut être constaté en cliquant sur la macro :



{MAP.NAME} est remplacée par 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.

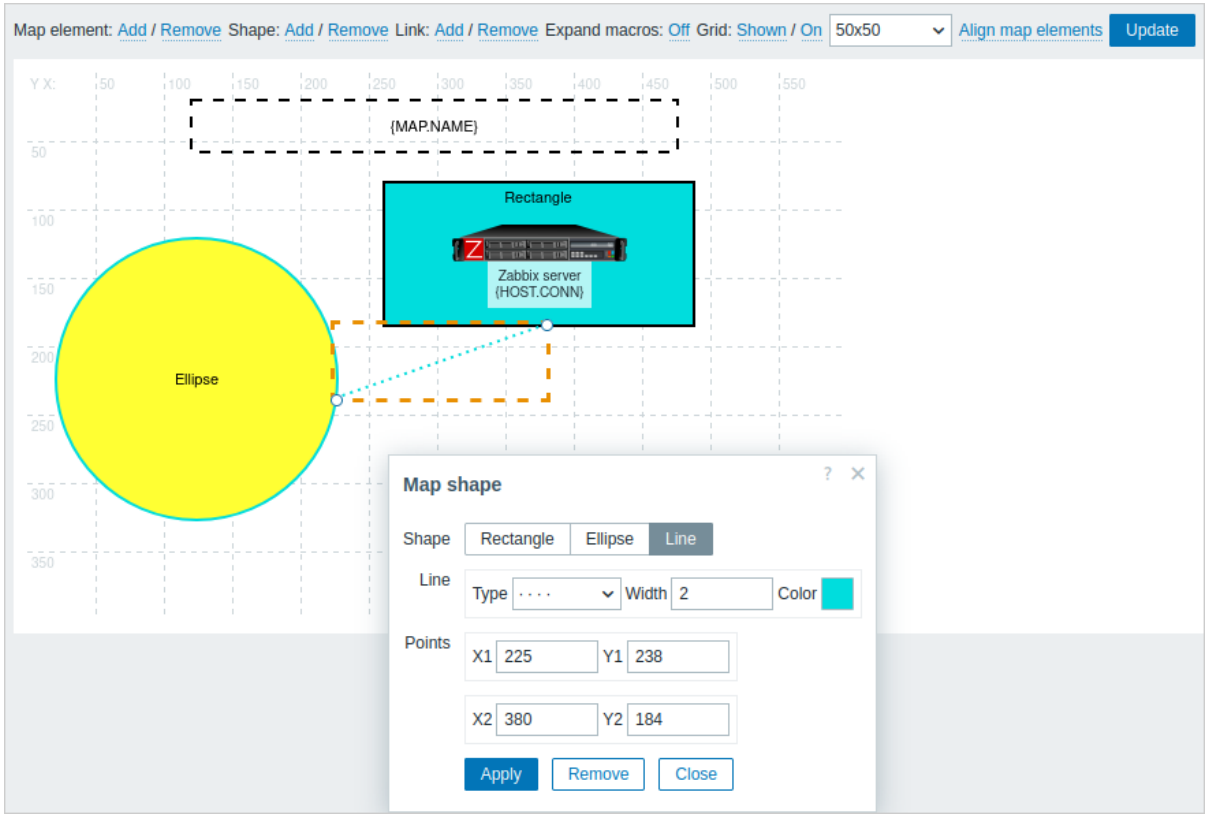
Le retour à la ligne du texte est toujours activé dans les formes. Cependant, dans une ellipse, les lignes sont renvoyées à la ligne

comme si l'ellipse était un rectangle. Le retour à la ligne par mot n'est pas implémenté ; ainsi, les mots longs (les mots qui ne tiennent pas dans la forme) ne sont pas renvoyés à la ligne, mais masqués (sur la page de modification de la carte) ou tronqués (autres pages contenant des cartes).

Ajout de lignes

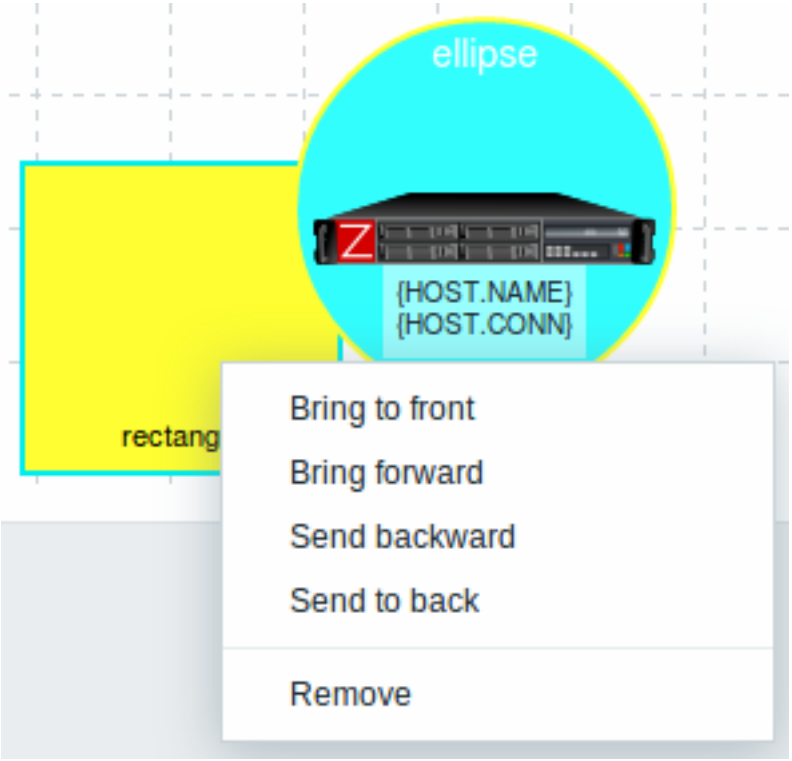
En plus des formes, il est également possible d'ajouter des lignes. Les lignes peuvent être utilisées pour relier des éléments ou des formes sur une carte.

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.



Ordonner les formes et les lignes

Pour placer une forme devant une autre (ou inversement), cliquez sur la forme avec le bouton droit de la souris pour afficher le menu des formes de la carte.

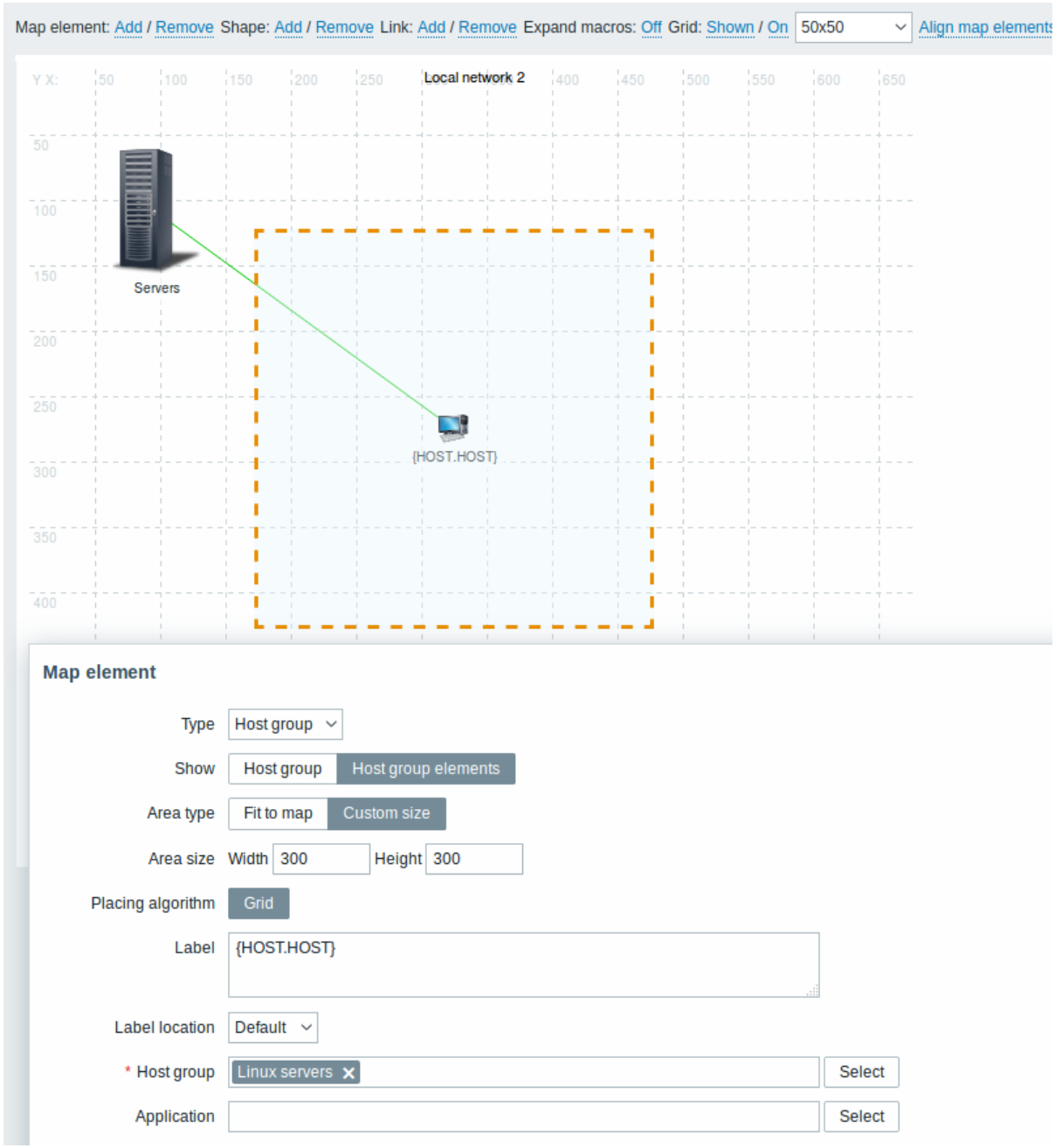


2 Éléments du groupe d’hôtes

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.

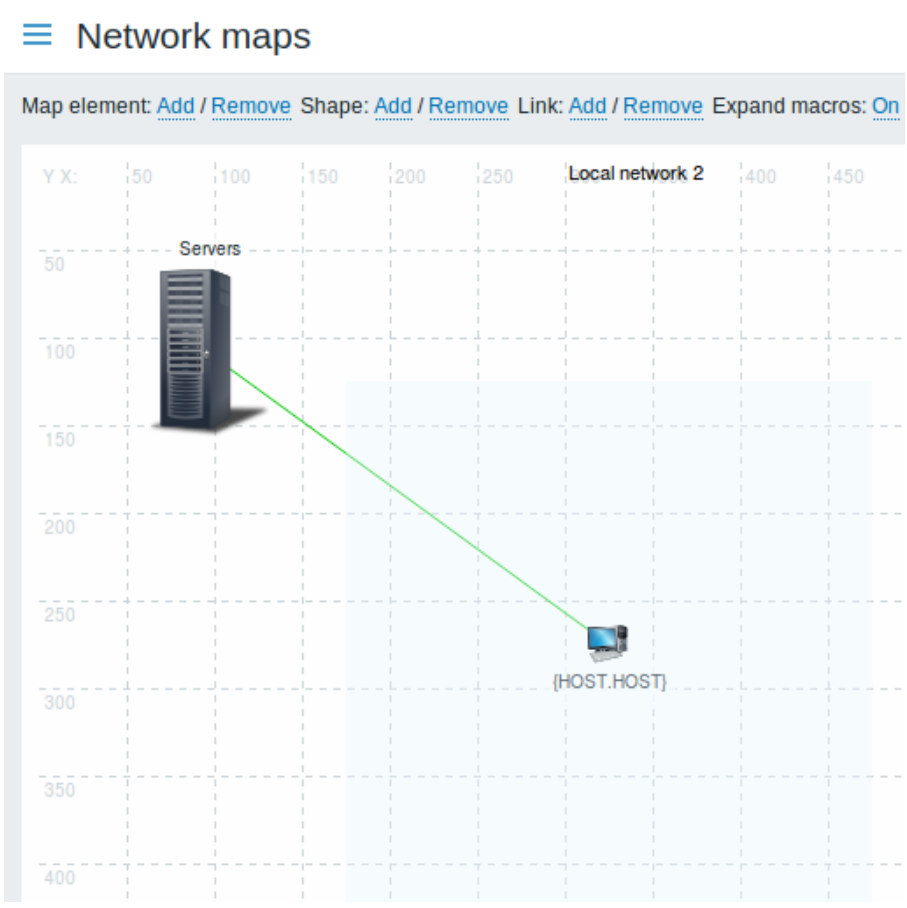
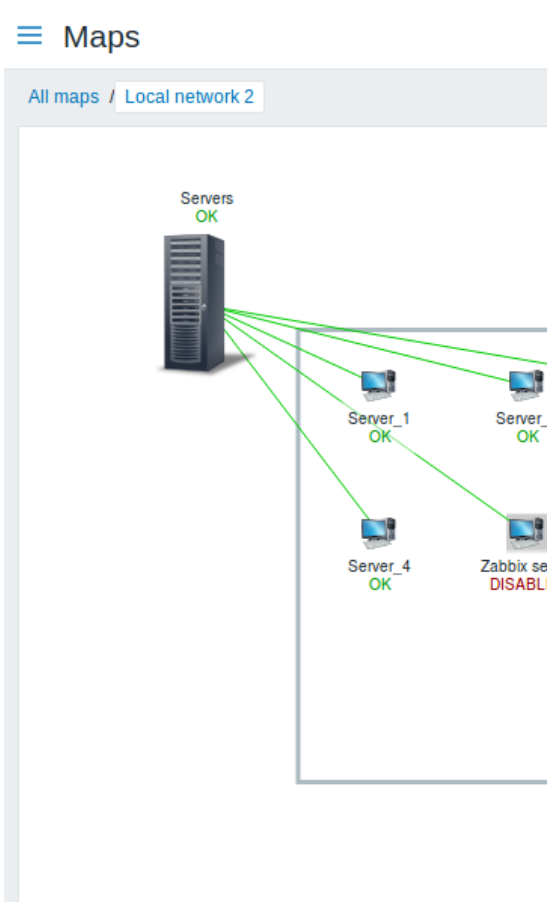
Ce tableau contient les paramètres typiques du type d'élément *Groupe d'hôtes* :

Parameter	Description
Type	Sélectionnez le type de l'élément : **Groupe d'hôtes** - Icône représentant l'état de tous les déclencheurs de tous les hôtes appartenant au groupe sélectionné.
Show	Options d'affichage : **Groupe d'hôtes** - La sélection de cette option affichera une seule icône présentant les informations correspondantes sur le groupe d'hôtes donné. **Éléments du groupe d'hôtes** - La sélection de cette option affichera plusieurs icônes présentant les informations correspondantes sur chaque élément (hôte) du groupe d'hôtes donné.
Area type	Ce paramètre est disponible si *Show* est défini sur *Éléments du groupe d'hôtes* : **Ajuster à la carte** - Tous les éléments du groupe d'hôtes sont placés de manière égale sur la carte. **Taille personnalisée** - Définition manuelle de la zone de la carte dans laquelle tous les éléments du groupe d'hôtes seront affichés.

Parameter	Description
Area size	Ce paramètre est disponible si *Area type* est défini sur *Taille personnalisée* : **Largeur** - Valeur en pixels permettant de spécifier la largeur de la zone de la carte. **Hauteur** - Valeur en pixels permettant de spécifier la hauteur de la zone de la carte.
Placing algorithm	**Grille** - Seule option disponible pour afficher tous les éléments du groupe d'hôtes.
Label	Libellé de l'icône, chaîne quelconque. Les macros et les chaînes multilignes peuvent être utilisées dans les libellés. Si le type de l'élément de carte est *Groupe d'hôtes*, la spécification de certaines macros a un impact sur la vue de la carte en affichant les informations correspondantes pour chaque hôte. Par exemple, si la macro {HOST.IP} est utilisée, la vue de modification de la carte n'affichera que la macro {HOST.IP} elle-même, tandis que la vue de la carte inclura et affichera 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 choisissez *Éléments du groupe d'hôtes* comme option *afficher*, vous ne verrez d'abord qu'une seule icône pour le groupe d'hôtes.
Cependant, lorsque vous enregistrez la carte puis accédez à la vue de la carte, vous verrez que la carte inclut tous les éléments (hôtes) du groupe d'hôtes concerné :

Vue d'édition de la carte	Vue de la carte
	

Remarquez comment la macro {HOST.HOST} est utilisée.
Lors de l'édition 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.

3 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 sur 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 du 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).

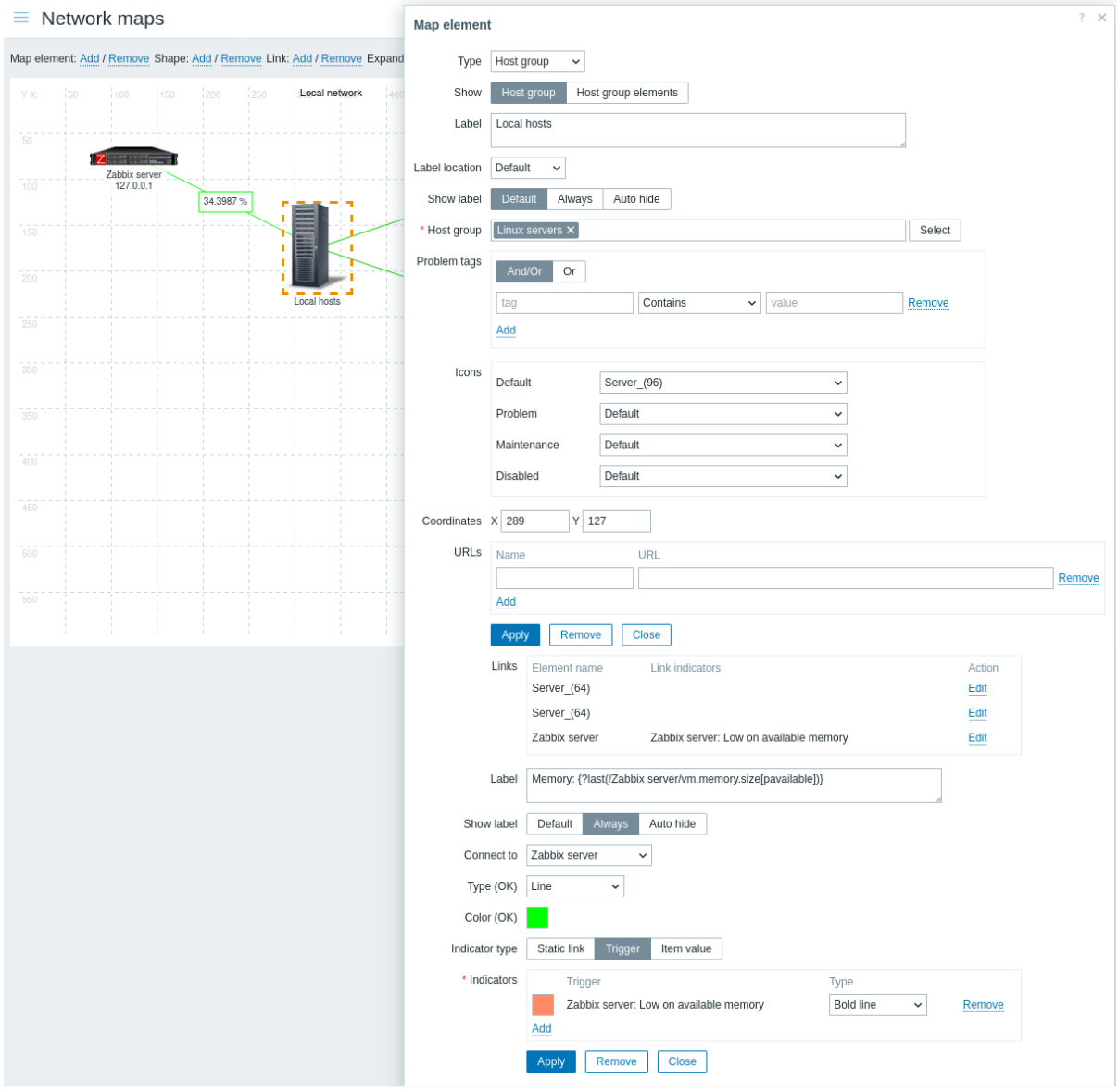
De même, si la valeur d'un élément atteint un seuil spécifié ou correspond à une expression régulière spécifiée, le style du lien peut le refléter.

Configuration

Déclencheurs

Pour attribuer des déclencheurs comme indicateurs de lien, procédez comme suit :

- sélectionnez un élément de carte
- cliquez sur *Modifier* dans la section *Liens* pour le lien approprié
- sélectionnez *Déclencheur* comme type d'indicateur
- cliquez sur *Ajouter* dans le bloc *Indicateurs* et sélectionnez un ou plusieurs déclencheurs

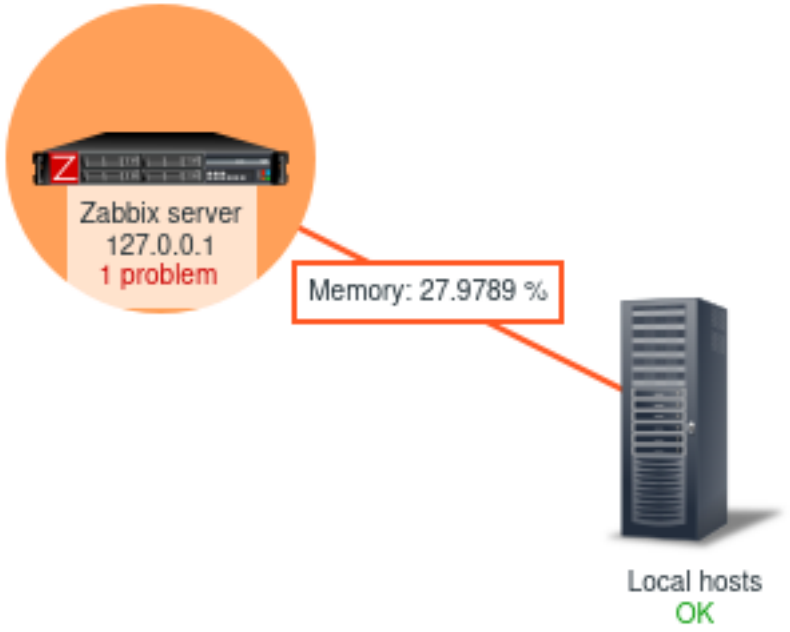


Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge.

Les déclencheurs ajoutés sont visibles dans la liste *Indicateurs*.

Vous pouvez définir le type et la couleur du lien pour chaque indicateur directement depuis la liste. Une fois terminé, cliquez sur *Appliquer*, fermez le formulaire puis cliquez sur *Mettre à jour* pour enregistrer les modifications de la carte.

Dans *Surveillance* → *Cartes*, le type et la couleur du lien correspondants seront affichés si le **déclencheur** passe à l'état de problème.



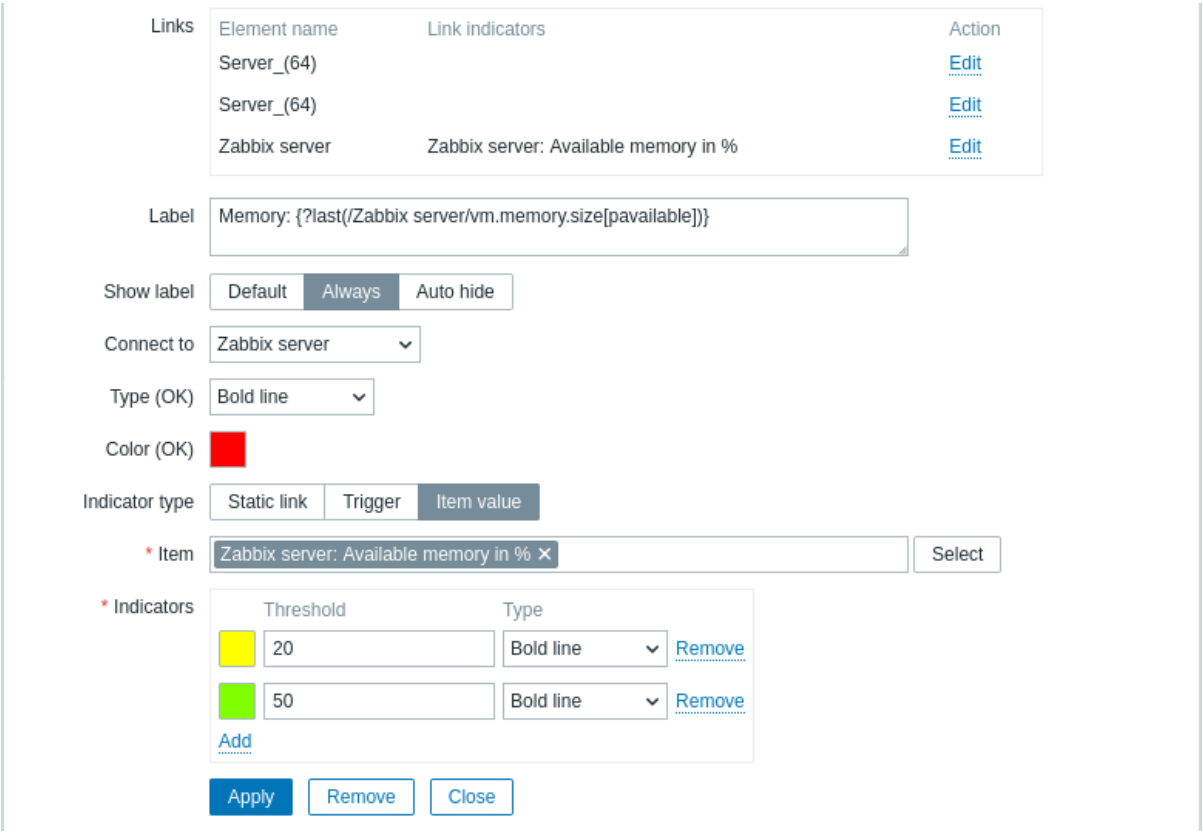
Note:
Si plusieurs déclencheurs passent à l'état de problème, le problème ayant la sévérité la plus élevée déterminera le style et la couleur du lien. Si plusieurs déclencheurs de même sévérité sont attribués au même lien de carte, celui ayant l'ID le plus faible est prioritaire. Notez également que :

1. les paramètres *Sévérité minimale du déclencheur* et *Afficher le problème supprimé* 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 :

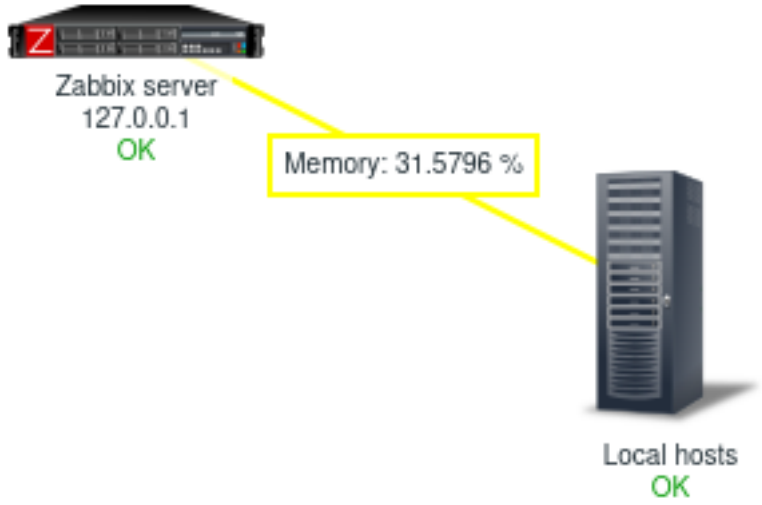
- sélectionnez un élément de carte
- cliquez sur *Modifier* dans la section *Liens* pour le lien approprié
- sélectionnez *Valeur d'élément* comme type d'indicateur
- sélectionnez l'élément
- ajoutez un ou plusieurs seuils ou motifs de valeur d'élément dans le bloc *Indicateurs*



Les seuils/motifs d'élément ajoutés sont visibles dans la liste *Indicateurs*.

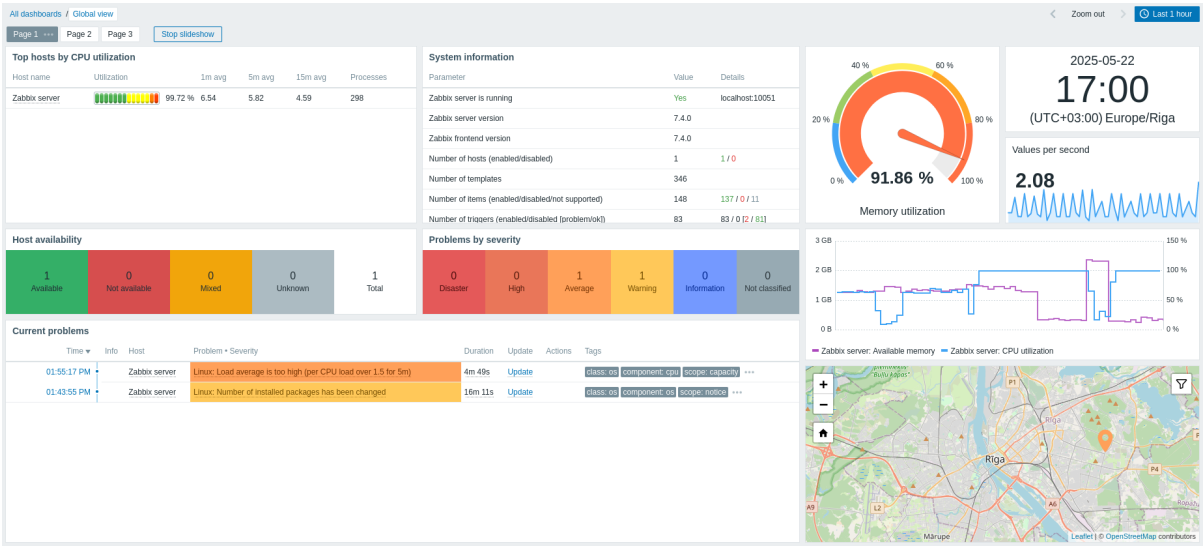
Vous pouvez définir le type et la couleur du lien pour chaque indicateur directement depuis la liste. Une fois terminé, cliquez sur *Appliquer*, fermez le formulaire puis cliquez sur *Mettre à jour* pour enregistrer les modifications de la carte.

Dans *Surveillance* → *Cartes*, le type et la couleur du lien 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 texte).



3 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 grâce à des widgets et des outils tels que des graphiques modernes, des cartes, des diaporamas, et bien plus encore.



8 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.

9 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*.

Lors de la mise à niveau de Zabbix, les modèles existants ne sont pas mis à jour automatiquement afin d'éviter l'écrasement des modifications personnalisées. Pour mettre à niveau les modèles existants ou en ajouter de nouveaux, consultez Mise à niveau des modèles.

Veuillez utiliser la barre latérale pour accéder aux informations concernant les types de modèles spécifiques et les exigences de fonctionnement.

Voir aussi :

- Lier un modèle
- Problèmes connus des modèles

Mise à niveau des modèles

Pour mettre à niveau des modèles existants ou en ajouter de nouveaux après une mise à niveau de Zabbix :

1. Accédez au dépôt Git Zabbix.
2. Dans le sélecteur de branche (sous l'étiquette *Source*), choisissez la branche correspondant à votre version de Zabbix (par exemple, *master* pour la dernière version de développement).
3. Ouvrez 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 web de Zabbix, accédez à *Data collection* > *Templates* et importez le fichier local.

Compatibilité avec l'assistant d'hôte

Pour mettre à niveau plusieurs modèles compatibles avec le Host Wizard, vous pouvez importer les lots de modèles suivants :

- cloud_aws_azure.yaml
- cloud_gcp_oracle.yaml
- database.yaml
- network_os.yaml
- software.yaml

Pour obtenir la liste complète des modèles compatibles, consultez templates_in_bundles.md.

Pour rendre un modèle personnalisé compatible avec le Host Wizard :

1. Dans l'interface web de Zabbix, accédez à *Collecte de données* > *Modèles* et exportez le modèle personnalisé.
2. Modifiez le fichier exporté :
   - Ajoutez l'élément de modèle `wizard_ready` pour indiquer la compatibilité avec le Host Wizard.
   - Si nécessaire, ajoutez également l'élément de modèle `readme` ou `config`.
3. Dans l'interface web de Zabbix, accédez à *Collecte de données* > *Modèles* et importez le fichier de modèle mis à jour.

1 Fonctionnement des modèles Zabbix agent

Étapes 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 importer le modèle).

3. Si nécessaire, ajustez les valeurs des macros du modèle.

4. Configurez l'instance surveillée pour autoriser le partage de données avec Zabbix.

Une description détaillée d'un modèle, y compris la liste complète des macros, éléments de données et déclencheurs, est disponible dans le fichier README du modèle (accessible en cliquant sur le nom du modèle).

Les modèles suivants sont disponibles :

- Apache par Zabbix agent
- Apache par Zabbix agent actif
- HAProxy par Zabbix agent
- IIS par Zabbix agent
- IIS par Zabbix agent actif
- Microsoft Exchange Server 2016 par Zabbix agent
- Microsoft Exchange Server 2016 par Zabbix agent actif
- MySQL par Zabbix agent
- MySQL par Zabbix agent actif
- Nginx par Zabbix agent
- Nginx par Zabbix agent actif

- PHP-FPM par Zabbix agent
- PHP-FPM par Zabbix agent actif
- PostgreSQL par Zabbix agent
- PostgreSQL par Zabbix agent actif
- Cluster RabbitMQ par Zabbix agent

2 Fonctionnement des modèles Zabbix agent 2

Étapes pour garantir le bon fonctionnement des modèles qui collectent des métriques avec Zabbix agent 2 :

1. Assurez-vous que l'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 l'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 du modèle. Notez que les macros utilisateur peuvent être utilisées pour remplacer les paramètres de configuration.

4. Configurez l'instance surveillée afin d'autoriser le partage des données avec Zabbix.

Attention:
Les modèles Zabbix agent 2 fonctionnent conjointement avec les plugins. Bien que la configuration de base puisse être effectuée simplement en ajustant les macros utilisateur, une personnalisation plus poussée peut être obtenue 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 ex. MySQL1 et MySQL2) en spécifiant, dans le fichier de configuration, une session nommée avec son propre URI, nom d'utilisateur et mot de passe pour chaque entité.

Une description détaillée d'un modèle, y compris la liste complète des macros, éléments de données et déclencheurs, est disponible dans le fichier README du modèle (accessible en cliquant sur le nom du modèle).

Les modèles suivants sont disponibles :

- Ceph par Zabbix agent 2
- Docker
- Memcached
- Cluster MongoDB par Zabbix agent 2
- Nœud MongoDB par Zabbix agent 2
- MySQL par Zabbix agent 2
- MySQL par Zabbix agent 2 actif
- Oracle par Zabbix agent 2
- PostgreSQL par Zabbix agent 2
- PostgreSQL par Zabbix agent 2 actif
- Redis
- SMART par Zabbix agent 2
- SMART par Zabbix agent 2 actif
- Systemd par Zabbix agent 2
- Certificat de site web par Zabbix agent 2

3 Fonctionnement des modèles HTTP

Étapes pour garantir le bon fonctionnement des modèles qui collectent des métriques avec l'agent HTTP :

1. Créez dans Zabbix un hôte correspondant à la cible de supervision. Indiquez l'adresse IP/le nom DNS de la cible de supervision comme interface principale de l'hôte. Cela 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 pour autoriser le partage de données avec Zabbix.

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).

Les modèles suivants sont disponibles :

- Acronis Cyber Protect Cloud par HTTP
- Apache par HTTP
- Asterisk par HTTP
- AWS par HTTP
- AWS Cost Explorer par HTTP
- AWS EC2 par HTTP
- AWS ECS Cluster par HTTP
- AWS ECS Serverless Cluster par HTTP
- AWS ELB Application Load Balancer par HTTP
- AWS ELB Network Load Balancer par HTTP
- AWS Lambda par HTTP
- AWS RDS instance par HTTP
- AWS S3 bucket par HTTP
- Azure par HTTP
- Cisco Meraki organization par HTTP
- Cisco SD-WAN par HTTP
- Cisco Secure Firewall Threat Defense par HTTP
- ClickHouse par HTTP
- Cloudflare par HTTP
- CockroachDB par HTTP
- Control-M enterprise manager par HTTP
- Control-M server par HTTP
- Cradlepoint NCM v2 par HTTP
- DELL PowerEdge R720 par HTTP
- DELL PowerEdge R740 par HTTP
- DELL PowerEdge R820 par HTTP
- DELL PowerEdge R840 par HTTP
- Domain RDAP par HTTP
- Elasticsearch Cluster par HTTP
- Envoy Proxy par HTTP
- Etcd par HTTP
- FortiGate par HTTP
- GitHub repository par HTTP
- GitLab par HTTP
- Google Cloud Platform (GCP) par HTTP
- Hadoop par HTTP
- HAProxy par HTTP
- HashiCorp Consul Cluster par HTTP
- HashiCorp Consul Node par HTTP
- HashiCorp Nomad par HTTP
- HashiCorp Vault par HTTP
- Hikvision camera par HTTP
- HPE iLO par HTTP
- HPE MSA 2040 Storage par HTTP
- HPE MSA 2060 Storage par HTTP
- HPE Primera par HTTP
- HPE Synergy par HTTP
- InfluxDB par HTTP
- Jenkins par HTTP
- Kubernetes API server par HTTP
- Kubernetes cluster state par HTTP
- Kubernetes Controller manager par HTTP
- Kubernetes kubelet par HTTP
- Kubernetes nodes par HTTP
- Kubernetes Scheduler par HTTP
- MantisBT par HTTP

- Microsoft 365 reports par HTTP
- Microsoft SharePoint par HTTP
- NetApp AFF A700 par HTTP
- Nextcloud par HTTP
- NGINX par HTTP
- NGINX Plus par HTTP
- Nutanix Prism Element par HTTP
- OpenAI Platform par HTTP
- OpenStack par HTTP
- OpenWeatherMap par HTTP
- Oracle Cloud par HTTP
- Palo Alto PA-440 par HTTP
- PHP-FPM par HTTP
- Proxmox VE par HTTP
- Pure Storage FlashArray v1 and v2 par HTTP
- RabbitMQ cluster par HTTP
- Ribbon SBC Edge par HTTP
- Ribbon SBC SWe CE par HTTP
- Ribbon SBC SWe Core par HTTP
- TiDB par HTTP
- TiDB PD par HTTP
- TiDB TiKV par HTTP
- Travis CI par HTTP
- Veeam Backup Enterprise Manager par HTTP
- Veeam Backup and Replication par HTTP
- VeloCloud SD-WAN par HTTP
- VeloCloud SD-WAN Edge par HTTP
- YugabyteDB par HTTP
- ZooKeeper par HTTP

4 Fonctionnement des modèles IPMI

Les modèles IPMI ne nécessitent aucune configuration spécifique. Pour commencer la supervision, 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).

Une description détaillée d'un modèle, y compris la liste complète des macros, des éléments de données et des déclencheurs, est disponible dans le fichier README du modèle (accessible en cliquant sur le nom du modèle).

Modèle disponible :

- Chassis by IPMI

5 Fonctionnement des modèles 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 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 de données 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

6 Fonctionnement des modèles ODBC

Étapes pour garantir le bon fonctionnement des modèles qui collectent des métriques via la supervision 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 importer le modèle).
3. Si nécessaire, ajustez les valeurs des macros du modèle.
4. Configurez l'instance supervisée pour autoriser le partage des données avec Zabbix.

Une description détaillée d'un modèle, y compris la liste complète des macros, éléments de données 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
- PostgreSQL by ODBC

7 Modèles standardisés pour les équipements réseau

Aperçu

Afin de fournir une supervision des équipements réseau tels que les commutateurs et les routeurs, nous avons créé deux modèles dits : pour l'équipement réseau lui-même (essentiellement son châssis) et pour l'interface réseau.

Des modèles pour de nombreuses familles d'équipements réseau sont fournis. Tous les modèles couvrent (lorsqu'il est possible d'obtenir ces éléments depuis l'équipement) :

- la supervision des défauts du châssis (alimentations, ventilateurs et température, état global)
- la supervision des performances du châssis (éléments CPU et mémoire)
- la collecte de l'inventaire du châssis (numéros de série, nom du modèle, version du firmware)
- la supervision des interfaces réseau avec IF-MIB et EtherLike-MIB (état de l'interface, charge du trafic de l'interface, état du duplex pour Ethernet)

Si le modèle n'est pas disponible dans votre installation Zabbix, vous devrez peut-être d'abord importer le modèle.

Si vous importez les nouveaux modèles prêts à l'emploi, vous pouvez également vouloir mettre à jour l'expression régulière globale `@Network interfaces for discovery` comme suit :

```
Result is FALSE: ^Software Loopback Interface
Result is FALSE: ^(In)?[lL]oop[bB]ack[0-9._]*$
Result is FALSE: ^NULL[0-9.]*$
Result is FALSE: ^[lL]o[0-9.]*$
Result is FALSE: ^[sS]ystem$
Result is FALSE: ^Nu[0-9.]*$
```

afin de filtrer les interfaces loopback et 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	Certifié
Aruba CX 8300s by SNMP	HPE Aruba Network-ing	Série de commu-tateurs 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	Certifié
Brocade FC SNMP	Brocade	Commutateurs FC Brocade	Brocade 300 SAN Switch-	-	SW-MIB,ENTITY-MIB	Performance Défaut
Brocade_Foundry Stackable SNMP	Brocade	Brocade ICX	Brocade ICX6610, Brocade ICX7250-48, Brocade ICX7450-48F		FOUNDRY-SN-AGENT-MIB, FOUNDRY-SN-STACKING-MIB	Certifié
Brocade_Foundry Nonstackable SNMP	Brocade, Foundry	Brocade MLX, Foundry	Brocade MLXe, Foundry FLS648, Foundry FWSX424		FOUNDRY-SN-AGENT-MIB	Performance Défaut
Check Point Next Generation Firewall by SNMP	Check Point	Pare-feu nouvelle généra-tion	-	Gaia	HOST-RESOURCES-MIB, CHECKPOINT-MIB, UCD-SNMP-MIB, SNMPv2-MIB, IF-MIB	Certifié
Ciena 3906 by SNMP	Ciena	CPE Ciena	Ciena 3906	SAOS	WWP-LEOS-BLADE-MIB, WWP-LEOS-CHASSIS-MIB, WWP-LEOS-SYSTEM-CONFIG-MIB	Certifié
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	Certifié
Cisco IOS SNMP	Cisco	Cisco IOS ver > 12.2 3.5	Cisco C2950	IOS	CISCO-PROCESS-MIB,CISCO-MEMORY-POOL-MIB,CISCO-ENVMON-MIB	Certifié
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	Certifié
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	Certifié

Nom du modèle	Fournisseur	Famille d'appareils	Modèles connus	OS	MIB utilisés	**Tags**
D-Link DES_DGS Switch SNMP	D-Link	Commutateurs D-Link DES/DGX	D-Link DES-xxxx/DGS-xxxx,DLINK DGS-3420-26SC	-	DLINK-AGENT-MIB,EQUIPMENT-MIB,ENTITY-MIB	Certifié
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 Défaut Interfaces
Dell Force S-Series SNMP	Dell	Dell Force série S	S4810		F10-S-SERIES-CHASSIS-MIB	Certifié
Extreme Exos SNMP	Extreme	Extreme EXOS	X670V-48x	EXOS	EXTREME-SYSTEM-MIB,EXTREME-SOFTWARE-MONITOR-MIB	Certifié
FortiGate by SNMP	Fortinet	FortiGate (NGFW)	-	FortiOS	HOST-RESOURCES-MIB FORTINET-FORTIGATE-MIB FORTINET-CORE-MIB SNMPv2-MIB IF-MIB ENTITY-MIB	Performance Inventaire
HP Comware HH3C SNMP	HP	HP (H3C) Comware	HP A5500-24G-4SFP HI Switch		HH3C-ENTITY-EXT-MIB,ENTITY-MIB	Certifié
HP Enterprise Switch SNMP	HP	Commutateur d'entreprise HP	HP ProCurve J4900B Switch 2626, HP J9728A 2920-48G Switch		STATISTICS-MIB,NETSWITCH-MIB,HP-ICF-CHASSIS,ENTITY-MIB,SEMI-MIB	Certifié
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	Certifié
Huawei OceanStor Dorado by SNMP	Huawei	Huawei OceanStor, Huawei OceanStor Dorado	Huawei OceanStor Dorado 3000/5000/6000/8000/18000/5300/5500/5600/6800/18500/18800 with V6/V700 software		HUAWEI-STORAGE-HARDWARE-MIB, HUAWEI-STORAGE-SPACE-MIB, ISM-PERFORMANCE-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Certifié
Huawei VRP by SNMP	Huawei	Huawei VRP	S2352P-EI		ENTITY-MIB,HUAWEI-ENTITY-EXTENT-MIB	Certifié

Nom du modèle	Fournisseur	Famille d'appareils	Modèles connus	OS	MIB utilisés	**Tags**
Intel_Qlogic Infiniband SNMP	Intel/QLogic	Appareils Infini-band In-tel/QLogic	Infiniband 12300		ICS-CHASSIS-MIB	Défaut Inventaire
Juniper SNMP	Juniper	Modèles MX,SRX,EX	Juniper MX240, Juniper EX4200-24F	JunOS	JUNIPER-MIB	Certifié
Juniper MX NETCONF	Juniper	Modèles MX	Juniper MX204 Edge Router	JunOS 24.2R1-S1.10	*Non applicable (utilise NETCONF sur SSH)*	Certifié
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	Certifié
Mellanox SNMP	Mellanox	Appareils Infini-band Mellanox	SX1036	MLNX-OS	HOST-RESOURCES-MIB,ENTITY-MIB,ENTITY-SENSOR-MIB,MELLANOX-MIB	Certifié
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	Certifié

Nom du modèle	Fournisseur	Famille d'appareils	Modèles connus	OS	MIB utilisés	**Tags**
MikroTik CRS<device model> SNMP	MikroTik	Commutateurs routeurs cloud MikroTik (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	Certifié
MikroTik CSS<device model> SNMP	MikroTik	Commutateurs intelligents cloud MikroTik (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	Certifié
MikroTik FiberBox SNMP	MikroTik	MikroTik FiberBox	MikroTik FiberBox	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Certifié
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	Certifié

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	Certifié
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	Certifié
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	Certifié
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	Certifié
Netgear Fastpath SNMP	Netgear	Netgear Fastpath	M5300-28G		FASTPATH-SWITCHING-MIB,FASTPATH-BOXSERVICES-PRIVATE-MIB	Défaut Inventaire
QTech QSW SNMP	QTech	Appareils Qtech	Qtech QSW-2800-28T	-	QTECH-MIB,ENTITY-MIB	Performance Inventaire

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	Certifié
TP-LINK SNMP	TP-LINK	TP-LINK	T2600G-28TS v2.0		TPLINK-SYSMONITOR-MIB,TPLINK-SYSINFO-MIB	Performance Inventaire
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 Inventaire

Conception des modèles

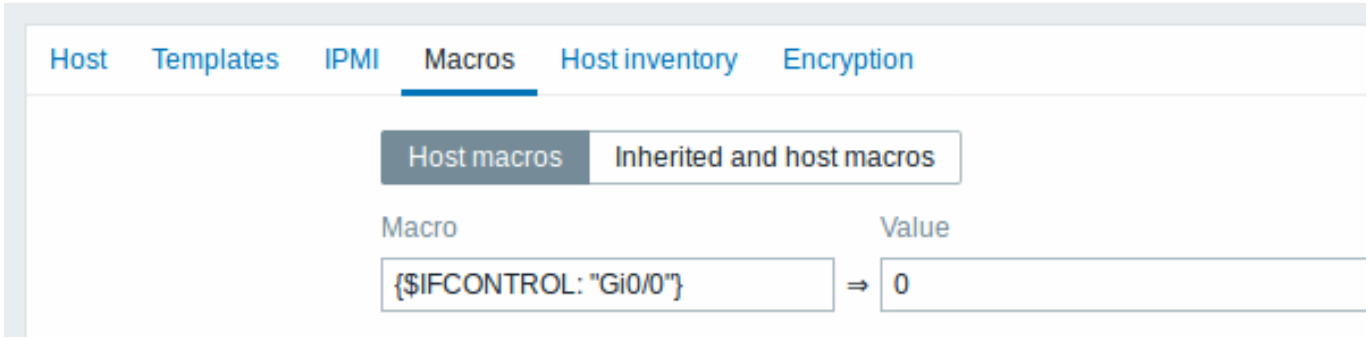
Les modèles ont été conçus en tenant compte des points suivants :

- 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 :

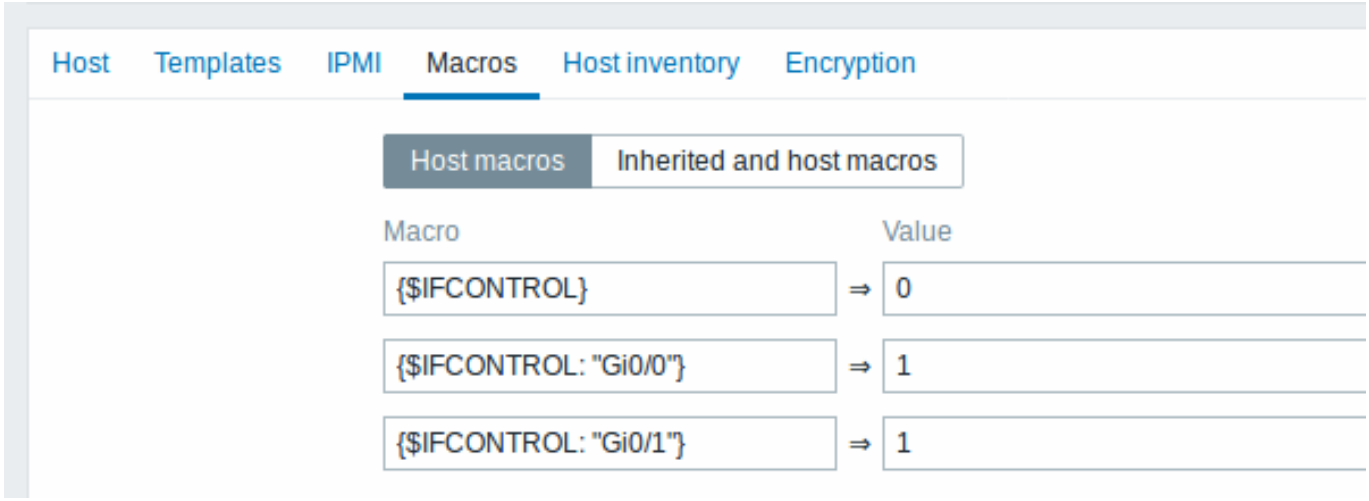
```
{$IFCONTROL:"{#IFNAME}"}=1 and last(/Alcatel Timetra TiMOS SNMP/net.if.status[ifOperStatus.{#SNMPINDEX}])
```

- 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 :



Tags

- Performances - Les MIBs de la famille de périphériques offrent un moyen de surveiller les éléments de la CPU et de la mémoire ;
- Erreur - Les MIBs de la famille des périphériques fournissent un moyen de surveiller au moins un capteur de température ;
- Inventaire - Les MIBs de la famille des périphériques offrent un moyen de collecter au moins le numéro de série du périphérique et le nom du modèle ;
- Certifié - les trois catégories principales ci-dessus sont couvertes.

8 Fonctionnement 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

10 Notifications sur les 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.

1 Types de Media

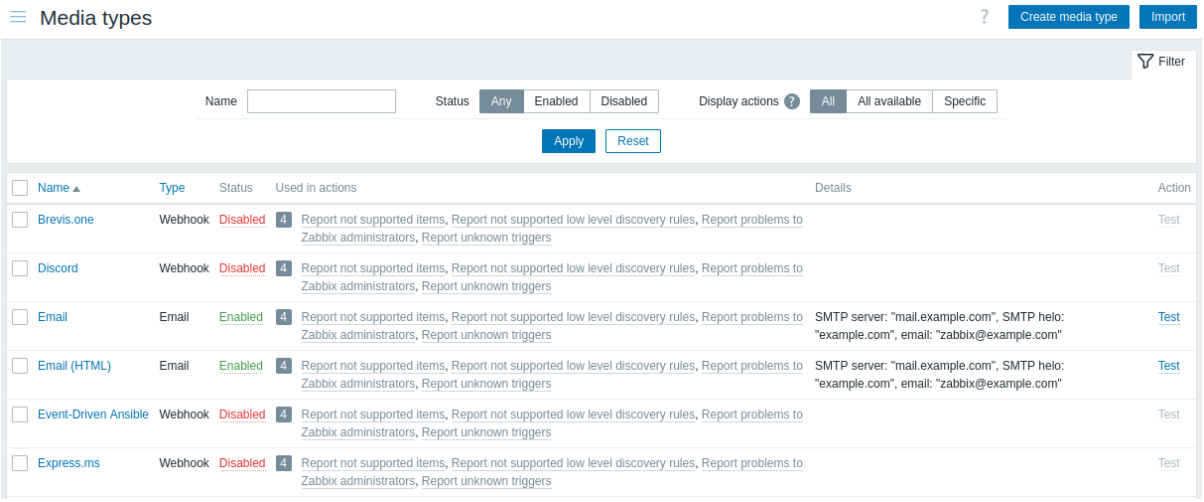
Aperçu

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
- SMS
- Script personnalisé
- Webhook

Les types de média sont gérés dans *Alertes* > *Types de média*. Certains types de média sont prédéfinis dans l'ensemble de données par défaut.
Il vous suffit d'ajuster leurs paramètres pour les rendre opérationnels.



Pour voir comment les types de média s'intègrent dans le processus d'alerte, examinons les trois conditions requises pour transmettre des notifications de Zabbix aux utilisateurs finaux :

1. Une opération d'action doit être définie pour envoyer des notifications
2. Un type de média fonctionnel doit être défini (par exemple *E-mail*, qui envoie des alertes via SMTP)
3. Les informations de diffusion au niveau de l'utilisateur (telles que les adresses e-mail, les numéros de téléphone, etc.) doivent être définies dans le média utilisateur

Test du type de média

Pour tester si un type de média configuré fonctionne, cliquez sur *Test* dans la liste des types de média.

La requête 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 web. L'interface web attendra que le serveur renvoie les résultats. Le test du type de média a un délai d'expiration de 65 secondes par défaut (configurable dans *Administration* > *General* > *Timeouts*).

Voir aussi les détails des tests pour :

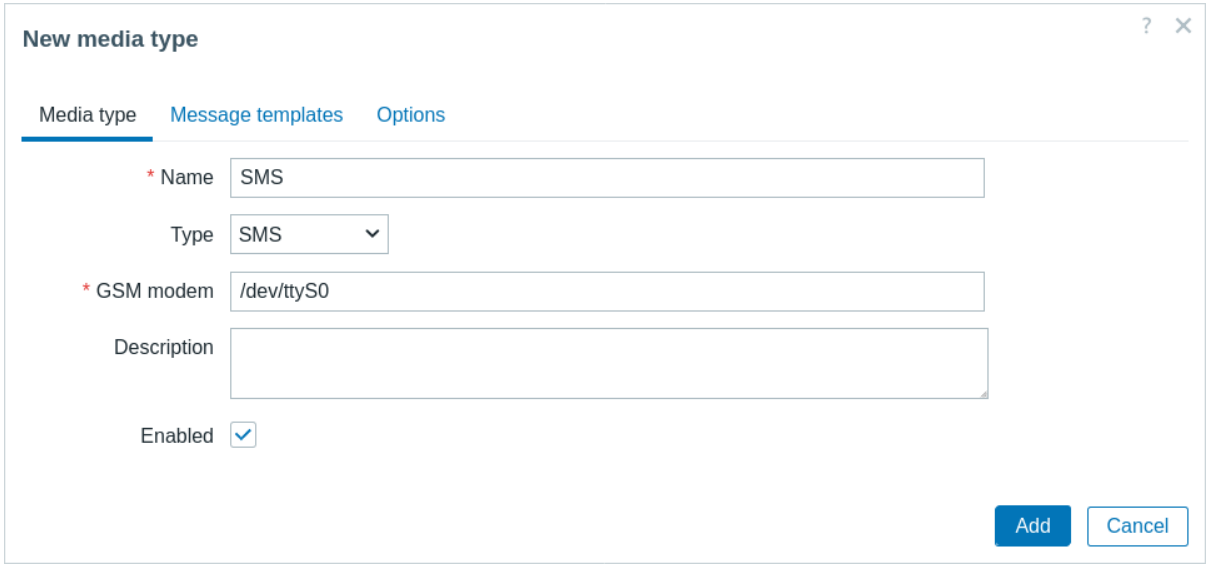
- Email
- Webhook
- Script

Configuration

Pour créer un type de média dans l'interface web de Zabbix :

- Allez à : *Alertes* > *Types de média*
- Cliquez sur *Créer un type de média*
- Saisissez les paramètres du type de média dans le formulaire

Certains paramètres sont **communs** à toutes les méthodes d'envoi.

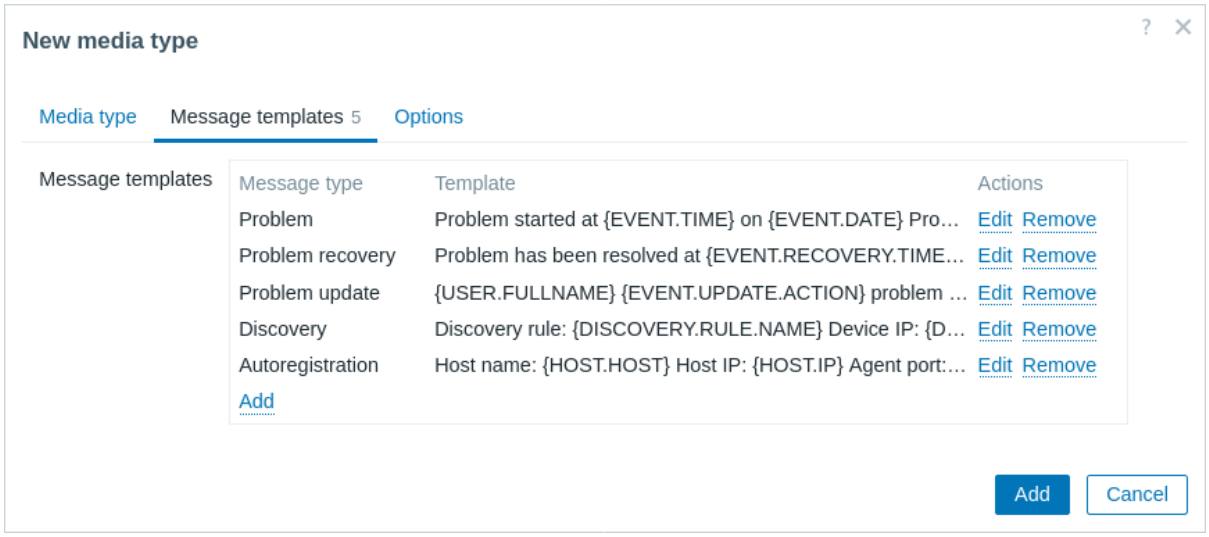


Paramètre	Description
Name	Nom du type de média.
Type	Sélectionnez la méthode d'envoi pour le type de média.
Description	Saisissez une description du type de média.
Enabled	Cochez la case pour activer le type de média.

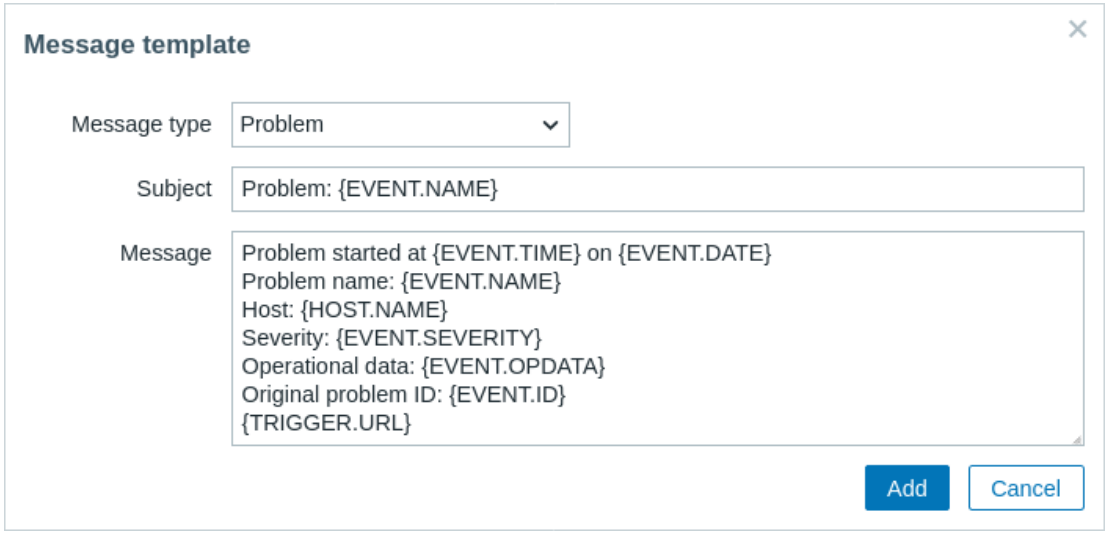
Pour les paramètres spécifiques à chaque méthode, consultez les pages email, SMS, script d'alerte personnalisé ou webhook.

Modèles de message

L'onglet **Modèles de message** contient les messages par défaut pour chaque type d'événement (problème, résolution de problème, découverte, etc.).



Cliquez sur **Ajouter** pour définir un message par défaut (ou sur **Modifier** pour mettre à jour un message existant) :



Paramètre	Description
Type de message	Type d'événement pour lequel le message par défaut doit être utilisé. Un seul message par défaut peut être défini pour chaque type d'événement.
Sujet	Sujet du message par défaut. Le sujet peut contenir des macros. Il est limité à 255 caractères. Le sujet 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 Envoi de 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)}`).

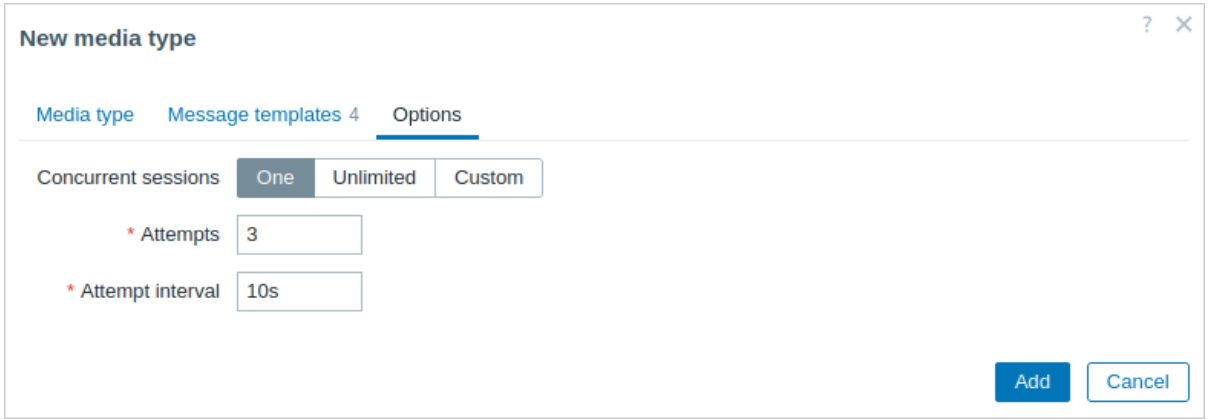
Notez que les messages par défaut sont remplacés par des messages personnalisés, s'ils sont définis dans les opérations d'action.

Warning:
La définition de modèles de message est obligatoire pour toutes les méthodes d'envoi, y compris les webhooks ou les scripts d'alerte personnalisés qui n'utilisent pas les messages par défaut pour les notifications. Par exemple, l'action « Send message to Pushover webhook » n'enverra pas de 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 par type de média,
le nombre total de processus d'alerte sur le serveur ne peut être limité que par le paramètre StartAlerters.
Les alertes générées par un même déclencheur sont traitées de manière séquentielle.
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 d'alerte parallèles pour le type de média : **Une** - une session **Illimité** - nombre illimité de sessions **Personnalisé** - sélectionnez un nombre personnalisé de sessions Des valeurs illimitées/élevées signifient davantage de sessions parallèles et une capacité accrue d'envoi des notifications. Les valeurs illimitées/élevées doivent être utilisées dans les grands environnements où de nombreuses notifications peuvent devoir être envoyées simultanément. Si davantage de notifications doivent être envoyées qu'il n'y a de sessions simultanées, les notifications restantes seront mises en file d'attente ; elles ne seront 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 de nouvelle tentative d'envoi d'une notification en cas d'échec, en secondes (0-3600). Si « 0 » est spécifié, Zabbix réessaiera immédiatement. Les suffixes de temps sont pris en charge, par exemple 5s, 3m, 1h.

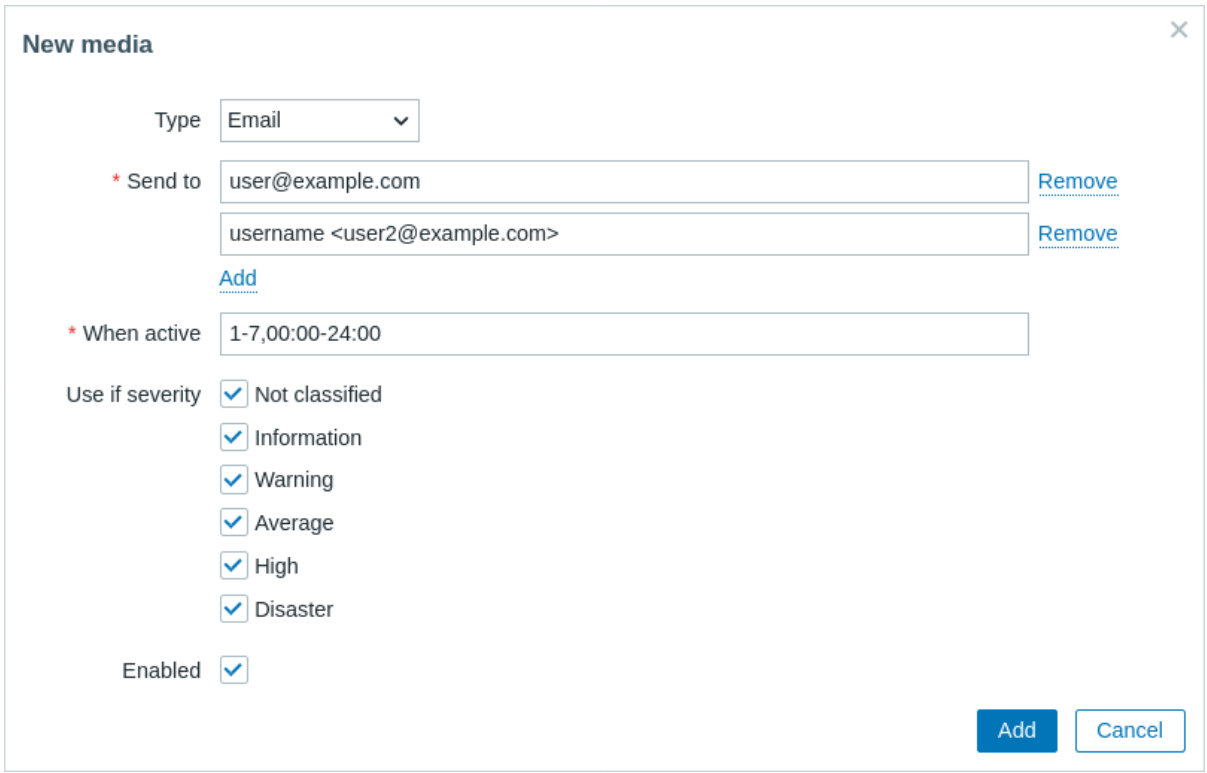
Médias utilisateur

Alors que les types de média définissent **comment** une notification sera envoyée, les médias utilisateur définissent **où** la notification doit être envoyée.

Les médias utilisateur (par ex. adresse e-mail, ID 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 être distribuée si les détails de livraison du webhook *X* ne sont pas définis dans le profil utilisateur d'Admin.

Pour définir des médias utilisateur :

- Allez dans *Users* > *Users* et ouvrez le formulaire des propriétés de l'utilisateur (ou allez dans *User settings* > *Notifications* de votre propre profil utilisateur)
- Cliquez sur *Add* dans l'onglet Media



Parameter	Description
Type	La liste déroulante contient les noms des types de média activés. Notez que lors de la modification d'un média d'un type de média désactivé, le type sera affiché en rouge.

Parameter	Description
Send to	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 sous le champ d'adresse. Dans ce cas, les notifications seront envoyées à toutes les adresses listées. Pour des exemples d'adresses, consultez la description du paramètre *Email* pour le type de média email.
When active	Vous pouvez limiter la période pendant laquelle les messages sont envoyés, par exemple en définissant uniquement les 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 du fuseau horaire du système, cette limite devra peut-être être ajustée en conséquence afin de ne pas manquer de messages importants. Consultez la page Spécification de période temporelle pour une description du format. Les macros utilisateur sont prises en charge.
Use if severity	Cochez les cases des sévérités de déclencheur pour lesquelles vous souhaitez recevoir des notifications. *Notez* que la sévérité par défaut (« Not classified ») **doit être** cochée si vous souhaitez recevoir des notifications pour des événements non liés à des déclencheurs. Après l'enregistrement, les sévérités de déclencheur sélectionnées seront affichées dans les couleurs de sévérité correspondantes, tandis que celles non sélectionnées seront grisées.
Status	Statut du média utilisateur. **Enabled** - est utilisé. **Disabled** - n'est pas utilisé.

1 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.

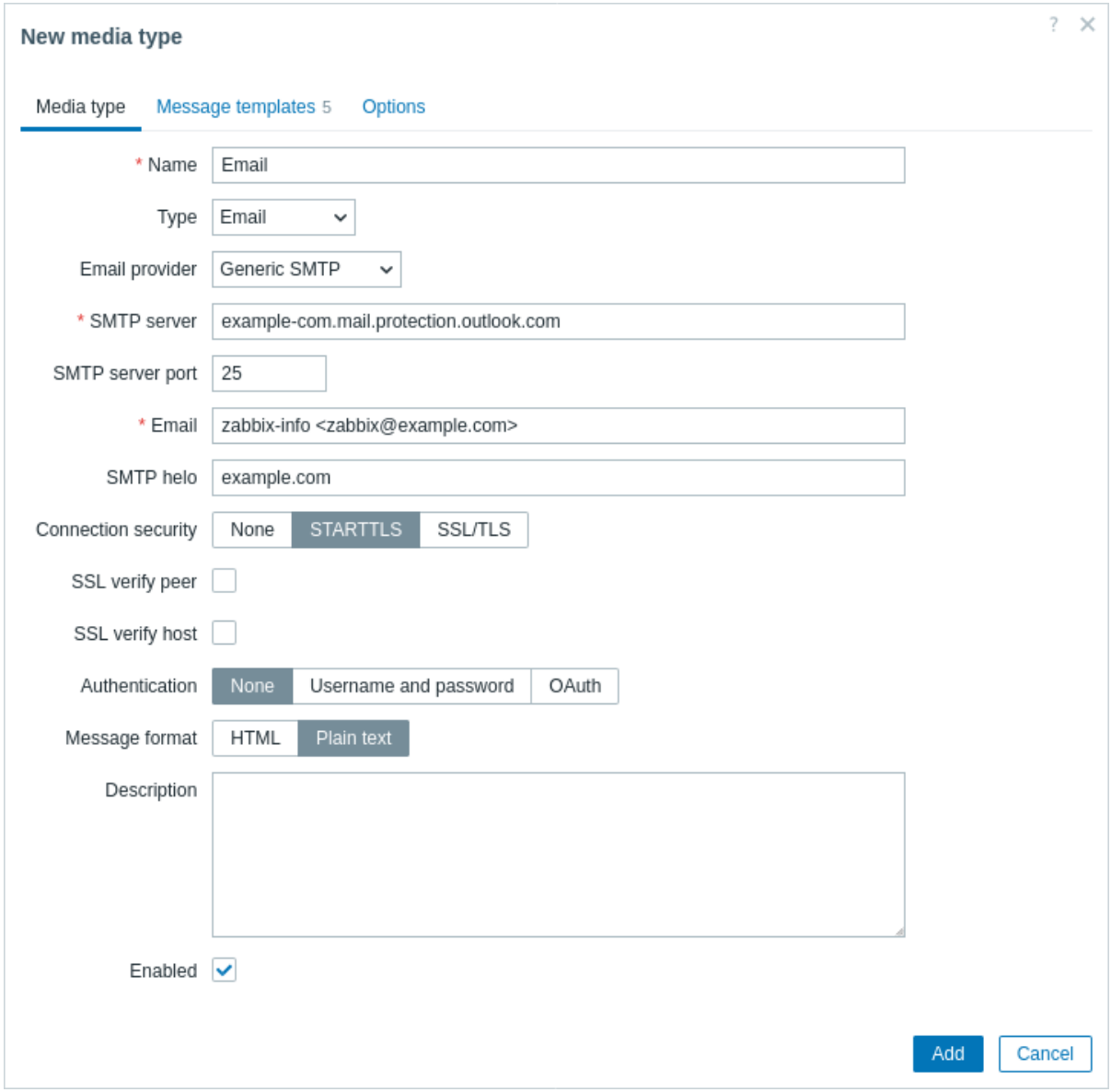
> **Note:**
> Plusieurs notifications pour un même événement seront regroupées dans le même fil de discussion e-mail.

Configuration

Pour configurer l'e-mail comme type de média :

1. Allez dans *Alerts* > *Media types*.
2. Cliquez sur *Create media type* (ou cliquez sur *Email* dans la liste des types de médias prédéfinis).

L'onglet **Media type** 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 texte clair lors de l'export de la configuration du type de média e-mail.

Les paramètres suivants sont spécifiques au type de média e-mail :

Paramètre	Description
Email provider	Sélectionnez le fournisseur d'e-mail : *Generic SMTP*, *Gmail*, *Gmail relay*, *Office365* ou *Office365 relay*. Si vous sélectionnez les options liées à Gmail/Office365, vous n'aurez qu'à fournir l'adresse e-mail de l'expéditeur et le mot de passe ; des options telles que *SMTP server*, *SMTP server port*, *SMTP helo* et *Connection security* seront automatiquement renseignées par Zabbix. Voir aussi : Types de médias Gmail/Office365 automatisés.
SMTP server	Définissez un serveur SMTP pour gérer les messages sortants. Ce champ est disponible si *Generic SMTP* est sélectionné comme fournisseur d'e-mail.
SMTP server port	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.

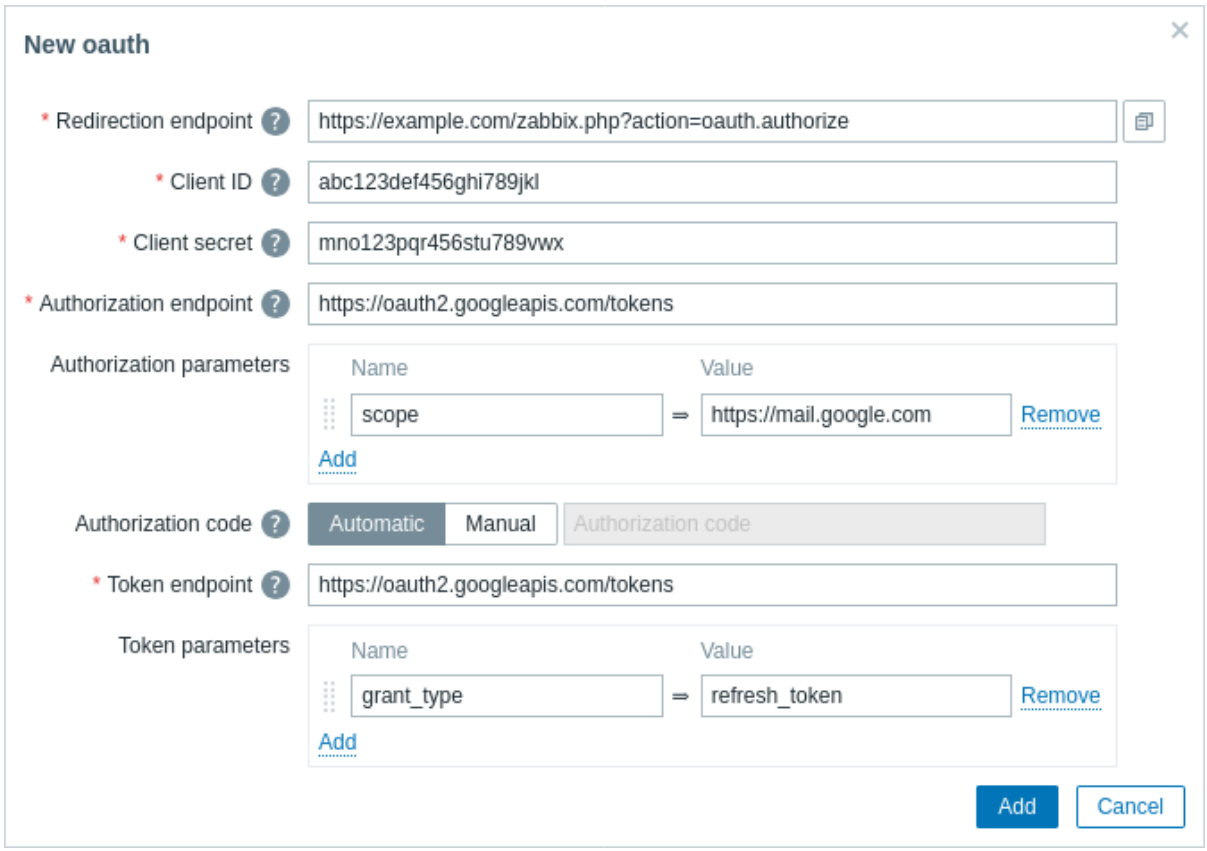
Paramètre	Description
Email	L'adresse saisie ici sera utilisée comme adresse **From** pour les messages envoyés. L'ajout d'un nom d'affichage de l'expéditeur (comme "Zabbix_info" dans *Zabbix_info <zabbix@company.com>* dans la capture d'écran ci-dessus) avec l'adresse e-mail réelle est pris en charge. Il existe certaines restrictions concernant les noms d'affichage dans les e-mails Zabbix par rapport à ce qui est autorisé par la RFC 5322, comme l'illustrent les exemples suivants : Exemples valides : *zabbix@company.com* (adresse e-mail uniquement, sans nécessité d'utiliser des chevrons) *Zabbix_info <zabbix@company.com>* (nom d'affichage et adresse e-mail entre chevrons) *∑Ω-monitoring <zabbix@company.com>* (caractères UTF-8 dans le nom d'affichage) Exemples non valides : *Zabbix HQ zabbix@company.com* (nom d'affichage présent mais sans chevrons autour de l'adresse e-mail) *"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, normalement un nom de domaine. S'il est vide, le nom de domaine de l'e-mail sera envoyé (c'est-à-dire ce qui se trouve après @ dans le champ *Email*). S'il est impossible de récupérer le nom de domaine, un avertissement de niveau debug sera enregistré dans le journal 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.
Connection security	Sélectionnez le niveau de sécurité de la connexion : **None** - 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
SSL verify peer	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. Cela définit l'option cURL CURLOPT_SSL_VERIFYPEER.
SSL verify host	Cochez la case pour vérifier que le champ *Common Name* ou le champ *Subject Alternate Name* du certificat du serveur SMTP correspond. Cela définit l'option cURL CURLOPT_SSL_VERIFYHOST.
Authentication	Sélectionnez le niveau d'authentification : **None** - aucune option cURL n'est définie **Username and password** - implique "AUTH=*" en 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*.
Username	Nom d'utilisateur à utiliser pour l'authentification. Cela définit la valeur de CURLOPT_USERNAME. Macros utilisateur prises en charge.
Password	Mot de passe à utiliser pour l'authentification. Cela définit la valeur de CURLOPT_PASSWORD. Macros utilisateur prises en charge. La valeur saisie ici est exportée telle quelle lors de l'export 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 de la macro sera exportée tandis que la valeur secrète devra être (re)configurée sur le système de destination.
OAuth tokens	Cliquez sur *Configure* pour définir les paramètres de récupération des OAuth tokens dans une nouvelle fenêtre. Ce champ n'est disponible que si "OAuth" est sélectionné dans le champ *Authentication*.
Message format	Sélectionnez le format du message : **HTML** - envoyer au format HTML **Plain text** - envoyer en texte brut

Attention:
Pour activer l'authentification SMTP, le serveur Zabbix doit être compilé avec l'option de compilation `--with-libcurl`, qui nécessite libcurl (voir les exigences server pour les détails de version).

Voir aussi les paramètres communs des types de médias pour plus de détails 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 :



Paramètre	Description
Point de terminaison de redirection	Saisissez l'URL du frontend Zabbix vers laquelle le service OAuth redirigera l'autorisation OAuth (en utilisant le format `https://<zabbix-frontend-url>/zabbix.php?action=oauth.authorize`). Elle est définie automatiquement pour un nouveau type de média e-mail lorsque l'URL du frontend est définie.
ID client	Saisissez l'identifiant unique de l'application cliente enregistrée sur le serveur d'autorisation OAuth.
Secret client	Saisissez le secret privé de l'application cliente enregistrée sur le serveur d'autorisation OAuth.
Point de terminaison d'autorisation	Saisissez l'URL du serveur d'autorisation OAuth pour demander l'autorisation de l'utilisateur.
Paramètres d'autorisation	Saisissez les paramètres du point de terminaison d'autorisation.
Code d'autorisation	Saisissez le code d'autorisation : **Automatique** - le code sera récupéré automatiquement via une page de redirection **Manuel** - saisissez le code manuellement si la récupération automatique échoue
Point de terminaison du jeton	Saisissez l'URL du serveur d'autorisation OAuth pour échanger le code d'autorisation contre des jetons d'accès et d'actualisation.
Paramètres du jeton	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 fournir les valeurs des paramètres *Point de terminaison de redirection*, *ID client* et *Secret client*. Zabbix renseignera automatiquement les autres valeurs requises (voir Valeurs par défaut des URL OAuth par fournisseur).

Récupération des jetons d'accès et d'actualisation OAuth

Après avoir soumis les paramètres OAuth :

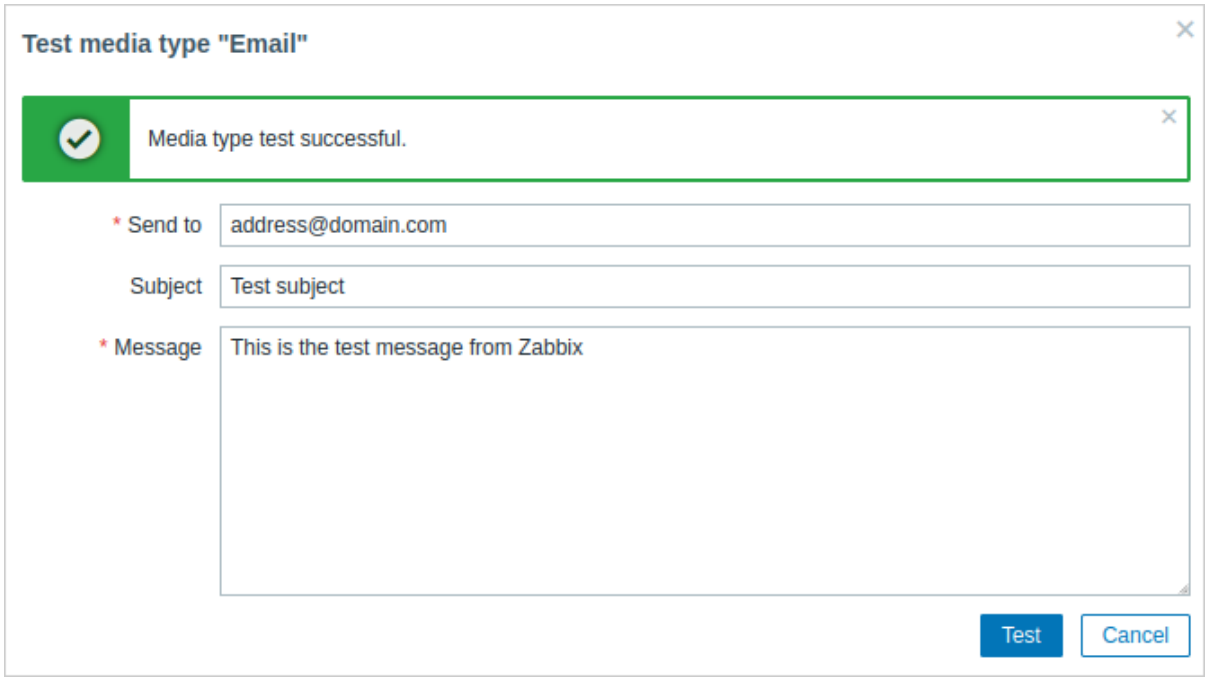
1. Une fenêtre contextuelle du navigateur s'ouvre et redirige l'utilisateur vers le *point de terminaison d'autorisation*.
2. L'utilisateur autorise Zabbix dans le service OAuth.
3. Le service OAuth redirige l'utilisateur vers l'action du frontend Zabbix `oauth.authorize` avec le code d'autorisation 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 tester si 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 :



Support utilisateur

Une fois le type de média e-mail configuré, accédez à la section *Utilisateurs* > *Utilisateurs* et modifiez le profil de l'utilisateur afin de lui attribuer le média e-mail. Les étapes de configuration du support utilisateur, communes à tous les types de médias, sont décrites sur la page Types de médias.

1 Types de média Gmail/Office365 automatisés

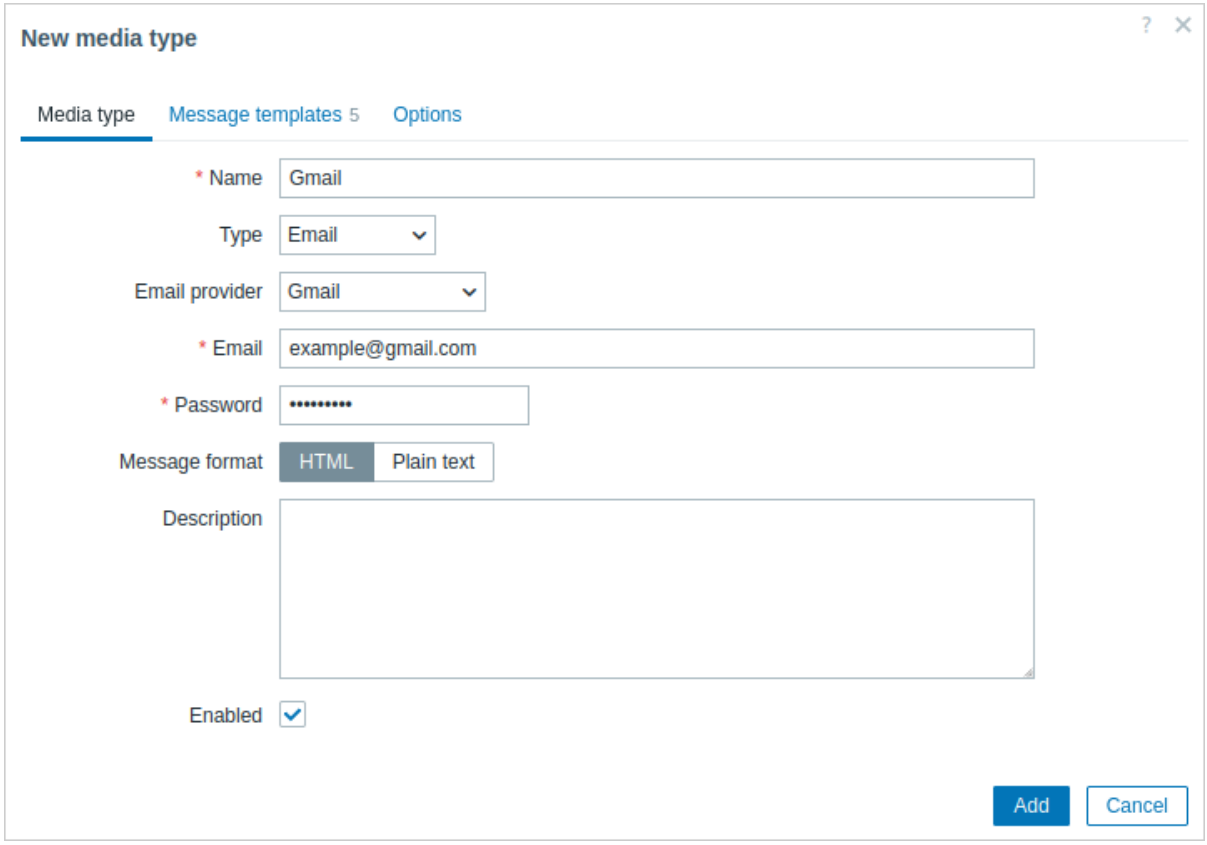
Vue d'ensemble

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 e-mail permet de sélectionner des options préconfigurées pour Gmail et Office 365.

Lors de la sélection des options liées à Gmail/Office365, il suffit de fournir l'adresse e-mail de l'expéditeur et le mot de passe 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*, *SMTP helo* et *Sécurité de la connexion*. En raison de cette automatisation, ces champs ne sont même pas affichés ; il est toutefois possible de voir les détails du serveur SMTP et de 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 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/authorize`	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.

2 SMS

Aperçu

Zabbix prend en charge l'envoi de SMS via un modem GSM connecté au port série du serveur Zabbix.

Assurez-vous que :

- La vitesse du périphérique série (normalement /dev/ttyS0 sous Linux) corresponde 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 sur le périphérique série. Exécutez la commande ls --l /dev/ttyS0 pour afficher les autorisations en cours du périphérique série.
- Le modem GSM a un code PIN entré et qu'il le conserve après la réinitialisation de l'alimentation. Vous pouvez également désactiver le code PIN sur la carte SIM. Vous pouvez entrer un code PIN en émettant la commande AT+CPIN = "NNNN" (NNNN est votre numéro PIN, les guillemets doivent être présents) dans un logiciel de terminal, tel que Unix minicom ou Windows HyperTerminal.

Zabbix a été testé avec ces modems GSM :

- Siemens MC35
- Teltonika ModemCOM / G10

Pour configurer les SMS en tant que canal de remise des messages, vous devez également configurer les SMS en tant que type de média et entrer les numéros de téléphone respectifs des utilisateurs.

Configuration

Pour configurer le SMS comme type de média :

- Allez dans *Alertes* → *Types de média*
- Cliquez sur *Créer un type de média* (ou cliquez sur *SMS* dans la liste des types de média prédéfinis).

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 serveur SMSDevices (s'il est spécifié).

Consultez les paramètres communs des types de média pour plus de détails 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.

Support utilisateur

Une fois le type de média SMS configuré, accédez à la section *Utilisateurs* → *Utilisateurs* et modifiez le profil de l'utilisateur afin de lui attribuer le média SMS. Les étapes de configuration du support utilisateur, communes à tous les types de médias, sont décrites sur la page Types de médias.

3 Scripts d'alerte personnalisés

Aperçu

Si vous n'êtes pas satisfait des types de média existants pour l'envoi d'alertes, il existe une autre façon de procéder. 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 être placés 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 d'exécution de commande. Dans ce cas, Zabbix tentera de répéter l'exécution ayant échoué.

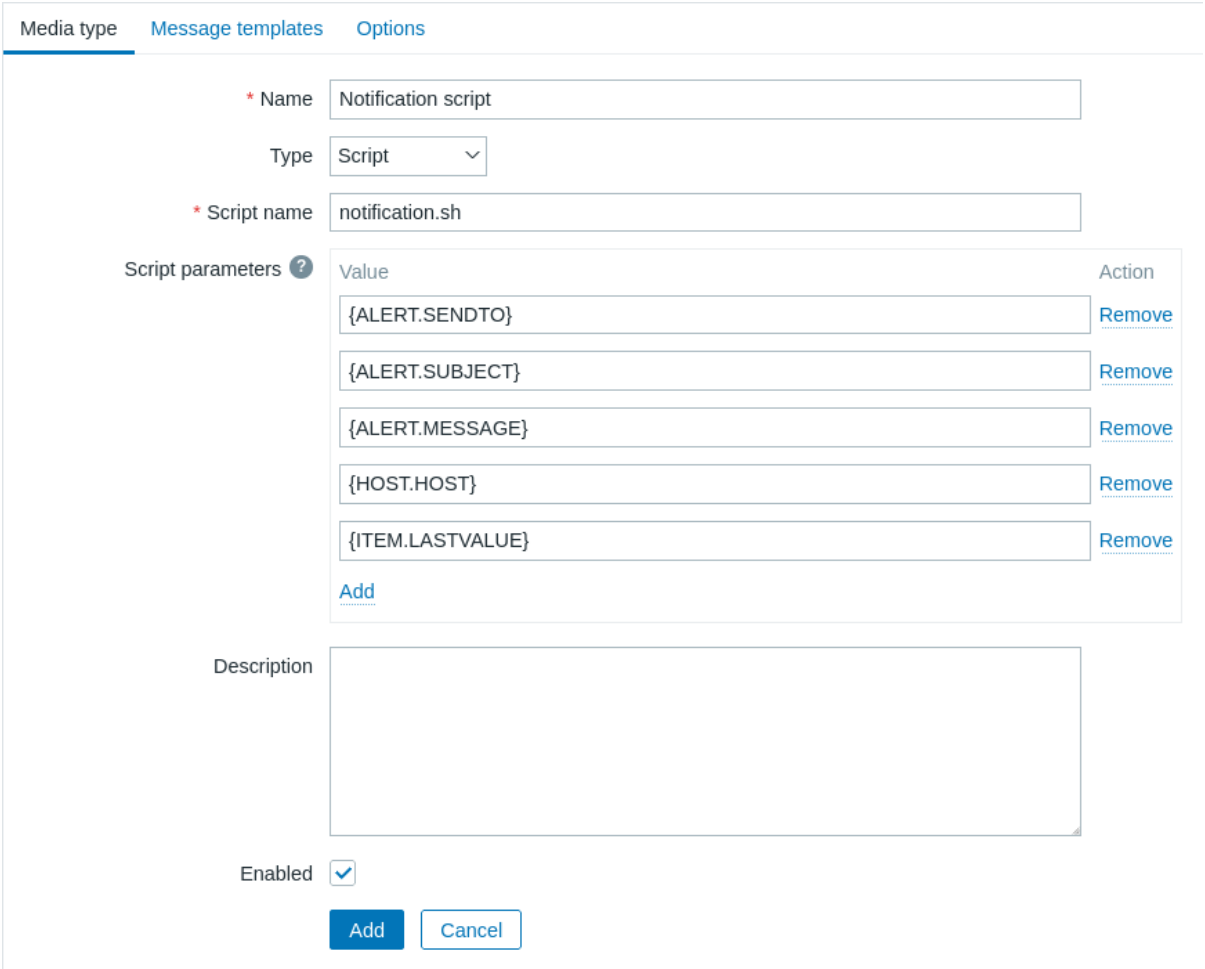
Les variables d'environnement ne sont ni conservées ni créées pour le script, elles doivent donc être gérées explicitement.

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*.

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 script :

Parameter	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.

Parameter	Description
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 où ils sont définis. Les paramètres du script prennent en charge les macros {ALERT.SENDTO}, {ALERT.SUBJECT}, {ALERT.MESSAGE}, ainsi que toutes les macros prises en charge dans les notifications, ainsi que les macros utilisateur.

Consultez les paramètres communs des types de média pour plus de détails sur la configuration des messages par défaut et des options de traitement des alertes.

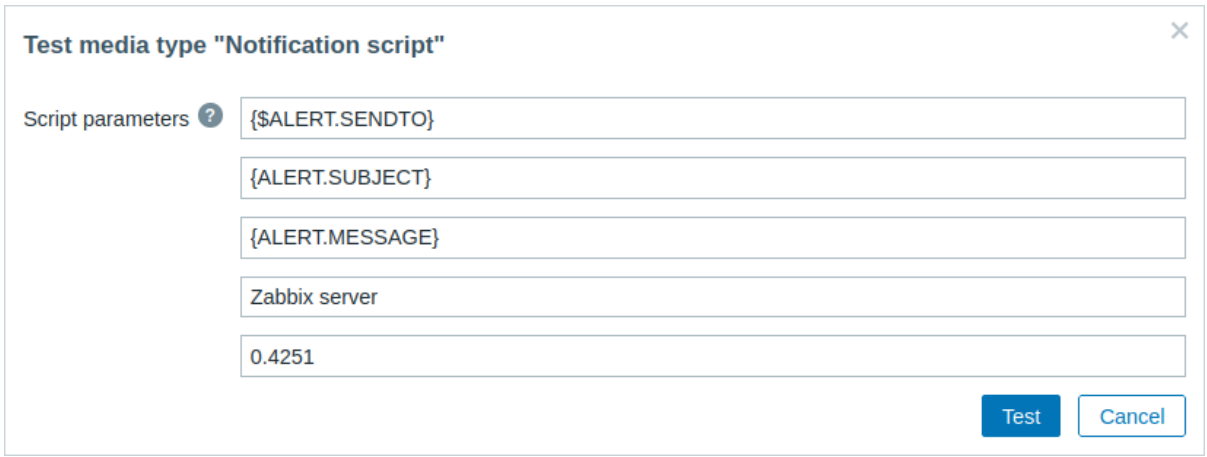
Warning:
Même si un script d'alerte n'utilise pas les messages par défaut, les modèles de message pour les types d'opération utilisés par ce type de média doivent tout de même être définis. Sinon, une notification ne sera pas envoyée.

Attention:
Si plusieurs types de média 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 seront résolues à leurs valeurs, mais les macros liées aux événements (par exemple, {HOST.HOST}, {ITEM.LASTVALUE}, etc.) ne seront pas résolues, car lors du test, il n'existe aucun événement associé dont les détails pourraient être récupérés. Notez également que les macros présentes 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.

Support utilisateur

Une fois le type de média configuré, accédez à la section *Utilisateurs* → *Utilisateurs* et modifiez un profil utilisateur en attribuant ce type de média à l'utilisateur. Les étapes de configuration du support utilisateur, communes à tous les types de médias, sont décrites sur la page Types de médias.

Notez que lors de la définition du support utilisateur, le champ *Envoyer à* 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.

4 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 des notifications Zabbix vers :

- brevis.one
- Discord
- Event-Driven Ansible
- Express.ms messenger
- GitHub
- GLPi
- iLert
- iTop
- Jira
- Jira Service Management
- ManageEngine ServiceDesk
- Mantis Bug Tracker
- Mattermost
- MS Teams
- 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 helpdesk Spiceworks (par exemple, help@zabbix.on.spiceworks.com) dans les paramètres de profil d'un utilisateur Zabbix désigné.

Configuration

Pour commencer à utiliser une intégration webhook :

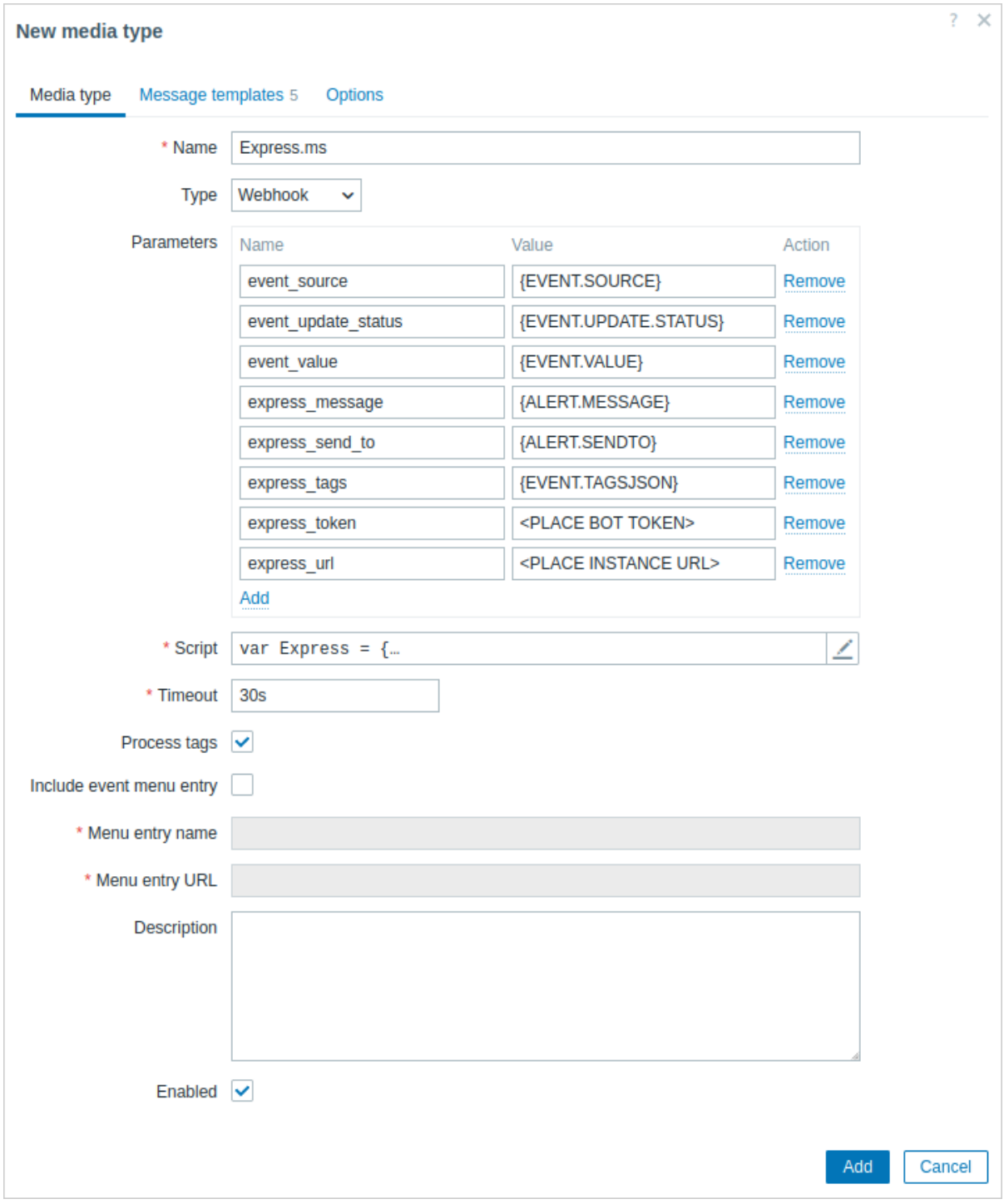
1. Localisez le fichier .yaml requis dans le répertoire `templates/media` de la version téléchargée de Zabbix ou téléchargez-le depuis le dépôt git de Zabbix.
2. Importez le fichier dans votre installation Zabbix. Le webhook apparaîtra dans la liste des types de média.
3. Configurez le webhook conformément 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*.

L'onglet **Type de média** contient divers attributs spécifiques à ce type de média :



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

Les paramètres suivants sont spécifiques au type de média webhook :

Parameter	Description
Parameters	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 communes 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 du webhook prennent en charge les macros utilisateur, toutes les macros prises en charge dans les notifications de problème et, en plus, les macros {ALERT.SENDTO}, {ALERT.SUBJECT} et {ALERT.MESSAGE}. Si vous spécifiez un proxy HTTP, le champ prend en charge la même fonctionnalité que le champ HTTP proxy dans la configuration des éléments. La chaîne du proxy peut être préfixée par `[scheme]://` pour spécifier le type de proxy utilisé (par exemple, https, socks4, socks5 ; voir 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 à côté. Ce code exécutera l'opération du webhook. Le script est un code de 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 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 statut OK avec une liste facultative de tags et de valeurs de tags (voir l'option *Process 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 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 ; par conséquent, les tags renvoyés par un script webhook ne sont ajoutés qu'après la création initiale de l'alerte et ne seront pas présents dans les macros `{EVENT.TAGS}` et `{EVENT.RECOVERY.TAGS}` du message initial de problème ou du message de rétablissement immédiat. *Remarque* : il est recommandé d'utiliser des variables locales (par ex. `var local = 1`) plutôt qu'une variable globale (par ex. `global = 1`) afin de garantir que chaque script fonctionne sur ses propres données et d'éviter les collisions entre appels simultanés (voir les problèmes connus). Voir aussi : Directives de développement des webhooks, Exemples de scripts webhook, Objets JavaScript supplémentaires.
Timeout	Délai d'expiration de l'exécution JavaScript (1-60s, 30s par défaut). Les suffixes de temps sont pris en charge, par ex., 30s, 1m.
Process tags	Cochez la case pour traiter les valeurs des propriétés JSON renvoyées comme des tags. Ces tags sont ajoutés à tous les tags de problème existants. Notez que lors de l'utilisation de tags webhook, le webhook doit renvoyer un objet JSON contenant au minimum un objet tags vide : `var result = {tags: {}};` Exemples de tags pouvant être renvoyés : *jira-id:prod-1234*, *responsible:John Smith*, *processed:<no value>*
Include event menu entry	Cochez la case pour inclure une entrée dans le menu d'événement pointant vers un ticket externe créé. Une entrée sera incluse pour chaque webhook activé ayant cette case cochée. Notez que si les paramètres *Menu entry name* et *Menu entry URL* 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 a ces tags définis). Si cette option est cochée, 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.

Parameter	Description
Menu entry name	Spécifiez le nom de l'entrée de menu. La macro {EVENT.TAGS.<tag name>} est prise en charge. Ce champ n'est obligatoire que si *Include event menu entry* est coché.
Menu entry URL	Spécifiez l'URL sous-jacente de l'entrée de menu. La macro {EVENT.TAGS.<tag name>} est prise en charge. Ce champ n'est obligatoire que si *Include event menu entry* est coché.

Consultez les paramètres communs des types de média pour plus de détails sur la configuration des messages par défaut et des options de traitement des alertes.

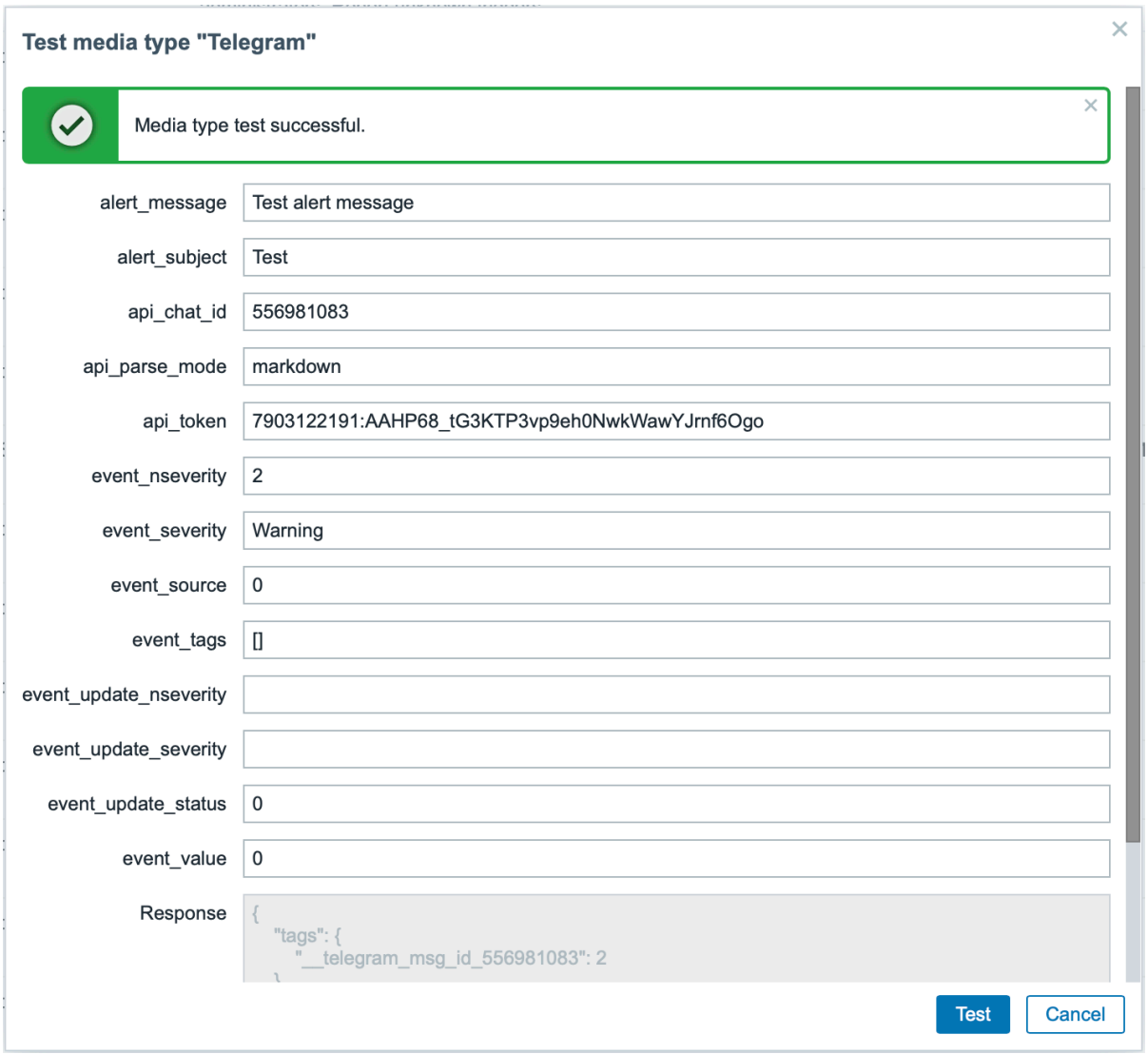
Warning:
Même si un webhook n'utilise pas les messages par défaut, des modèles de message pour les types d'opération utilisés par ce webhook doivent tout de même être définis.

Test

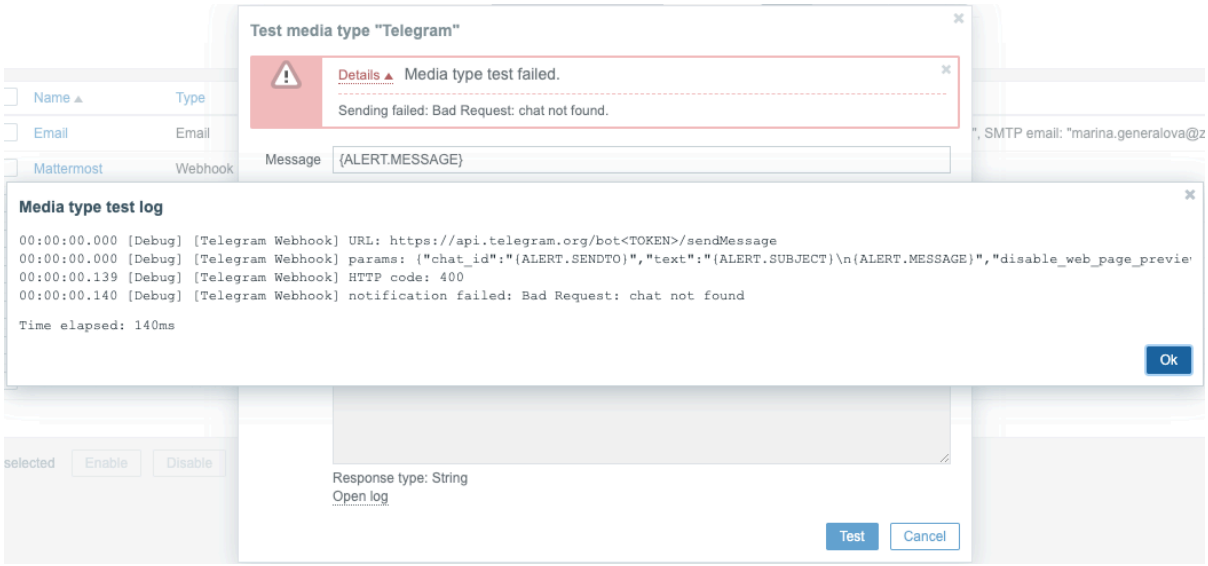
Pour tester un type de média webhook configuré :

1. Localisez le webhook concerné dans la liste des types de média.
2. Cliquez sur *Test* dans la dernière colonne de la liste (une fenêtre de test s'ouvrira).
3. Modifiez les valeurs des paramètres du webhook selon les besoins. Remplacez les macros par des valeurs d'exemple ; sinon, les macros ne seront pas résolues et le test échouera.
4. Cliquez sur *Test*.

Le remplacement ou la suppression de valeurs dans la fenêtre de test n'affecte que la procédure de test ; les valeurs réelles des attributs du webhook resteront inchangées.



Pour afficher les entrées du journal de test du type de média sans quitter la fenêtre de test, cliquez sur *Open log* (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.

Support utilisateur

Une fois le type de média configuré, accédez à la section *Users* > *Users* et assignez le média webhook à un utilisateur existant ou créez un nouvel utilisateur pour représenter le webhook. Les étapes de configuration du média 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 tags pour stocker l'ID du ticket\message, évitez d'assigner le même webhook comme média à différents utilisateurs, car cela peut provoquer des erreurs de webhook (cela s'applique à la majorité des webhooks qui utilisent l'option *Include event menu entry*). Dans ce cas, la bonne pratique consiste à créer un utilisateur dédié pour représenter le webhook :

1. Après avoir configuré le type de média webhook, accédez à la section *Users* > *Users* 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 être laissés à leurs valeurs par défaut, car cet utilisateur ne se connectera pas à Zabbix.
2. Dans le profil utilisateur, accédez à l'onglet *Media* et ajoutez un webhook avec les informations de contact requises. Si le webhook n'utilise pas de champ *Send to*, saisissez n'importe quelle combinaison de caractères pris en charge afin de contourner les exigences de validation.
3. Accordez à cet utilisateur au moins les autorisations de lecture sur tous les hôtes pour lesquels il doit envoyer des alertes.

Lors de la configuration de l'action d'alerte, ajoutez cet utilisateur dans le champ *Send to users* des détails de l'opération - cela indiquera à Zabbix d'utiliser 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 points 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 même 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 même 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.

1 Exemples de scripts webhook

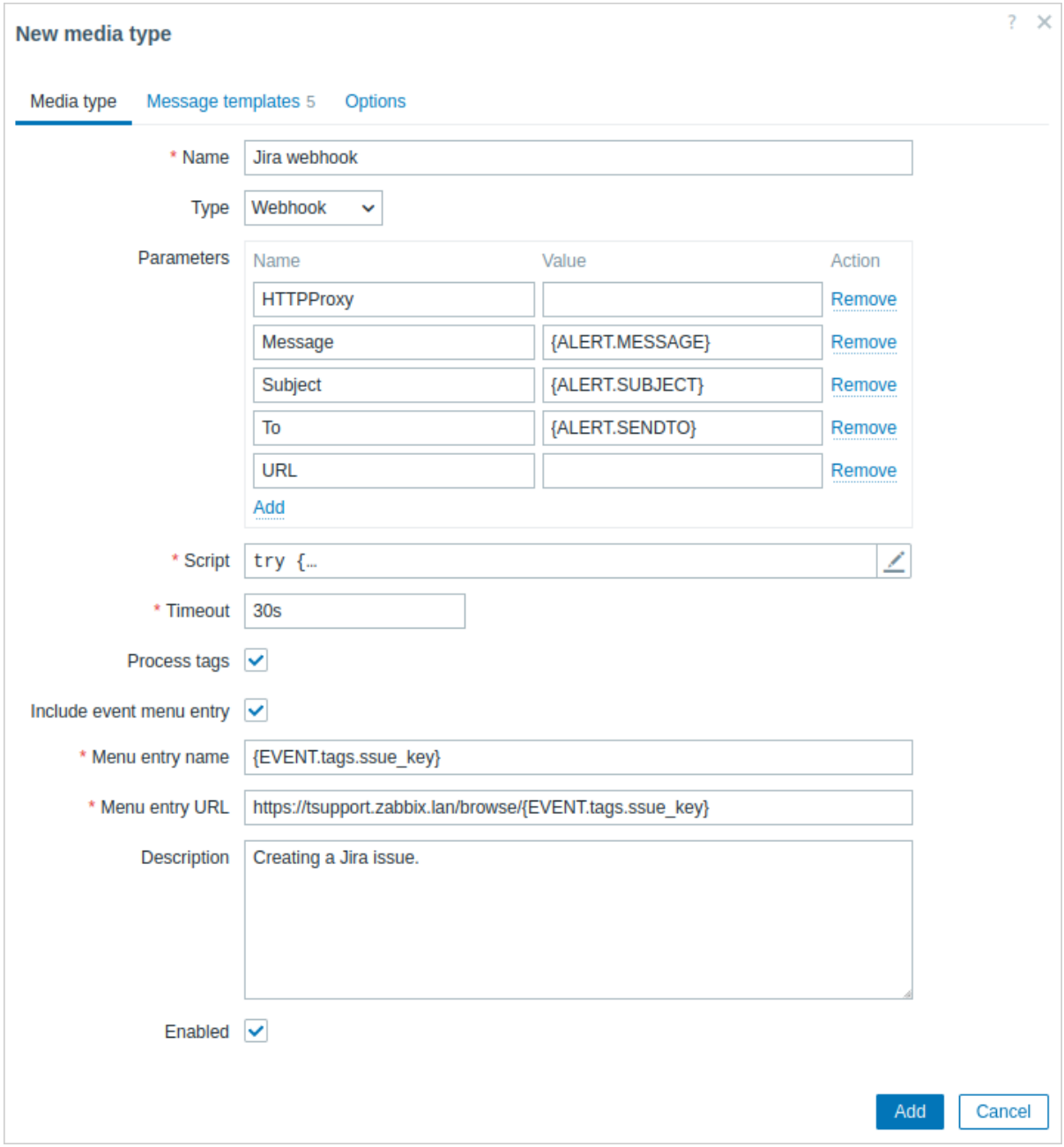
Aperçu

Bien que Zabbix propose un grand nombre d'intégrations webhook disponibles prêtes à l'emploi, vous pouvez préférer créer vos propres webhooks. Cette section fournit des exemples de scripts webhook personnalisés (utilisés dans le paramètre *Script*). Consultez la section webhook pour la description des autres paramètres de webhook.

Attention:
N'utilisez pas d'affectations non déclarées dans le prétraitement JavaScript. Utilisez `var` pour déclarer les variables locales.

Les événements de rétablissement (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 exige que les tags produits par un webhook soient présents dans la notification initiale, récupérez les tags d'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 effectuez 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();
    }

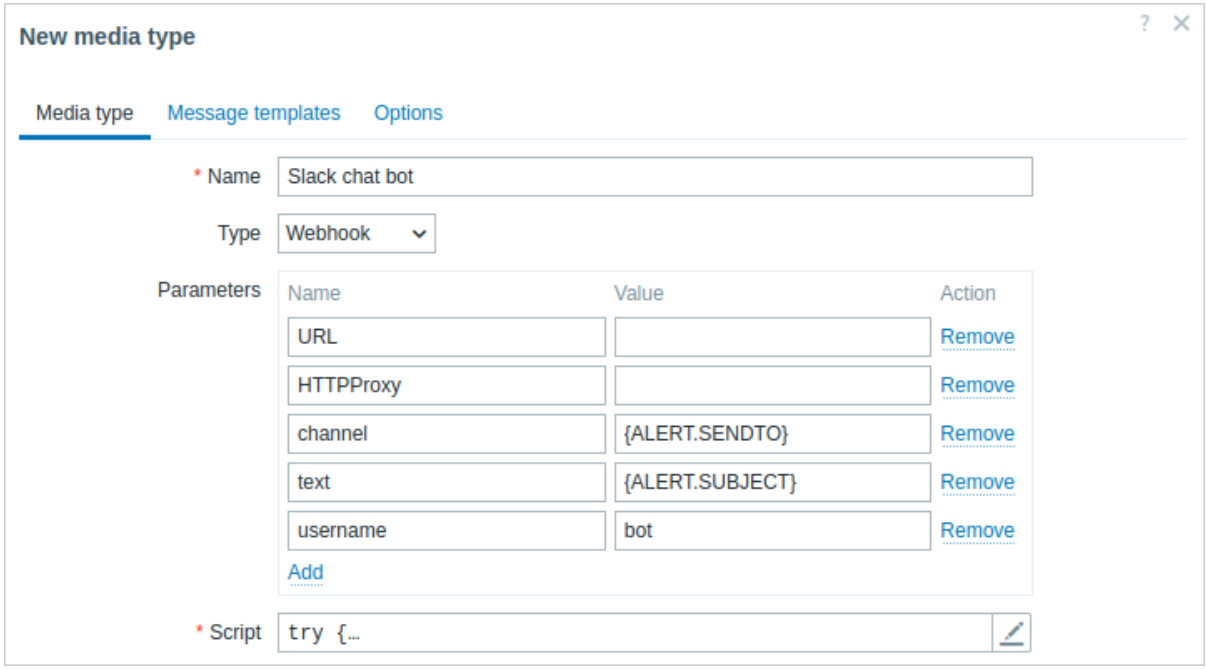
    resp = JSON.parse(resp);
    result.tags.issue_id = resp.id;
    result.tags.issue_key = resp.key;

    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 à 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.hook_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;
}
```

2 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 :

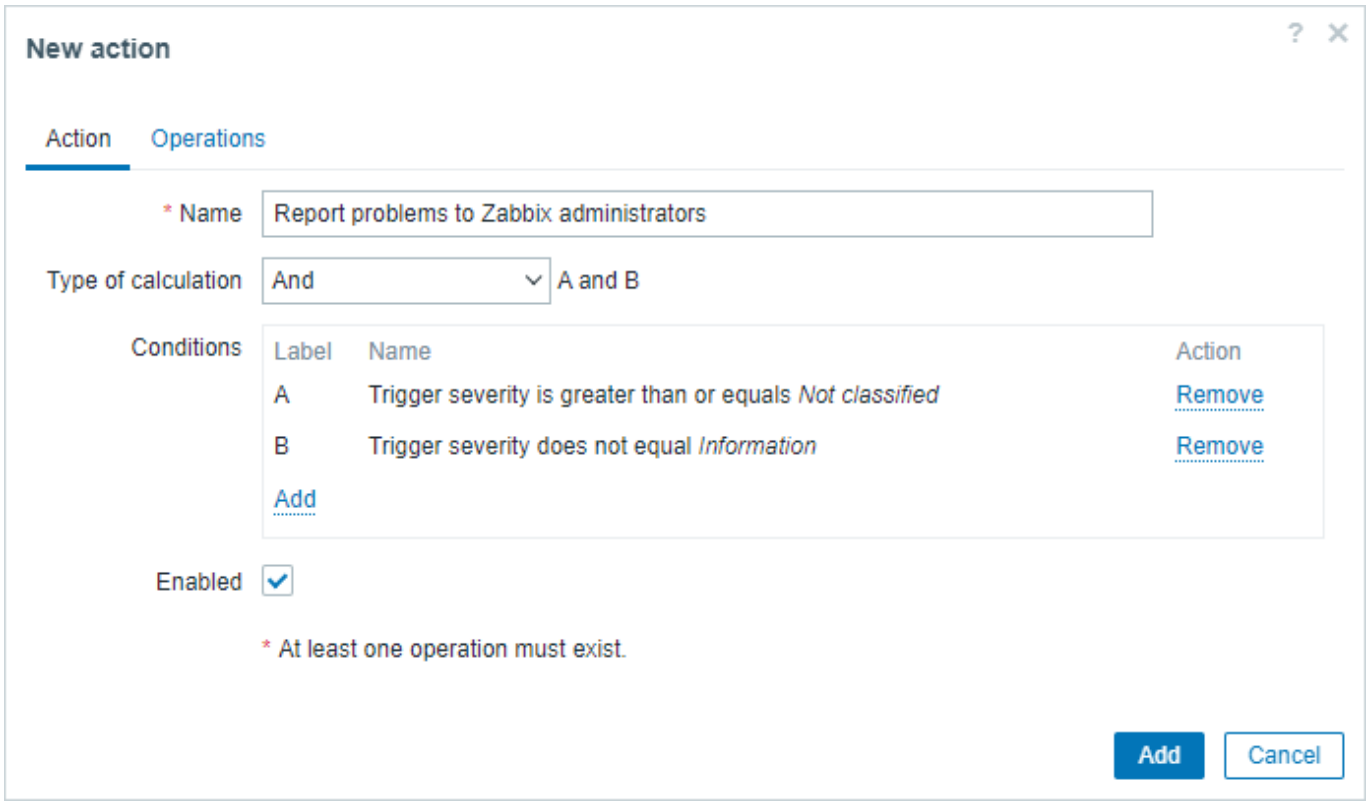
- L'accès des utilisateurs aux actions de service dépend des droits d'accès aux services accordés par le rôle de l'utilisateur
- Les actions de service prennent en charge un ensemble différent de conditions

Configuration d'une action

Pour configurer une action, procédez comme suit :

- Accédez à *Alertes* > *Actions* et sélectionnez le type d'action requis dans le sous-menu (vous pourrez passer à un autre type plus tard à l'aide de la liste déroulante du titre).
- Cliquez sur *Créer une action*.
- Nommez l'action.
- Choisissez les conditions selon lesquelles les opérations sont exécutées.
- Choisissez les opérations à exécuter.

Attributs généraux de l'action :



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

Paramètre	Description
Nom	Nom unique de l'action.

Paramètre	Description
Type de calcul	Sélectionnez l'option d'évaluation des conditions d'action (lorsqu'il y a plus d'une condition) : **Et** - toutes les conditions doivent être remplies. **Ou** - il suffit qu'une seule condition soit remplie. **Et/Ou** - combinaison des deux : ET avec des types de condition différents et OU avec le même type de condition. **Expression personnalisée** - formule de calcul définie par l'utilisateur pour évaluer les conditions d'action.
Conditions	Liste des conditions d'action. Cliquez sur *Ajouter* pour ajouter une nouvelle condition. Si aucune condition n'est configurée, l'action sera exécutée pour chaque événement correspondant au type d'action en cours de configuration.
Activé	Cochez la case pour activer l'action. Sinon, elle sera désactivée.

1 Conditions

Aperçu

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.

Type de condition	Opérateurs pris en charge	Description
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	égal à différent de	Spécifiez un service ou un service à exclure. **égal à** - l'événement appartient à ce service. **différent de** - 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 **différent de**.
Nom du service	contient ne contient pas	Spécifiez une chaîne dans le nom du service ou une chaîne à exclure. **contient** - l'événement est généré par un service dont le nom contient cette chaîne. **ne contient pas** - cette chaîne ne peut pas être trouvée dans le nom du service.
Nom du tag de service	égal à différent de contient ne contient pas	Spécifiez un tag d'événement ou un tag d'événement à exclure. Les tags d'événement de service peuvent être définis dans la section *Tags* de la configuration du service. **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.

Type de condition	Opérateurs pris en charge	Description
Valeur du tag de service	égal à différent de contient ne contient pas	Spécifiez une combinaison de tag d'événement et de valeur, ou une combinaison de tag et de valeur à exclure. Les tags d'événement de service peuvent être définis dans la section *Tags* de la configuration du service. **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.

Attention:
Assurez-vous de définir des modèles de message pour les actions de service dans le menu *Alertes* → *Types de médias*. 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
IP de l'hôte	égal à différent de	Spécifiez une plage d'adresses IP ou une plage à exclure pour un hôte découvert. **égal à** - l'IP de l'hôte se trouve dans la plage. **différent de** - l'IP de l'hôte ne se trouve pas dans la plage. Les formats suivants sont possibles : 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 service	égal à différent de	Spécifiez un type de service d'un service découvert ou un type de service à exclure. **égal à** - correspond au service découvert. **différent de** - 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	égal à différent de	Spécifiez une plage de ports TCP d'un service découvert ou une plage à exclure. **égal à** - le port du service se trouve dans la plage. **différent de** - le port du service ne se trouve pas dans la plage.
Règle de découverte	égal à différent de	Spécifiez une règle de découverte ou une règle de découverte à exclure. **égal à** - utilise cette règle de découverte. **différent de** - utilise toute autre règle de découverte, sauf celle-ci.
Vérification de découverte	égal à différent de	Spécifiez une vérification de découverte ou une vérification de découverte à exclure. **égal à** - utilise cette vérification de découverte. **différent de** - utilise toute autre vérification de découverte, sauf celle-ci.
Objet de découverte	égal à	Spécifiez l'objet découvert. **égal à** - égal à l'objet découvert (un périphérique ou un service).

Type de condition	Opérateurs pris en charge	Description
Statut de découverte	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	égal à différent de est supérieur ou égal à est inférieur ou égal à contient ne contient pas	Spécifiez 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). **égal à** - égal à la valeur. **différent de** - 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 fourni sous forme de chaîne. **ne contient pas** - ne contient pas la sous-chaîne. Le paramètre est fourni sous forme de chaîne.
Proxy	égal à différent de	Spécifiez un proxy ou un proxy à exclure. **égal à** - utilise ce proxy. **différent de** - utilise tout autre proxy sauf celui-ci.

Note:
Les vérifications de service dans une règle de découverte qui donnent lieu à des événements de découverte ne s'exécutent pas simultanément. Par conséquent, si **plusieurs** valeurs sont configurées pour les conditions `Service type`, `Service port` ou `Received value` 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 de l'agent actif :

Type de condition	Opérateurs pris en charge	Description
Métadonnées de l'hôte	contient ne contient pas correspond à ne correspond pas	Spécifiez 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 spécifiées 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.

Type de condition	Opérateurs pris en charge	Description
Nom de l'hôte	contient ne contient pas correspond à ne correspond pas	Spécifiez un nom d'hôte ou un nom d'hôte à exclure. **contient** - le nom d'hôte contient la chaîne. **ne contient pas** - le nom d'hôte ne contient pas la chaîne. **correspond à** - le nom d'hôte correspond à l'expression régulière. **ne correspond pas à** - le nom d'hôte ne correspond pas à l'expression régulière.
Proxy	égal n'est pas égal	Spécifiez un proxy ou un proxy à exclure. **égal** - en utilisant ce proxy. **n'est pas égal** - en utilisant un autre proxy que 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
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 :

- **Et** - 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.

- **Ou** - il suffit qu'une seule condition soit remplie
- **Et/Ou** - combinaison des deux : AND avec des types de condition différents 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 de l'événement contient 'Database is down'
Nom de l'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 de l'événement contient 'Database is down' **or** Nom de l'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.

2 Opérations

Vue d'ensemble

Vous pouvez définir les opérations suivantes pour tous les événements :

- Envoyer un message
- Exécuter une commande à distance

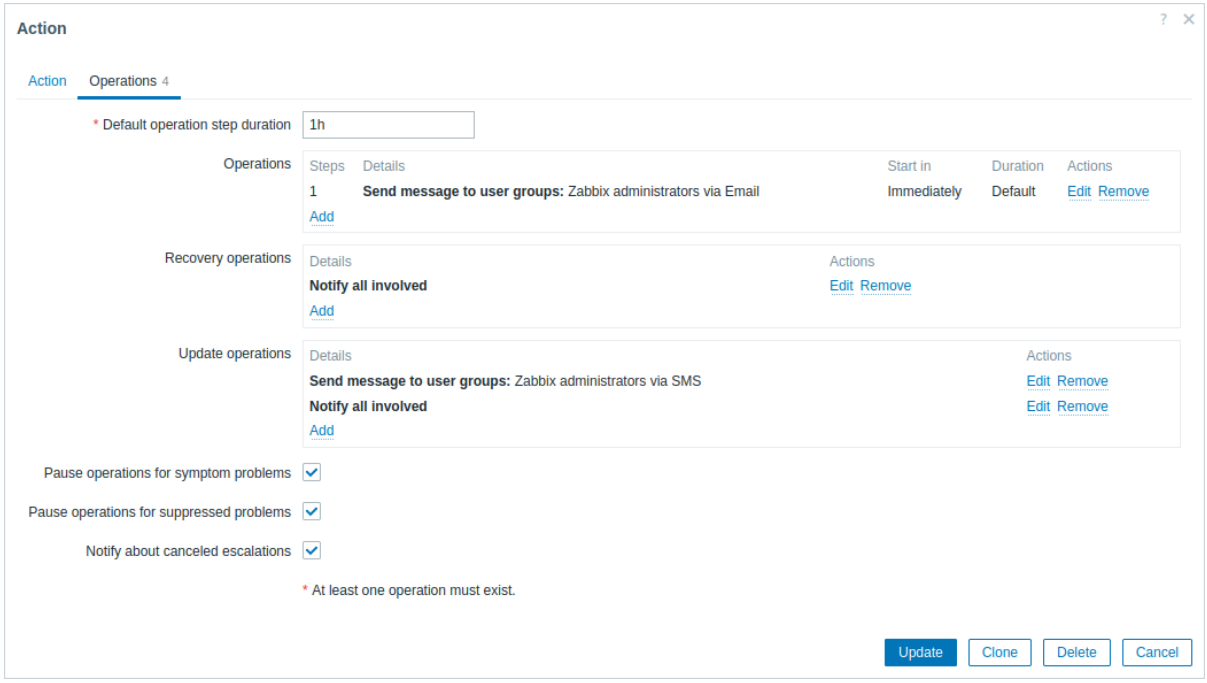
> **Attention:**
> Le serveur Zabbix ne crée pas d'alertes si l'accès à l'hôte est explicitement « refusé » pour l'utilisateur défini comme destinataire de l'opération d'action ou si aucun droit n'est défini pour cet utilisateur sur l'hôte.

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
- Délier un modèle
- Définir le mode d'inventaire de l'hôte

Configuration d'une opération

Pour configurer une opération, accédez à l'onglet *Operations* dans la configuration de l'action.



Attributs généraux de l'opération :

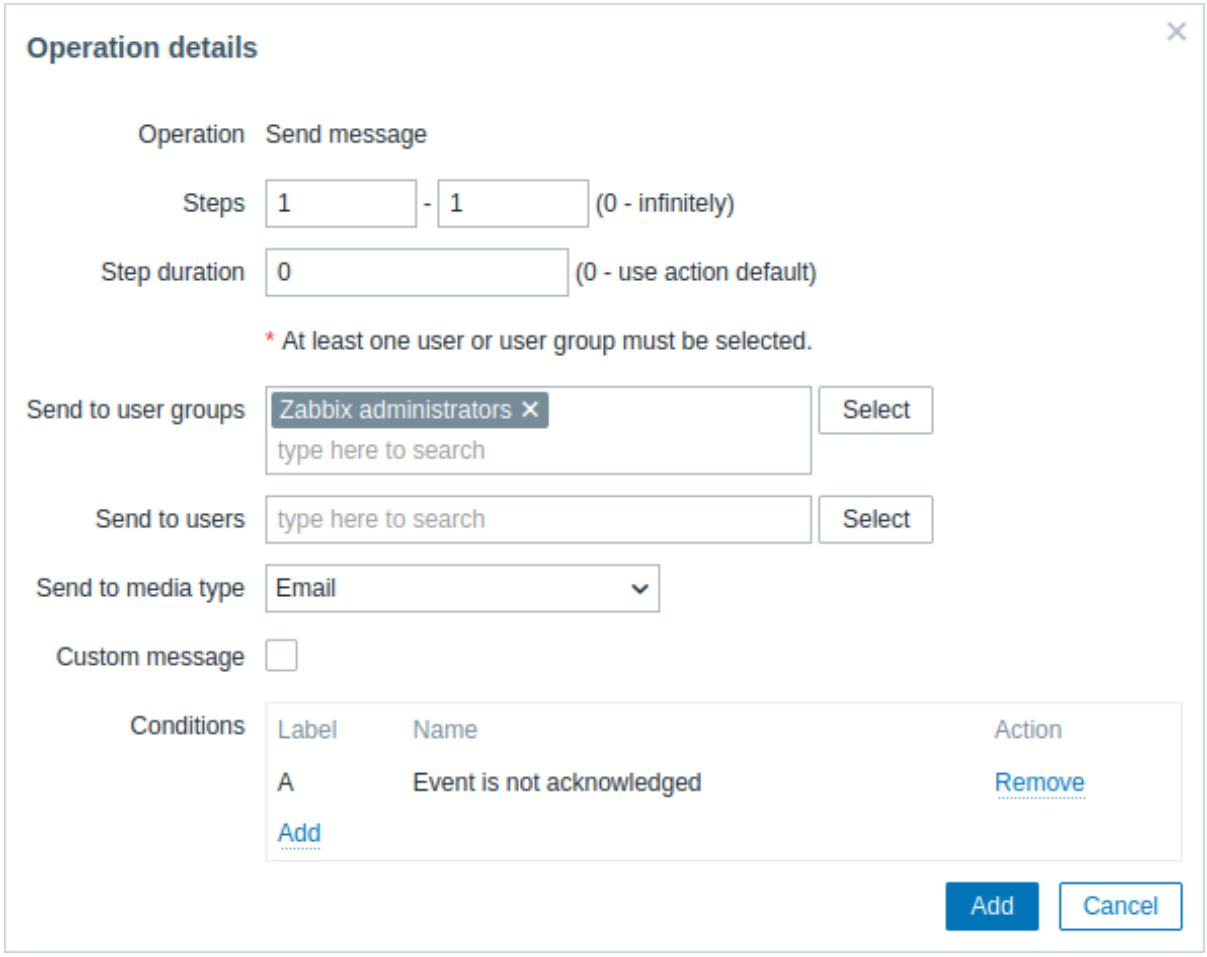
Paramètre	Description
Default operation step duration	Durée d'une étape d'opération par défaut (de 60 secondes à 1 semaine). Par exemple, une durée d'étape d'une heure signifie que si une opération est exécutée, une heure s'écoulera avant l'étape suivante. Les suffixes de temps sont pris en charge, par exemple 60s, 1m, 2h, 1d. Les macros utilisateur sont prises en charge.
Operations	Les opérations d'action (le cas échéant) sont affichées avec les détails suivants : **Steps** - étape(s) d'escalade auxquelles l'opération est affectée. **Details** - type d'opération et son destinataire/cible. La liste des opérations affiche également le type de média utilisé (e-mail, SMS ou script), ainsi que le nom et le prénom (entre parenthèses après le nom d'utilisateur) d'un destinataire de notification. **Start in** - délai après un événement au bout duquel l'opération est exécutée. **Duration (sec)** - la durée de l'étape est affichée. *Default* est affiché si l'étape utilise la durée par défaut, et une durée est affichée si une durée personnalisée est utilisée. **Actions** - des liens pour modifier et supprimer une opération sont affichés.
Recovery operations	Les opérations d'action (le cas échéant) sont affichées avec les détails suivants : **Details** - type d'opération et son destinataire/cible. La liste des opérations affiche également le type de média utilisé (e-mail, SMS ou script), ainsi que le nom et le prénom (entre parenthèses après le nom d'utilisateur) d'un destinataire de notification. **Actions** - des liens pour modifier et supprimer une opération sont affichés.
Update operations	Les opérations d'action (le cas échéant) sont affichées avec les détails suivants : **Details** - type d'opération et son destinataire/cible. La liste des opérations affiche également le type de média utilisé (e-mail, SMS ou script), ainsi que le nom et le prénom (entre parenthèses après le nom d'utilisateur) d'un destinataire de notification. **Actions** - des liens pour modifier et supprimer une opération sont affichés.
Pause operations for symptom problems	Cochez cette case pour mettre en pause les opérations (après la première opération) pour les problèmes symptômes. Notez que ce paramètre n'affecte que les escalades de problèmes ; les opérations de récupération et de mise à jour ne seront pas affectées. Cette option est disponible uniquement pour les *Trigger actions*.
Pause operations for suppressed problems	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 des événements survenus pendant la maintenance. Notez que ce paramètre n'affecte que les escalades de problèmes ; les opérations de récupération 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 *Service actions*.

Paramètre	Description
Notify about canceled escalations	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 *Operations*. 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 commande distante>** - exécuter une commande distante. Les commandes peuvent être exécutées si elles ont été définies au préalable dans les scripts globaux avec *Action operation* sélectionné comme portée. D'autres opérations sont disponibles pour les événements basés sur la découverte et l'auto-enregistrement (voir ci-dessus).
Étapes	Sélectionnez la ou les étapes auxquelles attribuer l'opération dans un calendrier d'escalation : **De** - exécuter 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 de temps sont pris en charge, par ex. 60s, 1m, 2h, 1d. Les macros utilisateur sont prises en charge. Plusieurs opérations peuvent être attribuées à la même étape. Si ces opérations ont des durées d'étape différentes définies, la plus courte est prise en compte et appliquée à l'étape.

Paramètre	Description
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 au minimum des permissions « lecture » sur l'hôte afin d'être notifié.
Envoyer aux utilisateurs	Sélectionnez les utilisateurs auxquels envoyer le message. L'utilisateur doit disposer au minimum des permissions « lecture » sur l'hôte afin d'être notifié.
Envoyer au type de média	Envoyer le message à tous les types de média disponibles (configurés et activés) ou à un seul type spécifique.
Message personnalisé	Si cette option est sélectionnée, le message personnalisé peut être configuré. Pour les notifications concernant des événements internes via des webhooks, le message personnalisé est obligatoire.
Sujet	Sujet du message personnalisé. Le sujet peut contenir des macros. Il est limité à 255 caractères.
Message	Le 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 Envoi d'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 l'hôte du déclencheur qui a provoqué l'événement de problème. Cette option ne fonctionnera pas si plusieurs hôtes sont présents dans le déclencheur. **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 ex. à partir de plusieurs groupes d'hôtes ; individuellement et à partir d'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 a seulement pour effet d'exécuter le script plus de fois sur le serveur. Notez que, pour les scripts globaux, la sélection de la cible dépend également du paramètre *Host group* dans la configuration du script global. L'option *Liste des cibles* n'est pas disponible pour les *Actions de service* car, dans ce cas, les commandes distantes sont toujours exécutées sur le serveur Zabbix.

Paramètre	Description
Conditions	Condition d'exécution de l'opération : **L'événement n'est pas acquitté** - uniquement lorsque l'événement n'est pas acquitté. **L'événement est acquitté** - uniquement lorsque l'événement est acquitté. L'option *Conditions* n'est disponible que pour les *Actions de déclencheur*.

Une fois terminé, cliquez sur *Ajouter* pour ajouter l'opération à la liste des *Opérations*.

1 Envoi d'un 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 depuis Zabbix, vous devez :

- définir le média auquel envoyer un message

Si l'opération a lieu en dehors de la période When active définie pour le média sélectionné dans la configuration utilisateur, le message ne sera pas envoyé.

La sévérité de déclencheur par défaut (« Not classified ») **doit être** cochée dans la configuration du média utilisateur si vous souhaitez recevoir des notifications pour des événements autres que les déclencheurs, tels que la découverte, l'autoréenregistrement de l'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 qui disposent au minimum des permissions « lecture » sur l'hôte ayant généré 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 « SMTP/MIME email », car Zabbix envoie des données UTF-8 (si l'objet contient uniquement des caractères ASCII, il n'est pas encodé en UTF-8). L'objet et le corps du message sont encodés en base64 afin de respecter le format standard « SMTP/MIME email ».

La limite de message après expansion de toutes les macros est identique à la limite de message pour les commandes distantes.

Suivi des messages

Vous pouvez consulter l'état des messages envoyés dans *Surveillance* → *Problèmes*.

Dans la colonne *Actions*, vous pouvez voir des informations récapitulatives sur les actions effectuées. Les nombres verts représentent les messages envoyés, les rouges les messages ayant échoué. *En cours* indique qu'une action est initiée. *Échec* signifie 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 *Rapports* → *Journal des actions*, vous verrez les détails de toutes les actions effectuées pour les événements pour lesquels une action est configurée.

2 Commandes à distance

Aperçu

Avec les commandes à distance, vous pouvez définir qu'une certaine commande prédéfinie est automatiquement exécutée sur l'hôte surveillé sous certaines conditions.

Ainsi, les commandes à distance constituent un mécanisme puissant pour la supervision proactive intelligente.

Dans les utilisations les plus évidentes de cette fonctionnalité, vous pouvez essayer de :

- Redémarrer automatiquement certaines applications (serveur Web, middleware, CRM) si elles ne répondent pas
- Utiliser la commande 'reboot' IPMI pour redémarrer un serveur distant s'il ne répond pas aux demandes
- Libérer automatiquement de l'espace disque (suppression des anciens fichiers, nettoyage de /tmp) si vous manquez d'espace disque
- Migrer une VM d'un boîtier physique à un autre en fonction de la charge du processeur
- Ajouter de nouveaux nœuds à un environnement cloud en cas de ressources CPU insuffisantes (disque, mémoire, etc.)

La configuration d'une action pour les commandes à distance est similaire à celle pour l'envoi d'un message, la seule différence étant que Zabbix exécutera une commande au lieu d'envoyer un message.

Les commandes à distance peuvent être exécutées par un serveur, un proxy ou un agent Zabbix. Les commandes à distance sur l'agent Zabbix peuvent être exécutées directement par le serveur Zabbix ou via un proxy Zabbix. Les commandes distantes de l'agent Zabbix et du proxy Zabbix sont désactivées par défaut. Elles peuvent être activées en : - ajoutant le paramètre `AllowKey=system.run[*]` dans la configuration de l'agent; - définissant le paramètre EnableRemoteCommands à '1' dans la configuration du proxy.

Les commandes distantes exécutées par le serveur Zabbix sont exécutées comme décrit dans Exécution de commandes y compris la vérification du code de sortie.

Les commandes à distance sont exécutées même si l'hôte cible est en maintenance.

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.

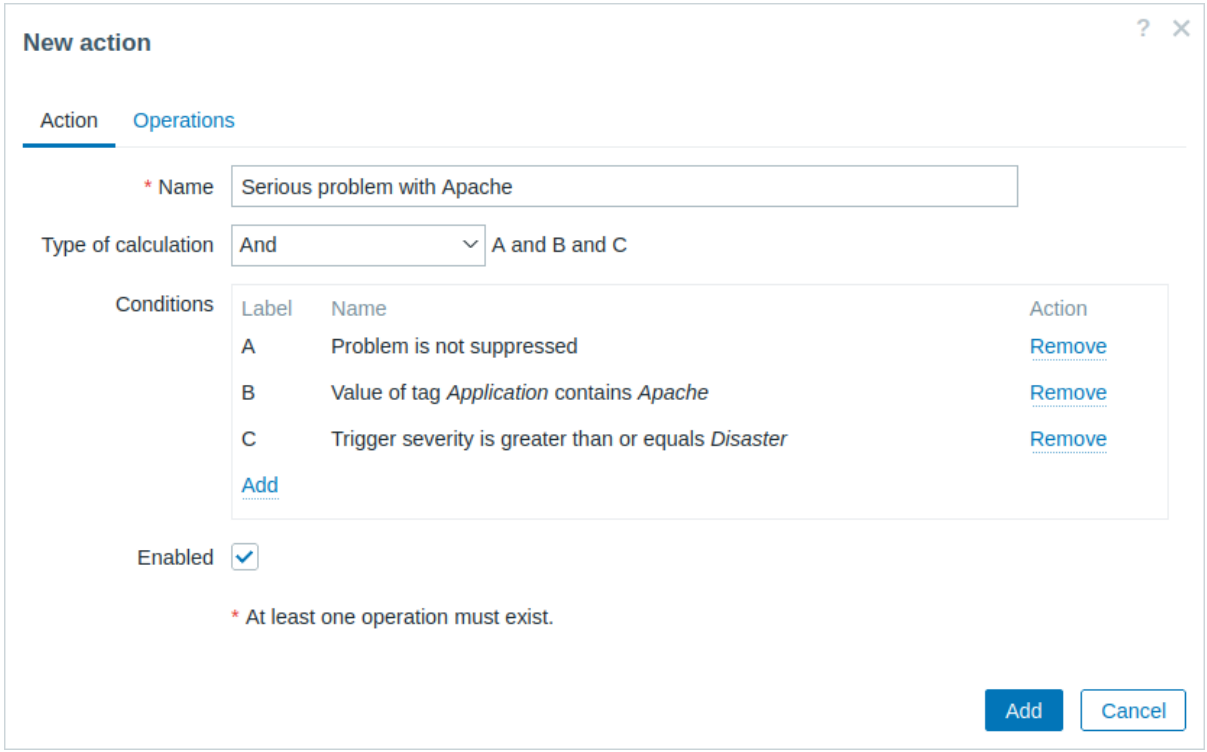
Configuration

Ces commandes distantes, exécutées sur l'agent Zabbix (scripts personnalisés), doivent d'abord être activées dans la configuration de l'agent.

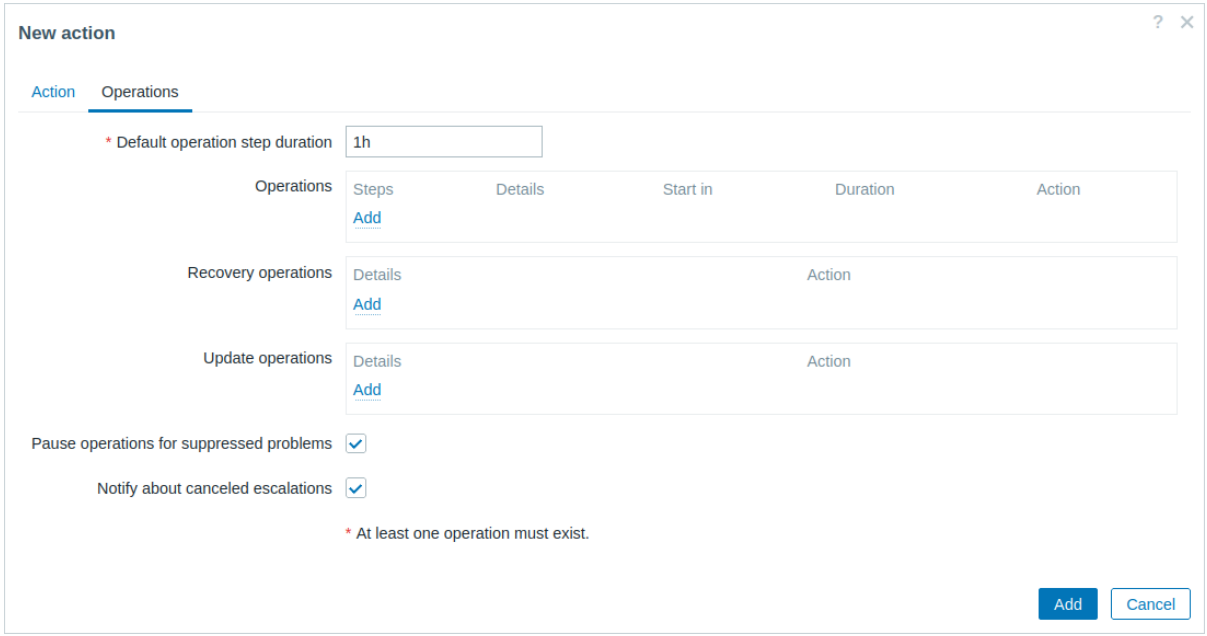
Assurez-vous que le paramètre AllowKey=system.run[<command>,*] est ajouté pour chaque commande autorisée dans la configuration de l'agent afin d'autoriser une commande spécifique avec le mode nowait. Redémarrez le démon de l'agent si vous modifiez ce paramètre.

Ensuite, lors de la configuration d'une nouvelle action dans *Alertes → Actions → Actions sur les déclencheurs* :

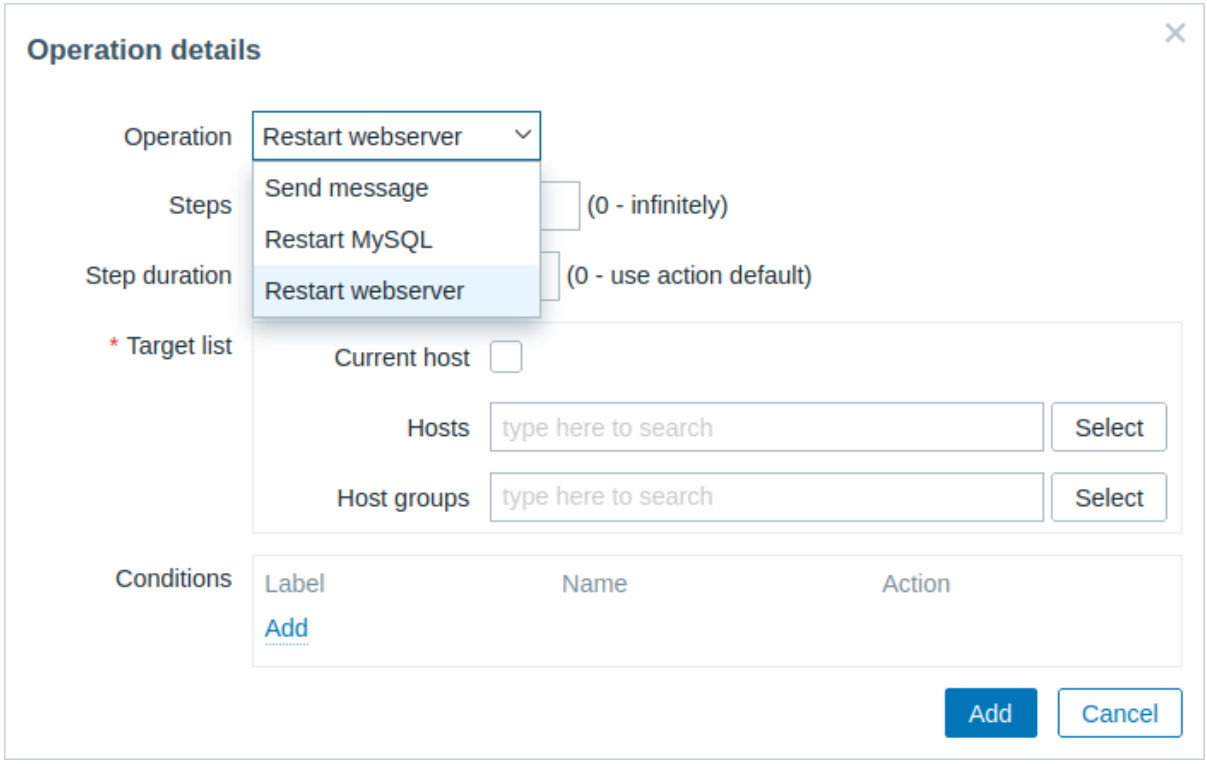
1. Définissez les conditions appropriées, par exemple, indiquez que l'action est activée pour tout problème de gravité désastre concernant l'une des applications Apache.



2. Dans l'onglet *Opérations*, cliquez sur *Ajouter* dans le bloc *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* pour le script.



Scripts prédéfinis

Les scripts disponibles pour les opérations d'action (webhook, script, SSH, Telnet, IPMI) sont définis dans les scripts globaux.

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 face de *Execute on*).

Attention:
Notez l'utilisation de **sudo** : l'utilisateur Zabbix n'a pas, par défaut, les autorisations nécessaires pour redémarrer les services système. Voir ci-dessous pour des conseils 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. Zabbix agent — 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'expiration par la clé system.run[,nowait] et ne sont pas vérifiées quant à leurs résultats d'exécution. Sur Zabbix server et Zabbix proxy, les commandes distantes sont exécutées avec le délai d'expiration défini par le paramètre TrapperTimeout du fichier zabbix_server.conf ou zabbix_proxy.conf et sont vérifiées quant à leurs résultats d'exécution. Pour plus d'informations, voir *Délai d'expiration du script*.

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* :

```
# autorise l'utilisateur 'zabbix' à exécuter toutes les commandes sans mot de passe.
zabbix ALL=NOPASSWD: ALL

# autorise l'utilisateur 'zabbix' à 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 cela, 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 sous une certaine condition.

Afin de redémarrer automatiquement Windows lorsqu'un problème est détecté par Zabbix, définissez le script suivant :

Paramètre du script	Valeur
Portée	'Action operation'
Type	'Script'
Commande	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

3 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.

4 Utiliser 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
```

3 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

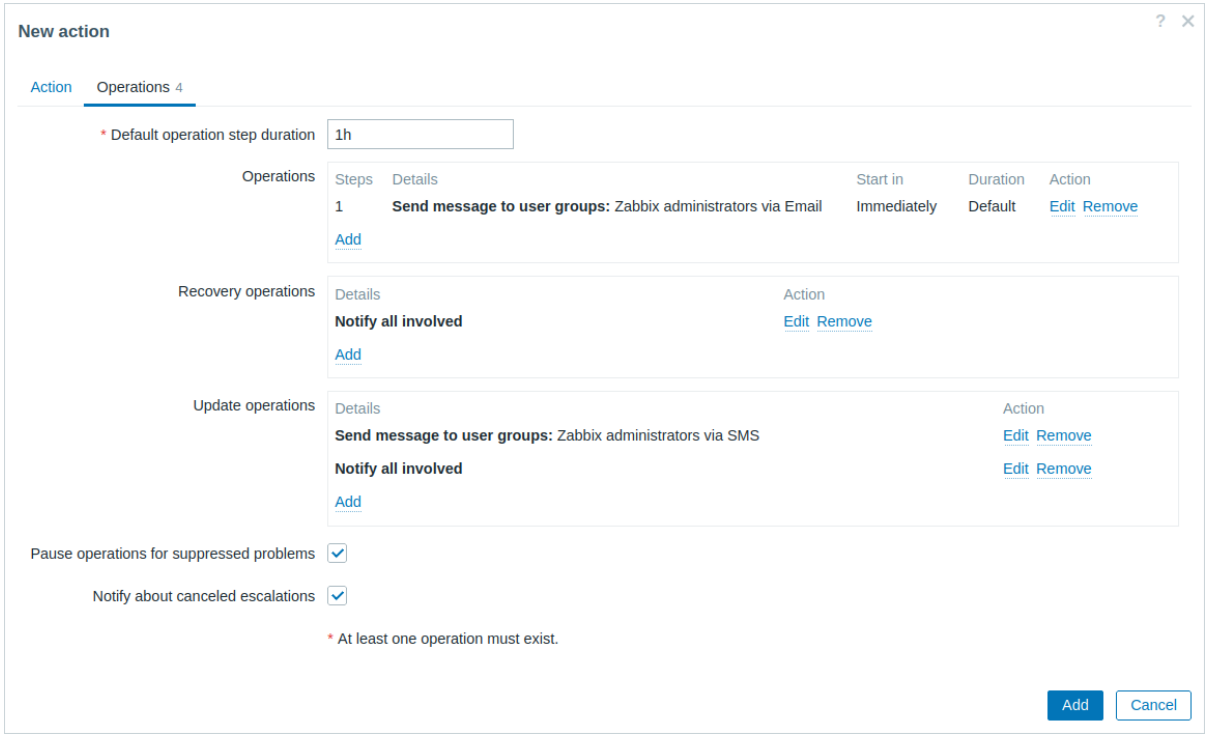
Certains cas d'usage pour les opérations de récupération sont les suivants :

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.

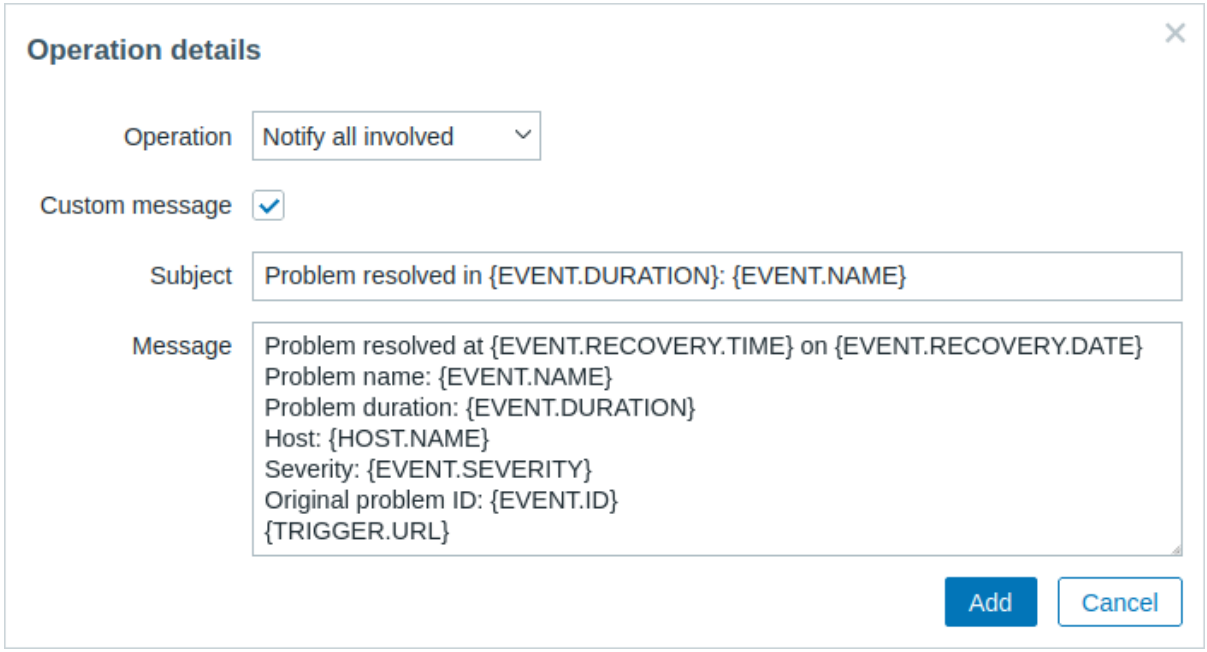
Configuration d'une opération de récupération

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 *Action operation* 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 qui a provoqué l'événement de problème. Cette option ne fonctionnera pas s'il y a plusieurs hôtes dans le déclencheur. **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, à partir de plusieurs groupes d'hôtes, individuellement et à partir d'un groupe d'hôtes). La liste des cibles n'a pas de sens si la commande est exécutée sur le serveur Zabbix. Dans ce cas, sélectionner davantage de cibles a seulement pour effet d'exécuter la commande sur le serveur un plus grand nombre de fois. Notez que, pour les scripts globaux, la sélection de la cible 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.

4 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 lorsqu'ils font l'objet d'un commentaire, sont reconnus, que leur gravité a été modifiée ou qu'ils ont été fermés (manuellement) ;
- *Services* - lorsque la gravité d'un service a changé mais que le service n'est toujours pas rétabli.

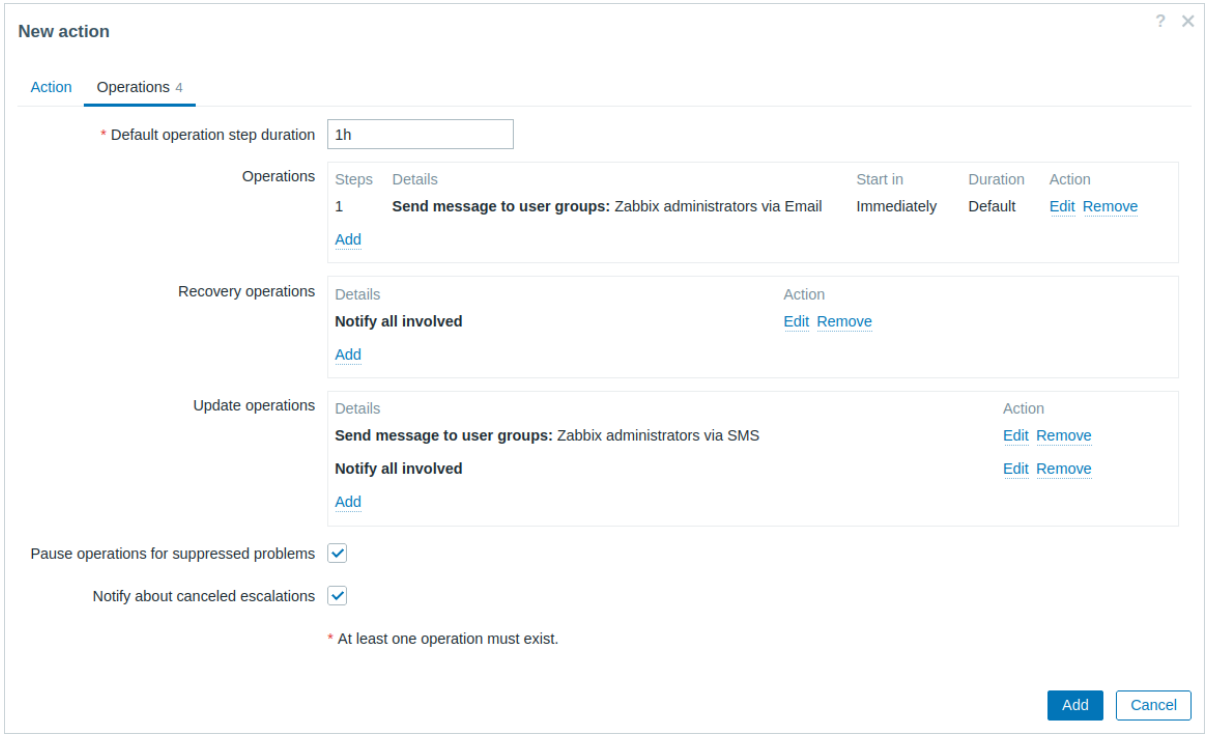
Veuillez noter que les utilisateurs ne reçoivent pas de notifications concernant leurs propres mises à jour.

Les messages ainsi que les commandes distantes sont 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.

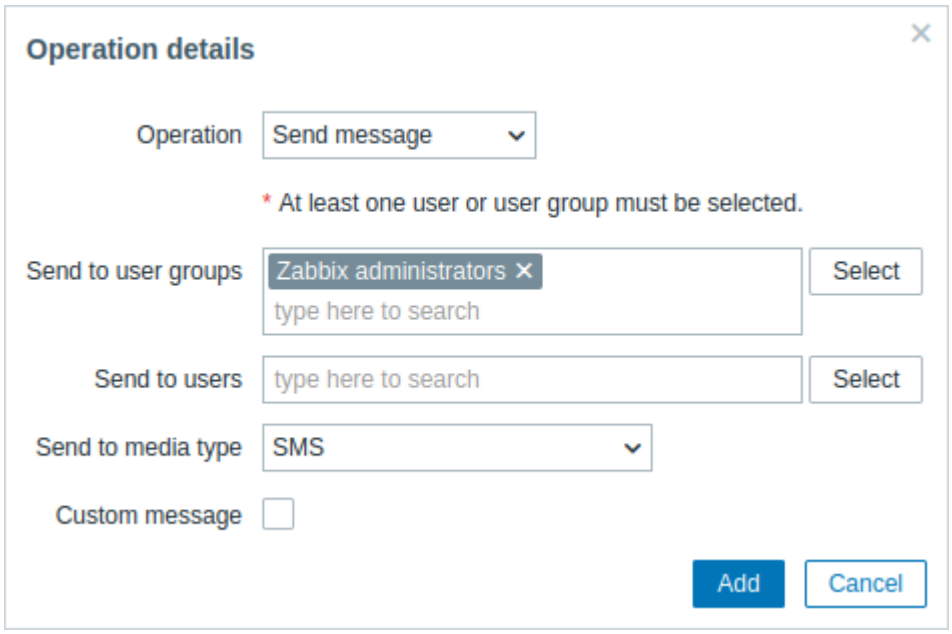
Configuration d'une opération de mise à jour

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.

5 Escalades

Aperçu

Avec les escalades, vous pouvez créer des scénarios personnalisés pour envoyer des notifications ou exécuter des commandes à distance.

Concrètement cela signifie que :

- Les utilisateurs peuvent être informés immédiatement des nouveaux problèmes
- Les notifications peuvent être répétées jusqu'à ce que le problème soit résolu
- L'envoi d'une notification peut être retardé
- Les notifications peuvent être transmises à un autre groupe d'utilisateurs "supérieur"
- Les 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 dans le temps.

Vous pouvez définir à la fois la durée par défaut et une durée personnalisée d'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. La première étape concerne les actions immédiates. Si vous souhaitez retarder 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 problématiques, pas pour la récupération.

Aspects divers du comportement d'escalade

Considérons ce qui se passe dans différentes circonstances si une action contient plusieurs étapes d'escalade.

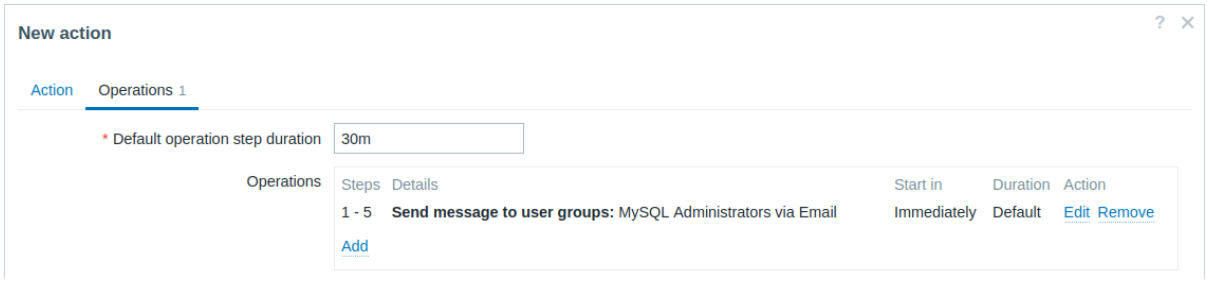
Situation	Comportement
L'hôte en question passe en maintenance après l'envoi de la notification de problème initiale	Selon le paramètre *Suspendre les opérations pour les problèmes supprimés* dans la configuration de l'action, tout les étapes d'escalade restantes sont exécutées soit avec un retard causé par la période de maintenance, soit sans retard. Une période de maintenance n'annule pas les opérations.
La période de temps définie dans la condition d'action ***Période de temps*** *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 ; il a un effet sur le moment où les actions sont démarrées/non démarrées, pas sur les opérations.
Un problème commence pendant la maintenance et continue (n'est pas résolu) après la fin de la maintenance	Selon le paramètre *Suspendre les opérations pour les problèmes supprimés* dans la configuration de l'action, toutes les étapes d'escalade sont exécutées dès la fin de la maintenance ou immédiatement.
Un problème commence pendant une maintenance sans donnée et continue (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 ne soient exécutées.
Différentes escalades se succèdent et se chevauchent	L'exécution de chaque nouvelle escalade remplace l'escalade précédente, mais pour au moins une étape d'escalade qui est toujours exécutée sur l'escalade précédente. Ce comportement est pertinent dans les actions sur les événements qui sont créés avec CHAQUE évaluation de problème du déclencheur.
Lors d'une escalade en cours (comme l'envoi d'un message), en fonction de tout type d'événement : - l'action est désactivée En fonction de l'événement déclencheur : - le déclencheur est désactivé - l'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 - l'hébergeur est désactivé	Le message en cours est envoyé puis un autre message sur l'escalade est envoyé. Le message de suivi contiendra le texte d'annulation au début du corps du message (*NOTE: Escalation canceled*) indiquant la raison (par exemple, **NOTE: Escalation canceled: action '<Action name>' disabled*). De cette façon, le destinataire est informé que l'escalade est annulée et qu'aucune autre étape ne sera exécutée. Ce message est envoyé à tous ceux qui ont reçu les notifications auparavant. La raison de l'annulation est également consignée dans le fichier journal du serveur (à partir du Niveau de débogage 3=Avertissement). Notez que le message* Escalation canceled* est également envoyé si les opérations sont terminées, mais les opérations de récupération sont configurées et ne sont pas encore exécutées.
Lors d'une escalade en cours (comme l'envoi d'un message) l'action est supprimée	Plus aucun message n'est envoyé. Les informations sont consignées 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).

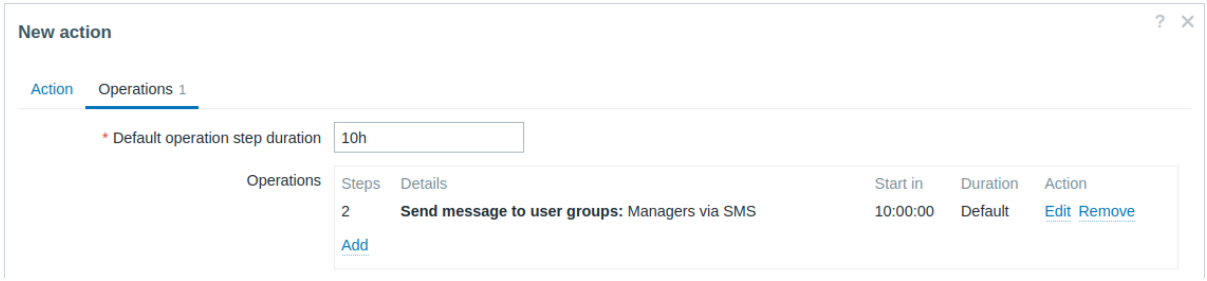
Si le problème est résolu et qu'un message de rétablissement est configuré, il sera envoyé à ceux qui ont reçu au moins un message de problème dans ce scénario d'escalade.

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 ».



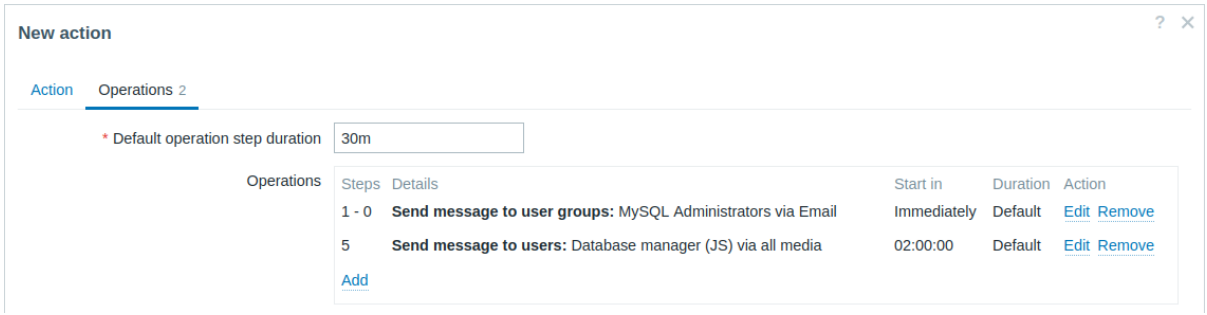
Une notification ne sera envoyée qu'à l'étape 2 du scénario d'escalade, soit 10 heures après le début du problème.

Vous pouvez personnaliser le texte du message avec quelque chose comme « Le problème dure depuis plus de 10 heures ».

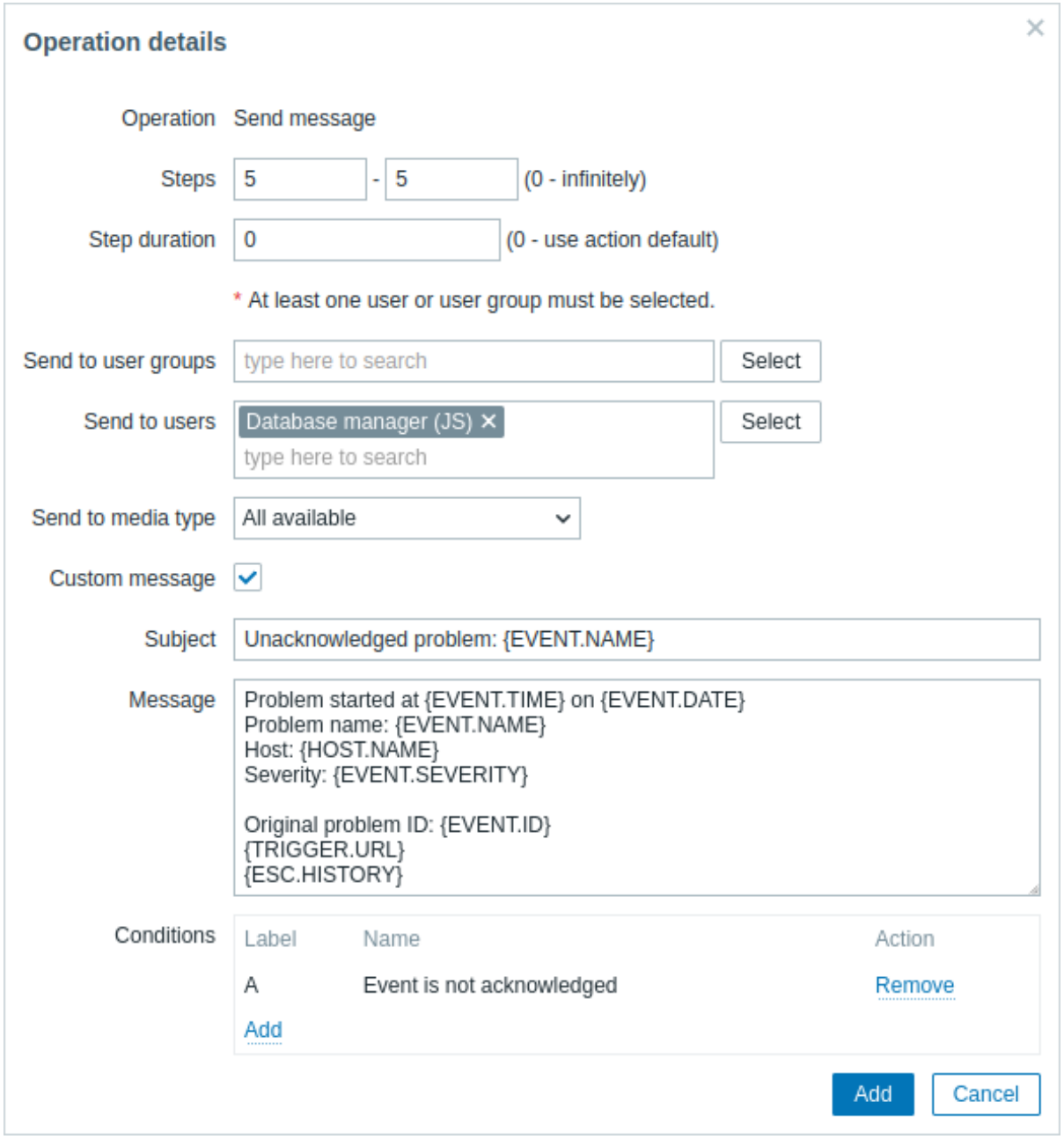
Exemple 3

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 :



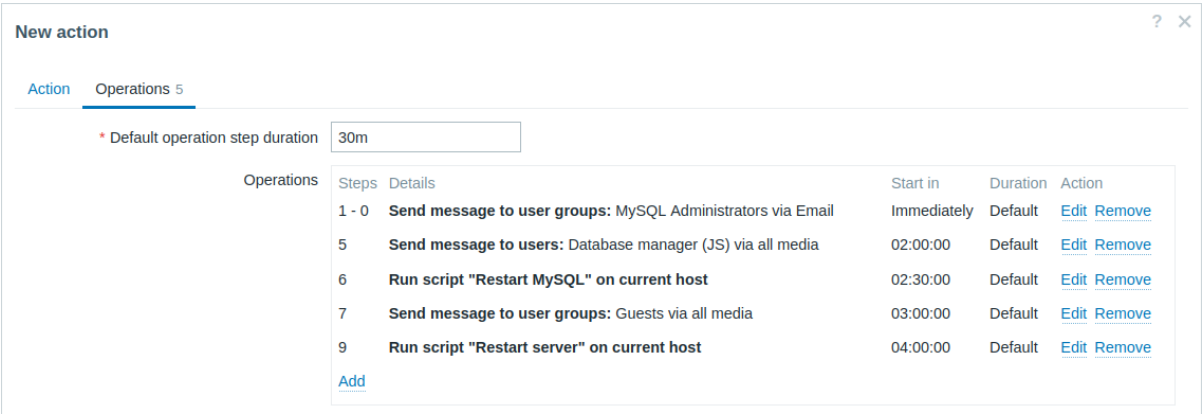
Notez l'utilisation de la macro {ESC.HISTORY} dans le message personnalisé. La macro contiendra des informations sur toutes les étapes précédemment exécutées de cette escalade, telles que les notifications envoyées et les commandes exécutées.

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.

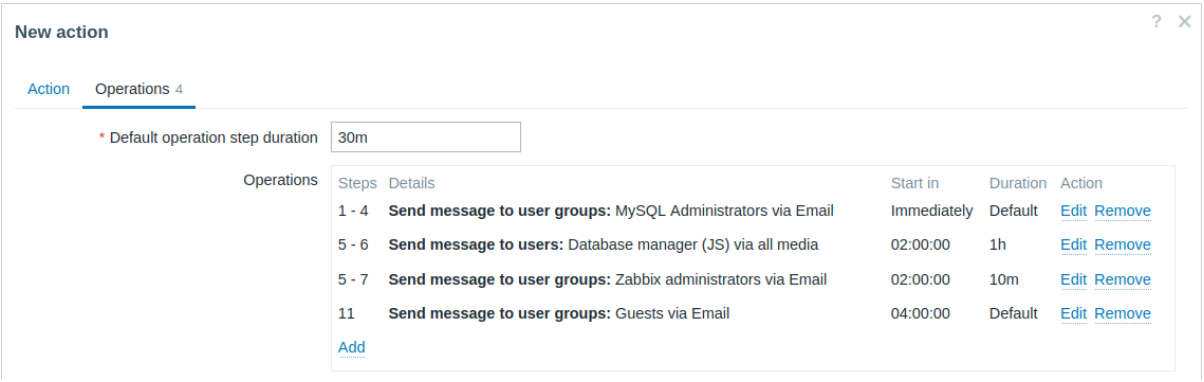
Si le problème persiste, après 30 minutes supplémentaires, Zabbix enverra un message à tous les utilisateurs invités.

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).

3 Réception de notification pour des éléments non supportés

Vue d'ensemble

Il est possible de recevoir des notifications concernant les éléments non pris en charge dans Zabbix.

Cela fait partie du concept des événements internes dans Zabbix, permettant aux utilisateurs d'être notifiés 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 média, 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 concernant les événements internes, la sévérité par défaut (« Not classified ») est utilisée ; laissez-la donc cochée lors de la configuration du média utilisateur si vous souhaitez recevoir des notifications pour les événements internes.

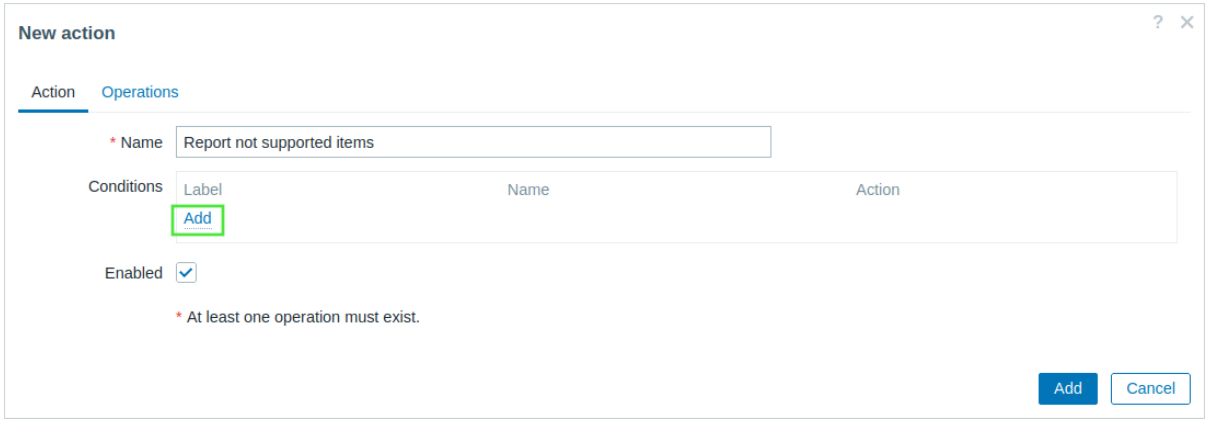
Étape 2

Allez dans *Alerts* → *Actions* → *Internal actions*.

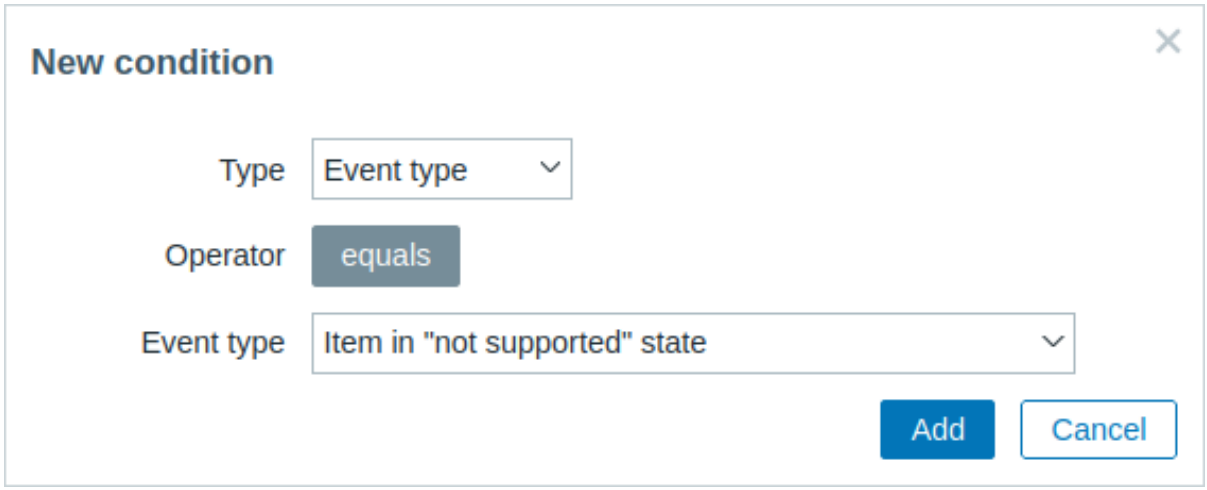
Cliquez sur *Create action* dans le coin supérieur droit de la page pour ouvrir un formulaire de configuration d'action.

Étape 3

Dans l'onglet *Action*, saisissez un nom pour l'action. Cliquez ensuite sur *Add* dans le bloc *Conditions* pour ajouter une nouvelle condition.



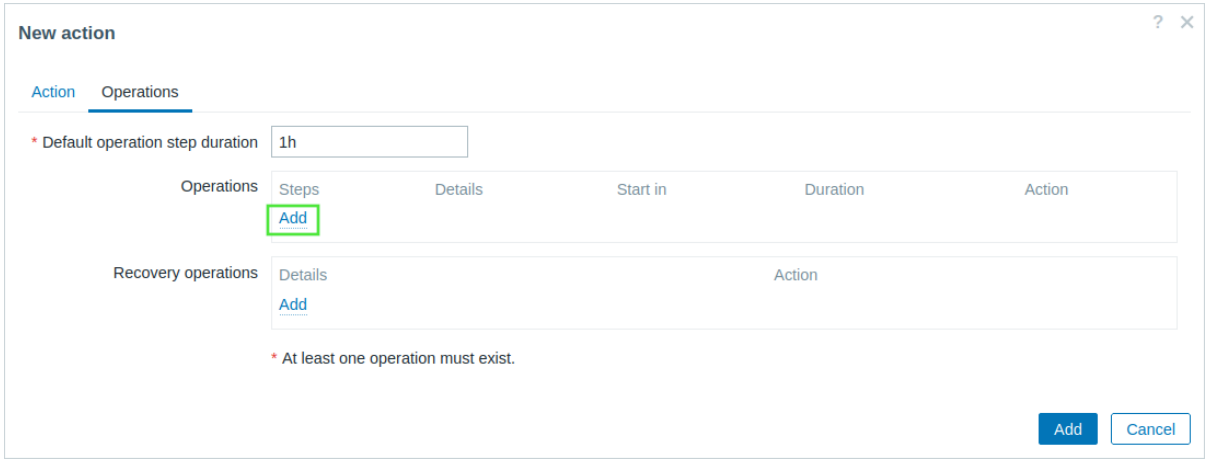
Dans la fenêtre contextuelle *New condition*, sélectionnez « Event type » comme type de condition, puis sélectionnez « Item in 'not supported' state » comme type d'événement.



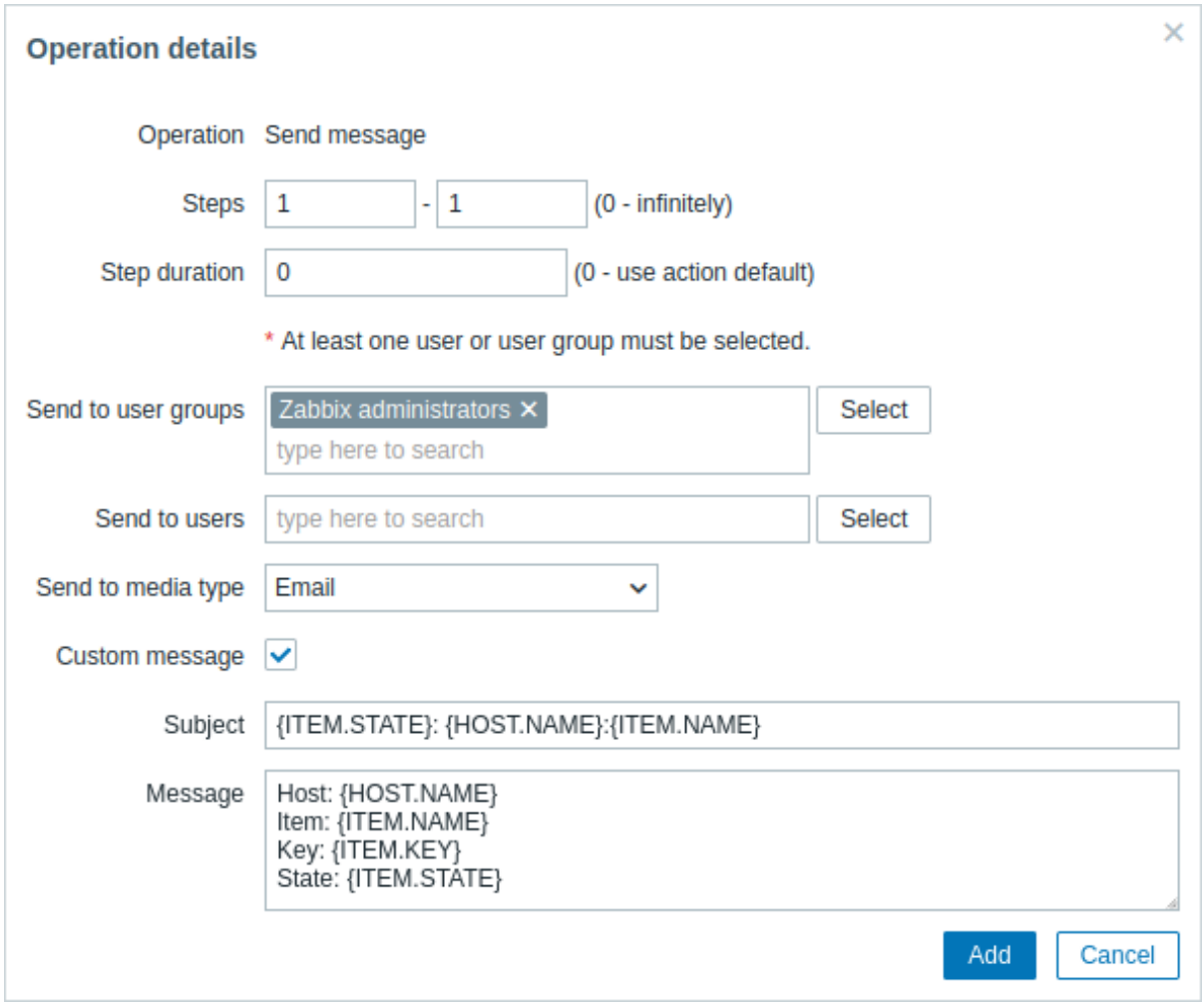
N'oubliez pas de cliquer sur *Add* pour que la condition soit effectivement ajoutée à la liste dans le bloc *Conditions*.

Étape 4

Dans l'onglet *Operations*, cliquez sur *Add* dans le bloc *Operations* pour ajouter une nouvelle opération.



Sélectionnez certains destinataires du message (groupes d'utilisateurs/utilisateurs) ainsi que le type de média (ou « All ») à utiliser pour l'envoi. Cochez la case *Custom message* si vous souhaitez saisir l'objet/le contenu personnalisé du message de problème.

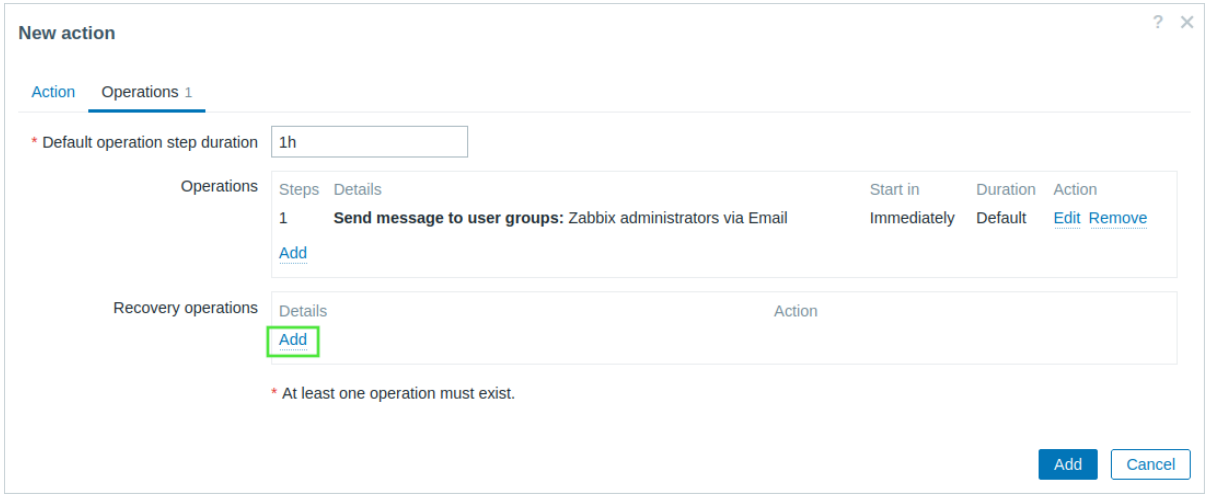


Cliquez sur *Add* pour que l'opération soit effectivement ajoutée à la liste dans le bloc *Operations*.

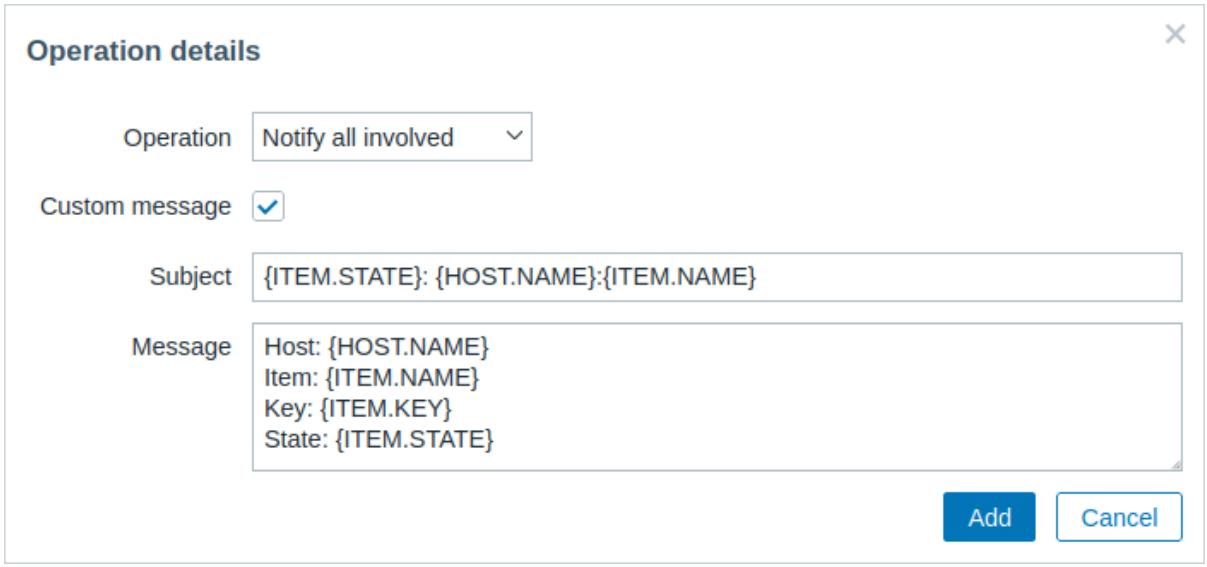
Si vous souhaitez recevoir plus d'une notification, définissez la durée de l'étape de l'opération (intervalle entre les messages envoyés) et ajoutez une autre étape.

Étape 5

Le bloc *Opérations de récupération* permet de configurer une notification de récupération lorsqu'un élément revient à l'état normal. Cliquez sur *Ajouter* dans le bloc *Opérations de récupération* pour ajouter une nouvelle opération de récupération.



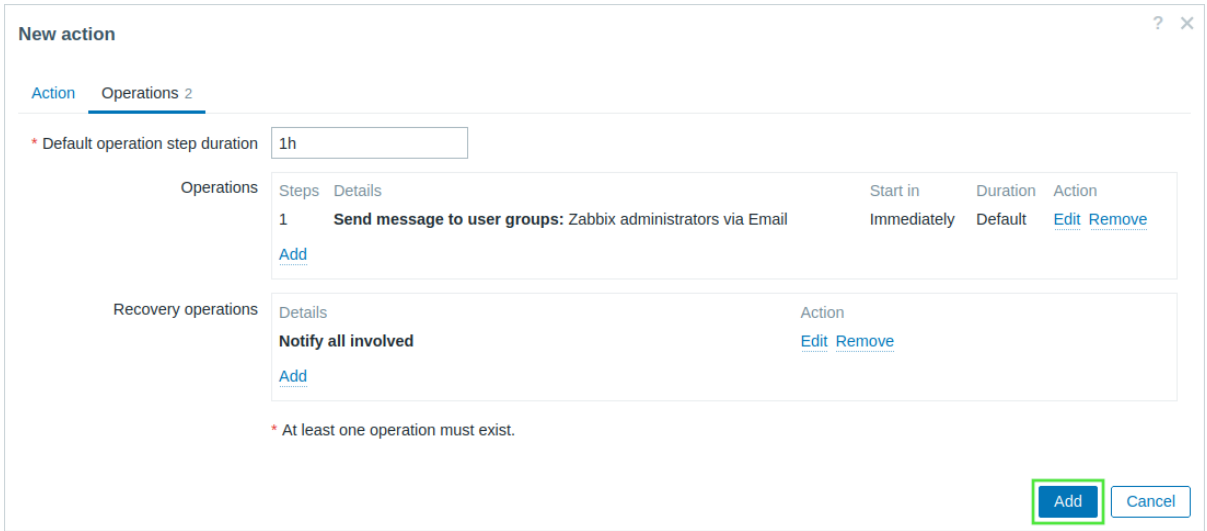
Sélectionnez le type d'opération « Notifier toutes les parties impliquées ». Cochez la case *Message personnalisé* si vous souhaitez saisir l'objet/le contenu personnalisé du message de récupération.



Cliquez sur *Ajouter* dans la fenêtre contextuelle *Détails de l'opération* pour afficher effectivement l'opération dans le bloc *Opérations de récupération*.

Étape 6

Une fois terminé, cliquez sur le bouton *Ajouter* 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.

11 Macros

Aperçu

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 se résolvent en une valeur spécifique en fonction du contexte.

L'utilisation efficace des macros permet de gagner du temps et de rendre la configuration de Zabbix plus transparente.

Dans l'une des utilisations typiques, une macro peut être utilisée dans un modèle. Ainsi, un déclencheur sur un modèle peut être nommé "La charge du processeur est trop élevée sur {HOST.NAME}". Lorsque le modèle est appliqué à l'hôte, tel que le serveur Zabbix, le nom se résoudra en "La charge du processeur est trop élevée sur le serveur Zabbix" lorsque le déclencheur s'affiche dans la section Surveillance.

Des macros peuvent être utilisées dans les paramètres clés de l'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 prendra soin de tous les symboles spéciaux ambigus, s'ils sont 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 liste complète)
- `{<macro>.<func>(<params>)}` - fonctions de macro
- `{$MACRO}` - macro définie par l'utilisateur, éventuellement avec contexte
- `{#MACRO}` - macro pour découverte de bas niveau
- `{?EXPRESSION}` - macro d'expression

1 Fonctions de macro

Aperçu

Les fonctions de macro permettent de personnaliser les valeurs des macros (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 les fonctions prises en charge) ;
- **params** - une liste de paramètres de fonction séparés par des virgules, qui doivent être **placés entre guillemets doubles** s'ils :
  - commencent par un espace ou des guillemets doubles ;
  - contiennent une parenthèse fermante 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 de bas niveau
- les macros d'expression

Les fonctions de macro peuvent être utilisées dans tous les emplacements prenant en charge les macros listées. Cela s'applique sauf s'il est explicitement indiqué qu'une macro seule est attendue (par exemple, lors de la configuration des macros d'hôte ou des filtres de règle de découverte de bas niveau).

Une seule fonction par macro est prise en charge ; plusieurs fonctions de macro chaînées ne sont pas prises en charge.

> **Note:**
> Veuillez consulter les exemples d'échappement pour les cas où des 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 la fonction pour voir tous les détails.

Fonction	Description
btoa	Encodage de la valeur de macro en Base64.
fmtnum	Formatage des nombres pour contrôler le nombre de chiffres affichés après la virgule.
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 de sous-chaîne par correspondance avec une expression régulière (insensible à la casse).
lowercase	Transformation 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 de sous-chaîne par correspondance avec une expression régulière (sensible à la casse).
tr	Translittération des caractères de la valeur de macro.
uppercase	Transformation 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.

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) - le 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 décaler 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 priorité. Par exemple, `-1M/d+1h/w` sera interprété comme `((-1M/d)+1h)/w`.

Exemples :

```
{{TIME}.fmttime(%B)} - renverra "October" à partir d'une valeur reçue de "1633098961"
{{TIME}.fmttime(%d %B,-1M/M)} - renverra "1 September" à partir d'une valeur reçue de "1633098961"
```

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
```

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 de caractère/sous-chaîne dans la valeur de 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 les groupes de capture.

Commentaires :

- Les motifs 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 dans la valeur d
```

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"
```

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_1	1

Fonction de macro	Valeur reçue	Sortie
`{{#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.

2 Macros utilisateur

Vue d'ensemble

Les macros utilisateur sont prises en charge dans Zabbix pour offrir une plus grande flexibilité, en plus des macros prises en charge prêtes à l'emploi.

Les macros utilisateur peuvent être définies au niveau global, du modèle et de l'hôte. Ces macros ont une syntaxe spéciale :

`{$MACRO}`

Zabbix résout les macros selon l'ordre de priorité suivant :

1. macros 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 sur plusieurs modèles liés du même niveau, la macro du modèle ayant l'ID le plus faible sera utilisée. Ainsi, avoir des macros portant le même nom dans plusieurs modèles constitue 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) sont volontairement laissées 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 :

- nom de l'élément
- paramètre de clé d'élément

- intervalles de mise à jour de l'élément et intervalles flexibles
- nom et description du déclencheur
- paramètres et constantes d'expression du déclencheur (voir les exemples)
- de nombreux autres emplacements - voir 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.

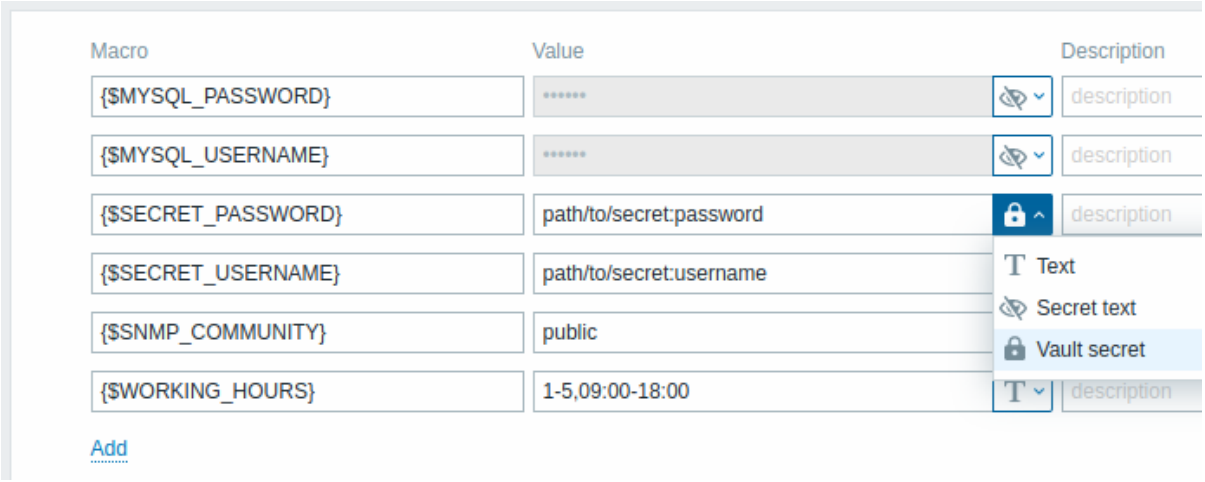
Note:
Il est recommandé d'utiliser des macros d'hôte plutôt que des macros globales, car l'ajout, la mise à jour ou la suppression de macros globales force une mise à jour incrémentielle de la configuration pour tous les hôtes. Pour plus d'informations, voir Passive and active agent checks.

Configuration

Pour définir des macros utilisateur, accédez à l'emplacement correspondant dans le frontend :

- 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 :



Paramètre	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/une 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 de 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) ; ainsi, 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 référencent un paramètre ou une constante ; elles ne seront PAS résolues si elles référencent un hôte, une clé d'élément,

une fonction, un opérateur ou 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 disposant d'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 d'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 de la macro au niveau de l'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 des hôtes individuels.

Note:
Si vous souhaitez utiliser la quantité de valeurs comme paramètre de fonction (pour exemple, **max(/host/key,#3)**), incluez le signe dièse dans la définition de la macro comme ceci : 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 de la fonction de déclenchement, dans cet exemple la fonction **min()**.

Exemple 4

Synchronisez la condition d'indisponibilité de l'agent avec l'intervalle de mise à jour de l'élément :

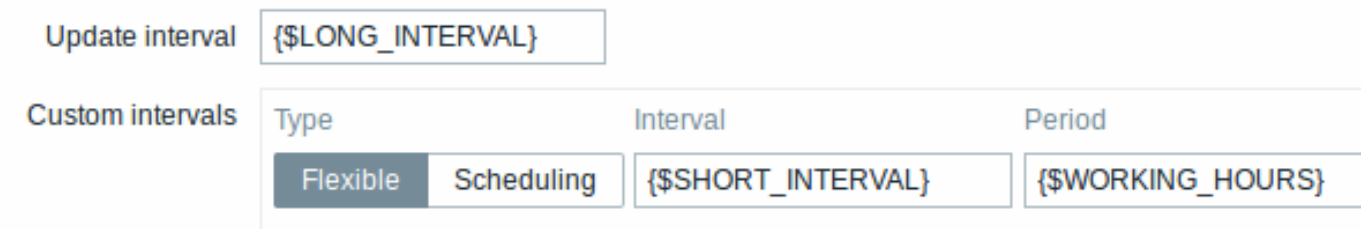
- définir la macro {$INTERVAL} et l'utiliser dans l'intervalle de mise à jour de l'élément ;
- utilisez {$INTERVAL} comme paramètre du déclencheur d'indisponibilité de l'agent :

```
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 *Working time* dans *Administration → General → GUI* ;
- l'utiliser dans le champ *When active* dans *Users → Users*, onglet *Media* d'un utilisateur ;
- l'utiliser pour configurer une interrogation des éléments plus fréquente pendant les heures de travail :

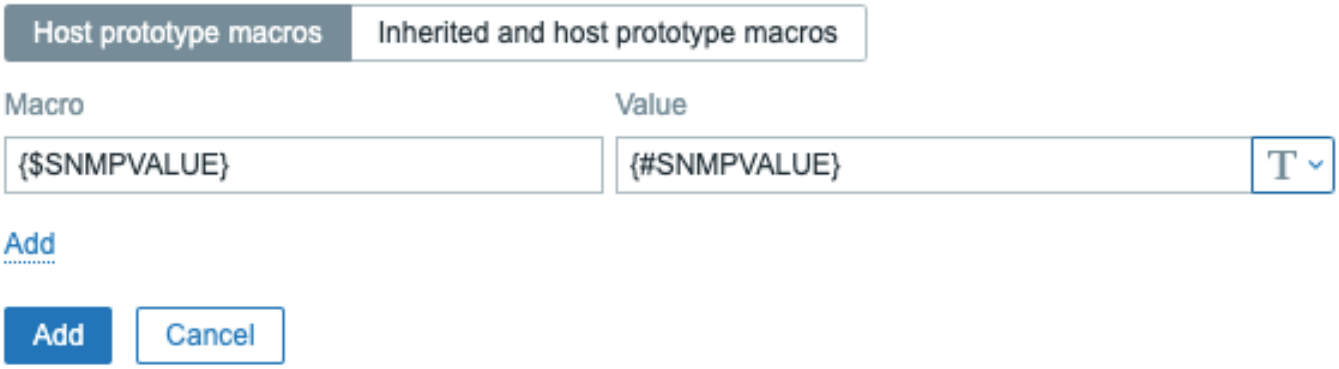


- l'utiliser dans la condition d'action *Time period* ;
- ajuster les heures de travail dans *Administration → Macros*, si nécessaire.

Exemple 6

Utilisez la macro de prototype d'hôte pour configurer des éléments pour les hôtes découverts :

- sur un prototype d'hôte, définissez la macro utilisateur {$SNMPVALUE} avec la macro {#SNMPVALUE} de découverte de bas niveau comme valeur :



- attribuer le modèle *Generic SNMPv2* au prototype hôte ;
- utilisez {$SNMPVALUE} dans le champ *SNMP OID* des éléments de modèle *Generic SNMPv2*.

Contexte de la macro utilisateur

Voir macros utilisateur avec contexte.

3 Macros utilisateur avec contexte

Aperçu

Un contexte facultatif peut être utilisé dans les macros utilisateur, permettant de remplacer la valeur par défaut par une valeur spécifique au contexte.

Le contexte est ajouté au nom de la macro ; la syntaxe varie selon que le contexte est une valeur de texte statique :

```
{$MACRO:"static text"}
```

ou une expression régulière :

```
{$MACRO:regex:"regular expression"}
```

Notez qu'une macro avec un contexte d'expression régulière ne peut être définie que dans la configuration de la macro utilisateur. Si le préfixe `regex:` est utilisé ailleurs comme contexte de macro utilisateur, comme dans une expression de déclencheur, il sera traité comme un contexte statique.

La citation contextuelle est facultative (voir aussi notes importantes).

Exemples de contexte 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). Identique à {$LOW_SPACE_LIMIT:/tmp}.
{$LOW_SPACE_LIMIT:regex:"^/var/log/.*$"}	Macro utilisateur avec contexte (expression régulière). Correspond à toutes les chaînes préfixées par /var/log/.

Cas d'utilisation

Les macros utilisateur avec contexte peuvent être définies afin d'obtenir des seuils plus flexibles dans les expressions de déclencheur (en fonction des 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 selon les points de montage ou les 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 correspondant au motif d'expression régulière (comme /etc, /tmp ou /var) ont moins de 30 % d'espace disque libre
- les dossiers qui ne correspondent pas au motif d'expression régulière et qui ne sont pas /home ont 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.

4 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

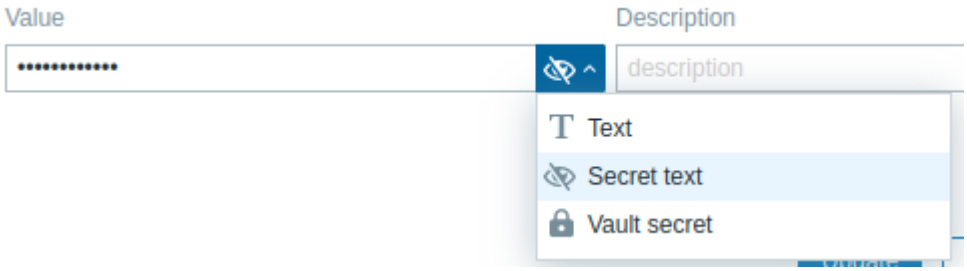
Note:
Bien que la valeur d'une macro secrète soit masquée, elle peut être révélée par son utilisation dans des éléments. Par exemple, dans un script externe, une instruction echo faisant référence à une macro secrète peut être utilisée pour révéler la valeur de la macro dans l'interface, car le serveur 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 *Valeur* et sélectionnez l'option *Texte secret* :



Une fois la configuration enregistrée, il ne sera plus possible d'afficher la valeur.

Pour modifier la valeur de la macro, survolez le champ *Valeur* et cliquez sur le bouton *Définir une nouvelle valeur* (apparaît au survol) :



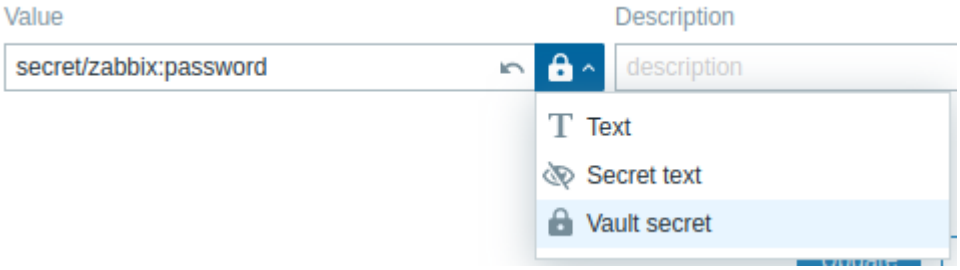
Lorsque vous cliquez sur le bouton *Définir une nouvelle valeur* (ou modifiez le type de valeur de la macro), la valeur actuelle sera effacée. Vous pouvez restaurer la valeur d'origine en cliquant sur la flèche à la fin du champ *Valeur* (disponible uniquement avant l'enregistrement de la nouvelle configuration). 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 au fournisseur, voir :

- HashiCorp
- CyberArk

Les valeurs des macros secrètes du coffre-fort 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*, les secrets du coffre-fort sont récupérés uniquement par le serveur, et le proxy Zabbix reçoit les valeurs des macros secrètes du coffre-fort depuis le serveur Zabbix à chaque synchronisation de la 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 la valeur d'une 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[192.0.2.0,80]`.

Contexte	Paramètre
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
	Post
	En-têtes
	Nom d'utilisateur
	Mot de passe
	Mot de passe de 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*
	Mot de passe
Agent JMX	*Nom d'utilisateur*
	Mot de passe
Prétraitement des valeurs 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 clé SSL
Connecteurs	
Connecteur	*URL*
	Nom d'utilisateur

Contexte	Paramètre
	Mot de passe
	Jeton
	Proxy HTTP
	Fichier de certificat SSL
	Fichier de clé SSL
	Mot de passe de 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

5 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.

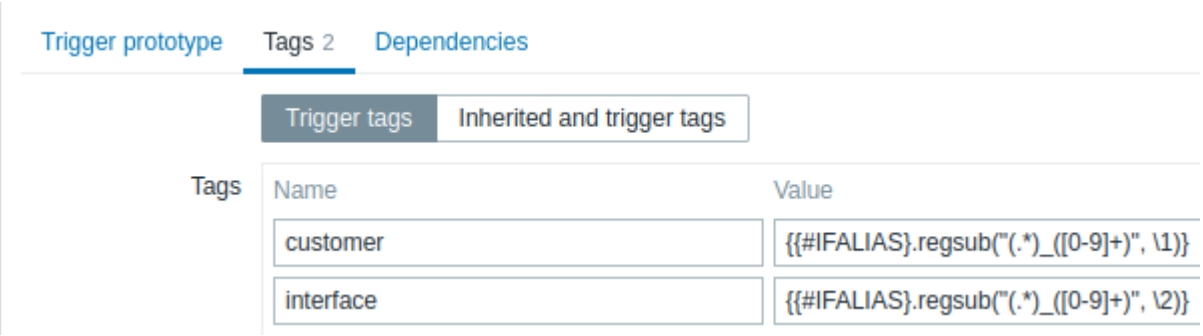
Utilisation des fonctions de macro

Les fonctions de macro sont prises en charge avec les macros de découverte de bas niveau, sauf dans le filtre de la règle de découverte de bas niveau, ce qui permet d'extraire une certaine partie 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 d'interface à partir de la macro LLD suivante à des fins de marquage des événements :

```
{#IFALIAS}=customername_1
```

Pour ce faire, la fonction de macro `regsub` peut être utilisée avec la macro dans le champ de valeur du tag d'événement d'un prototype de déclencheur :



Notez que les virgules ne sont pas autorisées dans les paramètres de clé d'élément non entre guillemets ; le paramètre contenant une fonction de macro doit donc être placé entre guillemets. Le caractère antislash (\) 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, voir : Fonctions de macro

Les fonctions de macro sont prises en charge dans les macros de découverte de bas niveau depuis Zabbix 4.0.

Item prototypes with no key parameters

Some item types (for example *Zabbix trapper* and *Simple checks*) do not expose regular key parameters in the same way as agent items. When creating an item prototype for such types, place the LLD macro inside the item key parameter brackets `[...]` so it is treated as a parameter and substituted during discovery:

```
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.

6 Macros d'expression

Vue d'ensemble

Les macros d'expression vous permettent d'effectuer des calculs dans les champs.

Leur valeur est calculée en résolvant d'abord 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.

Remarques sur l'utilisation :

- Utilisez les macros {FUNCTION.*} pour référencer les valeurs de fonction des expressions de déclencheur/des 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 des noms d'hôte lors de la liaison de modèles.

```
{?{FUNCTION.VALUE2} - {FUNCTION.VALUE3}}
{?max(/{HOST.HOST}/{ITEM.KEY},3h)}
```

Voir aussi Exemples d'expressions de déclencheur pour un exemple d'utilisation d'une macro d'expression dans un nom d'événement.

12 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 cryptés et stockés dans la base de données Zabbix. Les utilisateurs ne peuvent pas utiliser leur ID utilisateur et leur mot de passe pour se connecter directement au serveur UNIX à moins qu'ils n'aient également été configurés conformément à UNIX. La communication entre le serveur Web et le navigateur de l'utilisateur peut être protégée par SSL.

Avec un schéma d'autorisation utilisateur flexible, vous pouvez restreindre et différencier les droits pour :

- accéder aux fonctions d'administration de l'interface Zabbix
- effectuer certaines actions dans l'interface Web
- accéder aux hôtes surveillés dans les groupes d'hôtes
- utiliser des méthodes API spécifiques

1 Configuration d'un utilisateur

Vue d'ensemble

L'installation initiale de Zabbix comporte deux utilisateurs prédéfinis :

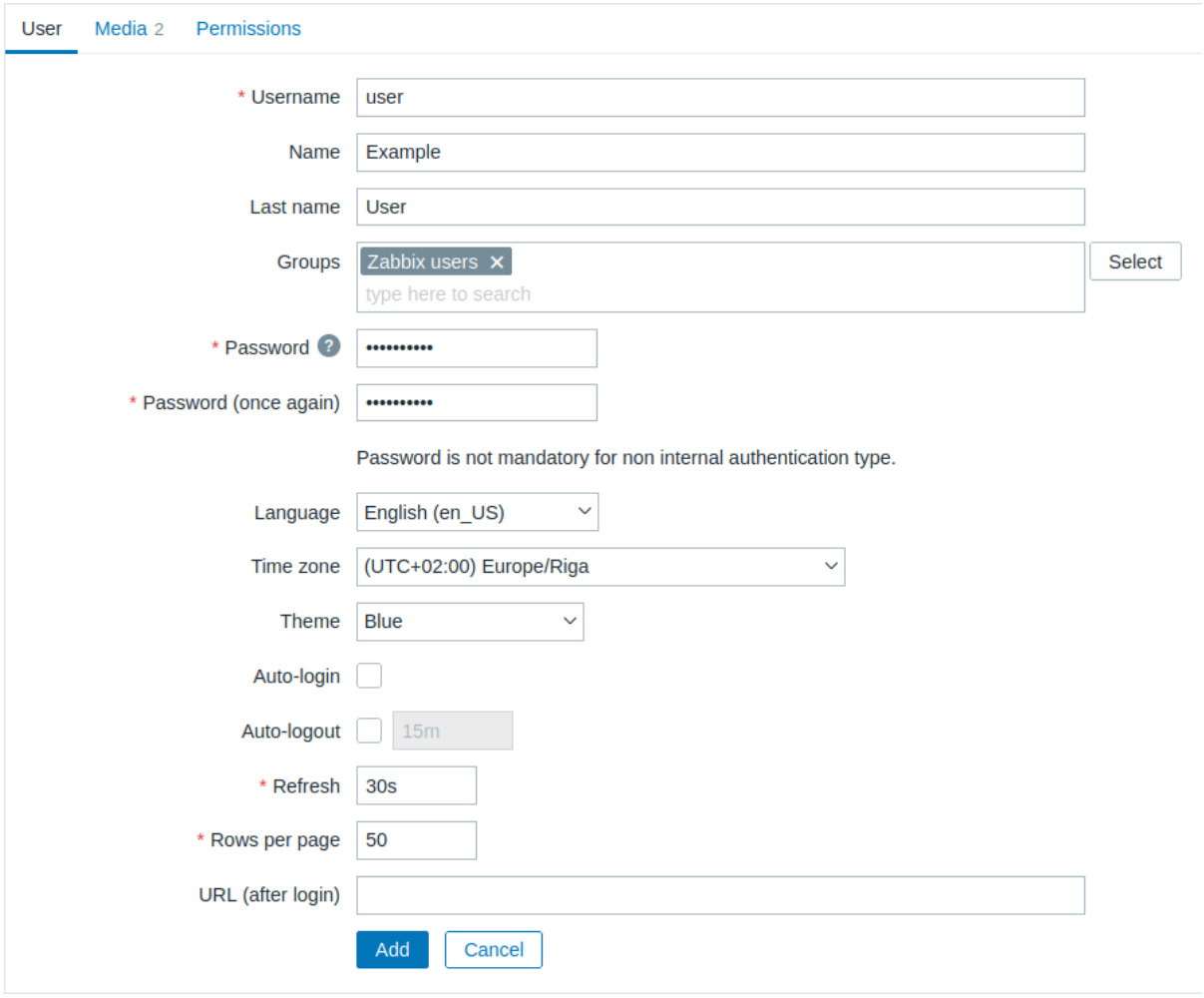
- *Admin* - un superutilisateur Zabbix disposant de toutes les autorisations.
- *guest* - un utilisateur Zabbix spécial. L'utilisateur « guest » est désactivé par défaut. Si vous l'ajoutez au groupe d'utilisateurs Guests, vous pourrez vous connecter avec cet utilisateur et accéder aux pages de supervision dans Zabbix. Notez que, par défaut, « guest » n'a aucune autorisation sur les objets Zabbix.

Pour configurer un utilisateur :

- Allez dans *Users* → *Users*.
- Cliquez sur *Create user* (ou sur un nom d'utilisateur pour modifier un utilisateur existant).
- Modifiez les attributs de l'utilisateur dans le formulaire.

Attributs généraux

L'onglet *Utilisateur* contient les attributs généraux de l'utilisateur :



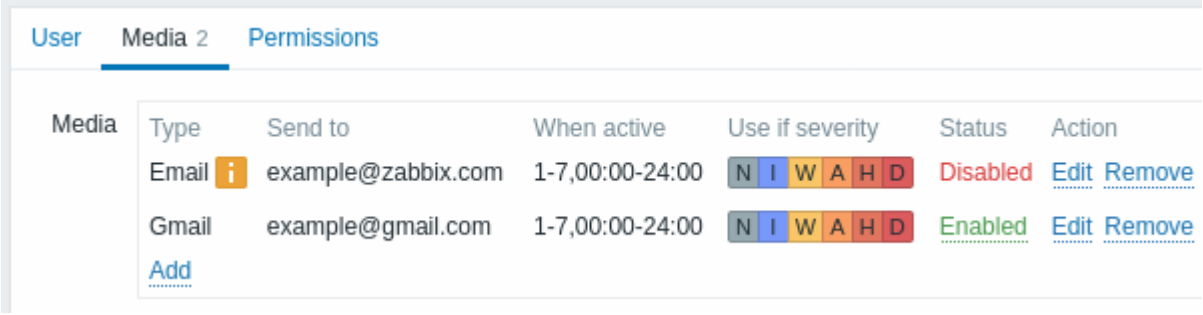
Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge.

Paramètre	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 des notifications.
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 des notifications.
Groupes	Sélectionnez les groupes d'utilisateurs auxquels l'utilisateur appartient. Ce champ prend en charge l'auto-complétion ; ainsi, lorsque vous commencez à saisir le nom d'un groupe d'utilisateurs, une liste déroulante des groupes correspondants est proposée. Faites défiler pour sélectionner. Vous pouvez également cliquer sur *Sélectionner* pour ajouter des groupes. Cliquez sur « x » pour supprimer l'élément sélectionné. L'appartenance aux groupes d'utilisateurs détermine à quels groupes d'hôtes et hôtes l'utilisateur aura accès.
Mot de passe	Deux champs permettent de saisir le mot de passe de l'utilisateur, ou un bouton *Changer le mot de passe* si l'utilisateur existe déjà. Cliquer sur le bouton *Changer le mot de passe* ouvre deux champs pour saisir un nouveau mot de passe. Pour l'utilisateur ayant le *rôle Super admin* qui modifie son propre mot de passe, cliquer sur le bouton *Changer le mot de passe* ouvre un champ supplémentaire pour saisir le mot de passe actuel (ancien). Après un changement de mot de passe réussi, l'utilisateur dont le mot de passe a été modifié sera déconnecté de toutes les 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 web Zabbix. L'extension php gettext est requise pour que les traductions fonctionnent.

Paramètre	Description
Fuseau horaire	Sélectionnez le fuseau horaire pour remplacer le fuseau horaire global au niveau de l'utilisateur, ou sélectionnez **Par défaut du système** pour utiliser les paramètres globaux de fuseau horaire.
Thème	Définit l'apparence de l'interface web : **Par défaut du système** - utiliser les paramètres système par défaut **Bleu** - thème bleu standard **Sombre** - thème sombre alternatif **Clair à contraste élevé** - thème clair avec contraste élevé **Sombre à contraste élevé** - thème sombre avec contraste élevé
Connexion automatique	Cochez cette case pour que Zabbix se souvienne de l'utilisateur et le connecte automatiquement pendant 30 jours. Lors de la connexion avec *Se souvenir pendant 30 jours* : - La déconnexion automatique est réinitialisée (la session persiste pendant 30 jours). - La connexion automatique est activée pour une réauthentification transparente. Lors de la connexion sans *Se souvenir pendant 30 jours* : - La connexion automatique est activée pour une réauthentification transparente. - La déconnexion automatique reste régie par le délai d'expiration standard. Des cookies du navigateur sont utilisés à cette fin.
Déconnexion automatique	Lorsque cette case est cochée, l'utilisateur sera 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 pour toute la période. Les suffixes temporels 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 indisponible" est activée et que l'interface web Zabbix reste ouverte. * Lorsque les pages du menu Monitoring effectuent des actualisations d'informations en arrière-plan. * En cas de connexion avec l'option *Se souvenir de moi pendant 30 jours* cochée.
Actualisation	Définissez le taux d'actualisation utilisé pour les graphiques, les données en texte brut, etc. Peut être défini sur 0 pour désactiver. Les suffixes temporels sont pris en charge, par exemple 90s, 5m, 1h.
Lignes par page	Vous pouvez déterminer combien de lignes par page seront affichées dans les listes.
URL (après 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 l'envoi de notifications.



Cliquez sur *Ajouter* pour attribuer des médias à l'utilisateur.

Si le type de média a été désactivé :

- Une icône d'information jaune s'affiche après le nom.
- *Désactivé* s'affiche dans la colonne Statut.

Note:
Les autorisations d'un utilisateur pour modifier les détails de ses propres médias peuvent être accordées/révoquées en fonction de son rôle utilisateur (voir l'option *Créer et modifier ses propres médias*). Les autorisations d'un utilisateur Super admin pour modifier les détails des médias d'autres utilisateurs peuvent également être accordées/révoquées en fonction de son rôle utilisateur (voir l'option *Créer et modifier les médias utilisateur*)

Consultez la section Types de médias pour plus de détails sur la configuration des médias utilisateur.

Autorisations

L'onglet *Autorisations* contient des informations sur les éléments suivants :

- Le rôle utilisateur (obligatoire pour tout utilisateur nouvellement créé), qui ne peut être modifié que par un utilisateur de type *Super admin*.

Attention:
Les utilisateurs ne peuvent pas être créés sans rôle utilisateur (sauf via l'API utilisateur 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 ne peuvent se connecter à Zabbix qu'en utilisant l'authentification LDAP ou SAML, à condition que leurs informations LDAP/SAML correspondent aux correspondances de groupes d'utilisateurs configurées dans Zabbix.

- Le type d'utilisateur (*User*, *Admin*, *Super admin*), défini dans la configuration du rôle utilisateur.
- Les groupes d'hôtes et de modèles auxquels l'utilisateur a accès.
  - Les utilisateurs de type *User* et *Admin* n'ont, par défaut, accès à aucun groupe, modèle ni hôte. Pour accorder cet accès, les utilisateurs doivent être inclus dans des groupes d'utilisateurs configurés avec des autorisations sur les entités concernées.
- Les droits d'accès aux sections et éléments de l'interface Zabbix, aux modules et aux méthodes de l'API.
  - Les éléments dont l'accès est autorisé sont affichés en vert, tandis que ceux dont l'accès est refusé le sont en gris clair.
- Les droits d'exécuter des actions spécifiques.
  - Les actions que l'utilisateur est autorisé à effectuer sont affichées en vert, tandis que celles qui sont refusées le sont en gris clair.

Consultez la page Autorisations pour plus de détails.

2 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 permissions dans Zabbix dépendent principalement du type d'utilisateur :

- *Utilisateur* - dispose de droits d'accès limités aux sections du menu (voir ci-dessous) et n'a accès à aucune ressource par défaut. Toute permission sur des groupes d'hôtes ou de modèles doit être explicitement attribuée ;
- *Admin* - dispose de droits d'accès incomplets aux sections du menu (voir ci-dessous). L'utilisateur n'a accès à aucun groupe d'hôtes par défaut. Toute permission sur des groupes d'hôtes ou de modèles doit être explicitement accordée ;
- *Super admin* - 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 permissions 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 utilisateur empêche uniquement l'ouverture de cette page ; il ne supprime pas la possibilité d'accéder aux données sous-jacentes dans 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	Admin	Super admin
Tableaux de bord		+	+	+
Surveillance		+	+	+
	Problèmes	+	+	+
	Hôtes	+	+	+
	Dernières données	+	+	+
	Cartes	+	+	+
	Découverte		+	+
Services		+	+	+
	Services	+	+	+

Section du menu		Utilisateur	Admin	Super admin
	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 d'événements			+
	Découverte		+	+
Alertes			+	+
	Actions de déclencheur		+	+
	Actions de service		+	+
	Actions de découverte		+	+
	Actions d'auto-enregistrement		+	+
	Actions internes		+	+
	Types de média			+
	Scripts			+
Utilisateurs				+
	Groupes d'utilisateurs			+
	Rôles utilisateur			+
	Utilisateurs			+
	Jetons d'API			+
	Authentification			+
Administration				+
	Général			+
	Journal d'audit			+
	Nettoyage interne			+
	Groupes de proxys			+
	Proxys			+
	Macros			+
	File d'attente			+

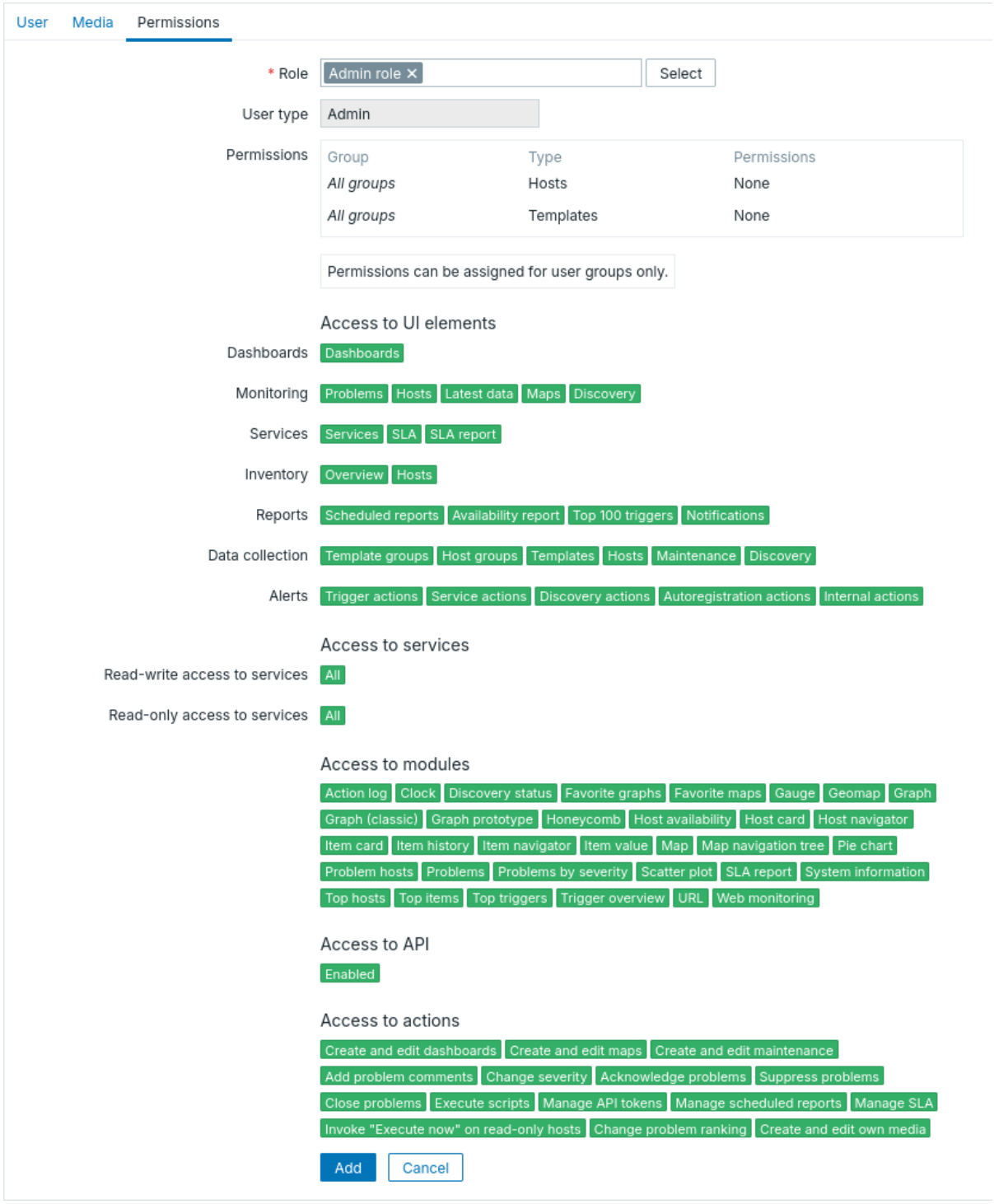
Rôles utilisateur

Les rôles utilisateur permettent d'apporter des ajustements personnalisés aux autorisations définies par le type d'utilisateur. Bien qu'aucune autorisation ne puisse être ajoutée (car cela dépasserait celles du type d'utilisateur), certaines autorisations peuvent être révoquées.

En outre, un rôle utilisateur détermine l'accès non seulement aux sections du menu, mais aussi aux services, aux modules, aux méthodes de l'API et à diverses actions dans l'interface web.

Les rôles utilisateur sont configurés dans la section *Utilisateurs* → *Rôles utilisateur* par les utilisateurs Super admin.

Les rôles utilisateur sont attribués aux utilisateurs dans le formulaire de configuration de l'utilisateur, onglet *Permissions*, par les utilisateurs Super admin.



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 du groupe d'hôtes/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 disposant d'un accès au groupe d'hôtes qui contient cet hôte.

De même, l'accès à un modèle ne peut être accordé à un utilisateur qu'en faisant partie d'un groupe d'utilisateurs disposant d'un accès au groupe de modèles qui contient ce modèle.

3 Groupes d'utilisateur

Vue d'ensemble

Les groupes d'utilisateurs permettent de regrouper les utilisateurs à la fois à des fins organisationnelles et pour attribuer des autorisations d'accès aux données. Les autorisations de consultation et de configuration des données des groupes d'hôtes et des

groupes de modèles sont attribuées aux groupes d'utilisateurs, et non aux utilisateurs individuels.

Il est souvent judicieux de distinguer les informations disponibles pour un groupe d'utilisateurs de celles accessibles à un autre. Cela peut être réalisé en regroupant les utilisateurs, puis en attribuant des autorisations différentes aux groupes d'hôtes et aux groupes de modèles.

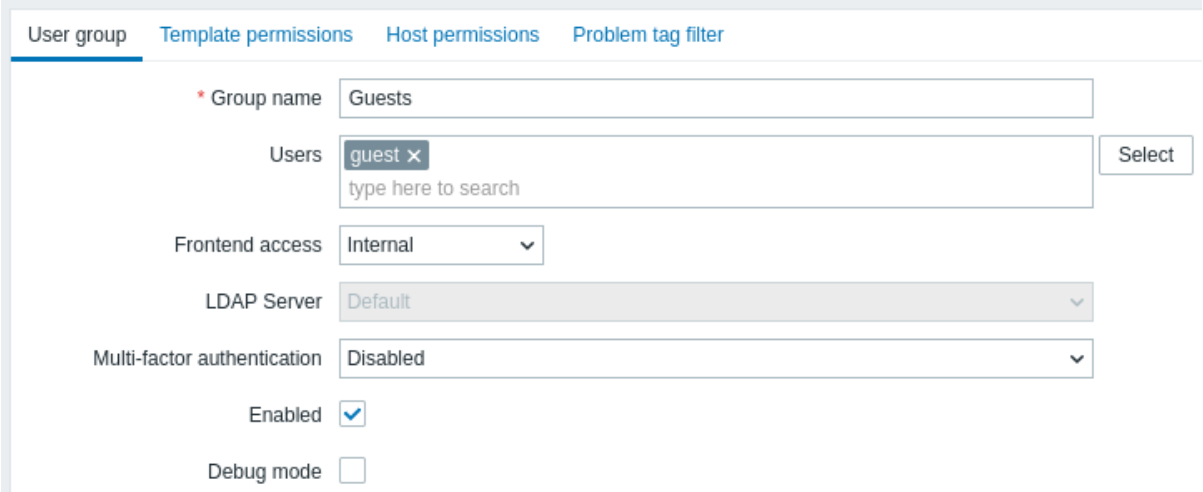
Un utilisateur peut appartenir à un nombre quelconque de groupes.

Configuration

Pour configurer un groupe d'utilisateurs :

- Allez dans *Utilisateurs* > *Groupes d'utilisateurs*
- Cliquez sur *Créer un groupe d'utilisateurs* (ou sur le nom du groupe pour modifier un groupe existant)
- Modifiez les attributs du groupe dans le formulaire

L'onglet **Groupe d'utilisateurs** contient les attributs généraux du groupe :

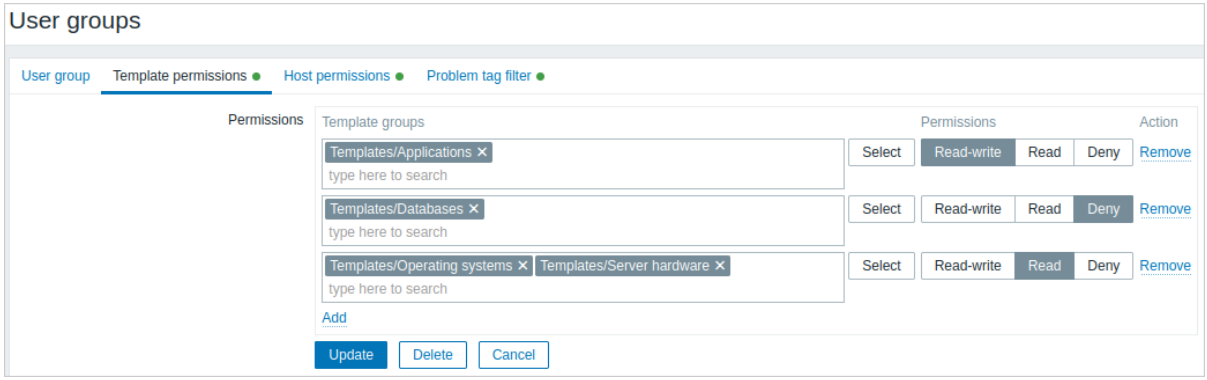


Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge.

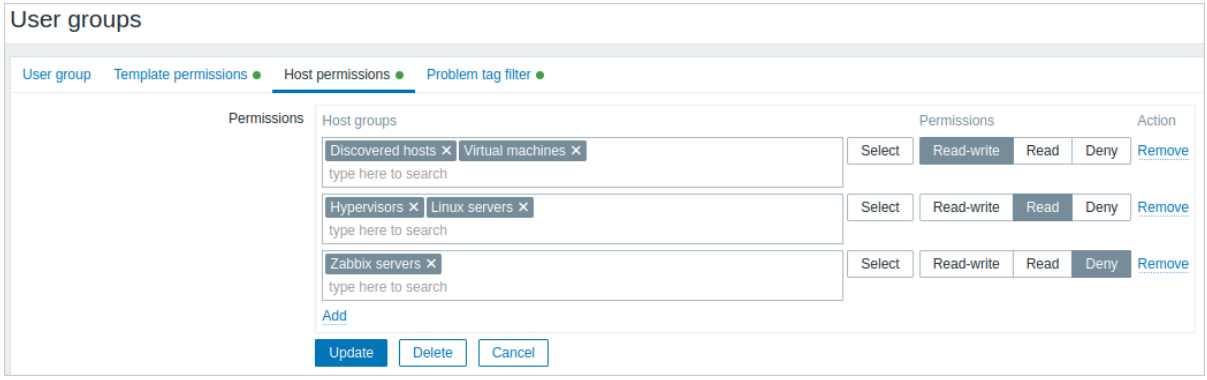
Parameter	Description
Nom du groupe	Nom de groupe unique.
Utilisateurs	Pour ajouter des utilisateurs au groupe, commencez à saisir le nom d'un utilisateur existant. Lorsque la liste déroulante avec les noms d'utilisateurs correspondants apparaît, faites défiler pour sélectionner. Vous pouvez également cliquer sur le bouton *Sélectionner* pour choisir des utilisateurs dans une fenêtre contextuelle.
Accès au frontend	Définit comment les utilisateurs du groupe sont authentifiés. **Par défaut du système** - utiliser la méthode d'authentification par défaut (définie globalement) **Interne** - utiliser l'authentification interne de Zabbix (même si l'authentification LDAP est utilisée globalement). Ignoré si l'authentification HTTP est la valeur par défaut globale. **LDAP** - utiliser l'authentification LDAP (même si l'authentification interne est utilisée globalement). Ignoré si l'authentification HTTP est la valeur par défaut globale. **Désactivé** - l'accès au frontend Zabbix est interdit pour ce groupe
Serveur LDAP	Sélectionnez quel serveur LDAP utiliser pour authentifier l'utilisateur. Ce champ est activé uniquement si *Accès au frontend* est défini sur LDAP ou Par défaut du système.
Authentification multifacteur	Sélectionnez quelle méthode d'authentification multifacteur utiliser pour authentifier l'utilisateur : **Par défaut** - utiliser la méthode définie par défaut dans la configuration MFA ; cette option est sélectionnée par défaut pour les nouveaux groupes d'utilisateurs si MFA est activée ; **<Nom de la méthode>** - utiliser la méthode sélectionnée (par exemple, "Zabbix TOTP") ; **Désactivé** - MFA est désactivée pour ce groupe ; cette option est sélectionnée par défaut pour les nouveaux groupes d'utilisateurs si MFA est désactivée. Notez que si un utilisateur appartient à plusieurs groupes d'utilisateurs avec MFA activée (ou si au moins un groupe a MFA activée), les règles d'authentification suivantes s'appliquent : si un groupe utilise la méthode MFA "Par défaut", elle servira à authentifier l'utilisateur ; sinon, la première méthode (par ordre alphabétique) sera utilisée pour l'authentification.

Parameter	Description
Activé	Statut du groupe d'utilisateurs et des membres du groupe. *Coché* - le groupe d'utilisateurs et les utilisateurs sont activés *Décoché* - le groupe d'utilisateurs et les utilisateurs sont désactivés
Mode de débogage	Cochez cette case pour activer le mode de débogage pour les utilisateurs.

L'onglet **Autorisations des modèles** permet de spécifier l'accès du groupe d'utilisateurs aux données des groupes de modèles (et donc des modèles) :



L'onglet **Autorisations des hôtes** permet de spécifier l'accès du groupe d'utilisateurs aux données des groupes d'hôtes (et donc des hôtes) :



Cliquez sur Add pour choisir les groupes de modèles/d'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 choisis. 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 à un groupe refusé

Si le même groupe de modèles/d'hôtes est ajouté sur plusieurs lignes avec des autorisations différentes, l'autorisation la plus restrictive sera appliquée.

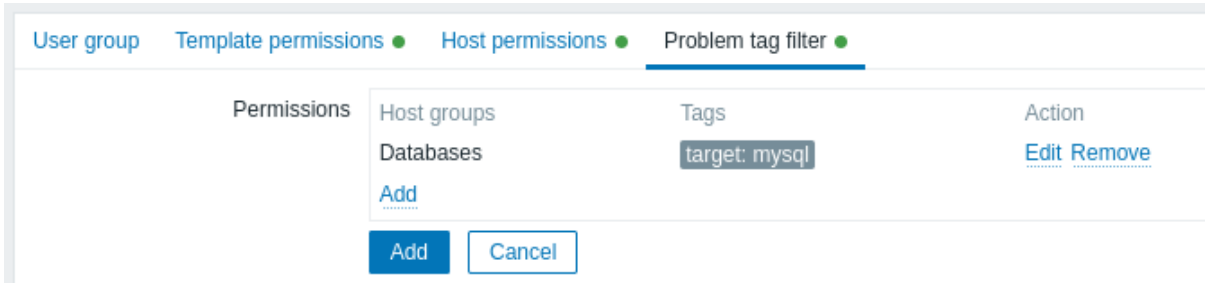
Notez qu'un utilisateur *Super admin* peut imposer aux groupes imbriqués d'avoir le même niveau d'autorisations que le groupe parent ; cela peut être fait dans le formulaire de configuration du groupe d'hôtes/de modèles.

Les onglets **Autorisations des modèles** et **Autorisations des hôtes** prennent en charge le même ensemble de paramètres.

Les autorisations actuelles sur les groupes sont affichées dans le bloc *Autorisations*, et elles peuvent être modifiées ou supprimées.

Note:
Si un groupe d'utilisateurs dispose des autorisations **Lecture-écriture** sur un hôte et des autorisations **Refuser** ou d'aucune autorisation sur un modèle lié à cet hôte, les utilisateurs de ce groupe ne pourront pas modifier les éléments hérités du modèle sur l'hôte, et le nom du modèle sera affiché comme *Modèle inaccessible*.

L'onglet **Filtre des tags de problème** permet de définir, pour les groupes d'utilisateurs, des autorisations basées sur les tags afin de voir les problèmes filtrés par nom et valeur de tag :

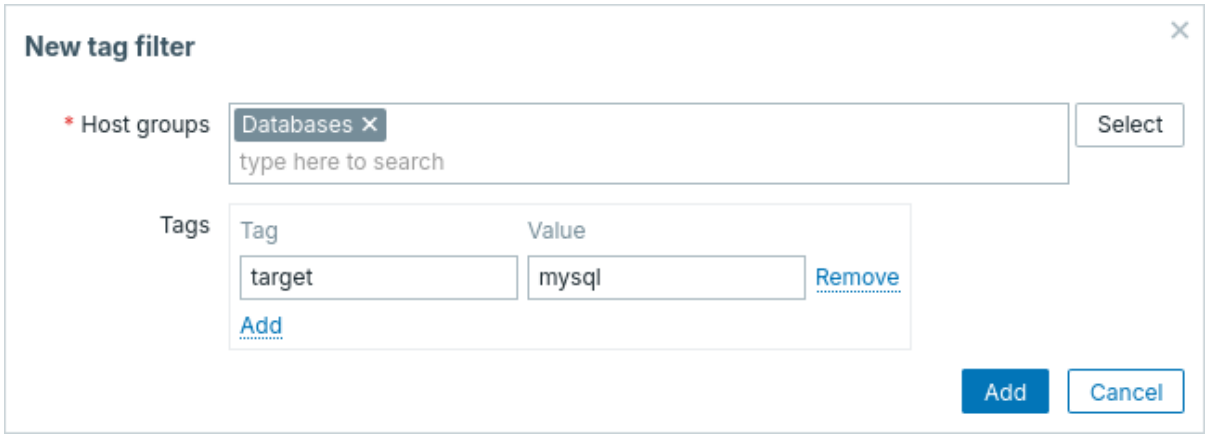


Cliquez sur Add pour choisir les groupes d'hôtes. Pour sélectionner un groupe d'hôtes auquel appliquer un filtre de tags, cliquez sur *Sélectionner* pour obtenir la liste complète des groupes d'hôtes existants ou commencez à saisir le nom d'un groupe d'hôtes pour obtenir une liste déroulante des groupes correspondants. Seuls les groupes d'hôtes seront affichés, car le filtre des tags de problème ne peut pas être appliqué aux groupes de modèles.

Il est ensuite possible de passer de *Tous les tags* à *Liste des tags* afin de définir des tags particuliers et leurs valeurs pour le filtrage. Des noms de tags sans valeurs peuvent être ajoutés, mais des valeurs sans noms ne le peuvent pas. Seuls les trois premiers tags (avec leurs valeurs, le cas échéant) sont affichés dans le bloc *Autorisations* ; s'il y en a davantage, ils peuvent être consultés en cliquant sur l'icône ••• ou en la survolant.

Le filtre des tags permet de dissocier 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 de base de données "MySQL", il faut d'abord créer un groupe d'utilisateurs pour les administrateurs de base de données, puis spécifier le nom de tag "target" et la valeur "mysql".



Si le nom de tag "target" est spécifié et que le champ de valeur est laissé vide, le groupe d'utilisateurs verra tous les problèmes avec le nom de tag "target" pour le groupe d'hôtes sélectionné. Si *Tous les tags* est sélectionné, le groupe d'utilisateurs verra tous les problèmes pour le groupe d'hôtes spécifié.

Assurez-vous que le nom du tag et la valeur du tag sont correctement spécifiés ; sinon, le groupe d'utilisateurs ne verra aucun problème.

Examinons un exemple où un utilisateur est membre de plusieurs groupes d'utilisateurs sélectionnés. Dans ce cas, le filtrage utilisera une condition OR pour les tags.

Groupe d'utilisateurs A			**Groupe d'utilisateurs B**			**Résultat visible pour un utilisateur (membre) des deux groupes**
Filtre de tags						
Groupe d'hôtes	*Nom du tag*	*Valeur du tag*	*Groupe d'hôtes*	*Nom du tag*	*Valeur du tag*	
Databases	target	mysql	Databases	target	oracle	problèmes target:mysql ou target:oracle visibles
Databases	défini sur : *Tous les tags*		Databases	target	oracle	Tous les problèmes visibles

Non configuré dans le **Filtre des tags de problème**	Databases	target	oracle	problèmes target:oracle visibles

Attention:
L'ajout d'un filtre (par exemple, tous les tags dans un certain groupe d'hôtes "Databases") a pour conséquence qu'il n'est plus possible de voir les problèmes des autres groupes d'hôtes.

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 ».

13 Stockage des secrets

Aperçu

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 :

- les valeurs de macros utilisateur
- les identifiants d'accès à la base de données

Zabbix fournit un accès en lecture seule aux secrets dans un coffre-fort, en partant du principe que les secrets sont gérés par quelqu'un d'autre.

For information about specific vault provider configuration, see: - HashiCorp configuration - CyberArk configuration

Mise en cache des valeurs secrètes

Par défaut, les valeurs des macros secrètes du coffre-fort 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 coffre-fort depuis le serveur Zabbix à 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 consigné dans le journal.

Il est également possible de configurer la récupération indépendante des valeurs de macros par le serveur Zabbix et le proxy Zabbix.

Pour déclencher manuellement l'actualisation des valeurs secrètes mises en cache depuis un coffre-fort, utilisez l'option de ligne de commande 'secrets_reload'.

Pour les identifiants de base de données du frontend Zabbix, la mise en cache est désactivée par défaut, mais peut être activée en définissant l'option `$DB['VAULT_CACHE'] = true` dans zabbix.conf.php. Les identifiants seront stockés dans un cache local à l'aide du 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, indiquez le répertoire dans le paramètre de configuration SSLCALocation du serveur/proxy Zabbix, placez le fichier de certificat dans ce répertoire, puis exécutez la commande CLI :

```
c_rehash .
```

1 Configuration de 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 de CyberArk.

Pour en savoir plus sur la configuration de TLS dans Zabbix, consultez *Stockage des secrets*.

Identifiants de base de données

L'accès à un secret contenant les identifiants de base de données est configuré séparément pour chaque composant Zabbix.

Serveur et proxys

Pour obtenir les identifiants de base de données depuis le coffre-fort pour le server ou le proxy Zabbix, spécifiez les paramètres de configuration suivants dans le fichier de configuration :

- `Vault` - indique quel fournisseur de coffre-fort doit être utilisé ;
- `VaultURL` - URL HTTP[S] du serveur de coffre-fort ;
- `VaultDBPath` - requête vers le secret du coffre-fort contenant les identifiants de 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 SSL et de clé ; 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 secret par le serveur Zabbix (et le proxy Zabbix, s'il est configuré). Le serveur Zabbix et les proxys Zabbix n'ouvriront pas les macros de secret du coffre-fort qui contiennent des identifiants de base de données provenant de VaultDBPath.

Le serveur Zabbix et le proxy Zabbix lisent les paramètres de configuration liés au coffre-fort depuis les fichiers *zabbix_server.conf* et *zabbix_proxy.conf* au démarrage.

Exemple

1. Dans *zabbix_server.conf*, indiquez 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>
}
```

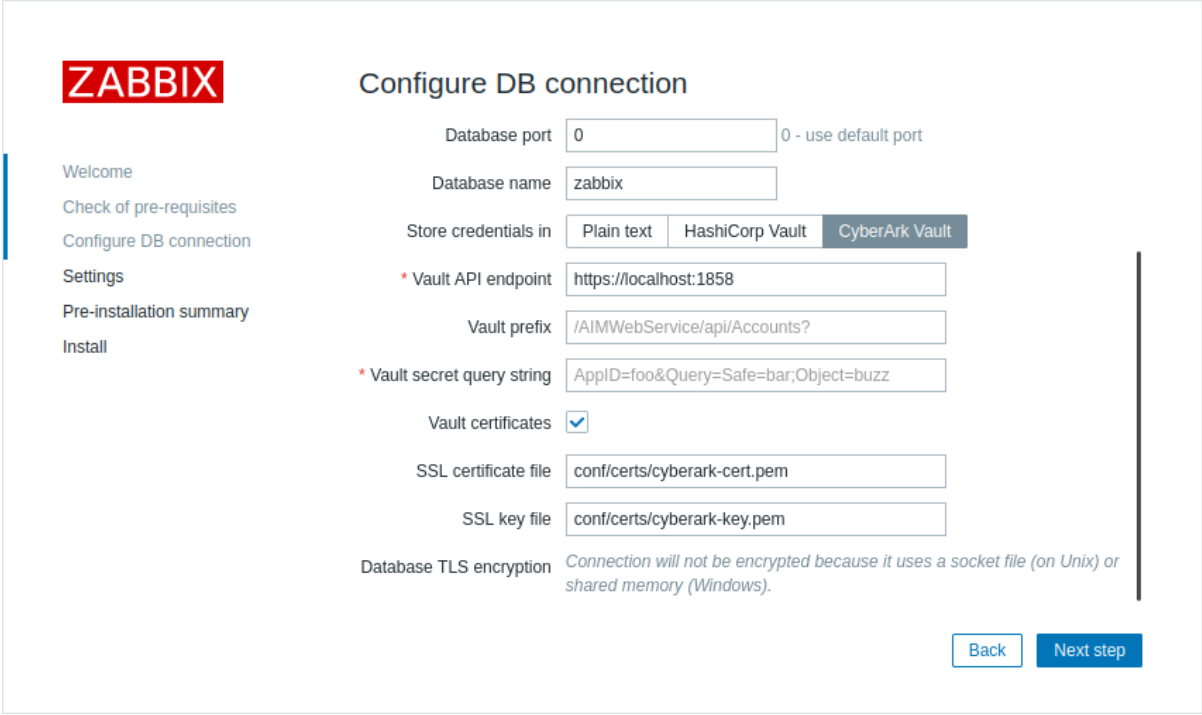
4. Par conséquent, Zabbix utilisera les informations d'identification suivantes pour l'authentification à la base de données :

- Nom d'utilisateur : <username>
- Mot de passe : <password>

Interface web

Pour obtenir les identifiants de la base de données depuis le coffre-fort pour l'interface web Zabbix, spécifiez les paramètres suivants lors de l'installation de l'interface web.

1. À l'étape *Configurer la connexion à la base de données*, définissez le paramètre *Stocker les identifiants dans* sur « CyberArk Vault ».



2. Ensuite, renseignez les paramètres supplémentaires :

Paramètre	Obligatoire	Valeur par défaut	Description
Point de terminaison de l'API du coffre-fort	oui	https://localhost:1858	Spécifiez l'URL de connexion au coffre-fort au format `scheme://host:port`

Paramètre	Obligatoire	Valeur par défaut	Description
Préfixe du coffre-fort	non	/AIMWebService/api/Accounts?	Fournissez un préfixe personnalisé pour le chemin ou la requête du coffre-fort. S'il n'est pas spécifié, la valeur par défaut est utilisée.
Chaîne de requête du secret du coffre-fort	oui		Une requête qui indique d'où les identifiants de la base de données doivent être récupérés. Exemple : `AppID=foo&Query=Safe=bar;Object=buzz`
Certificats du coffre-fort	non		Après avoir coché la case, des paramètres supplémentaires apparaîtront pour permettre de configurer l'authentification du client. Bien que ce paramètre soit facultatif, il est fortement recommandé de l'activer pour la communication avec CyberArk Vault.
Fichier de certificat SSL	non	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 du fichier de clé SSL vide.
Fichier de clé SSL	non	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 pour 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) doit avoir accès aux valeurs des macros *Vault secret* depuis le coffre-fort. Le frontend 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 des informations détaillées sur le traitement des valeurs de macro par Zabbix.

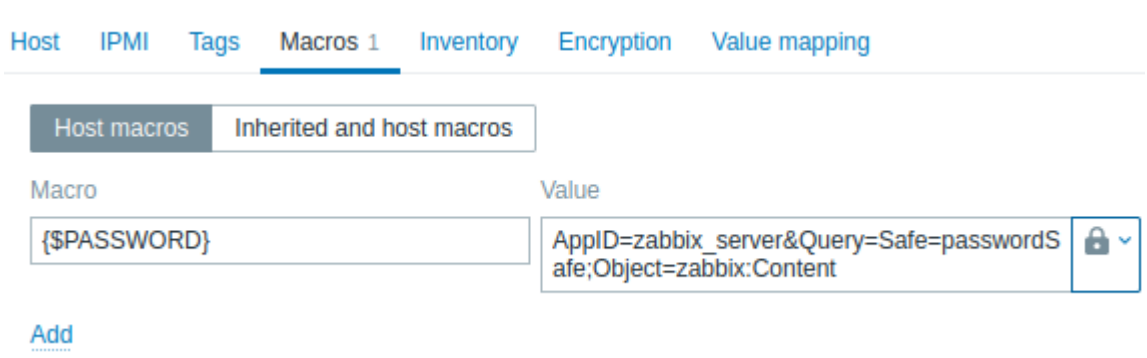
Syntaxe de requête

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 en URL ("/" est encodé en "%2F", ":" est encodé en "%3A").

Exemple

1. Dans Zabbix, ajoutez une macro utilisateur {$PASSWORD} de type *Vault secret* avec la valeur `AppID=zabbix_server&Query=Safe=p`



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 coffre 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 de connexion à la base de données en reconfigurant le frontend Zabbix et en spécifiant les paramètres requis comme décrit dans la section *Frontend*. Pour reconfigurer le frontend Zabbix, ouvrez l'URL de configuration du frontend dans le navigateur :

- pour Apache : http://<server_ip_or_name>/zabbix/setup.php
- pour Nginx : http://<server_ip_or_name>/setup.php

Sinon, ces paramètres peuvent être définis dans le fichier de configuration du frontend (*zabbix.conf.php*) :

```
$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 coffre HashiCorp Vault, consultez *Configuration HashiCorp*.

2 Configuration de HashiCorp

Aperçu

Cette section explique comment configurer Zabbix pour récupérer des secrets depuis le moteur de secrets KV de HashiCorp Vault - version 2.

Le coffre-fort 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 avec succès un secret contenant les identifiants de la base de données, il est nécessaire de configurer les deux éléments suivants :

- serveur/proxy Zabbix
- interface web Zabbix

Serveur/proxy

Pour configurer le serveur ou le proxy Zabbix, spécifiez les paramètres de configuration suivants dans le fichier de configuration :

- `Vault` - indique quel fournisseur de coffre-fort doit être utilisé ;

- `VaultToken` - jeton d'authentification du coffre-fort (voir le fichier de configuration du serveur/proxy Zabbix pour plus de détails) ;
- `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.

Attention:
Les paramètres de configuration `Vault`, `VaultToken`, `VaultURL` et `VaultPrefix` sont également utilisés pour l'authentification au coffre-fort lors du traitement des macros de coffre-fort secret par le serveur Zabbix (et le proxy Zabbix, s'il est configuré). Le serveur Zabbix et les proxys Zabbix n'ouvriront pas les macros de secret du coffre-fort contenant les identifiants de base de données provenant de VaultDBPath.

L'utilisation de jetons différents pour différents proxys 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 liront la variable d'environnement `VAULT_TOKEN` une seule fois au démarrage, puis la supprimeront afin qu'elle ne soit pas accessible via des 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 :

```
Vault=HashiCorp
VaultToken=hvs.CAESIIG_PILmULFYOsEyWHxkZ2mF2a8VPKNLE8eHqd4autYGGh4KHGh2cy5aeTY0NFNSaUp3ZnpWbDF1RUNjUkNTZEg
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 coffre :

```
#### Activez 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

#### Ajoutez 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>

#### Vérifiez que le secret a bien été ajouté.
vault kv get secret/zabbix/database

#### Enfin, testez 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. Par conséquent, le serveur Zabbix récupérera les informations d'identification suivantes pour l'authentification à la base de données :

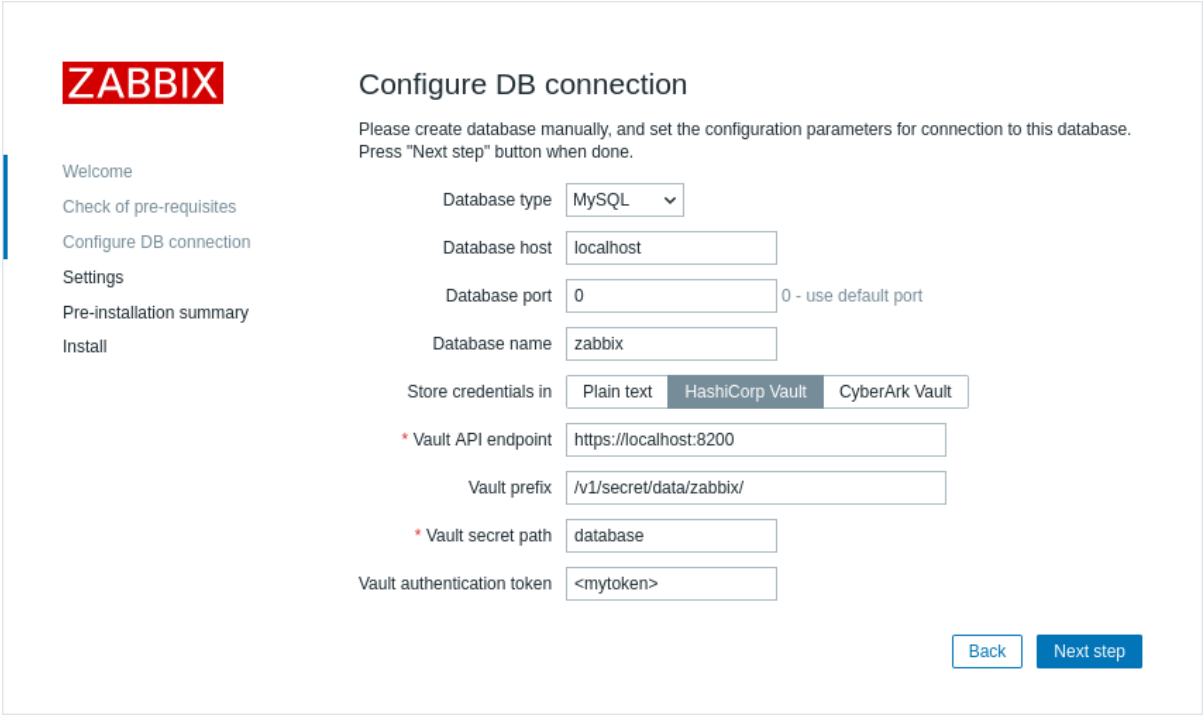
- Nom d'utilisateur : zabbix
- Mot de passe : <password>

Frontend

Le frontend Zabbix peut être configuré pour récupérer les identifiants de la base de données depuis le coffre-fort, soit lors de l'installation du frontend, soit en mettant à jour le fichier de configuration du frontend (`zabbix.conf.php`).

Attention:
Si les identifiants du coffre-fort ont été modifiés depuis la précédente installation du frontend, relancez l'installation du frontend ou mettez à jour `zabbix.conf.php`. Voir aussi : Mise à jour de la configuration existante.

Lors de l'**installation du frontend**, les paramètres de configuration doivent être spécifiés à l'étape *Configurer la connexion à la base de données* :



- Définissez le paramètre *Stocker les identifiants dans* sur « HashiCorp Vault ».
- Spécifiez les paramètres de connexion :

Paramètre	Obligatoire	Valeur par défaut	Description
Point de terminaison de l'API Vault	oui	https://localhost:8200	Indiquez l'URL de connexion au coffre-fort au format `scheme://host:port`
Préfixe Vault	non	/v1/secret/data/	Fournissez 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/`
Chemin du secret Vault	non		Chemin vers le secret à partir duquel les identifiants de la base de données seront récupérés à l'aide des clés « password » et « username ». Exemple : `database`
Jeton d'authentification Vault	non		Fournissez un jeton d'authentification avec un accès en lecture seule au chemin du secret. Consultez la documentation HashiCorp pour plus d'informations sur la création de jetons et les politiques 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 :

- le serveur/proxy Zabbix est configuré pour fonctionner avec HashiCorp Vault ;
- le paramètre *Vault provider* dans *Administration → General → Other* est défini sur « HashiCorp Vault » (par défaut) ;



Note:
Le serveur Zabbix (et le proxy Zabbix, si configuré) nécessitent un accès aux valeurs des macros *Vault secret* depuis le coffre-fort. Le frontend 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 du serveur/proxy Zabbix (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 macros par Zabbix.

Syntaxe du chemin

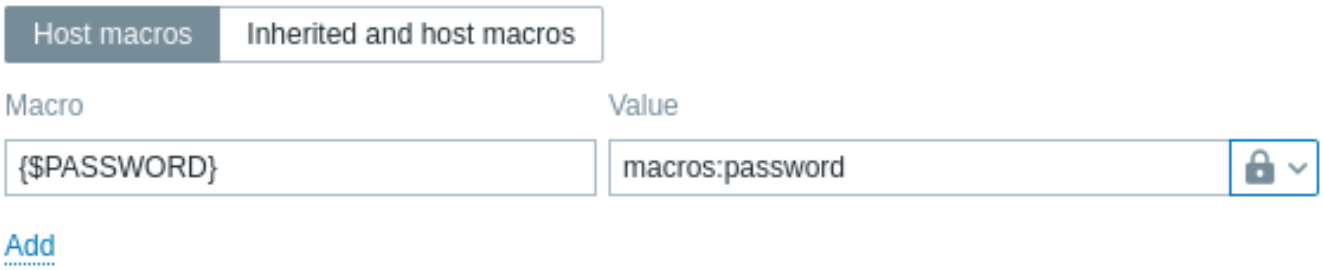
Les symboles barre oblique ("/") et deux-points (":") sont réservés.

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/une 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 est séparé par une barre oblique (par exemple, *foo/bar/zabbix*, où le point de montage est « foo/bar » et le chemin est « zabbix », peut être encodé sous la forme « foo%2Fbar/zabbix ») et si un nom de point de montage ou un chemin doit contenir un deux-points.

Exemple

1. Dans Zabbix, ajoutez une macro utilisateur {$PASSWORD} de type « Vault secret » avec la valeur `macros:password`



2. Exécutez les commandes CLI suivantes pour créer le secret requis dans le coffre :

```
#### Activez 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

#### Ajoutez 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>

#### Vérifiez que le secret a bien été ajouté.
vault kv get secret/zabbix/macros

#### Enfin, testez 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. Par conséquent, Zabbix résoudra la macro {$PASSWORD} à 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 base de données*.
2. Mettez à jour les paramètres de connexion à la base de données en reconfigurant le frontend Zabbix et en spécifiant les paramètres requis comme décrit dans la section *Frontend*. Pour reconfigurer le frontend Zabbix, ouvrez l'URL de configuration du frontend dans le navigateur :

- pour Apache : http://<server_ip_or_name>/zabbix/setup.php
- pour Nginx : http://<server_ip_or_name>/setup.php

Vous pouvez également définir ces paramètres dans le fichier de configuration du frontend (*zabbix.conf.php*) :

```
$DB['VAULT']                    = 'HashiCorp';
$DB['VAULT_URL']                = 'https://localhost:8200';
$DB['VAULT_DB_PATH']            = 'database';
$DB['VAULT_TOKEN']              = '<mytoken>';
$DB['VAULT_CERT_FILE']          = '';
```

```
$DB['VAULT_KEY_FILE']           = '';
$DB['VAULT_PREFIX']             = '/v1/secret/data/zabbix/';
```

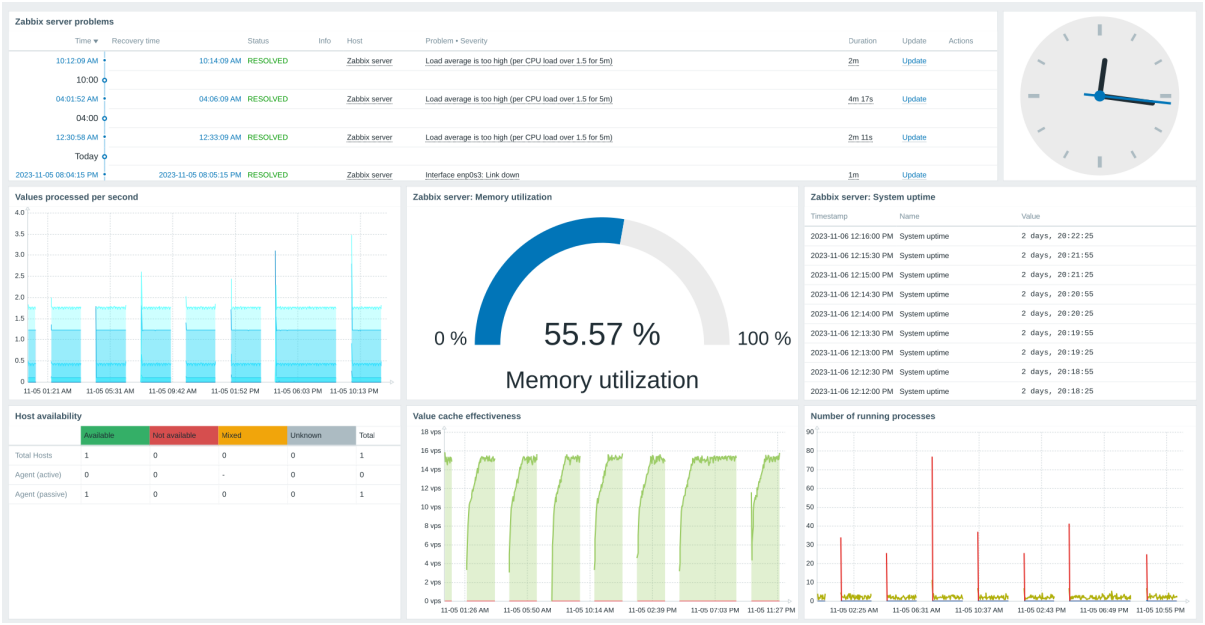
3. Configurez les macros utilisateur comme décrit dans la section *Valeurs des macros utilisateur*, si nécessaire.

Pour mettre à jour une configuration existante afin de récupérer des secrets depuis un CyberArk Vault, voir *Configuration CyberArk*.

14 Rapports planifiés

Aperçu

Avec la fonctionnalité *Rapports planifiés*, vous pouvez configurer l'envoi, à intervalles réguliers, d'une version PDF d'un tableau de bord donné à des destinataires spécifiés.



Prérequis :

- Le service web Zabbix doit être installé et correctement configuré pour permettre la génération de rapports planifiés - voir Configuration des rapports planifiés pour les instructions.
- Un utilisateur doit avoir un rôle utilisateur de type *Admin* ou *Super admin* avec les autorisations suivantes :
  - *Rapports planifiés* dans le bloc *Accès aux éléments de l'interface utilisateur* (pour afficher les paramètres du rapport)
  - *Gérer les rapports planifiés* dans le bloc *Accès aux actions* (pour créer/modifier des rapports)

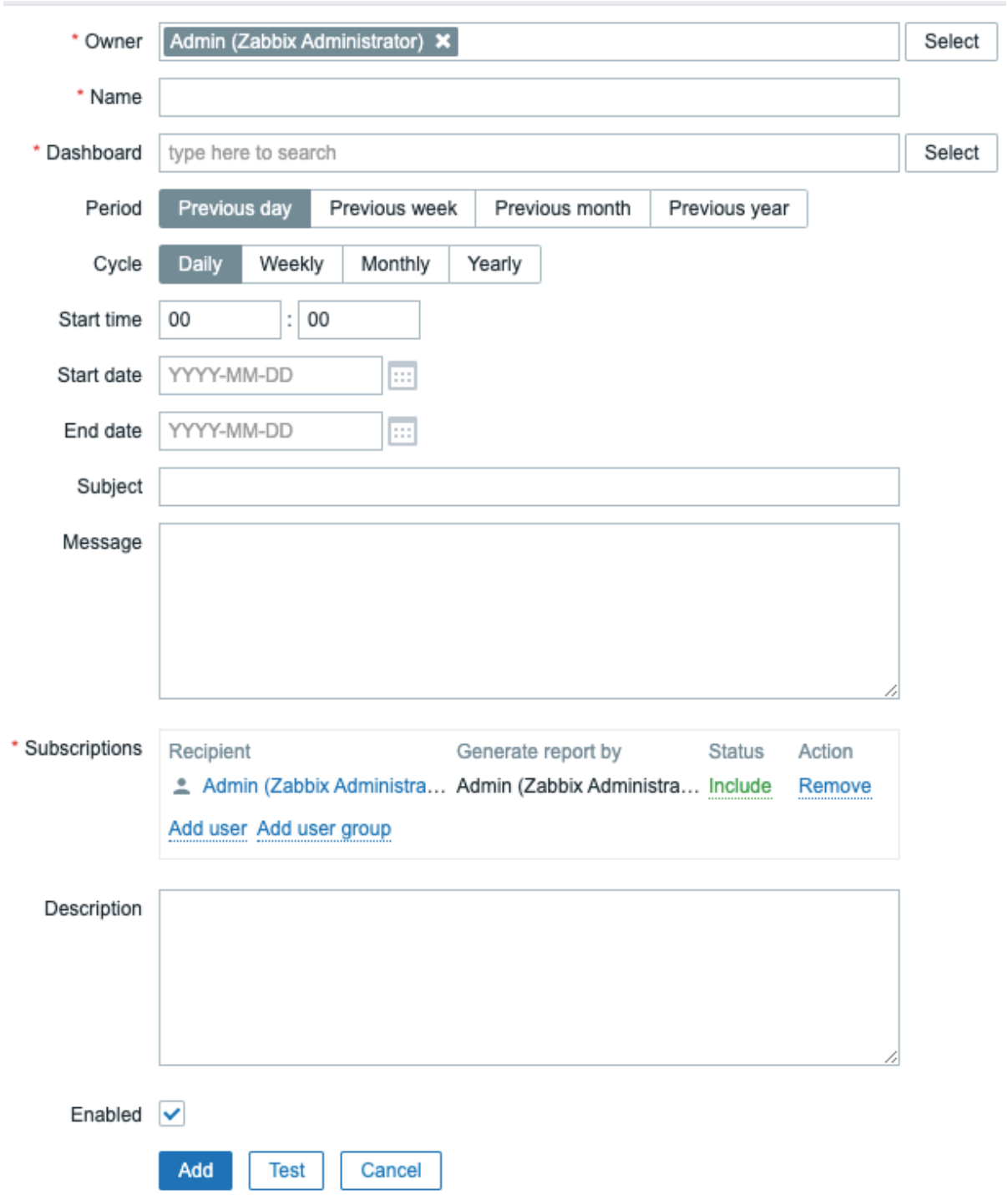
Pour créer un rapport planifié dans l'interface Zabbix, procédez comme suit :

- Allez à : *Rapports* > *Rapports planifiés*.
- Cliquez sur *Créer un rapport* dans le coin supérieur droit de l'écran.
- 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
Owner	Utilisateur qui crée un rapport. Les utilisateurs de niveau *Super admin* sont autorisés à modifier le propriétaire. Pour les utilisateurs de niveau *Admin*, ce champ est en lecture seule.
Name	Nom du rapport ; il doit être unique.
Dashboard	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 apparaîtra ; faites défiler pour sélectionner. Vous pouvez également cliquer sur *Select* à côté du champ et sélectionner un tableau de bord dans la liste affichée.
Period	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 « Weekly » permet de sélectionner les jours de la semaine auxquels le rapport sera envoyé.
Start time	Heure de la journée au format hh:mm à laquelle le rapport sera préparé. Notez que le fuseau horaire du serveur Zabbix sera utilisé.

Parameter	Description
Repeat on	Jours de la semaine auxquels le rapport sera envoyé. Ce champ n'est disponible que si *Cycle* est défini sur « Weekly ».
Start date	Date à laquelle la génération régulière du rapport doit commencer.
End date	Date à laquelle la génération régulière du rapport doit s'arrêter.
Subject	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}.
Subscriptions	Liste des destinataires du rapport. Par défaut, elle inclut uniquement le propriétaire du rapport. Tout utilisateur Zabbix ayant un média e-mail configuré peut être spécifié comme destinataire du rapport. Cliquez sur *Add user* ou *Add user group* pour ajouter d'autres destinataires. Cliquez sur le nom d'utilisateur pour modifier les paramètres : *Generate report by* - indique si les données du rapport doivent être générées en fonction des autorisations du tableau de bord de l'utilisateur actuel ou du destinataire. *Status* - sélectionnez « Include » pour envoyer le rapport à l'utilisateur ou « Exclude » pour empêcher l'envoi du rapport à cet utilisateur. Au moins un utilisateur doit avoir le statut « Include ». Le statut « Exclude » 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 *Admin* qui ne sont pas membres du même groupe d'utilisateurs que le destinataire ou le propriétaire du rapport) verront « Inaccessible user » ou « Inaccessible user group » au lieu des noms réels dans les champs *Recipient* et *Generate report by* ; les champs *Status* et *Action* seront affichés en lecture seule.
Enabled	Statut du rapport. Décocher cette case désactivera 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 de formulaire

Des boutons en bas du formulaire permettent d'effectuer plusieurs opérations.

Add	Ajouter un rapport. Ce bouton n'est disponible que pour les nouveaux rapports.
Update	Mettre à jour les propriétés d'un rapport.
Clone	Créer un autre rapport basé sur les 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.

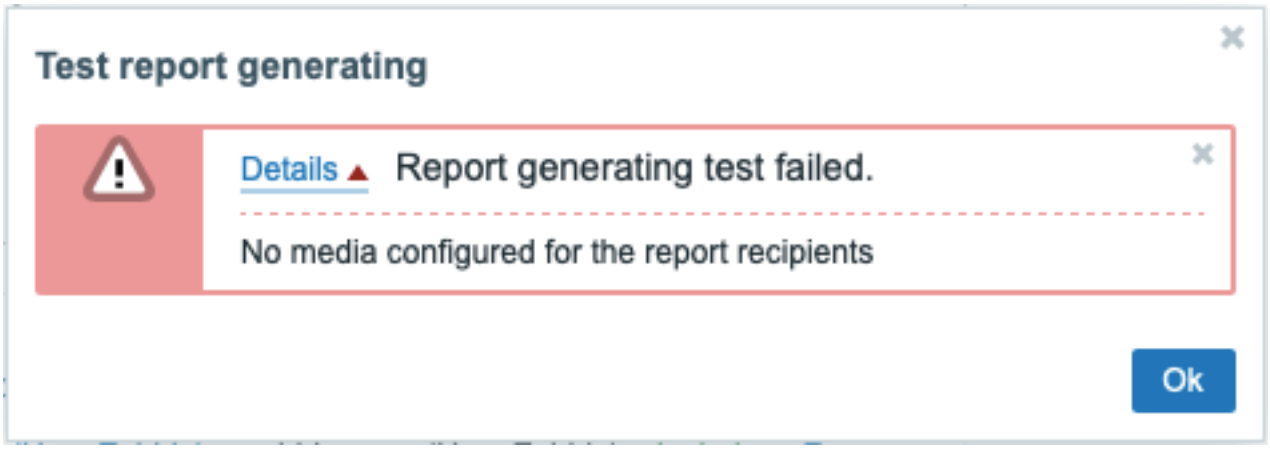
Test

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'action du tableau de bord.

Si la configuration est correcte, le rapport de test est envoyé immédiatement à l'utilisateur actuel. Pour les rapports de test, les abonnés et les paramètres utilisateur *Generate report by* sont ignorés.

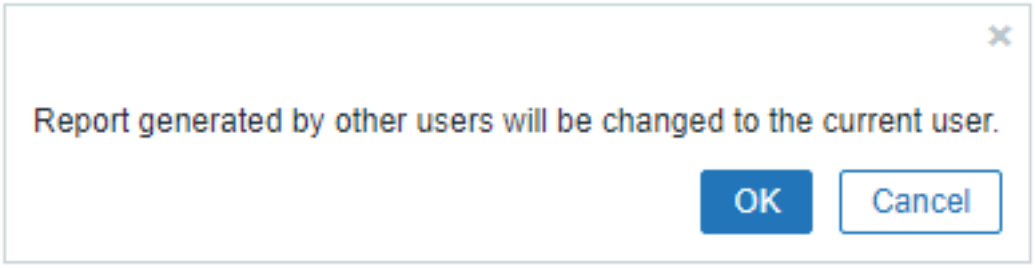
Si la configuration est incorrecte, un message d'erreur s'affiche, décrivant la cause possible.



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*.

15 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)

1 Exporter vers des fichiers

Aperçu

Il est possible de configurer l'exportation en temps réel des événements de déclencheur, des valeurs d'élément et des tendances dans un format JSON délimité par des retours à la ligne.

L'exportation s'effectue dans des fichiers, où chaque ligne du fichier d'exportation 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'exportation, ou le fichier d'exportation 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'exportation. Il est écrit uniquement dans la base de données Zabbix. L'écriture des données dans le fichier d'exportation reprend lorsque le problème d'écriture est résolu.

Le fichier d'exportation est créé avec des autorisations de lecture et d'écriture pour le propriétaire du fichier uniquement. De plus, le fichier est lisible par le groupe propriétaire. Toutes les autres autorisations sont refusées.

Pour des détails précis sur les informations exportées, consultez la page du protocole d'exportation.

Notez qu'un hôte/élément peut ne pas avoir de métadonnées (groupes d'hôtes, nom d'hôte, nom d'élément) si l'hôte/l'élément a été supprimé après la réception des données, mais avant que le serveur n'exporte les données.

Configuration

L'exportation en temps réel des événements de déclencheur, des valeurs d'élément et des tendances est configurée en spécifiant un répertoire pour les fichiers d'exportation - voir 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 du fichier. Si elle dépasse la limite de taille configurée, le fichier est renommé en ajoutant .old à son nom et un nouveau fichier portant le nom d'origine est créé.

> **Attention:**
> Un fichier sera créé pour chaque processus qui écrira des données (c'est-à-dire environ 4 à 30 fichiers). Comme la taille par défaut de chaque fichier d'exportation est de 1G, conserver de gros fichiers d'exportation peut rapidement épuiser l'espace disque.

- `ExportType` permet de spécifier quels types d'entités (événements, historique, tendances) seront exportés.

2 Diffusion en continu vers des systèmes externes

Aperçu

Il est possible de diffuser en continu les valeurs des éléments et les événements de Zabbix vers des systèmes externes via HTTP (voir les détails du protocole).

Le filtre de tags peut être utilisé pour diffuser en continu des sous-ensembles de valeurs d'éléments ou d'événements.

Deux processus du serveur Zabbix sont responsables de la diffusion des données : `connector manager` et `connector worker`. Un élément interne Zabbix `zabbix[connector_queue]` permet de surveiller le nombre de valeurs mises en file d'attente dans la file du connecteur.

Configuration

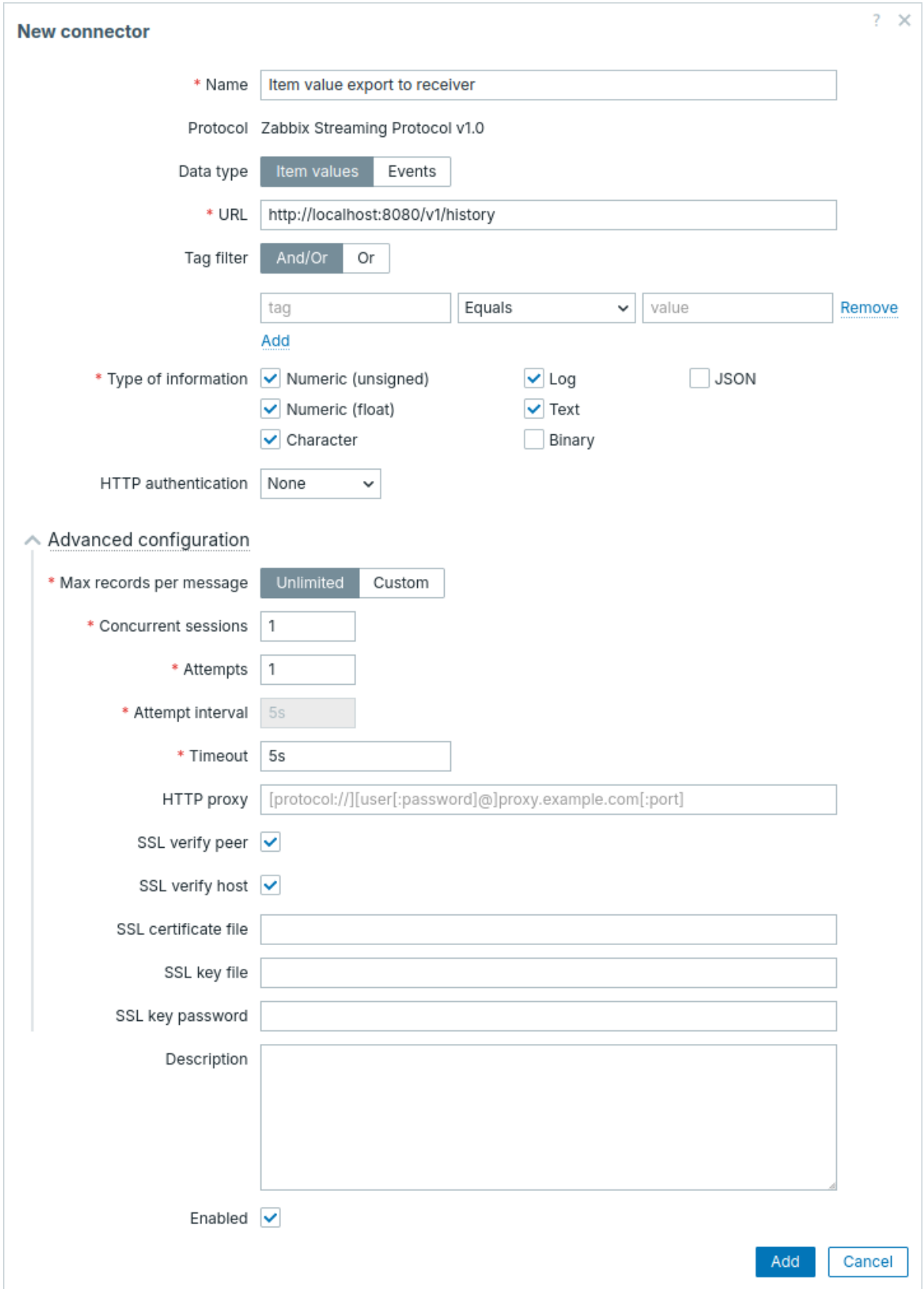
Les étapes suivantes sont nécessaires pour configurer le streaming de données vers un système externe :

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 Zabbix server - un serveur léger écrit en Go, conçu pour transférer les valeurs d'éléments et les événements d'un Zabbix server vers un broker Kafka.

2. Définissez 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 (ou être supérieur si le nombre de sessions simultanées est supérieur à 1) au nombre de connecteurs configuré dans le frontend Zabbix. Redémarrez ensuite Zabbix server.

3. Configurez un nouveau connecteur dans le frontend Zabbix (*Administration* > *General* > *Connectors*) et rechargez le cache du serveur avec la commande `zabbix_server -R config_cache_reload`.



Les champs obligatoires sont marqués d'un astérisque.

Parameter	Description
Name	Saisissez le nom du connecteur.
Data type	Sélectionnez le type de données à diffuser : **Item values** - diffuser les valeurs d'éléments de Zabbix vers des systèmes externes ; **Events** - diffuser les événements de Zabbix vers des systèmes externes.
URL	Saisissez l'URL du récepteur. Les macros utilisateur sont prises en charge.
Tag filter	Exportez uniquement les valeurs d'éléments ou les événements correspondant au filtre de tags. S'il n'est pas défini, tout est exporté. 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.
Type of information	Sélectionnez le type d'information (numeric (unsigned), numeric (float), character, etc.) permettant de filtrer les valeurs d'éléments que le connecteur doit diffuser. Ce champ est disponible si *Data type* est défini sur "Item values".
HTTP authentication	Sélectionnez l'option d'authentification : **None** - aucune authentification n'est utilisée ; **Basic** - l'authentification de base est utilisée ; **NTLM** - l'authentification NTLM (Windows NT LAN Manager) est utilisée ; **Kerberos** - l'authentification Kerberos est utilisée (voir aussi : Configuration de Kerberos avec Zabbix) ; **Digest** - l'authentification Digest est utilisée ; **Bearer** - l'authentification Bearer est utilisée.
Username	Saisissez le nom d'utilisateur (jusqu'à 255 caractères). Les macros utilisateur sont prises en charge. Ce champ est disponible si *HTTP authentication* est défini sur "Basic", "NTLM", "Kerberos" ou "Digest".
Password	Saisissez le mot de passe de l'utilisateur (jusqu'à 255 caractères). Les macros utilisateur sont prises en charge. Ce champ est disponible si *HTTP authentication* est défini sur "Basic", "NTLM", "Kerberos" ou "Digest".
Bearer token	Saisissez le jeton Bearer. Les macros utilisateur sont prises en charge. Ce champ est disponible et obligatoire si *HTTP authentication* est défini sur "Bearer".
Advanced configuration	Cliquez sur le libellé *Advanced configuration* pour afficher les options de configuration avancée (voir ci-dessous).
Max records per message	Spécifiez le nombre maximal de valeurs ou d'événements pouvant être diffusés dans un seul message.
Concurrent sessions	Sélectionnez le nombre de processus d'envoi à exécuter pour ce connecteur. Jusqu'à 100 sessions peuvent être spécifiées ; la valeur par défaut est "1".
Attempts	Nombre de tentatives de diffusion des données. Jusqu'à 5 tentatives peuvent être spécifiées ; la valeur par défaut est "1".
Attempt interval	Spécifiez combien de temps le connecteur doit attendre après une tentative infructueuse de diffusion des données. Jusqu'à 10s peuvent être spécifiées ; la valeur par défaut est "5s". Ce champ est disponible si *Attempts* est défini sur "2" ou plus. Les tentatives infructueuses sont celles où l'établissement d'une connexion a échoué, ou lorsque le code de réponse HTTP n'est pas 200, 201, 202, 203, 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, 422. Les redirections sont suivies ; ainsi, 302 -> 200 est une réponse positive, tandis que 302 -> 503 déclenchera une nouvelle tentative.

Parameter	Description
Timeout	Spécifiez 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.
HTTP proxy	Vous pouvez spécifier un proxy HTTP à utiliser 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 sera traité comme un proxy HTTP. Par défaut, le port 1080 sera utilisé. Si *HTTP proxy* est spécifié, le proxy remplacera les variables d'environnement liées au proxy telles que `http_proxy`, `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 aucun contrôle de validité. 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 le proxy HTTP.
SSL verify peer	Cochez la case pour vérifier le certificat SSL du serveur web. Le certificat du serveur sera automatiquement pris à partir de l'emplacement système des autorités de certification (CA). Vous pouvez remplacer l'emplacement des fichiers CA à l'aide du paramètre de configuration Zabbix server ou proxy `SSLCALocation`.
SSL verify host	Cochez la case pour vérifier que le champ *Common Name* ou le champ *Subject Alternate Name* du certificat du serveur web correspond. Cela définit l'option cURL `CURLOPT_SSL_VERIFYHOST`.
SSL certificate file	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, laissez le champ *SSL key file* vide. Si la clé est chiffrée, indiquez le mot de passe dans le champ *SSL key password*. Le répertoire contenant ce fichier est spécifié par le paramètre de configuration Zabbix server ou proxy `SSLCertLocation`.
SSL key file	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 Zabbix server ou proxy `SSLKeyLocation`.
SSL key password	Mot de passe du fichier de clé privée SSL. Les macros utilisateur sont prises en charge.
Description	Saisissez la description du connecteur.
Enabled	Cochez la case pour activer le connecteur.

Lorsque le connecteur Kafka est configuré avec une liste d'adresses de brokers bootstrap 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 IP peuvent être résolus vers des adresses de bouclage (par exemple `127.0.0.1/localhost`) ou être normalisés par le client, ce qui peut rendre ces avertissements trompeurs. Pour réduire toute 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 récepteur s'effectue via HTTP en utilisant l'API REST, NDJSON, « Content-Type: application/x-ndjson ».

Pour plus de détails, voir Protocole d'export JSON délimité par des sauts de ligne.

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, toute autre valeur indiquant des requêtes en échec.

Exemple de réussite :

```
localhost:8080/v1/history": HTTP/1.1 200 OK
Date: Wed, 11 Jan 2023 16:40:30 GMT
Content-Length: 0
```

Exemple avec des erreurs :

```
localhost:8080/v1/history": HTTP/1.1 422 Unprocessable Entity
Content-Type: application/json
Date: Wed, 11 Jan 2023 17:07:36 GMT
Content-Length: 55

{"error":"invalid character '{' after top-level value"}
```

3 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`)

6 Surveillance des services

Vue d'ensemble La supervision des services est une supervision au niveau métier qui peut être utilisée pour obtenir une vue d'ensemble de l'arborescence complète des services de l'infrastructure IT, identifier les points faibles de l'infrastructure, calculer les SLA de divers services IT et consulter d'autres informations à un niveau plus élevé. La supervision des services se concentre sur la disponibilité globale d'un service plutôt que sur des détails de bas niveau, tels que le manque d'espace disque, une charge processeur élevée, etc. La supervision des services fournit également des fonctionnalités permettant de trouver la cause racine d'un problème si un service ne fonctionne pas comme prévu.

La supervision des services permet de créer une représentation hiérarchique des données supervisées.

Une structure de service très simple peut ressembler à ceci :

```
Service
|
|-Postes de travail
| |
| |-Poste de travail1
| |
| |-Poste de travail2
```

```
|
|-Serveurs
```

Chaque nœud de la structure possède un attribut de statut. Le statut est calculé et propagé vers les niveaux supérieurs selon l'algorithme sélectionné. Le statut des nœuds individuels est affecté par le statut des problèmes associés. L'association des problèmes est réalisée à l'aide du tagging.

Zabbix peut envoyer des notifications ou exécuter automatiquement un script sur le serveur Zabbix lorsqu'un changement d'état de service est détecté. Il est possible de définir des règles flexibles pour déterminer si un service parent doit passer dans un « état de problème » en fonction des statuts des services enfants. Les données de problème 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 se configure dans le menu Services, qui se compose des sections suivantes :

- Services

La section Services permet de construire une hiérarchie de votre infrastructure supervisée en ajoutant des services parents, puis des services enfants aux services parents.

En plus de la configuration de l'arborescence des services, cette section fournit une vue d'ensemble de toute l'infrastructure et permet d'identifier rapidement les problèmes ayant entraîné un changement d'état de 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.

- SLA report

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 :

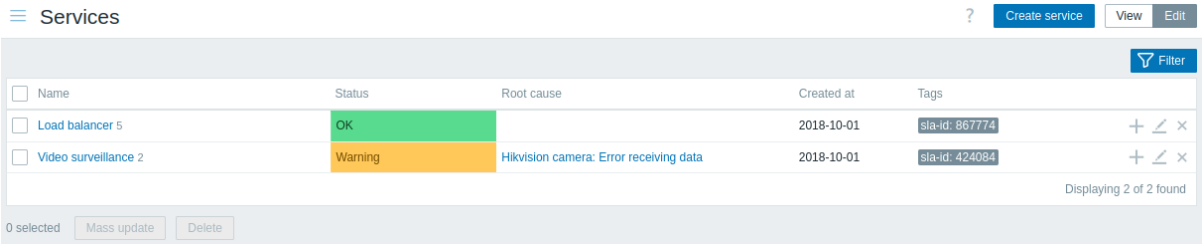
- envoyer une notification lorsqu'un service est indisponible
- exécuter une commande distante sur le serveur Zabbix lors d'un changement d'état de service
- envoyer une notification de rétablissement lorsqu'un service est de nouveau disponible.

Voir aussi :

- Exemple de configuration de la supervision SLA
- Remarques sur la mise à niveau des services depuis des versions de Zabbix antérieures à 6.0

1 Arborescence des services

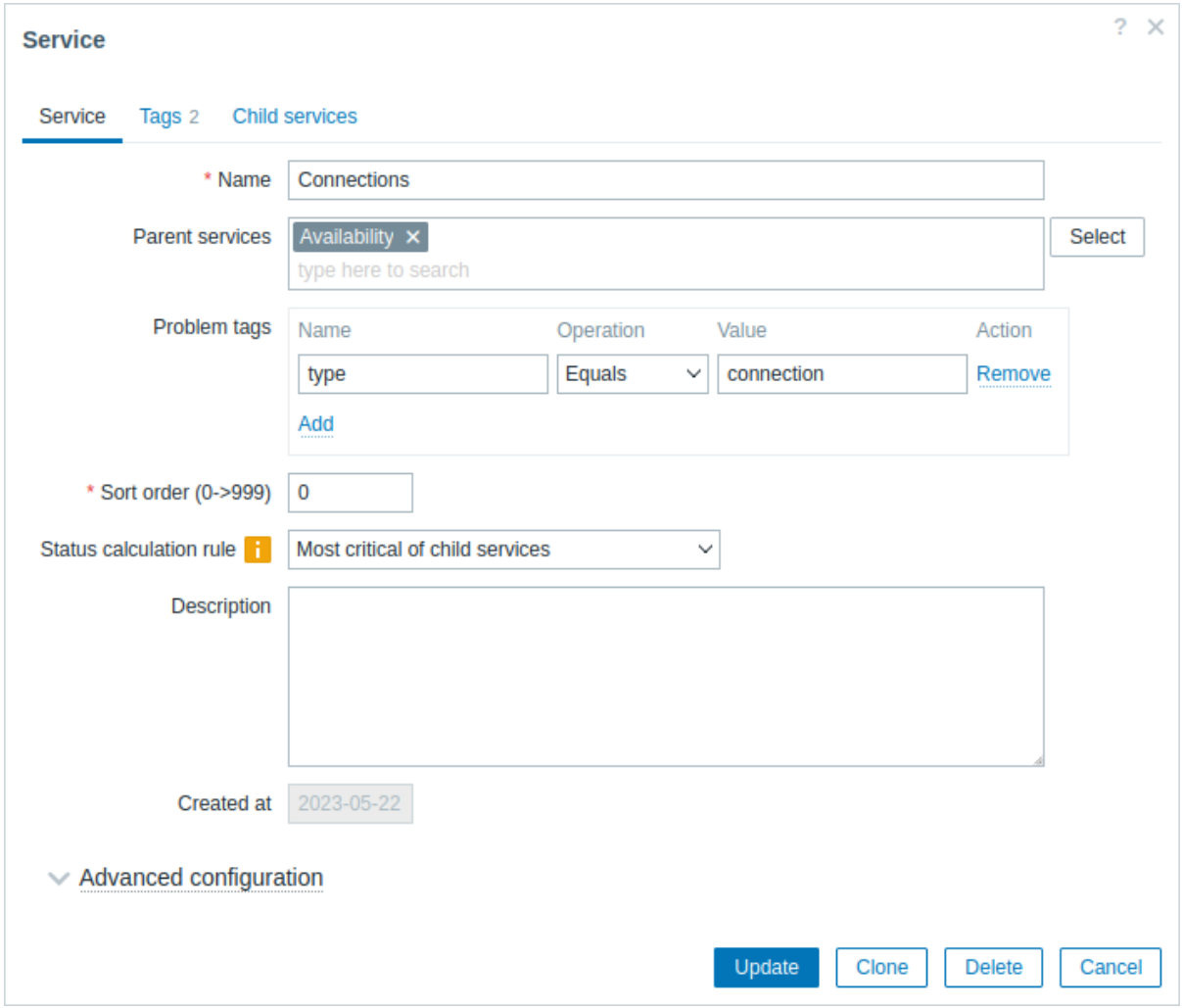
L'arborescence des services se configure dans la section de menu *Services -> Services*. Dans le coin supérieur droit, passez de view au mode d'édition.



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 une icône plus à côté du service parent. Cela ouvrira le même formulaire de configuration du service, mais le paramètre *Parent services* sera prérempli.

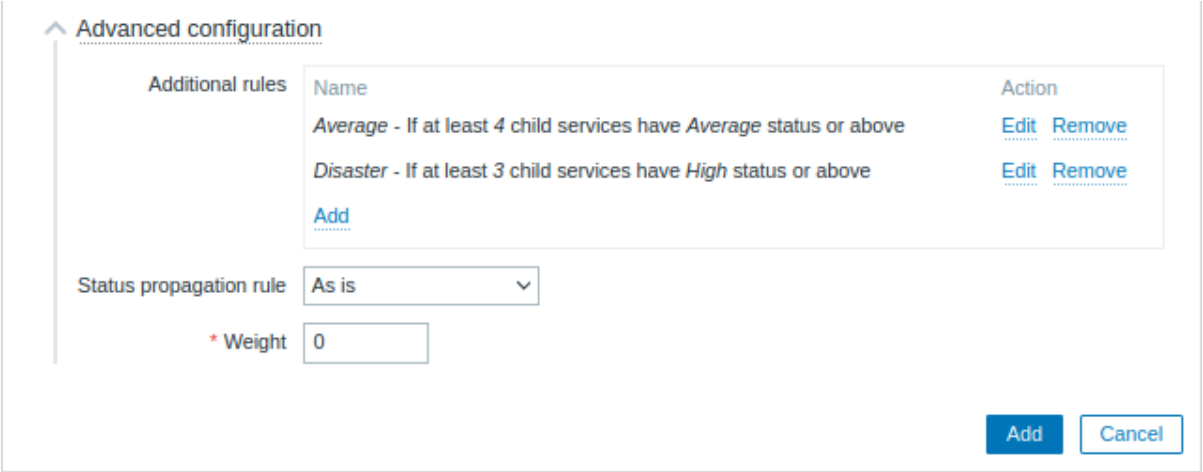
Configuration du service Dans l'onglet **Service**, spécifiez les paramètres requis du service :



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

Paramètre	Description
Name	Nom du service.
Parent services	Services parents auxquels le service appartient. Laissez ce champ vide si vous ajoutez le service de niveau le plus élevé. 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 mapper les données de problème au service. Plusieurs conditions peuvent être définies. La correspondance des noms de tags est toujours sensible à la casse. Deux opérateurs sont disponibles pour chaque condition : **Equals** - inclure les noms et valeurs de tags spécifiés (sensible à la casse) **Contains** - inclure les noms de tags spécifiés lorsque les valeurs des tags contiennent la chaîne saisie (correspondance de sous-chaîne, sensible à la casse)
Sort order	Ordre de tri pour l'affichage, le plus petit en premier.
Status calculation rule	Règle de calcul du statut du service : **Most critical if all children have problems** - le problème le plus critique parmi les services enfants est utilisé pour colorer le statut du service, si tous les services enfants ont des problèmes **Most critical of child services** - le problème le plus critique parmi les services enfants est utilisé pour colorer le statut du service **Set status to OK** - ne pas calculer le statut du service Des règles supplémentaires de calcul du statut peuvent être configurées dans les options de configuration avancée.
Description	Description du service.
Created at	Heure de création du service ; affichée lors de la modification d'un service existant.
Advanced configuration	Cliquez sur le libellé *Advanced configuration* pour afficher les options de configuration avancée.

Configuration avancée



Paramètre	Description
Règles supplémentaires	Cliquez sur *Ajouter* pour configurer des règles supplémentaires de calcul du statut.
Définir le statut sur	Définissez le statut du service sur *OK* (par défaut), *Non classé*, *Information*, *Avertissement*, *Moyen*, *Élevé* ou *Désastre* 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 (Statut) ou supérieur** **si au moins (N%) des services enfants ont le statut (Statut) ou supérieur** **si moins de (N) services enfants ont le statut (Statut) ou inférieur** **si moins de (N%) des services enfants ont le statut (Statut) ou inférieur** **si le poids des services enfants ayant le statut (Statut) ou supérieur est d'au moins (W)** **si le poids des services enfants ayant le statut (Statut) ou supérieur est d'au moins (N%)** **si le poids des services enfants ayant le statut (Statut) ou inférieur est inférieur à (W)** **si le poids des services enfants ayant le statut (Statut) ou inférieur est inférieur à (N%)** Si plusieurs conditions sont spécifiées et que la situation correspond à plus d'une condition, la sévérité la plus élevée sera définie.
N (W)	Définissez la valeur de N ou W (1-100000), ou N% (1-100) dans la condition.
Statut	Sélectionnez la valeur de *Statut* dans la condition : *OK* (par défaut), *Non classé*, *Information*, *Avertissement*, *Moyen*, *Élevé* ou *Désastre*.
Règle de propagation du statut	Règle de propagation du statut du service vers le service parent : **Tel quel** - le statut est propagé sans modification **Augmenter de** - vous pouvez augmenter le statut propagé de 1 à 5 niveaux de sévérité **Diminuer de** - vous pouvez diminuer le statut propagé de 1 à 5 niveaux de sévérité **Ignorer ce service** - le statut n'est pas du tout propagé au service parent **Statut fixe** - le statut est propagé de manière statique, c'est-à-dire toujours identique
Poids	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 sévérité au-delà du niveau calculé selon le paramètre principal *Règle de calcul du statut*. Si, selon les règles supplémentaires, le statut doit être Avertissement, mais que selon la *Règle de calcul du statut* le statut est Désastre, le service aura le statut Désastre.

L'onglet **Tags** contient des tags de niveau de service. Les tags de niveau de service sont utilisé pour identifier un service. Les tags de ce type ne sont pas utilisées pour mapper les problèmes au service (pour cela, utilisez les *Tags de problème* du premier onglet).

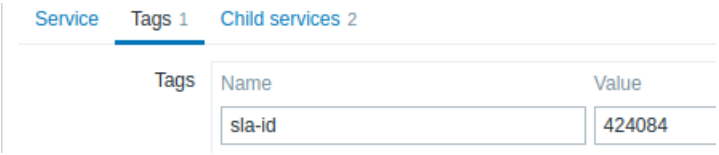
L'onglet **Services enfants** permet de spécifier les services dépendants. Cliquez sur *Ajouter* pour ajouter un service à partir de la liste des services existants. Si vous souhaitez ajouter un nouveau service enfant, enregistrez d'abord ce service, puis cliquez sur une icône plus à côté du service que vous venez de créer.

Balises Il existe deux types différents de balises dans les services :

- Balises de service
- Balises de problème

Balises de service

Les balises de service sont utilisées pour faire correspondre les services avec les actions de service et les SLA. Ces balises sont spécifiées dans l'onglet de configuration du service *Balises*.
Pour l'association des SLA, une 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 d'association sont configurables et peuvent utiliser une logique *AND*, *OR* ou *AND/OR*.

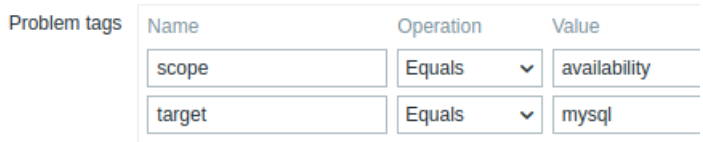


Balises de problème

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 le plus bas de la hiérarchie peuvent avoir des balises de problème définies et être directement corrélés aux problèmes. Si les balises de problème correspondent, le statut du service prendra le même statut que celui du problème. En cas de plusieurs problèmes, un service aura le statut du plus grave. Le statut d'un service parent est ensuite calculé en fonction des statuts des services enfants, conformément aux règles de calcul du statut.

Si plusieurs balises sont spécifiées, une logique *ET* est utilisée : un problème doit avoir toutes les balises spécifiées dans la configuration du service pour être associé au service.



Note:
Dans Zabbix, un problème hérite des balises de toute la chaîne de modèles, d'hôtes, d'éléments, de scénarios web et de déclencheurs. N'importe laquelle de ces balises peut être utilisée pour faire correspondre des problèmes à des services.

Exemple :

Le problème *La caméra Web 3 est hors service* a les tags `type:video-surveillance`, `floor:1` et `name:webcam-3`, ainsi que le statut *Avertissement*

Le service **Caméra Web 3** a le seul tag de problème spécifié : `name:webcam-3`



Le statut du service passera de *OK* à *Avertissement* lorsque ce problème sera détecté.

Si le service **Caméra Web 3** avait les tags de problème `name:webcam-3` et `floor:2`, son statut ne changerait pas lorsque le problème serait détecté, car les conditions ne seraient 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 parents sont conservés, tandis que les liens enfants 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 sont pas supprimés et remontent d'un niveau dans l'arborescence des services (les enfants de 1er niveau obtiennent le même niveau que le service parent supprimé).

Deux boutons sous la liste des services offrent des options de modification en masse :

- *Mass update* - mise à jour en masse des propriétés des services
- *Delete* - supprimer les services

Pour utiliser ces options, cochez les cases devant les services concernés, puis cliquez sur le bouton requis.

2 SLA

Aperçu Une fois les services créés, vous pouvez commencer à surveiller si leurs performances sont conformes au contrat de niveau de service (SLA).

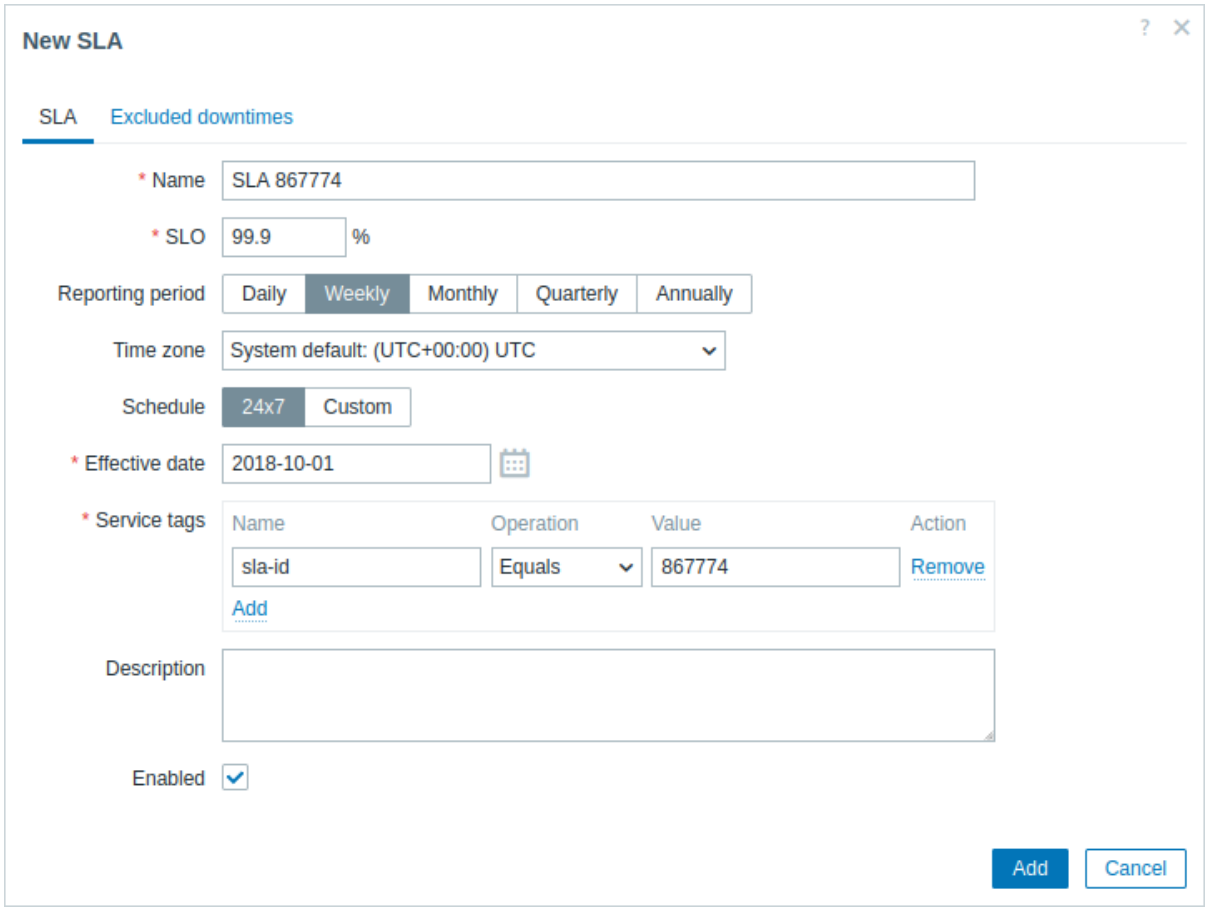
La section de menu *Services->SLA* permet de configurer les SLA pour divers services. Un SLA dans Zabbix définit l'objectif de niveau de service (SLO), le calendrier de disponibilité prévu et les temps d'arrêt planifiés.

Les SLA et les services correspondent aux tags de service. Le même SLA peut être appliqué à plusieurs services - les performances seront mesurées séparément pour chaque service correspondant. Un seul service peut avoir plusieurs SLA attribués - les données pour chacun des SLA seront affichées séparément.

Dans les rapports SLA, Zabbix fournit des données d'indicateur de niveau de service (SLI), qui mesurent la disponibilité réelle du service. La conformité d'un service aux objectifs du SLA est déterminée en comparant le SLO (disponibilité attendue en %) au SLI (disponibilité réelle en %).

Configuration Pour créer un nouveau SLA, cliquez sur le bouton *Créer un SLA*.

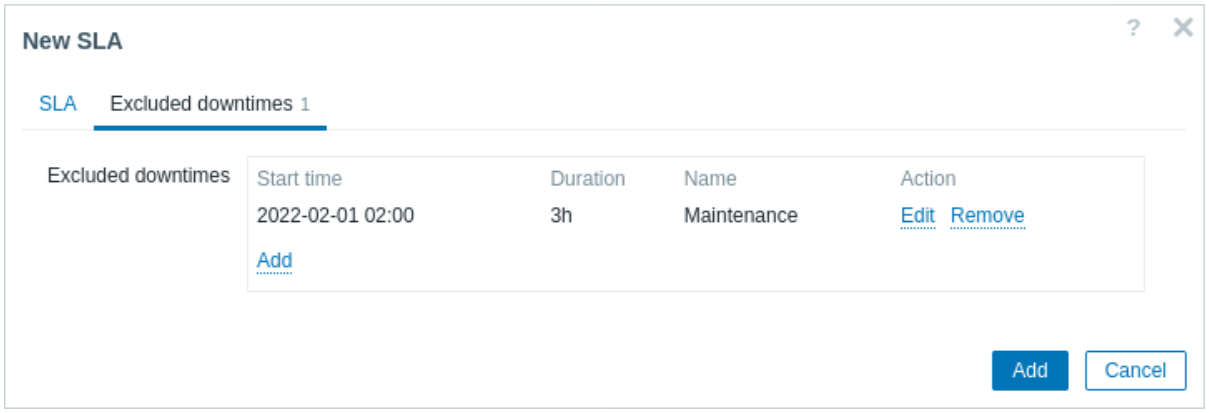
L'onglet **SLA** permet de spécifier les paramètres généraux du SLA.



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 00:00:00 au samedi 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 (janv.–mars), T2 (avr.–juin), T3 (juil.–sept.), T4 (oct.–déc.). **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, doit correspondre exactement, sensible à la casse. **Opération** - sélectionnez *Égale* si la valeur de l'étiquette doit correspondre exactement (sensible à la casse) ou *Contient* si une partie de la valeur de l'étiquette doit correspondre (insensible à 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 pour le 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 de SLA Les performances d'un service par rapport à un SLA sont visibles dans le rapport de SLA. Les rapports de SLA peuvent être consultés :

- depuis la section *SLA* en cliquant sur l'hyperlien du rapport de SLA ;
- depuis la section *Services* en cliquant sur le nom du SLA dans l'onglet d'information ;
- dans le widget de tableau de bord *Rapport de SLA*.

Une fois un SLA configuré, l'onglet *Info* de la section des services affichera également certaines informations sur les performances du service.

3 Exemple de configuration

Aperçu Cette section décrit une configuration simple pour surveiller le cluster haute disponibilité Zabbix en tant que service.

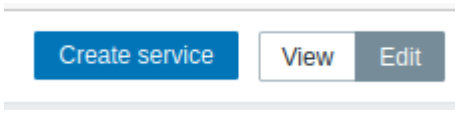
Prérequis Avant de configurer la surveillance des services, vous devez avoir configuré les hôtes :

- *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`

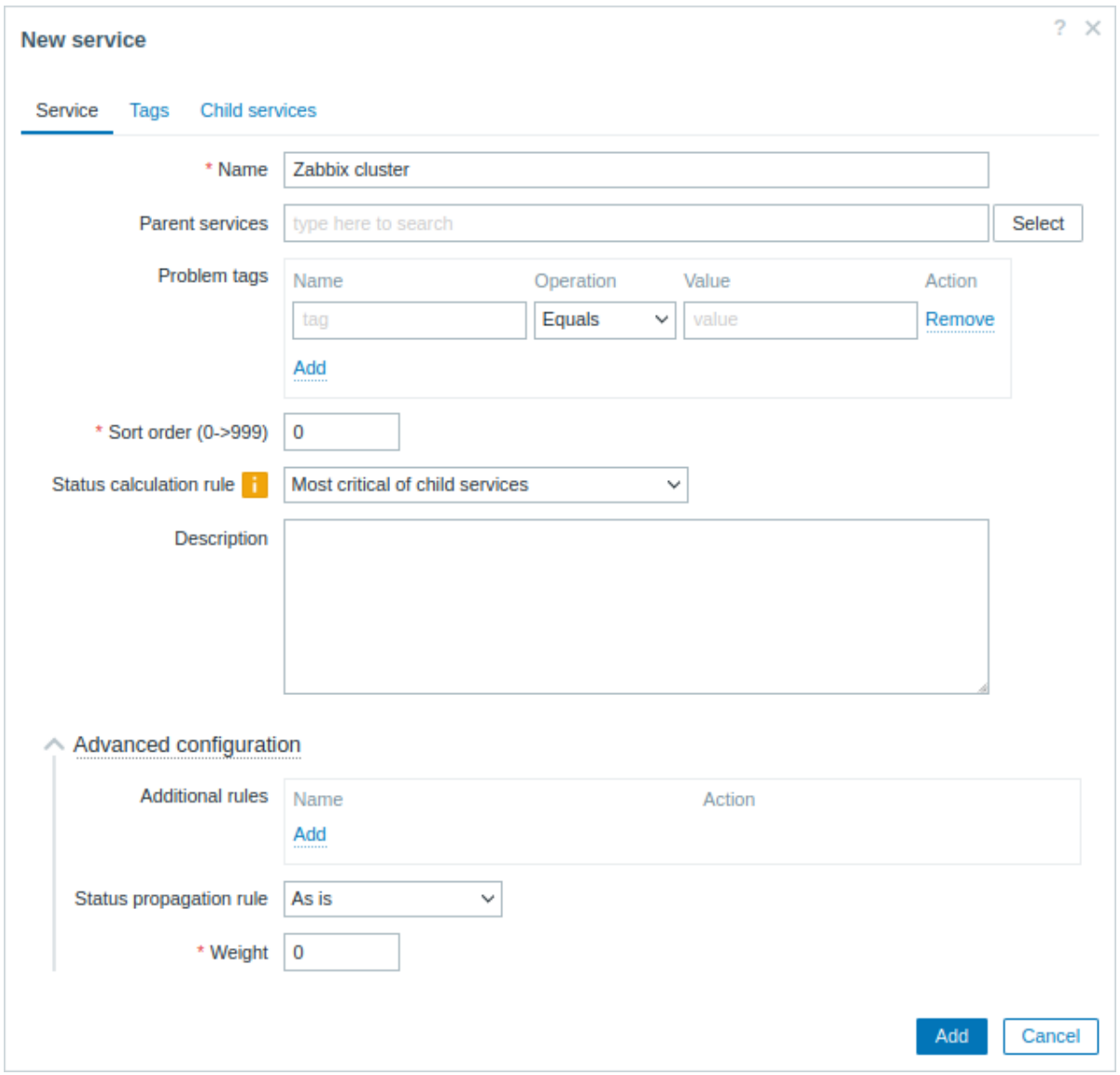
Arborescence des services L'étape suivante consiste à créer l'arborescence des services. Dans cet exemple, l'infrastructure est très basique et se compose de trois services : *Zabbix cluster* (parent) et deux services enfants *Zabbix server node 1* et *Zabbix server node 2*.

```
Zabbix cluster
|
|- Zabbix server node 1
|- Zabbix server node 2
```

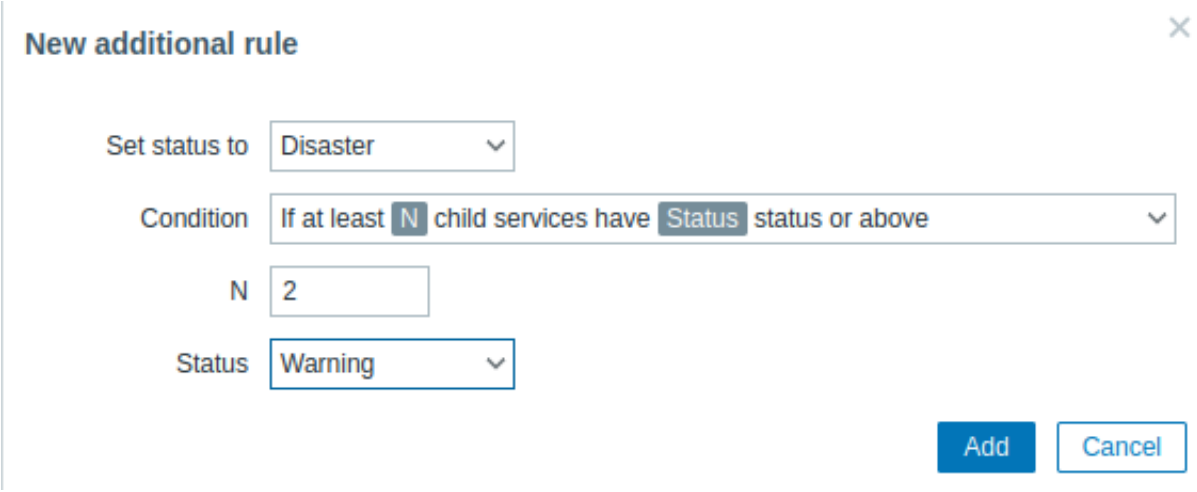
Sur la page Services, activez le mode *Edit* et cliquez sur Create service :



Dans la fenêtre de configuration du service, saisissez le nom *Zabbix cluster* et cliquez sur le libellé *Advanced configuration* pour afficher les options de configuration avancée.



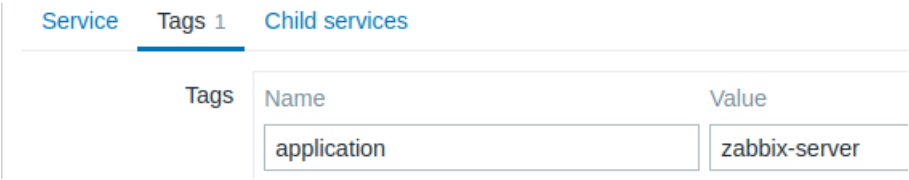
Configurez une règle supplémentaire :



Zabbix cluster aura deux services enfants, un pour chacun des nœuds HA. Si les deux nœuds HA rencontrent des problèmes avec un statut d'au moins *Warning*, le statut du service parent doit être défini sur *Disaster*. Pour y parvenir, la règle supplémentaire doit être configurée comme suit :

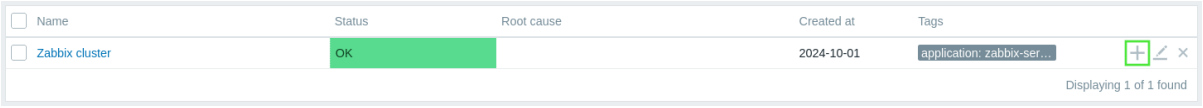
- Set status to: Disaster
- Condition: If at least `N` child services have `Status` status or above
- N: 2
- Status: Warning

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.

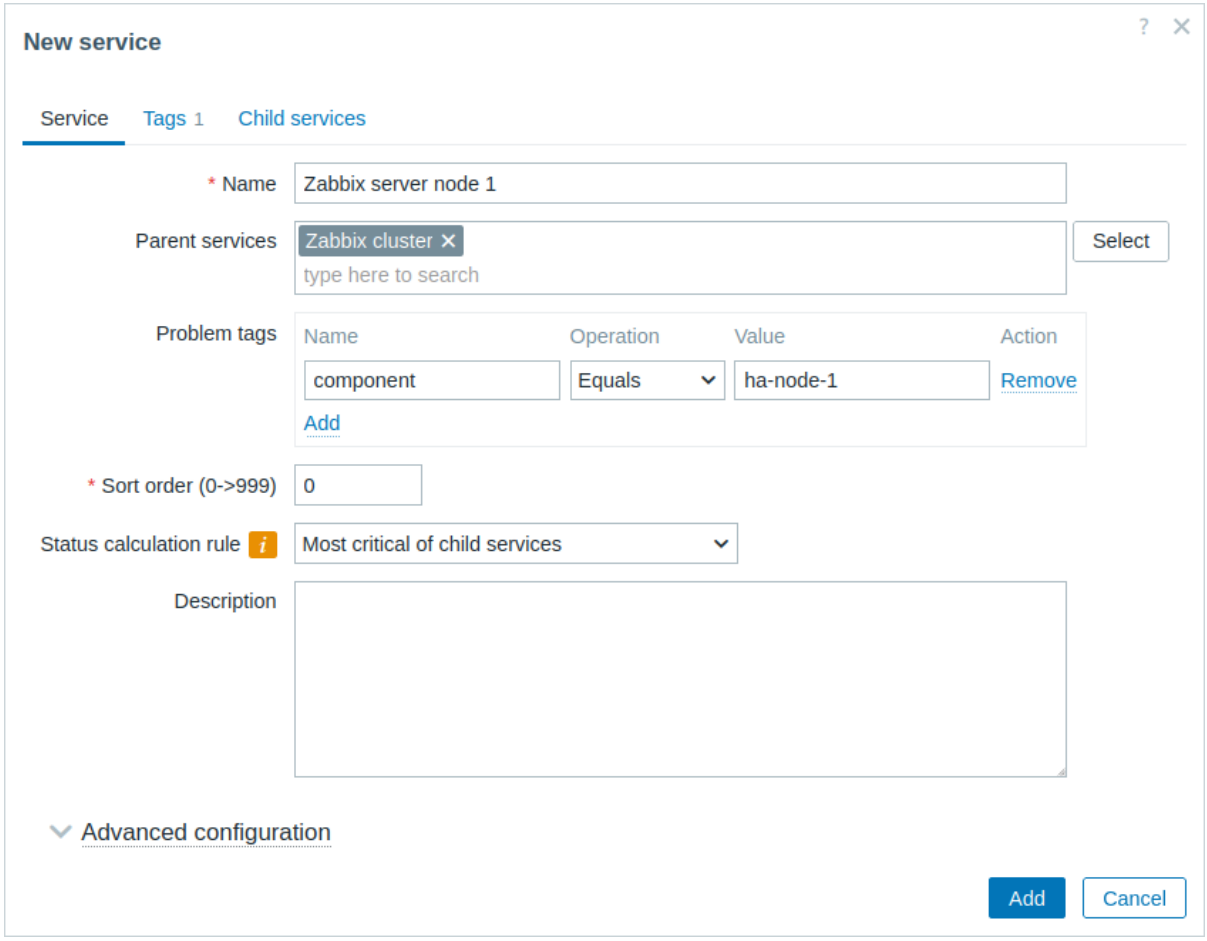
Pour ajouter un service enfant, cliquez sur l'icône plus à côté du service de cluster Zabbix (l'icône n'est visible qu'en mode *Édition*).



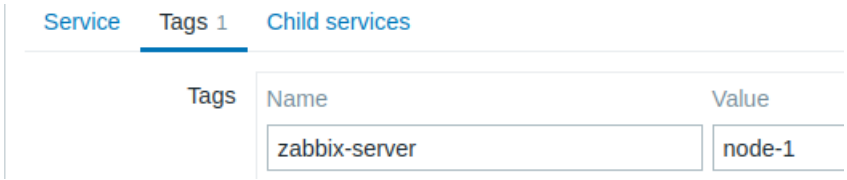
Dans la fenêtre de configuration du service, saisissez le nom *Nœud 1 du serveur Zabbix*. Notez que le paramètre Parent services est déjà prérempli avec *Cluster Zabbix*.

La disponibilité de ce service est affectée par les problèmes sur l'hôte *HA node 1*, marqués avec le tag de problème `component:ha-node-1`. Dans le paramètre Problem tags, saisissez :

- Nom : component
- Opération : Égal à
- Valeur : ha-node-1



Passez à l'onglet *Tags* et ajoutez un tag de service : `zabbix-server:node-1`. Ce tag sera utilisé 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.

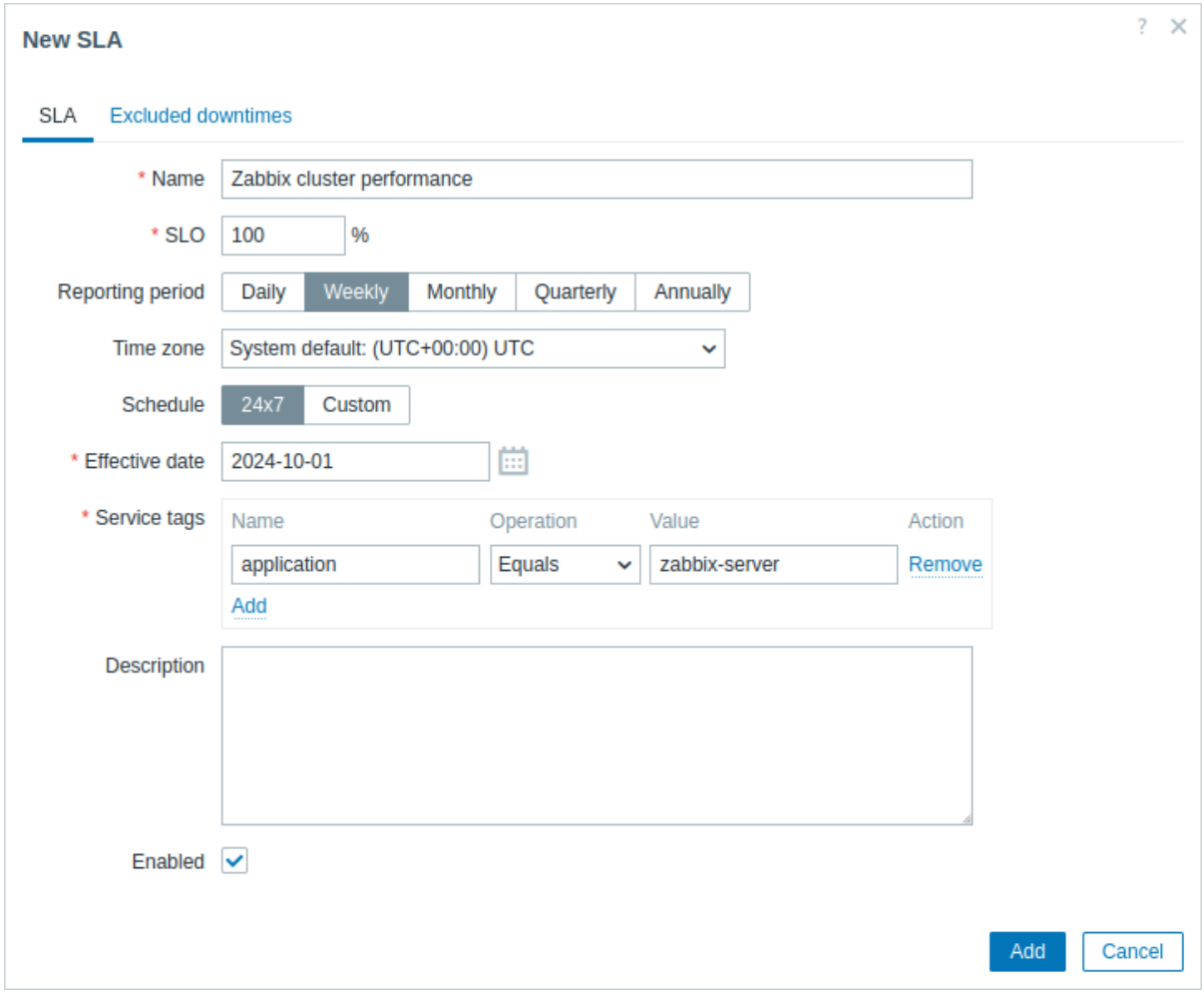
SLA Dans cet exemple, les performances attendues du cluster Zabbix sont de 100 %, à l'exclusion d'une période de maintenance semestrielle d'une heure.

Vous devez d'abord ajouter un nouveau contrat de niveau de service.

Accédez à la section de menu *Services->SLA* et cliquez sur Create SLA. Saisissez le nom *Zabbix cluster performance* et définissez le SLO sur 100 %.

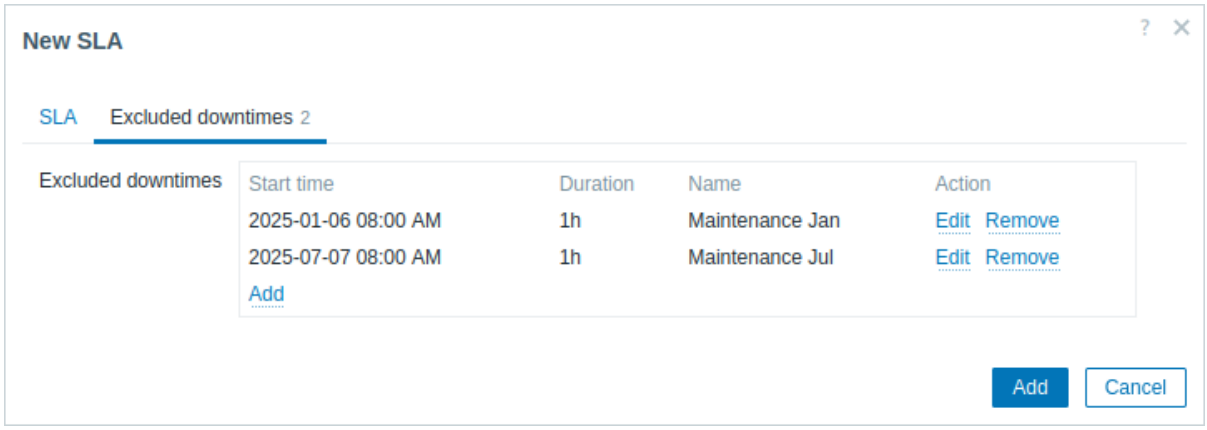
Le service Zabbix cluster possède le tag de service `application:zabbix-server`. Pour utiliser ce SLA afin de mesurer les performances du cluster Zabbix, dans le paramètre *Service tags*, indiquez :

- Nom : application
- Opération : Equals
- Valeur : zabbix-server



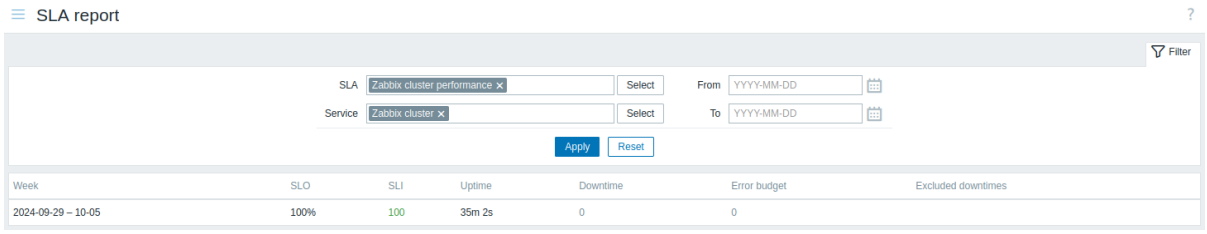
Dans une configuration réelle, vous pouvez également mettre à jour la période de rapport souhaitée, le fuseau horaire et la date de début, ou modifier la planification de 24/7 à personnalisée. Pour cet exemple, les paramètres par défaut sont suffisants.

Passez à l'onglet *Excluded downtimes* et ajoutez des indisponibilités pour les périodes de maintenance planifiées afin d'exclure ces périodes du calcul du SLA. Dans la section Excluded downtimes, cliquez sur le lien Add, saisissez le nom de l'indisponibilité, l'heure de début planifiée et la durée.

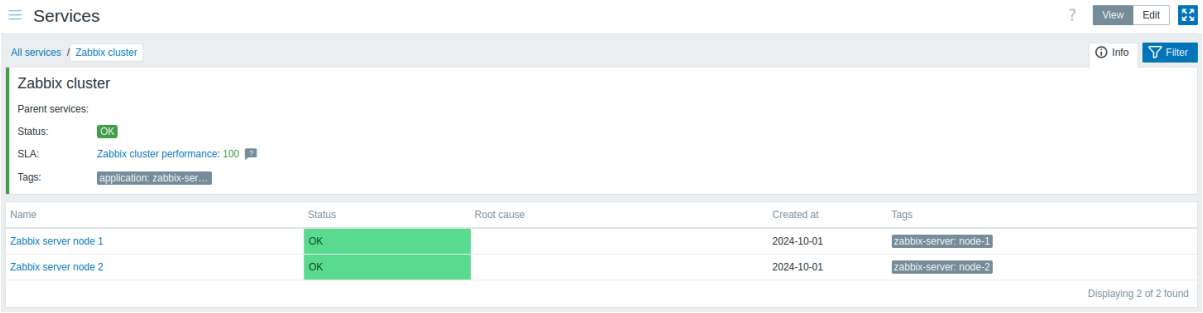


Cliquez sur Add pour enregistrer le nouveau SLA.

Passez à la section des rapports SLA pour afficher le rapport SLA du cluster Zabbix.



Les informations SLA peuvent également être consultées dans la section *Services*.



7 Surveillance Web

Aperçu Avec Zabbix, vous pouvez vérifier plusieurs aspects de la disponibilité des sites web.

> **Attention:**
> Pour effectuer la supervision web, le serveur Zabbix doit être initialement configuré avec la prise en charge de cURL (libcurl).

Pour activer la supervision web, vous devez définir des scénarios web. Un scénario web se compose d'une ou de plusieurs requêtes HTTP, ou « étapes ». Les étapes sont exécutées périodiquement par le serveur Zabbix dans un ordre prédéfini. Si un hôte est supervisé par un proxy, les étapes sont exécutées par le proxy.

Les scénarios web sont rattachés aux hôtes/modèles de la même manière que les items, les 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 pour chaque étape d'un scénario web :

- vitesse de téléchargement par seconde
- temps de réponse
- code de réponse

Pour plus de détails, voir les items de supervision 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.

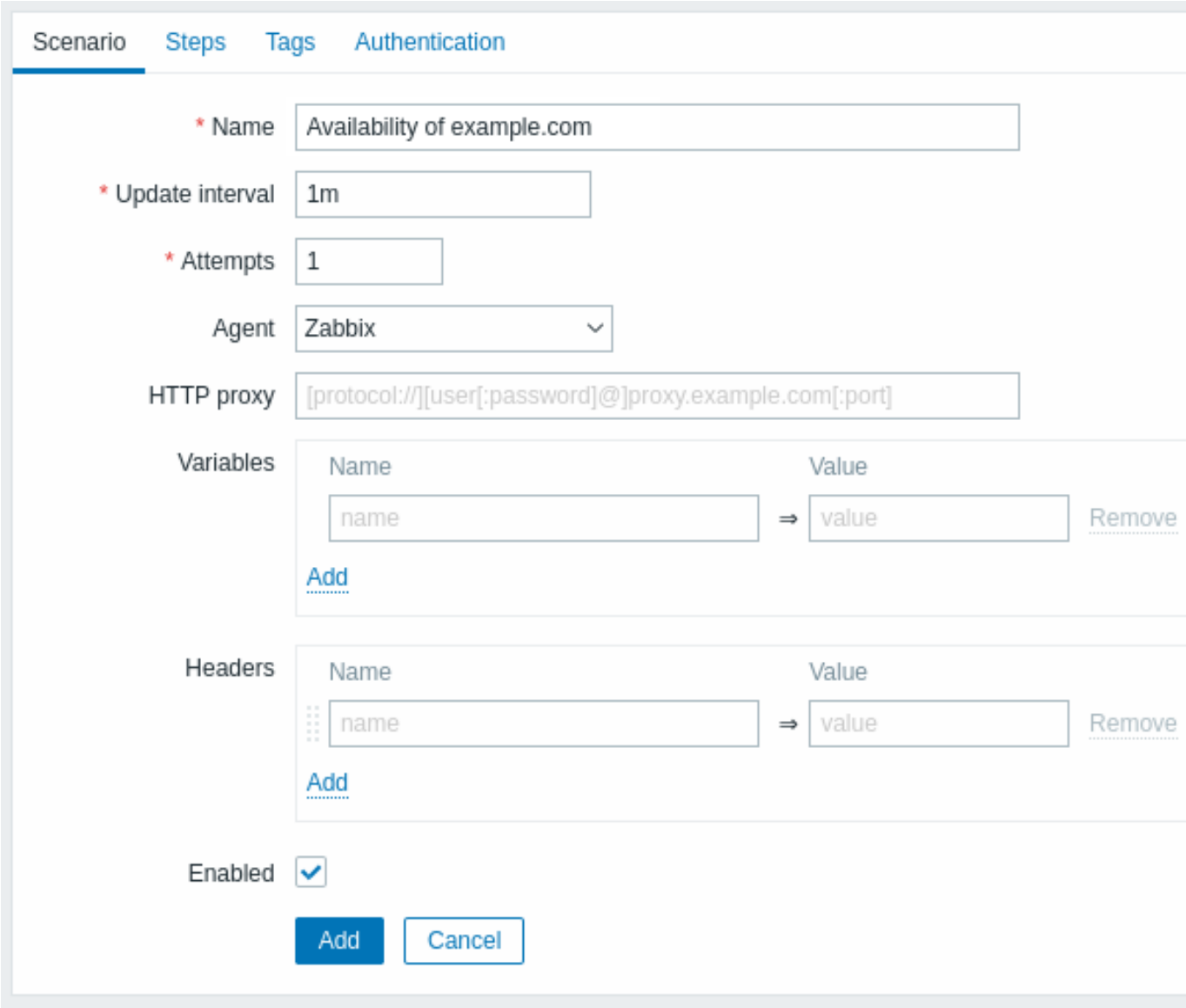
Zabbix peut également vérifier si une page HTML récupérée contient une chaîne prédéfinie. Il peut exécuter une connexion simulée et suivre un chemin de clics de souris simulés sur la page.

La supervision 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 *Follow redirects* ci-dessous). Le nombre maximal de redirections est codé en dur à 10 (en utilisant 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 :

- Allez dans : *Collecte de données* → *Hôtes* (ou *Modèles*)
- Cliquez sur *Web* dans la ligne de l'hôte/modèle
- Cliquez sur *Créer un scénario web* à droite (ou sur le nom du scénario pour modifier un scénario existant)
- Saisissez les paramètres du scénario dans le formulaire

L'onglet **Scénario** vous permet de configurer les paramètres généraux d'un scénario web.



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

Paramètres du scénario :

Parameter	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.
Intervalle de mise à jour	Fréquence d'exécution du scénario. Les suffixes temporels sont pris en charge, par ex. 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 ex. 5m → 30s), la prochaine vérification sera exécutée selon la valeur précédente (plus loin 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. 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 affectera de manière égale chaque étape du scénario. Jusqu'à 10 tentatives peuvent être spécifiées, la valeur par défaut est 1. *Remarque* : Zabbix ne répétera pas une étape en raison d'un code de réponse incorrect ou d'une non-correspondance d'une chaîne requise.
Agent	Sélectionnez un agent client. Zabbix se fera passer pour le navigateur sélectionné. Cela est utile lorsqu'un site web renvoie un contenu différent selon le navigateur. Des macros utilisateur peuvent être utilisées dans ce champ.

Parameter	Description
Proxy HTTP	Vous pouvez spécifier un proxy HTTP à utiliser, au format `[protocol://][username[:password]@]proxy.example.com[:port]`. Cela 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 spécifié, le proxy remplacera les variables d'environnement liées au proxy telles que http_proxy, 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 validité. 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 le 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 ont 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 les 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 dans la page web et, si elle est trouvée, stocke la correspondance 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 en URL lorsqu'elles sont utilisées dans des champs de requête ou des données de formulaire pour les variables post, mais doivent être encodées manuellement en URL lorsqu'elles sont utilisées dans un post brut ou directement dans l'URL.

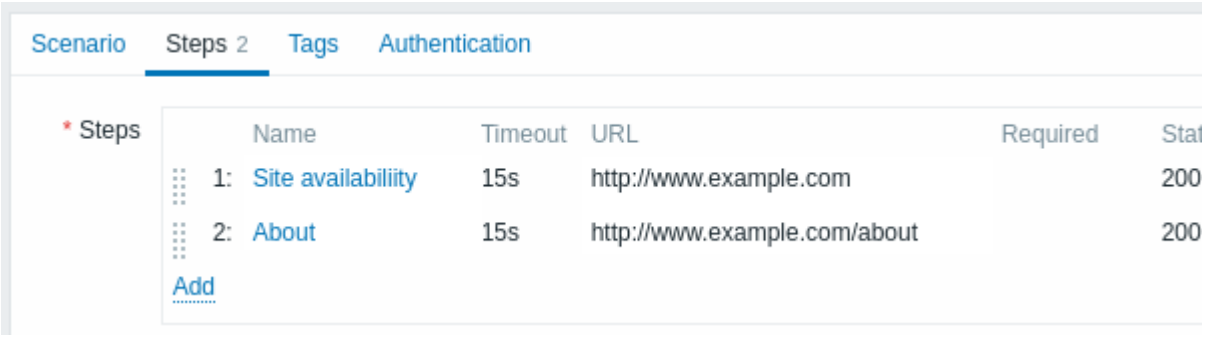
Parameter	Description
En-têtes	Les en-têtes HTTP sont utilisés lors de l'exécution d'une requête. Des en-têtes par défaut et personnalisés peuvent être utilisés. Les en-têtes seront attribués à l'aide des paramètres par défaut selon le type d'agent sélectionné dans une liste déroulante au niveau du scénario, et seront appliqués à toutes les étapes, sauf s'ils sont définis de manière personnalisée au niveau d'une étape. **Il convient de noter que la définition d'un en-tête au niveau d'une étape supprime automatiquement tous les en-têtes précédemment définis, à l'exception d'un en-tête par défaut attribué par la sélection de « 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 annuler l'en-tête au niveau du scénario, l'en-tête doit être nommé et attribué sans valeur au niveau d'une étape. Les en-têtes doivent être listés en utilisant la même syntaxe que celle utilisée dans le protocole HTTP, avec éventuellement 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 :

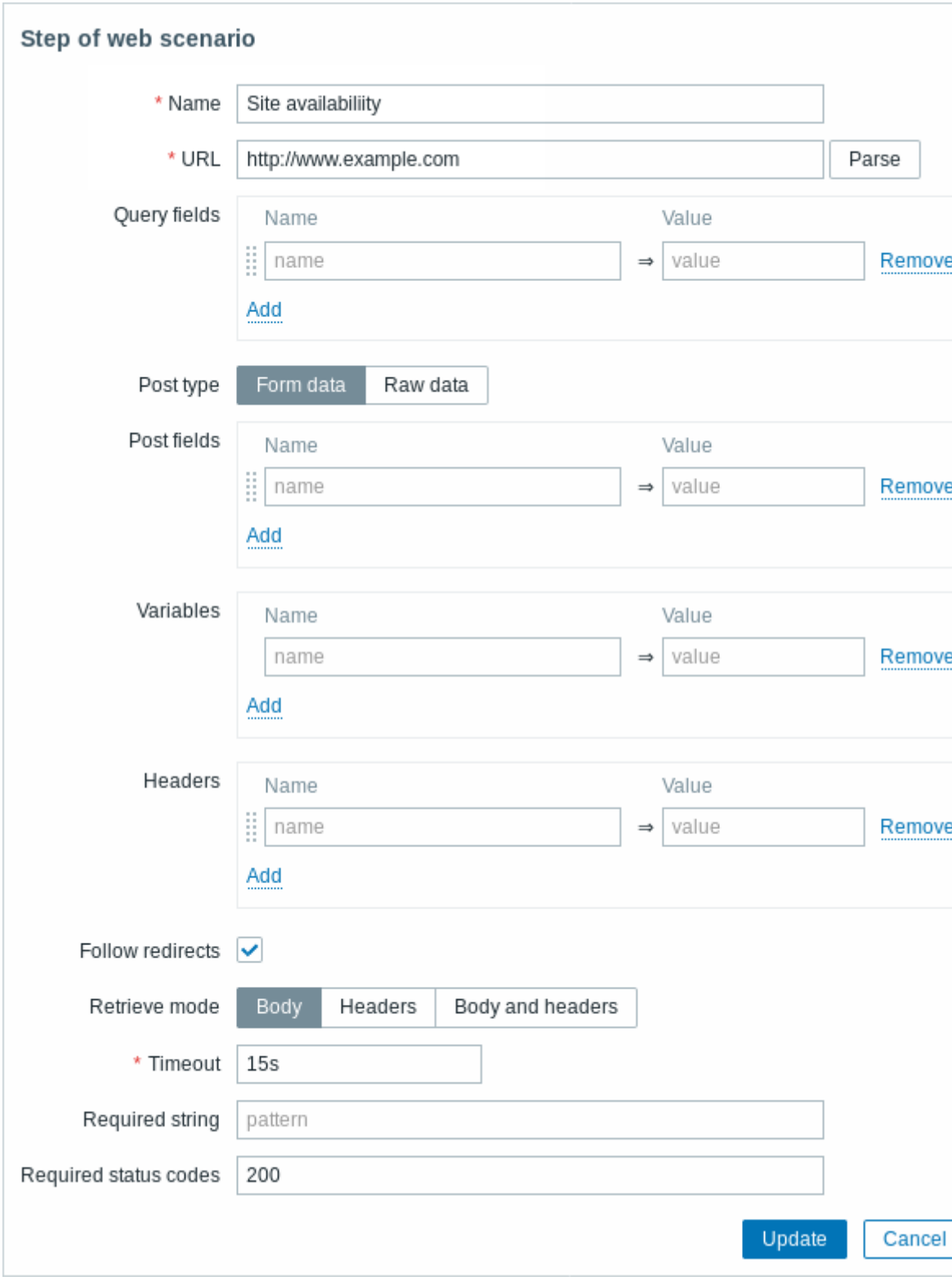
Clone	Créer un autre scénario basé sur les propriétés du scénario existant.
Clear history and trends	Supprimer les données d'historique et de tendances du scénario. Cela amènera le serveur à exécuter le scénario immédiatement après la suppression des données.

Note:
Si le champ *Proxy HTTP* est laissé vide, une autre façon d'utiliser un proxy HTTP consiste à définir des 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`. Plus de détails sont disponibles en exécutant une commande shell : *# man curl*.

L'onglet **Étapes** vous permet de configurer les étapes du scénario web. Pour ajouter une étape de 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 résolues en "******".



Configuration des étapes

Paramètres d'étape:

Paramètre	Description
Nom	Nom unique de l'étape.

Paramètre	Description
URL	URL où se connecter et où récupérer les données. Par exemple : https://www.example.com http://www.example.com/download Les noms de domaines peuvent être spécifiés en caractères Unicode. Ils sont automatiquement convertis en ASCII lors de l'exécution de l'étape du scénario web. Le bouton *Analyser* peut être utilisé pour séparer les champs de requête facultatifs (comme ?name=Admin&password=mypassword) de l'URL, et ainsi placer les attributs et les valeurs dans le *Champs de requête* pour un encodage automatique d'URL. Variables can be used in the URL, using the {macro} syntax. Les variables peuvent être utilisées dans l'URL, en utilisant la syntaxe {macro}. Les variables peuvent être encodée manuellement en utilisant la syntaxe {{macro}.urlencode()}. Les macros utilisateurs et les macros {HOST.*} sont supportées. Limité à 2048 caractères.
Champs de requête	Variables HTTP GET pour l'URL. Spécifié comme paires attribut/valeur. Les valeurs sont automatiquement encodées en URL. Les valeurs des variables de scénario, des macros utilisateur ou des macros {HOST.*} sont résolues puis automatiquement codées en URL. L'utilisation d'une syntaxe {{macro}.urlencode()} doublera leur encodage URL. Les macros utilisateurs et les macros {HOST.*} sont supportées.
Post	Variables HTTP POST. Dans le mode **Données de formulaire**, spécifié comme paires attribut/valeur. Les valeurs sont encodées automatiquement. Les valeurs issues des variables de scénario, les macros utilisateurs ou les macros {HOST.*} sont résolues et sont encodées automatiquement. Dans le mode **Données brutes**, les paires attribut/valeur sont affichées sur une seule ligne et concaténées avec le symbole **&**. Les valeurs brutes peuvent être encodées/décodées manuellement en utilisant une 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 au format URL, il faut substituer {user} par {{user}.urlencode()}. Les macros utilisateurs et les macros {HOST.*} sont supportées.
Variables	Variables qui peuvent être utilisées pour les fonctions GET et POST dans l'étape. Spécifié comme paires d'attribut/valeur. Les variables d'étape remplacent les variables de scénario ou les variables de l'étape précédente. Néanmoins, la valeur d'une variable d'étape affecte uniquement l'étape suivante (et non pas l'étape actuelle). Elles ont le format suivant : **{macro}**=value **{macro}**=regex:<regular expression> Pour plus d'informations, voir la description des variables au niveau du scenario. Les variables sont automatiquement encodées au format URL quand elles sont utilisées dans le champs de requête ou dans les données de formulaire pour les variables post, mais doivent être encodées manuellement quand elles sont utilisées dans les données brutes ou directement dans l'URL.
En-têtes	En-têtes HTTP personnalisés à envoyer lors de l'exécution d'une requête. Spécifié sous forme de paires d'attributs et de valeurs. Les en-têtes au niveau de l'étape remplaceront les en-têtes spécifiés pour le scénario. Par exemple, la définition d'un attribut 'User-Agent' sans valeur supprimera la valeur User-Agent définie au niveau du scénario. Les macros utilisateurs et les macros {HOST.*} sont supportées. Cela défini l'option cURL CURLOPT_HTTPHEADER. La spécification des en-têtes personnalisés est supportée depuis Zabbix 2.4*.
Suivre les redirections	Cochez la case pour suivre les redirections HTTP. Cela défini l'option cURL CURLOPT_FOLLOWLOCATION.
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
* Expiration*	Zabbix ne passera pas plus de temps que prévu à traiter l'URL (maximum 1 heure). En réalité, ce paramètre définit le délai maximal de connexion à l'URL et le délai maximal d'exécution d'une requête HTTP. Par conséquent, Zabbix ne passera pas plus de **2xExpiration* secondes sur l'étape. Les suffixes temporels sont supportés, ex : 30s, 1m, 1h. Les macros utilisateur sont supportées.

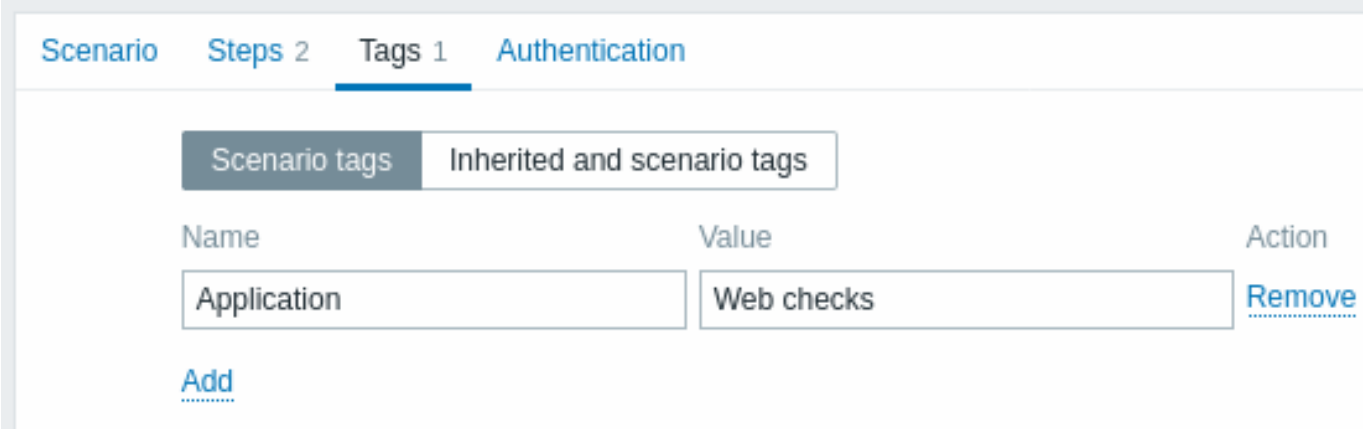
Paramètre	Description
Chaîne requise	Modèle requis d'expression régulière. À moins que le contenu récupéré (HTML) ne corresponde au modèle requis, l'étape échouera. Si vide, aucune vérification sur la chaîne n'est effectuée. Par exemple : Homepage of Zabbix Welcome.*admin *A noter* : Les expressions régulières créées dans l'interface Zabbix ne sont pas supportées dans ce champs. Les macros utilisateur et les macros {HOST.*} sont supportées.
Code d'état requis	Liste des codes d'état HTTP attendus. Si Zabbix obtient un code qui ne figure pas dans la liste, l'étape échouera. S'il est vide, aucune vérification de code d'état n'est effectuée. Par exemple : 200,201,210-299 Les macros utilisateur sont supportées.

Note:
Toute modification apportée aux étapes du scénario Web ne sera enregistrée que lorsque l'intégralité du scénario sera enregistré.

Voir aussi un exemple concret de la configuration des étapes de surveillance Web.

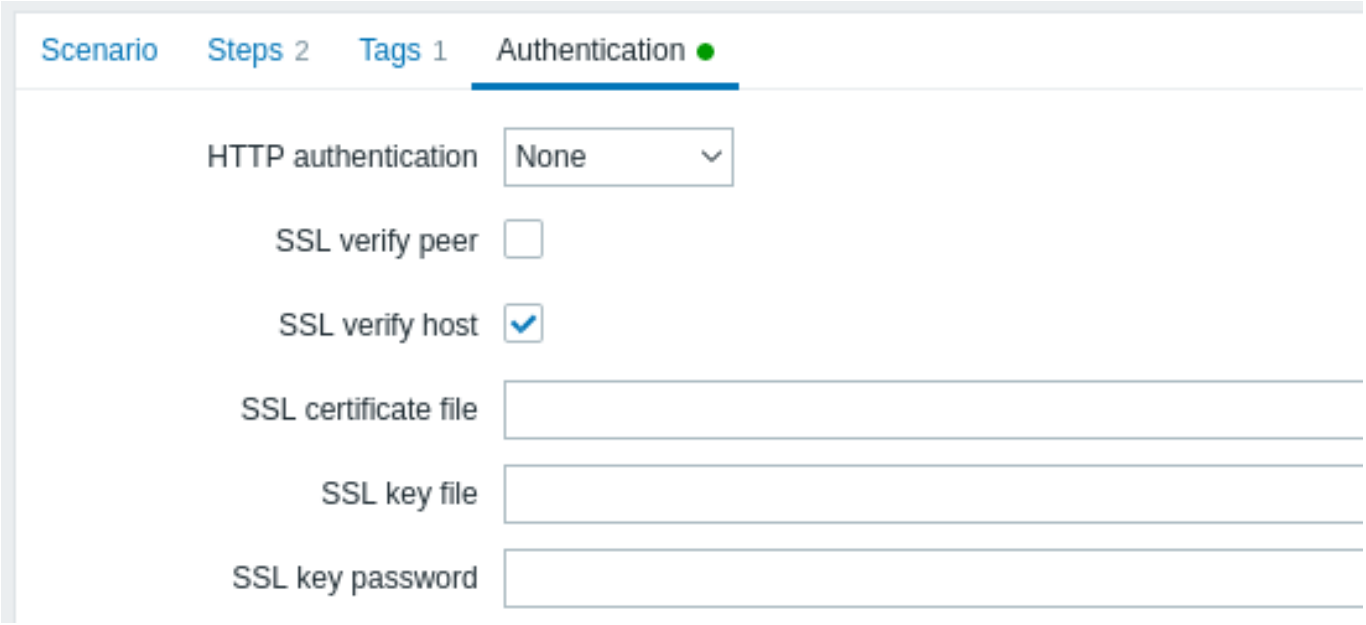
Configuration des tags

L'onglet **Tags** permet de définir des tags au niveau du scénario.



Les tags permettent de filtrer les scénarios Web et les éléments de surveillance Web.

Configuration de l'authentification L'onglet **Authentification** vous 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 :

Parameter	Description
Authentification HTTP	Sélectionnez l'option d'authentification : **Aucune** - aucune authentification n'est utilisée ; **Basic** - l'authentification de base est utilisée ; **NTLM** - l'authentification NTLM (Windows NT LAN Manager) est utilisée ; **Kerberos** - l'authentification Kerberos est utilisée (voir aussi : Configuration de Kerberos avec Zabbix) ; **Digest** - l'authentification Digest est utilisée.
Utilisateur	Saisissez le nom d'utilisateur (jusqu'à 255 caractères). Ce champ est disponible si *Authentification HTTP* est défini sur Basic, NTLM, Kerberos ou Digest. Les macros utilisateur sont prises en charge.
Mot de passe	Saisissez le mot de passe de l'utilisateur (jusqu'à 255 caractères). Ce champ est disponible si *Authentification HTTP* est défini 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 sera automatiquement pris à partir de l'emplacement global de l'autorité de certification (CA) du système. Vous pouvez remplacer l'emplacement des fichiers CA à l'aide du paramètre de configuration du serveur ou du proxy Zabbix SSLCALocation. Cela 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. Cela 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 ou du proxy Zabbix SSLCertLocation. Les macros `HOST.*` et les macros utilisateur peuvent être utilisées dans ce champ. Cela 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 ou du proxy Zabbix SSLKeyLocation. Les macros `HOST.*` et les macros utilisateur peuvent être utilisées dans ce champ. Cela 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. Cela définit l'option cURL CURLOPT_KEYPASSWD.

Attention:
[1] Zabbix prend en charge uniquement les fichiers de certificat et de clé privée au format PEM. Si vos données de certificat et de clé privée sont dans un fichier au format PKCS #12 (généralement avec l'extension *.p12 ou *.pfx), vous pouvez générer le fichier PEM à partir de celui-ci 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 des certificats sans redémarrage.

Note:
Si vous avez le certificat client et la clé privée dans un seul fichier, indiquez-le simplement 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, accédez à *Surveillance* → *Hôtes*, localisez l'hôte dans la liste et cliquez sur le lien hypertexte *Web* dans la dernière colonne. Cliquez sur le nom du scénario pour obtenir des informations détaillées.



Un aperçu des scénarios Web peut également être affiché dans *Surveillance* → *Tableau de bord* par un widget de surveillance Web.

Les résultats récents de l'exécution du scénario Web sont disponibles dans la section *Surveillance* → *Dernières données*.

Surveillance étendue Il est parfois nécessaire d'enregistrer le contenu HTML reçu d'une page.
Cela est particulièrement utile si une étape d'un scénario web échoue.
Le niveau de débogage 5 (trace) sert à cet usage.
Ce niveau peut être défini dans les fichiers de configuration server 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 démarrer la surveillance étendue, à condition que le niveau de débogage 4 soit déjà défini :

```
# Augmenter le niveau de journalisation de tous les processus http poller :
zabbix_server -R log_level_increase="http poller"

# Augmenter le niveau de journalisation du deuxième processus http poller :
zabbix_server -R log_level_increase="http poller,2"
```

Si la surveillance web étendue n'est pas nécessaire, elle peut être arrêtée à l'aide de l'option `-R log_level_decrease`.

1 Éléments de supervision 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
```

2 Scénario réel

Aperçu

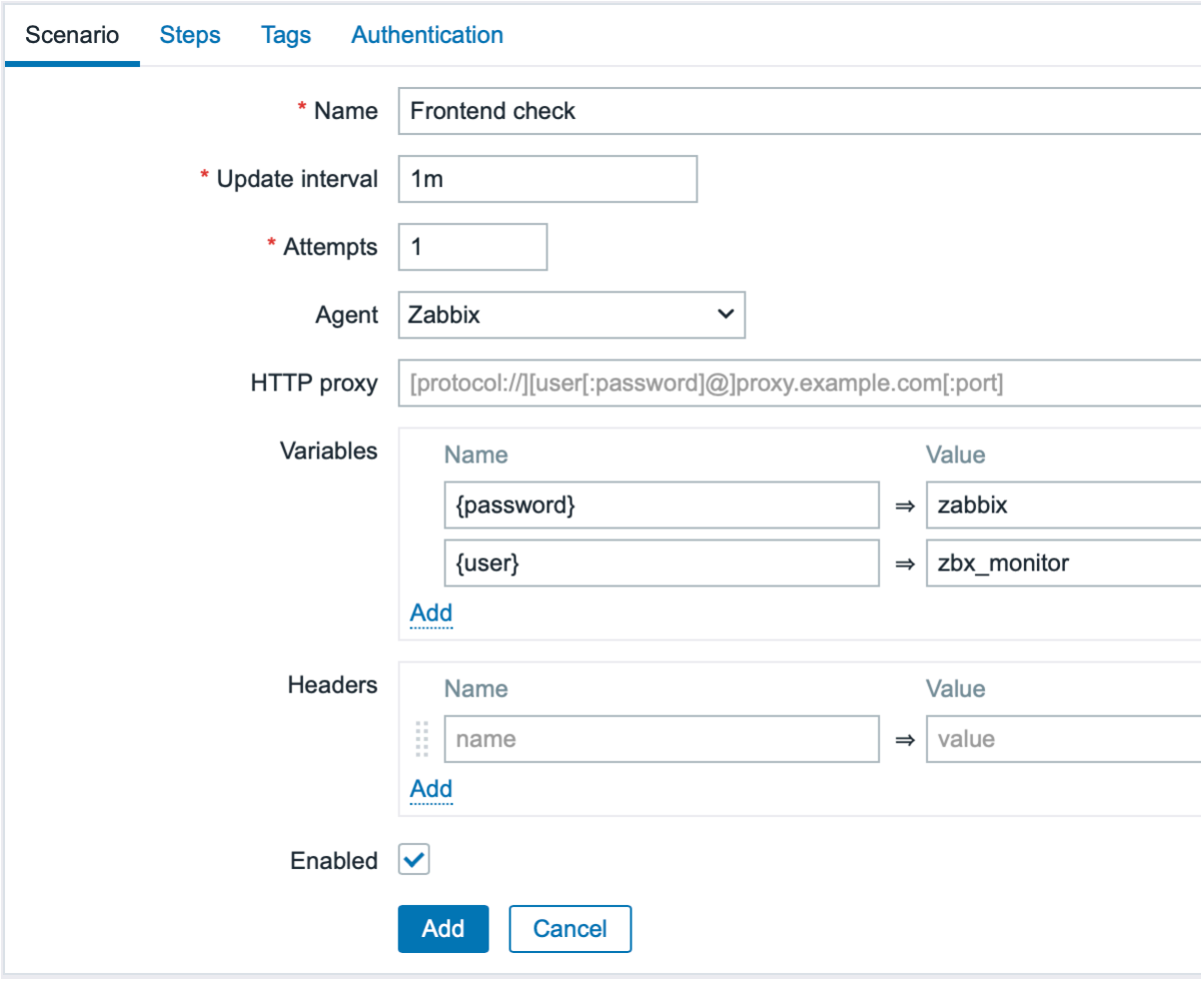
Cette section présente un exemple concret, étape par étape, de l'utilisation de la supervision web.

La supervision web de Zabbix sera utilisée pour surveiller le frontend de Zabbix. L'objectif est de déterminer s'il est disponible, s'il fournit le bon contenu et à quelle vitesse il 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

Accédez à *Collecte de données* → *Hôtes*, sélectionnez un hôte et cliquez sur *Web* sur la ligne de cet hôte.
Cliquez ensuite sur *Créer un scénario web*.



Dans le formulaire du nouveau scénario, renseignez les champs suivants :

- **Nom** - Vérification de l'interface web
- **Intervalle de mise à jour** - 1m
- **Tentatives** - 1
- **Agent** - Zabbix

Dans la section *Variables*, ajoutez deux variables : *{password}* et *{user}*.
Saisissez comme valeurs les identifiants de votre utilisateur Zabbix existant.

Pour des raisons de sécurité, il est recommandé de créer un utilisateur distinct avec des autorisations minimales à utiliser à des fins de supervision.

Si vous le souhaitez, passez à l'onglet *Tags* et ajoutez des tags au scénario web.

Note:
Une fois entièrement configuré, ce scénario web ajoutera automatiquement un élément Zabbix trapper à l'hôte.
Vous pouvez utiliser les tags du scénario web pour identifier rapidement les éléments et déclencheurs associés ou effectuer des recherches 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. Cliquez sur le bouton *Add* pour ajouter une étape individuelle.

Champs communs

Pour chaque étape décrite ci-dessous, renseignez les champs suivants en plus des champs spécifiques à l'étape :

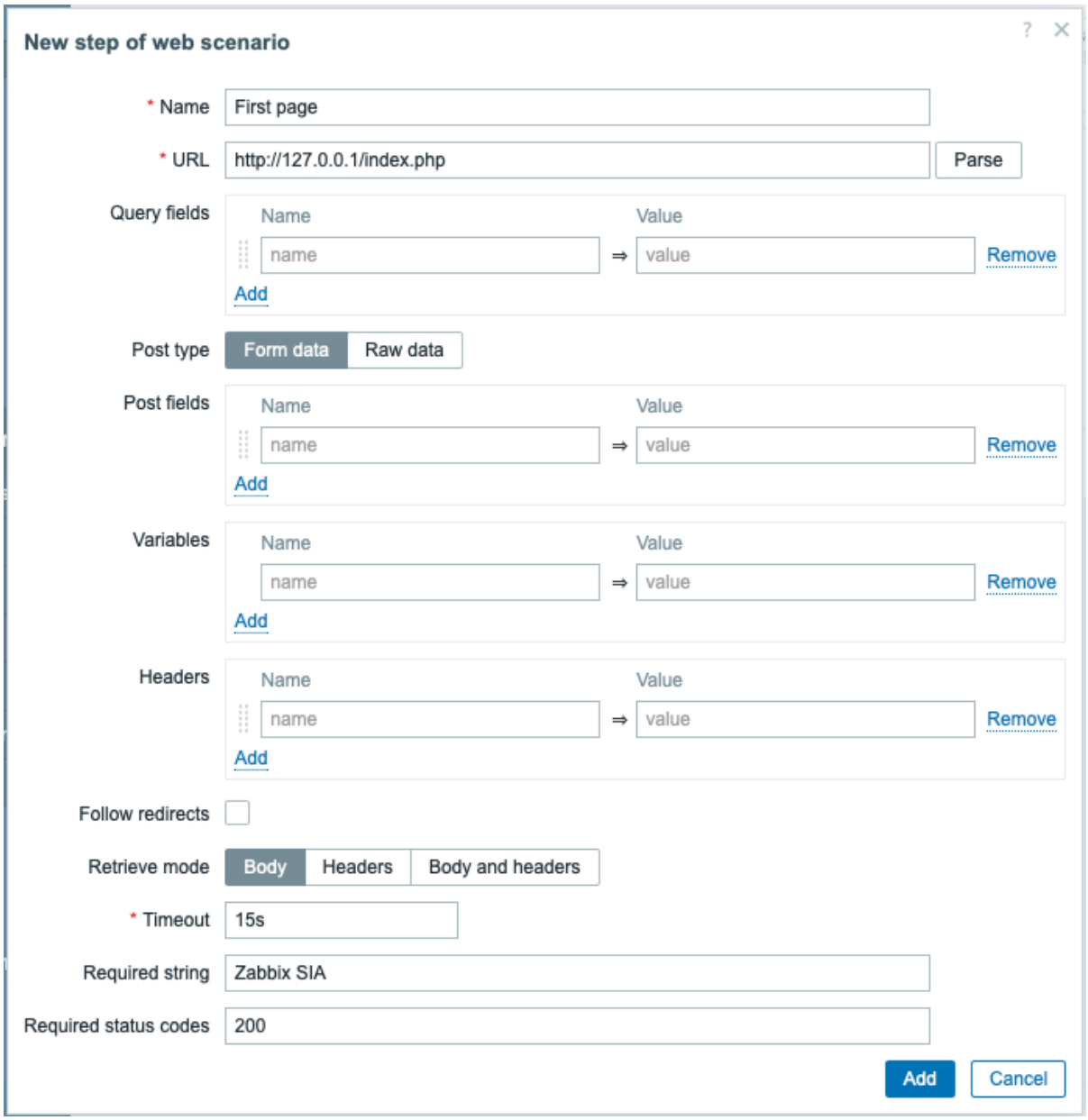
- **URL** - l'URL du frontend Zabbix
- **Timeout** - 15s
- **Required status codes** - 200

Étape 1 du scénario web

Vérifiez que la première page répond correctement, renvoie le code de réponse HTTP 200 et contient le texte « Zabbix SIA ».

- Dans le champ **Name**, saisissez *Première page*.
- Dans le champ **Required string**, saisissez *Zabbix SIA*.
- Renseignez les champs communs.

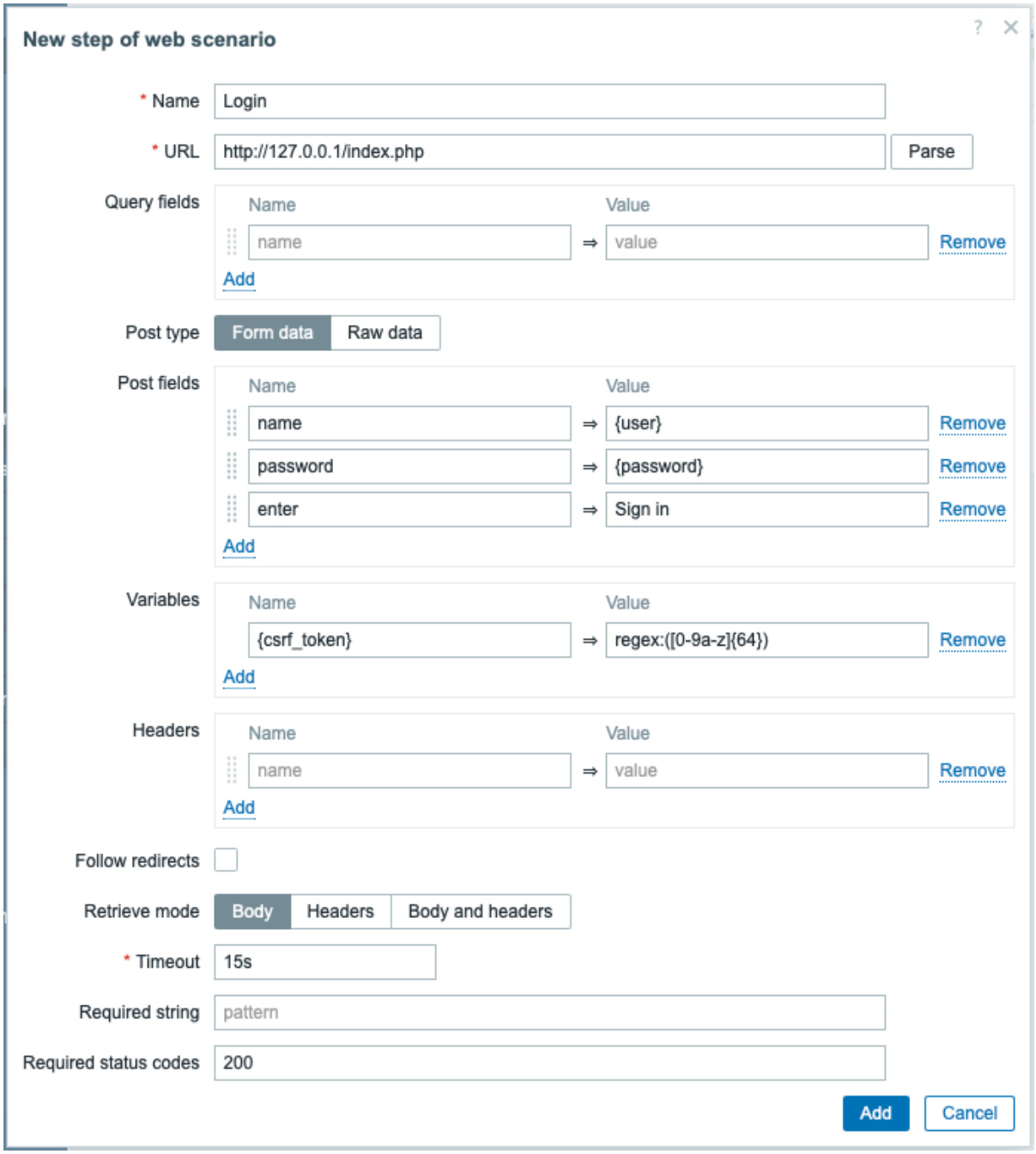
Une fois la configuration de l'étape terminée, appuyez sur le bouton *Add*.



Étape 2 du scénario web

Connectez-vous à l'interface web de Zabbix en utilisant les macros (variables) définies au niveau du scénario - *{user}* et *{password}*.

- Dans le champ **Name**, saisissez *Login*.
- Dans la section **Post fields**, ajoutez trois champs POST :
  - *name* avec la valeur *{user}*
  - *password* avec la valeur *{password}*
  - *enter* avec la valeur *Sign in*
- Dans la section **Variables**, ajoutez une nouvelle variable *{csrf_token}* avec la valeur *regex:([0-9a-z]{64})*. Cette variable récupérera la valeur du jeton CSRF attribué afin de la réutiliser à l'étape 4.
- Remplissez les champs communs.

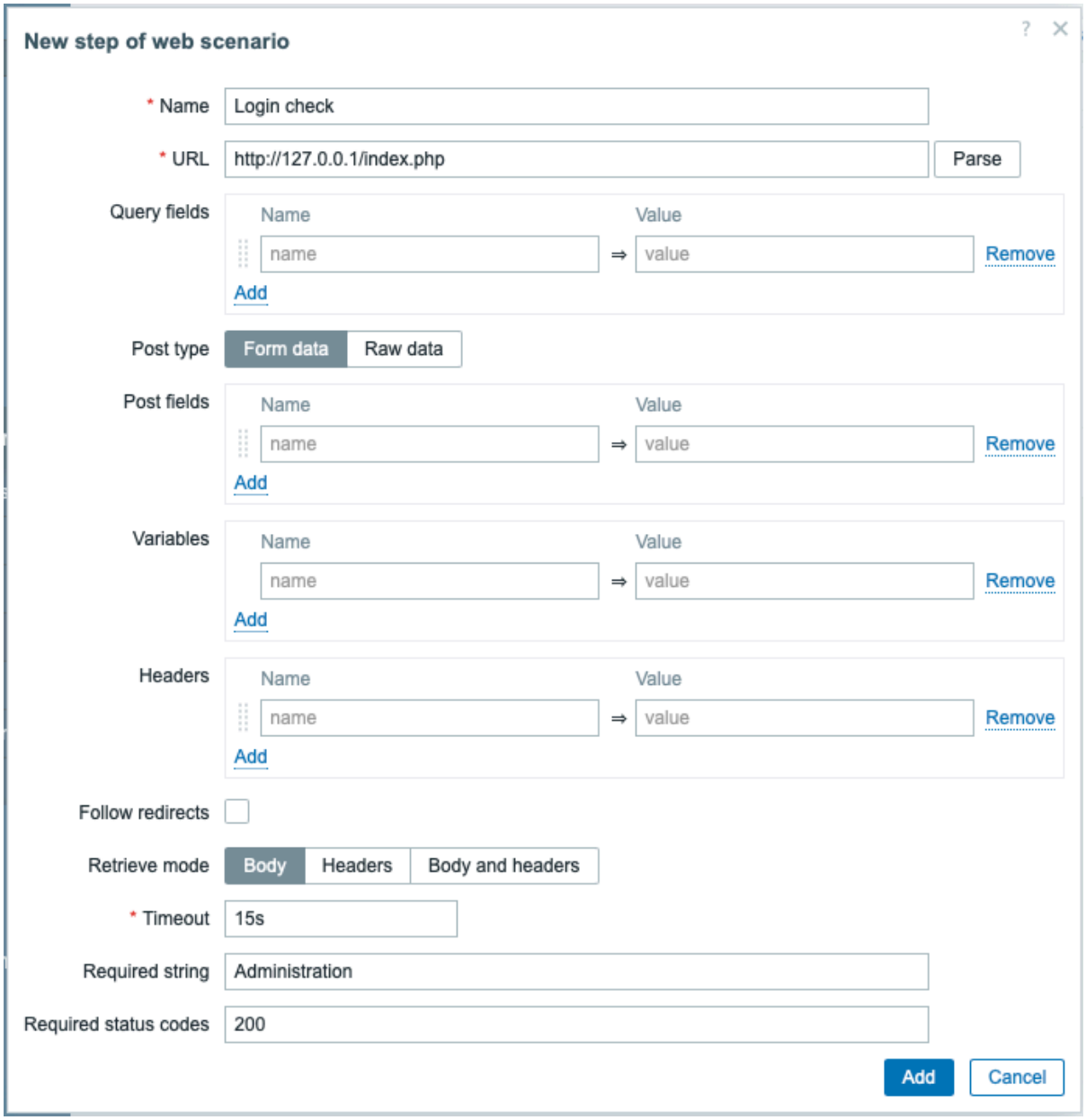


Attention:
Notez que l'interface web de Zabbix utilise une redirection JavaScript lors de la connexion ; la connexion doit donc avoir lieu en premier, et les fonctionnalités nécessitant une session ouverte ne peuvent être vérifiées que dans les étapes suivantes. De plus, l'étape de connexion doit utiliser l'URL complète vers le fichier **index.php**.

Étape 3 du scénario web

Après vous être connecté, vérifiez la réussite de l'opération en recherchant une chaîne visible uniquement lorsque vous êtes connecté, par exemple *Administration*.

- Dans le champ **Name**, saisissez *Login check*.
- Dans le champ **Required string**, saisissez *Administration*.
- Renseignez les champs communs.

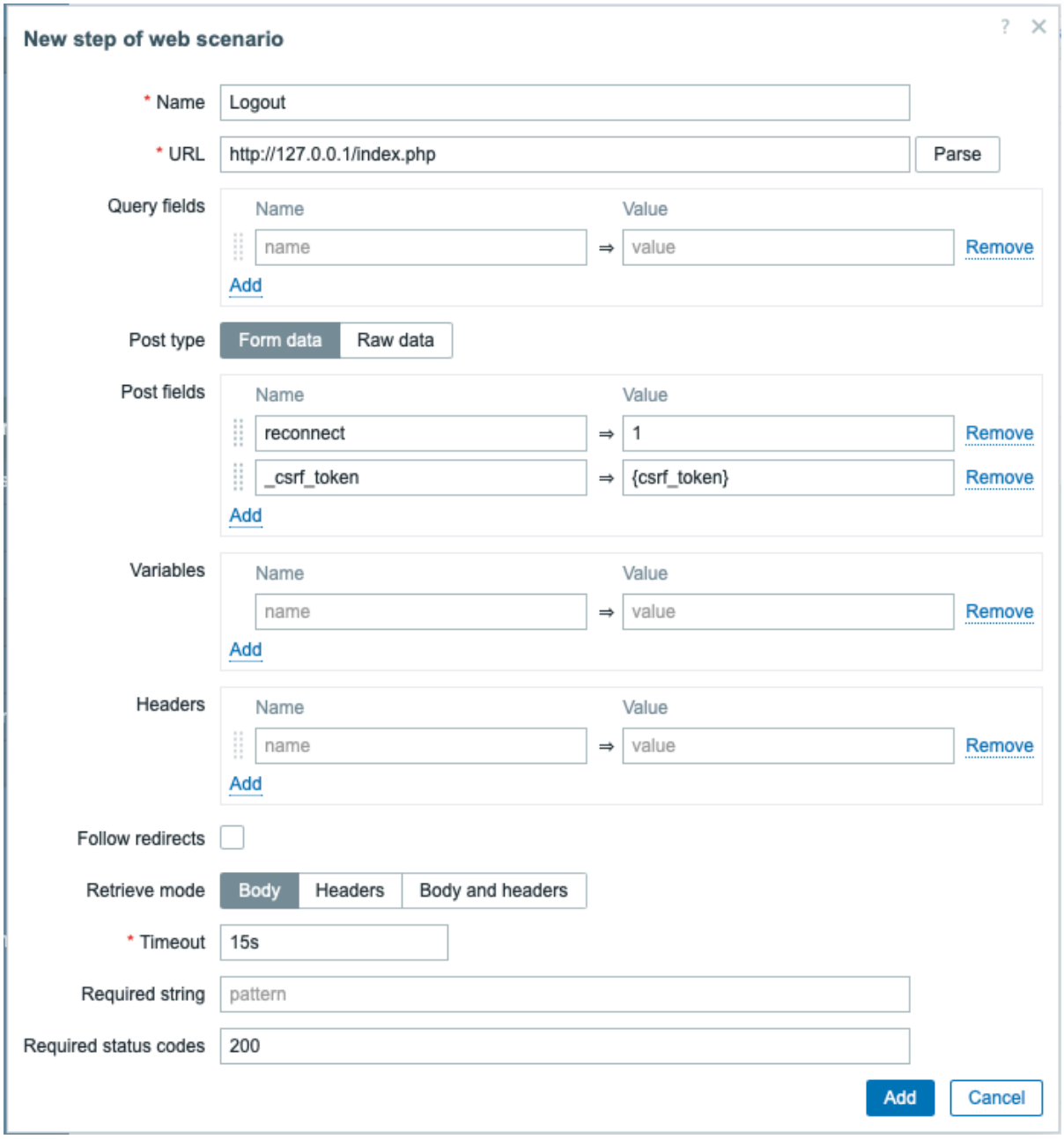


Étape 4 du scénario web

Une fois l'accessibilité du frontend et la connexion vérifiées, ajoutez une étape de déconnexion, sinon la base de données Zabbix sera encombrée par de nombreux enregistrements de sessions ouvertes.

- Dans le champ **Name**, saisissez *Logout*.
- Dans la section **Post fields**, ajoutez deux champs POST :
  - *reconnect* avec la valeur *1*
  - *_csrf_token* avec la valeur *{csrf_token}*.
- Renseignez les champs communs.

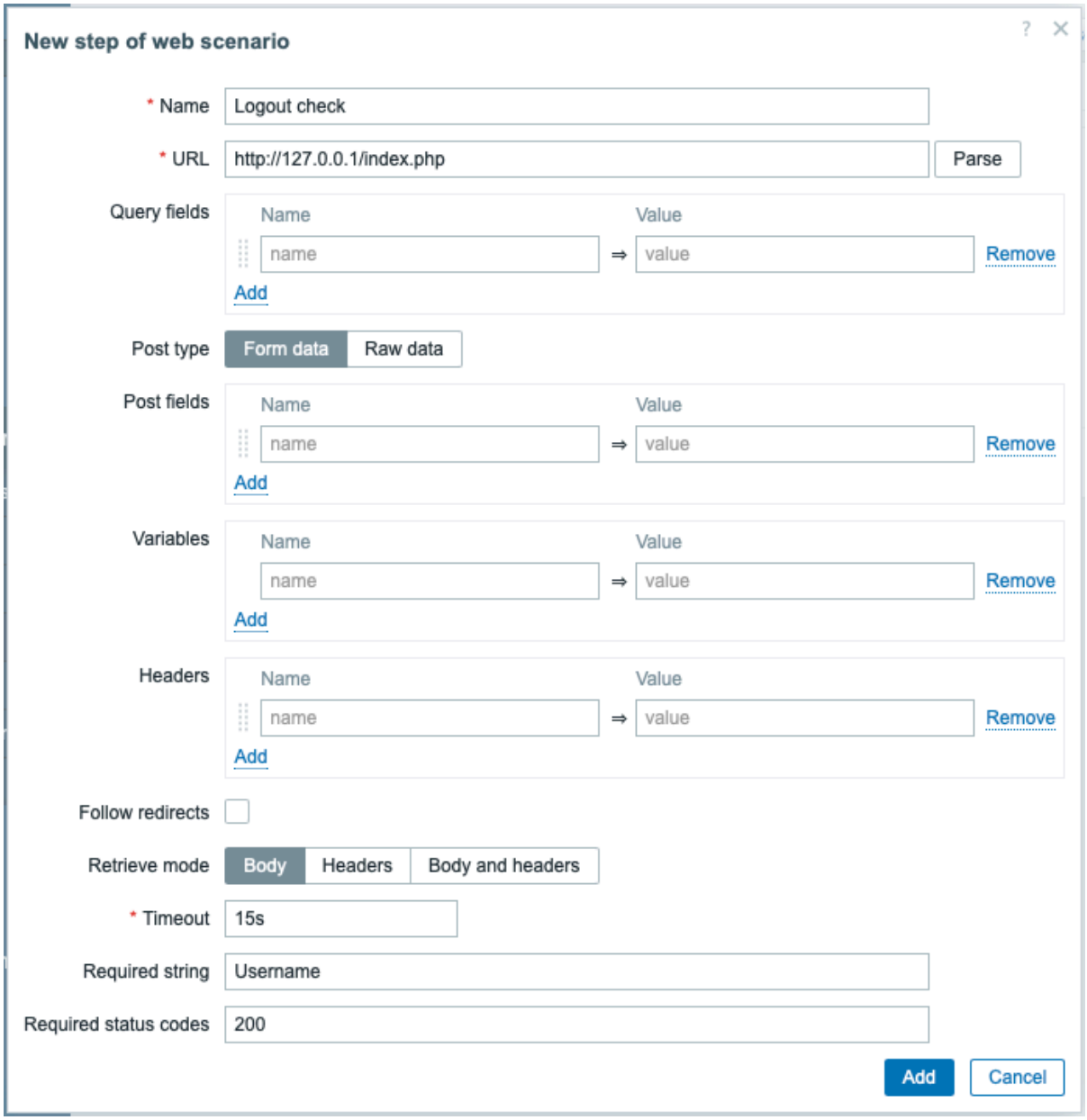
Cette étape utilise la variable {csrf_token} obtenue à l'étape 2



Étape 5 du scénario web

Pour confirmer la déconnexion, vérifiez la présence de la chaîne **Username**.

- Dans le champ **Name**, saisissez *Logout check*.
- Dans le champ **Required string**, saisissez *Username*.
- Renseignez les champs communs.



Configuration complète des étapes

Une configuration complète des étapes d'un scénario web devrait ressembler à ceci :

	Name	Timeout	URL	Required	Status codes	Action
1:	First page	15s	http://127.0.0.1/index.php	Zabbix SIA	200	Remove
2:	Log in	15s	http://127.0.0.1/index.php		200	Remove
3:	Log in Check	15s	http://127.0.0.1/index.php	Administrati…	200	Remove
4:	Log out	15s	http://127.0.0.1/index.php		200	Remove
5:	Log out check	15s	http://127.0.0.1/index.php	Username	200	Remove

Add

Vérifier les résultats

Enregistrez le scénario de surveillance web terminé.

Le scénario sera ajouté à l'hôte. Pour afficher les informations du scénario web, allez dans *Monitoring* → *Hosts*, repérez l'hôte dans la liste et cliquez sur le lien Web dans la dernière colonne.



Cliquez sur le nom du scénario pour voir des statistiques plus détaillées :



8 Supervision 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 supervision des machines virtuelles. Les paquets Zabbix sont déjà compilés avec ces options activées.

Les paramètres suivants du fichier de configuration du serveur Zabbix peuvent être modifiés pour la supervision des machines virtuelles :

- `StartVMwareCollectors`

Note:
Il est recommandé d'activer davantage de collecteurs que le nombre de services VMware supervisés ; sinon, la récupération des statistiques des compteurs de performance VMware peut être retardée par la récupération des données de configuration VMware (ce qui prend du temps pour les grandes installations).

 En général, la valeur de `StartVMwareCollectors` ne doit pas être inférieure à 2 et ne doit pas dépasser le double du nombre de services VMware supervisés : Nombre de services < `StartVMwareCollectors` < (Nombre de services * 2). Par exemple, lors de la supervision d'un service VMware, définissez `StartVMwareCollectors` sur 2 ; lors de la supervision 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 de vrais hôtes pour les entités découvertes. Pour plus d'informations, voir *Prototypes d'hôte*.

Configuration examples

For a basic example of how to set up Zabbix for monitoring VMware using the *VMware FQDN* template, see *Monitor VMware with Zabbix*.

For a more detailed example of how to create a host, a low-level discovery rule, and a host prototype for monitoring VMware, see *Setup example*.

Journalisation étendue Les données collectées par les processus *vmware collector* peuvent être consignées pour un débogage détaillé en utilisant le niveau de débogage 5. Le niveau de débogage peut être configuré dans les fichiers de configuration server et proxy ou à l'aide de l'option de contrôle d'exécution `-R log_level_increase="vmware collector,N"`, où « N » est le 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 ramener le niveau de débogage à sa valeur par défaut (3) en exécutant la commande `-R log_level_decrease`.

Dépannage

- En cas de métriques indisponibles, veuillez vous assurer qu'elles ne sont pas rendues indisponibles ou désactivées par défaut dans les versions récentes de VMware vSphere, et qu'aucune limite n'est appliquée aux requêtes de la base de données des métriques de performance. Pour plus d'informations, consultez ZBX-12094.
- Si l'erreur `config.vpxd.stats.maxQueryMetrics is invalid or exceeds the maximum number of characters permitted` s'affiche, ajoutez un paramètre `config.vpxd.stats.maxQueryMetrics` aux paramètres du serveur vCenter. La valeur de ce paramètre doit être identique à celle de `maxQuerysize` dans le fichier *web.xml* de VMware. Pour plus d'informations, consultez l'article de la *VMware Knowledge Base*.
- Si vous pensez que votre installation Zabbix utilise trop de mémoire, consultez Profiling excessive memory usage with tcmalloc.

1 Clés d'élément de supervision 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 afficher tous les détails.

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 de compteur de performances 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 performances du datastore VMware.	
vmware.datastore.property	La propriété du datastore VMware.	
vmware.datastore.read	Le temps nécessaire pour 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 pour 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.	

Clé d'élément	Description	Groupe d'éléments
vmware.dc.tags.get	Le tableau des balises du centre de données VMware.	
vmware.dvswitch.discovery	La découverte des commutateurs distribués VMware vSphere.	Commutateur distribué vSphere
vmware.dvswitch.fetchports.get	Les données des ports du commutateur distribué VMware vSphere.	
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, dépend de la gestion de l'alimentation ou du 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 pour 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 pour 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.	
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 gonflée 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 performances 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 récapitulatif 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 pool de ressources VMware.	Pool de ressources
vmware.rp.memory	Les métriques de mémoire du pool de ressources VMware.	
vmware.alarms.get	Les données des alarmes du centre virtuel VMware.	Centre virtuel
vmware.vm.alarms.get	Les données des alarmes de la machine virtuelle VMware.	Machine virtuelle
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.	

Clé d'élément	Description	Groupe d'éléments
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 du temps CPU passé à attendre le 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.swapped	La quantité de mémoire physique invitée 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 gonflée 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 hôte consommée pour sauvegarder les pages de mémoire physique invitée.	
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 invitée de la machine virtuelle VMware.	
vmware.vm.memory.size.usage.host	L'utilisation de la mémoire hôte de la machine virtuelle VMware.	
vmware.vm.memory.usage	Le pourcentage de mémoire physique hôte qui a été consommé.	
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 réseau de la machine virtuelle VMware pendant l'intervalle.	
vmware.vm.perfcounter	La valeur du compteur de performances 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 de l'instantané de la machine virtuelle VMware.	
vmware.vm.state	L'état de la machine virtuelle VMware.	
vmware.vm.storage.committed	L'espace de stockage validé 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 pour une lecture depuis le disque virtuel.	
vmware.vm.storage.totalwritelatency	Le temps moyen nécessaire pour une écriture vers le disque virtuel.	
vmware.vm.storage.uncommitted	L'espace de stockage non validé 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 invités 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 de retour : *Log*.

Paramètres :

- **url** - l'URL du service VMware ;
- **mode** - *all* (par défaut) ou *skip* - ignorer le traitement des données plus anciennes ;
- **severity** - filtrer par gravité : *error*, *warning*, *info* ou *user*. Ce paramètre doit être 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 plus récentes.

vmware.datastore.alarms.get[url,uuid]

 Les données des alarmes du datastore VMware.
 Valeur de retour : objet JSON.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique 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 global unique du datastore VMware.

Cet élément fonctionne avec vSphere 6.5 et les versions plus récentes.

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 centre de données VMware.
 Valeur de retour : objet JSON.

Paramètres :

- **url** - l'URL du service VMware ;
- **id** - l'ID du centre de données 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 commutateur distribué VMware vSphere.
 Valeur de retour : objet JSON.

Paramètres :

- **url** - l'URL du service VMware ;
- **uuid** - l'identifiant global unique du DVSwitch VMware.
- **filter** - une chaîne unique contenant 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 global unique 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 des disques 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.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 global unique 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 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.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 global unique de l'hyperviseur VMware.

Cet élément fonctionne avec vSphere 6.5 et les versions plus récentes.

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 global unique 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 de l'instantané 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.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 global unique de la machine virtuelle VMware.

Cet élément fonctionne avec vSphere 6.5 et 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.

2 Champs clés de la 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.

Clé d'élément		

	"resource_pool"	Un tableau contenant les données du pool de ressources, y compris l'identifiant du groupe de ressources, le tableau des tags, le chemin du pool de ressources et le nombre de machines virtuelles.
		
		Structure du tableau :
		`[{`
		`"rpid":"identifiant du groupe de ressources",`
		`"tags":[{}],`
		`"rpath":"chemin du groupe de ressources",`
		`"vm_count":0`
		`}]`

	"tags"	Pour la structure du tableau `"tags"`, voir le champ "tags".
		Un tableau contenant les tags avec le nom du tag, la description et la catégorie.
		
		Structure du tableau :
		`[{`
		`"tag":"nom du tag",`
		`"tag_description":"description du tag",`
		`"category":"catégorie du tag"`
		`}]`

vmware.datastore.discovery		
Effectue la découverte des datastores.	{#DATASTORE}	Nom du datastore.

	{#DATASTORE.EXTENT}	Un tableau contenant l'identifiant de partition de l'étendue du datastore et le nom de l'instance.
		
		Structure du tableau :
		`[{`
		`"partitionid":1,`
		`"instance":"nom"`
		`}]`

	{#DATASTORE.TYPE}	Type de datastore. Exemples de valeurs : `VMFS`, `NFS`, `vsan`, etc.
	{#DATASTORE.UUID}	Identifiant du datastore.

	"tags"	Un tableau contenant les tags avec le nom du tag, la description et la catégorie.
		
		Structure du tableau :
		`[{`
		`"tag":"nom du tag",`
		`"tag_description":"description du tag",`
		`"category":"catégorie du tag"`
		`}]`

vmware.dc.discovery		
Effectue la découverte des centres de données.	{#DATACENTER}	Nom du centre de données.
	{#DATACENTERID}	Identifiant du centre de données.

	"tags"	Un tableau contenant les tags avec le nom du tag, la description et la catégorie.
		
		Structure du tableau :
		`[{`
		`"tag":"nom du tag",`
		`"tag_description":"description du tag",`
		`"category":"catégorie du tag"`
		`}]`

vmware.dvswitch.discovery		

Clé d'élément		
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 : - préférence pour l'IP qui partage le sous-réseau IP avec l'IP du vCenter ; - préférence pour l'IP du sous-réseau IP avec la passerelle par défaut ; - préférence pour l'IP de l'interface ayant l'identifiant le plus faible.
	{#CLUSTER.NAME}	Nom du cluster, peut être vide.
	{#DATACENTER.NAME}	Nom du centre de données.
	{#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" représente un type de conteneur inconnu.

	"resource_pool"	Un tableau contenant les données du pool de ressources, y compris l'identifiant du groupe de ressources, le tableau des tags, le chemin du pool de ressources et le nombre de machines virtuelles.
		
		Structure du tableau :
		`[{`
		`"rpid":"identifiant du groupe de ressources",`
		`"tags":[{}],`
		`"rpath":"chemin du groupe de ressources",`
		`"vm_count":0`
		`}]`
		
		Pour la structure du tableau `"tags"`, voir le champ "tags".

	"tags"	Un tableau contenant les tags avec le nom du tag, la description et la catégorie.
		
		Structure du tableau :
		`[{`
		`"tag":"nom du tag",`
		`"tag_description":"description du tag",`
		`"category":"catégorie du tag"`
		`}]`

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.

Clé d'élément		

	"datastore_extent"	Un tableau contenant le nom de l'instance de l'étendue du datastore et l'identifiant de partition.
		
		Structure du tableau :
		`[{`
		`"partitionid":1,`
		`"instance":"nom"`
		`}]`

	"tags"	Un tableau contenant les tags avec le nom du tag, la description et la catégorie.
		
		Structure du tableau :
		`[{`
		`"tag":"nom du tag",`
		`"tag_description":"description du tag",`
		`"category":"catégorie du tag"`
		`}]`

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 centre de données.
	{#DATASTORE.NAME}	Nom du datastore.
	{#DATASTORE.UUID}	Identifiant du datastore.
	{#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}	La chaîne des dossiers parents de la machine virtuelle peut être utilisée comme valeur pour des groupes imbriqués ; les noms de dossiers sont combinés avec "/". Peut être vide.
	{#VM.TOOLS.STATUS}	État des outils de 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 "root" "Resources". Les noms de dossiers sont combinés avec "/".
	{#VM.SNAPSHOT.COUNT}	Nombre d'instantanés de la VM.

Clé d'élément		

	"tags"	Un tableau contenant les tags avec le nom du tag, la description et la catégorie.
		
		Structure du tableau :
		`[{`
		`"tag":"nom du tag",`
		`"tag_description":"description du tag",`
		`"category":"catégorie du tag"`
		`}]`

	"vm_customattribute"	Un tableau des attributs personnalisés de la machine virtuelle (s'ils sont définis).
		
		Structure du tableau :
		`[{`
		`"name":"nom du champ personnalisé",`
		`"value":"valeur du champ personnalisé"`
		`}]`

	"net_if"	Un tableau des interfaces réseau de la machine virtuelle.
		
		Structure du tableau :
		`[{`
		`"ifname": "nom de l'interface",`
		`"ifdesc": "description de l'interface",`
		`"ifmac": "00:00:00:00:00:00",`
		`"ifconnected": true,`
		`"iftype": "type d'interface",`
		`"ifbackingdevice": "périphérique de support de l'interface",`
		`"ifdvswitch_uuid": "uuid du commutateur de l'interface",`
		`"ifdvswitch_portgroup": "groupe de ports du commutateur de l'interface",`
		`"ifdvswitch_port": "port du commutateur de l'interface",`
		`"ifip": ["adresses IP de l'interface"]`
		`}]`
		
		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 d'interface.
	{#IFBACKINGDEVICE}	Nom du périphérique de support.
	{#IFDVSWITCH.UUID}	Identifiant unique du commutateur distribué vSphere.
	{#IFDVSWITCH.PORTGROUP}	Groupe de ports distribué.
	{#IFDVSWITCH.PORT}	Port du commutateur distribué 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		

Clé d'élément		
Effectue la découverte du système de fichiers de la machine virtuelle.	{#FSNAME}	Nom du système de fichiers.

3 exemples JSON pour les éléments VMware

Vue d'ensemble Cette section fournit des informations supplémentaires sur les objets JSON renvoyés par divers items.

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 ayant 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
        },
        {
            "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
        },
        {
            // autres alarmes
        }
    ]
}
```

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
}
```


4 Exemple de configuration de la supervision 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.

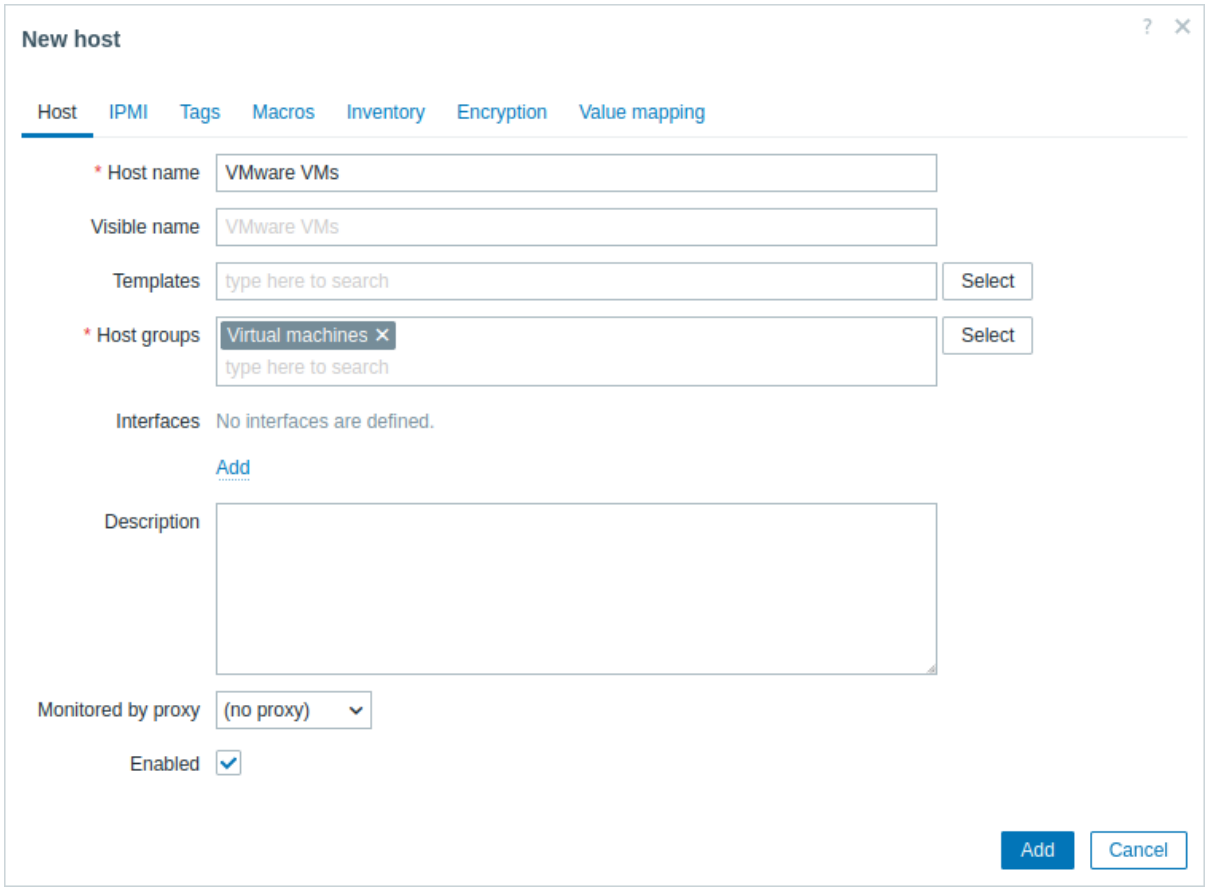
Avant de continuer, définissez le paramètre `StartVMwareCollectors` dans le fichier de configuration du serveur Zabbix à 2 ou plus (la valeur par défaut est 0).

Créer un hôte

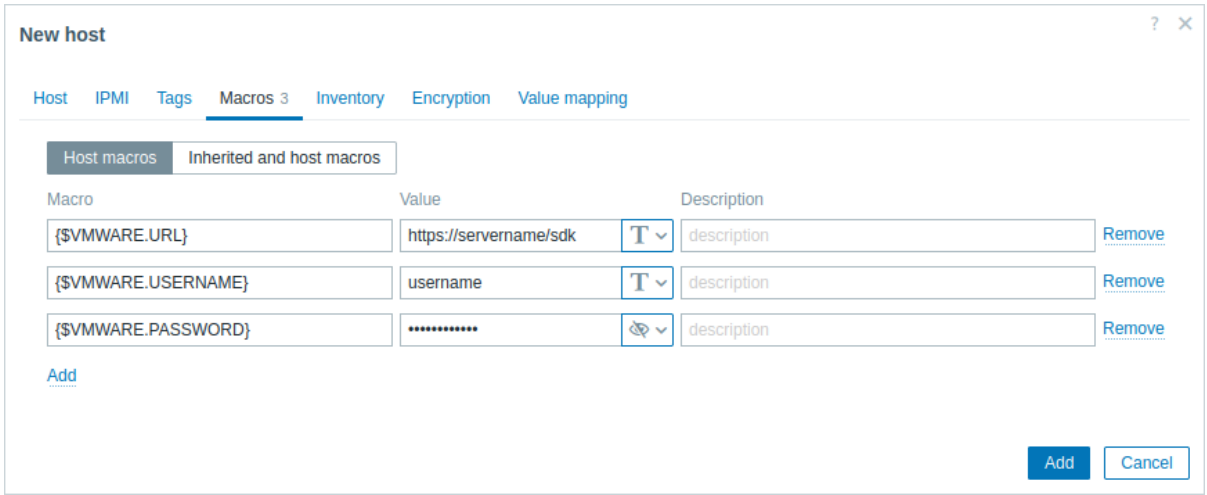
1. Accédez à *Collecte de données* → *Hôtes*.

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, « Virtual machines »).



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



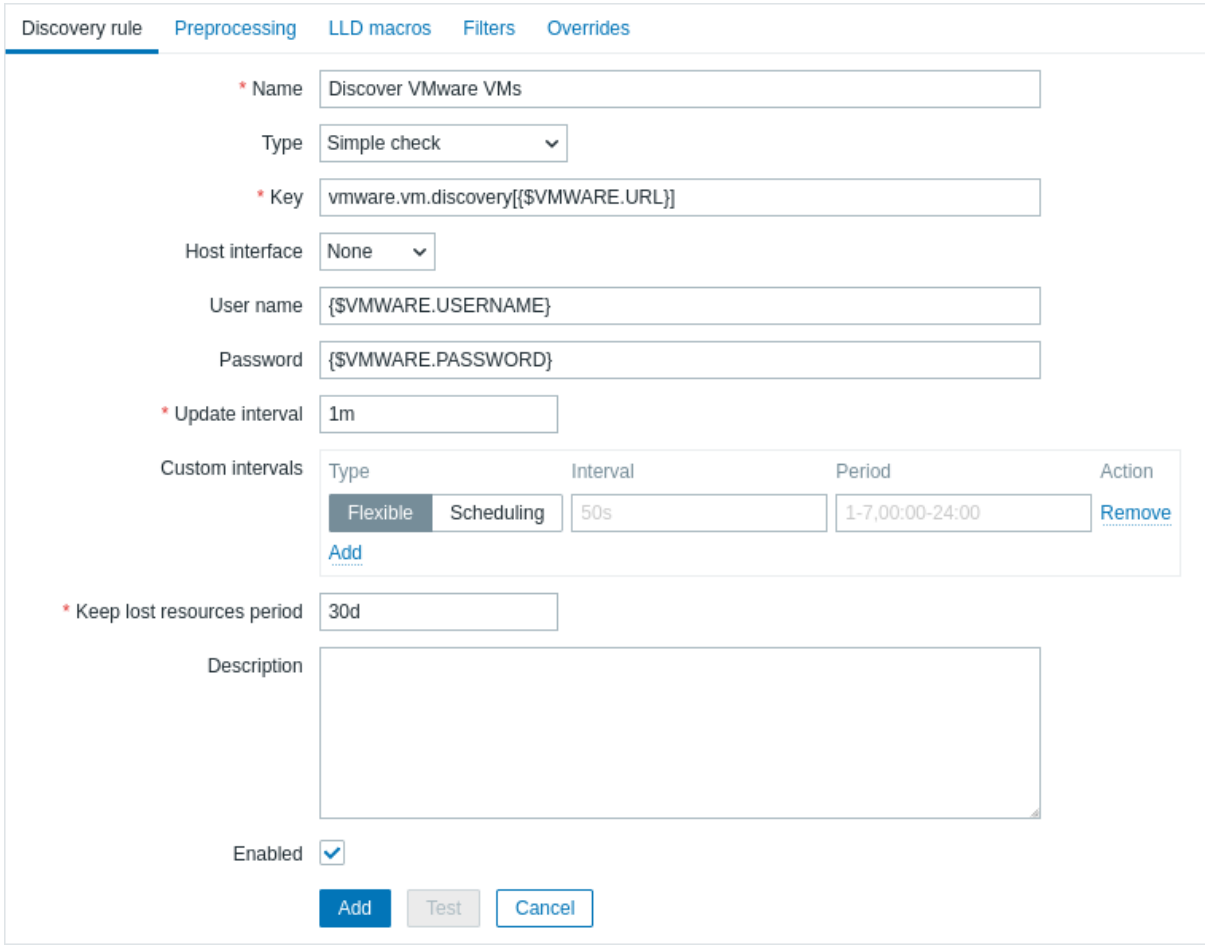
3. Cliquez sur le bouton *Ajouter* pour créer l'hôte. Cet hôte représentera votre environnement VMware.

Créer une règle de découverte de bas niveau

1. Cliquez sur *Discovery* 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 *Name*, saisissez un nom de règle de découverte de bas niveau (par exemple, « Discover VMware VMs »).
- Dans le champ *Type*, sélectionnez « Simple check ».
- Dans le champ *Key*, saisissez la clé d'élément intégrée pour découvrir les machines virtuelles VMware : `vmware.vm.discovery[{$VMW`
- Dans les champs *User name* et *Password*, saisissez les macros correspondantes précédemment configurées sur l'hôte.



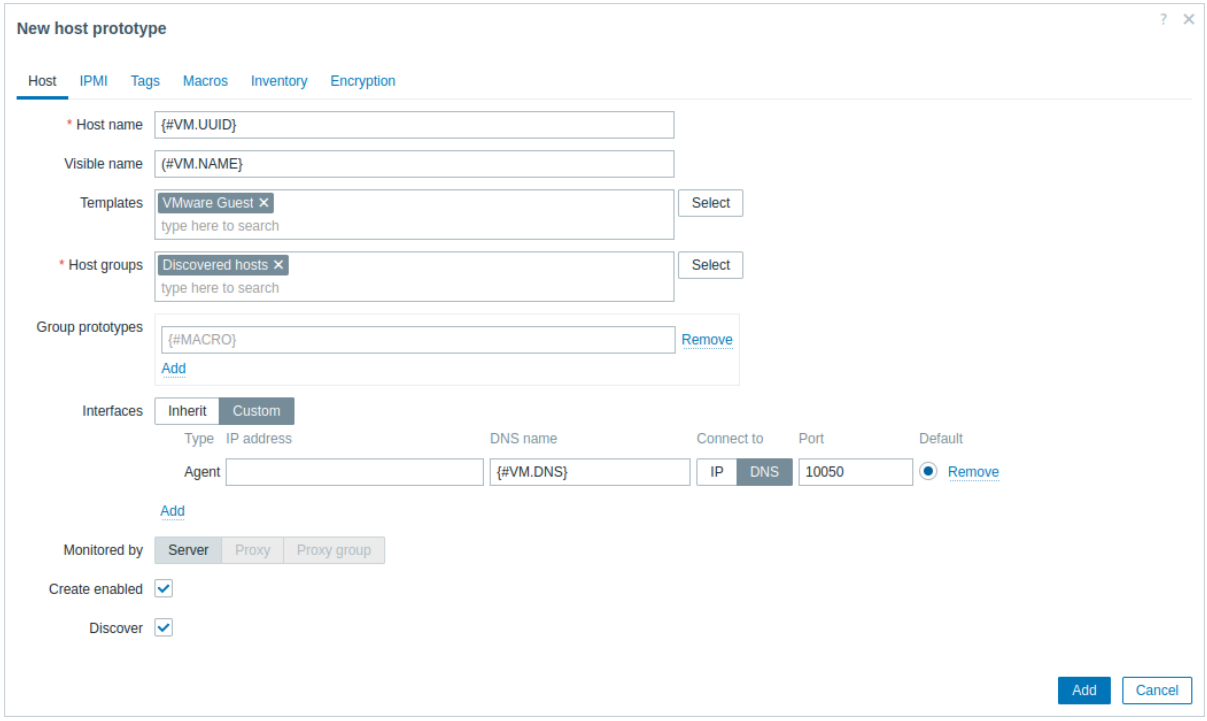
3. Cliquez sur le bouton *Add* pour créer la règle de découverte de bas niveau. Cette règle de découverte détectera les machines virtuelles dans votre environnement VMware.

Créer un prototype d'hôte

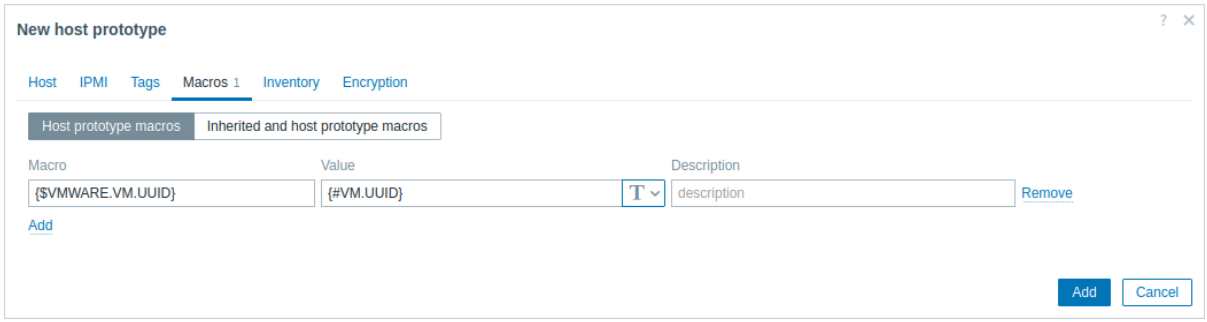
1. Dans la liste des règles de découverte de bas niveau, cliquez sur *Prototypes d'hôte* pour la règle de découverte de bas niveau créée précédemment.

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}]*).

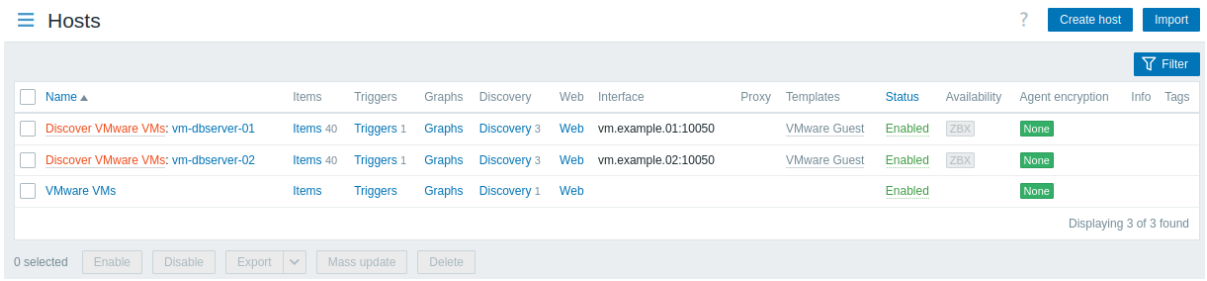


3. Cliquez sur le bouton *Ajouter* pour créer le prototype d'hôte. Ce prototype d'hôte sera utilisé pour créer des hôtes pour les machines virtuelles découvertes par la règle de découverte de bas niveau créée précédemment.

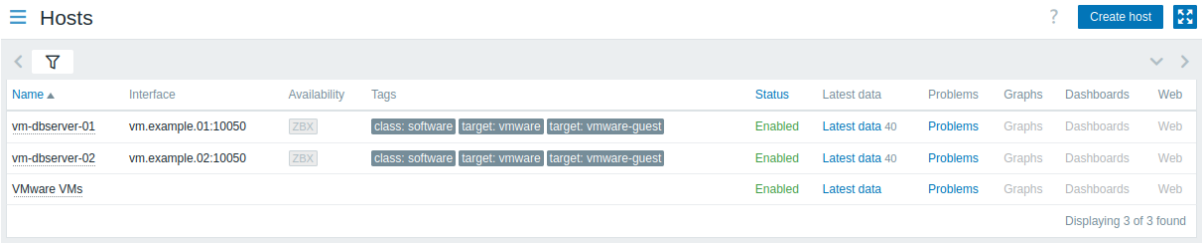
Voir les hôtes et les métriques

Une fois le prototype d'hôte créé, la règle de découverte de 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

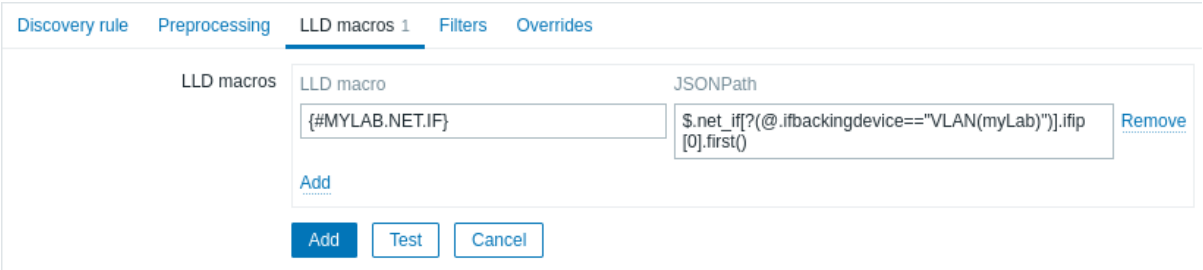
La clé d'élément `vmware.vm.discovery[{$VMWARE.URL}]`, configurée dans la section *Créer une règle de découverte de bas niveau*, renvoie les données des interfaces réseau dans le champ "net_if" :


```
"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": []
      }
    ]
```

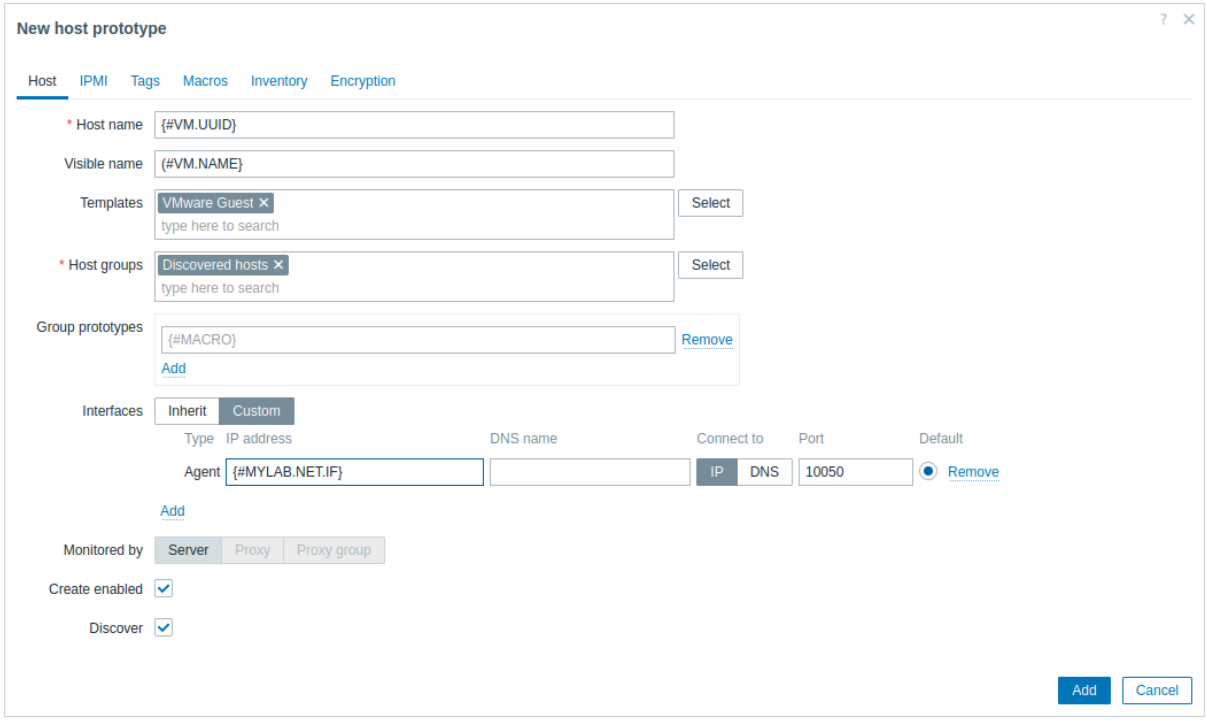

Ces données peuvent être utilisées pour configurer une interface d'hôte personnalisée.

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*.



9 Maintenance

Aperçu Vous pouvez définir des périodes de maintenance pour les hôtes et les groupes d'hôtes dans Zabbix.

En outre, il est possible de définir une maintenance uniquement pour un seul déclencheur (ou un sous-ensemble de déclencheurs) en spécifiant des tags de déclencheur. Dans ce cas, la maintenance sera activée uniquement pour ces déclencheurs ; 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 comme d'habitude et des événements sont créés si nécessaire. Cependant, les escalades de problèmes sont mises en pause pour les hôtes/déclencheurs en maintenance, si l'option *Pause operations for suppressed problems* est cochée dans la configuration de l'action. Dans ce cas, les étapes d'escalade pouvant inclure l'envoi de notifications ou des commandes distantes seront ignorées tant que dure la période de maintenance. Notez que les opérations de résolution et de mise à jour des problèmes ne sont pas supprimées pendant la maintenance, seules les escalades le sont.

Par exemple, si des étapes d'escalade sont planifiées à 0, 30 et 60 minutes après le début d'un problème, et qu'il y a une maintenance d'une demi-heure allant de 10 à 40 minutes après l'apparition d'un problème réel, les deuxième et troisième étapes seront exécutées une demi-heure plus tard, c'est-à-dire à 60 minutes et 90 minutes (à condition que le problème existe toujours). De même, si un problème survient pendant la maintenance, l'escalade commencera après la maintenance.

Pour recevoir normalement les notifications de problème pendant la maintenance (sans délai), vous devez décocher l'option *Pause operations for suppressed problems* 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 cours d'exécution pendant la maintenance. Les maintenances sont recalculées chaque minute ou dès que le cache de configuration est rechargé si des modifications sont apportées à la période de maintenance.

Les processus timer vérifient si le statut de l'hôte doit être changé vers/depuis la maintenance à 0 seconde de chaque minute. De plus, chaque seconde, le processus timer vérifie si des maintenances doivent être démarrées/arrêtées selon qu'il y a des modifications dans les périodes de maintenance après la mise à jour de la configuration. Ainsi, la vitesse de démarrage/arrêt des périodes de maintenance dépend de l'intervalle de mise à jour de la configuration (10 secondes par défaut). Notez que les modifications de période de maintenance n'incluent pas les paramètres *Active since/Active till*. De plus, si un hôte/groupe d'hôtes est ajouté à une période de maintenance active existante, les modifications ne seront activées par le processus timer qu'au début de la minute suivante.

Notez que lorsqu'un hôte entre en maintenance, les processus timer du serveur Zabbix liront tous les problèmes ouverts pour vérifier s'il est nécessaire de les supprimer. Cela peut avoir un impact sur les performances s'il y a de nombreux problèmes ouverts. Le serveur Zabbix lira 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 la maintenance « sans données »). Les données sont ensuite ignorées par le serveur si l'option « sans collecte de données » est définie.

Lorsque la maintenance « sans données » se termine, les déclencheurs utilisant la fonction nodata() ne se déclencheront pas avant la prochaine vérification pendant la période qu'ils surveillent.

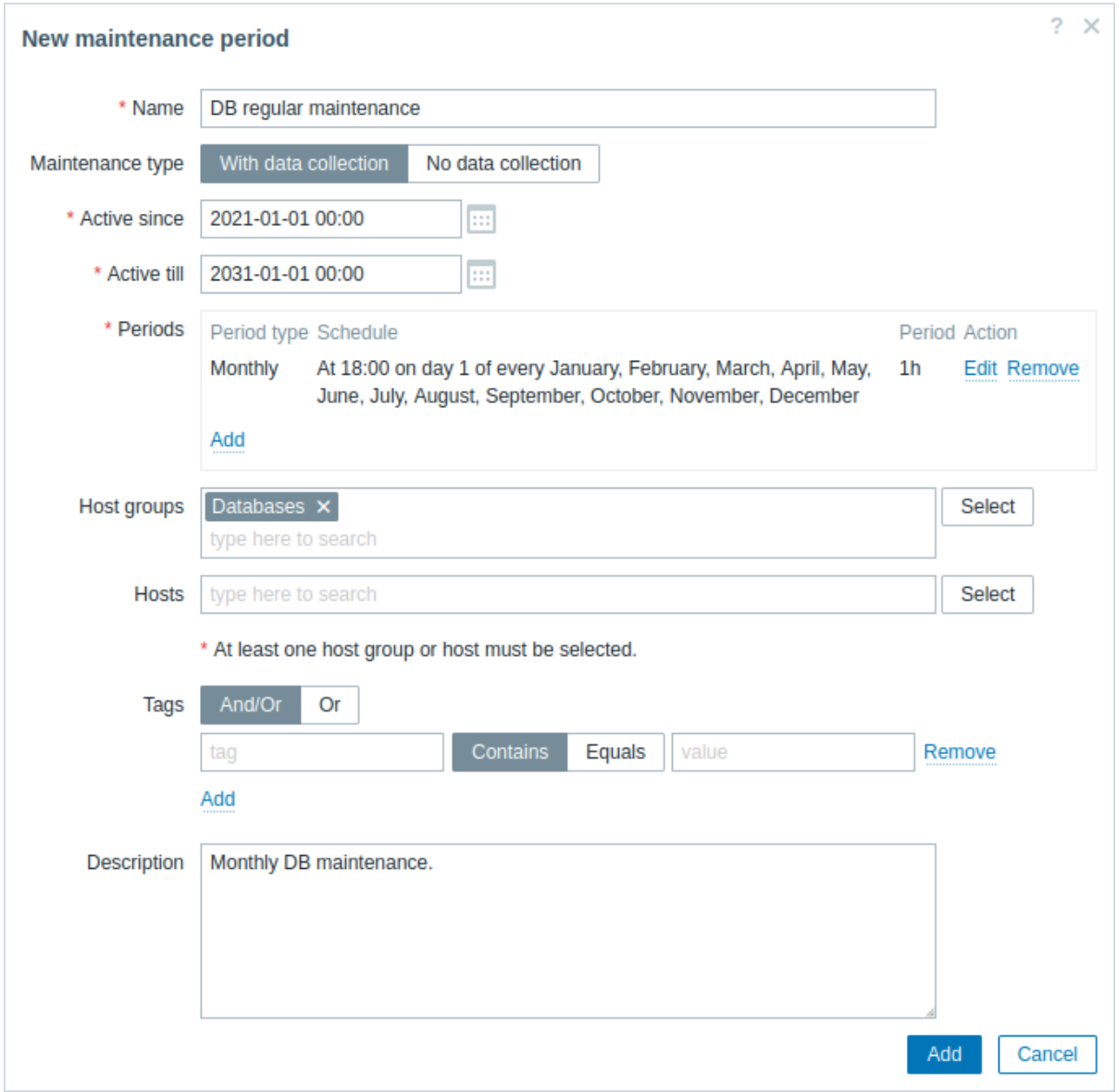
Si un élément journal est ajouté alors qu'un hôte est en maintenance et que la maintenance se termine, seules les nouvelles entrées du fichier journal depuis la fin de la maintenance seront collectées.

Si une valeur horodatée est envoyée pour un hôte qui est dans un type de maintenance « sans données » (par exemple en utilisant Zabbix sender), cette valeur sera rejetée ; toutefois, il est possible d'envoyer une valeur horodatée pour une période de maintenance expirée et elle sera acceptée.

Si la période de maintenance, les hôtes, les groupes ou les tags sont modifiés par l'utilisateur, les changements ne prendront effet qu'après la synchronisation du cache de configuration.

Configuration Pour configurer une période de maintenance :

1. Allez dans : *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.

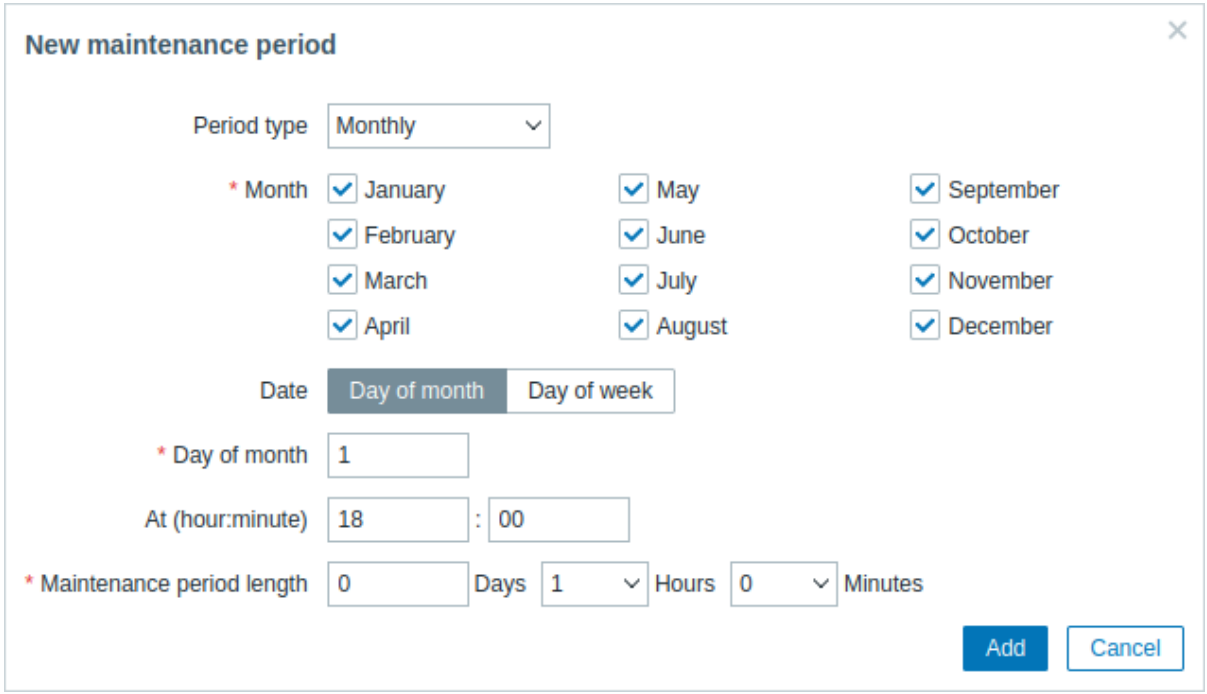


Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge.

Parameter	Description
Name	Nom de la période de maintenance.
Maintenance type	Deux types de maintenance peuvent être définis : **With data collection** - les données seront collectées par le serveur pendant la maintenance, les déclencheurs seront traités ; **No data collection** - les données ne seront pas collectées par le serveur pendant la maintenance. Voir Effect of maintenance periods pour savoir comment chaque type affecte le rapport de disponibilité.
Active since	La date et l'heure à partir desquelles l'exécution des périodes de maintenance devient active. *Remarque :* définir uniquement cette heure n'active pas une période de maintenance ; les périodes de maintenance doivent être configurées dans *Periods* (voir ci-dessous).
Active till	La date et l'heure auxquelles l'exécution des périodes de maintenance cesse d'être active.
Periods	Ce bloc vous permet de définir les jours et heures exacts auxquels la maintenance a lieu. Un clic sur Add ouvre une fenêtre contextuelle avec un formulaire flexible *Maintenance period* dans lequel vous pouvez définir le calendrier de maintenance. Voir Maintenance periods pour une description détaillée.
Host groups	Sélectionnez les groupes d'hôtes pour lesquels la maintenance sera activée. La maintenance sera activée pour tous les hôtes du ou des groupes d'hôtes spécifiés. Ce champ prend en charge l'autocomplétion ; lorsque vous commencez à y saisir du texte, une liste déroulante de tous les groupes d'hôtes disponibles s'affiche. La spécification d'un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués. Ainsi, la maintenance sera également activée sur les hôtes des groupes imbriqués.
Hosts	Sélectionnez les hôtes pour lesquels la maintenance sera activée. Ce champ prend en charge l'autocomplétion ; lorsque vous commencez à y saisir du texte, une liste déroulante de tous les hôtes disponibles s'affiche.
Tags	Spécifiez des tags pour supprimer les problèmes avec des tags correspondants sur les hôtes en maintenance. Plusieurs conditions peuvent être définies. La correspondance des noms de tags est toujours sensible à la casse. Deux opérateurs sont disponibles pour chaque condition : **Contains** - inclure les noms de tags spécifiés lorsque les valeurs des tags contiennent la chaîne saisie (correspondance de sous-chaîne, sensible à la casse) ; **Equals** - inclure les noms et valeurs de tags spécifiés (sensible 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. Les tags ne peuvent être spécifiés que si le mode *With data collection* est sélectionné.
Description	Description de la période de maintenance.

Périodes de maintenance

La fenêtre de période de maintenance permet de planifier une maintenance récurrente ou ponctuelle. Le formulaire est dynamique et les champs disponibles changent 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.
Quotidienne	Configurez une période de maintenance quotidienne : *Tous les X 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 X jour(s)* est supérieur à ”1”, le jour de départ est le jour auquel correspond l’heure *Actif depuis*. Exemples : - si *Actif depuis* est défini sur ”2021-01-01 12:00”, *Tous les X 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 X 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 X 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 X semaine(s)* est supérieur à ”1”, la semaine de départ est la semaine à laquelle correspond l’heure *Actif depuis*. Pour des exemples, voir la description du paramètre *Quotidienne* ci-dessus.

Type de période	Description
Mensuelle	Configurez une période de maintenance mensuelle : *Mois* - sélectionnez tous les mois durant 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 (*Quotidiennes*, *Hebdomadaires*, *Mensuelles*) sont planifiées, le fuseau horaire du serveur Zabbix est utilisé. Afin de garantir un comportement prévisible des périodes de maintenance récurrentes, il est nécessaire d'utiliser un fuseau horaire commun pour toutes les parties de Zabbix.

Une fois terminé, appuyez sur *Add* pour ajouter la période de maintenance au bloc *Periods*.

Notez que les changements d'heure d'été/hiver (DST) n'affectent pas la durée de la maintenance. Par exemple, supposons qu'une maintenance de deux heures soit configurée pour commencer habituellement à 01:00 et se terminer à 03:00 :

- si, après une heure de maintenance (à 02:00), un changement DST se produit et que l'heure courante passe de 02:00 à 03:00, la maintenance se poursuivra pendant encore une heure (jusqu'à 04:00) ;
- si, après deux heures de maintenance (à 03:00), un changement DST se produit et que l'heure courante 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 qui est sautée lors d'un changement DST, alors la maintenance ne démarrera pas.

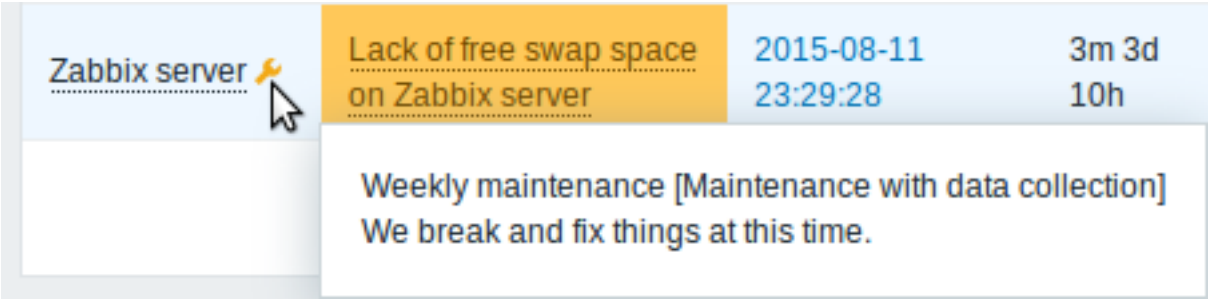
Si une période de maintenance est définie sur « 1 day » (la durée réelle de la maintenance est de 24 heures, puisque Zabbix calcule les jours en heures), commence à 00:00 et se termine à 00:00 le jour suivant :

- la maintenance s'arrêtera à 01:00 le jour suivant si l'heure courante avance d'une heure ;
- la maintenance s'arrêtera à 23:00 ce jour-là si l'heure courante 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 :

- *Tableaux de bord*
- *Monitoring* > *Problems*
- *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.

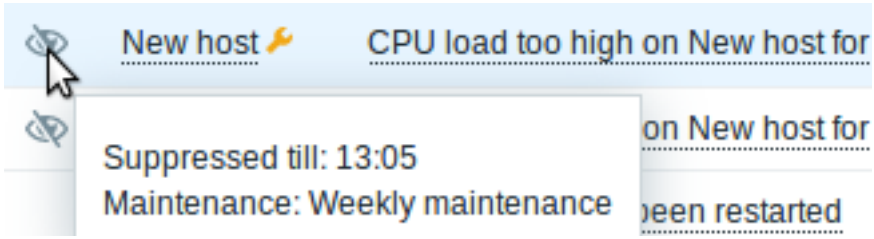
De plus, les hôtes en maintenance ont un arrière-plan orange dans *Monitoring* > *Maps*.

Affichage des problèmes supprimés

Normalement, les problèmes des hôtes en maintenance sont supprimés, c'est-à-dire qu'ils ne sont pas affichés dans l'interface web. Cependant, il est également possible de configurer l'affichage des problèmes supprimés en sélectionnant l'option *Afficher les problèmes supprimés* 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 supprimés sont affichés, l'icône suivante est affichée : . Le survol de l'icône avec la souris affiche davantage de détails.



Calcul des files d'attente pendant la maintenance Les files d'attente affichées dans le frontend Zabbix (Administration > Queue) sont calculées par le serveur Zabbix. Elles n'incluent pas les éléments en maintenance sans collecte de données : la longueur de la file d'attente est toujours nulle pour ces éléments, même lorsque leurs valeurs sont retardées. Les éléments retardés en maintenance avec collecte de données 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 contrôles internes calculés sur les proxys Zabbix (par exemple, `zabbix[queue,,]` et `zabbix[stats,,,queue,,]`) signalent les éléments retardés indépendamment du statut de maintenance sur le serveur Zabbix.

Par conséquent, des longueurs de file d'attente différentes peuvent être signalées pour les mêmes éléments en maintenance sans collecte de données par le frontend Zabbix et par les contrôles internes sur les proxys Zabbix.

10 Expressions régulières

Aperçu Perl Compatible Regular Expressions (PCRE, PCRE2) sont pris en charge dans Zabbix.

Il existe deux manières d'utiliser les expressions régulières dans Zabbix :

- saisie manuelle d'une expression régulière
- en utilisant 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 peut pas commencer par @, car ce symbole est utilisé dans Zabbix pour faire référence aux expressions régulières globales.

> **Warning:**
> Il est possible d'épuiser la pile lors de l'utilisation d'expressions régulières. Consultez la page de manuel pcrestack pour plus d'informations.

Notez qu'en correspondance multiligne, les ancres ^ et $ correspondent respectivement au début/à la fin de chaque ligne, au lieu du début/de la fin de la chaîne entière.

Voir aussi des exemples d'échappement correct dans différents contextes.

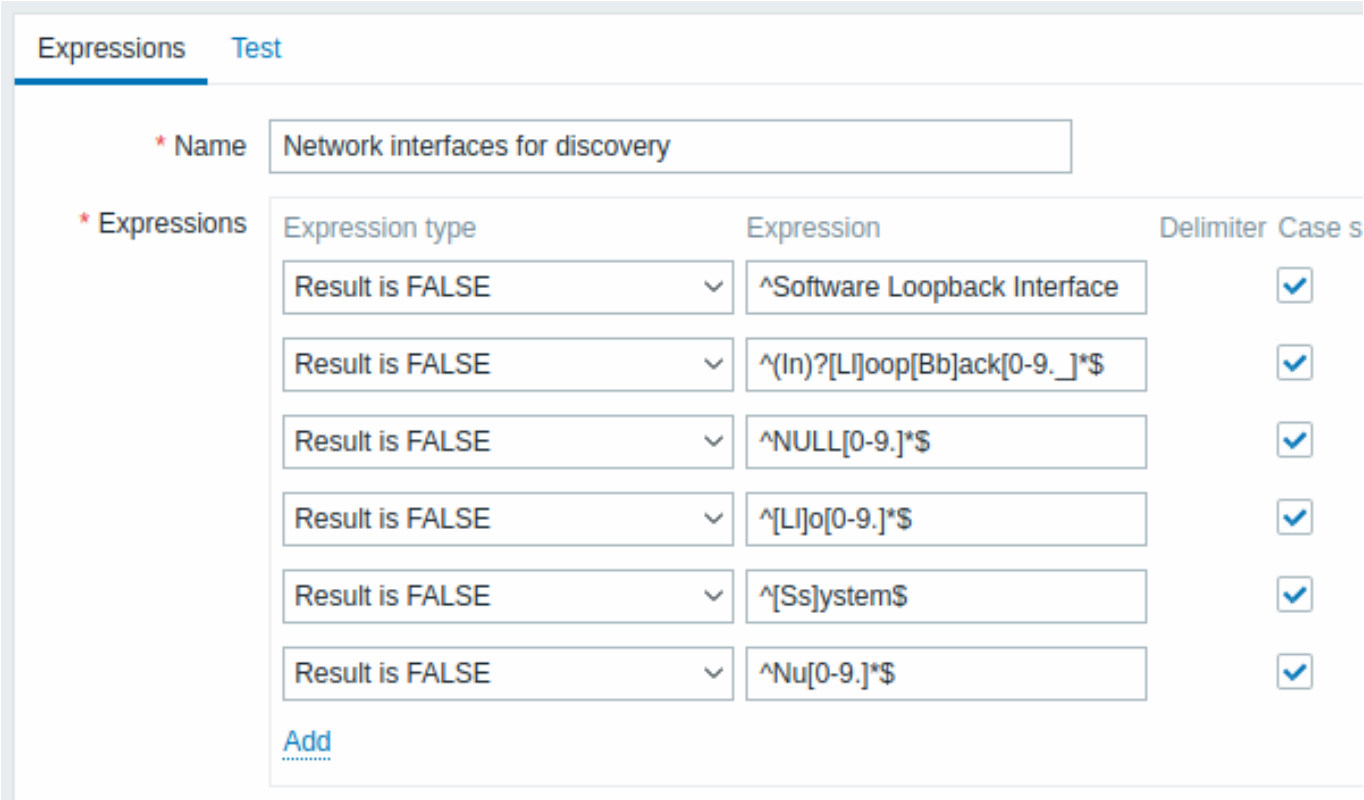
Expressions régulières globales Il existe un éditeur avancé pour créer et tester des expressions régulières complexes dans l'interface Zabbix.

Une fois qu'une expression régulière a été créée de cette façon, elle peut être utilisée à plusieurs endroits dans le frontend en se référant à son nom, préfixé par @, par exemple, *@mycustomregexp*.

Pour créer une expression régulière globale :

- Allez dans : *Administration* → *Général*
- Sélectionnez *Expressions régulières* dans le menu déroulant
- Cliquez sur *Nouvelle expression régulière*

L'onglet **Expressions** permet de définir le nom de l'expression régulière et d'ajouter des sous-expressions.



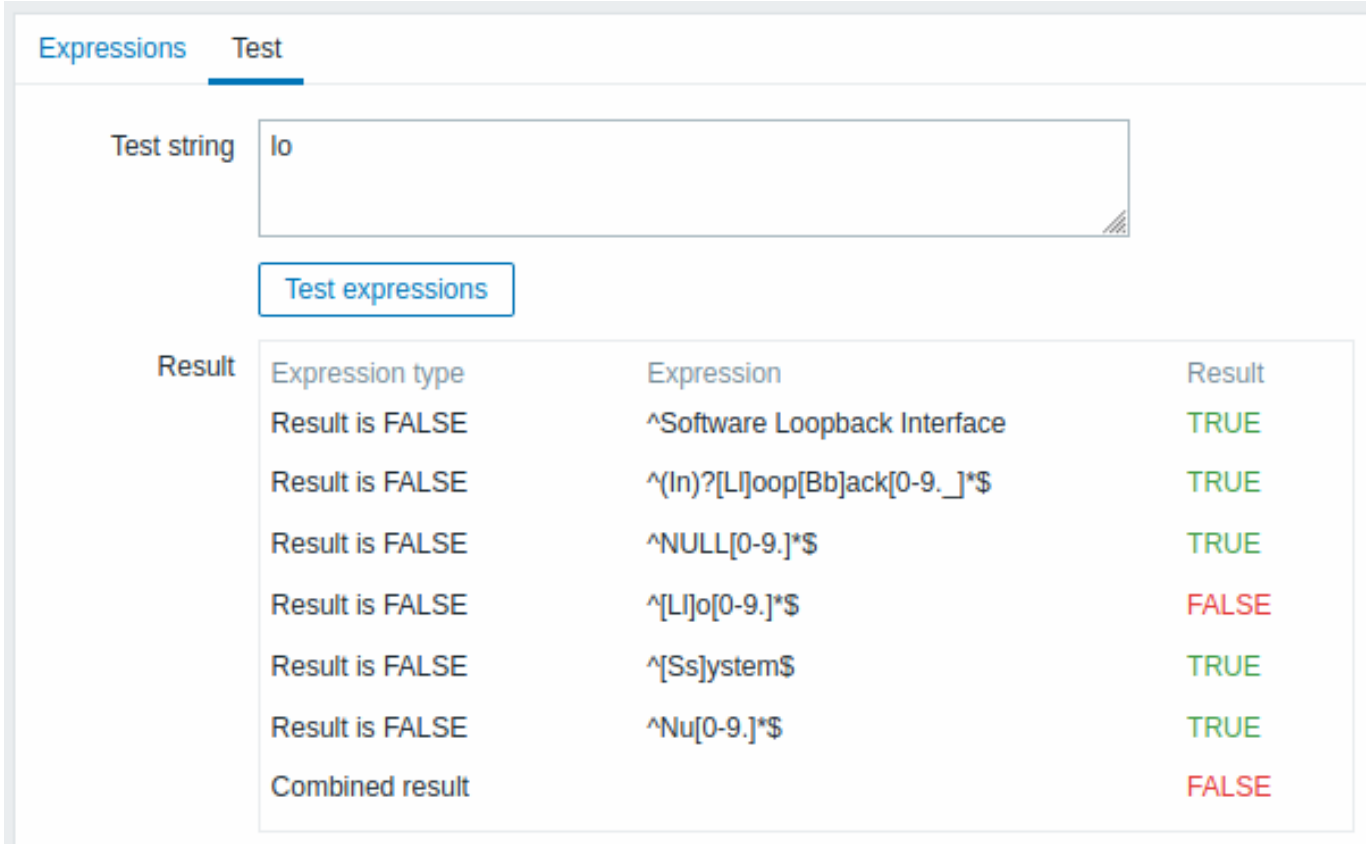
Tous les champs de saisie obligatoires sont marqués d'un astérisque rouge.

Paramètre	Description
Nom	Définissez le nom de l'expression régulière. Tous les caractères Unicode sont autorisés.
Expressions	Cliquez sur *Ajouter* dans le bloc Expressions pour ajouter une nouvelle sous-expression.
Type d'expression	Sélectionnez le type d'expression : **Chaîne de caractères incluse** - correspond à la sous-chaîne **Toute chaîne de caractères incluse** - correspond à toutes les sous-chaînes d'une liste délimitée. La liste délimitée comprend une virgule (,), un point (.) ou une barre oblique (/). **Chaîne de caractères non incluse** - correspond à n'importe quelle chaîne sauf la sous-chaîne **Le résultat est VRAI** - correspond à l'expression régulière **Le résultat est FAUX** - ne correspond pas à l'expression régulière
Expression	Saisissez une sous-chaîne/expression régulière.
Délimiteur	Une virgule (,), un point (.) ou une barre oblique (/) pour séparer les chaînes de texte dans une expression régulière. Ce paramètre n'est actif que lorsque le type d'expression "*Toute chaîne de caractères incluse*" est sélectionné.
Sensible à la casse	Une case à cocher pour spécifier si une expression régulière est sensible à la casse des lettres.

Une barre oblique (/) dans l'expression est traitée littéralement, plutôt qu'un délimiteur. De cette façon, il est possible d'enregistrer des expressions contenant une barre oblique, sans erreur.

Attention:
Un nom d'expression régulière personnalisé dans Zabbix peut contenir des virgules, des espaces, etc. Dans les cas où cela peut conduire à une mauvaise interprétation lors du référencement (par exemple, une virgule dans le paramètre d'une clé d'élément), la référence entière peut être mise entre guillemets comme ceci : *"@My custom regexp for purpose1, purpose2"*.
Les noms d'expressions régulières ne doivent pas être entre guillemets dans d'autres emplacements (par exemple, dans les propriétés de règle LLD).

Dans l'onglet **Test**, l'expression régulière et ses sous-expressions peuvent être testées en fournissant une chaîne de test.



Les résultats affichent l'état de chaque sous-expression et l'état total de l'expression personnalisée.

L'état total de l'expression personnalisée est défini comme *Résultat combiné*. Si plusieurs sous-expressions sont définies, Zabbix utilise l'opérateur logique ET pour calculer le *résultat combiné*. Cela signifie que si au moins un résultat est Faux, *Résultat combiné* a également le statut Faux.

Expressions régulières globales par défaut Zabbix est livré avec plusieurs expressions régulières globales dans son jeu de données par défaut.

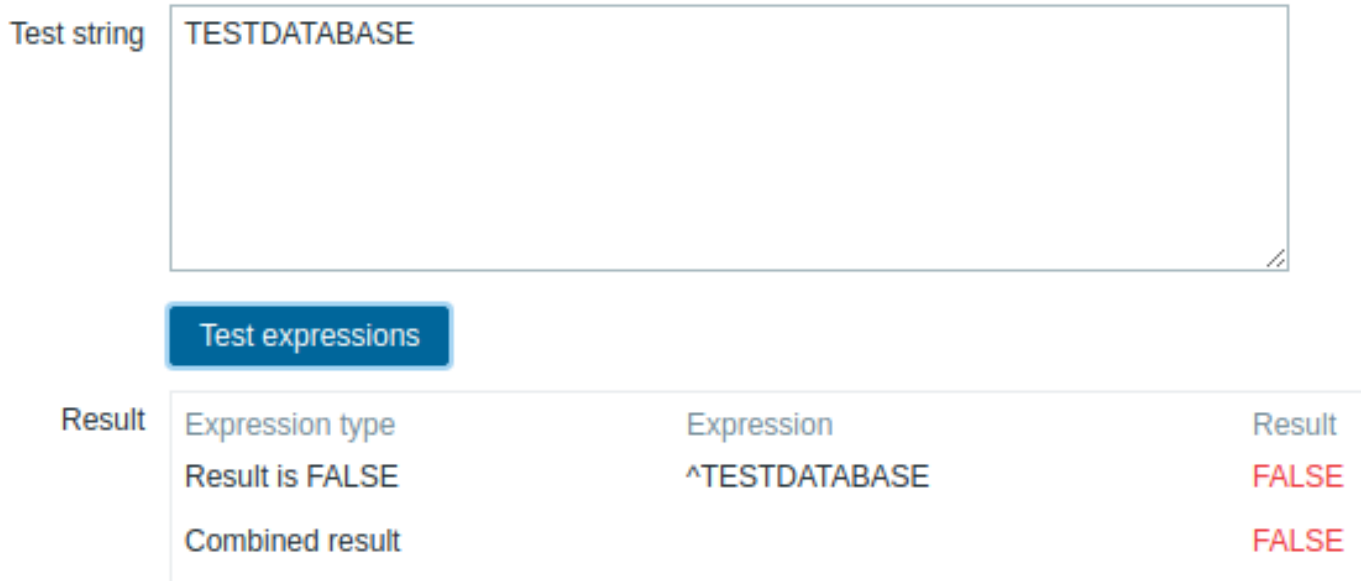
Nom	Expression	Correspondances
Systèmes de fichiers pour la découverte	`^(btrfs|ext2|ext3|ext4|jfs|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|refs|apfs|ntfs|fat32|zfs)$`	"btrfs" ou "ext2" ou "ext3" ou "ext4" ou "jfs" ou "reiser" ou "xfs" ou "ffs" ou "ufs" ou "jfs" ou "jfs2" ou "vxfs" ou "hfs" ou "refs" ou "apfs" ou "ntfs" ou "fat32" ou "zfs"
Interfaces réseau pour la découverte	`^Software Loopback Interface`	Chaînes commençant par "Software Loopback Interface".
	`^lo$`	"lo"
	`^(In)?[Ll]oop[Bb]ack[0-9._]*$`	Chaînes commençant éventuellement par "In", puis contenant "L" ou "l", puis "oop", puis "B" ou "b", puis "ack", éventuellement suivi de n'importe quel nombre de chiffres, de points ou de traits de soulignement.
	`^NULL[0-9.]*$`	Chaînes commençant par "NULL", éventuellement suivies de n'importe quel nombre de chiffres ou de points.
	`^[Ll]o[0-9.]*$`	Chaînes commençant par "Lo" ou "lo", éventuellement suivies de n'importe quel nombre de chiffres ou de points.
	`^[Ss]ystem$`	"System" ou "system"
	`^Nu[0-9.]*$`	Chaînes commençant par "Nu", éventuellement suivies de n'importe quel nombre de chiffres ou de points.

Nom	Expression	Correspondances
Périphériques de stockage pour la découverte SNMP	`^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$`	”Physical memory” ou ”Virtual memory” ou ”Memory buffers” ou ”Cached memory” ou ”Swap space”
Noms des services Windows pour la découverte	`^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.30319_32)$`	”MMCSS”, ”gupdate”, ”SysmonLog” ou des chaînes comme ”clr_optimization_v2.0.50727_32” et ”clr_optimization_v4.0.30319_32”, où à la place des points, vous pouvez mettre n’importe quel caractère sauf un saut de ligne.
États de démarrage des services Windows pour la découverte	`^(automatic|automatic delayed)$`	”automatic” ou ”automatic delayed”

Exemples Exemple 1

Utilisation de 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$
```

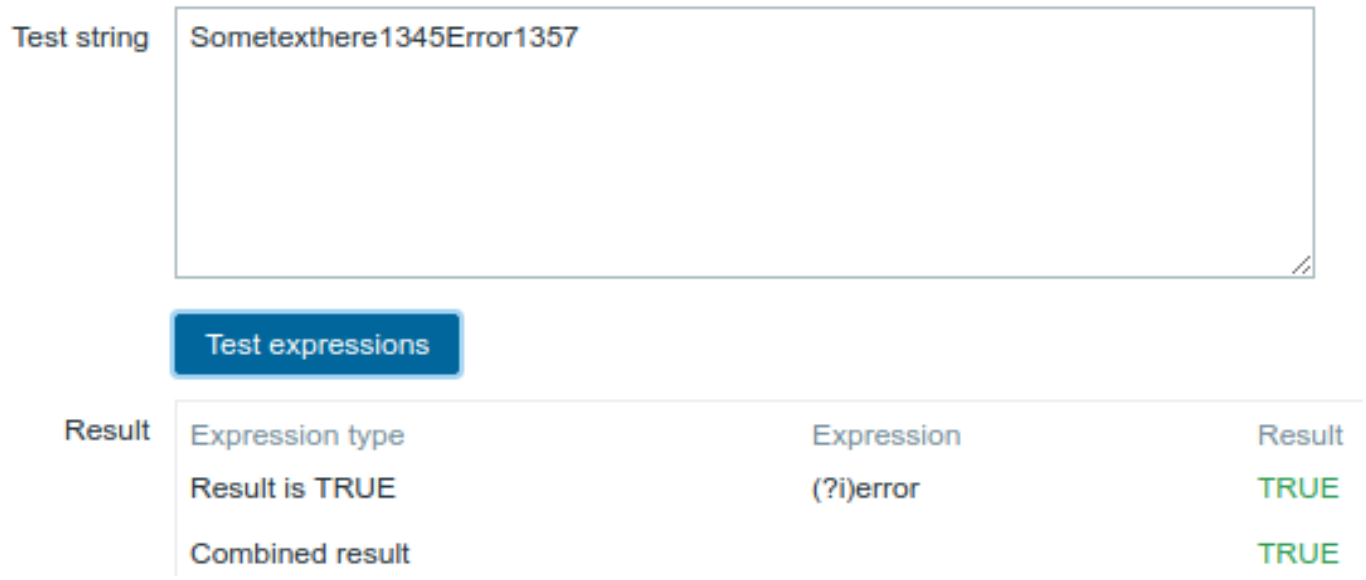


Type d’expression choisi : ”Le résultat est FAUX”. Ne correspond pas au nom, contenant la chaîne ”*TESTDATABASE*”.

Exemple avec un modificateur de regex en ligne

Utilisation de l’expression régulière suivante incluant un modificateur en ligne (?i) pour faire correspondre les caractères ”error”:

```
(?i)error
```



Type d'expression choisi : "Le résultat est VRAI". Les caractères "error" sont mis en correspondance.

Un autre exemple avec un modificateur de regex en ligne

Utilisation de l'expression régulière suivante, y compris plusieurs modificateurs en ligne pour faire correspondre les caractères après une ligne spécifique :

```
(?<=match (?i)everything(?-i) after this line\n)(?sx).*# nous ajoutons le modificateur s pour autoriser .
```



Type d'expression choisi : "Le résultat est VRAI". Les caractères après une ligne spécifique sont mis en correspondance.

Attention:
Le modificateur **g** ne peut pas être spécifié en ligne. La liste des modificateurs disponibles se trouve dans page de manuel pcresyntax. Pour plus d'informations sur la syntaxe PCRE, veuillez consulter la documentation HTML PCRE.

Prise en charge des expressions régulières selon l'emplacement

Emplacement	Expression régulière	Expression régulière globale	Correspondance multiligne	Commentaires
Éléments d'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

Emplacement	Expression régulière	Expression régulière globale	Correspondance multiligne	Commentaires
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`
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[]			Oui	paramètre `regexp`
vfs.file.regmatch[]				
web.page.regexp[]				
Traps SNMP				
snmptrap[]	Oui	Oui	Non	paramètre `regexp`
Prétraitement de la valeur d'élément	Oui	Non	Non	paramètre `pattern`
Fonctions pour les déclencheurs/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 d'*Opération*
Conditions d'action	Oui	Non	Non	Dans les options *correspond à*, *ne correspond pas à* pour les conditions d'auto-enregistrement *Nom d'hôte* et *Métadonnées de l'hôte*
Scripts	Oui	Oui	Non	champ *Règle de validation d'entrée*
Supervision web	Oui	Non	Oui	*Variables* avec un préfixe **regex:** champ *Chaîne requise*
Contexte de macro utilisateur	Oui	Non	Non	Dans le contexte de macro avec un préfixe **regex:**

Emplacement	Expression régulière	Expression régulière globale	Correspondance multiligne	Commentaires
Fonctions de macro regsub() iregsub()	Oui	Non	Non	paramètre `pattern`
Indicateurs de lien dans les cartes	Oui	Non	Non	champ *Motif* (pour les éléments texte)
Mappage d'icônes	Oui	Oui	Non	champ *Expression*
Mappage de valeurs	Oui	Non	Non	champ *Valeur* si le type de mappage est `regexp`

11 Acquittement de problème

Aperçu Les événements de problème dans Zabbix peuvent être acquittés par les utilisateurs.

Si un utilisateur reçoit une notification concernant un événement de problème, il peut accéder au frontend Zabbix, ouvrir la fenêtre contextuelle de mise à jour du problème correspondant à l'aide de l'une des méthodes indiquées ci-dessous, puis acquitter le problème. Lors de l'acquittement, il peut saisir un commentaire, indiquant qu'il est en train de le traiter ou toute autre information qu'il souhaite ajouter à son sujet.

Ainsi, si un autre utilisateur du système remarque le même problème, il voit immédiatement s'il a été acquitté ainsi que les commentaires ajoutés jusqu'à présent.

De cette manière, le processus de résolution des problèmes impliquant plusieurs utilisateurs du système peut se dérouler de façon coordonnée.

Le statut d'acquittement 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'est pas acquitté pendant un certain temps.

Pour acquitter des événements et les commenter, un utilisateur doit disposer au minimum de droits de lecture sur les déclencheurs correspondants. Pour modifier la sévérité d'un problème ou fermer un problème, un utilisateur doit disposer de droits de lecture-écriture sur les déclencheurs correspondants.

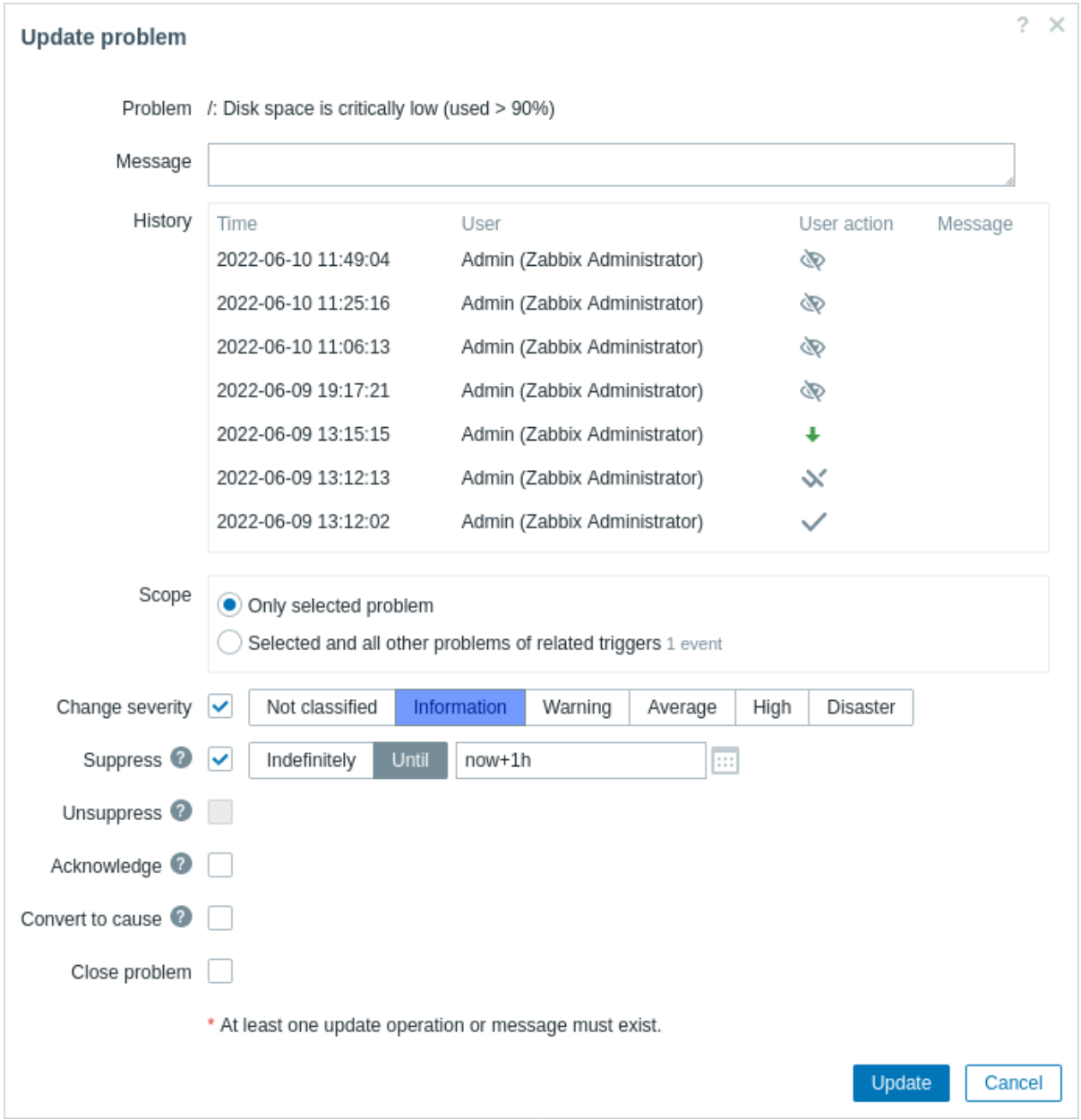
Il existe **plusieurs** façons d'accéder à la fenêtre contextuelle de mise à jour du problème, qui permet d'acquitter un problème.

- 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.

Mise à jour des problèmes La fenêtre contextuelle de mise à jour des problèmes permet de :

- commenter le problème
- afficher les commentaires et les actions effectuées jusqu'à présent
- modifier la sévérité du problème
- supprimer/annuler la suppression du problème
- acquitter/annuler l'acquittement du problème
- transformer un problème symptôme en problème cause
- fermer manuellement le problème



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

Parameter	Description
Problem	Si un seul problème est sélectionné, le nom du problème est affiché. Si plusieurs problèmes sont sélectionnés, *N problems selected* s'affiche.
Message	Saisissez un texte pour commenter le problème (2048 caractères maximum).
History	Les activités et commentaires précédents sur le problème sont listés, avec l'heure et les informations sur l'utilisateur. Pour la signification des icônes utilisées pour indiquer les actions de l'utilisateur, consultez la page détail de l'événement. Notez que l'historique est affiché uniquement si un seul problème est sélectionné pour la mise à jour.
Scope	Définissez la portée d'actions telles que la modification de la sévérité, l'acquittement ou la fermeture manuelle des problèmes : **Only selected problem** - n'affectera que cet événement **Selected and all other problems of related triggers** - dans le cas de l'acquittement/de la fermeture du problème, affectera cet événement et tous les autres problèmes qui ne sont pas encore acquittés/fermés. Si la portée contient des problèmes déjà acquittés ou fermés, ces problèmes ne seront pas acquittés/fermés à nouveau. En revanche, le nombre d'opérations de message et de modification de sévérité n'est pas limité.

Parameter	Description
Change severity	Cochez la case et cliquez sur le bouton de sévérité pour mettre à jour la sévérité du problème. La case à cocher pour modifier la sévérité est disponible si des droits de lecture-écriture existent pour au moins un des problèmes sélectionnés. Seuls les problèmes pour lesquels les droits de lecture-écriture sont disponibles seront mis à jour en cliquant sur *Update*. Si aucun des déclencheurs sélectionnés ne dispose de droits de lecture-écriture, la case à cocher est désactivée.
Suppress	Cochez la case pour supprimer le problème : **Indefinitely** - supprimer indéfiniment **Until** - supprimer jusqu'à une heure donnée. Les formats d'heure absolus et relatifs sont pris en charge, par exemple : `now+1d` - pendant un jour à partir de maintenant (par défaut) `now/w` - jusqu'à la fin de la semaine en cours `2022-05-28 12:00:00` - jusqu'à la 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 "Suppress problems". Voir aussi : Suppression des problèmes
Unsuppress	Cochez la case pour annuler la suppression du problème. Cette case à cocher est active uniquement si au moins un des problèmes sélectionnés est supprimé. La disponibilité de cette option dépend des paramètres du rôle utilisateur "Suppress problems".
Acknowledge	Cochez la case pour acquitter le problème. Cette case à cocher est disponible s'il y a au moins un problème non acquitté parmi ceux 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).
Unacknowledge	Cochez la case pour annuler l'acquittement du problème. Cette case à cocher est disponible s'il y a au moins un problème acquitté parmi ceux sélectionnés.
Convert to cause	Cochez la case pour transformer le ou les problèmes symptômes en problème(s) cause(s).
Close problem	Cochez la case pour fermer manuellement le ou les problèmes sélectionnés. La case à cocher pour fermer un problème est disponible si l'option *Allow manual close* est cochée dans la configuration du déclencheur pour au moins un des problèmes sélectionnés. Seuls les problèmes dont la fermeture est autorisée seront fermés en cliquant sur *Update*. Si aucun problème ne peut être fermé manuellement, la case à cocher est désactivée. Les problèmes déjà fermés ne seront pas fermés à nouveau.

Affichage En fonction des informations d'acquittement, il est possible de configurer la manière dont le nombre de problèmes est affiché dans le tableau de bord ou sur les cartes. Pour ce faire, vous devez effectuer des sélections dans l'option *Affichage des problèmes*, disponible à la fois dans la configuration des cartes 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.

En fonction des informations de mise à jour du problème (acquittement, etc.), il est possible de configurer des opérations de mise à jour : envoyer un message ou exécuter des commandes à distance.

###1 Suppression des problèmes {#manual-acknowledgment-suppression}

Aperçu

La suppression des problèmes offre un moyen de masquer temporairement un problème qui peut être résolu ultérieurement. Ceci est utile pour nettoyer la liste des problèmes afin de donner la plus haute priorité aux problèmes les plus urgents. Par exemple, il arrive parfois qu'un problème survienne le week-end et qu'il ne soit pas suffisamment urgent pour être traité immédiatement. Il peut donc être « mis en attente » jusqu'au lundi matin.

La suppression des problèmes permet de masquer un *unique* problème, contrairement à la suppression des problèmes via la maintenance de l'hôte lorsque tous les problèmes de l'hôte de maintenance sont masqués.

Les opérations pour les actions de déclenchement seront suspendues pour les problèmes supprimés de la même manière qu'avec maintenance de l'hôte.

Configuration

Un problème peut être supprimé via **Mise à jour d'un problème** fenêtre, où la suppression est l'une des options de mise à jour du problème avec commenter, modifier la gravité, reconnaître, etc.

Un problème peut également être non supprimé via le même problème fenêtre de mise à jour.

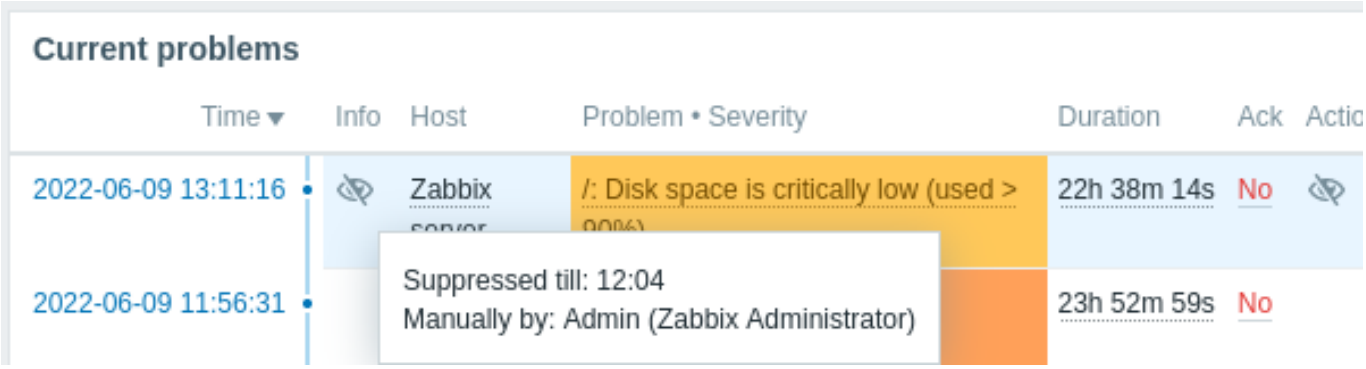
Affichage

Une fois supprimé, le problème est marqué par une icône de suppression clignotante dans la colonne *Info*, avant d'être masqué.

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 longtemps, cela peut indiquer un problème du 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 l'annulation de la suppression. Pendant la courte période qui suit l'envoi de la tâche, tant que le serveur ne l'a pas encore terminée, l'icône d'annulation de suppression clignote.

Un problème supprimé peut être soit masqué, soit affiché, selon les paramètres du filtre de problèmes/widget.

Lorsqu'il est affiché dans la liste des problèmes, un problème supprimé est marqué par l'icône de suppression et les détails de la suppression s'affichent au survol de la souris :



Les détails de la suppression sont également affichés dans une fenêtre contextuelle lorsque le pointeur de la souris est placé sur l'icône de suppression dans la colonne *Actions*.

12 Export/import de configuration

Aperçu La fonctionnalité d'exportation/importation de Zabbix permet d'échanger diverses entités de configuration entre un système Zabbix et un autre.

Cas d'utilisation typiques pour cette fonctionnalité :

- partager des modèles ou des cartes réseau - les utilisateurs de Zabbix peuvent partager leurs paramètres de configuration
- téléchargez un modèle sur les modèles de la communauté Zabbix. Ensuite, d'autres peuvent télécharger le modèle et importer le fichier dans Zabbix.
- intégration avec des outils tiers - les formats universels YAML, XML et JSON permettent l'intégration et l'importation/exportation de données avec des outils et des applications tiers

Que peut-on exporter/importer ?

Les objets pouvant être exportés/importés sont :

- Les groupes d'hôtes (*via l'API Zabbix uniquement*)
- Les modèles de groupe (*via l'API Zabbix uniquement*)
- Les modèles
- Les hôtes
- Les cartes réseau
- Les types de media
- Les images

Format d'exportation

Les données peuvent être exportées à l'aide de l'interface Web Zabbix ou de l'API Zabbix. Les formats d'exportation pris en charge sont YAML, XML and JSON.

Détails sur l'exportation

- Tous les éléments pris en charge sont exportés dans un seul fichier.
- Les entités d'hôte et de modèle (éléments, déclencheurs, graphiques, règles de découverte) héritées de modèles liés ne sont pas exportées. Toute modification apportée à ces entités au niveau de l'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'exportation ; lors de l'importation, toutes les entités des modèles liés sont recréées comme dans le modèle lié d'origine.
- Les entités créées par la découverte de bas niveau ainsi que 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/modèle exporté contient des entités prenant en charge les délais d'expiration, les valeurs de délai d'expiration seront exportées si ces entités ont leurs propres délais d'expiration configurés.

Détails sur l'importation

- L'importation s'arrête à la première erreur.
- Lors de la mise à jour d'images existantes pendant l'importation d'images, le champ "imagetype" est ignoré, c'est-à-dire qu'il est impossible de modifier le type d'image via l'importation.
- Lors de l'importation d'hôtes/modèles avec l'option "Delete missing", les macros d'hôte/modèle qui ne sont pas présentes dans le fichier d'importation seront supprimées de l'hôte/modèle après l'importation.
- 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'expiration configurés, ils seront appliqués ; sinon, les délais d'expiration du proxy/globaux seront appliqués.
- L'importation prend en charge YAML, XML et JSON ; le fichier d'importation doit avoir une extension 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'importation prend en charge uniquement les fichiers de configuration encodés en UTF-8 (avec ou sans BOM) ; les autres encodages (UTF16LE, UTF16BE, UTF32LE, UTF32BE, etc.) entraîneront une erreur de conversion lors de l'importation.

Format de base YAML Le format d'export YAML contient les nœuds suivants :

- Nœud racine pour l'export YAML de Zabbix
- Version de l'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 des documents XML
- Balise racine pour l'export XML de Zabbix
- Version de l'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 de l'export

```
{
    "zabbix_export": {
        "version": "8.0"
    }
}
```

Les autres objets dépendent des objets exportés.

1 Groupes de modèles

Aperçu

Dans l'interface web, les groupes de modèles ne peuvent être exportés qu'avec l'exportation de modèles. Lorsqu'un modèle est exporté, tous les groupes auxquels il appartient sont automatiquement exportés avec lui.

L'API permet d'exporter les groupes de modèles indépendamment des modèles.

Format d'exportation

```
template_groups:
  - uuid: 36bff6c29af64692839d077febfc7079
    name: 'Périphériques réseau'
```

Éléments exportés

Élément	Type	Description
uuid	string	Identifiant unique de ce groupe de modèles.
name	string	Nom du groupe.

2 Groupes d'hôtes

Aperçu

Dans l'interface web, les groupes d'hôtes peuvent être exportés uniquement avec l'export d'hôtes. 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'exportation

```
host_groups:
  - uuid: 6f6799aa69e844b4b3918f779f2abf08
    name: 'Serveurs Zabbix'
```

Éléments exportés

Élément	Type	Description
uuid	string	Identifiant unique de ce groupe d'hôtes.
name	string	Nom du groupe.

3 Modèles

Aperçu

Les modèles sont exportés avec de nombreux objets associés et relations entre objets.

L'exportation d'un modèle contient :

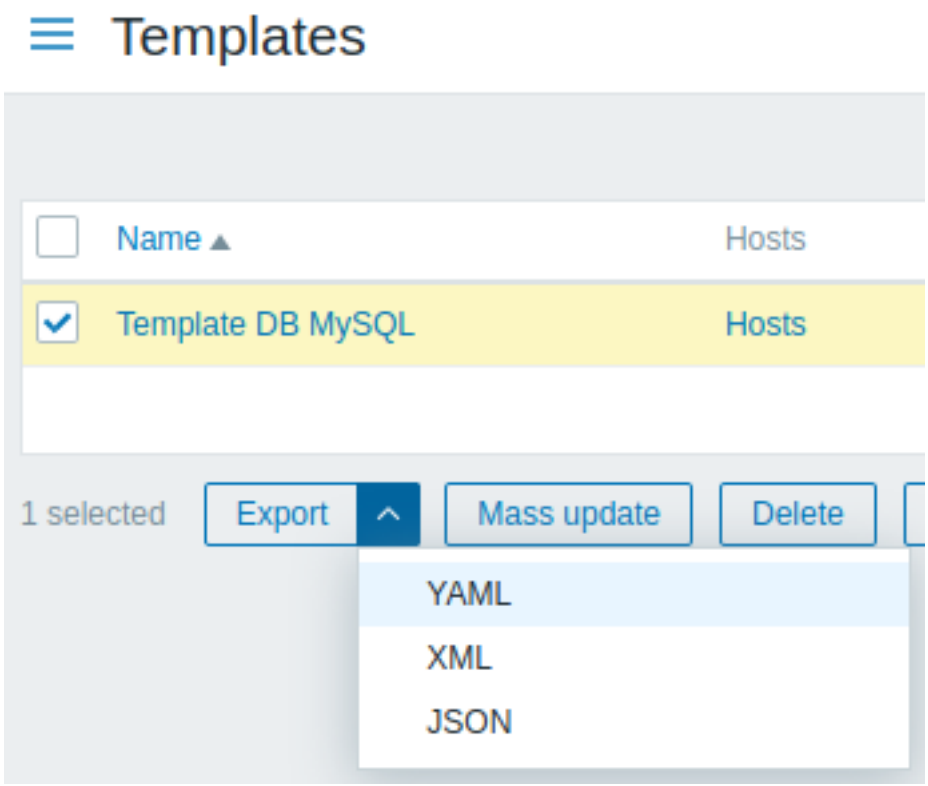
- Groupes de modèles liés
- Groupes d'hôtes liés (s'ils sont utilisés dans la configuration des prototypes d'hôte)
- Données du modèle
- Liaison avec d'autres modèles
- Liaison avec des groupes de modèles
- Éléments directement liés
- Déclencheurs directement liés
- Graphiques directement liés
- Tableaux de bord directement liés
- Règles de découverte directement liées avec tous les prototypes

- Scénarios web directement liés
- Mappages 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.



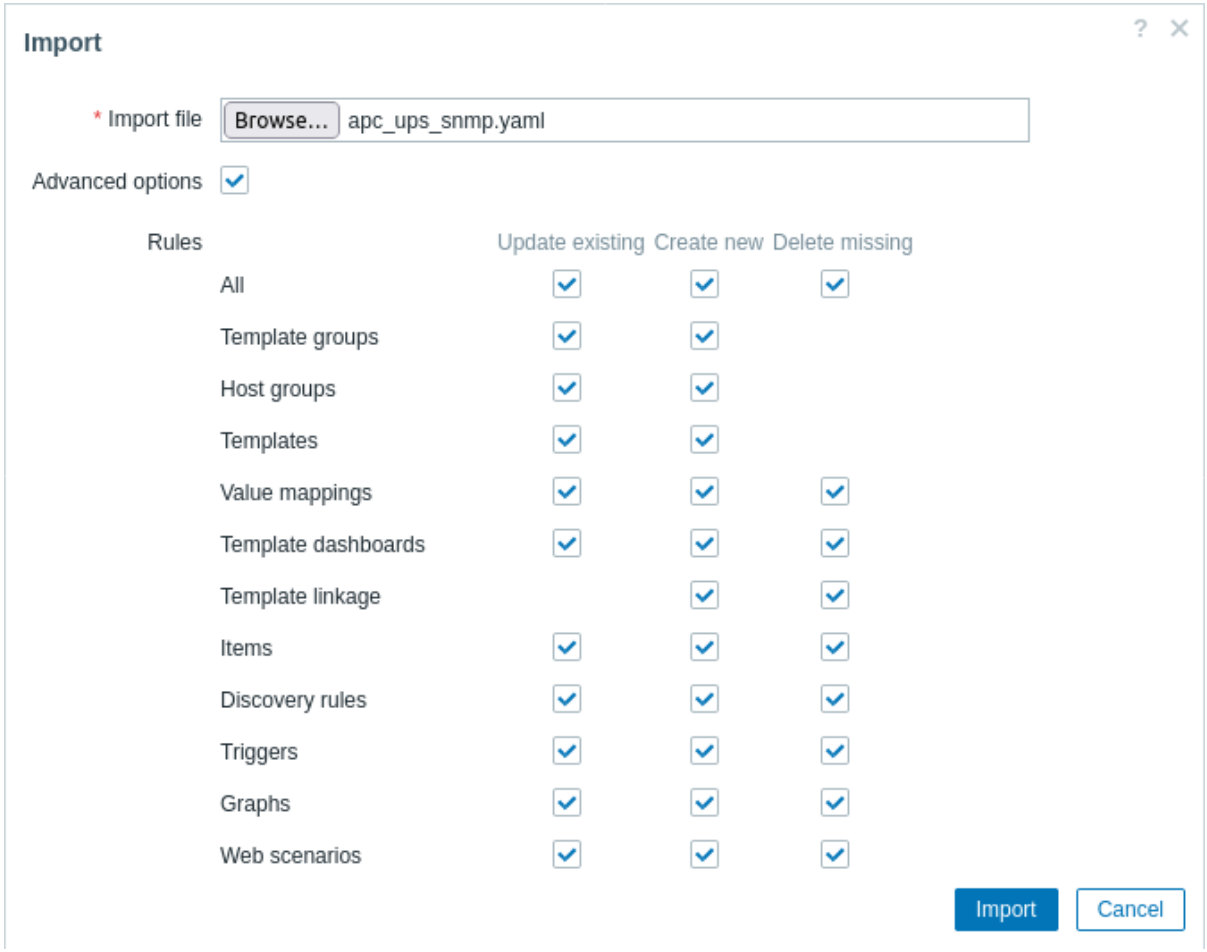
Selon le format sélectionné, les modèles sont exportés vers un fichier local avec un nom par défaut :

- `zabbix_export_templates.yaml` - en export YAML (option par défaut pour l'exportation) ;
- `zabbix_export_templates.xml` - en export XML ;
- `zabbix_export_templates.json` - en export JSON.

Importation

Pour importer des modèles, procédez comme suit :

1. Allez dans *Collecte de données* → *Modèles*.
2. Cliquez sur *Importer* dans le coin supérieur droit.
3. Sélectionnez le fichier d'importation.
4. Cliquez sur *Importer* dans le coin inférieur droit du formulaire de configuration.



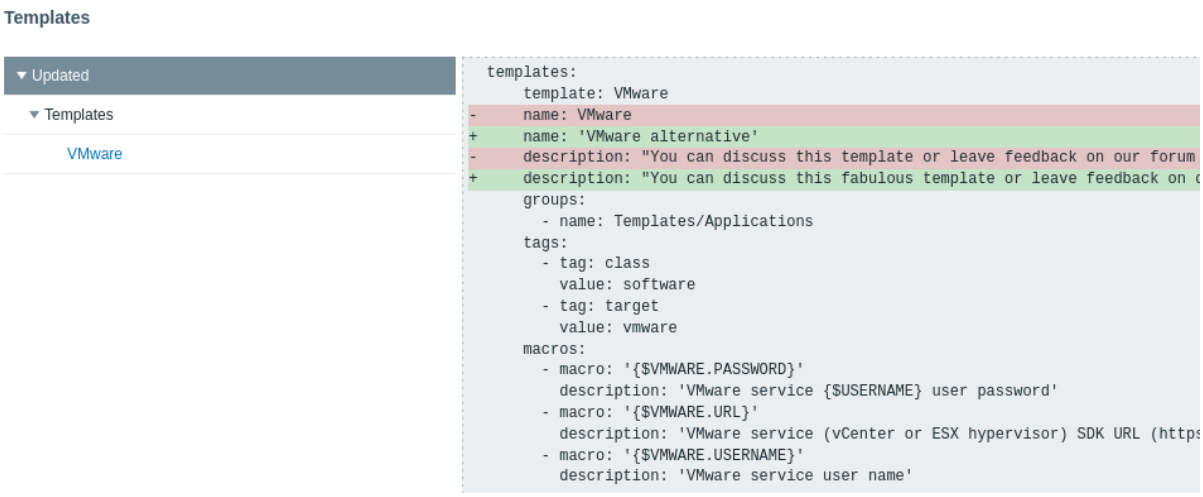
Si vous cochez la case *Options avancées*, une liste détaillée de tous les éléments importables s'affichera ; cochez ou décochez chaque règle d'importation selon vos besoins.

Si vous cliquez sur la case à cocher de la ligne *Tous*, tous les éléments en dessous seront cochés/décochés.

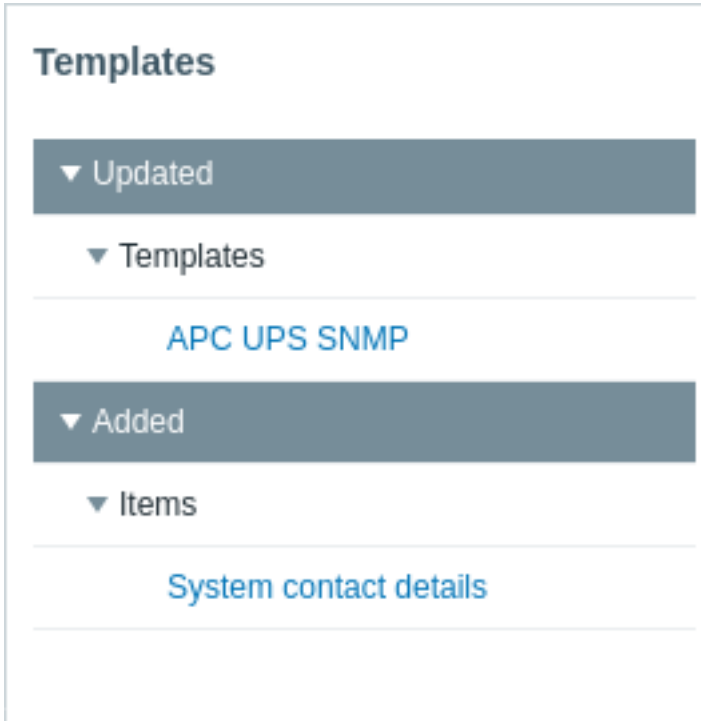
Règles d'importation :

Règle	Description
Mettre à jour l'existant	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 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 sont pas présents dans le fichier d'importation seront supprimés. Sinon, ils ne seront pas supprimés. Si *Supprimer les éléments manquants* est coché pour *Liaison de modèles*, la liaison actuelle des modèles qui n'est pas présente dans le fichier d'importation sera supprimée. Les entités (éléments, déclencheurs, graphiques, etc.) héritées des modèles dissociés ne seront pas supprimées (sauf si l'option *Supprimer les éléments manquants* est également sélectionnée pour chaque entité).

Sur l'écran suivant, vous pourrez 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 du modèle seront mis en évidence en rouge ; les éléments qui n'ont pas changé seront affichés sur un fond gris.



Le menu de gauche peut être utilisé pour parcourir la liste des modifications. La section *Mis à jour* met en évidence toutes les modifications apportées aux éléments existants du modèle. La section *Ajouté* 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.



Passez en revue les modifications du modèle, puis cliquez sur *Importer* pour lancer l'importation du modèle. Un message indiquant le succès ou l'échec de l'importation s'affichera dans l'interface web.

Format d'export Lorsqu'un modèle est exporté, Zabbix génère un format structuré YAML, JSON ou XML. L'export inclut des éléments du modèle, tels que les métadonnées du modèle, les éléments de données, les macros, les déclencheurs, les tableaux de bord, et plus encore.

Chaque élément remplit un objectif spécifique et peut contenir des éléments imbriqués.

Les sections suivantes décrivent chaque élément du format d'export. Les exemples utilisent le modèle Linux by Zabbix agent.

Une ellipse (`...`) indique des éléments omis par souci de concision. La note (`see table below`) 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: (voir le tableau ci-dessous)
```

```
  triggers: (voir le tableau ci-dessous)
  graphs: (voir le tableau ci-dessous)
```

Métadonnées du modèle

Élément	Type	Description
version	string	(obligatoire) Version de l'export.
template_groups		(obligatoire) Élément racine pour les groupes de modèles.
uuid	string	(obligatoire) Identifiant unique de ce groupe de modèles.
name	string	(obligatoire) Nom du groupe de modèles.
host_groups		Élément racine pour les groupes d'hôtes utilisés par les prototypes d'hôte.
uuid	string	(obligatoire) Identifiant unique de ce groupe d'hôtes.
name	string	(obligatoire) Nom du groupe d'hôtes.
templates		Élément racine pour les modèles.
triggers		Élément racine pour les éléments de déclencheur de modèle, identiques à ceux des déclencheurs d'élément de modèle.
graphs		Élément racine pour les éléments de graphique de modèle, identiques à ceux des graphiques d'hôte.

```
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: |
        ## Aperçu

        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: (voir le tableau ci-dessous)
      discovery_rules: (voir le tableau ci-dessous)
      tags:
        - tag: class
          value: os
        - tag: target
          value: linux
      macros:
        - macro: '{$AGENT.TIMEOUT}'
          value: 3m
          description: 'Délai après lequel l''agent est considéré comme indisponible. Fonctionne uniquemen
          config: (voir le tableau ci-dessous)
        - macro: '{$CPU.UTIL.CRIT}'
          value: '90'
          description: 'Seuil critique d''utilisation du processeur exprimé en %.'
          config: (voir le tableau ci-dessous)
        (...)
      dashboards: (voir le tableau ci-dessous)
      valuemaps: (voir le tableau ci-dessous)
    (...)
```

Modèles

Element	Type	Description
uuid	string	(obligatoire) Identifiant unique de ce modèle.
template	string	(obligatoire) 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 : Template object (`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	(obligatoire) Nom du fournisseur du modèle.
version	string	(obligatoire) Version du modèle. 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.
templates		Élément racine pour les modèles liés.
name	string	(obligatoire) Nom du modèle.
groups		Élément racine pour les groupes de modèles.
name	string	(obligatoire) 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	(obligatoire) Nom de la balise.
value	string	Valeur de la balise.
macros		Élément racine pour les macros utilisateur du modèle.
macro	string	(obligatoire) 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 : User macro object (`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 correspondances de valeurs du modèle.

Éléments de 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 en combinaison a
          valuemap:
            name: 'Zabbix agent ping status'
          tags:
            - tag: component
              value: system
        (...)
        - uuid: 58818005e76d46dda14d6592f601ab00
          name: 'Number of installed packages'
          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: 'Memory utilization'
      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).
snmp_oid	string	(obligatoire pour les éléments SNMP_AGENT) ID d'objet SNMP.
key	string	(obligatoire) Clé de l'élément.
delay	string	Intervalle de mise à jour de l'élément. Par défaut : 1m. La valeur sera toujours 0 pour les éléments TRAP.
history	string	Période (à l'aide d'un suffixe temporel, d'une macro utilisateur ou d'une macro LLD) pendant laquelle les données d'historique doivent être conservées. Par défaut : 31d.
trends	string	Période (à l'aide d'un suffixe temporel, d'une macro utilisateur ou d'une macro LLD) pendant laquelle les données de tendances doivent être conservées. Par défaut : 365d.
status	string	Statut 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 d'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).
username	string	(obligatoire pour les éléments SSH et TELNET) Nom d'utilisateur pour l'authentification. Pris en charge pour les éléments SIMPLE, ODBC, JMX et HTTP_AGENT. Lorsqu'il est utilisé pour des éléments JMX, `password` (voir ci-dessous) doit également être spécifié, ou les deux éléments doivent être laissés vides.

Élément	Type	Description
password	string	(obligatoire pour les éléments SSH et TELNET) Mot de passe pour l'authentification. Pris en charge pour les éléments SIMPLE, ODBC, JMX et HTTP_AGENT. Lorsqu'il est utilisé pour des éléments JMX, `username` (voir ci-dessus) doit également être spécifié, ou les deux éléments doivent être laissés 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 renseigné par l'élément. Valeurs possibles :[1] NONE (0), ALIAS (4), etc. (voir Host inventory pour les champs pris en charge).
valuemap		Élément racine pour les correspondances de valeurs de l'élément.
name	string	(obligatoire) Nom de la correspondance 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 pour le prétraitement des valeurs de l'élément.
step		Élément racine pour les étapes de prétraitement des valeurs d'élément 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 pour l'élément maître d'un 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 Timeouts.
url	string	(obligatoire pour les éléments HTTP_AGENT) Chaîne URL.
query_fields		Élément racine pour les 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 pour les 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 des codes d'état HTTP requis, séparées par des virgules. Pris en charge pour les éléments HTTP_AGENT.
follow_redirects	string	Suivre les redirections de 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 POST. 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 du proxy HTTP(S). Pris en charge pour les éléments HTTP_AGENT.
headers		Élément racine pour les 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.
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).

Élément	Type	Description
allow_traps	string	Autoriser le renseignement de la valeur de manière similaire à 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 valider l'authenticité du certificat de l'hôte. Pris en charge pour les éléments HTTP_AGENT. Valeurs possibles :[1] NO (0, par défaut), YES (1).
verify_host	string	Indique s'il faut valider que le nom d'hôte de la connexion correspond à celui figurant dans le certificat de l'hôte. Pris en charge pour les éléments HTTP_AGENT. Valeurs possibles :[1] NO (0, par défaut), YES (1).
tags		Élément racine pour les tags de l'élément.
tag	string	(obligatoire) Nom du tag.
value	string	Valeur du tag.
triggers		Élément racine pour les déclencheurs d'élément de modèle.

Note:
Voir aussi : Item object (reportez-vous à la propriété pertinente portant le même nom).

Étapes de prétraitement des valeurs d'élément de modèle

```
zabbix_export:
  (...)
  templates:
    (...)
      items:
        (...)
        - uuid: 58818005e76d46dda14d6592f601ab00
          (...)
          preprocessing:
            - type: JSONPATH
              parameters:
                - $.length()
            - type: DISCARD_UNCHANGED_HEARTBEAT
              parameters:
                - 12h
          (...)
```

Élément	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 des 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.

Élément	Type	Description
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 : Item preprocessing object (reportez-vous 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)>{$MEMORY.UTIL.MAX}'
              name: 'Linux : utilisation élevée de la mémoire'
              event_name: 'Linux : utilisation élevée de la mémoire (>{$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 pour la 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 (pas de corrélation d'événements ou corrélation d'événements 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.

Element	Type	Description
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, 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 (événement de problème unique ou événements de problèmes multiples). 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 de dépendance.
expression	string	(obligatoire) Expression du déclencheur de dépendance.
recovery_expression	string	Expression de récupération du déclencheur de dépendance.
tags		Élément racine pour les tags du déclencheur.
tag	string	(obligatoire) Nom du tag.
value	string	Valeur du tag.

Note:
Voir aussi : Trigger object (reportez-vous à la propriété correspondante portant le même nom).

Règles de découverte de bas niveau de template

```
zabbix_export:
  (...)
  templates:
    (...)
      discovery_rules:
        - uuid: acfdea9c46ef48c68e6636f43b8f96a2
          name: 'Découverte des interfaces réseau'
          key: net.if.discovery
          delay: 1h
          filter: (voir le tableau ci-dessous)
          description: 'La découverte des interfaces réseau.'
          item_prototypes: (voir le tableau ci-dessous)
          trigger_prototypes: (voir le tableau ci-dessous)
          graph_prototypes: (voir le tableau ci-dessous)
        (...)
```

Attention:
La plupart des éléments des règles de découverte de bas niveau de template sont identiques à ceux des éléments de template. Le tableau ci-dessous décrit les éléments qui diffèrent des éléments de template.

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 de règle de découverte de bas niveau de template.
lifetime	string	Période (en utilisant des secondes, un suffixe temporel ou une macro utilisateur), après laquelle les ressources qui ne sont plus découvertes seront supprimées. Par défaut : 7d.
lifetime_type	string	Scénario de suppression des ressources LLD perdues. Valeurs possibles : DELETE_NEVER, DELETE_IMMEDIATELY, DELETE_AFTER.

Element	Type	Description
enabled_lifetime	string	Période (en utilisant des secondes, un suffixe temporel ou une macro utilisateur), après laquelle les ressources qui ne sont plus découvertes 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 de prototype d'élément de template, qui sont identiques à ceux des éléments de template.
trigger_prototypes		Élément racine pour les éléments de prototype de déclencheur de template, qui sont identiques à ceux des déclencheurs d'élément de template.
graph_prototypes		Élément racine pour les éléments de prototype de graphique de template, qui sont identiques à ceux des graphiques d'hôte.
host_prototypes		Élément racine pour les éléments de prototype d'hôte de template, 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/du 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 règle de découverte de bas niveau.
lld_macro	string	(obligatoire) Nom de la macro de règle de découverte de bas niveau.
path	string	(obligatoire) Sélecteur de valeur, qui sera attribuée à la macro correspondante.
preprocessing		Élément racine pour le prétraitement des valeurs de la règle de découverte de bas niveau.
step		Élément racine pour les éléments d'étape de prétraitement des valeurs de la règle de découverte de bas niveau, qui sont identiques à ceux des étapes de prétraitement des valeurs d'élément de template, sauf avec moins de valeurs possibles. Voir aussi : objet de prétraitement de règle LLD.
overrides		Élément racine pour les règles de substitution de la règle de découverte de bas niveau.
name	string	(obligatoire) Nom unique de la substitution.
step	string	(obligatoire) Numéro d'ordre unique de la substitution.
stop	string	Arrêter le traitement des substitutions suivantes en cas de correspondance.
filter		Élément racine pour les éléments de filtre de règle de substitution de découverte de bas niveau de template, qui sont identiques à ceux des filtres de règle de découverte de bas niveau de template.
operations		Élément racine pour les opérations de substitution de règle de découverte de bas niveau de template.

Note:
Voir aussi : objet de règle LLD (reportez-vous à la propriété pertinente portant le même nom).

Filtres des règles de découverte de bas niveau de template

```
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
          (...)
```

Élément	Type	Description
evaltype	string	Remplacer 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 des conditions du filtre.
macro	string	(obligatoire) Nom de la macro de découverte de 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 du filtre, mais sera généré à nouveau lors de leur demande ultérieure.

Note:
Voir aussi : objet de filtre de règle LLD (reportez-vous à la propriété correspondante portant le même nom).

Opérations de substitution de règle de découverte de bas niveau de modèle

Élément	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).
operator	string	Opérateur de condition de substitution. 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 substitution.
status	string	Statut de l'objet lors de l'opération de substitution.
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 substitution.
history	string	Période de stockage de l'historique définie pour le prototype d'élément lors de l'opération de substitution.
trends	string	Période de stockage des tendances définie pour le prototype d'élément lors de l'opération de substitution.
severity	string	Sévérité du prototype de déclencheur définie lors de l'opération de substitution.
tags		Élément racine pour les tags définis pour l'objet lors de l'opération de substitution.
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 substitution.
name	string	(obligatoire) Nom du modèle.
inventory_mode	string	Mode d'inventaire du prototype d'hôte défini lors de l'opération de substitution.

Note:
Voir aussi : LLD rule override operation object (reportez-vous à la propriété correspondante portant le même nom).

Scénarios web de modèle

Élément	Type	Description
uuid	string	(obligatoire) Identifiant unique de ce scénario web.
name	string	(obligatoire) Nom du scénario web.
delay	string	Fréquence d'exécution du scénario web (en utilisant des secondes, un suffixe temporel ou une macro utilisateur). Par défaut : 1m.
attempts	integer	Nombre de tentatives d'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, indiqué sous la forme : `http://[username[:password]@]proxy.example.com[:port]`
variables		Élément racine des 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 des en-têtes HTTP à envoyer lors de l'exécution d'une requête. Les en-têtes doivent être listés en utilisant la même syntaxe que 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).
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 des étapes de scénario web de modèle.
tags		Élément racine des tags du scénario web.
tag	string	(obligatoire) Nom du tag.
value	string	Valeur du tag.

Note:
Voir aussi : Web scenario object (reportez-vous à la propriété correspondante portant le même nom).

Étapes du scénario web

Élément	Type	Description
name	string	(requis) Nom de l'étape du scénario web.
url	string	(requis) 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	(requis) Nom du paramètre de requête.
value	string	Valeur du paramètre de requête.

Élément	Type	Description
posts		Élément racine pour les variables HTTP POST (une chaîne de caractères (données POST brutes) ou un tableau de champs HTTP (données de champ de formulaire)).
name	string	(requis) Nom du champ POST.
value	string	(requis) 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:', sa valeur est alors extraite des données renvoyées par cette étape selon le motif d'expression régulière suivant le préfixe 'regex:'
name	string	(requis) Nom de la variable.
value	text	(requis) Valeur de la variable.
headers		Élément racine pour les en-têtes HTTP à envoyer lors de l'exécution d'une requête.
name	string	(requis) Nom de l'en-tête.
value	text	(requis) 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'expiration (en secondes, suffixe temporel 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 des codes d'état HTTP acceptés, séparés par des virgules (par exemple, 200-201,210-299 ; ignoré si vide).

Note:
Voir aussi : Web scenario step object (reportez-vous à la propriété correspondante portant le même nom).

Configuration des macros de modèle

```
zabbix_export:
  (...)
  templates:
    - 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: Seuils
            label: 'Seuil d'utilisation du processeur exprimé'
            description: 'Seuil critique d'utilisation du processeur exprimé en %. Dans la plage de 0 à 100
            regex: '^-?([0-9]+|([0-9]+)\.([0-9]+))$'
        (...)
        - macro: '{$IFCONTROL}'
          (...)
          config:
            type: CHECKBOX
            priority: '19'
            label: 'Contrôle des interfaces'
            description: 'Déclencher un trigger lorsque l'état opérationnel de l'interface passe à "Link do
            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	Marque la macro comme 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 utilisateur dans un champ de zone 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 un 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 un état décoché. Valeurs possibles :[1] 0 (false), 1 (true). Pris en charge pour les macros CHECKBOX.

Note:
Voir aussi : Macro configuration object (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: (voir le tableau ci-dessous)
      (...)
```

Élément	Type	Description
uuid	string	(obligatoire) Identifiant unique de ce tableau de bord.
name	string	(obligatoire) 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, par défaut).
pages		Élément racine pour les pages du tableau de bord du modèle.
name	string	Nom de la page.
display period	integer	Période d'affichage de la page.
sortorder	integer	Ordre de tri de la page.
widgets		Élément racine pour les widgets de tableau de bord de modèle.

Note:
Voir aussi : Template dashboard object (reportez-vous à la propriété correspondante portant le même nom).

Widgets de tableau de bord de modèle

```
zabbix_export:
  (...)
  templates:
    (...)
      dashboards:
        - uuid: c689ad3115fd46a4b927d1f70ee2e5a4
          name: Systèmes de fichiers
          pages:
            - name: Vue d'ensemble
              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})] : graphique d''utilisation de l''espace (par rapp
                    - 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})] : graphique d''utilisation de l''espace, en % (pa
                    - 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.

Element	Type	Description
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 : Template dashboard widget object (reportez-vous à la propriété correspondante portant le même nom).

Mappages de valeurs de modèle

```
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
        (...)
```

Élément	Type	Description
uuid	string	(obligatoire) Identifiant unique pour ce mappage de valeurs.
name	string	(obligatoire) Nom du mappage de valeurs.
mapping		Élément racine pour les correspondances.
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	(obligatoire) Valeur à laquelle la valeur d'origine est mappée.

Note:
Voir aussi : Value map 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.

4 Hôtes

Vue d'ensemble

Les hôtes sont exportés avec de nombreux objets liés et relations entre objets.

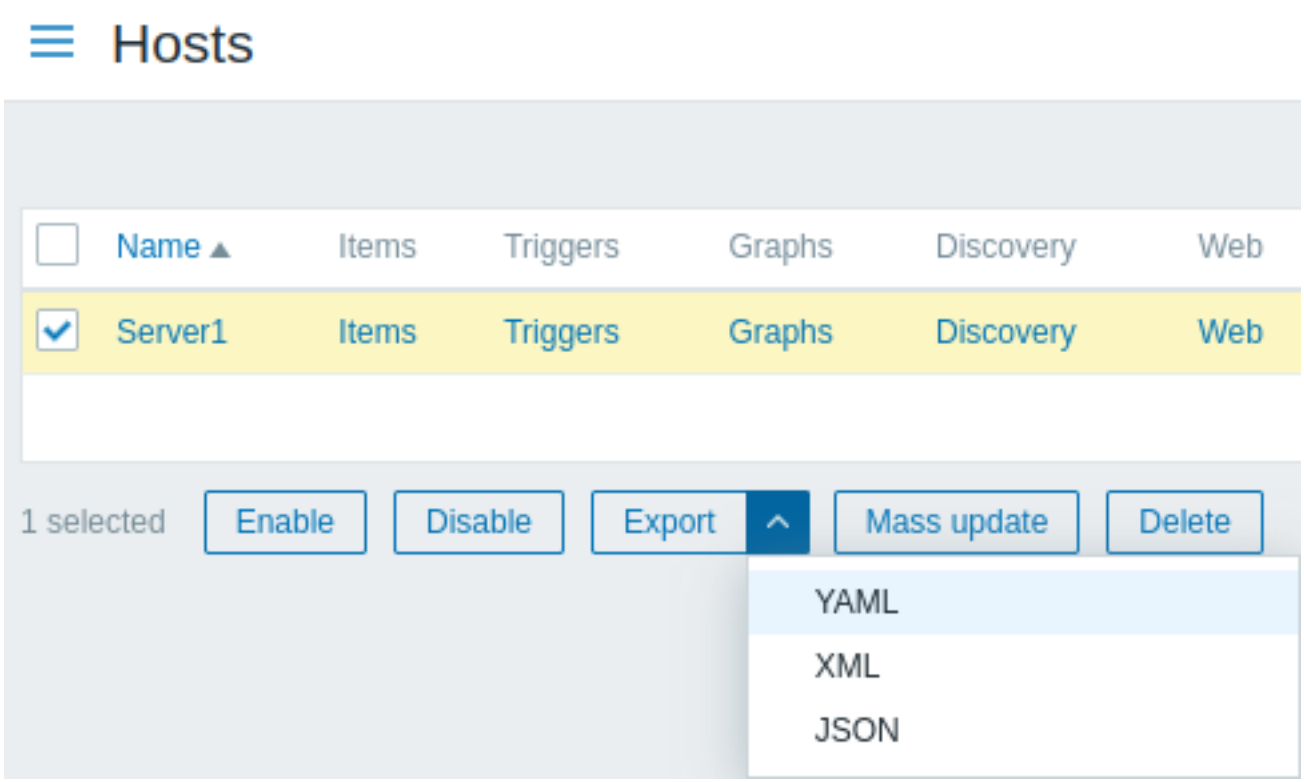
L'export d'hôte contient :

- Les groupes d'hôtes liés
- Les données de l'hôte
- La liaison au modèle
- La liaison au groupe d'hôtes
- Les interfaces de l'hôte
- Les éléments directement liés
- Les déclencheurs directement liés
- Les règles de découverte directement liées avec tous les prototypes
- Les scénarios web directement liés
- Les macros d'hôte
- Les données d'inventaire de l'hôte
- Les correspondances de valeurs
- Les 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.



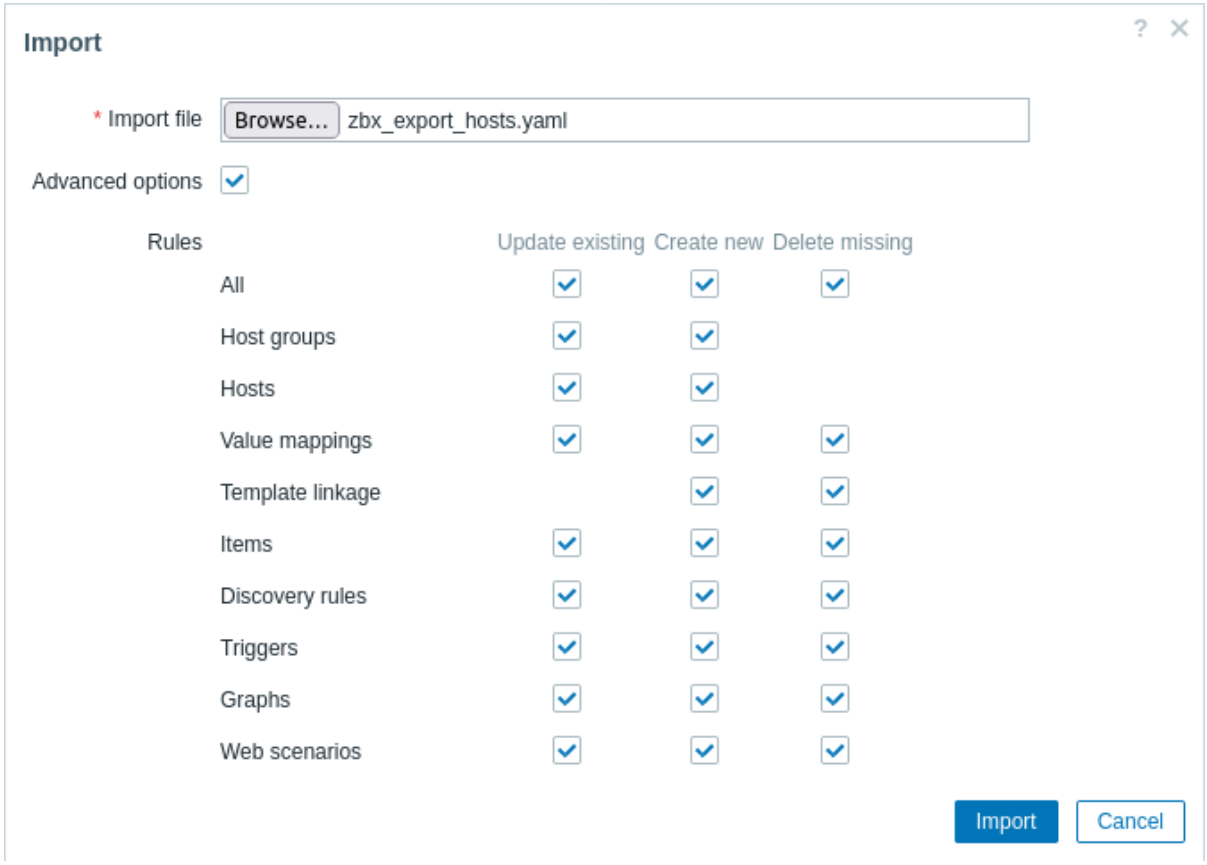
Selon le format sélectionné, les hôtes sont exportés vers un fichier local portant un nom par défaut :

- `zabbix_export_hosts.yaml` - pour l'exportation au format YAML (option par défaut pour l'exportation) ;
- `zabbix_export_hosts.xml` - pour l'exportation au format XML ;
- `zabbix_export_hosts.json` - pour l'exportation au format 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. 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 importables s'affichera ; cochez ou décochez chaque règle d'importation selon les besoins.

Si vous cliquez sur la case à cocher de la ligne *Tous*, tous les éléments en dessous seront cochés/décochés.

Règles d'importation :

Règle	Description
Mettre à jour l'existant	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	De nouveaux éléments seront créés à l'aide des données du fichier d'importation. Sinon, ils ne seront pas créés.
Supprimer les manquants	Les éléments existants non présents dans le fichier d'importation seront supprimés. Sinon, ils ne seront pas supprimés. Si *Supprimer les manquants* est coché pour *Liaison de modèles*, la liaison actuelle des modèles non présente dans le fichier d'importation sera dissociée. Les entités (éléments, déclencheurs, graphiques, etc.) héritées des modèles dissociés ne seront pas supprimées (sauf si l'option *Supprimer les manquants* est également sélectionnée pour chaque entité).

Un message indiquant le succès ou l'échec de l'importation sera affiché dans l'interface web.

Format d'exportation

Lorsqu'un hôte est exporté, Zabbix génère un format structuré YAML, JSON ou XML. L'exportation inclut des éléments de l'hôte, tels que les métadonnées de l'hôte, les éléments de données, les macros, les déclencheurs, et plus encore.

Chaque élément remplit un objectif spécifique et peut contenir des éléments imbriqués.

Les sections suivantes décrivent chaque élément du format d'exportation. 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 l'effacement, qui les suppriment).

Une ellipse (`...`) indique des éléments omis par souci de concision. La note (`see table below`) est utilisée lorsque l'élément est expliqué plus en détail dans les sections suivantes.

```
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

Élément		Type	Description
version		string	(obligatoire) Version de Zabbix.
host_groups			(obligatoire) Élément racine pour les groupes d'hôtes.
	uuid	string	(obligatoire) Identifiant unique de ce groupe d'hôtes.
	name	string	(obligatoire) Nom du groupe d'hôtes.
hosts			Élément racine pour les hôtes.
graphs			Élément racine pour les graphiques d'hôte.

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      name: 'MySQL server'
      groups:
        - name: Databases
      interfaces: (voir le tableau ci-dessous)
      items: (voir le tableau ci-dessous)
      discovery_rules: (voir le tableau ci-dessous)
      tags:
        - tag: location
          value: Riga
      macros:
        (...)
        - macro: '{$MYSQL.DSN}'
          value: 192.0.2.0
        (...)
        - macro: '{$MYSQL.PASSWORD}'
          type: SECRET_TEXT
          description: 'Mot de passe de l'utilisateur MySQL.'
        (...)
      valuemaps: (voir le tableau ci-dessous)
  (...)
```

Hôtes

Element	Type	Description
host	string	(obligatoire) Nom d'hôte unique.
name	string	Nom visible de l'hôte.
description	text	Description de l'hôte.
monitored_by	string	Comment l'hôte est surveillé. Valeurs possibles :[1] SERVER (0, par défaut), PROXY (1) ou PROXY_GROUP (2).
proxy		Élément racine pour le proxy.
name	string	(obligatoire) Nom du proxy (le cas échéant) qui surveille l'hôte.
proxy_group		Élément racine pour le groupe de proxys.
name	string	(obligatoire) Nom du groupe de proxys (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 contrôles IPMI.
ipmi_password	string	Mot de passe pour les contrôles IPMI.
templates		Élément racine pour les modèles liés.
name	string	(obligatoire) Nom du modèle.
groups		Élément racine pour les groupes d'hôtes auxquels l'hôte appartient.
name	string	(obligatoire) 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 d'hôte.
http_tests		Élément racine pour les scénarios web d'hôte.
tags		Élément racine pour les tags d'hôte.
tag	string	(obligatoire) Nom du tag.
value	string	Valeur du tag.
macros		Élément racine pour les macros d'hôte.
macro	string	(obligatoire) 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 élément respectif (`type`, `name`, `os`, etc. ; voir par exemple 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 d'hôte.

Note:
Voir aussi : Host object (reportez-vous à la propriété correspondante portant le même nom).

Interfaces d'hôte

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      interfaces:
        - ip: 192.0.2.0
          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 type donné 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	(obligatoire pour les connexions IP) Adresse IP (IPv4 ou IPv6).
dns	string	(obligatoire 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 de SNMP. Valeurs possibles :[1] SNMPV1 (1), SNMP_V2C (2, par défaut), SNMP_V3 (3).
community	string	(obligatoire pour les éléments SNMPv1 et SNMPv2) Communauté SNMP.

Element	Type	Description
max_repetitions	string	Valeur maximale de répétition pour les requêtes groupées SNMP natives (GetBulkRequest-PDUs). Pris en charge pour les éléments SNMPv2 et SNMPv3 (éléments `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 groupées 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 : Host interface object (reportez-vous à la propriété correspondante portant le même nom).

Éléments d'hôte

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      items:
        (...)
        - name: 'Utilisation du disque du cache binlog'
          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 buffer pool'
          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 buf
      tags:
        - tag: component
          value: memory
      triggers: (voir le tableau ci-dessous)
    (...)
    - name: Temps de fonctionnement
      type: DEPENDENT
      key: mysql.uptime
      units: uptime
      description: 'Nombre de secondes pendant lesquelles le serveur a été en fonctionnement.'
      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)
    (...)
```

Element	Type	Description
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), ITEM_TYPE_NESTED (23).
snmp_oid	string	(obligatoire pour les éléments SNMP_AGENT) ID d'objet SNMP.
key	string	(obligatoire) Clé de l'élément.
delay	string	Intervalle de mise à jour de l'élément. Par défaut : 1m. La valeur sera toujours 0 pour les éléments TRAP.
history	string	Période (à l'aide d'un suffixe temporel, d'une macro utilisateur ou d'une macro LLD) pendant laquelle les données d'historique doivent être conservées. Par défaut : 31d.
trends	string	Période (à l'aide d'un suffixe temporel, d'une macro utilisateur ou d'une macro LLD) pendant laquelle les données de tendances doivent être conservées. Par défaut : 365d.
status	string	Statut 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 d'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).

Element	Type	Description
username	string	(obligatoire pour les éléments SSH et TELNET) Nom d'utilisateur pour l'authentification. Pris en charge pour les éléments SIMPLE, ODBC, JMX et HTTP_AGENT. Lorsqu'il est utilisé pour des éléments JMX, `password` (voir ci-dessous) doit également être spécifié, ou les deux éléments doivent être laissés vides.
password	string	(obligatoire pour les éléments SSH et TELNET) Mot de passe pour l'authentification. Pris en charge pour les éléments SIMPLE, ODBC, JMX et HTTP_AGENT. Lorsqu'il est utilisé pour des éléments JMX, `username` (voir ci-dessus) doit également être spécifié, ou les deux éléments doivent être laissés 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 renseigné par l'élément. Valeurs possibles :[1] NONE (0), ALIAS (4), etc. (voir Host inventory pour les champs pris en charge).
valuemap		Élément racine pour les correspondances de valeurs de l'élément.
name	string	(obligatoire) Nom de la correspondance 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 pour le prétraitement des valeurs de l'élément.
step		Élément racine pour les 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		(obligatoire pour les éléments DEPENDENT) Élément racine pour l'élément maître d'un 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 Timeouts des types d'éléments.
url	string	(obligatoire pour les éléments HTTP_AGENT) Chaîne URL.
query_fields		Élément racine pour les 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 pour les 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 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 POST. 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 du proxy HTTP(S). Pris en charge pour les éléments HTTP_AGENT.
headers		Élément racine pour les 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.
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).

Element	Type	Description
output_format	string	Comment traiter la réponse. Pris en charge pour les éléments HTTP_AGENT. Valeurs possibles :[1] RAW (0, par défaut), JSON (1).
allow_traps	string	Autoriser le renseignement de la valeur de manière similaire à 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 valider 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 valider que le nom d'hôte de la connexion correspond à celui figurant dans le certificat de l'hôte. Pris en charge pour les éléments HTTP_AGENT. Valeurs possibles :[1] NO (0, par défaut), YES (1).
tags		Élément racine pour les tags de l'élément.
tag	string	(obligatoire) Nom du tag.
value	string	Valeur du tag.
triggers		Élément racine pour les déclencheurs des éléments d'hôte.

Note:
Voir aussi : Item object (reportez-vous à la propriété pertinente portant le même nom).

Étapes de prétraitement de la valeur d'un élément d'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
          (...)
```

Élément	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 : Item preprocessing object (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 has not received any data for items for the last 30 minutes.'
              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: 'MySQL uptime is less than 10 minutes.'
              tags:
                - tag: scope
                  value: notice
        (...)
```

Élément	Type	Description
uuid	string	(obligatoire) Identifiant unique de ce déclencheur.
expression	string	(obligatoire) Expression du déclencheur.

Élément	Type	Description
recovery_mode	string	Base pour la 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 (pas de corrélation d'événements ou corrélation d'événements 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	Statut 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 (événement de problème unique ou événements de problème multiples). 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 de dépendance.
expression	string	(obligatoire) Expression du déclencheur de dépendance.
recovery_expression	string	Expression de récupération du déclencheur de dépendance.
tags		Élément racine pour les tags du déclencheur.
tag	string	(obligatoire) Nom du tag.
value	string	Valeur du tag.

Note:
Voir aussi : Trigger object (reportez-vous à la propriété pertinente 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	(obligatoire) La clé de la règle de découverte de bas niveau.
filter		Élément racine des filtres de règle de découverte de bas niveau de l'hôte.
lifetime	string	Période (en utilisant des secondes, un suffixe temporel ou une macro utilisateur) après laquelle les ressources qui ne sont plus découvertes 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 (en utilisant des secondes, un suffixe temporel ou une macro utilisateur) après laquelle les ressources qui ne sont plus découvertes 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 des éléments de prototype d'élément d'hôte, qui sont identiques à ceux des éléments d'hôte.
trigger_prototypes		Élément racine des éléments de prototype de déclencheur d'hôte, qui sont identiques à ceux des déclencheurs d'élément d'hôte.
graph_prototypes		Élément racine des prototypes de graphique d'hôte, qui sont identiques à ceux des graphiques d'hôte.
host_prototypes		Élément racine des prototypes d'hôte, qui sont identiques à ceux des hôtes.
parent_discovery_rule		Élément racine de la règle de découverte de bas niveau parente (ou du 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/du 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 de l'élément maître de la règle dépendante.
lld_macro_paths		Élément racine des chemins de macro de règle de découverte de bas niveau.
lld_macro	string	(obligatoire) Nom de la macro de règle de découverte de bas niveau.
path	string	(obligatoire) Sélecteur de la valeur qui sera affectée à la macro correspondante.
preprocessing		Élément racine du prétraitement des valeurs de la règle de découverte de bas niveau.
step		Élément racine des éléments d'étape de prétraitement des valeurs de la règle de découverte de bas niveau, identiques à ceux des étapes de prétraitement des valeurs d'élément d'hôte, mais avec moins de valeurs possibles. Voir aussi : objet de prétraitement de règle LLD.
overrides		Élément racine des règles de substitution de la règle de découverte de bas niveau.
name	string	(obligatoire) Nom unique de la substitution.
step	string	(obligatoire) Numéro d'ordre unique de la substitution.
stop	string	Arrêter le traitement des substitutions suivantes en cas de correspondance.

Element	Type	Description
filter		Élément racine des éléments de filtre de règle de substitution de découverte de bas niveau, identiques à ceux des filtres de règle de découverte de bas niveau de l'hôte.
operations		Élément racine des opérations de substitution de règle de découverte de bas niveau de l'hôte.

Note:
Voir aussi : objet de règle LLD (reportez-vous à la propriété pertinente portant le même nom).

Filtres de règle de découverte de bas niveau d'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
          (...)
```

Élément	Type	Description
evaltype	string	Remplacer 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 de 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 du filtre, mais sera généré à nouveau lors de leur demande ultérieure.

Note:
Voir aussi : objet de filtre de règle LLD (reportez-vous à la propriété correspondante portant le même nom).

Opérations de substitution de règle de découverte de bas niveau d'hôte

Élément	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 substitution. 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 substitution.
status	string	Statut de l'objet lors de l'opération de substitution.
discover	string	Indique si l'objet est ajouté à la suite de la découverte.

Élément	Type	Description
delay	string	Intervalle de mise à jour défini pour le prototype d'élément lors de l'opération de substitution.
history	string	Période de stockage de l'historique définie pour le prototype d'élément lors de l'opération de substitution.
trends	string	Période de stockage des tendances définie pour le prototype d'élément lors de l'opération de substitution.
severity	string	Sévérité du prototype de déclencheur définie lors de l'opération de substitution.
tags		Élément racine pour les tags définis pour l'objet lors de l'opération de substitution.
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 substitution.
name	string	(obligatoire) Nom du modèle.
inventory_mode	string	Mode d'inventaire du prototype d'hôte défini lors de l'opération de substitution.

Note:
Voir aussi : LLD rule override operation object (reportez-vous à la propriété correspondante portant le même nom).

Scénarios web d'hôte

Élément	Type	Description
uuid	string	(obligatoire) Identifiant unique de ce scénario web.
name	string	(obligatoire) Nom du scénario web.
delay	string	Fréquence d'exécution du scénario web (en utilisant des secondes, un suffixe temporel ou une macro utilisateur). Par défaut : 1m.
attempts	integer	Nombre de tentatives pour exécuter les 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 le navigateur. 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 des 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 des 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 du 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).
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.

Élément	Type	Description
steps		(obligatoire) Élément racine des étapes du scénario web d'hôte.
tags		Élément racine des tags du scénario web.
tag	string	(obligatoire) Nom du tag.
value	string	Valeur du tag.

Note:
Voir aussi : Web scenario object (reportez-vous à la propriété correspondante portant le même nom).

Étapes du scénario web

Élément	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 de caractères (données POST brutes) 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 comporte le préfixe 'regex:', sa valeur est alors extraite des données renvoyées par cette étape conformément au motif 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'expiration (en secondes, suffixe temporel, 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 des 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 d'hôte

```
zabbix_export:
  (...)
  graphs:
    - name: 'MySQL: Bandwidth'
      graph_items: (voir le tableau ci-dessous)
    (...)
```

Élément	Type	Description
uuid	string	Identifiant unique du graphique.
name	string	(obligatoire) Nom du graphique.

Élément	Type	Description
width	integer	Largeur du graphique, en pixels. Utilisée pour l'aperçu et pour les graphiques en secteurs/explosés. Valeurs possibles:[1] 20-65535 (par défaut : 900).
height	integer	Hauteur du graphique, en pixels. Utilisée pour l'aperçu et pour les graphiques en secteurs/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 de déclencheur simples 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 d'élément individuels.
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 d'élément individuels.
host	string	(obligatoire) Hôte de l'élément.
key	string	(obligatoire) Clé de l'élément.
graph_items		(obligatoire) Élément racine pour les éléments de graphique d'hôte.

Note:
Voir aussi : Graph object (reportez-vous à la propriété correspondante portant le même nom).

Éléments de graphique d'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 plus petite valeur 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 plusieurs valeurs existent 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 secteurs/éclatés).
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 secteur ; pris en charge pour les graphiques en secteurs/éclatés).
item		(obligatoire) Élément individuel.
host	string	(obligatoire) Hôte de l'élément.
key	string	(obligatoire) Clé de l'élément.

Note:
Voir aussi : Objet d'élément de graphique (reportez-vous à la propriété correspondante portant le même nom).

Correspondances de valeurs d'hôte

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      valuemaps:
        - name: Example value map
          mappings:
            - value: '1'
              newvalue: Example value
  (...)
```

Élément	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. 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 à laquelle la valeur d'origine est associée.

Note:
Voir aussi : Value map 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.

5 Cartes du réseau

Aperçu

La carte réseau exportée contient :

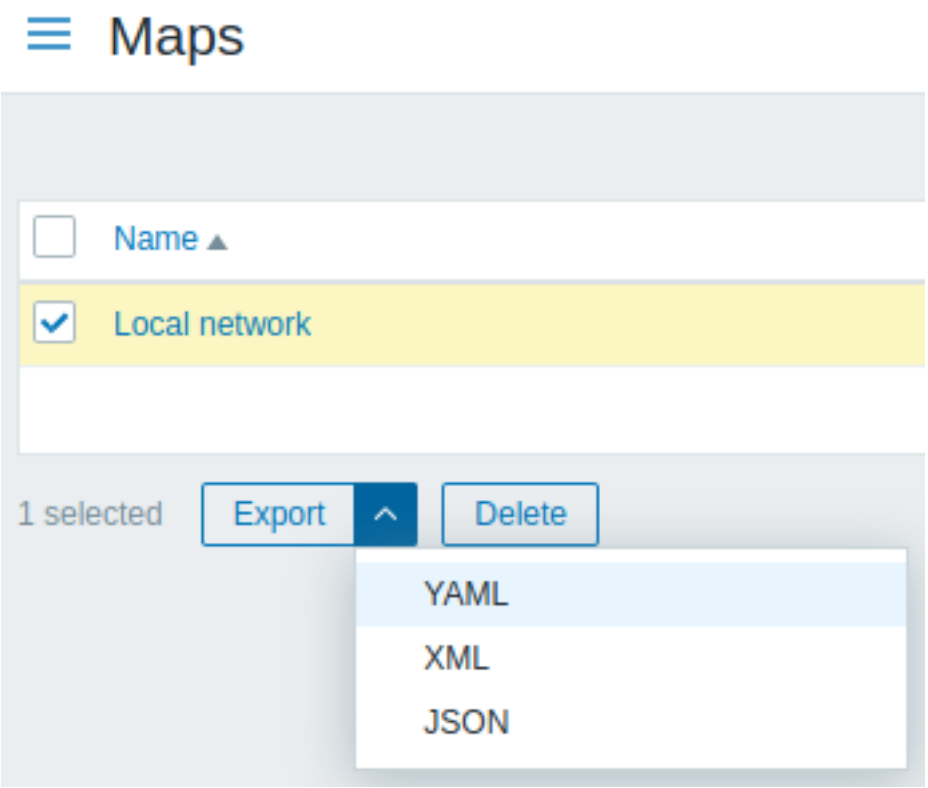
- Toutes les images associées
- La structure de la carte (tous les paramètres de la carte, tous les éléments contenus avec leurs paramètres, liens de carte et indicateurs d'état des liens de carte)

Warning:
Les groupes d'hôtes, hôtes, déclencheurs, autres cartes ou autres éléments pouvant être liés à la carte exportée ne sont pas exportés. Ainsi, si au moins un des éléments auxquels la carte fait référence est manquant, son importation échouera.

Exportation

Pour exporter des cartes de réseau, procédez comme suit :

1. Accédez à *Surveillance* → *Cartes*.
2. Cochez les cases des cartes de réseau à exporter.
3. Cliquez sur *Exporter* sous la liste.



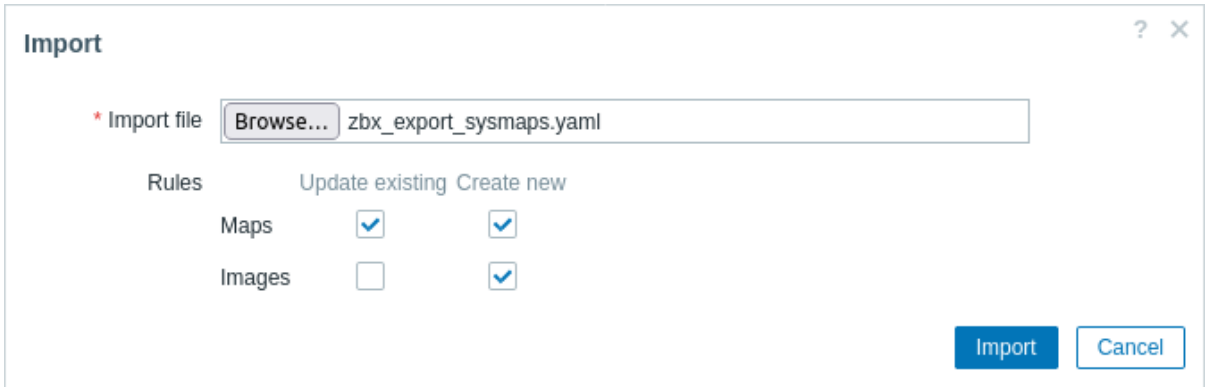
Selon le format sélectionné, les cartes sont exportées vers un fichier local avec un nom par défaut :

- `zabbix_export_maps.yaml` - dans l'export YAML (option par défaut pour l'export) ;
- `zabbix_export_maps.xml` - dans l'export XML ;
- `zabbix_export_maps.json` - dans l'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 options requises dans les règles d'importation.
5. Cliquez sur *Import* dans le coin inférieur droit du formulaire de configuration.



Règles d'importation :

Règle	Description
Update existing	Les cartes existantes seront mises à jour à l'aide des données du fichier d'importation. Sinon, elles ne seront pas mises à jour.
Create new	De nouvelles cartes seront créées à l'aide des données du fichier d'importation. Sinon, elles ne seront pas créées.

Si vous décochez les deux options de carte et cochez les options correspondantes pour les images, seules les images seront importées. L'importation d'images est disponible uniquement pour les utilisateurs *Super admin*.

Un message indiquant le succès ou l'échec de l'importation sera affiché dans le frontend.

Warning:
Si vous remplacez une image existante, cela affectera toutes les cartes qui utilisent cette image.

Format d'exportation

Exporter au format YAML :

```yaml
zabbix_export:
  version: '8.0'
  images:
    - name: Zabbix_server_3D_(128)
      imagetype: '1'
      encodedImage: iVBOR...5CYII=
  maps:
    - name: 'Local network'
      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.

Élément	Type	Description
images		Élément racine pour les images.
name	string	Nom unique de l'image.

Élément	Type	Description
imagetype	integer	Type d'image. Valeurs possibles : 1 - Image ; 2 - Arrière-plan.
encodedImage	string	Image encodée en Base64.
maps		Élément racine pour les cartes.

Cartes

Élément	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 par défaut de l'étiquette de l'élément de carte. Valeurs possibles : 0 - Bas ; 1 - Gauche ; 2 - Droite ; 3 - Haut.
highlight	integer	Activer la mise en surbrillance 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 surbrillance les éléments de carte dont le statut a récemment changé. Valeurs possibles : 0 - Non ; 1 - Oui.
show_unack	integer	Affichage des problèmes. Valeurs possibles : 0 - Nombre de tous les problèmes ; 1 - Nombre de problèmes non acquittés ; 2 - Nombre de problèmes acquittés et non acquittés séparément.
severity_min	integer	Gravité minimale du déclencheur à afficher sur la carte par défaut. Valeurs possibles : 0 - Non classé ; 1 - Information ; 2 - Avertissement ; 3 - Moyenne ; 4 - Haute ; 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 de cellule d'une grille de carte en pixels. Pris en charge si `grid_show` est défini sur 0. Valeurs possibles : 20, 40, 50, 75 ou 100.

Élément	Type	Description
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 la 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.
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.

Élément	Type	Description
expand_macros	integer	Développer les macros dans les étiquettes dans 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 d'icônes (le cas échéant).
name	string	Nom du mappage d'icônes.
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.
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.
shapes		Élément racine pour les formes de carte.
type	integer	Type de forme. Valeurs possibles : 0 - Rectangle ; 1 - Ellipse.
x	integer	Coordonnées X de la forme en pixels.
y	integer	Coordonnées 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.

Élément		Type	Description
	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 bordure représentée en code hexadécimal.
	background_color	string	Couleur d'arrière-plan (remplissage) représentée en code hexadécimal.
	zindex	integer	Valeur pour ordonner toutes les formes et lignes (z-index).
lines			Élément racine pour les lignes de carte.
	x1	integer	Coordonnées X du point 1 de la ligne en pixels.
	y1	integer	Coordonnées Y du point 1 de la ligne en pixels.
	x2	integer	Coordonnées X du point 2 de la ligne en pixels.
	y2	integer	Coordonnées 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 ligne en pixels.
	line_color	string	Couleur de ligne représentée en code hexadécimal.
	zindex	integer	Valeur pour ordonner toutes les formes et lignes (z-index).
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 symboles, hex).
	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 (reportez-vous à la propriété correspondante portant le même nom).

Éléments de carte

Élément	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 élément respectif (`host`, etc. ; voir, par exemple, Format d'export).
label	string	Libellé de l'icône.

Élément	Type	Description
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 la carte entière ; 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 - Répartir uniformément dans la zone.
use_iconmap	integer	Utiliser le mapping d'icônes pour cet élément. Pertinent uniquement si le mapping d'icônes est activé au niveau de la carte. Valeurs possibles : 0 - Non ; 1 - Oui.
selementid	id	ID unique de l'enregistrement de l'élément.
icon_off		Élément racine pour l'image à utiliser lorsque l'élément est à l'état `OK`.
name	string	Nom unique de l'image.
icon_on		Élément racine pour l'image à utiliser lorsque l'élément est à 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 tags.
tags		Élément racine pour les tags de problème (pour les éléments hôte et groupe d'hôtes). Si des tags sont fournis, seuls les problèmes avec ces tags seront affichés sur la carte.
tag	string	Nom du tag.
value	string	Valeur du tag.
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

Élément	Type	Description
drawtype	integer	Style du lien lorsque le déclencheur est dans l'état « Problème ». Valeurs possibles : 0 - Ligne ; 2 - Ligne en gras ; 3 - Ligne pointillée ; 4 - Ligne en tirets.
color	string	Couleur du lien (6 caractères, hexadécimal) lorsque le déclencheur est dans l'état « Problème ».
trigger		Élément racine pour le déclencheur utilisé afin d'indiquer l'état du lien.
description	string	Nom du déclencheur.
expression	string	Expression du déclencheur.
recovery_expression	string	Expression de récupération du déclencheur.

Note:
Voir aussi : Objet de déclencheur de lien de carte (reportez-vous à la propriété correspondante portant le même nom).

6 Types de média

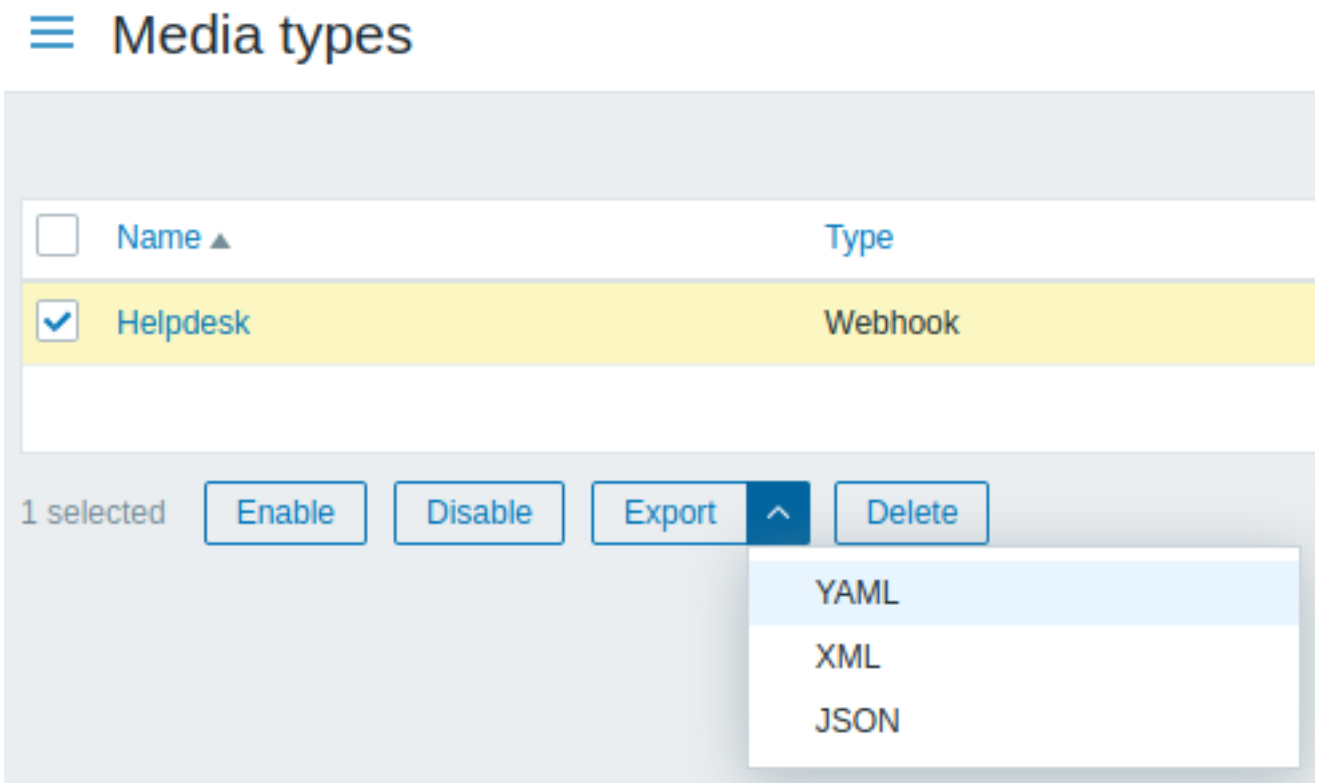
Vue d'ensemble

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 *Alertes* → *Types de média*.
2. Cochez les cases des types de média à exporter.
3. Cliquez sur *Exporter* sous la liste.



Selon le format sélectionné, les types de média sont exportés vers un fichier local portant un nom par défaut :

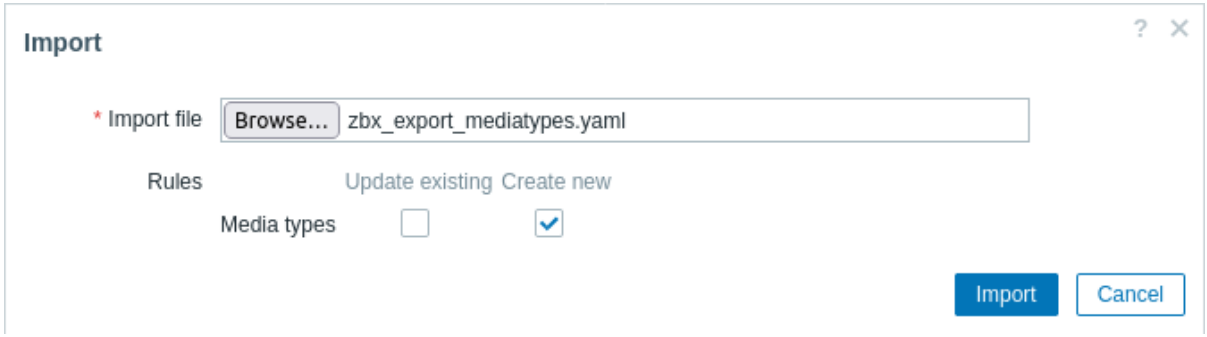
- `zabbix_export_mediatypes.yaml` - pour l'exportation YAML (option par défaut pour l'exportation) ;
- `zabbix_export_mediatypes.xml` - pour l'exportation XML ;

- `zabbix_export_mediatypes.json` - pour l'exportation 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 options requises dans les règles d'importation.
5. Cliquez sur *Importer* dans le coin inférieur droit du formulaire de configuration.



Règles d'importation :

Règle	Description
Mettre à jour l'existant	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	De nouveaux éléments seront créés à l'aide des données du fichier d'importation. Sinon, ils ne seront pas créés.

Un message indiquant le succès ou l'échec de l'importation sera affiché dans l'interface web.

Format d'exportation

Exporter 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: '<JETON PUSHOVER ICI>'
      - 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 êt
          }

          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
          }

          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'au moins 30 si "priority" est défini sur
    }

    if (isNaN(params.expire) || params.expire > 10800) {
        throw 'le champ "expire" doit être un nombre d'au plus 10800 si "priority" est défini s
    }

    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 d'état ' + 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 Pushover
        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 votre clé d'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}, acquitté : {EV
    - event_source: DISCOVERY
      operation_mode: PROBLEM
      subject: 'Découverte : {DISCOVERY.DEVICE.STATUS} {DISCOVERY.DEVICE.IPADDRESS}'
      message: |
        Règle de découverte : {DISCOVERY.RULE.NAME}

        IP du périphérique : {DISCOVERY.DEVICE.IPADDRESS}
        DNS du périphérique : {DISCOVERY.DEVICE.DNS}
        État du périphérique : {DISCOVERY.DEVICE.STATUS}
        Temps de fonctionnement du périphérique : {DISCOVERY.DEVICE.UPTIME}

        Nom du service du périphérique : {DISCOVERY.SERVICE.NAME}
        Port du service du périphérique : {DISCOVERY.SERVICE.PORT}
```

```
            État du service du périphérique : {DISCOVERY.SERVICE.STATUS}
            Temps de fonctionnement du service du périphérique : {DISCOVERY.SERVICE.UPTIME}
        - event_source: AUTOREGISTRATION
          operation_mode: PROBLEM
          subject: 'Enregistrement automatique : {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.

Élément	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	Le 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	Le nombre maximal de tentatives d'envoi d'une alerte. Valeurs possibles :[1] 1-10 (par défaut : 3).
attempt_interval	string	L'intervalle (en utilisant des secondes ou un suffixe temporel) entre les tentatives de nouvel essai. Valeurs possibles :[1] 0-60s (par défaut : 10s).
description	string	Description du type de média.
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	Sujet du message.
message	string	Corps du message.

Note:
Voir aussi : Media type object (reportez-vous à la propriété correspondante portant le même nom).

E-mail

Les éléments supplémentaires suivants sont exportés uniquement pour le type de média *E-mail*.

Élément	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 de l'hôte SSL pour SMTP. Valeurs possibles :[1] NO (0, par défaut), YES (1).
smtp_verify_peer	string	Vérification du pair SSL 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.

Élément	Type	Description
password	string	Mot de passe d'authentification.
redirection_url	string	URL du frontend Zabbix vers laquelle rediriger après l'autorisation OAuth.
client_id	string	Identifiant du client enregistré sur le serveur d'autorisation OAuth.
authorization_url	string	URL OAuth, avec paramètres, pour obtenir les 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 : Media type object (reportez-vous à la propriété correspondante portant le même nom).

SMS

Les éléments supplémentaires suivants sont exportés uniquement pour le type de média *SMS*.

Élément	Type	Description
gsm_modem	string	(obligatoire) Nom du périphérique série du modem GSM.

Note:
Voir aussi : Objet de type de média (reportez-vous à la propriété correspondante portant le même nom).

Script

Les éléments supplémentaires suivants sont exportés uniquement pour le type de média *Script*.

Élément	Type	Description
nom du script	string	(obligatoire) Nom du script.
parameters		Élément racine pour les paramètres du script.
sortorder	string	(obligatoire) Ordre des paramètres du script passés au script comme arguments de ligne de commande.
value	string	Valeur du paramètre du script.

Note:
Voir aussi : Objet de type de média (reportez-vous à la propriété correspondante portant le même nom).

Webhook

Les éléments supplémentaires suivants sont exportés uniquement pour le type de média *Webhook*.

Élément	Type	Description
script	string	Script.
timeout	string	Intervalle de délai d'expiration des requêtes 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 des événements 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 des événements. Prend en charge la macro {EVENT.TAGS.*}.
event_menu_name	string	Nom de l'entrée du menu des événements. 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.

Élément	Type	Description
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.

13 Découverte

Veuillez utiliser la barre latérale pour accéder au contenu de la section Découverte.

1 Découverte du réseau

Aperçu

Zabbix offre une fonctionnalité de découverte automatique du réseau, à la fois efficace et très flexible.

Avec une 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 repose 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 du 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 comporte un ensemble de contrôles de service définis pour être exécutés sur la plage d'adresses IP.

Les règles de découverte sont traitées par le gestionnaire de découverte. Le gestionnaire de découverte crée une tâche pour chaque règle avec une liste de sous-tâches (vérifications réseau). Les vérifications réseau sont effectuées en parallèle par les workers de découverte disponibles (leur nombre est configurable dans le frontend pour chaque règle). Seules les vérifications ayant la même adresse IP et le même port sont planifiées de manière séquentielle, car certains appareils n'acceptent pas les connexions parallèles sur le même port.

La taille de la file d'attente des vérifications réseau est limitée à environ 2000000 ou 4 Go de mémoire. Si la file d'attente est pleine, la règle de découverte sera ignorée et un message d'avertissement sera écrit 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 seront tout de même traitées.

Note:
Si une règle de découverte est modifiée pendant son exécution, l'exécution de 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.

Événement	Résultat de la vérification du service
Service découvert	Le service est « en ligne » après avoir été « hors ligne » ou lorsqu'il est découvert pour la première fois.
Service en ligne	Le service est « en ligne », après avoir déjà été « en ligne ».
Service perdu	Le service est « hors ligne » après avoir été « en ligne ».
Service hors ligne	Le service est « hors ligne », après avoir déjà été « hors ligne ».
Hôte découvert	Au moins un service d'un hôte est « en ligne » après que tous les services de cet hôte ont été « hors ligne », ou un service est découvert et appartient à un hôte non enregistré.
Hôte en ligne	Au moins un service d'un hôte est « en ligne », après qu'au moins un service a déjà été « en ligne ».
Hôte perdu	Tous les services d'un hôte sont « hors ligne » après qu'au moins un a été « en ligne ».
Hôte hors ligne	Tous les services d'un hôte sont « hors ligne », après avoir déjà été « hors ligne ».

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'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 un modèle à un hôte

Les hôtes créés sont ajoutés au groupe *Discovered hosts* (par défaut, configurable dans *Administration* > *General* > *Other*). Si vous souhaitez que les hôtes soient ajoutés à un autre groupe, ajoutez une opération *Retirer des groupes d'hôtes* (en spécifiant « Discovered hosts ») et ajoutez également une opération *Ajouter à des groupes d'hôtes* (en spécifiant un autre groupe d'hôtes), car un hôte doit appartenir à un groupe d'hôtes.

L'adresse IP du périphérique découvert, ainsi que la source de découverte (serveur Zabbix, proxy Zabbix ou groupe de proxys) 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 de l'exécution 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 prochaine découverte.

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'interfaces 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 répond à la fois aux requêtes de l'agent Zabbix et aux requêtes SNMP, les deux types d'interfaces seront créés.
- Si les critères d'unicité sont des données renvoyées par l'agent Zabbix ou par SNMP, la première interface trouvée pour un hôte sera créée comme interface par défaut. D'autres adresses IP seront ajoutées comme interfaces supplémentaires. Les conditions de l'action (telles que l'IP de l'hôte) n'ont pas d'impact sur l'ajout d'interfaces. *Notez* que cela fonctionnera si toutes les interfaces sont découvertes par la même règle de découverte. Si une autre règle de découverte découvre une interface différente du même hôte, un hôte supplémentaire sera ajouté.
- Si un hôte répond uniquement aux vérifications de l'agent, il sera créé avec une interface d'agent uniquement. S'il commence plus tard à répondre à SNMP, des interfaces SNMP supplémentaires seront ajoutées.
- Si 3 hôtes distincts ont été initialement 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. 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 de proxy

Les hôtes découverts par différents proxys ne sont pas toujours traités comme des hôtes différents. La découverte et les vérifications d'unicité dépendent de la structure du groupe de proxys : lorsqu'un proxy exécute une règle de découverte et crée un hôte, cet hôte est ajouté au groupe de proxys parent du proxy, et non attribué au proxy lui-même. Lorsque Zabbix évalue l'unicité de l'adresse IP pendant la découverte, il vérifie les hôtes supervisés par le groupe de proxys parent. Les hôtes supervisés par des proxys individuels au sein de ce groupe (y compris le proxy qui a exécuté la découverte) sont ignorés lors de la vérification d'unicité, ce qui peut entraîner des doublons d'hôtes si plusieurs proxys supervisent 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 sont utilisées par différents sous-réseaux, la modification du proxy attribué à un sous-réseau déjà supervisé est plus complexe, car les changements de proxy doivent être appliqués de manière cohérente aux hôtes découverts ainsi qu'à l'appartenance au groupe de proxys parent afin d'éviter les doublons.

Par exemple, voici 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 (s'assurer que les hôtes du groupe de proxys parent ainsi que tous les hôtes supervisés par des proxys individuels de ce groupe sont mis à jour afin d'éviter les doublons)
5. activer la règle de découverte

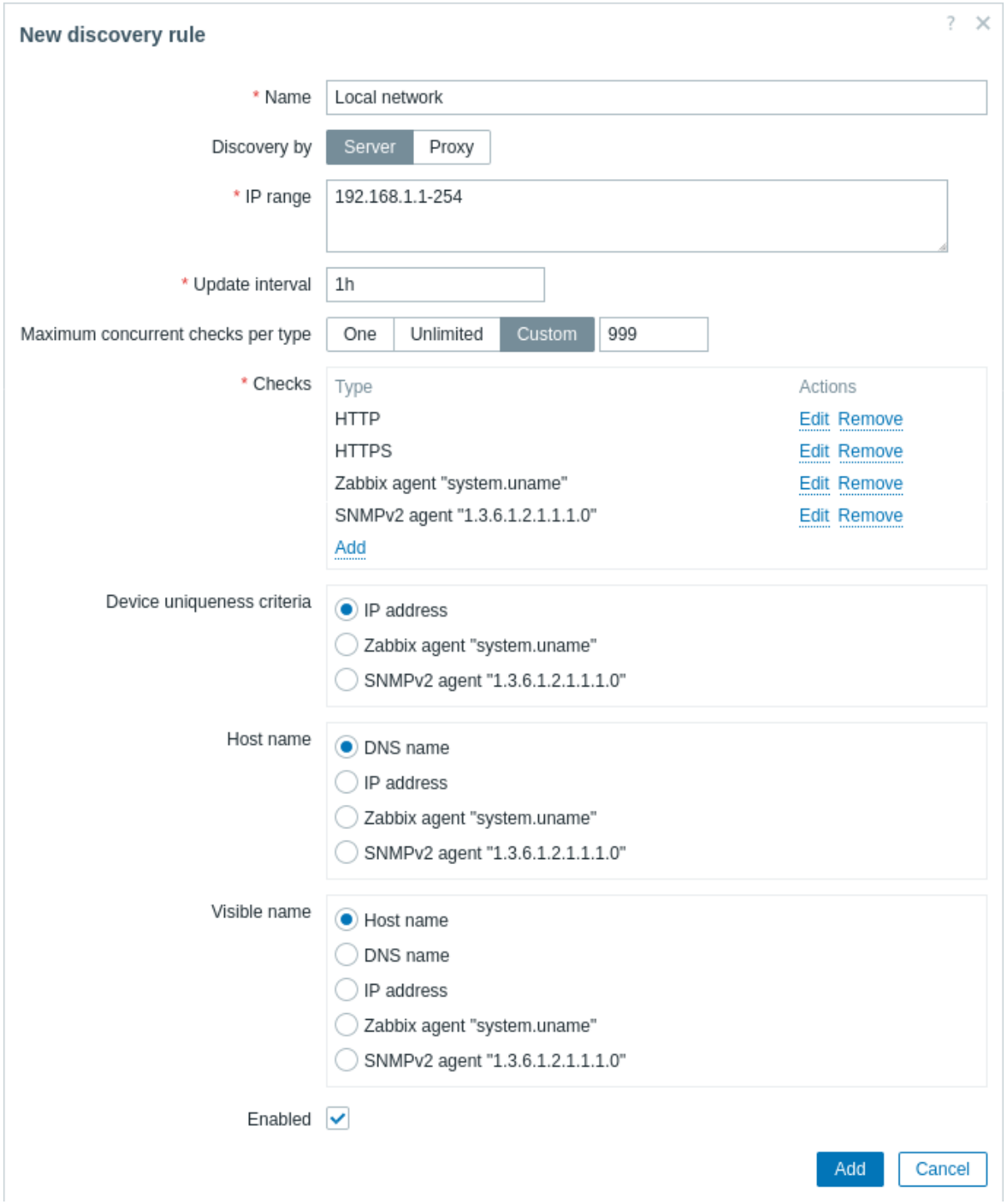
1 Configuration d'une règle de découverte réseau

Vue d'ensemble

Pour configurer une règle de découverte réseau utilisée par Zabbix pour découvrir des hôtes et des services :

- Allez dans *Collecte de données* → *Découverte*
- Cliquez sur *Créer une règle de découverte* (ou sur le nom de la règle pour modifier une règle existante)
- Modifiez les attributs de la règle de découverte

Attributs de la règle



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

Paramètre	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	La 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.
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, il n'y a donc pas de chevauchement. 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 ex. 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) Notez que 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 de libsnmp, c.-à-d. qu'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, agent Zabbix, agent SNMPv1, agent SNMPv2, agent SNMPv3, ping ICMP. Une découverte basée sur un protocole utilise la fonctionnalité **net.tcp.service[]** pour tester chaque hôte, à l'exception de SNMP qui interroge un OID SNMP. L'agent Zabbix est testé en interrogeant un élément en mode non chiffré. Veuillez consulter éléments d'agent pour plus de détails. Le paramètre « Ports » peut prendre l'une des formes suivantes : 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 le serveur/proxy Zabbix est compilé sans libcurl, les vérifications HTTP fonctionneront comme dans les versions précédentes (c.-à-d. comme des vérifications TCP), mais les vérifications HTTPS ne fonctionneront pas.
Device uniqueness criteria	Les critères d'unicité du périphérique peuvent être : **IP address** - ne pas traiter plusieurs périphériques à IP unique. Si un périphérique avec la même IP existe déjà, il sera considéré comme déjà découvert et un nouvel hôte ne sera pas ajouté. **<discovery check>** - soit une vérification de l'agent Zabbix, soit une vérification de l'agent SNMP. Notez que le critère d'unicité utilisé pendant la découverte n'est pas le même 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 périphériques découverts ou plus sont identiques (ou différents), alors que seule l'adresse IP est le critère d'identification de l'hôte dans Zabbix (voir Création d'hôte).

Paramètre	Description
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 de chaîne reçue de la vérification de découverte (par ex. agent Zabbix, vérification de l'agent SNMP) Voir aussi : Nommage 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 de chaîne reçue de la vérification de découverte (par ex. agent Zabbix, vérification de l'agent SNMP) Voir aussi : Nommage 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'épuiser la limite des descripteurs de fichiers pour le gestionnaire de découverte.

Le nombre de descripteurs de fichiers requis pour la détection est égal au nombre de workers de découverte * 1000. Par défaut, il y a 5 workers de découverte, tandis que la limite logicielle 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

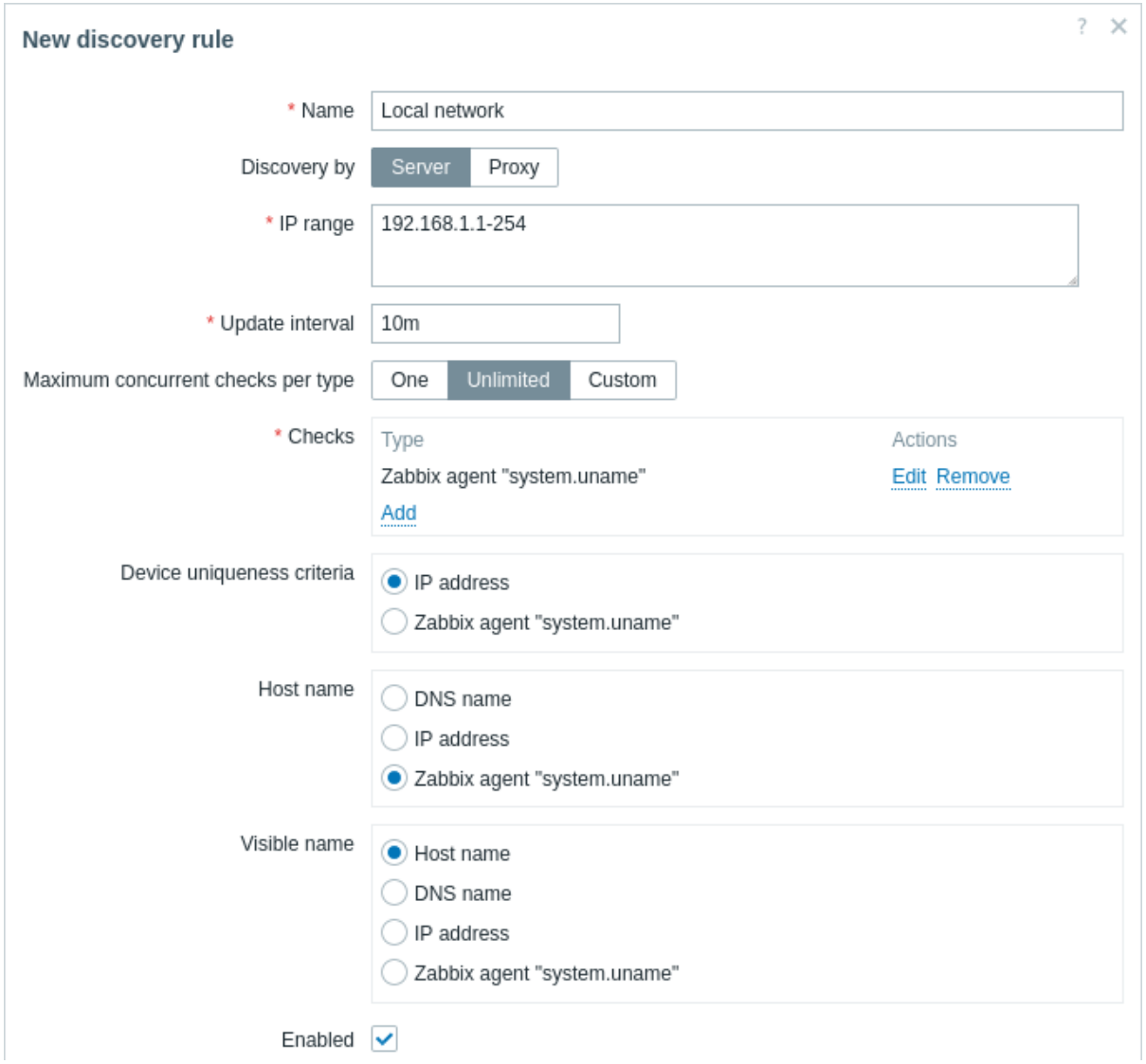
Dans cet exemple, nous souhaiterions configurer la détection de réseau pour le réseau local ayant une plage d'adresses IP : 192.168.1.1-192.168.1.254.

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.



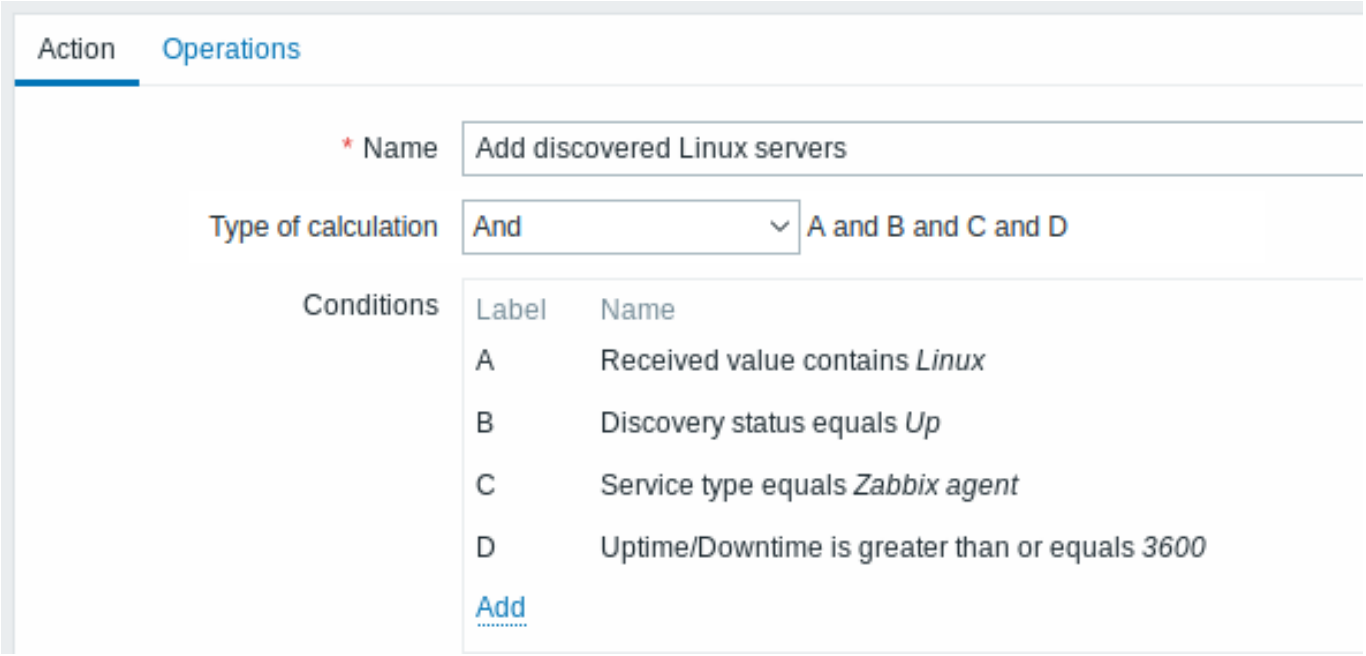
Zabbix tentera de découvrir les hôtes dans la plage d'adresses IP 192.168.1.1-192.168.1.254 en se connectant aux agents Zabbix et en récupérant la valeur de la clé **system.uname**. La valeur reçue de l'agent peut être utilisée pour nommer les hôtes et également pour appliquer différentes actions selon 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.

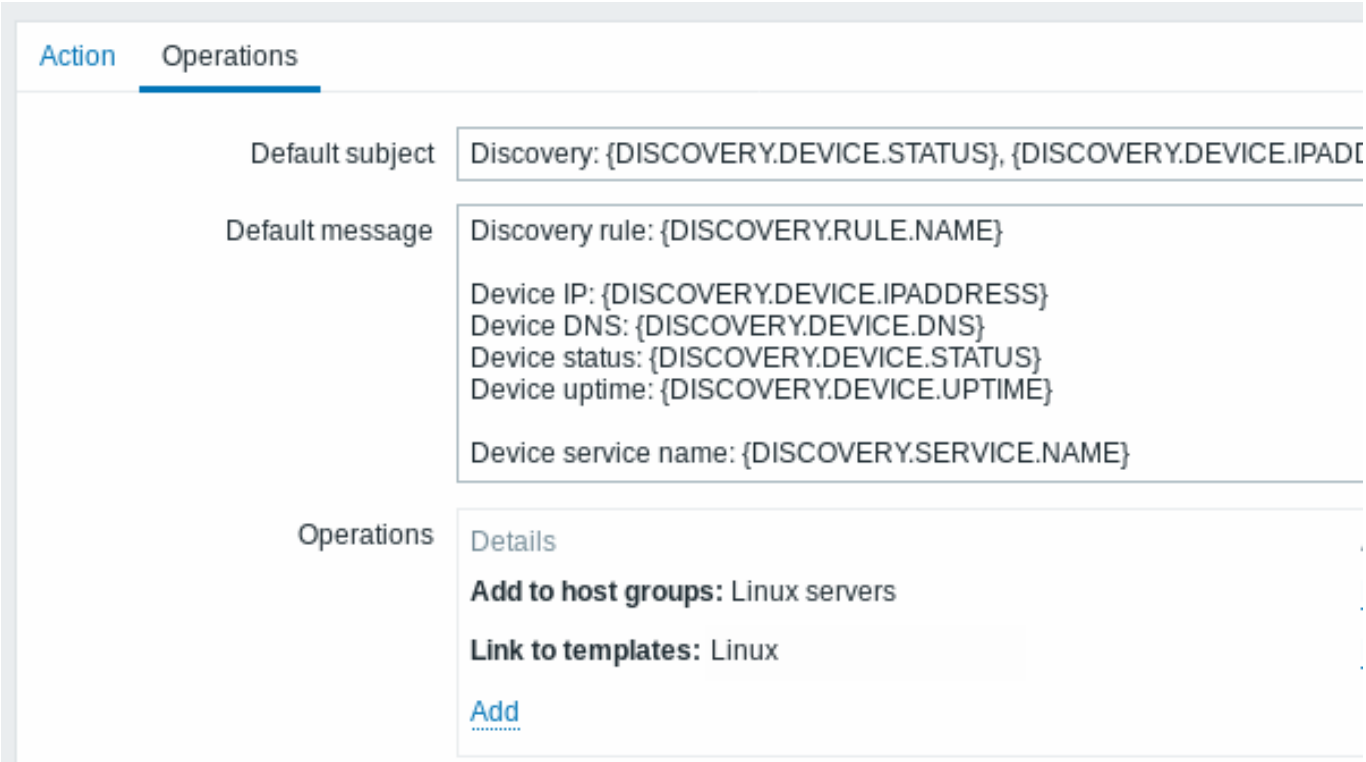
Étape 2

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

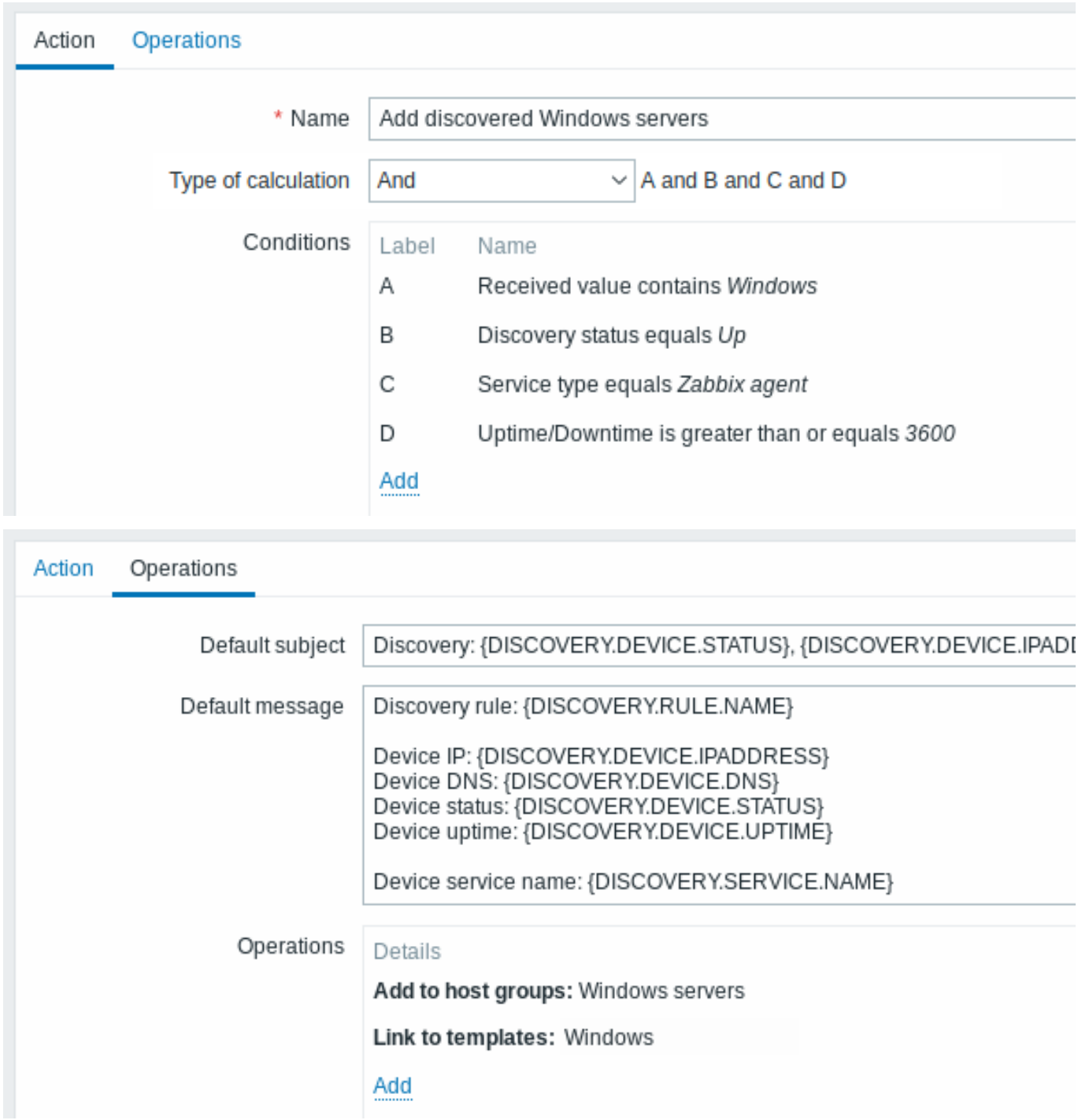


L'action exécutera les opérations suivantes :

- 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".

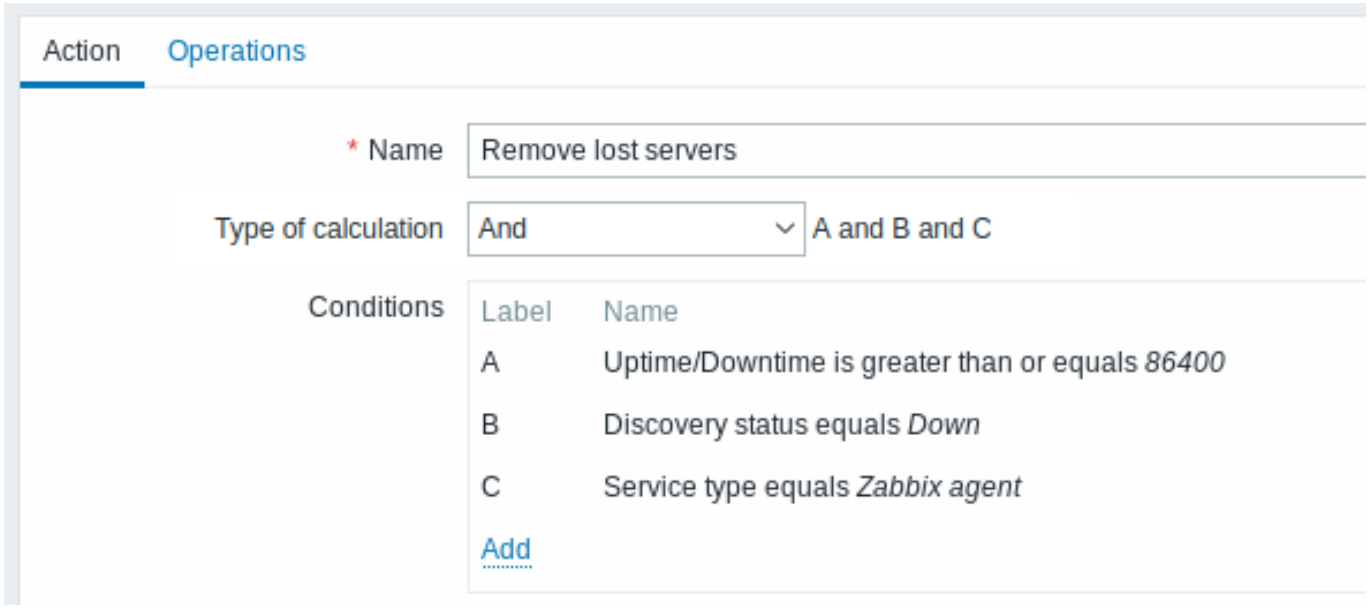
Étape 3

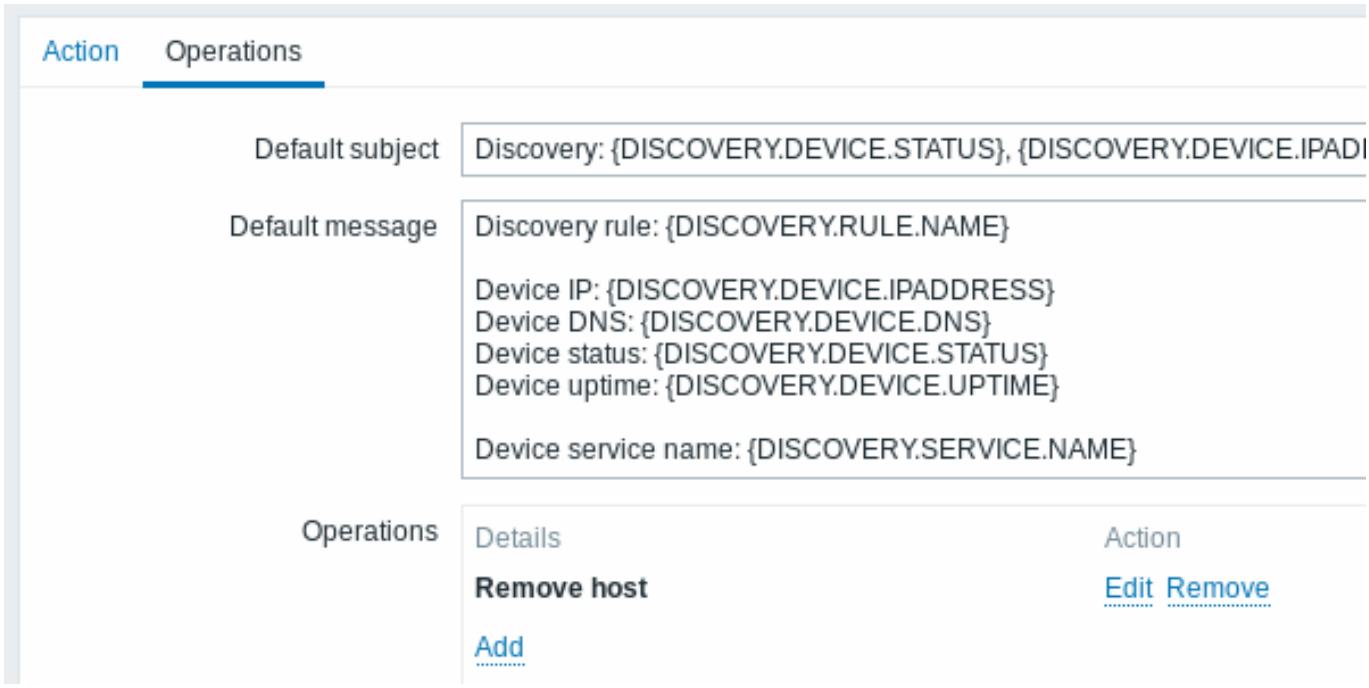
Définir une action pour ajouter les serveurs Windows découverts au groupe/modèle correspondant.



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).

2 Enregistrement automatique d’agent actif

Aperçu

Il est possible d’autoriser l’auto-enregistrement des agents Zabbix actifs, après quoi le serveur peut commencer à les surveiller. De cette manière, de nouveaux hôtes peuvent être ajoutés à la supervision sans les configurer manuellement sur le serveur.

L’auto-enregistrement peut se produire lorsqu’un agent actif auparavant inconnu demande des contrôles.

Cette fonctionnalité peut être très pratique pour la supervision automatique de nouveaux nœuds Cloud. Dès que vous disposez d’un nouveau nœud dans le Cloud, Zabbix commencera automatiquement à collecter les données de performance et de disponibilité de l’hôte.

L’auto-enregistrement des agents actifs prend également en charge la supervision des hôtes ajoutés avec des contrôles passifs. Lorsque l’agent actif demande des contrôles, à condition que les paramètres de configuration `ListenIP` ou `ListenPort` soient définis dans le fichier de configuration, ceux-ci sont également envoyés au serveur. Si plusieurs adresses IP sont spécifiées, la première est envoyée au serveur.

Lors de l’ajout du nouvel hôte auto-enregistré, le serveur 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 auto-enregistré avec un nom DNS comme interface d'agent par défaut.

L'auto-enregistrement est relancé :

- 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 dont les métadonnées sont absentes
- si un hôte est modifié manuellement pour être supervisé par un autre proxy Zabbix
- si l'auto-enregistrement du même hôte provient d'un nouveau proxy Zabbix

L'intervalle de pulsation de l'auto-enregistrement des agents actifs pour Zabbix server et Zabbix proxy est de 120 secondes. Ainsi, si un hôte découvert est supprimé, l'auto-enregistrement sera relancé dans 120 secondes.

Configuration

Spécifier le serveur

Assurez-vous d'avoir identifié le serveur Zabbix dans le fichier de configuration de l'agent - `zabbix_agentd.conf` :

```
ServerActive=10.0.0.1
```

À moins que vous ne définissiez explicitement un *Hostname* dans `zabbix_agentd.conf`, le nom d'hôte système de l'emplacement de l'agent sera utilisé par le serveur pour nommer l'hôte. Le nom d'hôte système sous Linux 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 avoir apporté des modifications au fichier de configuration.

Action pour l'autoréenregistrement de l'agent

Lorsque le serveur reçoit une demande d'autoréenregistrement d'un agent, il appelle une action. Une action avec la source d'événement « Autoréenregistrement » doit être configurée pour l'autoréenregistrement de l'agent.

> **Note:**
> La configuration de la découverte réseau n'est pas nécessaire pour que les agents actifs s'autoréenregistrent.

Dans l'interface Zabbix, allez dans *Alertes → Actions*, sélectionnez *Actions d'autoréenregistrement* et cliquez sur *Créer une action* :

- Dans l'onglet Action, donnez un nom à votre action
- Vous pouvez éventuellement spécifier des conditions. Vous pouvez effectuer une correspondance par sous-chaîne ou par expression régulière dans les conditions pour le nom d'hôte/les métadonnées de l'hôte. Si vous allez utiliser la condition « Métadonnées de l'hôte », consultez la section suivante.
- Dans l'onglet Opérations, ajoutez les opérations appropriées, telles que « Ajouter un hôte », « Ajouter à un groupe d'hôtes » (par exemple, *Hôtes découverts*), « Lier des modèles* », etc.

> **Note:**
> Si les hôtes qui vont s'autoréenregistrer ne sont susceptibles d'être pris en charge que pour la surveillance active uniquement (par exemple, des hôtes protégés par un pare-feu vis-à-vis de votre serveur Zabbix), vous pouvez créer un modèle spécifique comme *Template_Linux-active* à lier.

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 *Retirer du groupe d'hôtes* (en spécifiant « Hôtes découverts ») et ajoutez également une opération *Ajouter à un groupe d'hôtes* (en spécifiant un autre groupe d'hôtes), car un hôte doit appartenir à un groupe d'hôtes.

Autorenregistrement sécurisé

Une méthode sécurisée d'autorenregistrement est possible en configurant une authentification basée sur PSK avec des connexions chiffrées.

Le niveau de chiffrement est configuré globalement dans *Administration > Général > Autorenregistrement*. 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 le fassent via un chiffrement).

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/clé pré-partagée identique à celle du paramètre global d'autorenregistrement.

Attention:
Pour garantir la sécurité de l'autorenregistrement sur les installations utilisant des proxys, le chiffrement entre le serveur Zabbix et le proxy doit être activé.

Utilisation de DNS comme interface par défaut

Les paramètres de configuration HostInterface et HostInterfaceItem 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 d'agent par défaut plutôt que son adresse IP. Dans ce cas, le nom DNS doit être spécifié ou renvoyé comme valeur du paramètre HostInterface ou HostInterfaceItem. Si la valeur de l'un de ces paramètres change — par exemple, d'une adresse IP à un nom DNS ou inversement — l'interface par défaut de l'hôte enregistré automatiquement sera 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 inverse de l'adresse IP de l'agent. Si le DNS inverse n'est pas correctement configuré ou renvoie un nom non valide, cela peut entraîner un enregistrement automatique incorrect ou échoué en raison d'une valeur d'interface non valide.

Utilisation des métadonnées d'hôte

Lorsqu'un agent envoie une requête d'auto-enregistrement au serveur, il envoie son nom d'hôte. Dans certains cas (par exemple, les nœuds du cloud Amazon), un nom d'hôte ne suffit pas au serveur Zabbix pour différencier les hôtes découverts. Les métadonnées d'hôte peuvent être utilisées de manière facultative pour envoyer d'autres informations d'un agent au serveur.

Les métadonnées d'hôte sont configurées dans le fichier de configuration de l'agent - `zabbix_agentd.conf`. Il existe 2 façons de spécifier les métadonnées d'hôte dans le fichier de configuration :

```
HostMetadata
HostMetadataItem
```

Voir 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 sous 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 sur 3 octets sera limitée à 21844 caractères au total, tandis qu'une valeur contenant uniquement des caractères sur 4 octets sera limitée à 16383 caractères.

Attention:
Une tentative d'auto-enregistrement a lieu chaque fois qu'un agent actif envoie au serveur une requête pour actualiser les contrôles actifs. Le délai entre les requêtes est spécifié dans le paramètre RefreshActiveChecks de l'agent. La première requête 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.

3 Découverte de bas niveau

Aperçu La découverte de bas niveau (LLD) permet de créer automatiquement des éléments, des déclencheurs et des graphiques pour différentes entités sur un hôte. Par exemple, Zabbix peut commencer automatiquement à surveiller les systèmes de fichiers ou les interfaces réseau sur votre machine, sans qu'il soit nécessaire de créer manuellement des éléments pour chaque système de fichiers ou interface réseau. La LLD peut également créer des hôtes, par exemple pour renseigner des machines virtuelles découvertes sur un hyperviseur, ainsi que des règles de découverte imbriquées, permettant une découverte à plusieurs niveaux. De plus, il est possible de configurer Zabbix pour supprimer automatiquement les entités inutiles en fonction des résultats réels de découvertes effectuées périodiquement.

Un utilisateur peut définir ses propres types de découverte, à condition de suivre un protocole JSON particulier.

L'architecture générale du processus de découverte est la suivante.

Tout d'abord, un utilisateur crée une règle de découverte dans *Collecte de données* > *Modèles*, dans la colonne *Découverte*. Une règle de découverte se compose de (1) un élément qui découvre les entités nécessaires (par exemple, les systèmes de fichiers ou les interfaces réseau) et de (2) prototypes d'éléments, de déclencheurs et de graphiques qui doivent être créés en fonction de la valeur de cet élément.

Un élément qui découvre les entités nécessaires est semblable à un élément classique que l'on voit ailleurs : le serveur demande à un agent Zabbix (ou à tout autre type défini pour l'élément) la valeur de cet élément, et l'agent répond avec une valeur textuelle. La différence est que la valeur renvoyée par l'agent doit contenir une liste des entités découvertes au format JSON. Bien que les détails de ce format ne soient importants que pour les implémenteurs de contrôles de découverte personnalisés, il est nécessaire de savoir que la valeur renvoyée contient une liste de paires macro → valeur. Par exemple, l'élément "net.if.discovery" peut renvoyer deux paires : "{#IFNAME}" → "lo" et "{#IFNAME}" → "eth0".

Ces macros sont utilisées dans les noms, les clés et d'autres champs de prototype, où elles sont ensuite remplacées par les valeurs reçues afin de créer de véritables éléments, déclencheurs, graphiques, voire des hôtes, pour chaque entité découverte. Consultez la liste complète des options pour l'utilisation des macros LLD.

Lorsque le serveur reçoit une valeur pour un élément de découverte, il examine les paires macro → valeur et, pour chaque paire, génère de véritables éléments, déclencheurs et graphiques à partir de leurs prototypes. Dans l'exemple avec "net.if.discovery" ci-dessus, le serveur générerait un ensemble d'éléments, de déclencheurs et de graphiques pour l'interface de bouclage "lo", et un autre ensemble pour l'interface "eth0".

Notez que depuis **Zabbix 4.2**, le format du JSON renvoyé par les règles de découverte de bas niveau a changé. Il n'est plus attendu que le JSON contienne l'objet "data". La découverte de bas niveau accepte désormais un JSON normal contenant un tableau, afin de prendre en charge de nouvelles fonctionnalités telles que le prétraitement des valeurs d'élément et les chemins personnalisés vers les valeurs de macro de découverte de bas niveau dans un document JSON.

Les clés de découverte intégrées ont été mises à jour pour renvoyer un tableau de lignes LLD à la racine du document JSON. Zabbix extraira automatiquement une macro et une valeur si un champ de tableau utilise la syntaxe {#MACRO} comme clé. Toute nouvelle vérification de découverte native utilisera la nouvelle syntaxe sans les éléments "data". Lors du traitement d'une valeur de découverte de bas niveau, la racine est d'abord localisée (tableau à `$.` ou `$.data`).

Bien que l'élément "data" ait été supprimé de tous les éléments natifs liés à la découverte, pour des raisons de rétrocompatibilité, Zabbix acceptera toujours la notation JSON avec un élément "data", bien que son utilisation soit déconseillée. Si le JSON contient un objet avec un seul élément de tableau "data", alors le contenu de cet élément sera automatiquement extrait à l'aide de JSONPath `$.data`. La découverte de bas niveau accepte désormais des macros LLD facultatives définies par l'utilisateur avec un chemin personnalisé spécifié en syntaxe JSONPath.

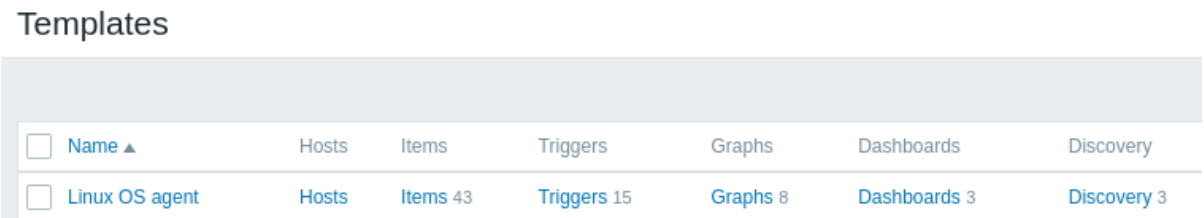
Warning:
En conséquence des changements ci-dessus, les agents plus récents ne pourront plus fonctionner avec un ancien serveur Zabbix.

Voir aussi : Entités découvertes

Configuration de la découverte de bas niveau Nous allons illustrer la découverte de bas niveau à partir d'un exemple de découverte de systèmes de fichiers.

Pour configurer la découverte, procédez comme suit :

- Allez dans : *Collecte de données* > *Modèles* ou *Hôtes*.
- Cliquez sur *Découverte* sur la ligne du modèle/hôte approprié.



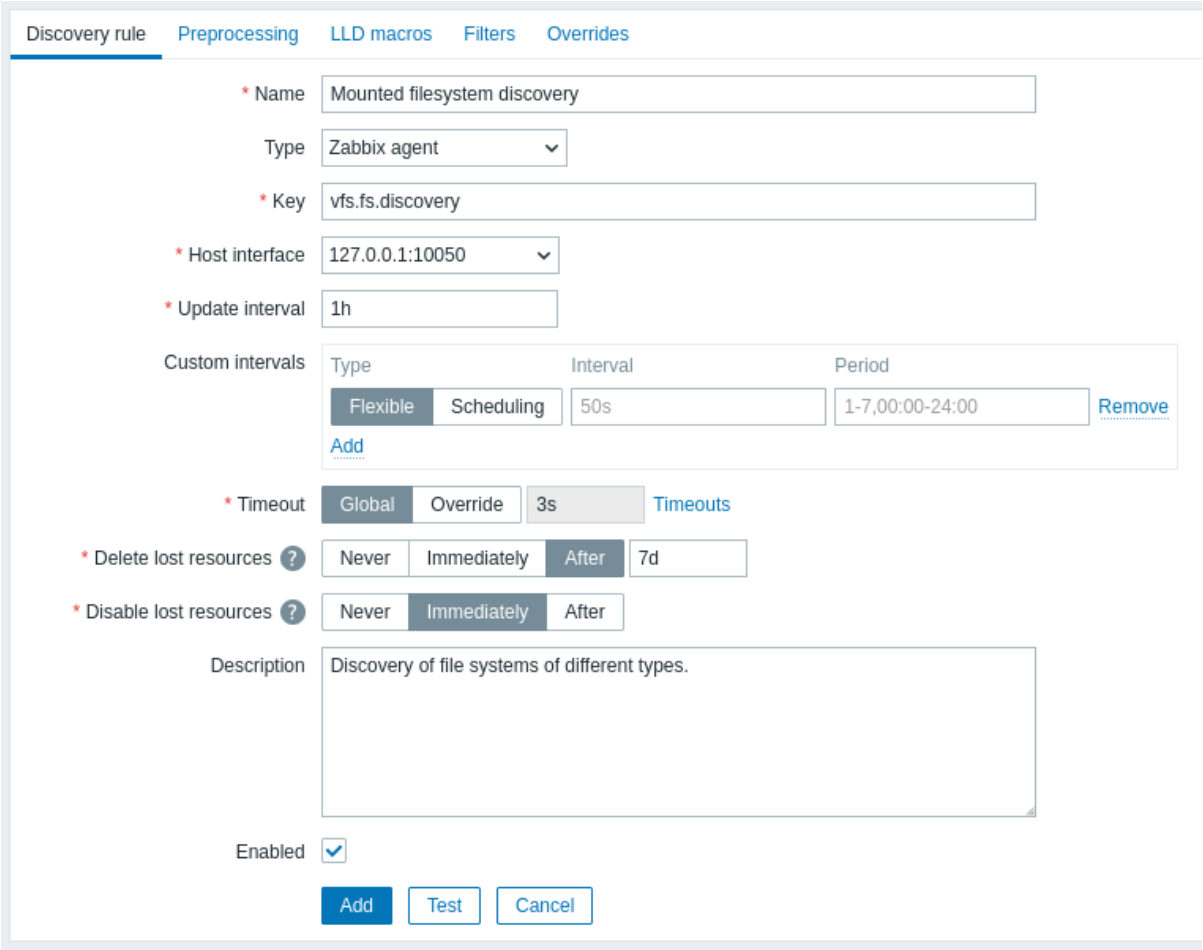
- Cliquez sur *Créer une règle de découverte* dans le coin supérieur droit de l'écran.
- Remplissez le formulaire de la règle de découverte avec les informations requises.

Règle de découverte

Le formulaire de règle de découverte contient cinq onglets, représentant, de gauche à droite, le flux de données pendant la découverte :

- *Règle de découverte* - spécifie, surtout, l'élément intégré ou le script personnalisé à utiliser pour récupérer les données de découverte.
- *Prétraitement* - applique un certain prétraitement aux données découvertes.
- *Macros LLD* - permet d'extraire certaines valeurs de macro à utiliser dans les éléments, déclencheurs, etc. découverts.
- *Filtres* - permet de filtrer les valeurs découvertes.
- *Surcharges* - permet de modifier les éléments, déclencheurs, graphiques ou prototypes d'hôte lors de l'application à des objets découverts spécifiques.

L'onglet **Règle de découverte** contient la clé d'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.

Parameter	Description
Name	Nom de la règle de découverte.
Type	Le type de contrôle à effectuer pour la découverte. Dans cet exemple, nous utilisons un type d'élément *Agent Zabbix*. La règle de découverte peut également être un élément dépendant, dépendant d'un élément normal. Elle ne peut pas dépendre d'une autre règle de découverte. Pour un élément dépendant, sélectionnez le type correspondant (*Élément dépendant*) et indiquez l'élément maître dans le champ 'Master item'. L'élément maître doit exister.
Key	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 option pour la découverte des systèmes de fichiers consiste à utiliser les résultats de découverte de la clé d'agent "vfs.fs.get" (voir exemple).
Update interval	Ce champ indique à quelle fréquence Zabbix effectue la découverte. Au début, lorsque vous configurez juste la découverte du système de fichiers, vous pouvez souhaiter définir un intervalle court, mais une fois que vous savez que cela fonctionne, vous pouvez le définir à 30 minutes ou plus, car les systèmes de fichiers ne changent généralement pas très souvent. Les suffixes temporels 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 à '0' que s'il existe des intervalles personnalisés avec une valeur non nulle. S'il est défini à '0' et qu'un intervalle personnalisé (flexible ou planifié) existe avec une valeur non nulle, 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'une planification ou d'un intervalle de mise à jour flexible et que *Update interval* est défini à 0. *Remarque* : pour une règle de découverte existante, la découverte peut être effectuée immédiatement en appuyant sur le bouton *Execute now*.

Parameter	Description
Custom intervals	Vous pouvez créer des règles personnalisées pour vérifier l'élément : **Flexible** - crée une exception à *Update interval* (intervalle avec une fréquence différente) **Scheduling** - crée une planification d'interrogation personnalisée. Pour des informations détaillées, voir Intervalles personnalisés.
Timeout	Définissez le délai d'expiration du contrôle de découverte. Sélectionnez l'option de délai d'expiration : **Global** - le délai d'expiration du proxy/global est utilisé (affiché dans le champ *Timeout* grisé) ; **Override** - un délai d'expiration personnalisé est utilisé (défini dans le champ *Timeout* ; plage autorisée : 1 - 600s). Les suffixes temporels, par exemple 30s, 1m, et les macros utilisateur sont pris en charge. Cliquer sur le lien *Timeouts* vous permet de configurer les délais d'expiration du proxy ou les délais d'expiration globaux (si un proxy n'est pas utilisé). Notez que le lien *Timeouts* n'est visible que pour les utilisateurs de type *Super admin* disposant des autorisations pour les sections du frontal *Administration* > *General* ou *Administration* > *Proxies*.
Delete lost resources	Indiquez dans quel délai l'entité découverte sera supprimée une fois que son statut de découverte devient "Not discovered anymore" : *Never* - elle ne sera pas supprimée ; *Immediately* - elle sera supprimée immédiatement ; *After* - elle sera supprimée après la période spécifiée. La valeur doit être supérieure à la valeur de *Disable lost resources*. Les suffixes temporels sont pris en charge, par exemple 2h, 1d. Les macros utilisateur sont prises en charge. *Remarque* : l'utilisation de "Immediately" n'est pas recommandée, car une simple modification erronée du filtre peut entraîner la suppression de l'entité avec toutes les données historiques. Notez que les ressources désactivées manuellement ne seront pas supprimées par la découverte de bas niveau.
Disable lost resources	Indiquez dans quel délai l'entité découverte sera désactivée une fois que son statut de découverte devient "Not discovered anymore" : *Never* - elle ne sera pas désactivée ; *Immediately* - elle sera désactivée immédiatement ; *After* - 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 réactivées si elles sont à nouveau découvertes par la découverte de bas niveau. Les ressources désactivées manuellement ne seront pas réactivées si elles sont à nouveau découvertes. Ce champ n'est pas affiché si *Delete lost resources* est défini sur "Immediately". Les suffixes temporels sont pris en charge, par exemple 2h, 1d. Les macros utilisateur sont prises en charge.
Description	Saisissez une description.
Enabled	Si cette case est cochée, la règle sera traitée.

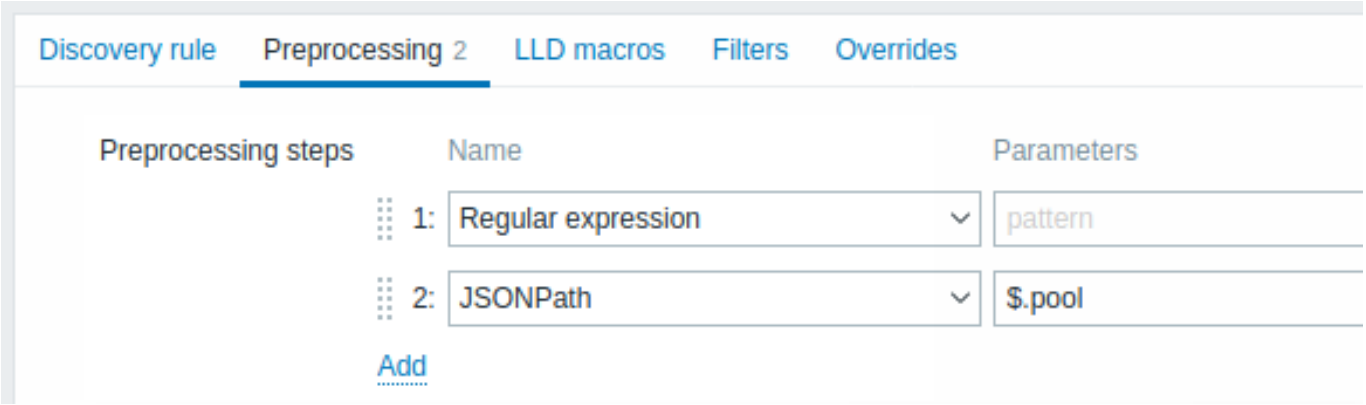
Note:
L'historique de la règle de découverte n'est pas conservé.

Prétraitement

L'onglet **Prétraitement** 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 aussi :

- 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 <output> extraite. L'expression régulière prend en charge l'extraction d'un maximum de 10 groupes capturés avec la séquence \N. Paramètres : **pattern** - expression régulière **output** - modèle de formatage de 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*, il est possible de spécifier des options personnalisées de gestion des erreurs : soit ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié.
	Remplacer	Rechercher la chaîne de recherche et la remplacer par une autre (ou rien). Toutes les occurrences de la chaîne de recherche seront remplacées. Paramètres : **search string** - la chaîne à rechercher et à remplacer, sensible à la casse (obligatoire) **replacement** - la chaîne qui remplace la chaîne de recherche. La chaîne de remplacement peut également être vide, ce qui permet effectivement de supprimer la chaîne de recherche lorsqu'elle est trouvée. Il est possible d'utiliser des séquences d'échappement pour rechercher ou remplacer des sauts de ligne, retours chariot, tabulations et espaces "\n \r \t \s" ; l'antislash peut être échappé sous la forme "\\" et les séquences d'échappement peuvent être échappées sous la forme "\\n". L'échappement des sauts de ligne, retours chariot et tabulations est effectué automatiquement lors de la découverte de bas niveau.
Données structurées		
	JSONPath	Extraire une valeur ou un fragment à partir de données JSON à l'aide de la fonctionnalité JSONPath. Si vous cochez la case *Personnalisé en cas d'échec*, il est possible de spécifier des options personnalisées de gestion des erreurs : soit ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié.

	XML XPath	Extraire une valeur ou un fragment à partir 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*, il est possible de spécifier des options personnalisées de gestion des erreurs : soit ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié.

Type	
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*, il est possible de spécifier des options personnalisées de gestion des erreurs : soit ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié.
SNMP	
Valeur SNMP walk	Extraire une valeur par le nom OID/MIB spécifié et appliquer des options de formatage : **Inchangé** - renvoyer Hex-STRING sous forme de chaîne hexadécimale non échappée (*notez* que les indications d'affichage sont toujours appliquées) ; **UTF-8 à partir de Hex-STRING** - convertir Hex-STRING en chaîne UTF-8 ; **MAC à partir de Hex-STRING** - convertir Hex-STRING en chaîne d'adresse MAC (dans laquelle `' '` sera remplacé par `':'`) ; **Entier à partir de BITS** - convertir les 8 premiers octets d'une chaîne de bits exprimée comme une séquence de caractères hexadécimaux (par ex. "1A 2B 3C 4D") en un entier non signé 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*, il est possible de spécifier des options personnalisées de gestion des erreurs : soit ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié.
SNMP walk vers JSON	Convertir des valeurs SNMP en JSON. Spécifiez un nom de champ dans le JSON et le chemin OID SNMP correspondant. Les valeurs des champs seront renseignées par les valeurs du chemin OID SNMP spécifié. Vous pouvez utiliser cette étape de prétraitement pour la découverte d'OID SNMP. Des options de formatage de valeur similaires à celles de l'étape *Valeur SNMP walk* sont disponibles. Si vous cochez la case *Personnalisé en cas d'échec*, il est possible de spécifier des options personnalisées de gestion des erreurs : soit ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié.
Valeur SNMP get	Appliquer des options de formatage à la valeur SNMP get : **UTF-8 à partir de Hex-STRING** - convertir Hex-STRING en chaîne UTF-8 ; **MAC à partir de Hex-STRING** - convertir Hex-STRING en chaîne d'adresse MAC (dans laquelle `' '` sera remplacé par `':'`) ; **Entier à partir de BITS** - convertir les 8 premiers octets d'une chaîne de bits exprimée comme une séquence de caractères hexadécimaux (par ex. "1A 2B 3C 4D") en un entier non signé 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*, il est possible de spécifier des options personnalisées de gestion des erreurs : soit ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié.
Scripts personnalisés	
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. Notez que la longueur disponible pour JavaScript dépend de la base de données utilisée. Pour plus d'informations, voir : Prétraitement JavaScript
Validation	
Ne correspond pas à l'expression régulière	Spécifiez une expression régulière à laquelle une valeur ne doit pas correspondre. Par ex. `Error:(.*?)\.` Si vous cochez la case *Personnalisé en cas d'échec*, il est possible de spécifier des options personnalisées de gestion des erreurs : soit ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié.

Type	
Vérifier une erreur dans JSON	Vérifier un message d'erreur au niveau de l'application situé à l'emplacement JSONPath. Arrêter le traitement si la vérification réussit et que le message n'est pas vide ; sinon, poursuivre le traitement avec la valeur qui existait avant cette étape de prétraitement. Notez que ces erreurs de service externe sont signalées à l'utilisateur telles quelles, sans ajout d'informations sur l'étape de prétraitement. Par ex. `$.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*, il est possible de spécifier des options personnalisées de gestion des erreurs : soit ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié.
Vérifier une erreur dans XML	Vérifier un message d'erreur au niveau de l'application situé à l'emplacement Xpath. Arrêter le traitement si la vérification réussit et que le message n'est pas vide ; sinon, poursuivre le traitement avec la valeur qui existait avant cette étape de prétraitement. Notez que ces erreurs de service externe sont signalées à l'utilisateur telles quelles, sans ajout d'informations sur l'étape de prétraitement. Aucune erreur ne sera signalée en cas d'échec de l'analyse d'un XML invalide. Si vous cochez la case *Personnalisé en cas d'échec*, il est possible de spécifier des options personnalisées de gestion des erreurs : soit ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié.
Correspond à l'expression régulière	Spécifiez une expression régulière à laquelle une valeur doit correspondre. Si vous cochez la case *Personnalisé en cas d'échec*, il est possible de spécifier des options personnalisées de gestion des erreurs : soit ignorer la valeur, définir une valeur spécifiée ou définir un message d'erreur spécifié.
Limitation du débit	
Ignorer les valeurs inchangées avec heartbeat	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 les secondes (minimum - 1 seconde). Des suffixes de temps peuvent être utilisés dans ce champ (par ex. 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 du débit peut être spécifiée pour un élément de découverte. Par ex. `1m`. Si un texte identique est transmis deux fois à cette règle dans un délai de 60 secondes, il sera ignoré. *Remarque* : la modification des prototypes d'élément ne réinitialise pas la limitation du débit. La limitation du débit n'est réinitialisée que lorsque les étapes de prétraitement sont modifiées.
Prometheus	
Prometheus vers JSON	Convertir les métriques Prometheus requises en JSON. Voir Contrôles Prometheus pour plus de détails.

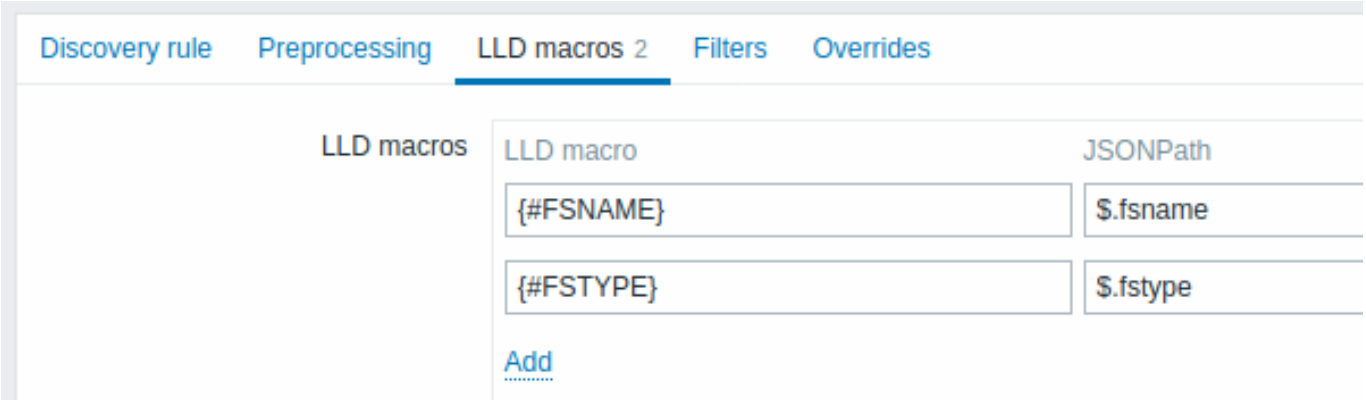
Notez que si la règle de découverte a été appliquée à l'hôte via un modèle, le contenu de cet onglet est en lecture seule.

Macros personnalisées

L'onglet **Macros LLD** permet de spécifier des macros personnalisées pour la découverte de bas niveau.

Les macros personnalisées sont utiles lorsque le JSON renvoyé ne contient pas déjà les macros requises définies. Ainsi, par exemple :

- La clé native `vfs.fs.discovery` pour la découverte des systèmes de fichiers renvoie un JSON avec certaines macros LLD prédéfinies telles que {#FSNAME}, {#FSTYPE}. Ces macros peuvent être utilisées directement dans les prototypes d'éléments et de déclencheurs (voir les sections suivantes de la page) ; il n'est pas nécessaire de définir des macros personnalisées ;
- L'élément d'agent `vfs.fs.get` renvoie également un JSON avec des données sur les systèmes de fichiers, mais sans macros LLD prédéfinies. Dans ce cas, vous pouvez définir vous-même les macros et les associer aux valeurs du JSON à l'aide de JSONPath :

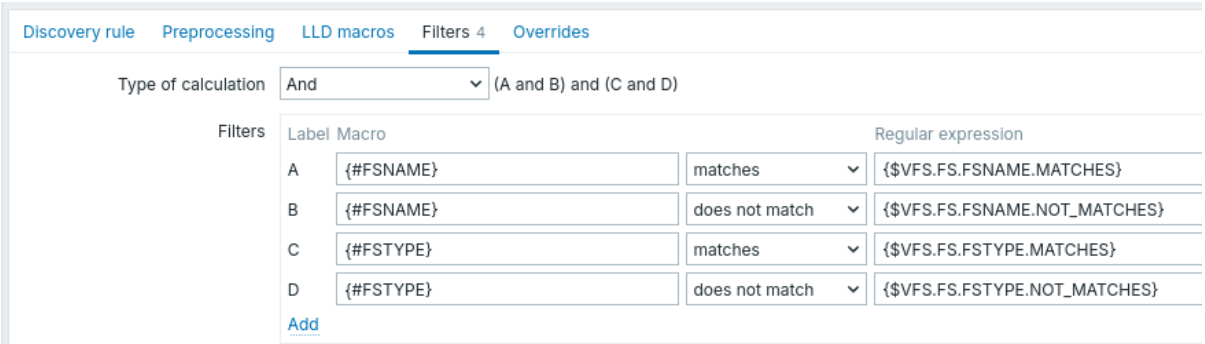


Les valeurs extraites peuvent être utilisées dans les éléments découverts, les déclencheurs, etc. Notez que les valeurs seront extraites du résultat de la découverte et de toutes les étapes de prétraitement effectuées jusque-là.

Parameter	Description
Macro LLD	Nom de la macro de découverte de bas niveau, en utilisant la syntaxe suivante : `{#MACRO}`.
JSONPath	Chemin utilisé pour extraire la valeur de la macro LLD à partir d'une ligne LLD, en utilisant la syntaxe JSONPath. Les valeurs extraites du JSON renvoyé sont utilisées pour remplacer les macros LLD dans les champs des prototypes d'éléments, de déclencheurs, etc. JSONPath peut être spécifié en utilisant la notation par point ou la notation entre crochets. La notation entre crochets doit être utilisée en présence de caractères spéciaux et d'Unicode, comme `$['unicode + special chars #1']['unicode + special chars #2']`. Par exemple, `$.foo` extraira "bar" et "baz" de ce JSON : `[{"foo":"bar"}, {"foo":"baz"}]` Notez que `$.foo` extraira également "bar" et "baz" de ce JSON : `{"data":[{"foo":"bar"}, {"foo":"baz"}]}` car un seul objet "data" est traité automatiquement (pour la compatibilité ascendante avec l'implémentation de la découverte de bas niveau dans les versions de Zabbix antérieures à 4.2).

Filtre

Un filtre peut être utilisé pour générer des éléments, déclencheurs et graphiques réels uniquement pour les entités qui correspondent aux critères. L'onglet **Filtres** contient des définitions de filtres de règle de découverte permettant de filtrer les valeurs de découverte :



Paramètre	Description
Type de calcul	Les options suivantes sont disponibles pour le calcul des filtres : **Et** - tous les filtres doivent être validés ; **Ou** - il suffit qu'un seul filtre soit validé ; **Et/Ou** - utilise *Et* avec des noms de macro différents et *Ou* avec le même nom de macro ; **Expression personnalisée** - offre la possibilité de définir un calcul personnalisé des filtres. La formule doit inclure tous les filtres de la liste. Limitée à 255 caractères.

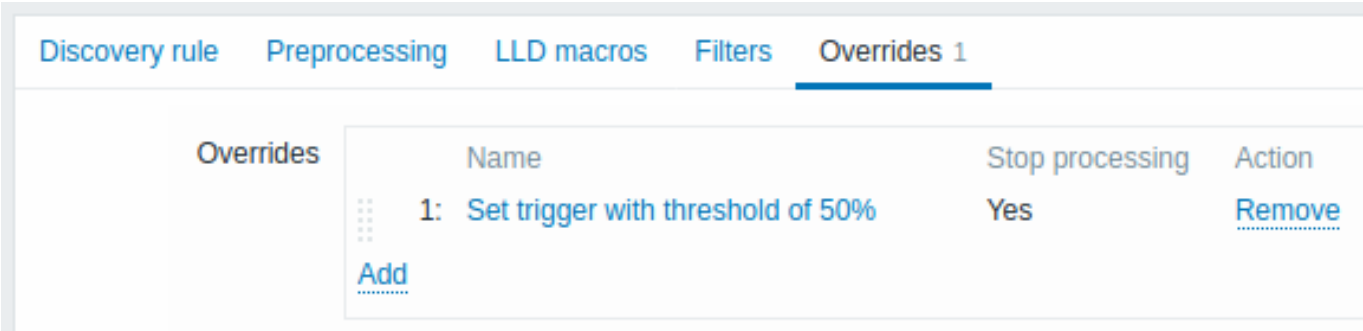
Paramètre	Description
Filtres	Les opérateurs de condition de filtre suivants sont disponibles : *correspond à*, *ne correspond pas à*, *existe*, *n'existe pas*. Les opérateurs *correspond à* et *ne correspond pas à* attendent une expression régulière compatible Perl (PCRE). Par exemple, si vous vous intéressez uniquement aux systèmes de fichiers C:, D: et E:, vous pouvez saisir {#FSNAME} dans le champ « Macro » et l'expression régulière "^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 l'agent Windows) à l'aide de la macro {#FSDRIVETYPE} (par exemple, "fixed"). Vous pouvez saisir une expression régulière ou référencer une expression régulière globale dans le champ « Expression régulière ». 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` 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 une condition « n'existe pas » est spécifiée pour cette macro. Un avertissement sera affiché si l'absence d'une macro affecte le résultat de l'expression. Par exemple, si {#B} est absente dans : {#A} matches 1 and {#B} matches 2 - affichera un avertissement {#A} matches 1 or {#B} matches 2 - aucun avertissement

Warning:
Une erreur ou une faute de frappe dans l'expression régulière utilisée dans la règle LLD (par exemple, une expression régulière incorrecte pour « File systems for discovery ») peut entraîner la suppression de milliers d'éléments de configuration, de valeurs historiques et d'événements pour de nombreux hôtes.

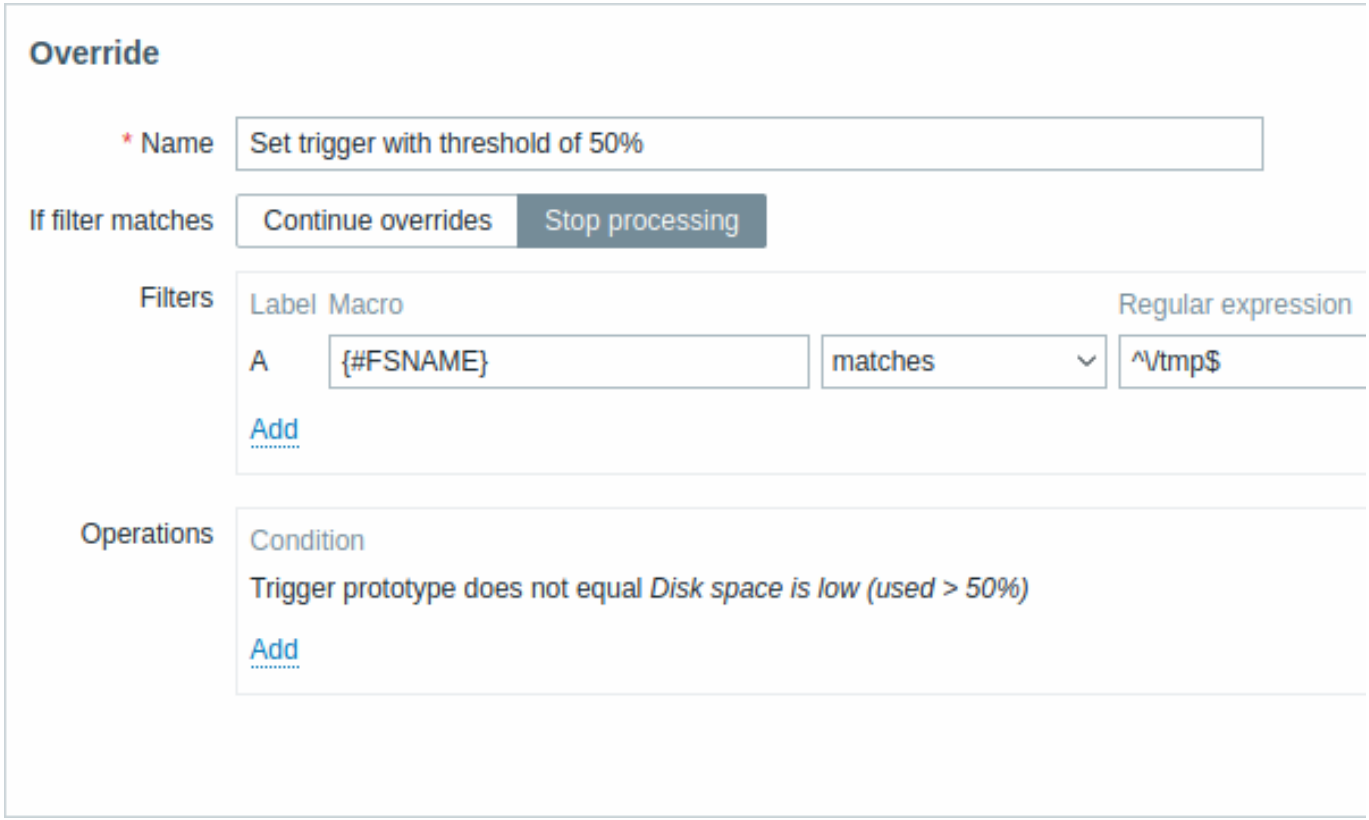
Attention:
La base de données Zabbix dans MySQL doit être créée en mode sensible à la casse afin que les noms de systèmes de fichiers qui ne diffèrent que par la casse soient correctement découverts.

Remplacement

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 des critères donnés.



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 le nom du remplacement. Une fenêtre contextuelle s'ouvrira, permettant de modifier les détails de la règle de remplacement.

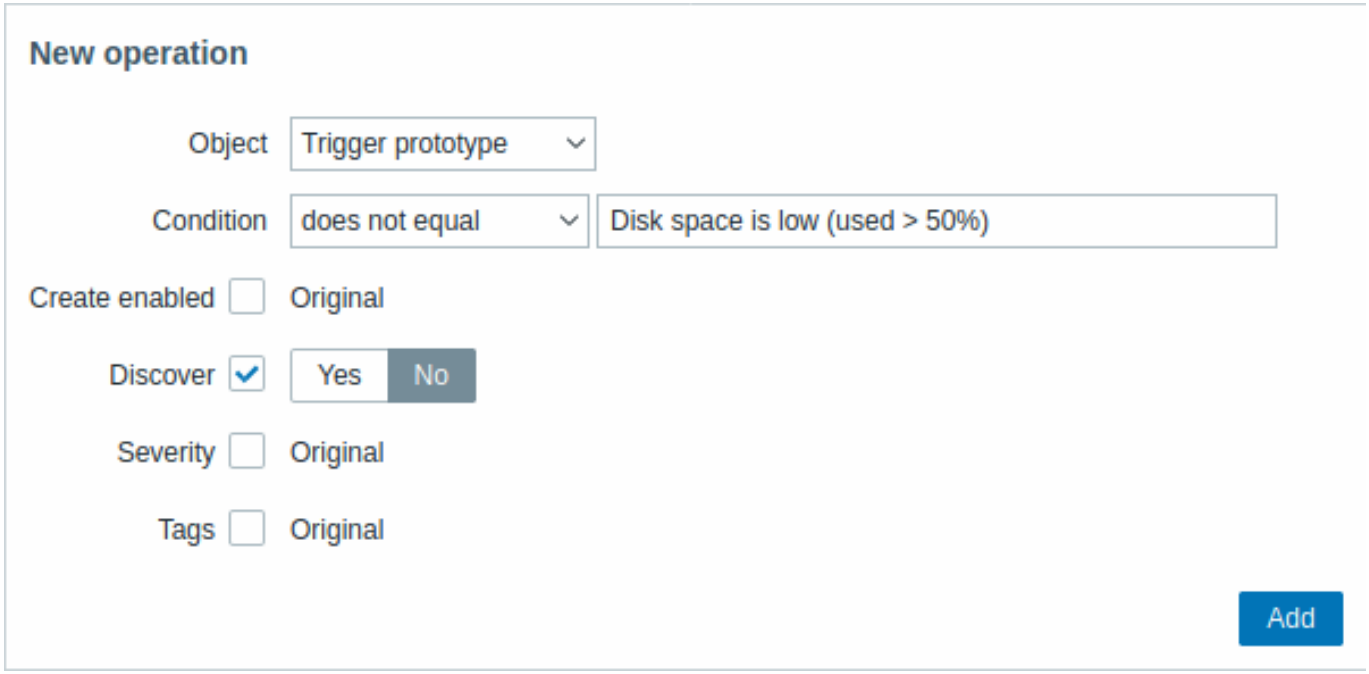


Tous les paramètres obligatoires sont marqués d'un astérisque rouge.

Paramètre	Description
Nom	Un nom de remplacement unique (par règle LLD).
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 *Disk space is low (used > 50%)*. **Actions** - des liens pour modifier et 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 dans laquelle vous pouvez modifier les détails de l'opération s'ouvrira.



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 : **égal à** - appliquer à ce prototype **différent de** - 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 à** - appliquer si le nom du prototype correspond à l'expression régulière **ne correspond pas à** - appliquer si le nom du prototype ne correspond pas à l'expression régulière
Motif	Une expression régulière ou une chaîne à rechercher.
Objet : *Prototype d'élément*	
Créer activé	Lorsque la case est cochée, les boutons apparaissent, permettant de remplacer les paramètres d'origine du prototype d'élément : *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, permettant de remplacer les paramètres d'origine du prototype d'élément : *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, permettant de définir un intervalle différent pour l'élément : *Délai* - intervalle de mise à jour de l'élément. Les macros utilisateur et les suffixes temporels (par ex. 30s, 1m, 2h, 1d) sont pris en charge. Doit être défini sur 0 si *Intervalle personnalisé* est utilisé. *Intervalle personnalisé* - cliquez sur Add pour spécifier des intervalles flexibles/de planification. Pour des informations détaillées, voir Intervalles personnalisés.

Paramètre	Description
Historique	Lorsque la case est cochée, les boutons apparaissent, permettant de définir une période de stockage de l'historique différente pour l'élément : *Ne pas stocker* - si sélectionné, l'historique ne sera pas stocké. *Stocker jusqu'à* - si sélectionné, un champ de saisie permettant de spécifier la période de stockage apparaîtra à droite. Les macros utilisateur et les macros LLD sont prises en charge.
Tendances	Lorsque la case est cochée, les boutons apparaissent, permettant de définir une période de stockage des tendances différente pour l'élément : *Ne pas stocker* - si sélectionné, les tendances ne seront pas stockées. *Stocker jusqu'à* - si sélectionné, un champ de saisie permettant de spécifier la période de stockage apparaîtra à droite. Les macros utilisateur et les macros LLD sont prises en charge.
Balises	Lorsque la case est cochée, un nouveau bloc apparaît, permettant de spécifier des paires balise-valeur. Ces balises seront ajoutées à celles spécifiées dans le prototype d'élément, même si les noms des balises correspondent.
Objet : *Prototype de déclencheur*	
Créer activé	Lorsque la case est cochée, les boutons apparaissent, permettant de remplacer les paramètres d'origine du prototype de déclencheur : *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, permettant de remplacer les paramètres d'origine du prototype de déclencheur : *Oui* - le déclencheur sera ajouté. *Non* - le déclencheur ne sera pas ajouté.
Sévérité	Lorsque la case est cochée, les boutons de sévérité du déclencheur apparaissent, permettant de modifier la sévérité du déclencheur.
Balises	Lorsque la case est cochée, un nouveau bloc apparaît, permettant de spécifier des paires balise-valeur. Ces balises seront ajoutées à celles spécifiées dans le prototype de déclencheur, même si les noms des balises correspondent.
Objet : *Prototype de graphique*	
Découvrir	Lorsque la case est cochée, les boutons apparaissent, permettant de remplacer les paramètres d'origine du prototype de graphique : *Oui* - le graphique sera ajouté. *Non* - le graphique ne sera pas ajouté.
Objet : *Prototype d'hôte*	
Créer activé	Lorsque la case est cochée, les boutons apparaissent, permettant de remplacer les paramètres d'origine du prototype d'hôte : *Oui* - l'hôte sera créé à l'état activé. *Non* - l'hôte sera créé à l'état désactivé.

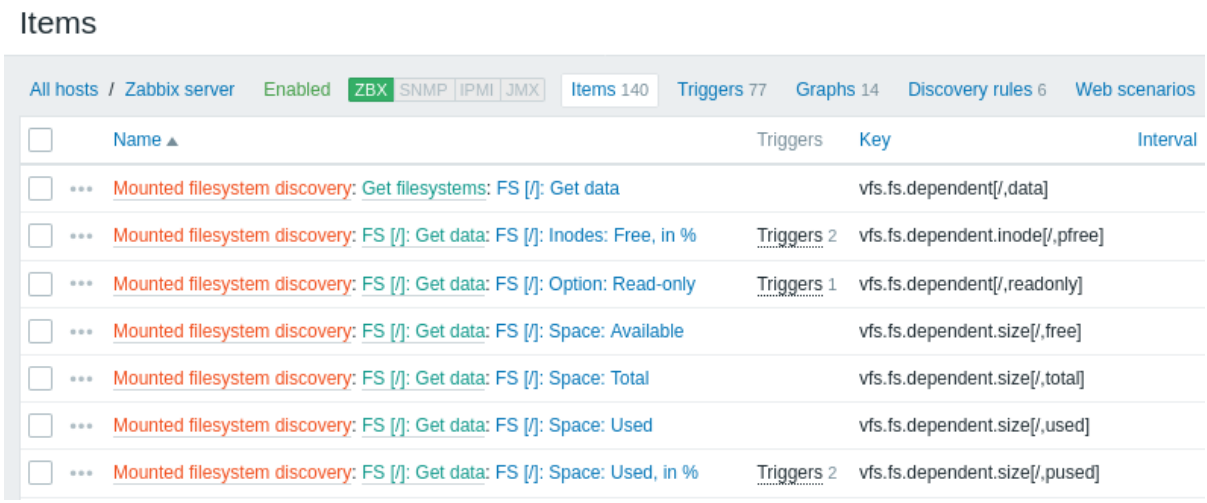
Paramètre	Description
Découvrir	Lorsque la case est cochée, les boutons apparaissent, permettant de remplacer les paramètres d'origine du prototype d'hôte : *Oui* - l'hôte sera découvert. *Non* - l'hôte ne sera pas découvert.
Lier des modèles	Lorsque la case est cochée, un champ de saisie permettant de spécifier des modèles apparaîtra. Commencez à saisir le nom du modèle ou cliquez sur *Sélectionner* à côté du champ et sélectionnez des modèles dans la liste de la fenêtre contextuelle. Les modèles de ce remplacement sont ajoutés à tous les modèles déjà liés au prototype d'hôte.
Balises	Lorsque la case est cochée, un nouveau bloc apparaît, permettant de spécifier des paires balise-valeur. Ces balises seront ajoutées à celles spécifiées dans le prototype d'hôte, même si les noms des balises correspondent.
Inventaire de l'hôte	Lorsque la case est cochée, les boutons apparaissent, permettant 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* - remplir automatiquement les données d'inventaire de l'hôte en fonction des métriques collectées.

Boutons du formulaire

Les boutons situés en bas du formulaire permettent d'effectuer plusieurs opérations.

Add	Ajouter une règle de découverte. Ce bouton n'est disponible que pour les nouvelles règles de découverte.
Update	Mettre à jour les propriétés d'une règle de découverte. Ce bouton n'est disponible que pour les règles de découverte existantes.
Clone	Créer une autre règle de découverte à partir des propriétés de la règle de découverte actuelle.
Execute now	Exécuter 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 la validité des paramètres) sans appliquer définitivement les 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.

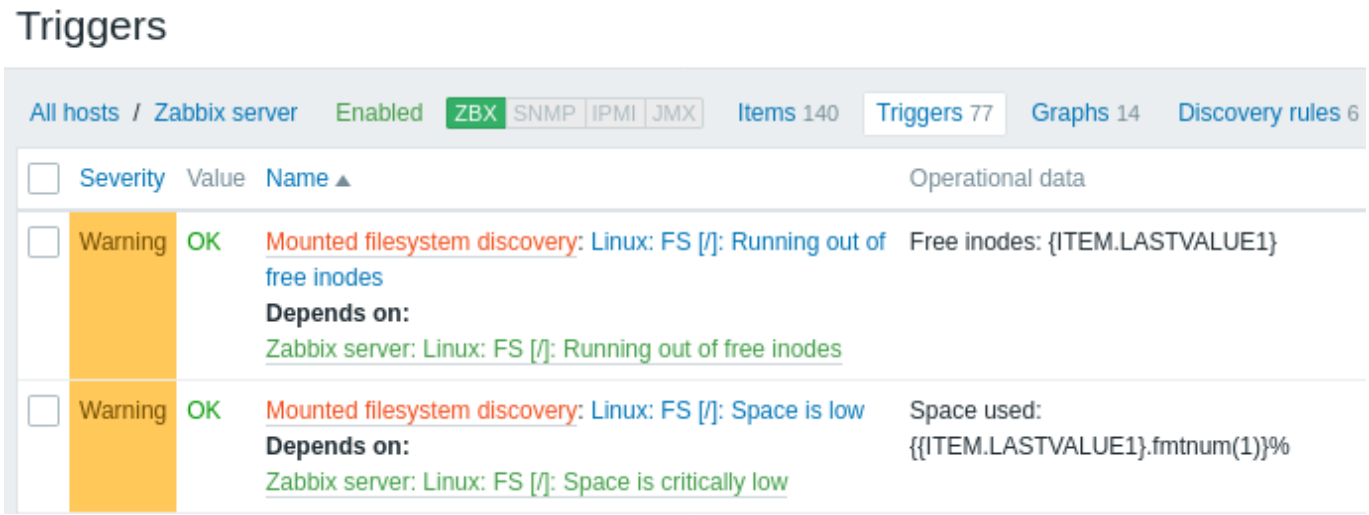
Les ressources perdues peuvent être automatiquement supprimées selon la valeur du paramètre *Delete lost resources*. Cela affecte les hôtes, groupes d'hôtes, éléments, déclencheurs et graphiques perdus.

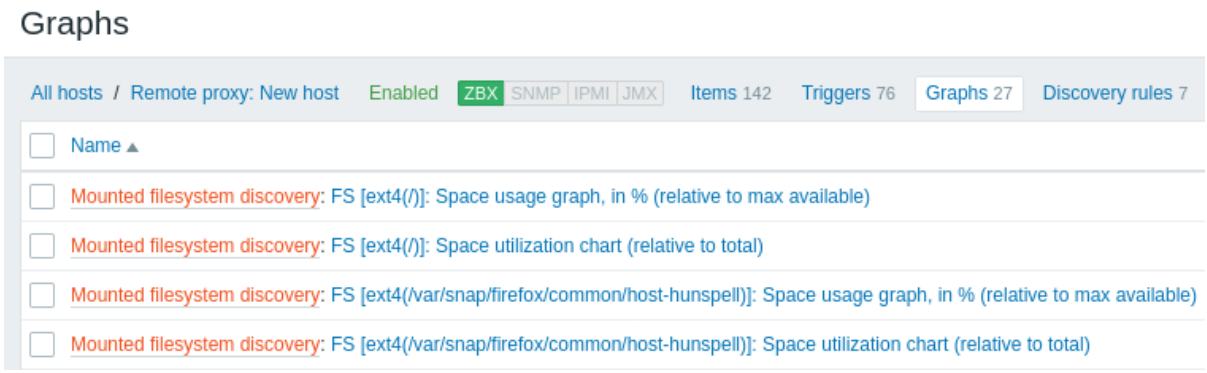
Lorsque des entités découvertes deviennent « Non découvertes désormais », un indicateur de durée de vie s'affiche dans la liste des entités. Placez le pointeur de la souris dessus et un message s'affichera indiquant les détails de son statut.



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 :

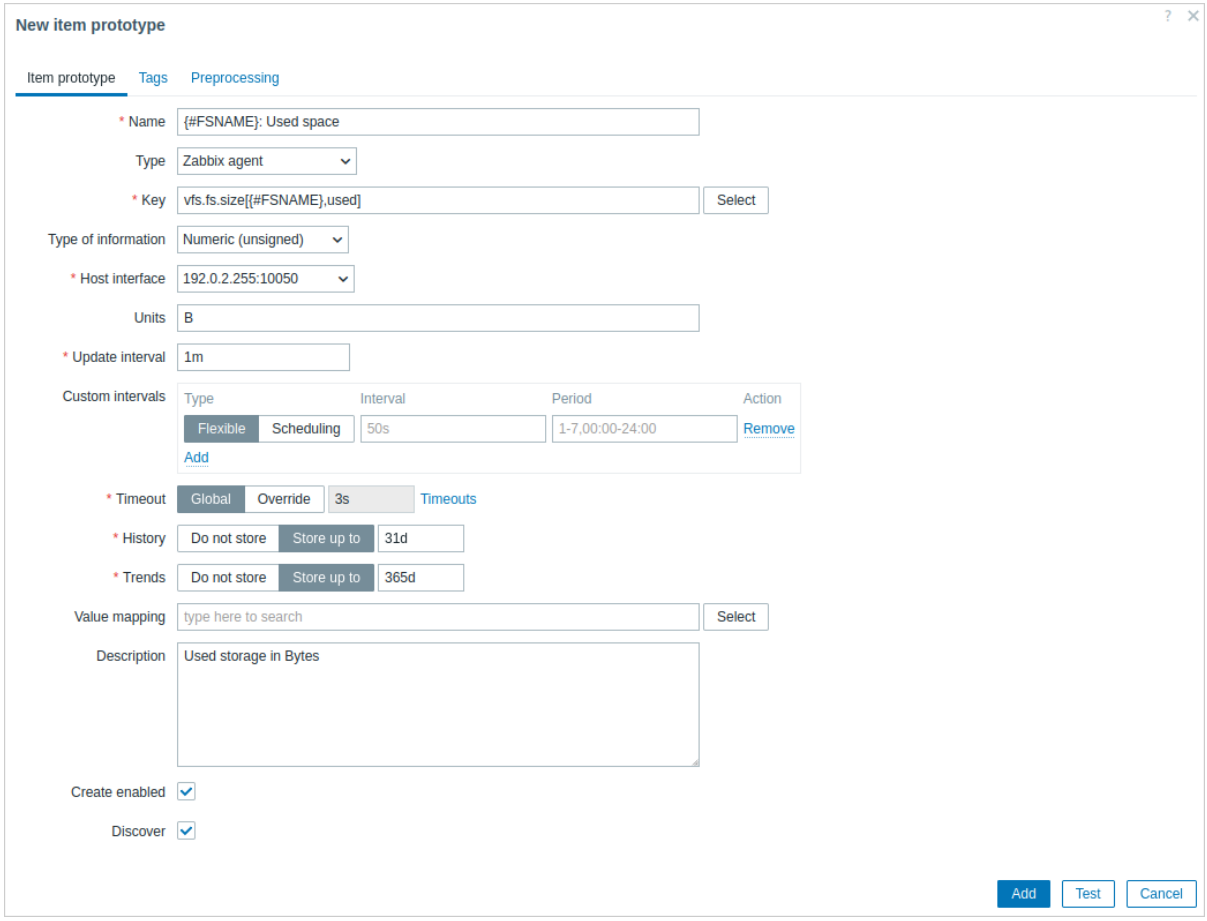
- découverte des interfaces réseau
- découverte des processeurs et des cœurs de processeur
- découverte des OID SNMP
- découverte des objets JMX ;
- découverte à l'aide de requêtes SQL ODBC
- découverte des services Windows
- découverte des interfaces d'hôte dans Zabbix

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.

1 Prototypes d'éléments

Une fois qu'une règle est créée, accédez aux éléments de cette règle et cliquez sur « Create item prototype » 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 que la découverte est traitée correctement. Lorsque la règle de découverte est traitée, cette macro sera remplacée par le système de fichiers découvert.



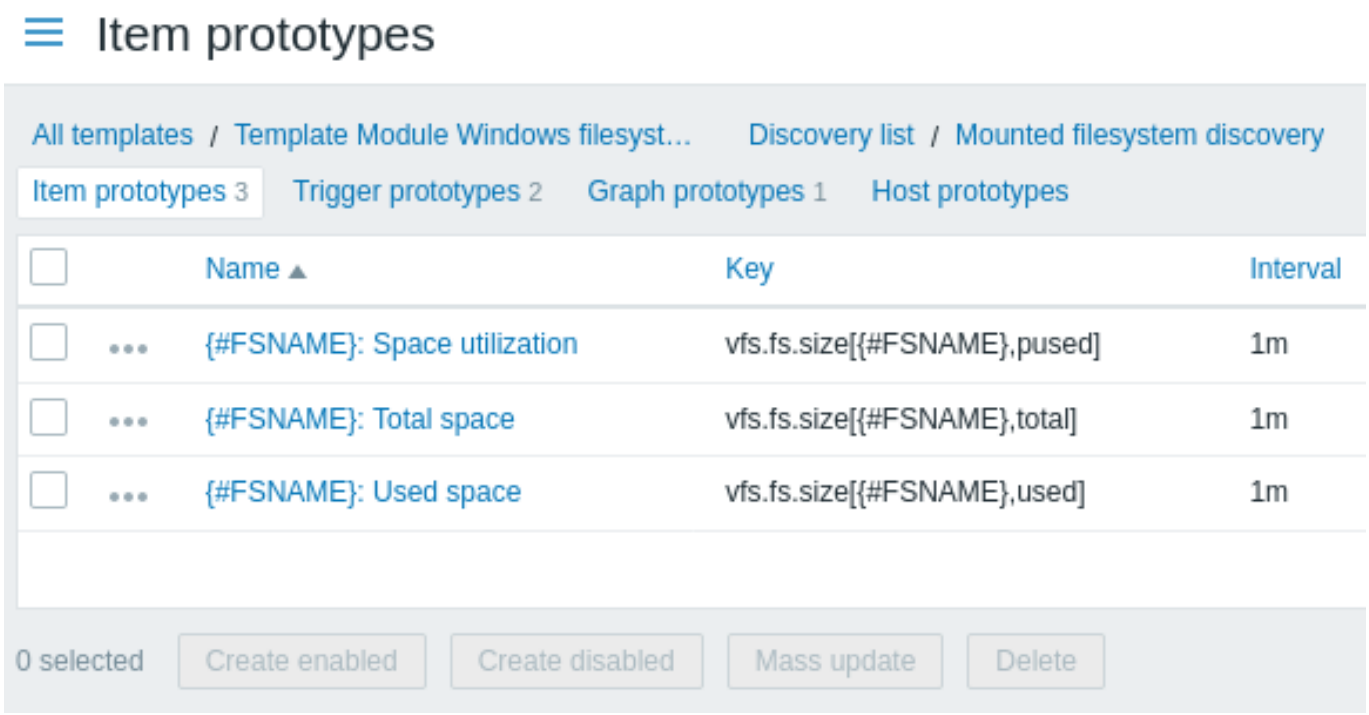
Les macros de découverte de bas niveau et les macros utilisateur sont prises en charge dans la configuration des prototypes d'éléments et dans les paramètres de prétraitement des valeurs d'élément. Notez que, lorsqu'elle est utilisée dans les intervalles de mise à jour, une seule macro doit remplir l'intégralité du champ. Plusieurs macros dans un même champ ou des macros mélangées avec du texte ne sont pas prises en charge.

Note:
L'échappement contextuel des macros de découverte de bas niveau est effectué pour une utilisation sûre dans les paramètres de prétraitement des expressions régulières et de XPath.

Attributs spécifiques aux prototypes d'éléments :

Parameter	Description
Create enabled	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.
Discover	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 :



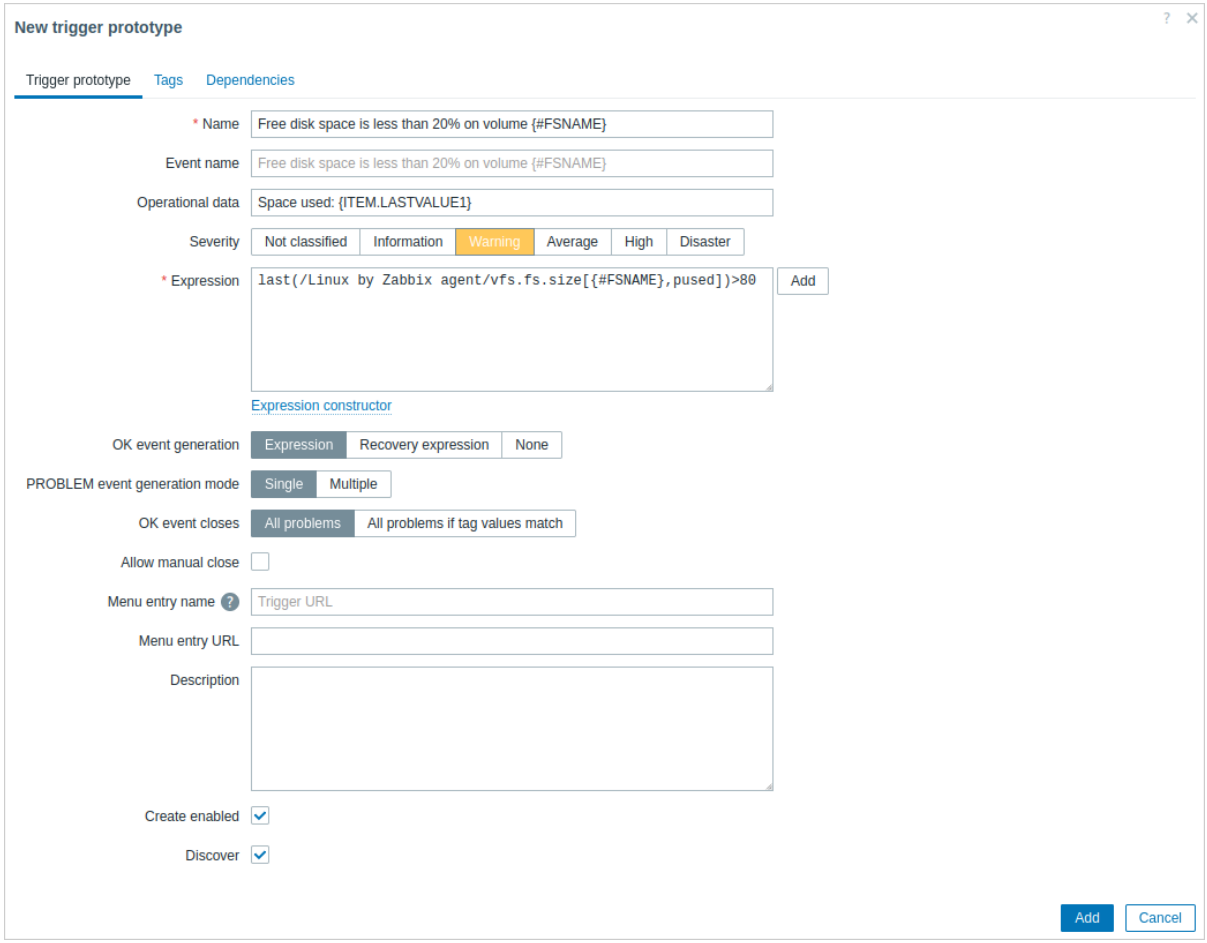
Cliquez sur l'icône à trois points pour ouvrir le menu du prototype d'élément spécifique avec les options suivantes :

- *Create trigger prototype* - créer un prototype de déclencheur basé sur ce prototype d'élément
- *Trigger prototypes* - cliquer pour voir une liste avec des liens vers les prototypes de déclencheurs déjà configurés de ce prototype d'élément
- *Create dependent item* - créer un élément dépendant pour ce prototype d'élément

L'option *Mass update* est disponible si vous souhaitez mettre à jour les propriétés de plusieurs prototypes d'éléments à la fois.

2 Prototypes de déclencheurs

Nous créons des prototypes de déclencheurs de manière similaire aux prototypes d'éléments :

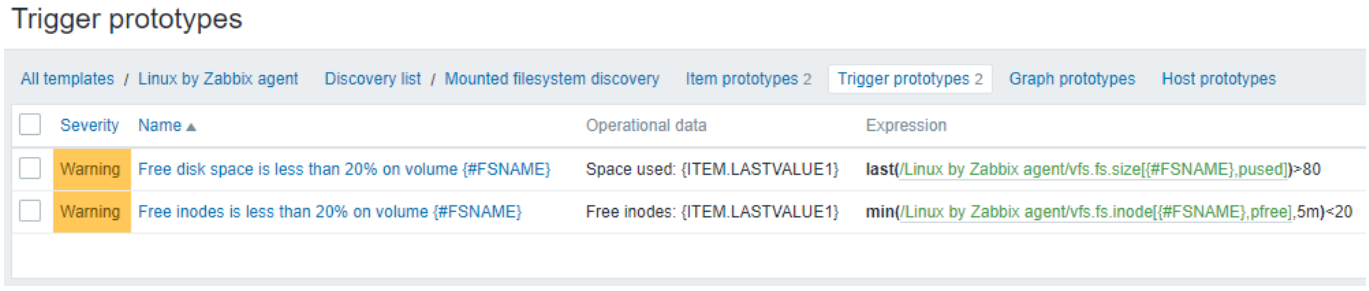


Attributs spécifiques aux prototypes de déclencheurs :

Paramètre	Description
Créer activé	Si cette option est cochée, le déclencheur sera ajouté à l'état activé. Si elle n'est pas cochée, le déclencheur sera ajouté à une entité découverte, mais à l'état désactivé.
Découvrir	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.

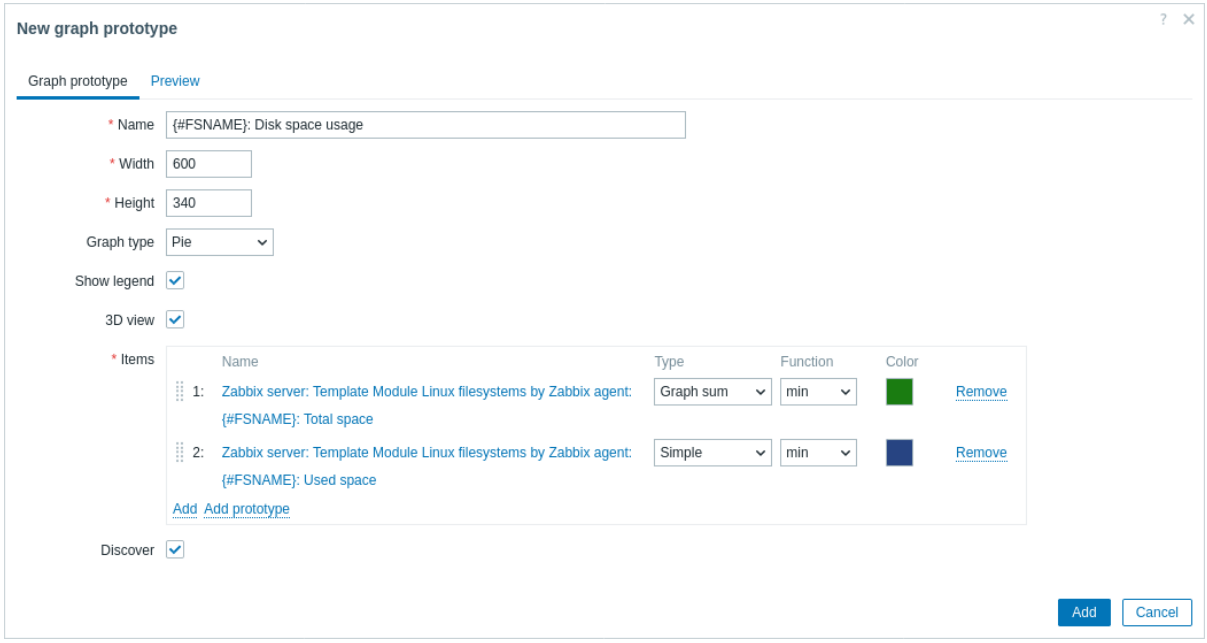
Lorsque de vrais déclencheurs sont créés à partir des prototypes, il peut être nécessaire de faire preuve de souplesse quant à la constante (« 20 » dans notre exemple) utilisée pour la comparaison dans l'expression. Voyez comment les macros utilisateur avec contexte peuvent être utiles pour obtenir cette flexibilité.

Vous pouvez définir des dépendances entre les prototypes de déclencheurs. Pour cela, accédez à l'onglet *Dépendances*. Un prototype de déclencheur peut dépendre d'un autre prototype de déclencheur de la même règle de découverte de bas niveau (LLD) ou d'un déclencheur standard. 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.



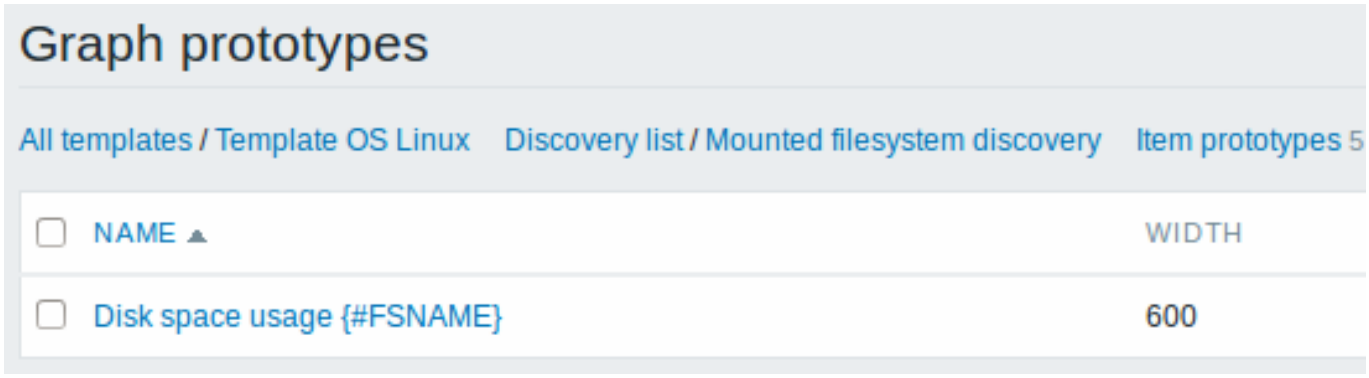
3 Prototypes de graphiques

Nous pouvons également créer des prototypes de graphiques :

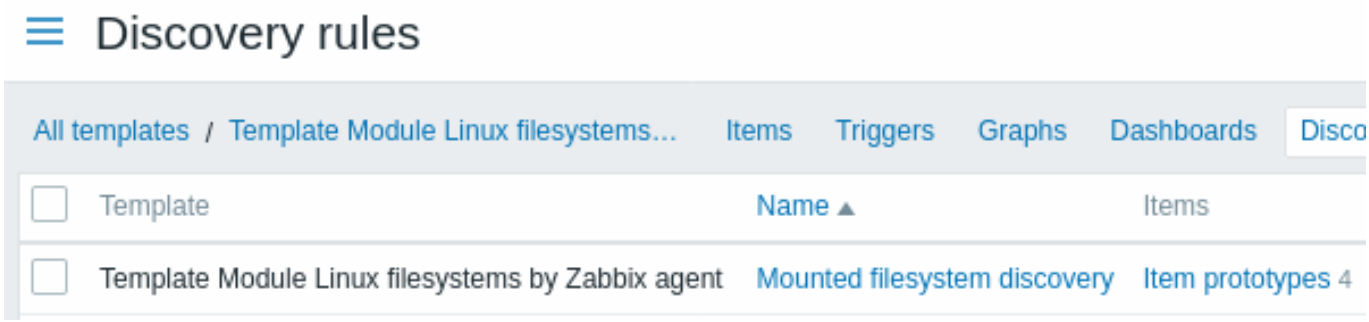


Attributs spécifiques aux prototypes de graphiques :

Paramètre	Description
Discover	Si cette option est cochée (par défaut), le graphique sera ajouté à une entité découverte. Si cette option 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 se présente comme indiqué ci-dessous. Elle comporte cinq prototypes d'éléments, deux prototypes de déclencheurs et un prototype de graphique.



4 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 dans les règles de découverte de bas niveau.

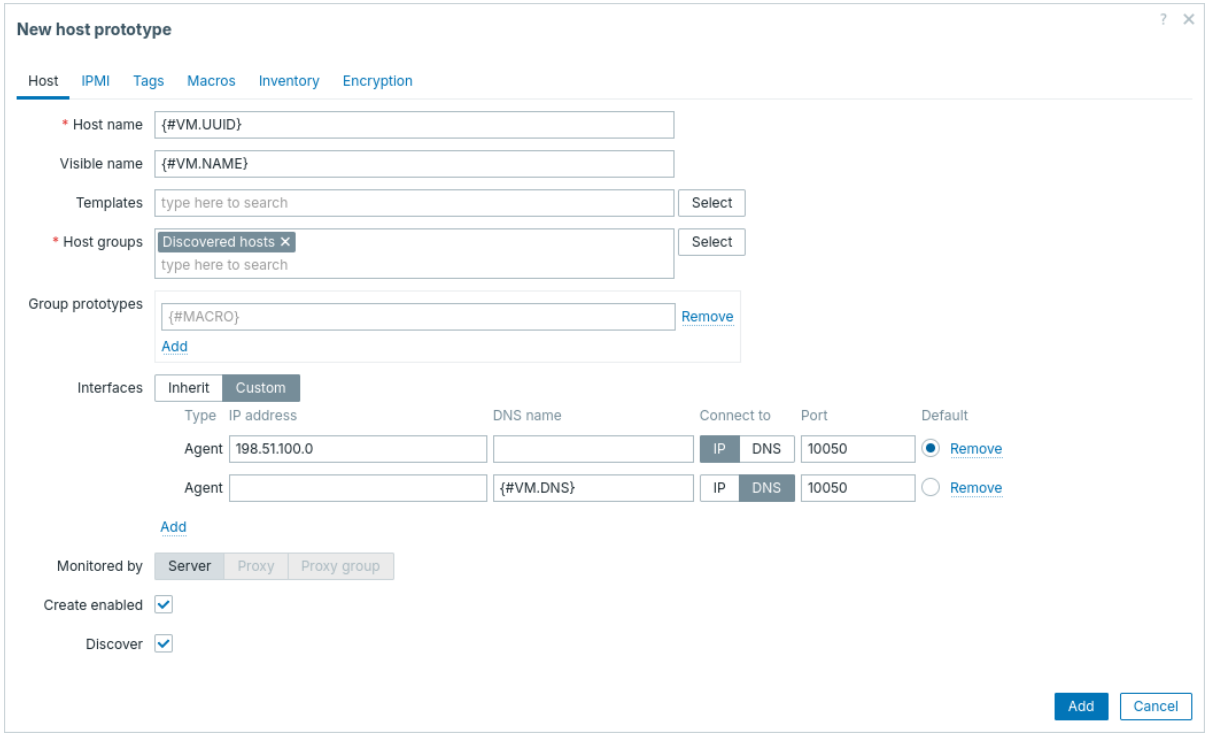
Pour créer un prototype d'hôte :

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 de saisie obligatoires sont marqués d'un astérisque rouge.

Paramètre	Description
Nom d'hôte	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.
Nom visible	Les macros de découverte de bas niveau sont prises en charge.
Prototypes de groupe	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 groupe 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. Cependant, 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'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.
Surveillé par	Ce champ est synchronisé automatiquement avec le champ correspondant dans la configuration de l'hôte.
Créer activé	Définissez l'état des hôtes découverts ; si cette option n'est pas cochée, les hôtes seront créés comme désactivés.
Découvrir	Définissez si des hôtes seront créés à partir du prototype d'hôte ; si cette option n'est pas cochée, aucun hôte ne sera créé 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 tags et les valeurs des 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 *Supervision des machines virtuelles*.

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).

5 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 de créer une découverte multi-niveaux d'objets avec leurs propres éléments de données, déclencheurs, etc. Par exemple, vous pouvez vouloir découvrir toutes les instances de base de données sur un serveur de base de données, puis découvrir les tablespaces pour chaque instance, puis découvrir les tables pour chaque tablespace.

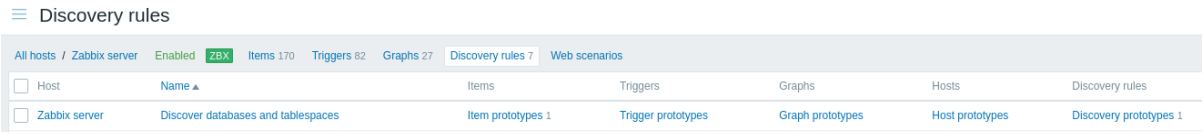
Les prototypes de découverte disposent de leurs propres prototypes d'élément de données, de déclencheur, de graphique, d'hôte et de découverte. Un prototype de découverte imbriqué utilisera la même valeur JSON que la règle parente si vous spécifiez un type *Imbriqué*.

Le nombre de niveaux d'imbrication des prototypes de découverte est illimité.

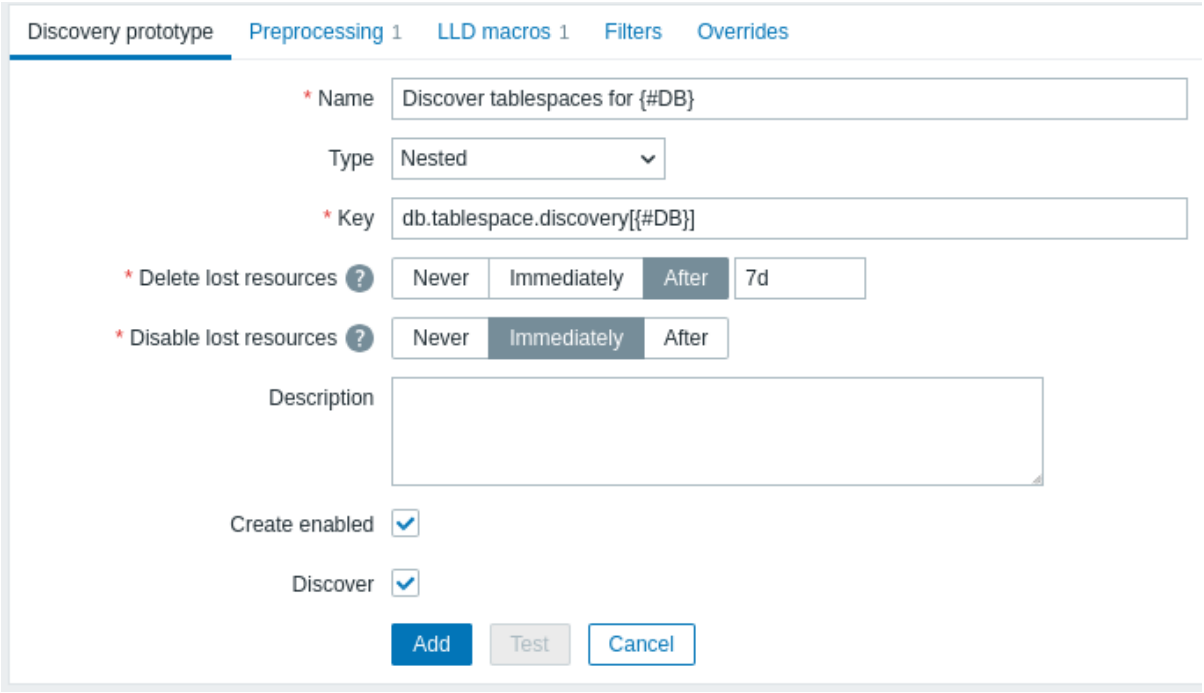
Configuration

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 communs avec la découverte de bas niveau classique.

Si vous sélectionnez « Nested » comme *Type* dans le formulaire de prototype de découverte ouvert, des règles de découverte (à partir du prototype de découverte) sont alors 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 [<objet A>, <objet B>] et qu'il existe un prototype de règle de découverte imbriquée, alors deux règles de découverte seront générées respectivement à partir des données de l'objet A et de l'objet B.

Dans ce cas, le prototype de découverte est activé en même temps que la règle parente. La règle imbriquée peut ainsi utiliser le prétraitement pour travailler sur une autre « tranche » des mêmes données, déjà acquises par le parent.

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 : imbriqué, 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 dans ce modèle un élément 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 *Prétraitement* 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

6 Remarques sur la découverte de bas niveau

Utilisation de macros LLD dans des contextes de macros utilisateur

Les macros LLD peuvent être utilisées dans le contexte de la macro utilisateur, par exemple, dans les prototypes de déclencheur.

Plusieurs règles LLD 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 d'agent Alias, ce qui 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.

7 Règles de découverte

Veuillez utiliser la barre latérale pour voir des exemples de configuration des règles de découverte pour divers cas.

1 Découverte des systèmes de fichiers montés

Vue d'ensemble

Il est possible de découvrir les systèmes de fichiers montés et leurs propriétés :

- nom du point de montage
- type de système de fichiers
- taille du système de fichiers
- statistiques des inodes
- options de montage

Pour ce faire, vous pouvez utiliser une combinaison de :

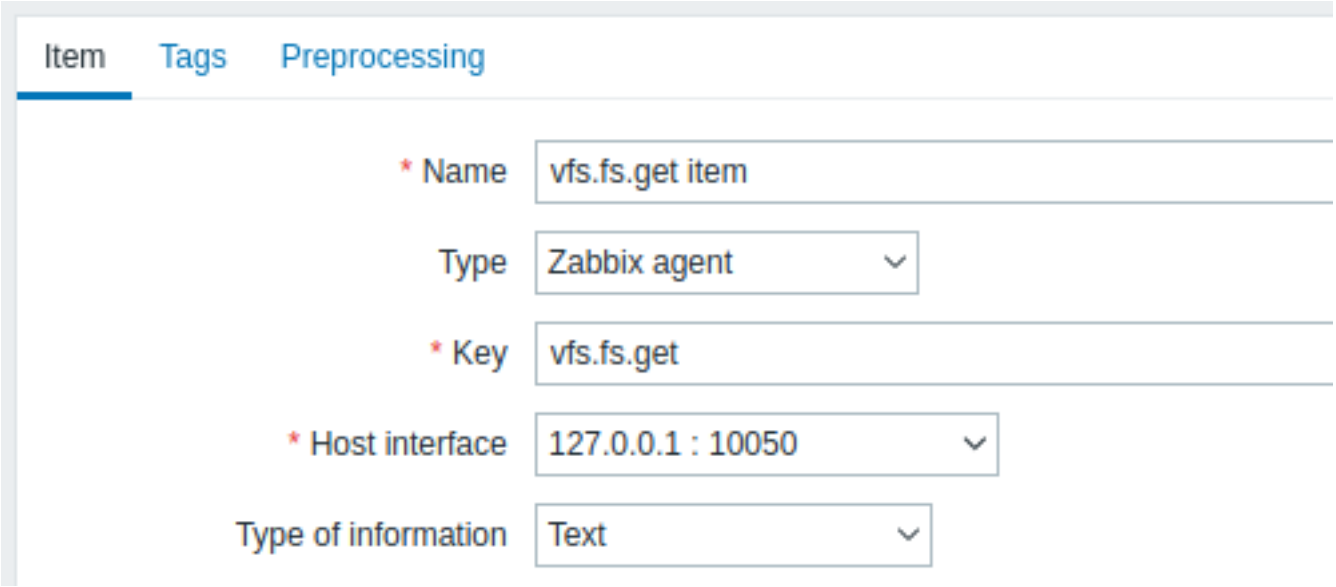
- l'élément d'agent `vfs.fs.get` comme élément maître
- une règle de découverte de bas niveau dépendante et des prototypes d'éléments

Configuration

Élément maître

Créez un élément d'agent Zabbix en utilisant la clé suivante :

`vfs.fs.get`



Définissez le type d'information sur « Texte » pour les données JSON potentiellement volumineuses.

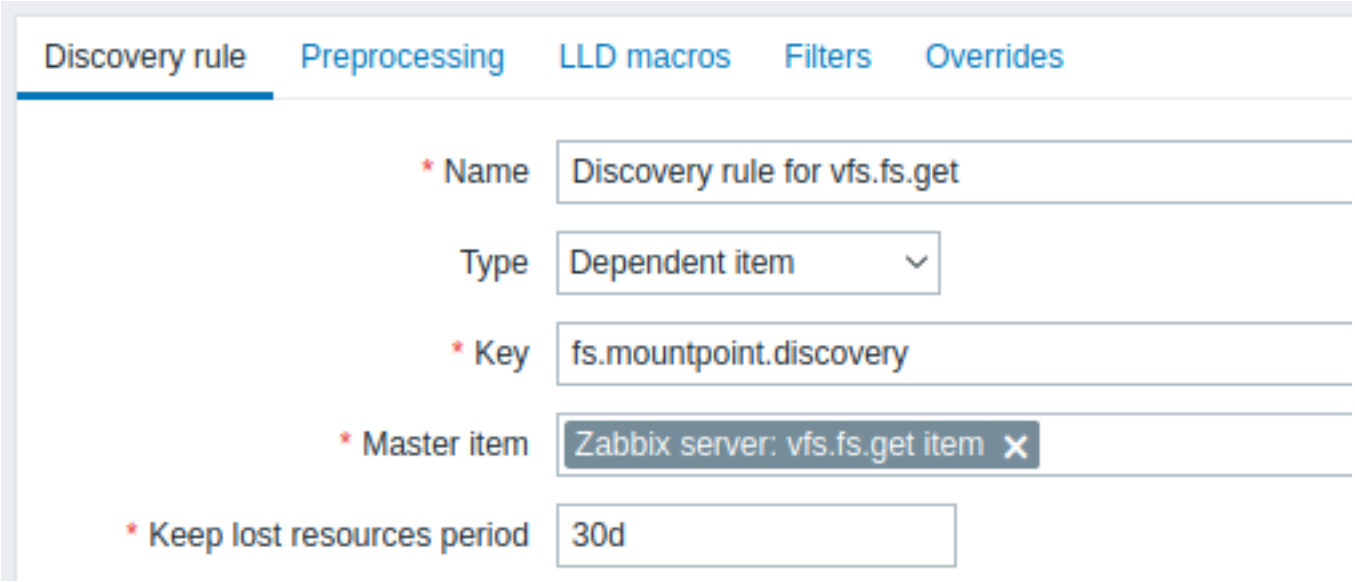
Les données renvoyées par cet élément contiendront quelque chose de similaire à ce qui suit pour un système de fichiers monté :


```
[
  {
    "fsname": "/",
    "fstype": "ext4",
    "bytes": {
      "total": 249405239296,
      "free": 24069537792,
      "used": 212595294208,
      "pfree": 10.170306,
      "pused": 89.829694
    },
    "inodes": {
      "total": 15532032,
      "free": 12656665,
      "used": 2875367,
```

```
      "pfree": 81.487503,
      "pused": 18.512497
    },
    "options": "rw,noatime,errors=remount-ro"
  }
]
```

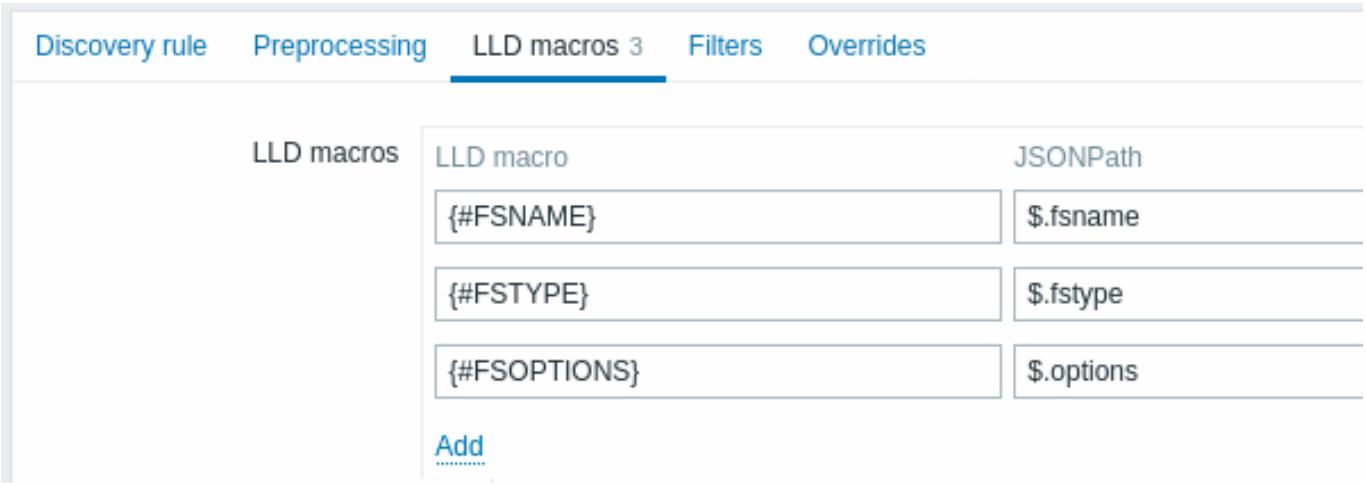
Règle LLD dépendante

Créez une règle de découverte de bas niveau de type « Élément dépendant » :

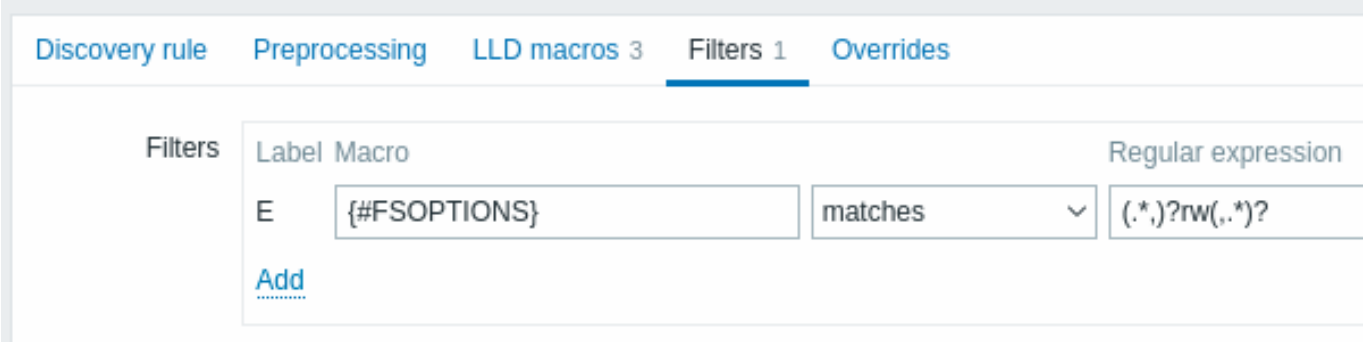


Comme élément maître, sélectionnez l'élément `vfs.fs.get` que nous avons créé.

Dans l'onglet « Macros LLD », définissez des macros personnalisées avec le JSONPath correspondant :

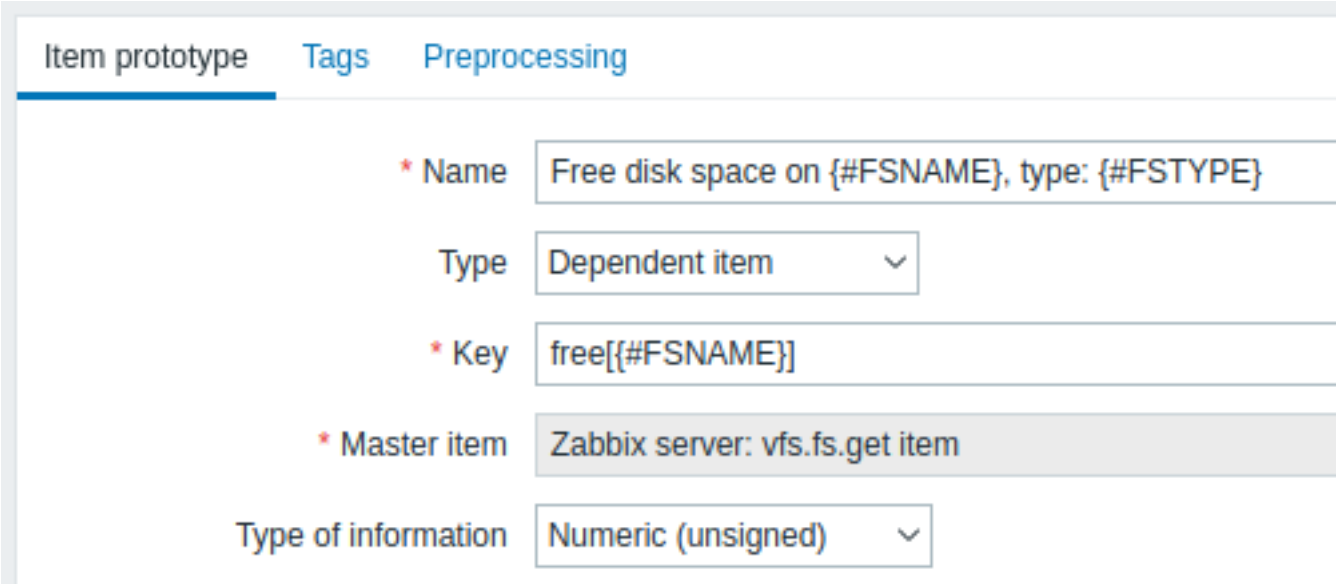


Dans l'onglet « Filtres », vous pouvez ajouter une expression régulière qui filtre uniquement les systèmes de fichiers **lecture-écriture** :



Prototype d'élément dépendant

Créez un prototype d'article avec le type "Élément dépendant" dans cette règle LLD. Comme élément maître pour ce prototype, sélectionnez l'élément `vfs.fs.get` que nous avons créé.



Notez l'utilisation de macros personnalisées dans le nom et la clé du prototype d'élément :

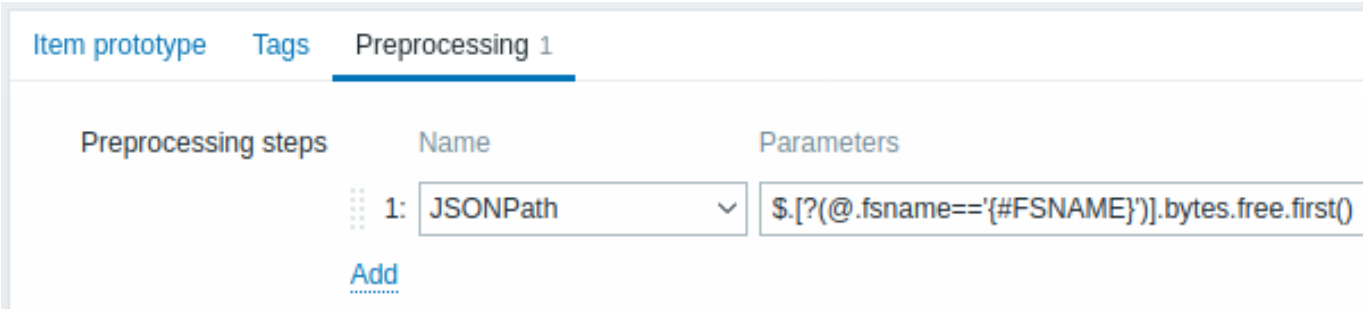
- *Nom*: Free disk space on {#FSNAME}, type: {#FSTYPE}
- *Clé*: Free[{#FSNAME}]

Comme type d'information, utilisez :

- *Numérique (non signé)* pour les métriques telles que 'free', 'total', 'used'
- *Numérique (flottant)* pour les métriques telles que 'pfree', 'pused' (pourcentage)

Dans l'onglet "Prétraitement" du prototype d'élément, sélectionnez JSONPath et utilisez l'expression JSONPath suivante comme paramètre :

```
$.[?(@.fsname=='{#FSNAME}')].bytes.free.first()
```



Lorsque la découverte démarre, un élément par point de montage sera créé. Cet élément renverra le nombre d'octets libres pour le point de montage donné.

2 Découverte des interfaces réseau

De la même manière que les systèmes de fichiers sont découverts, il est également possible de découvrir les interfaces réseau.

Clé d'élément

La clé d'élément à utiliser dans la règle de découverte est

`net.if.discovery`

Macros supportées

Vous pouvez utiliser la macro {#IFNAME} dans le filtre de règle de découverte et les prototypes d'éléments, de déclencheurs et de graphiques.

Exemples de prototypes d'articles que vous pourriez souhaiter créer en fonction de "net.if.discovery" :

- "net.if.in[{#IFNAME},bytes]",
- "net.if.out[{#IFNAME},bytes]".

Notez que sous Windows {#IFGUID} est également renvoyé.

3 Découverte des processeurs et des cœurs

De la même manière que les systèmes de fichiers sont découverts, il est également possible de découvrir des CPU et des cœurs de CPU.

Clé d'élément

La clé d'élément à utiliser dans la règle de découverte est

`system.cpu.discovery`

Macros prises en charge

Cette clé de découverte renvoie deux macros - {#CPU.NUMBER} et {#CPU.STATUS} identifiant respectivement le numéro d'ordre et l'état du CPU. Elle énumère les processeurs de 0 à N - 1, où N correspond au nombre total de processeurs.

```
[
    {
        "{#CPU.NUMBER}": 0,
        "{#CPU.STATUS}": "online"
    },
    {
        "{#CPU.NUMBER}": 1,
        "{#CPU.STATUS}": "offline"
    },
    {
        "{#CPU.NUMBER}": 2,
        "{#CPU.STATUS}": "unknown" /* "unknown" n'existe que sous Windows */
    },
    {
        "{#CPU.NUMBER}": 3,
        "{#CPU.STATUS}": "online"
    }
]
```

Notez qu'il n'est pas possible d'établir une distinction claire entre les processeurs réels, physiques, les cœurs et les hyperthreads. {#CPU.STATUS} sur les systèmes Linux, UNIX et BSD renvoie l'état du processeur, qui peut être soit "online", soit "offline". Sur les systèmes Windows, cette même macro peut représenter une troisième valeur - "unknown" - qui indique qu'un processeur a été détecté, mais qu'aucune information n'a encore été collectée à son sujet.

La découverte des CPU s'appuie sur le processus collecteur de l'agent afin de rester cohérente avec les données fournies par le collecteur et d'économiser des ressources lors de l'obtention des données. Cela a pour effet que cette clé d'élément ne fonctionne pas avec l'option de ligne de commande de test (-t) du binaire de l'agent, qui renverra un état NOT_SUPPORTED ainsi qu'un message indiquant que le processus collecteur n'a pas été démarré.

Les prototypes d'élément pouvant être créés sur la base de la découverte des CPU incluent, par exemple :

- `system.cpu.util[{#CPU.NUMBER},<type>,<mode>]`
- `system.hw.cpu[{#CPU.NUMBER},<info>]`

Pour une description détaillée des clés d'élément, voir clés d'élément de l'agent Zabbix.

Remarques sur les éléments de compteur de performance découverts sur les systèmes NUMA

Certains compteurs de performance Windows peuvent ne pas être disponibles pour certains processeurs logiques sur les systèmes NUMA.

Par exemple, les éléments découverts avec un prototype d'élément utilisant la clé ci-dessous peuvent fonctionner uniquement pour le premier nœud NUMA. Les éléments des autres nœuds NUMA peuvent être dans un état non pris en charge.

`perf_counter[\Processor({#CPU.NUMBER})\% Processor Time,60]`

De plus, une règle de découverte avec la clé "system.cpu.discovery" ne fournit aucune valeur de macro de découverte de bas niveau à substituer dans des compteurs de performance tels que :

`perf_counter["\Processor Information(<NUMA node index>,<CPU index in NUMA node>)\% Processor Time",60]`

Alors que les éléments system.cpu.util s'appuient sur différents compteurs de performance pour surveiller le pourcentage d'utilisation du processeur en fonction du nombre de groupes de processeurs et du nombre de processeurs. Il existe également un bogue rare lié aux groupes de processeurs sous Windows.

4 Découverte des OID SNMP

Aperçu

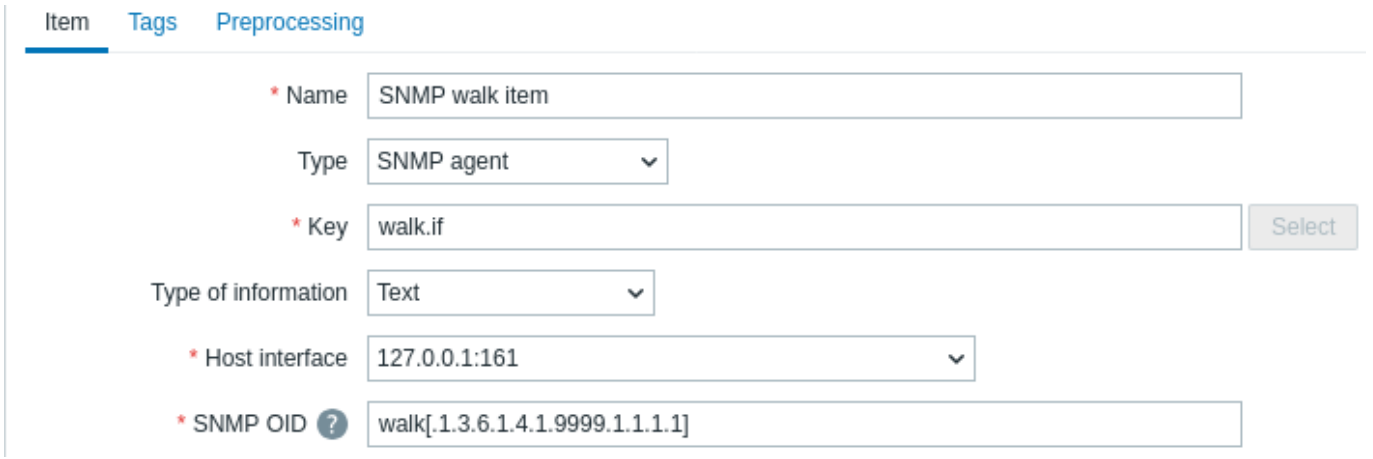
Dans cette section, nous allons effectuer une découverte de bas niveau sur un périphérique SNMP.

Cette méthode de découverte des OID SNMP est prise en charge depuis Zabbix server/proxy 6.4.

Exemple de configuration

1. Créez un élément d'agent SNMP avec une clé telle que :

```
walk[.1.3.6.1.4.1.9999.1.1.1.1]
```



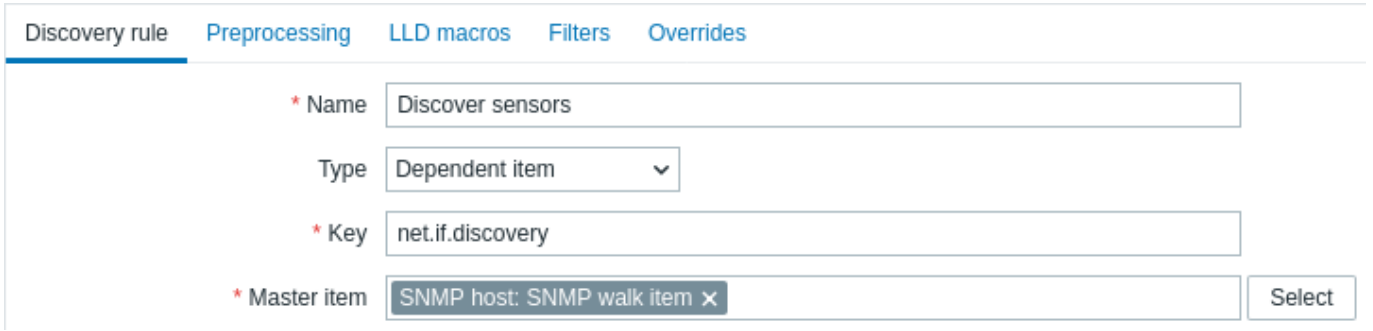
Cet élément effectue un seul parcours de table SNMP et renvoie toutes les entrées de la table en une seule requête, dans un format correspondant à la sortie de l'utilitaire `snmpwalk` avec les options de formatage `-Oe -Ot -On`.

Il renverra la valeur de texte multiligne suivante :

```
.1.3.6.1.4.1.9999.1.1.1.1.1.1 = STRING: "Temperature Sensor"
.1.3.6.1.4.1.9999.1.1.1.1.2.1 = STRING: "temp"
.1.3.6.1.4.1.9999.1.1.1.1.3.1 = 100
.1.3.6.1.4.1.9999.1.1.1.1.1.2 = STRING: "Humidity Sensor"
.1.3.6.1.4.1.9999.1.1.1.1.2.2 = STRING: "humidity"
.1.3.6.1.4.1.9999.1.1.1.1.3.2 = 200
```

2. Créez une règle de découverte :

- Dans le champ *Name*, saisissez un nom descriptif pour la règle de découverte (par exemple, "Découvrir les capteurs").
- Dans le champ *Type*, sélectionnez "Dependent item".
- Dans le champ *Key*, saisissez une clé descriptive (par exemple, "net.if.discovery").
- Dans le champ *Master item*, sélectionnez "SNMP walk item".



3. Dans l'onglet *Preprocessing*, ajoutez une étape de prétraitement avec "SNMP walk to JSON" dans la liste déroulante *Name* avec 3 paramètres :

- *Field name* : "{#SENSORNAME}" ; *OID prefix* : ".1.3.6.1.4.1.9999.1.1.1.1.1" : *Format* : "Unchanged".
- *Field name* : "{#SENSORTYPE}" ; *OID prefix* : ".1.3.6.1.4.1.9999.1.1.1.1.2" : *Format* : "Unchanged".

- *Field name* : "{#SENSORVALUE}" ; *OID prefix* : ".1.3.6.1.4.1.9999.1.1.1.1.3" : *Format* : "Unchanged".

Après le prétraitement, la règle de découverte renvoie un tableau JSON d'ensembles de macros.

Par exemple :

```
[
    {
        "{#SNMPINDEX}": "1",
        "{#SENSORNAME}": "Temperature Sensor",
        "{#SENSORTYPE}": "temp",
        "{#SENSORVALUE}": "100"
    },
    {
        "{#SNMPINDEX}": "2",
        "{#SENSORNAME}": "Humidity Sensor",
        "{#SENSORTYPE}": "humidity",
        "{#SENSORVALUE}": "200"
    }
]
```

Chaque objet représente un capteur découvert et fournit des macros telles que {#SNMPINDEX}, {#SENSORNAME}, {#SENSORTYPE} et {#SENSORVALUE}.

Elles sont regroupées par l'index SNMP, qui est le suffixe numérique à la fin de chaque OID (par exemple, .1, .2) — cet index identifie de manière unique chaque ligne de la table SNMP et est automatiquement extrait sous la forme {#SNMPINDEX}.

4. Sous la règle de découverte, créez un ou plusieurs prototypes d'élément (avec la règle de découverte comme élément maître).

Par exemple, un élément dépendant pour la valeur du capteur :

- Dans le champ *Name*, saisissez "Sensor {#SNMPINDEX}: {#SENSORNAME}".
- Dans le champ *Type*, sélectionnez "Dependent item".
- Dans le champ *Key*, saisissez "sensor.value[{#SNMPINDEX}]".
- Dans le champ *Master item*, sélectionnez "SNMP walk item".



Dans l'onglet *Preprocessing*, ajoutez une étape de prétraitement avec le nom "SNMP walk value" et l'OID ".1.3.6.1.4.1.9999.1.1.1.1.3.{#SNMPINDEX}" dans le champ *Parameter*. *Format* : "Unchanged".

Les éléments suivants seront découverts :

Name	Key	OID à partir duquel la valeur est extraite	Valeur de l'élément
Sensor 1: Temperature Sensor	sensor.value[1]	.1.3.6.1.4.1.9999.1.1.1.1.3.1	100
Sensor 2: Humidity Sensor	sensor.value[2]	.1.3.6.1.4.1.9999.1.1.1.1.3.2	200

Lorsque la règle de découverte s'exécute, des éléments tels que `sensor.value[1]`, `sensor.value[2]` sont créés.

Chaque élément dépendant extrait sa valeur du résultat du parcours SNMP de l'élément maître à l'aide du prétraitement, sans effectuer lui-même de requêtes SNMP distinctes.

5. Référencez les prototypes d'éléments dépendants dans les prototypes de déclencheur en utilisant les mêmes macros de la règle de découverte. Exemple :

```
{Template_Sensor:sensor.value[{#SNMPINDEX}].last()} > 75
```

Cela produit un déclencheur pour chaque capteur découvert (par exemple, sensor.value[1], sensor.value[2]) et se déclenche si la dernière valeur (température ou humidité) dépasse 75.

6. Incluez des éléments dépendants pour chaque entité découverte. Exemple de clé d'élément de graphique :

```
sensor.value[{#SNMPINDEX}]
```

Un graphique est créé par {#SNMPINDEX}, traçant la température et l'humidité au fil du temps.

Cette configuration n'effectue qu'une seule requête de parcours SNMP par cycle d'interrogation, quel que soit le nombre d'éléments découverts. Tous les éléments dépendants extraient leurs valeurs du résultat du parcours SNMP maître à l'aide du prétraitement, ce qui réduit considérablement le trafic et la charge SNMP.

Index dynamiques avec walk[]

Les index dynamiques (par exemple, les index d'interface) peuvent changer lorsque le matériel est reconfiguré. Pour prendre en charge ce comportement, une règle de découverte maître SNMP walk est créée avec une clé telle que :

```
walk[1.3.6.1.2.1.2.2.1.10]
```

Après le prétraitement SNMP walk vers JSON, le résultat peut ressembler à :

```
[
    {
        "{#SNMPINDEX}": "2",
        "{#VALUE}": "123456"
    },
    {
        "{#SNMPINDEX}": "3",
        "{#VALUE}": "654321"
    }
]
```

Un prototype d'élément dépendant utilise la macro {#SNMPINDEX} pour construire la clé :

```
net.if.in[{#SNMPINDEX}]
```

Le prétraitement de ce prototype inclut le nom « SNMP walk value » avec l'OID « 1.3.6.1.2.1.2.2.1.10.{#SNMPINDEX} » dans le champ *Parameter*. *Format* : « Unchanged ».

Au moment de l'exécution, des éléments réels tels que net.if.in[2] et net.if.in[3] sont créés. Si un index d'interface donné change (par exemple, si l'index 2 est remplacé par 5 dans la table SNMP), alors lors de la prochaine exécution de la règle de découverte :

- L'ancien élément dépendant net.if.in[2] est marqué comme « perdu » ou supprimé, et aucune nouvelle donnée n'est collectée pour cet élément.
- Un nouvel élément dépendant net.if.in[5] est créé, en commençant avec un historique vide.
- Les données historiques de net.if.in[2] ne sont pas automatiquement déplacées vers net.if.in[5].

Exemple de prototype de déclencheur :

```
{Template_Interface:net.if.in[{#SNMPINDEX}].last()} > 1000000000
```

Exemple de prototype de graphique incluant les éléments :

```
net.if.in[{#SNMPINDEX}]
net.if.out[{#SNMPINDEX}]
```

Cette configuration garantit une surveillance fiable des tables avec des index dynamiques tout en minimisant le trafic SNMP — un seul SNMP walk par cycle d'interrogation est nécessaire, les prototypes d'éléments dépendants extrayant les valeurs requises.

Entités découvertes

Lorsque le serveur est en cours d'exécution, il crée de véritables éléments dépendants, déclencheurs et graphiques en fonction des valeurs renvoyées par la règle de découverte SNMP.

5 Découverte des OID SNMP (hérité)

Overview

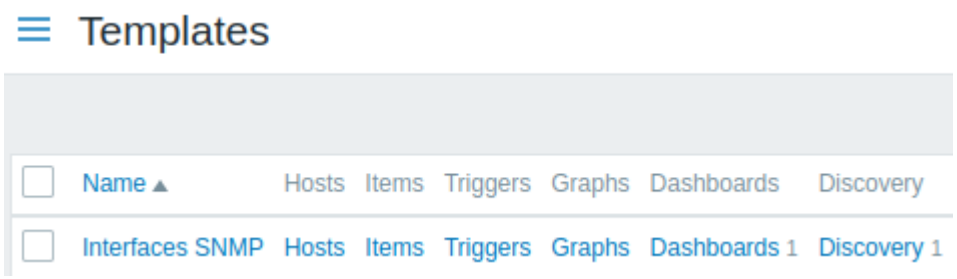
ans cette section, nous allons effectuer une découverte SNMP sur un switch.

Clé d'élément

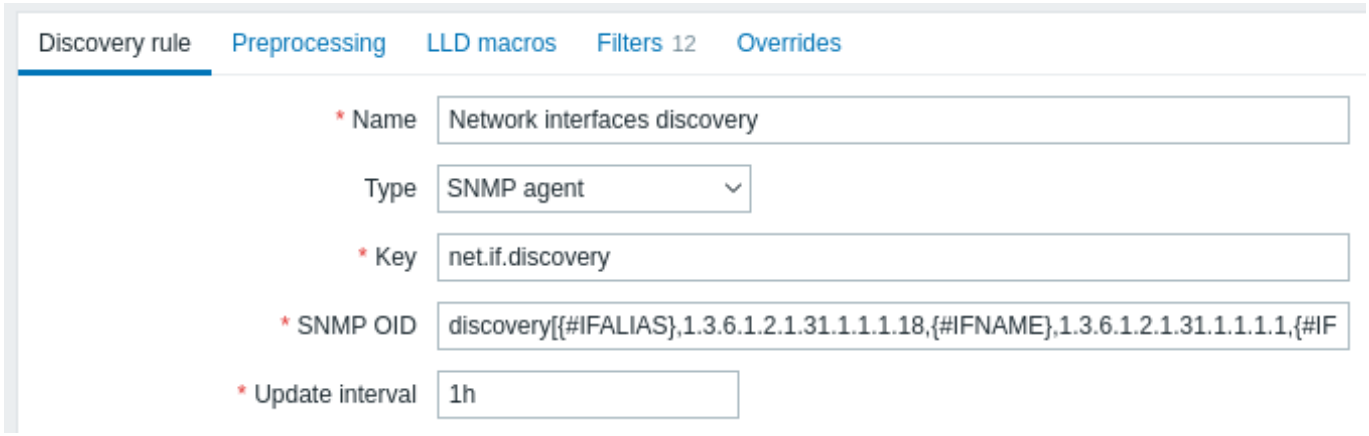
Contrairement à la découverte du système de fichiers et des interfaces réseau, l'élément n'a pas nécessairement besoin d'avoir une clé "snmp.discovery" - le type d'élément SNMP agent est suffisant.

Pour configurer la règle de découverte, procédez comme suit :

- Allez dans : *Collecte de données → Modèles*
- Cliquez sur *Découverte* dans la ligne du modèle approprié



- Cliquez sur *Créer une règle de découverte* dans le coin supérieur droit de l'écran
- Remplissez le formulaire de la règle de découverte avec les détails requis, comme dans la capture d'écran ci-dessous



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

Les OID à découvrir sont définis dans le champ SNMP OID au format suivant : `discovery[{#MACRO1}, oid1, {#MACRO2}, oid2, …,]`

où *{#MACRO1}*, *{#MACRO2}* ... sont des noms de macros lld valides et *oid1*, *oid2*... sont des OID capables de générer des valeurs significatives pour ces macros. Une macro intégrée *{#SNMPINDEX}* contenant l'index de l'OID découvert est appliquée aux entités découvertes. Les entités découvertes sont regroupées par la valeur de la macro *{#SNMPINDEX}*.

Voici un exemple où quelques commandes snmpwalk sont exécutées sur le commutateur :

```
$ snmpwalk -v 2c -c public 192.168.1.1 IF-MIB::ifDescr
IF-MIB::ifDescr.1 = STRING: WAN
IF-MIB::ifDescr.2 = STRING: LAN1
IF-MIB::ifDescr.3 = STRING: LAN2

$ snmpwalk -v 2c -c public 192.168.1.1 IF-MIB::ifPhysAddress
IF-MIB::ifPhysAddress.1 = STRING: 8:0:27:90:7a:75
IF-MIB::ifPhysAddress.2 = STRING: 8:0:27:90:7a:76
IF-MIB::ifPhysAddress.3 = STRING: 8:0:27:2b:af:9e
```

Et le SNMP OID est défini sur : `discovery[{#IFDESCR}, ifDescr, {#IFPHYSADDRESS}, ifPhysAddress]`

Cette règle découvrira alors des entités avec les macros {#IFDESCR} définies sur **WAN**, **LAN1** et **LAN2**, les macros {#IFPHYSADDRESS} définies sur **8:0:27:90:7a:75**, **8:0:27:90:7a:76** et **8:0:27:2b:af:9e**, et les macros {#SNMPINDEX} définies sur les index des OID découverts **1**, **2** et **3** :

```
[
    {
        "{#SNMPINDEX}": "1",
        "{#IFDESCR}": "WAN",
        "{#IFPHYSADDRESS}": "8:0:27:90:7a:75"
    },
    {
        "{#SNMPINDEX}": "2",
        "{#IFDESCR}": "LAN1",
        "{#IFPHYSADDRESS}": "8:0:27:90:7a:76"
    },
    {
        "{#SNMPINDEX}": "3",
        "{#IFDESCR}": "LAN2",
        "{#IFPHYSADDRESS}": "8:0:27:2b:af:9e"
    }
]
```


Si une entité ne possède pas l'OID spécifié, la macro correspondante sera alors omise pour cette entité. Par exemple, si nous avons les données suivantes :

```
ifDescr.1 "Interface #1"
ifDescr.2 "Interface #2"
ifDescr.4 "Interface #4"

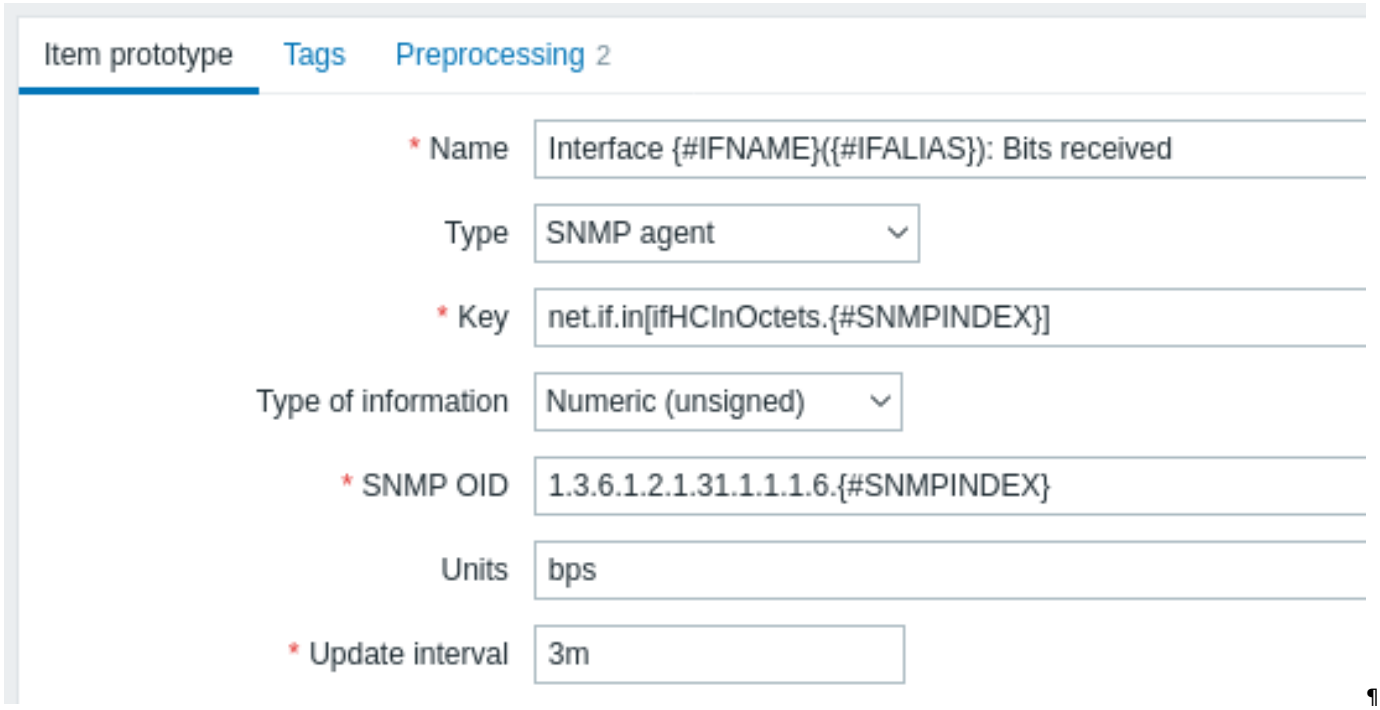
ifAlias.1 "eth0"
ifAlias.2 "eth1"
ifAlias.3 "eth2"
ifAlias.5 "eth4"
```

Alors, dans ce cas, la découverte SNMP `discovery[{#IFDESCR}, ifDescr, {#IFALIAS}, ifAlias]` renverra la structure suivante :


```
[
    {
        "{#SNMPINDEX}": 1,
        "{#IFDESCR}": "Interface #1",
        "{#IFALIAS}": "eth0"
    },
    {
        "{#SNMPINDEX}": 2,
        "{#IFDESCR}": "Interface #2",
        "{#IFALIAS}": "eth1"
    },
    {
        "{#SNMPINDEX}": 3,
        "{#IFALIAS}": "eth2"
    },
    {
        "{#SNMPINDEX}": 4,
        "{#IFDESCR}": "Interface #4"
    },
    {
        "{#SNMPINDEX}": 5,
        "{#IFALIAS}": "eth4"
    }
]
```


Prototypes d'élément

La capture d'écran suivante illustre comment nous pouvons utiliser ces macros dans les prototypes d'éléments :



¶

Vous pouvez créer autant de prototypes d'articles que nécessaire :

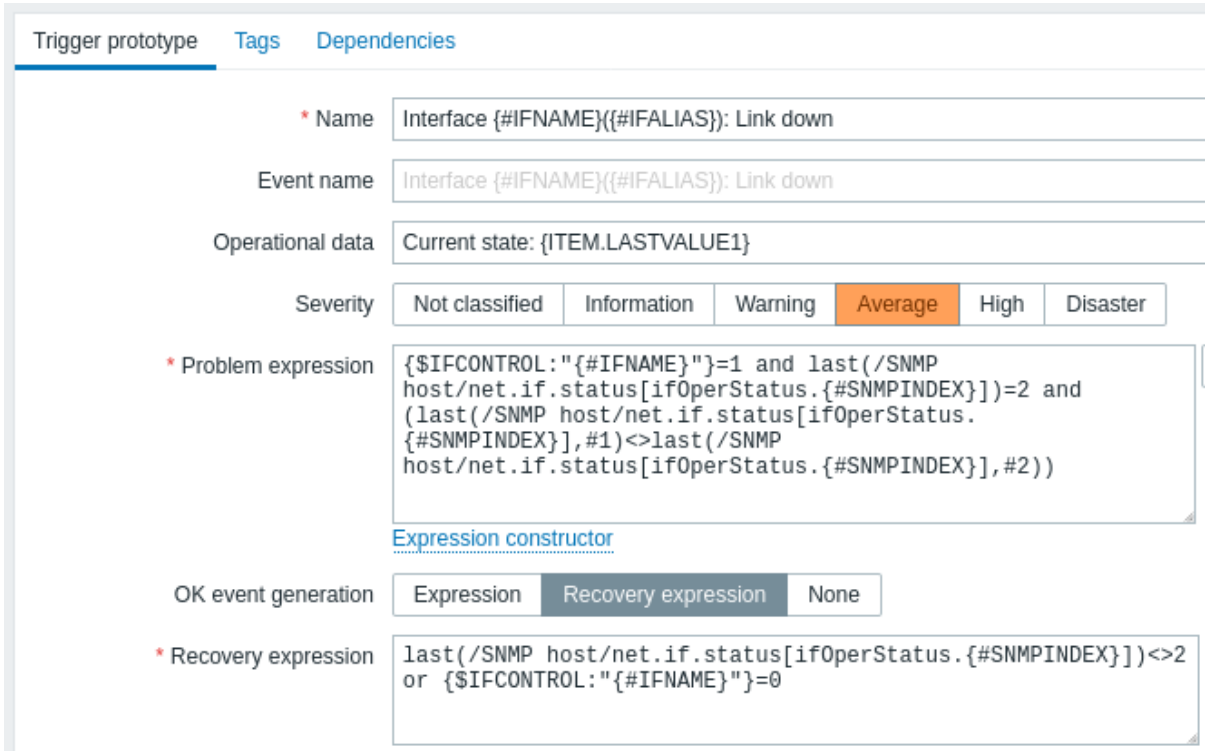
Item prototypes

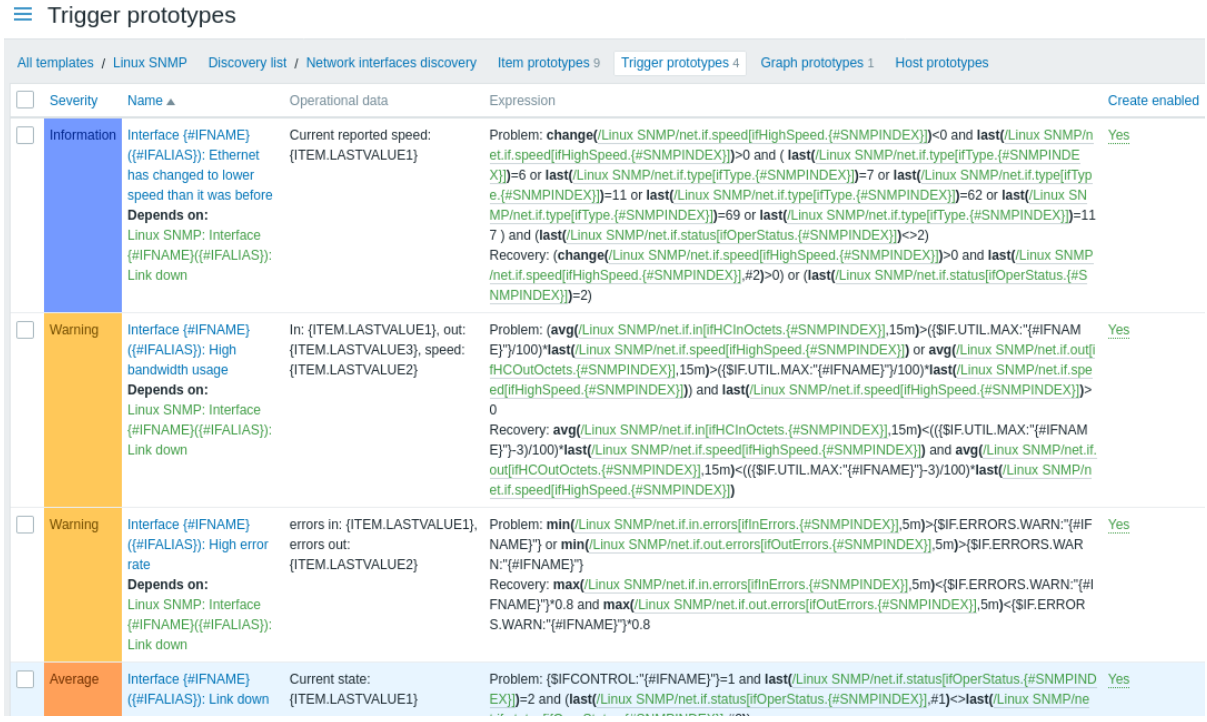
All templates / Linux SNMP Discovery list / Network interfaces discovery Item prototypes 9 Trigger prototypes 4 Graph prototypes 1 Host prototypes

Name ▲	Key	Interval	History	Trends	Type	Create enabled
Interface {#IFNAME}({#IFALIAS}): Bits received	net.if.in[ifHCInOctets.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Bits sent	net.if.out[ifHCOutOctets.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Inbound packets discarded	net.if.in.discards[ifInDiscards.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Inbound packets with errors	net.if.in.errors[ifInErrors.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Interface type	net.if.type[ifType.{#SNMPINDEX}]	1h	7d	0d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Operational status	net.if.status[ifOperStatus.{#SNMPINDEX}]	1m	7d	0	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Outbound packets discarded	net.if.out.discards[ifOutDiscards.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Outbound packets with errors	net.if.out.errors[ifOutErrors.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Speed	net.if.speed[ifHighSpeed.{#SNMPINDEX}]	5m	7d	0d	SNMP agent	Yes

Prototypes de déclencheur

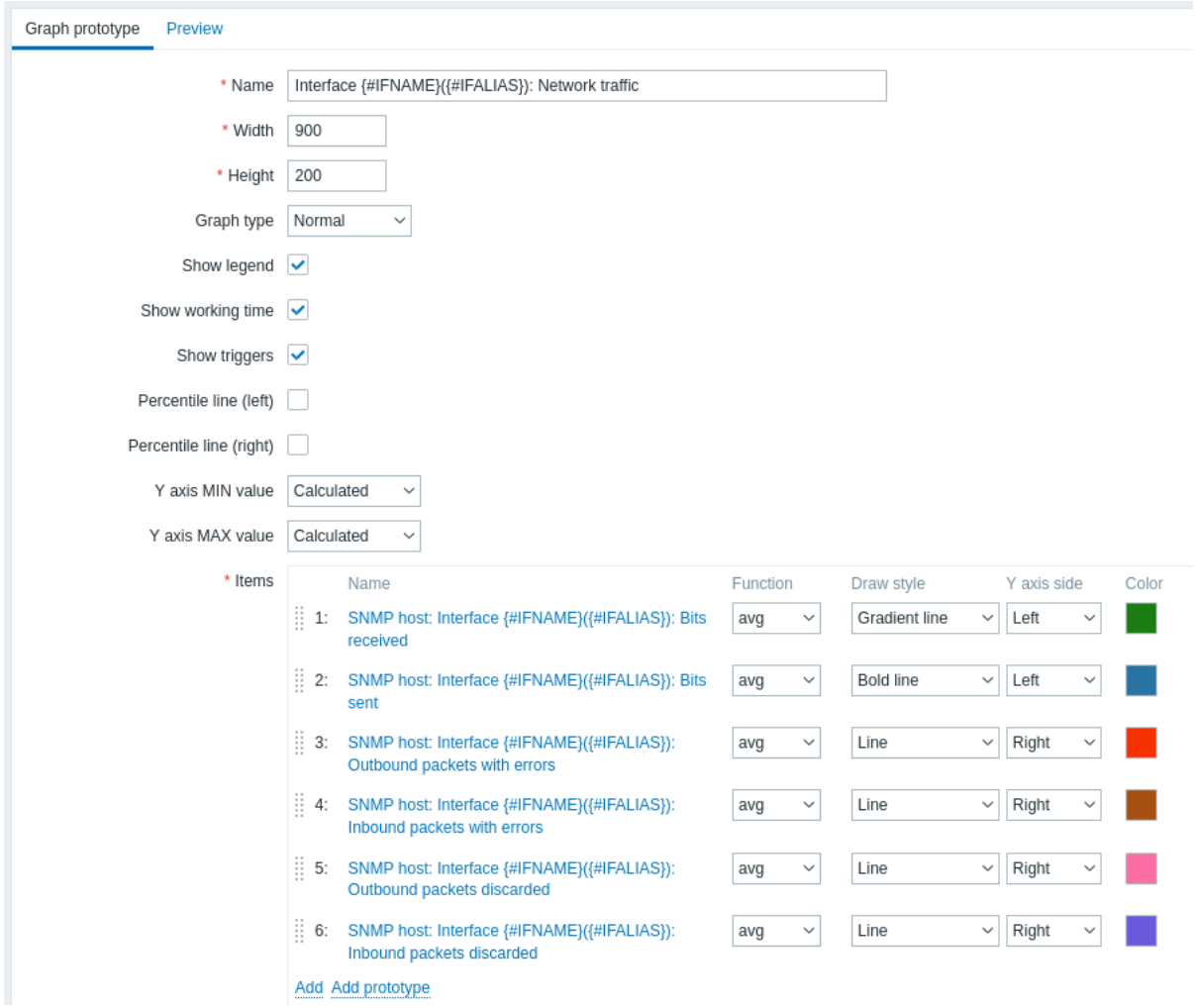
La capture d'écran suivante illustre comment nous pouvons utiliser ces macros dans des prototypes de déclencheur :

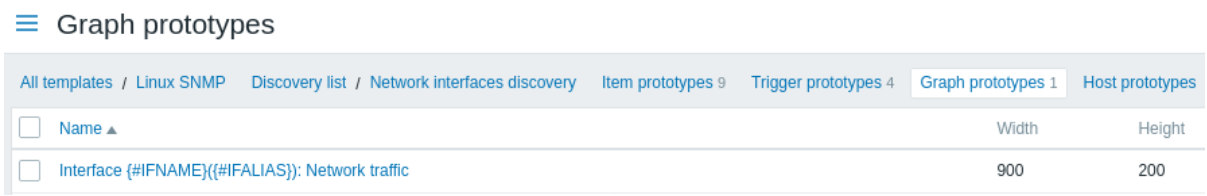




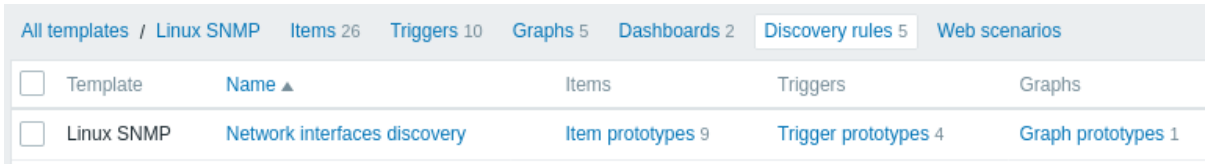
Prototypes de graphiques

La capture d’écran suivante illustre comment nous pouvons utiliser ces macros dans des prototypes de graphe :





Un résumé de notre règle de découverte :



Entités découvertes

Lorsque le serveur s'exécute, il crée des éléments réels, des déclencheurs et des graphiques en fonction des valeurs renvoyées par la règle de découverte SNMP. Dans la configuration de l'hôte, ils sont précédés d'un lien orange vers une règle de découverte d'origine.

Items

All hosts / SNMP host Enabled SNMP Items 81 Triggers 23 Graphs 14 Discovery rules 6 Web scenarios

Name ▲	Triggers	Key	Interval	History	Trends	Type	Status
Network interfaces discovery: Interface enp4s0(): Bits received	Triggers 1	net.if.in[ifHCInOctets.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Bits sent	Triggers 1	net.if.out[ifHCOutOctets.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Inbound packets discarded		net.if.in.discards[ifInDiscards.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Inbound packets with errors	Triggers 1	net.if.in.errors[ifInErrors.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Interface type	Triggers 1	net.if.type[ifType.2]	1h	7d	0d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Operational status	Triggers 2	net.if.status[ifOperStatus.2]	1m	7d	0	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Outbound packets discarded		net.if.out.discards[ifOutDiscards.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Outbound packets with errors	Triggers 1	net.if.out.errors[ifOutErrors.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Speed	Triggers 2	net.if.speed[ifHighSpeed.2]	5m	7d	0d	SNMP agent	Enabled

Triggers

All hosts / SNMP host Enabled SNMP Items 81 Triggers 23 Graphs 14 Discovery rules 6 Web scenarios

Severity	Value	Name ▲	Operational data	Expression
Information	OK	Network interfaces discovery: Interface enp4s0(): Ethernet has changed to lower speed than it was before **Depends on:** SNMP host: Interface enp4s0(): Link down	Current reported speed: {ITEM.LASTVALUE1}	Problem: **change(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**<0 and **last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**>0 and (**last(**/SNMP host/net.if.type[ifType.2]**)**=6 or **last(**/SNMP host/net.if.type[ifType.2]**)**=7 or **last(**/SNMP host/net.if.type[ifType.2]**)**=11 or **last(**/SNMP host/net.if.type[ifType.2]**)**=62 or **last(**/SNMP host/net.if.type[ifType.2]**)**=69 or **last(**/SNMP host/net.if.type[ifType.2]**)**=117) and (**last(**/SNMP host/net.if.status[ifOperStatus.2]**)**<>2) Recovery: (**change(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**>0 and **last(**/SNMP host/net.if.speed[ifHighSpeed.2],#2**)**>0) or (**last(**/SNMP host/net.if.status[ifOperStatus.2]**)**=2)
Warning	OK	Network interfaces discovery: Interface enp4s0(): High bandwidth usage **Depends on:** SNMP host: Interface enp4s0(): Link down	In: {ITEM.LASTVALUE1}, out: {ITEM.LASTVALUE3}, speed: {ITEM.LASTVALUE2}	Problem: (**avg(**/SNMP host/net.if.in[ifHCInOctets.2],15m**)**>({$IF.UTIL.MAX:"enp4s0"}/100)***last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**) or **avg(**/SNMP host/net.if.out[ifHCOutOctets.2],15m**)**>({$IF.UTIL.MAX:"enp4s0"}/100)***last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**) and **last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**>0 Recovery: **avg(**/SNMP host/net.if.in[ifHCInOctets.2],15m**)**<(({$IF.UTIL.MAX:"enp4s0"}-3)/100)***last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)** and **avg(**/SNMP host/net.if.out[ifHCOutOctets.2],15m**)**<(({$IF.UTIL.MAX:"enp4s0"}-3)/100)***last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**
Warning	OK	Network interfaces discovery: Interface enp4s0(): High error rate **Depends on:** SNMP host: Interface enp4s0(): Link down	errors in: {ITEM.LASTVALUE1}, errors out: {ITEM.LASTVALUE2}	Problem: **min(**/SNMP host/net.if.in.errors[ifInErrors.2],5m**)**>{$IF.ERRORS.WARN:"enp4s0"} or **min(**/SNMP host/net.if.out.errors[ifOutErrors.2],5m**)**>{$IF.ERRORS.WARN:"enp4s0"} Recovery: **max(**/SNMP host/net.if.in.errors[ifInErrors.2],5m**)**<{$IF.ERRORS.WARN:"enp4s0"}*0.8 and **max(**/SNMP host/net.if.out.errors[ifOutErrors.2],5m**)**<{$IF.ERRORS.WARN:"enp4s0"}*0.8
Average	OK	Network interfaces discovery: Interface enp4s0(): Link down	Current state: {ITEM.LASTVALUE1}	Problem: {$IFCONTROL:"enp4s0"}=1 and **last(**/SNMP host/net.if.status[ifOperStatus.2]**)**=2 and (**last(**/SNMP host/net.if.status[ifOperStatus.2],#1**)**<>**last(**/SNMP host/net.if.status[ifOperStatus.2],#2**)**) Recovery: **last(**/SNMP host/net.if.status[ifOperStatus.2]**)**<>2 or {$IFCONTROL:"enp4s0"}=0

Graphs

All hosts / SNMP host Enabled SNMP Items 81 Triggers 23 Graphs 14 Discovery rules 6 Web scenarios

Name ▲
Mounted filesystem discovery: /: Disk space usage
Linux SNMP: CPU jumps
CPU discovery: CPU usage
CPU discovery: CPU utilization
Network interfaces discovery: Interface enp4s0(): Network traffic

6 Découverte des objets JMX

Aperçu

Il est possible de découvrir tous les attributs MBeans ou MBean JMX ou de spécifier un modèle pour la découverte de ces objets.

Il est indispensable de comprendre la différence entre Mbean et les attributs Mbean pour la configuration des règles de découverte. Un MBean est un objet pouvant représenter un périphérique, une application ou toute ressource devant être gérée.

Par exemple, il existe un Mbean qui représente un serveur Web. Ses attributs sont : nombre de connexions, nombre de threads, délai de requête, cache de fichier http, utilisation de la mémoire, etc. En exprimant cette pensée en langage humain compréhensif, nous pouvons définir une machine à café comme un Mbean dont les attributs à surveiller sont les suivants : quantité d'eau par tasse, consommation moyenne d'eau pendant une certaine période, nombre de grains de café requis par tasse, grains de café et temps de remplissage en eau, etc.

Clé d'élément

Dans la configuration de la règle de découverte, sélectionnez **Agent JMX** dans le champ *Type*.

Deux clés d'élément sont prises en charge pour la découverte d'objets JMX : jmx.discovery[] et jmx.get[] :

Clé d'élément	Valeur de retour	Paramètres	Commentaire
jmx.discovery[<mode de découverte>,<nom d'objet>,<description courte unique>]	Cet élément renvoie un tableau JSON contenant des macros LLD décrivant des objets MBean ou leurs attributs.	**mode de découverte** - l'une des valeurs suivantes : *attributes* (récupérer les attributs des MBean JMX, par défaut) ou *beans* (récupérer les MBean JMX) **nom d'objet** - modèle de nom d'objet (voir la documentation) identifiant les noms des MBean à récupérer (vide par défaut, ce qui récupère tous les beans enregistrés) **description courte unique** - une description unique permettant d'avoir plusieurs éléments JMX avec le même mode de découverte et le même nom d'objet sur l'hôte (facultatif)	Exemples : → jmx.discovery - récupérer tous les attributs des MBean JMX → jmx.discovery[beans] - récupérer tous les MBean JMX → jmx.discovery[attributes,"*:type=GarbageCollector,name - récupérer tous les attributs des collecteurs de déchets → jmx.discovery[beans,"*:type=GarbageCollector,name=* - récupérer tous les collecteurs de déchets Il existe certaines limitations concernant les propriétés MBean que cet élément peut renvoyer, en raison du nombre limité de caractères pris en charge pour la génération des noms de macros (les caractères pris en charge peuvent être décrits par l'expression régulière suivante : `A-Z0-9_\.`). Ainsi, par exemple, pour découvrir des propriétés MBean contenant un mot avec trait d'union ou des caractères non ASCII, vous devez utiliser `jmx.get[]`.
jmx.get[<mode de découverte>,<nom d'objet>,<description courte unique>]			

Clé d'élément			
	Cet élément renvoie un tableau JSON contenant des objets MBean ou leurs attributs. Comparé à `jmx.discovery[]`, il ne définit pas de macros LLD.	**mode de découverte** - l'une des valeurs suivantes : *attributes* (récupérer les attributs des MBean JMX, par défaut) ou *beans* (récupérer les MBean JMX) **nom d'objet** - modèle de nom d'objet (voir la documentation) identifiant les noms des MBean à récupérer (vide par défaut, ce qui récupère tous les beans enregistrés) **description courte unique** - une description unique permettant d'avoir plusieurs éléments JMX avec le même mode de découverte et le même nom d'objet sur l'hôte (facultatif)	Lors de l'utilisation de cet élément, il est nécessaire de définir des macros de découverte de bas niveau personnalisées, pointant vers des valeurs extraites du JSON renvoyé à l'aide de JSONPath.

Attention:
Si aucun paramètre n'est transmis, tous les attributs MBean de JMX sont demandés. Ne pas spécifier de paramètres pour la découverte JMX ou tenter de recevoir tous les attributs pour une plage étendue comme `*:type=*,name=*` peut entraîner des problèmes potentiels de performances.

Utilisation de jmx.discovery

Cet élément renvoie un objet JSON avec des macros de découverte de bas niveau décrivant des objets ou des attributs MBean. Par exemple, dans la découverte des attributs MBean (reformaté pour plus de clarté) :


```
[
    {
        "{#JMXVALUE}":"0",
        "{#JMXTYPE}":"java.lang.Long",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,CollectionCount",
        "{#JMXATTR}":"CollectionCount"
    },
    {
        "{#JMXVALUE}":"0",
        "{#JMXTYPE}":"java.lang.Long",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,CollectionTime",
        "{#JMXATTR}":"CollectionTime"
    },
    {
        "{#JMXVALUE}":"true",
        "{#JMXTYPE}":"java.lang.Boolean",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,Valid",
        "{#JMXATTR}":"Valid"
    },
    {
        "{#JMXVALUE}":"PS Scavenge",
        "{#JMXTYPE}":"java.lang.String",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,Name",
        "{#JMXATTR}":"Name"
    },
    {
        "{#JMXVALUE}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXTYPE}":"javax.management.ObjectName",
```

```
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,ObjectName",
        "{#JMXATTR}":"ObjectName"
    }
]
```

Dans la découverte des MBeans (reformaté pour plus de clarté) :

```
[
    {
        "{#JMXDOMAIN}":"java.lang",
        "{#JMXTYPE}":"GarbageCollector",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXNAME}":"PS Scavenge"
    }
]
```

Macros prises en charge

Les macros suivantes sont prises en charge pour une utilisation dans le filter de la règle de découverte et les prototypes d'éléments, de déclencheurs et de graphiques :

Macro	Description
Découverte des attributs MBean	
{#JMXVALUE}	Valeur d'attribut.
{#JMXTYPE}	Type d'attribut.
{#JMXOBJ}	Nom de l'objet.
{#JMXDESC}	Nom d'objet incluant le nom d'attribut.
{#JMXATTR}	Nom d'attribut.
Découverte des MBeans	
{#JMXDOMAIN}	Domaine MBean. (*nom réservé Zabbix*)
{#JMXOBJ}	Nom de l'objet. (*nom réservé Zabbix*)
{#JMX<key property>}	Propriétés MBean (comme {#JMXTYPE}, {#JMXNAME}) (voir Limitations ci-dessous).

Limitations

Il existe certaines limitations liées à l'algorithme de création des noms de macros LLD à partir des noms de propriétés MBean :

- les noms d'attribut sont convertis en majuscules
- les noms d'attribut sont ignorés (aucune macro LLD n'est générée) s'ils contiennent des caractères non pris en charge pour les noms de macros LLD. Les caractères pris en charge peuvent être décrits par l'expression régulière suivante : `A-Z0-9_\.`
- si un attribut s'appelle « obj » ou « domain », il sera ignoré en raison du chevauchement avec les valeurs des propriétés Zabbix réservées {#JMXOBJ} et {#JMXDOMAIN}

Veuillez considérer cet exemple de jmx.discovery (avec le mode « beans »). Le MBean a les propriétés suivantes définies (dont certaines seront ignorées ; voir ci-dessous) :

```
name=test
  =Type
attributes []=1,2,3
Name=NameOfTheTest
domAin=some
```

À la suite de la découverte JMX, les macros LLD suivantes seront générées :

- {#JMXDOMAIN} - interne à Zabbix, décrit le domaine du MBean
- {#JMXOBJ} - interne à Zabbix, décrit l'objet MBean
- {#JMXNAME} - créée à partir de la propriété « name »

Les propriétés ignorées sont :

- тип : son nom contient des caractères non pris en charge (non ASCII)
- attributes[] : son nom contient des caractères non pris en charge (les crochets ne sont pas pris en charge)
- Name : il est déjà défini (name=test)
- domAin : c'est un nom réservé de Zabbix

Exemples

Examinons deux autres exemples pratiques de création d'une règle LLD avec l'utilisation de MBean. Pour mieux comprendre la différence entre une règle LLD collectant des MBeans et une règle LLD collectant des attributs de MBean, veuillez consulter le tableau suivant :

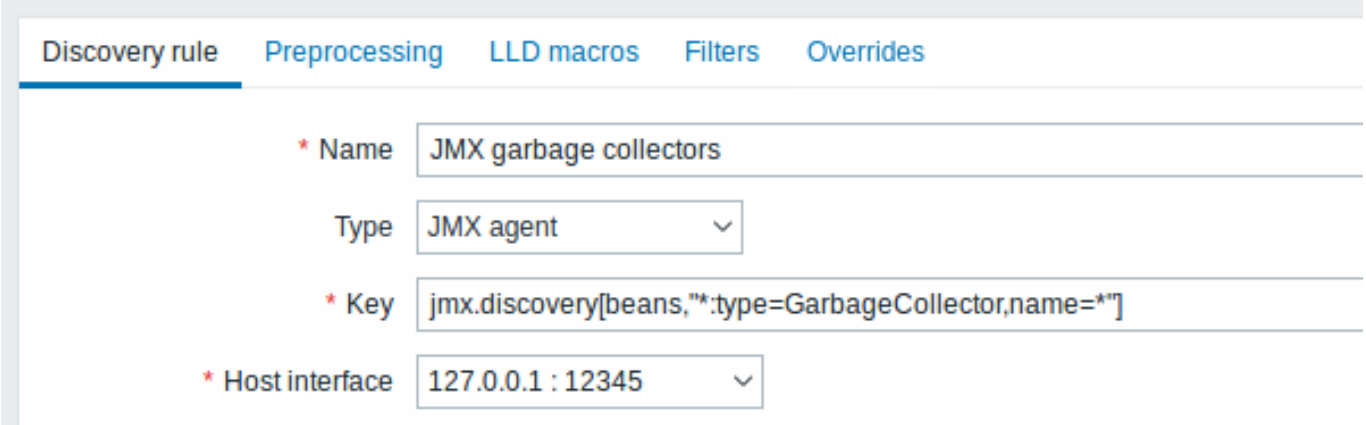
MBean1	**MBean2**	**MBean3**
MBean1Attribute1	MBean2Attribute1	MBean3Attribute1
MBean1Attribute2	MBean2Attribute2	MBean3Attribute2
MBean1Attribute3	MBean2Attribute3	MBean3Attribute3

Exemple 1 : Découverte des MBeans

Cette règle renverra 3 objets : la ligne supérieure de la colonne : MBean1, MBean2, MBean3.

Pour plus d'informations sur les objets, veuillez consulter le tableau des macros prises en charge, section *Découverte des MBeans*.

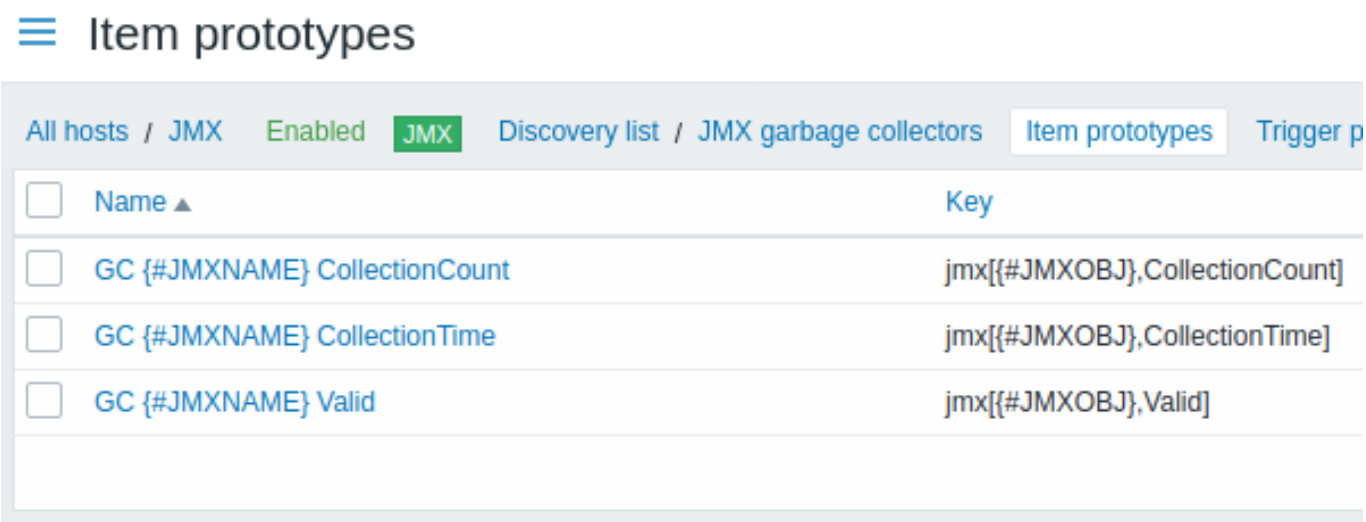
La configuration de la règle de découverte collectant les MBeans (sans les attributs) ressemble à ce qui suit :



La clé utilisée ici :

```
jmx.discovery[beans,"*:type=GarbageCollector,name=*"]
```

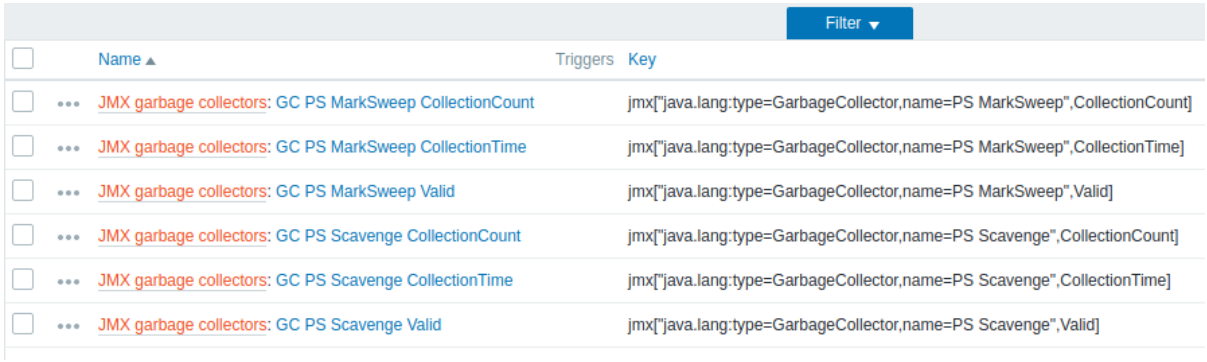
Tous les collecteurs de déchets sans attributs seront découverts. Comme les collecteurs de déchets ont le même ensemble d'attributs, nous pouvons utiliser les attributs souhaités dans les prototypes d'éléments de la manière suivante :



Les clés utilisées ici :

```
jmx[{#JMXOBJ},CollectionCount]
jmx[{#JMXOBJ},CollectionTime]
jmx[{#JMXOBJ},Valid]
```

La règle de découverte LLD produira un résultat proche de celui-ci (des éléments sont découverts pour deux collecteurs de déchets) :

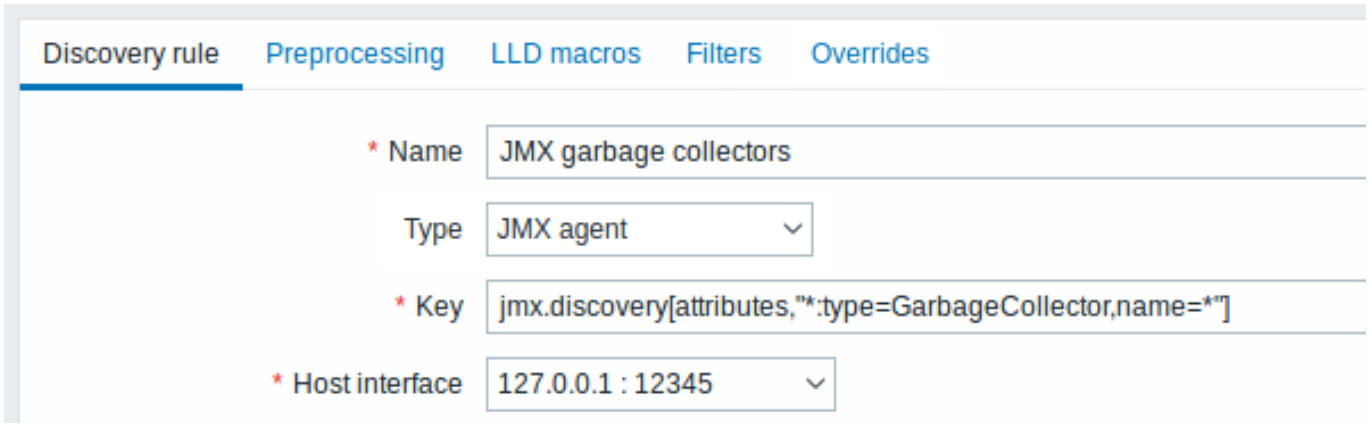


Exemple 2 : Découverte des attributs MBean

Cette règle renverra 9 objets avec les champs suivants : MBean1Attribute1, MBean2Attribute1, MBean3Attribute1,MBean1Attribute2,MBean2A MBean3Attribute2, MBean1Attribute3, MBean2Attribute3, MBean3Attribute3.

Pour plus d'informations sur les objets, veuillez consulter le tableau des macros prises en charge, section *Découverte des attributs MBean*.

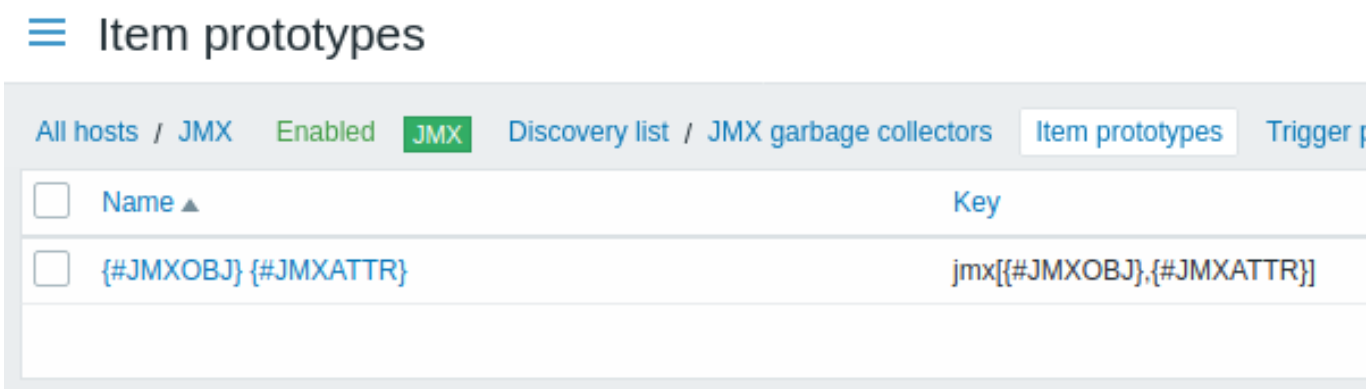
La configuration de la règle de découverte collectant les attributs MBean ressemble à ce qui suit :



La clé utilisée ici :

`jmx.discovery[attributes,"*:type=GarbageCollector,name=*"]`

Tous les collecteurs de déchets avec un attribut d'élément unique seront découverts.



Dans ce cas particulier, un élément sera créé à partir du prototype pour chaque attribut MBean. Le principal inconvénient de cette configuration est que la création de déclencheurs à partir de prototypes de déclencheur est impossible, car il n'existe qu'un seul prototype d'élément pour tous les attributs. Cette configuration peut donc être utilisée pour la collecte de données, mais n'est pas recommandée pour la supervision automatique.

Utilisation de jmx.get

`jmx.get[]` est similaire à l'élément `jmx.discovery[]`, mais il ne transforme pas les propriétés des objets Java en noms de macros de découverte de bas niveau et peut donc renvoyer des valeurs sans les limitations qui sont associées à la génération des noms de macros LLD, telles que les tirets ou les caractères non ASCII.

Lors de l'utilisation de `jmx.get[]` pour la découverte, les macros de découverte de bas niveau peuvent être définies séparément dans l'onglet personnalisé macro LLD de la configuration de la règle de découverte, en utilisant JSONPath pour pointer vers les

valeurs requises.

Découvrir les MBeans

Elément de découverte : `jmx.get[beans,"com.example:type=*,*"]`

Réponse :

```
[
    {
        "object": "com.example:type=Hello,data-src=data-base,  =    ",
        "domain": "com.example",
        "properties": {
            "data-src": "data-base",
            "  ": "    ",
            "type": "Hello"
        }
    },
    {
        "object": "com.example:type=Atomic",
        "domain": "com.example",
        "properties": {
            "type": "Atomic"
        }
    }
]
```

Découvrir les attributs MBean

Elément de découverte : `jmx.get[attributes,"com.example:type=*,*"]`

Réponse :

```
[
    {
        "object": "com.example:type=*",
        "domain": "com.example",
        "properties": {
            "type": "Simple"
        }
    },
    {
        "object": "com.zabbix:type=yes,domain=zabbix.com,data-source=/dev/rand,  =    ,obj=true",
        "domain": "com.zabbix",
        "properties": {
            "type": "Hello",
            "domain": "com.example",
            "data-source": "/dev/rand",
            "  ": "    ",
            "obj": true
        }
    }
]
```

7 Découverte des capteurs IPMI

Aperçu

Il est possible de découvrir automatiquement les capteurs IPMI.

Pour ce faire, vous pouvez utiliser une combinaison de :

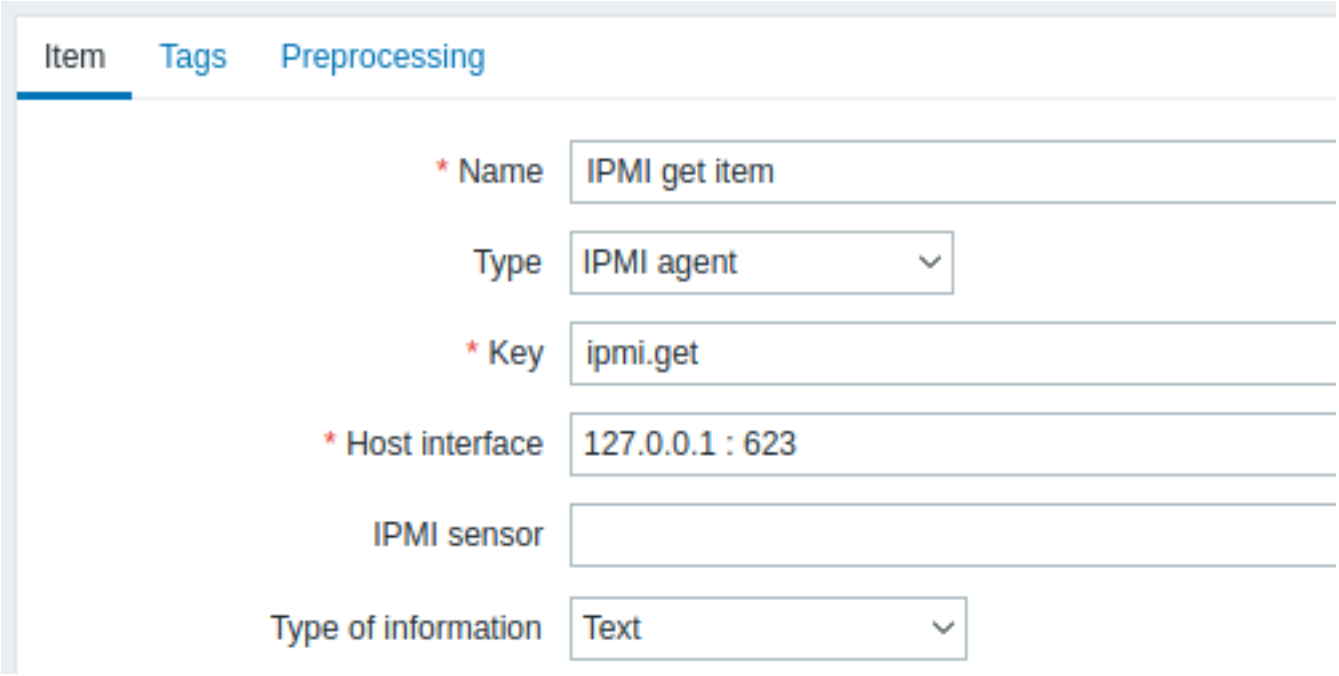
- l'élément IPMI `ipmi.get` comme élément maître
- une règle de découverte de bas niveau dépendante et des prototypes d'élément

Configuration

Élément principal

Créez un élément IPMI à l'aide de la clé suivante :

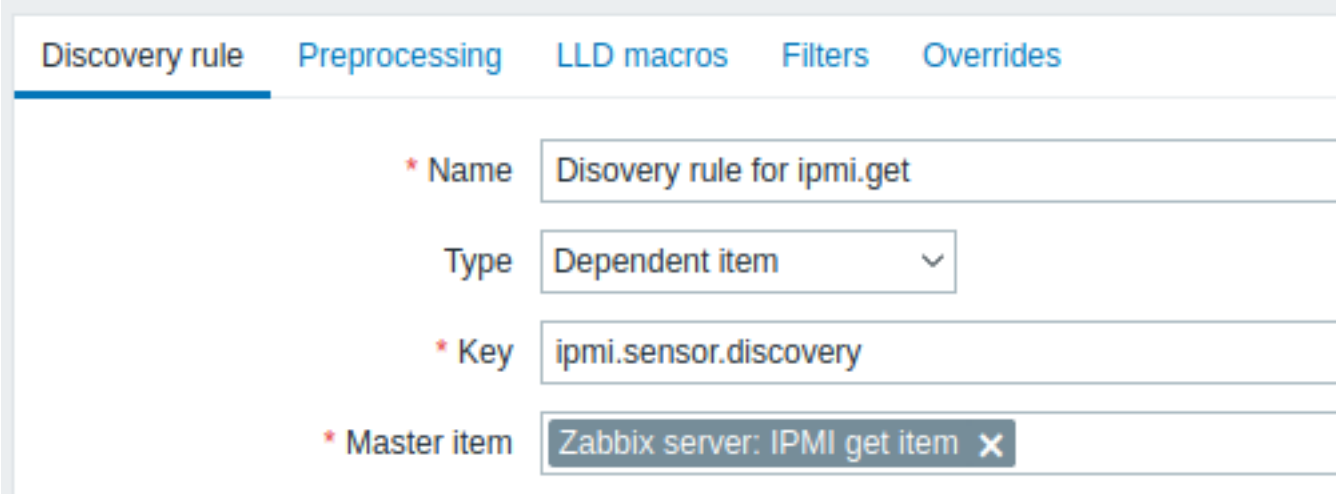
`ipmi.get`



Définissez le type d'informations sur "Texte" pour les données JSON éventuellement volumineuses.

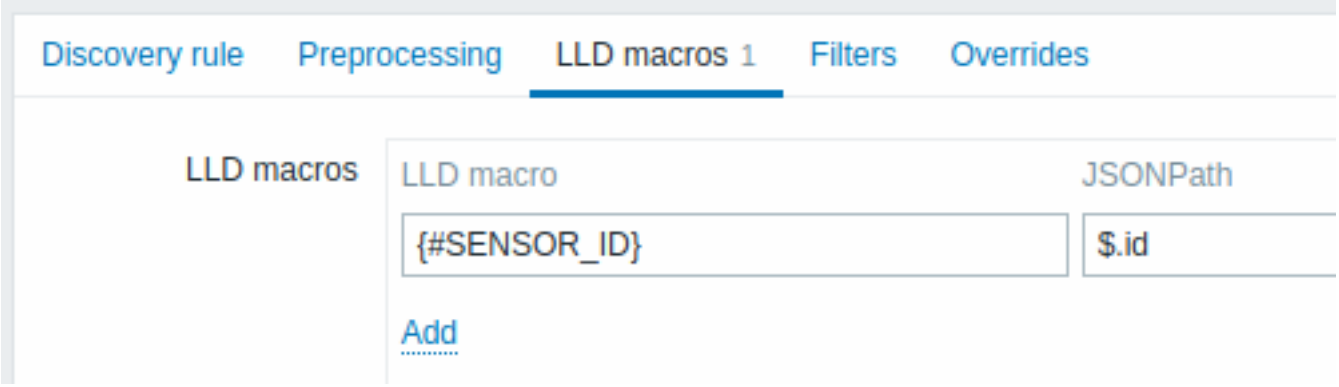
Règle LLD dépendante

Créez une règle de découverte de bas niveau en tant que type "Élément dépendant" :



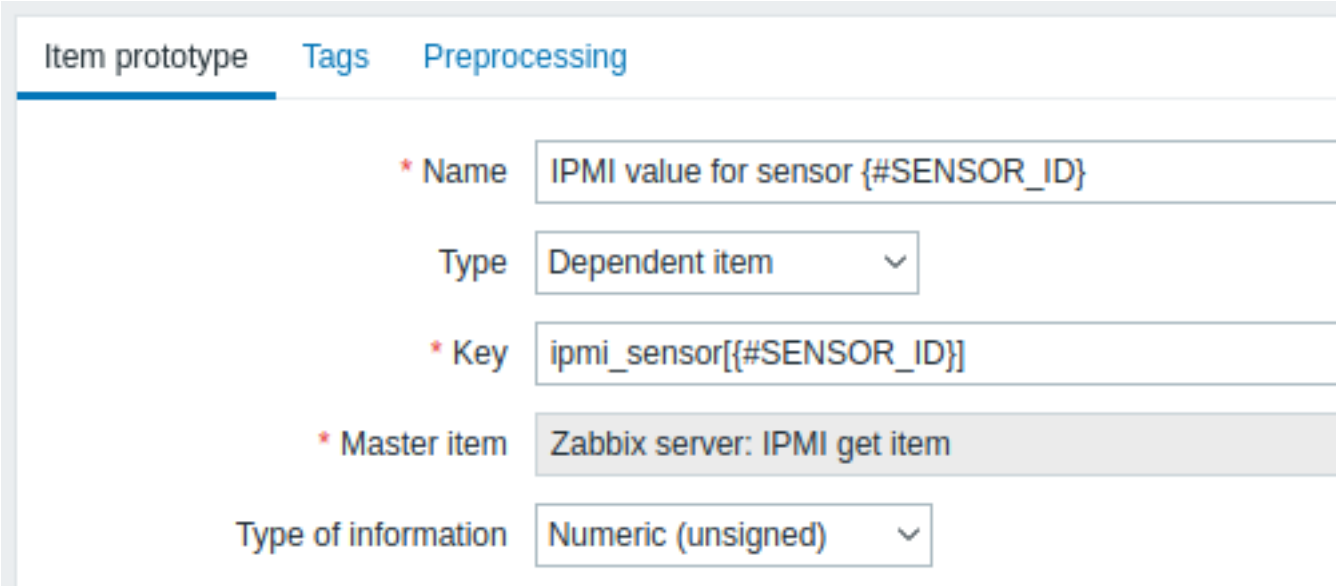
En tant qu'élément principal, sélectionnez l'élément `ipmi.get` que nous avons créé.

Dans l'onglet "Macros LLD", définissez une macro personnalisée avec le JSONPath correspondant :



Prototype d'élément dépendant

Créez un prototype d'élément avec le type "Elément dépendant" dans cette règle LLD. En tant qu'élément maître pour ce prototype, sélectionnez l'élément `ipmi.get` que nous avons créé.



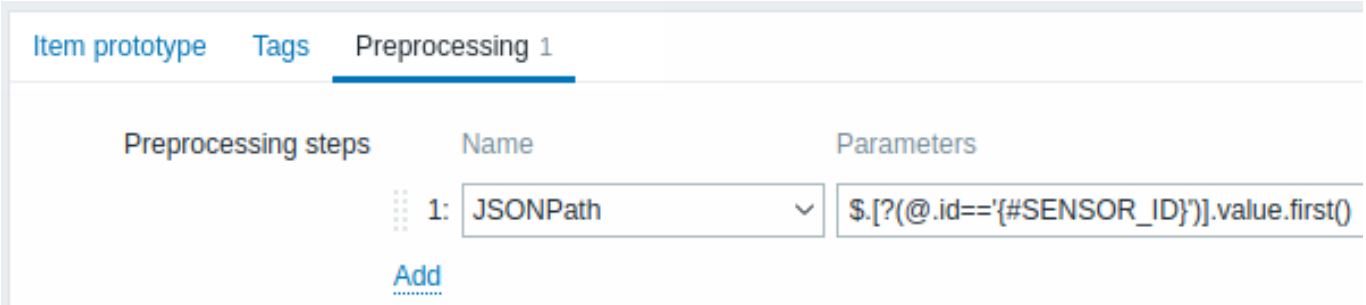
Notez l'utilisation de la macro {#SENSOR_ID} dans le nom et la clé du prototype d'élément :

- *Nom*: IPMI value for sensor {#SENSOR_ID}
- *Clé*: ipmi_sensor[{#SENSOR_ID}]

Comme type d'information, *Numérique (non signé)*.

Dans l'onglet "Prétraitement" du prototype d'élément, sélectionnez JSONPath et utilisez l'expression JSONPath suivante comme paramètre :

`$.[?(@.id=='{#SENSOR_ID}')].value.first()`



Lorsque la découverte démarre, un élément par capteur IPMI est créé. Cet élément renverra la valeur entière du capteur donné.

8 Découverte des services systemd

Aperçu

Il est possible de découvrir des unités systemd (services, par défaut) avec Zabbix.

Clé d'élément

La clé d'élément à utiliser dans la règle de découverte est

`systemd.unit.discovery`

> **Attention:**
> Cette clé d'élément n'est prise en charge que par Zabbix agent 2.

Cet élément renvoie un JSON contenant des informations sur les unités systemd, par exemple :

```
[{
    "{#UNIT.NAME}": "mysqld.service",
    "{#UNIT.DESCRIPTION}": "Serveur MySQL",
    "{#UNIT.LOADSTATE}": "loaded",
```

```
    "{#UNIT.ACTIVESTATE}": "active",
    "{#UNIT.SUBSTATE}": "running",
    "{#UNIT.FOLLOWED}": "",
    "{#UNIT.PATH}": "/org/freedesktop/systemd1/unit/mysqld_2eservice",
    "{#UNIT.JOBID}": 0,
    "{#UNIT.JOBTYPE}": "",
    "{#UNIT.JOBPATH}": "/",
    "{#UNIT.UNITFILESTATE}": "enabled"
    "{#UNIT.SERVICETYPE}": "simple"
}, {
    "{#UNIT.NAME}": "systemd-journald.socket",
    "{#UNIT.DESCRIPTION}": "Socket du journal",
    "{#UNIT.LOADSTATE}": "loaded",
    "{#UNIT.ACTIVESTATE}": "active",
    "{#UNIT.SUBSTATE}": "running",
    "{#UNIT.FOLLOWED}": "",
    "{#UNIT.PATH}": "/org/freedesktop/systemd1/unit/systemd_2djournald_2esocket",
    "{#UNIT.JOBID}": 0,
    "{#UNIT.JOBTYPE}": "",
    "{#UNIT.JOBPATH}": "/",
    "{#UNIT.UNITFILESTATE}": "enabled"
}]
```

Découverte des unités systemd désactivées

Il est également possible de découvrir les unités systemd **désactivées**. Dans ce cas, trois macros sont renvoyées dans le JSON résultant :

- {#UNIT.PATH}
- {#UNIT.ACTIVESTATE}
- {#UNIT.UNITFILESTATE}.

Attention:
Pour que les éléments et les déclencheurs soient créés à partir des prototypes pour les unités systemd désactivées, veillez à ajuster (ou supprimer) les filtres LLD d'exclusion pour {#UNIT.ACTIVESTATE} et {#UNIT.UNITFILESTATE}.

Macros prises en charge

Les macros suivantes sont prises en charge pour une utilisation dans le filtre de la règle de découverte et dans les prototypes d'éléments, de déclencheurs et de graphiques :

Macro	Description
{#UNIT.NAME}	Nom principal de l'unité.
{#UNIT.DESCRIPTION}	Description lisible par l'homme.
{#UNIT.LOADSTATE}	État de chargement (c'est-à-dire si le fichier d'unité a été chargé avec succès).
{#UNIT.ACTIVESTATE}	État actif (c'est-à-dire si l'unité est actuellement démarrée ou non).
{#UNIT.SUBSTATE}	Sous-état (une version plus fine de l'état actif, spécifique au type d'unité, contrairement à l'état actif).
{#UNIT.FOLLOWED}	Unité dont l'état est suivi par cette unité, le cas échéant ; sinon, une chaîne vide.
{#UNIT.PATH}	Chemin de l'objet unité.
{#UNIT.JOBID}	ID numérique de tâche si une tâche est en file d'attente pour l'unité de tâche ; 0 sinon.
{#UNIT.JOBTYPE}	Type de tâche.
{#UNIT.JOBPATH}	Chemin de l'objet tâche.
{#UNIT.UNITFILESTATE}	État d'installation du fichier d'unité.
{#UNIT.SERVICETYPE}	Type de l'unité de service (par exemple, `simple`, `forking`, `oneshot`, `idle`, etc.). Cette macro n'est renvoyée que si l'unité est un service.

Prototypes d'éléments

Les prototypes d'éléments pouvant être créés sur la base de la découverte des services systemd incluent, par exemple :

- Nom de l'élément : {#UNIT.DESCRIPTION} `active state info` ; clé de l'élément : `systemd.unit.info["{#UNIT.NAME}"]`
- Nom de l'élément : {#UNIT.DESCRIPTION} `load state info` ; clé de l'élément : `systemd.unit.info["{#UNIT.NAME}",Load`

9 Découverte des services Windows

Aperçu

De la même manière que les systèmes de fichiers sont découverts, il est également possible de découvrir les services Windows.

Clé d'élément

L'élément à utiliser dans la règle de découverte est

`service.discovery`

Macros prises en charge

Les macros suivantes sont prises en charge pour une utilisation dans le filtre de la règle de découverte et dans les prototypes d'éléments, de déclencheurs et de graphiques :

Macro	Description
{#SERVICE.NAME}	Nom du service.
{#SERVICE.DISPLAYNAME}	Nom affiché du service.
{#SERVICE.DESCRIPTION}	Description du service.
{#SERVICE.STATE}	Valeur numérique de l'état du service. Voir l'élément service.info pour plus de détails.
{#SERVICE.STATENAME}	Nom de l'état du service. Voir l'élément service.info pour plus de détails.
{#SERVICE.PATH}	Chemin du service.
{#SERVICE.USER}	Utilisateur du service.
{#SERVICE.STARTUP}	Valeur numérique du type de démarrage du service. Voir l'élément service.info pour plus de détails.
{#SERVICE.STARTUPNAME}	Nom du type de démarrage du service. Voir l'élément service.info pour plus de détails.
{#SERVICE.STARTUPTRIGGER}	Valeur numérique indiquant si le type de démarrage du service a : 0 - aucun déclencheur de démarrage 1 - a des déclencheurs de démarrage Elle est utile pour découvrir des types de démarrage de service tels que *Automatique (démarrage par déclencheur)*, *Automatique différé (démarrage par déclencheur)* et *Manuel (démarrage par déclencheur).*

Sur la base de la découverte des services Windows, vous pouvez créer un prototype d'élément comme

`service.info[{#SERVICE.NAME},<param>]`

où `param` accepte les valeurs suivantes : *state*, *displayname*, *path*, *user*, *startup* ou *description*.

Par exemple, pour obtenir le nom affiché d'un service, vous pouvez utiliser un élément "service.info[{#SERVICE.NAME},displayname]". Si la valeur `param` n'est pas spécifiée ("service.info[{#SERVICE.NAME}]"), le paramètre par défaut *state* est utilisé.

10 Découverte des instances de compteurs de performance Windows

Vue d'ensemble

Il est possible de découvrir les instances d'objet des compteurs de performance Windows. Cela est utile pour les compteurs de performance à instances multiples.

Clé d'élément

Pour configurer la règle de découverte, utilisez l'élément suivant :

- `perf_instance.discovery[object]`

Notez que le nom de l'objet peut être localisé. Par exemple :

```
perf_instance.discovery[Processor] # Le nom de l'objet est en anglais.
perf_instance.discovery[Processador] # Le nom de l'objet est en portugais.
```

Sinon, pour garantir que le nom de l'objet soit fourni en anglais, indépendamment de la localisation du système d'exploitation, utilisez l'élément suivant :

- `perf_instance_en.discovery[object]`

Par exemple :

```
perf_instance_en.discovery[Processor]
perf_instance_en.discovery[Memory]
```

Macros prises en charge

Le processus de découverte renverra toutes les instances de l'objet spécifié dans la macro `{#INSTANCE}` :

```
[
    {"{#INSTANCE}":"0"},
    {"{#INSTANCE}":"1"},
    {"{#INSTANCE}":"_Total"}
]
```

Cette macro peut être utilisée dans les prototypes des éléments `perf_counter[]` et `perf_counter_en[]`.

Par exemple, si la clé d'élément utilisée dans la règle de découverte est `perf_instance.discovery[Processor]`, vous pouvez créer le prototype d'élément suivant :

```
perf_counter["\Processor({#INSTANCE})\% Processor Time"]
```

Remarque :

- Si l'objet spécifié est introuvable ou ne prend pas en charge les instances variables, l'élément de découverte deviendra NOTSUPPORTED.
- Si l'objet spécifié prend en charge les instances variables mais ne possède actuellement aucune instance, un tableau JSON vide sera renvoyé.
- Les instances en double seront ignorées.

11 Découverte à l'aide de requêtes WMI

Aperçu

WMI est une interface puissante de Windows qui peut être utilisée pour récupérer diverses informations sur les composants, les services, l'état et les logiciels installés de Windows.

Il peut être utilisé pour la découverte de disques physiques et leur collecte de données de performances, la découverte d'interfaces réseau, la découverte d'invités Hyper-V, la surveillance des services Windows et bien d'autres choses dans le système d'exploitation Windows.

Ce type de découverte de bas niveau est effectué à l'aide de requêtes WQL dont les résultats sont automatiquement transformés en un objet JSON adapté à la découverte de bas niveau.

Clé d'élément

L'élément à utiliser dans la règle de découverte est :

`wmi.getall[<namespace>,<query>]`

Cet élément transforme le résultat de la requête en un tableau JSON. Par exemple :

`select * from Win32_DiskDrive where Name like '%PHYSICALDRIVE%'`

peut renvoyer quelque chose comme ceci :

```
[
    {
        "DeviceID" : "\\.\PHYSICALDRIVE0",
        "BytesPerSector" : 512,
        "Capabilities" : [
            3,
            4
        ],
        "CapabilityDescriptions" : [
            "Random Access",
            "Supports Writing"
        ],
        "Caption" : "VBOX HARDDISK ATA Device",
        "ConfigManagerErrorCode" : 0,
```

```
        "ConfigManagerUserConfig" : "False",
        "CreationClassName" : "Win32_DiskDrive",
        "Description" : "Disk drive",
        "FirmwareRevision" : "1.0",
        "Index" : 0,
        "InterfaceType" : "IDE"
    },
    {
        "DeviceID" : "\\.\PHYSICALDRIVE1",
        "BytesPerSector" : 512,
        "Capabilities" : [
            3,
            4
        ],
        "CapabilityDescriptions" : [
            "Random Access",
            "Supports Writing"
        ],
        "Caption" : "VBOX HARDDISK ATA Device",
        "ConfigManagerErrorCode" : 0,
        "ConfigManagerUserConfig" : "False",
        "CreationClassName" : "Win32_DiskDrive",
        "Description" : "Disk drive",
        "FirmwareRevision" : "1.0",
        "Index" : 1,
        "InterfaceType" : "IDE"
    }
]
```


Macros de découverte de bas niveau

Même si aucune macro de découverte de bas niveau n'est créée dans le JSON renvoyé, ces macros peuvent être définies par l'utilisateur comme une étape supplémentaire, à l'aide de macros LLD personnalisées avec un JSONPath pointant vers les valeurs découvertes dans le JSON renvoyé.

Les macros peuvent ensuite être utilisées pour créer des prototypes d'éléments, de déclencheurs, etc.

12 Découverte à l'aide de requêtes SQL ODBC

Aperçu

Ce type de découverte de bas niveau est effectué à l'aide de requêtes SQL, dont les résultats sont automatiquement transformés en un objet JSON adapté à la découverte de bas niveau.

Clé d'élément

Les requêtes SQL sont exécutées à l'aide d'un type d'élément « Database monitor ». Par conséquent, la plupart des instructions de la page ODBC monitoring s'appliquent afin d'obtenir une règle de découverte « Database monitor » fonctionnelle.

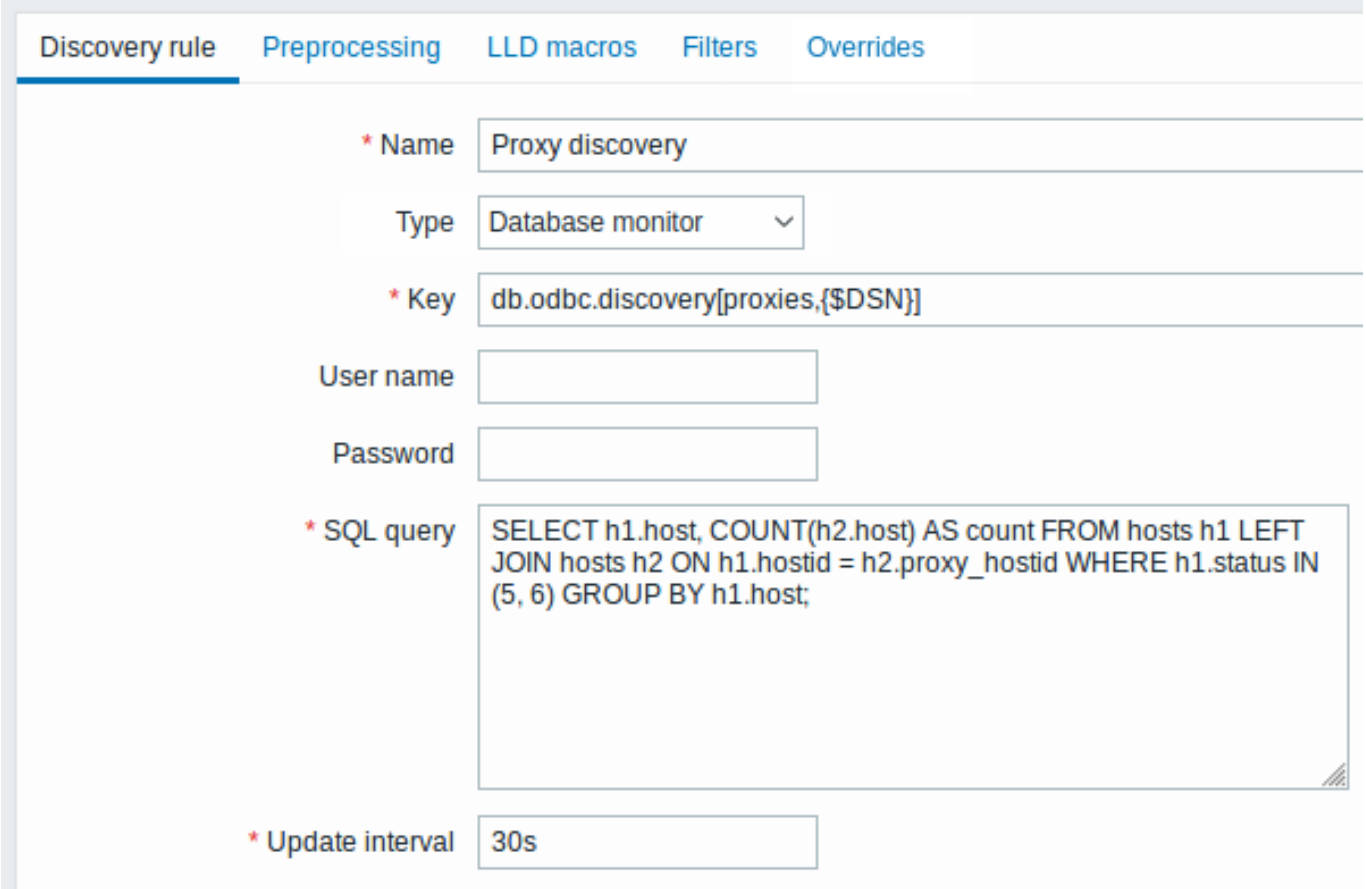
Deux clés d'élément peuvent être utilisées dans les règles de découverte « Database monitor » :

- **db.odbc.discovery**[<description courte unique>,<dsn>,<chaîne de connexion>] - cet élément transforme le résultat de la requête SQL en un tableau JSON, en convertissant les noms de colonnes du résultat de la requête en noms de macros de découverte de bas niveau, associés aux valeurs des champs découverts. Ces macros peuvent être utilisées pour créer des prototypes d'éléments, de déclencheurs, etc. Voir aussi : Utilisation de db.odbc.discovery.
- **db.odbc.get**[<description courte unique>,<dsn>,<chaîne de connexion>] - cet élément transforme le résultat de la requête SQL en un tableau JSON, en conservant les noms de colonnes d'origine du résultat de la requête comme noms de champs dans le JSON, associés aux valeurs découvertes. Comparé à `db.odbc.discovery[]`, cet élément ne crée pas de macros de découverte de bas niveau dans le JSON renvoyé ; il n'est donc pas nécessaire de vérifier si les noms de colonnes peuvent être des noms de macros valides. Les macros de découverte de bas niveau peuvent être définies comme étape supplémentaire selon les besoins, à l'aide de la fonctionnalité custom LLD macro avec JSONPath pointant vers les valeurs découvertes dans le JSON renvoyé. Voir aussi : Utilisation de db.odbc.get.

Utilisation de db.odbc.discovery

L'exemple suivant montre comment une requête SQL est transformée en JSON à l'aide de la découverte de bas niveau des proxies Zabbix, sur la base d'une requête ODBC sur la base de données Zabbix. Cela est utile pour la création automatique d'éléments internes "zabbix[proxy,<name>,lastaccess]" afin de surveiller quels proxies sont actifs.

Commencez par la configuration de la règle de découverte :



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

Ici, la requête directe suivante sur la base de données Zabbix est utilisée pour sélectionner tous les proxies Zabbix, ainsi que le nombre d'hôtes qu'ils surveillent. Le nombre d'hôtes peut être utilisé, par exemple, pour exclure les proxies vides :

```
mysql> SELECT h1.host, COUNT(h2.host) AS count FROM hosts h1 LEFT JOIN hosts h2 ON h1.hostid = h2.proxyid
+---------+-------+
| host    | count |
+---------+-------+
| Japan 1 |     5 |
| Japan 2 |    12 |
| Latvia  |     3 |
+---------+-------+
3 rows in set (0.01 sec)
```

Par le fonctionnement interne de l'élément "db.odbc.discovery[,{$DSN}]", le résultat de cette requête est automatiquement transformé en JSON suivant :

```
[
    {
        "{#HOST}": "Japan 1",
        "{#COUNT}": "5"
    },
    {
        "{#HOST}": "Japan 2",
        "{#COUNT}": "12"
    },
    {
        "{#HOST}": "Latvia",
        "{#COUNT}": "3"
    }
```

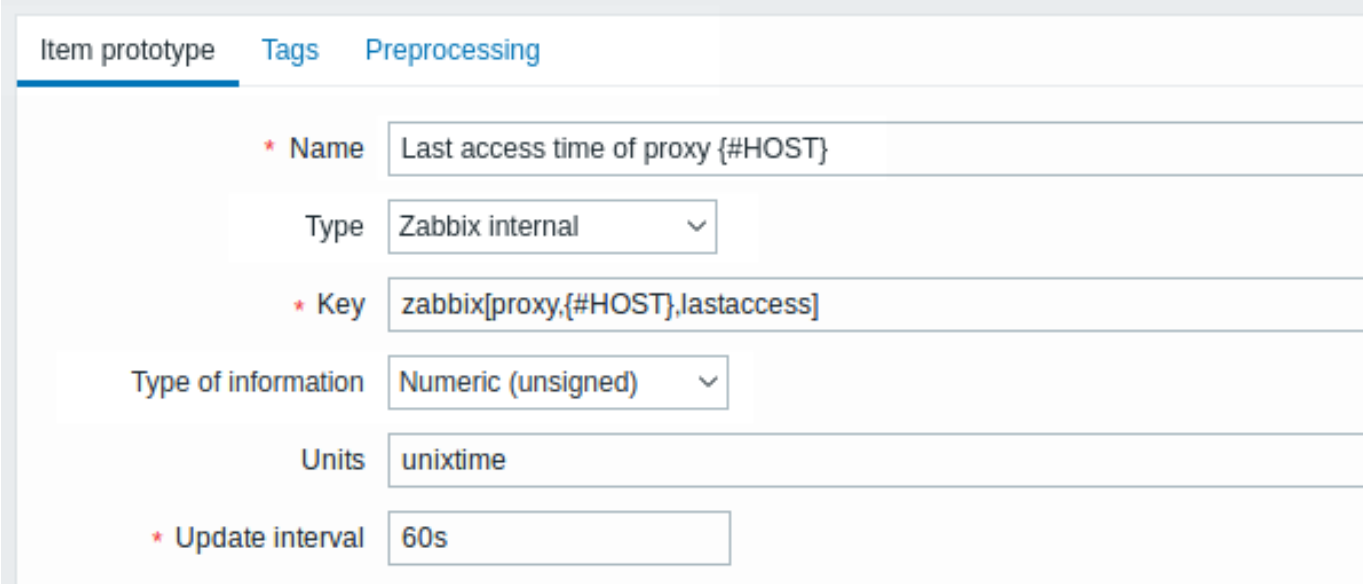
```
]
```

On peut voir que les noms de colonnes deviennent des noms de macros et que les lignes sélectionnées deviennent les valeurs de ces macros.

Note:
S'il n'est pas évident de savoir comment un nom de colonne sera transformé en nom de macro, il est recommandé d'utiliser des alias de colonnes comme "COUNT(h2.host) AS count" dans l'exemple ci-dessus.
Si un nom de colonne ne peut pas être converti en un nom de macro valide, la règle de découverte devient non prise en charge, avec un message d'erreur détaillant le numéro de la colonne en cause. Si une aide supplémentaire est souhaitée, les noms de colonnes obtenus sont fournis avec DebugLevel=4 dans le fichier journal du serveur Zabbix :

```
$ grep db.odbc.discovery /tmp/zabbix_server.log
 ...
 23876:20150114:153410.856 In db_odbc_discovery() query:'SELECT h1.host, COUNT(h2.host) FROM hosts h1 L
 23876:20150114:153410.860 db_odbc_discovery() column[1]:'host'
 23876:20150114:153410.860 db_odbc_discovery() column[2]:'COUNT(h2.host)'
 23876:20150114:153410.860 End of db_odbc_discovery():NOTSUPPORTED
 23876:20150114:153410.860 Item [Zabbix server:db.odbc.discovery[proxies,{$DSN}]] error: Cannot convert
```

Maintenant que nous comprenons comment une requête SQL est transformée en objet JSON, nous pouvons utiliser la macro {#HOST} dans les prototypes d'éléments :



Une fois la découverte effectuée, un élément sera créé pour chaque proxy :



Utilisation de db.odbc.get

En utilisant `db.odbc.get[,{$DSN}]` et l'exemple SQL suivant :

```
mysql> SELECT h1.host, COUNT(h2.host) AS count FROM hosts h1 LEFT JOIN hosts h2 ON h1.hostid = h2.proxyid
+---------+-------+
| host    | count |
+---------+-------+
| Japan 1 |     5 |
| Japan 2 |    12 |
| Latvia  |     3 |
```

```
+---------+-------+
3 rows in set (0.01 sec)
```

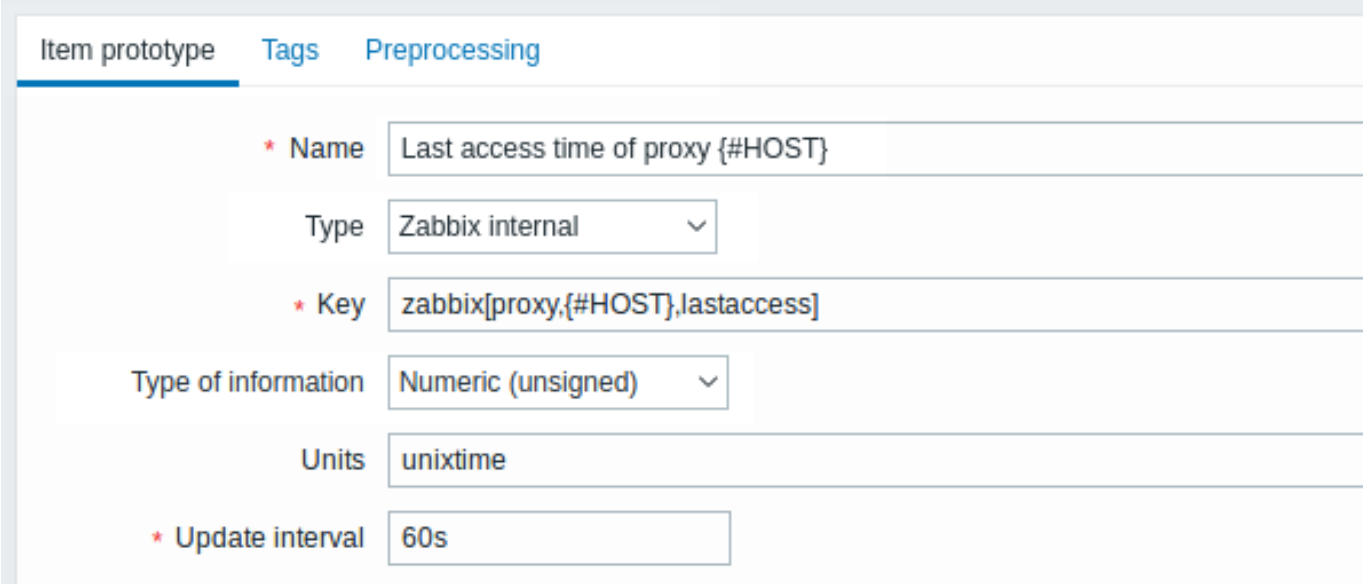
ce JSON sera renvoyé :

```
[
    {
        "host": "Japan 1",
        "count": "5"
    },
    {
        "host": "Japan 2",
        "count": "12"
    },
    {
        "host": "Latvia",
        "count": "3"
    }
]
```

Comme vous pouvez le voir, il n'y a pas de macros de découverte de bas niveau. Cependant, des macros personnalisées de découverte de bas niveau peuvent être créées dans l'onglet Macros LLD d'une règle de découverte à l'aide de JSONPath, par exemple :

```
{#HOST} → $.host
```

Cette macro {#HOST} peut maintenant être utilisée dans les prototypes d'élément :



13 Découverte à l'aide des données Prometheus

Aperçu

Les données fournies au format de ligne Prometheus peuvent être utilisées pour la découverte de bas niveau.

Voir les vérifications Prometheus pour plus de détails sur la façon dont l'interrogation des données Prometheus est implémentée dans Zabbix.

Configuration

La règle de découverte de bas niveau doit être créée en tant qu'élément dépendant de l'élément principal HTTP qui collecte les données Prometheus.

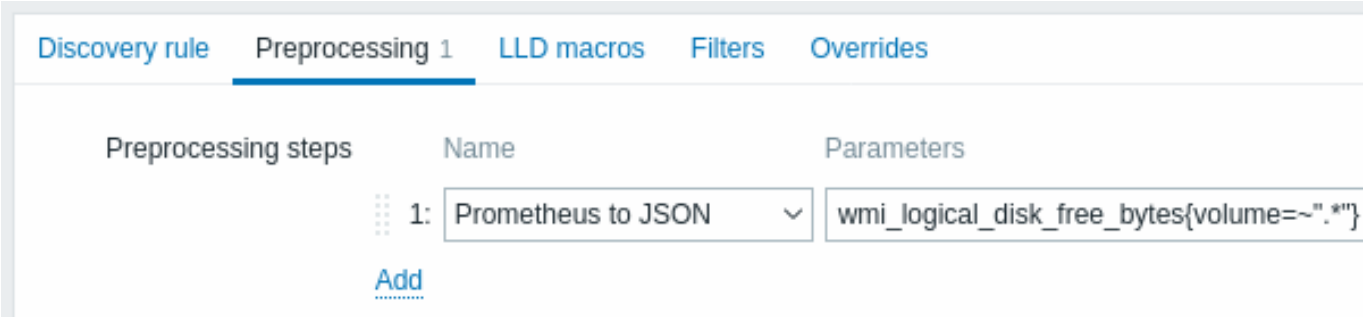
Prometheus vers JSON

Dans la règle de découverte, accédez à l'onglet Prétraitement et sélectionnez l'option de prétraitement *Prometheus vers JSON*. Les données au format JSON sont nécessaires pour la découverte et l'option de prétraitement *Prometheus vers JSON* renverra exactement cela, avec les attributs suivants :

- nom de la métrique

- valeur métrique
- aide (si présente)
- type (si présent)
- étiquettes (si présentes)
- ligne brute

Par exemple, en interrogeant `wmi_logical_disk_free_bytes` :



à partir de ces lignes Prometheus :

```
# HELP wmi_logical_disk_free_bytes Free space in bytes (LogicalDisk.PercentFreeSpace)
# TYPE wmi_logical_disk_free_bytes gauge
wmi_logical_disk_free_bytes{volume="C:"} 3.5180249088e+11
wmi_logical_disk_free_bytes{volume="D:"} 2.627731456e+09
wmi_logical_disk_free_bytes{volume="HarddiskVolume4"} 4.59276288e+08
```

renverra :

```
[
    {
        "name": "wmi_logical_disk_free_bytes",
        "help": "Free space in bytes (LogicalDisk.PercentFreeSpace)",
        "type": "gauge",
        "labels": {
            "volume": "C:"
         },
        "value": "3.5180249088e+11",
        "line_raw": "wmi_logical_disk_free_bytes{volume=\"C:\"} 3.5180249088e+11"
    },
    {
        "name": "wmi_logical_disk_free_bytes",
        "help": "Free space in bytes (LogicalDisk.PercentFreeSpace)",
        "type": "gauge",
        "labels": {
            "volume": "D:"
         },
        "value": "2.627731456e+09",
        "line_raw": "wmi_logical_disk_free_bytes{volume=\"D:\"} 2.627731456e+09"
    },
    {
        "name": "wmi_logical_disk_free_bytes",
        "help": "Free space in bytes (LogicalDisk.PercentFreeSpace)",
        "type": "gauge",
        "labels": {
            "volume": "HarddiskVolume4"
         },
        "value": "4.59276288e+08",
        "line_raw": "wmi_logical_disk_free_bytes{volume=\"HarddiskVolume4\"} 4.59276288e+08"
    }
]
```

Correspondance des macros LLD

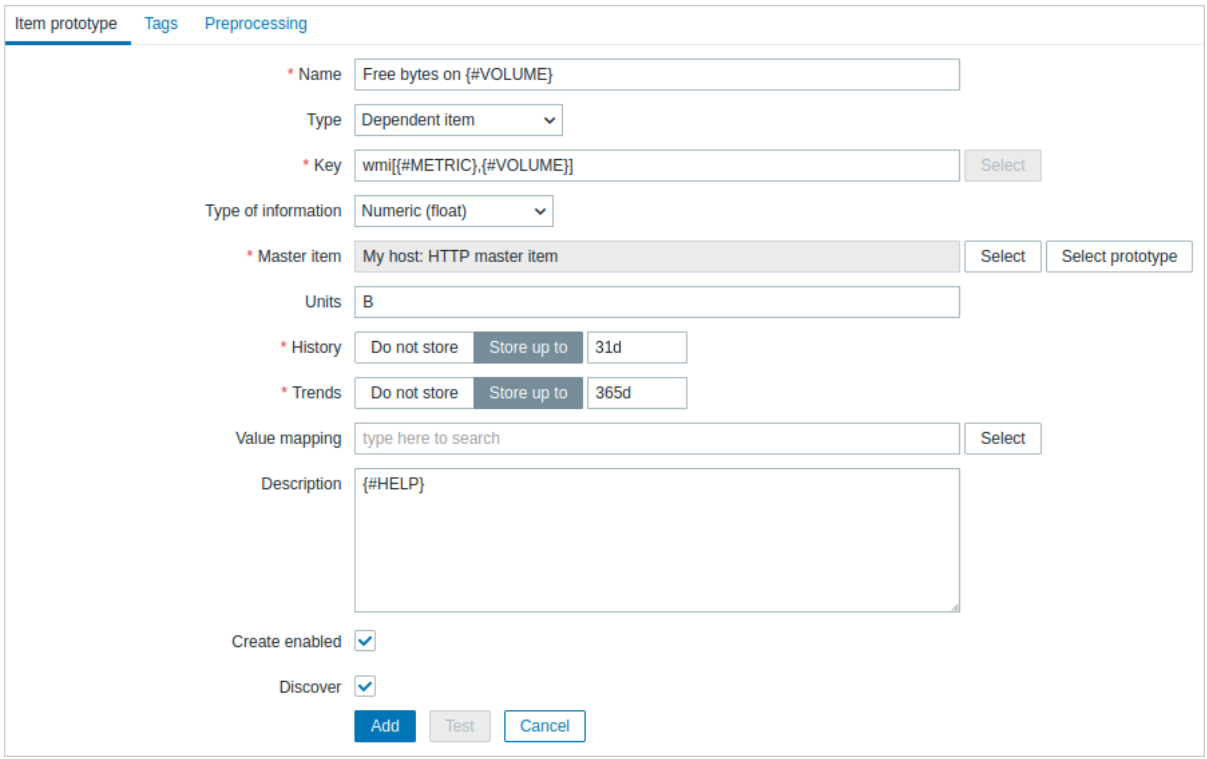
Ensuite, vous devez vous rendre dans l'onglet Macros LLD et effectuer les correspondances suivantes :

`{#VOLUME}=$.labels['volume']`

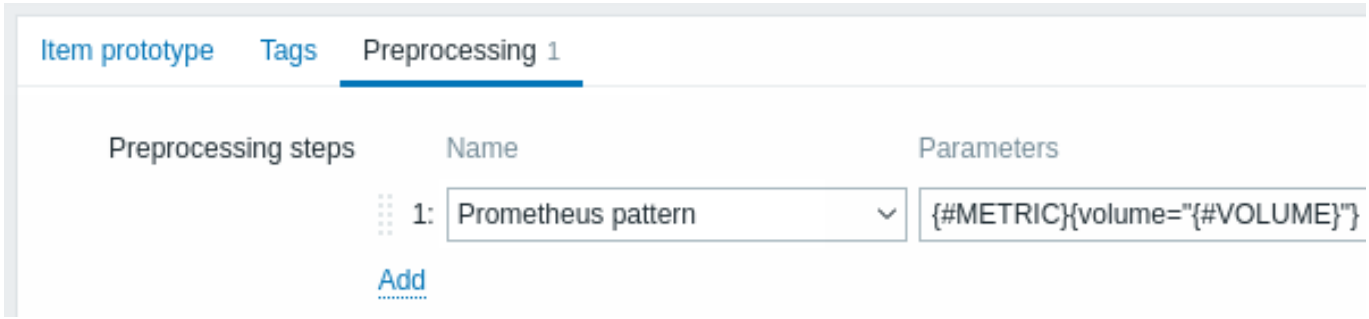
```
{#METRIC}=$['name']
{#HELP}=$['help']
```

Prototype d'élément

Vous pouvez créer un prototype d'élément comme celui-ci :



avec les options de prétraitement :



14 Découverte des périphériques de bloc

De manière similaire à la découverte des systèmes de fichiers, il est également possible de découvrir les périphériques de bloc ainsi que leur type.

Clé d'élément

La clé d'élément à utiliser dans la règle de découverte est

`vfs.dev.discovery`

Cet élément n'est pris en charge que sur les plateformes Linux.

Vous pouvez créer des règles de découverte à l'aide de cet élément de découverte et :

- filtre : **{#DEVNAME} correspond à** `sd[\D]$` - pour découvrir les périphériques nommés "sd0", "sd1", "sd2", ...
- filtre : **{#DEVTYPE} correspond à** `disk` **ET {#DEVNAME} ne correspond pas à** `^loop.*` - pour découvrir les périphériques de type disque dont le nom ne commence pas par "loop"

Macros supportées

Cette clé de découverte renvoie deux macros - {#DEVNAME} et {#DEVTYPE} identifiant respectivement le nom et le type du périphérique bloc, par exemple :

```
[
   {
      "{#DEVNAME}":"loop1",
      "{#DEVTYPE}":"disk"
   },
   {
      "{#DEVNAME}":"dm-0",
      "{#DEVTYPE}":"disk"
   },
   {
      "{#DEVNAME}":"sda",
      "{#DEVTYPE}":"disk"
   },
   {
      "{#DEVNAME}":"sda1",
      "{#DEVTYPE}":"partition"
   }
]
```


La découverte de périphériques en mode bloc permet d'utiliser les éléments `vfs.dev.read[]` et `vfs.dev.write[]` pour créer des prototypes d'éléments à l'aide de la macro {#DEVNAME}, par exemple :

- "vfs.dev.read[{#DEVNAME},sps]"
- "vfs.dev.write[{#DEVNAME},sps]"

{#DEVTYPE} est destiné au filtrage des périphériques.

15 Découverte des interfaces d'hôte dans Zabbix

Aperçu

Il est possible de découvrir toutes les interfaces configurées dans l'interface web de Zabbix pour un hôte.

Clé d'élément

L'élément à utiliser dans la règle de découverte est l'élément interne Zabbix :

`zabbix[host,discovery,interfaces]`

Cet élément renvoie un JSON avec la description des interfaces, notamment :

- l'adresse IP/le nom d'hôte DNS (selon le paramètre d'hôte « Connect to »)
- le numéro de port
- le type d'interface (agent Zabbix, SNMP, JMX, IPMI)
- si c'est l'interface par défaut ou non
- si la fonctionnalité de requête groupée est activée — uniquement pour les interfaces SNMP.

Par exemple :

`[{"{#IF.CONN}":"192.168.3.1","{#IF.IP}":"192.168.3.1","{#IF.DNS}":"","{#IF.PORT}":"10050","{#IF.TYPE}":"AG`

Avec plusieurs interfaces, leurs enregistrements dans le JSON sont ordonnés selon :

- le type d'interface,
- le statut par défaut — l'interface par défaut est placée avant les interfaces non par défaut,
- l'ID de l'interface (par ordre croissant).

Macros prises en charge

Les macros suivantes peuvent être utilisées dans le filtre de la règle de découverte et dans les prototypes d'éléments, de déclencheurs et de graphiques :

Macro	Description
{#IF.CONN}	Adresse IP de l'interface ou nom d'hôte DNS.
{#IF.IP}	Adresse IP de l'interface.
{#IF.DNS}	Nom d'hôte DNS de l'interface.
{#IF.PORT}	Numéro de port de l'interface.
{#IF.TYPE}	Type d'interface (« AGENT », « SNMP », « JMX » ou « IPMI »).

Macro	Description
{#IF.DEFAULT}	Statut par défaut de l'interface : 0 - interface non par défaut 1 - interface par défaut
{#IF.SNMP.BULK}	Statut du traitement SNMP bulk pour l'interface : 0 - désactivé 1 - activé Cette macro n'est renvoyée que si le type d'interface est « SNMP ».

8 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, 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 seulement (par exemple, deux pour la découverte des systèmes de fichiers), il est possible d'en renvoyer davantage.

Exemple

Le format requis pour la chaîne JSON est plus facile à comprendre à l'aide d'un exemple. Supposons que vous utilisiez 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 (reformatée 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 de ce type :

```
####!/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 (reformatée 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 peut aussi être 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.

14 Supervision distribuée

Aperçu Zabbix fournit un moyen efficace et fiable de surveiller une infrastructure informatique distribuée à l'aide de serveurs proxys Zabbix.

Les proxys peuvent être utilisés pour collecter des données localement pour le compte d'un serveur Zabbix centralisé, puis rapporter les données au serveur.

Fonctionnalités du proxy

Lors du choix d'utiliser ou non un proxy, plusieurs éléments doivent être pris en compte.

	Proxy
Léger	**Oui**
Interface graphique	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 unidirectionnelles	**Oui**
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.

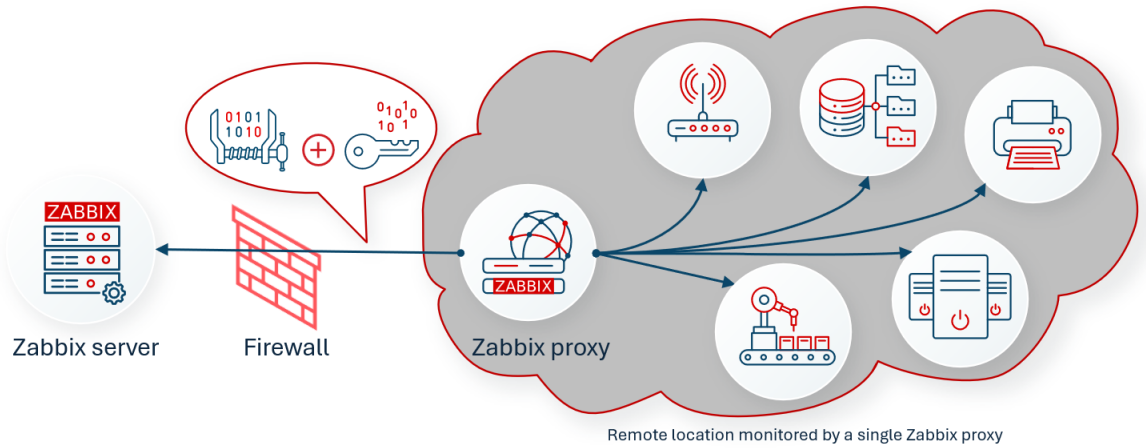
1 Proxys

Aperçu Un proxy Zabbix peut collecter des données de performance et de disponibilité pour le compte du serveur Zabbix. De cette manière, un proxy peut prendre en charge une partie de la charge liée à la collecte des données et ainsi soulager le serveur Zabbix.

De plus, l'utilisation d'un proxy est le moyen le plus simple de mettre en œuvre une supervision centralisée et distribuée, lorsque tous les agents et proxies rapportent à un seul serveur Zabbix et que toutes les données sont collectées de manière centralisée.

Un proxy Zabbix peut être utilisé pour :

- Superviser des sites distants
- Superviser des sites avec des communications peu fiables
- Soulager le serveur Zabbix lors de la supervision de milliers d'équipements
- Simplifier la maintenance d'une supervision distribuée



Le proxy ne nécessite qu'une seule connexion TCP au serveur Zabbix. Ainsi, il est plus facile de contourner un pare-feu, car vous n'avez besoin de configurer qu'une seule règle de pare-feu.

Attention:
Le proxy Zabbix doit utiliser une base de données distincte. Le faire pointer vers la base de données du serveur Zabbix endommagera 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 mise à jour aussi rapidement en raison de la valeur de CacheUpdateFrequency. Par conséquent, le proxy peut commencer à collecter des données et à les envoyer au serveur Zabbix, qui ignore alors 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 un aperçu des fonctionnalités du proxy, consultez le tableau suivant :

Fonction		Pris en charge par le proxy
Éléments		
	Contrôles Zabbix agent	**Oui**
	Contrôles Zabbix agent (actifs)	**Oui** [1]
	Contrôles simples	**Oui**
	Éléments trapper	**Oui**
	Contrôles SNMP	**Oui**
	Traps SNMP	**Oui**
	Contrôles IPMI	**Oui**
	Contrôles JMX	**Oui**
	Surveillance des fichiers journaux	**Oui**
	Contrôles internes	**Oui**
	Contrôles SSH	**Oui**
	Contrôles Telnet	**Oui**
	Contrôles externes	**Oui**
	Éléments dépendants	**Oui**
	Éléments de script	**Oui**
	Éléments de navigateur	**Oui**
Supervision web intégrée		**Oui**
Prétraitement des valeurs d'élément		**Oui**
Découverte réseau		**Oui**
Autoréenregistrement des agents actifs		**Oui**
Découverte de bas niveau		**Oui** [2]
Commandes distantes		**Oui**
Calcul des déclencheurs		*Non*
Traitement des événements		*Non*

Fonction	Pris en charge par le proxy
Corrélation des événements	*Non*
Envoi d'alertes	*Non*

Note:
[1] Pour s'assurer qu'un agent interroge le proxy (et non le serveur) pour les contrôles actifs, le proxy doit être indiqué dans le paramètre **ServerActive** du fichier de configuration de l'agent.
 [2] Pour la LLD, le proxy Zabbix collecte et prétraite uniquement les données, puis les envoie au serveur Zabbix pour un traitement ultérieur.

Protection contre la surcharge

Si le serveur Zabbix a été arrêté pendant un certain temps, que les proxys ont collecté beaucoup de données, puis que le serveur redémarre, 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 contrôles étant traités plus lentement qu'ils ne le devraient. Une protection contre ce scénario a été mise en place afin d'éviter les problèmes dus 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 proxys téléversent les données collectées. Pour éviter cela, une limitation des proxys a été ajoutée (elle ne peut actuellement 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 proxys que lorsque l'utilisation du cache d'historique est inférieure à 60 %, en faisant tourner les proxys acceptés. Une fois que l'utilisation du cache d'historique descend en dessous de 20 %, le serveur revient en mode normal.

De plus, en mode normal, le serveur Zabbix limite les proxys individuels qui envoient de très gros paquets (plus de 10 000 enregistrements) si l'utilisation du cache d'historique dépasse 60 %. Cette décision est appliquée au moment précis où le serveur évalue un téléversement de proxy et, par conséquent, peut ne pas toujours être immédiatement visible dans les graphiques d'utilisation du cache d'historique (l'élément interne `zabbix[wcache,history,pused]` et son intervalle de mise à jour peuvent ne pas détecter de brèves pointes).

Ce mode de limitation se poursuivra jusqu'à ce que l'utilisation du cache atteigne à nouveau 80 %, descende à 20 % ou que la liste de limitation soit vide. Dans le premier cas, le serveur cessera à nouveau d'accepter les données des proxys. Dans les deux autres cas, le serveur recommencera à fonctionner normalement, en acceptant les données de tous les proxys.

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 proxys, mais conserve une *liste de limitation* (liste priorisée des proxys à contacter ultérieurement).
Atteint 60 %	Normal, mais prêt à être limité	Peut refuser de très gros téléversements de proxy (plus de 10 k enregistrements) lorsqu'il décide d'accepter ou non les données ; continue d'accepter les autres données des proxys.
Descend à 20 %	Normal	Supprime la liste de limitation et recommence à accepter normalement les données des proxys.

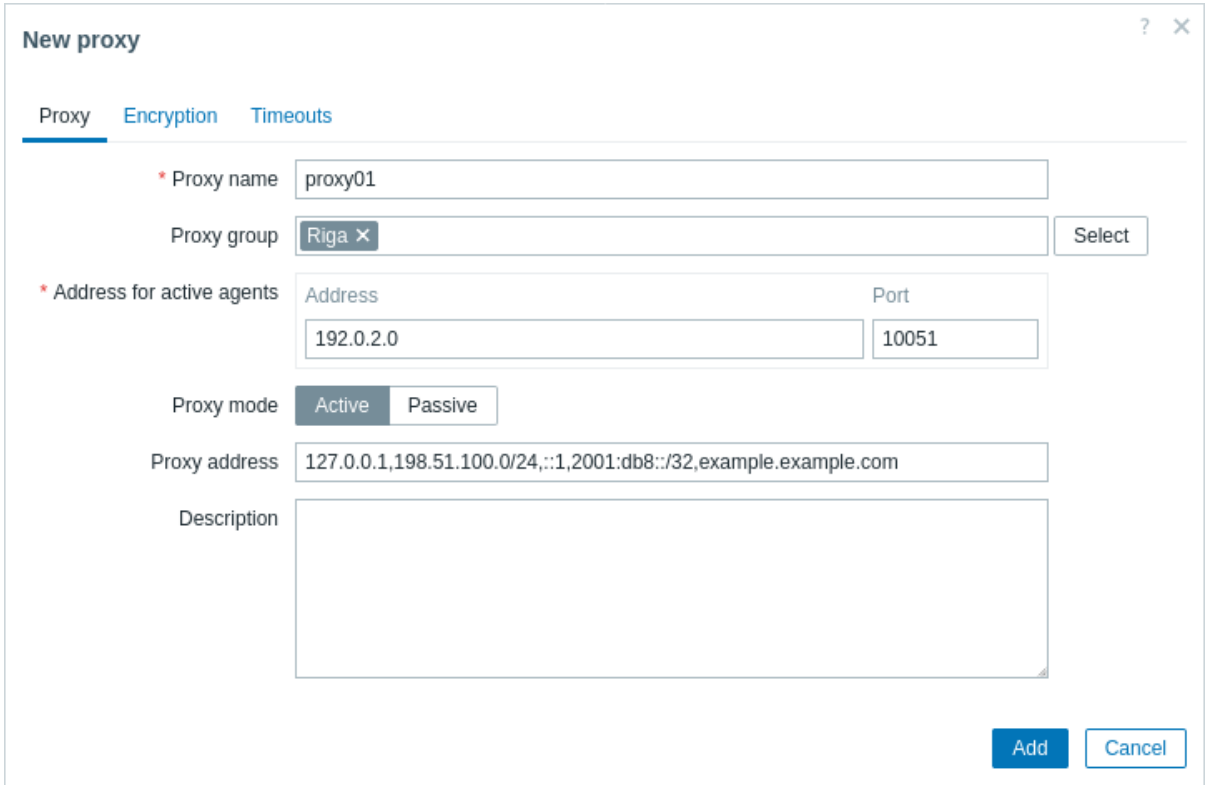
Vous pouvez utiliser l'élément interne zabbix[wcache,history,pused] pour corréler ce comportement du serveur Zabbix avec une métrique.

Configuration Une fois que vous avez installé et configuré un proxy, il est temps de le configurer dans l'interface web de Zabbix.

Ajout de proxys

Pour configurer un proxy dans l'interface web Zabbix :

- Allez à : *Administration > Proxies*
- Cliquez sur *Create proxy*

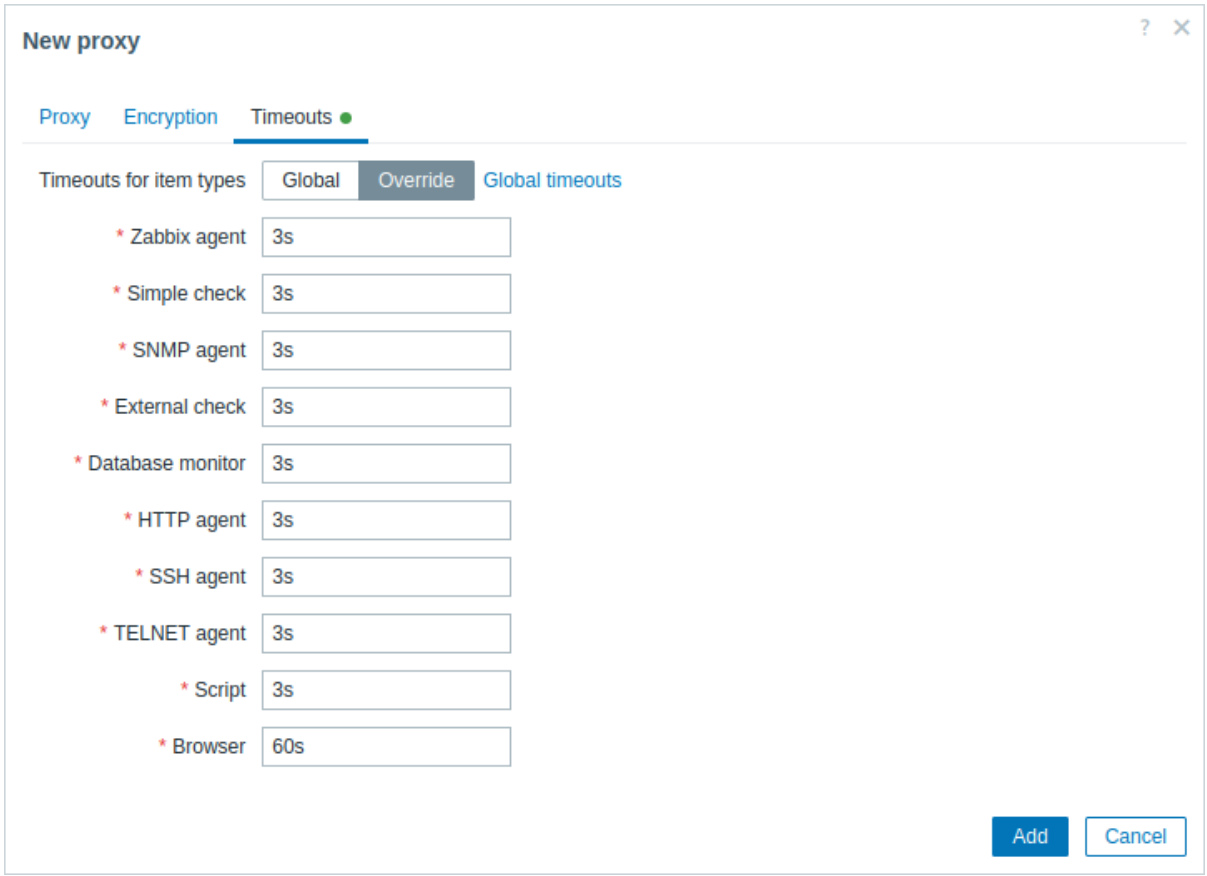


Paramètre	Description
Proxy name	Saisissez le nom du proxy. Il doit être identique au nom indiqué dans le paramètre *Hostname* du fichier de configuration du proxy.
Proxy group	Sélectionnez un groupe de proxys pour l'équilibrage de charge/haute disponibilité du proxy.
Address for active agents	Saisissez l'adresse à laquelle les agents actifs surveillés ou les expéditeurs doivent se connecter. Pris en charge **uniquement** pour les agents Zabbix 7.0 ou version ultérieure. Cette adresse est utilisée pour se connecter aux proxys actifs comme passifs. Ce champ n'est disponible que si un groupe de proxys 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 connectera au serveur Zabbix et demandera les données de configuration **Passive** - le serveur Zabbix se connecte au proxy Veuillez noter que sans communications chiffrées, les données de configuration du proxy (sensibles) peuvent devenir accessibles à des parties 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 si l'authentification n'a pas lieu ou si les adresses de proxy ne sont pas limitées dans le champ *Proxy address*.
Proxy address	Si ce champ est renseigné, les requêtes du proxy actif ne sont acceptées que depuis cette liste d'adresses IP séparées par des virgules, éventuellement en notation CIDR, ou de noms DNS du proxy Zabbix actif. Ce champ n'est disponible que 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 pour un proxy passif. Ce champ n'est disponible que si un proxy passif est sélectionné dans le champ Proxy mode.
Address	Adresse IP/nom DNS du proxy passif.
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.

Paramètre	Description
Connections to proxy	Définit comment le serveur se connecte au proxy passif : sans chiffrement (par défaut), à l'aide d'une PSK (clé pré-partagée) ou d'un certificat.
Connections from proxy	Sélectionnez le type de connexions autorisées depuis le proxy actif. Plusieurs types de connexion peuvent être sélectionnés en même temps (utile pour les tests et le basculement 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 la CA (autorité de certification). S'il est valide et signé par la CA, le champ *Issuer* peut être utilisé pour restreindre davantage les CA autorisées. Ce champ est facultatif et destiné à être utilisé si votre installation Zabbix utilise des certificats provenant de plusieurs CA.
Subject	Sujet autorisé du certificat. Le certificat est d'abord validé avec la CA. S'il est valide et signé par la CA, 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 la CA configurée est accepté.
PSK identity	Chaîne d'identité de la clé pré-partagée. Ne placez pas d'informations sensibles dans l'identité PSK, car elle est transmise sans chiffrement sur le réseau pour informer le destinataire de la PSK à utiliser.
PSK	Clé pré-partagée (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 **Timeouts** vous permet de remplacer les délais d'expiration globaux pour les types d'éléments qui le prennent en charge.



Paramètre	Description
Timeouts for item types	Définissez le délai d'expiration des éléments (en fonction de leur 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* pour chaque type d'élément). Plage autorisée : 1 - 600s (par défaut : hérité des délais d'expiration globaux). Les suffixes de temps, par exemple 30s, 1m, ainsi que les macros utilisateur sont pris en charge. Cliquer sur le lien *Global timeouts* vous permet de configurer les délais d'expiration globaux. Notez que le lien *Global timeouts* n'est visible que pour les utilisateurs de type *Super admin* disposant des autorisations pour la section de l'interface *Administration* > *General*. Notez que, bien que les délais d'expiration au niveau du proxy remplacent les délais globaux, ils seront 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'affichera 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 utilisera 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 spécifier qu'un hôte individuel doit être surveillé par un proxy ou un groupe de proxys dans le formulaire de configuration de l'hôte, en utilisant le champ *Monitored by*.



La mise à jour en masse des hôtes est une autre façon de spécifier que les hôtes doivent être surveillés par un proxy ou un groupe de proxys.

1 Synchronisation de la configuration de supervision

Aperçu

Cette page fournit des détails sur la mise à jour de la configuration de supervision pour le proxy, c'est-à-dire sur la manière dont les modifications apportées à la configuration de supervision sur le serveur sont synchronisées avec 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 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 :

Entité	Détails
données TLS d'auto-enregistrement	Toutes les données TLS d'auto-enregistrement.
expressions	Toutes les expressions (expressions régulières, tables d'expressions).
configuration globale	Configuration globale définie dans la table « config »

Entité	Détails
hôte	Toutes les propriétés, interfaces, inventaire, éléments de données, prétraitement des éléments de données, paramètres des éléments de données, 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 :

- Si un élément de données 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, car elles sont également envoyées si quoi que ce soit sur l'hôte a été modifié.

La commande `-R config_cache_reload` sur le proxy lancera également une mise à jour incrémentielle.

Notez qu'une synchronisation complète de la configuration aura lieu au démarrage/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 de synchronisation de la configuration du proxy avec le serveur (10 secondes par défaut).

Notez que ProxyConfigFrequency est :

- un paramètre du serveur pour les proxys passifs
- un paramètre du proxy pour les proxys actifs

Sur les proxys 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 enregistrera 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.

2 Équilibrage de charge des proxys et haute disponibilité

Aperçu

Les proxys Zabbix peuvent être organisés en groupes de proxys afin d'activer l'équilibrage de charge des proxys et la haute disponibilité.

L'équilibrage de charge des proxys et la haute disponibilité correspondent à la redistribution automatique des hôtes entre les proxys au sein d'un groupe de proxys :

- Si un proxy passe hors ligne, ses hôtes seront déplacés vers d'autres proxys, ce qui permet de maintenir une haute disponibilité des proxys.
- Si un proxy a un nombre d'hôtes beaucoup plus élevé ou plus faible que les autres proxys, ses hôtes seront déplacés vers d'autres proxys afin d'équilibrer la charge des proxys.

La redistribution des hôtes fonctionne uniquement entre les proxys d'un groupe qui remplissent les conditions suivantes :

- Les proxys exécutent Zabbix 7.0 ou une version ultérieure.
- La version du proxy correspond à la version du serveur Zabbix. Si vous utilisez l'agent Zabbix (passif), la version du proxy doit correspondre à celle de l'agent. Les agents actifs nécessitent uniquement Zabbix 7.0 ou une version ultérieure.
- Le groupe de proxys a un état en ligne.
- Les hôtes sont configurés pour être surveillés par un groupe de proxys plutôt que par des proxys individuels.

Note:
L'état de santé du groupe de proxys peut être surveillé à l'aide de contrôles internes par tout hôte attribué à un groupe de proxys. Cependant, pour surveiller l'état de santé d'un seul proxy dans un groupe, attribuez l'hôte à ce proxy ; sinon, les résultats peuvent être incohérents.

Redistribution des hôtes

L'équilibrage de charge des proxys et la haute disponibilité sont gérés par le serveur Zabbix via le gestionnaire de groupes de proxys, qui surveille en continu l'état de tous les proxys de chaque groupe de proxys ainsi que la répartition de leurs hôtes.

La haute disponibilité des proxys au sein d'un groupe est assurée par le basculement des proxys : lorsqu'un proxy devient hors ligne, ses hôtes sont immédiatement redistribués vers d'autres proxys. L'équilibrage de charge des proxys a également lieu, les hôtes étant réaffectés aux proxys ayant le moins d'hôtes assignés.

En outre, l'équilibrage de charge des proxys 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 ou déficit d'hôtes). Si le déséquilibre persiste après une période de grâce (10 x délai de basculement), le groupe de proxys est placé en file d'attente pour la redistribution des hôtes.

Le gestionnaire de groupes de proxys redistribue les hôtes selon la logique suivante :

1. Calculer le nombre moyen d'hôtes par proxy.
2. Pour les proxys présentant un excédent d'hôtes, déplacer les hôtes excédentaires vers le pool non assigné de proxys.
3. Pour les proxys présentant un déficit d'hôtes, calculer combien d'hôtes sont nécessaires pour rétablir l'équilibre.
4. Retirer le nombre requis d'hôtes des proxys ayant le plus grand nombre d'hôtes.
5. Déplacer les hôtes non assignés vers les proxys ayant le moins d'hôtes.

Exemples de redistribution des hôtes :

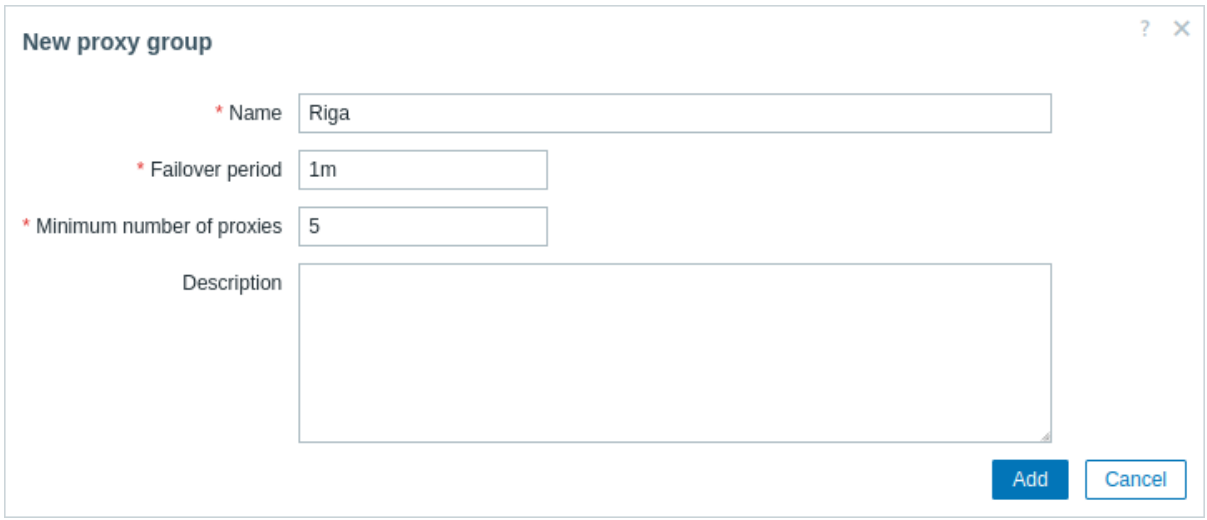
Hôtes sur le proxy	Moyenne du groupe	Réaffectation des hôtes
100	50	Oui
60	50	Non
40	50	Non
25	50	Oui
15	5	Oui
10	5	Non

Attention:
Avoir moins de 10 hôtes surveillés par un groupe de proxys peut entraîner une répartition inégale des hôtes entre les proxys du groupe.

Configuration d'un groupe de proxys

Pour configurer un groupe de proxys dans l'interface web de Zabbix :

1. Allez dans *Administration* > *Groupes de proxys*
2. Cliquez sur *Créer un groupe de proxys*



Paramètre	Description
Nom	Nom du groupe de proxys.
Période de basculement	Période, en secondes, pendant laquelle un proxy du groupe de proxys doit communiquer avec le serveur Zabbix pour être considéré comme en ligne (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 à d'autres proxys. L'équilibrage de charge des proxys commence après 10 x cette période. Prend en charge les suffixes de temps (par exemple, 30s, 1m) et les macros utilisateur.
Nombre minimal de proxys	Nombre minimal de proxys en ligne requis pour maintenir le groupe de proxys en ligne (par défaut : 1 ; plage : 1–1000). Prend en charge les macros utilisateur. Cette valeur doit être inférieure au nombre total de proxys dans le groupe. Par exemple, dans un groupe de 10 proxys, définir le minimum à 10 fera passer le groupe hors ligne si un proxy échoue. Notez que les proxys en ligne dans un groupe hors ligne continuent de fonctionner normalement, mais l'équilibrage de charge/la haute disponibilité n'auront pas lieu.
Description	Description du groupe de proxys.
Proxys	Affiche une liste de cinq proxys maximum (sous forme de liens ou en texte brut, selon les autorisations de l'utilisateur sur les proxys) lors de la modification d'un groupe contenant des proxys.

Configuration de l'équilibrage de charge des proxys

Pour utiliser l'équilibrage de charge des proxys, vous devez configurer un groupe de proxys dans l'interface web Zabbix (voir ci-dessus) et vous assurer que les hôtes sont surveillés par un groupe de proxys, et non par des proxys individuels (vous pouvez utiliser la mise à jour de masse des hôtes pour déplacer des hôtes des proxys vers le groupe de proxys).

Si vous utilisez Zabbix agent, configurez-le également comme suit :

- Pour les contrôles passifs, listez tous les proxys du groupe de proxys dans le paramètre Server.
- Pour les contrôles actifs, il est recommandé de lister tous les proxys du groupe de proxys ou le serveur Zabbix dans le paramètre ServerActive. Notez que seul Zabbix agent 7.0 (ou version ultérieure) fonctionnera avec les groupes de proxys en mode actif.

Si le paramètre `ServerActive` n'inclut qu'un seul proxy d'un groupe de proxys (ou le serveur Zabbix), l'agent pourra tout de même se connecter au proxy correct. Lorsque le service de l'agent démarre et se connecte au proxy spécifié, l'agent reçoit et met en cache la liste complète des IP des proxys ainsi que leur charge actuelle au sein du groupe. Ensuite, les contrôles actifs seront redirigés vers le proxy en ligne correct pour l'hôte, en fonction de l'affectation actuelle hôte-proxy dans le groupe de proxys.

Warning:
Le fait de n'avoir qu'un seul proxy spécifié dans le paramètre `ServerActive` de Zabbix agent peut entraîner une perte de données de supervision si l'agent est démarré/redémarré alors que le proxy spécifié est hors ligne.

Note:
Lors de l'utilisation de Zabbix sender, les requêtes de données sont également redirigées vers le proxy en ligne correct pour l'hôte, en fonction de l'affectation actuelle hôte-proxy dans le groupe de proxys. Cependant, si vous envoyez les valeurs de plusieurs hôtes à partir d'un fichier d'entrée, utilisez l'`option -g` pour éviter d'envoyer des données au mauvais proxy.

Zabbix agent doit également pouvoir se connecter à tous les proxys du groupe de proxys à travers le pare-feu. Sinon, les contrôles actifs peuvent se bloquer ou échouer pendant la redirection ou le basculement. Par exemple :

- Pendant les contrôles actifs, un proxy peut rediriger l'agent vers un autre proxy. Si ce proxy est bloqué par un pare-feu, la communication se bloquera en attendant une réponse.
- Dans des configurations stables à haute disponibilité sans rééquilibrage récent, les agents peuvent ne jamais contacter les proxys de secours. Si les règles du pare-feu ont changé et n'ont pas été testées, le basculement peut échouer.

Test de l'équilibrage de charge des proxies

Pour tester l'équilibrage de charge des proxies :

1. Configurez un groupe de proxies.
2. Assurez-vous que le groupe de proxies a un état en ligne.
3. Assurez-vous que les hôtes sont surveillés par un groupe de proxies, et non par des proxies individuels (vous pouvez utiliser la mise à jour de masse des hôtes pour déplacer les hôtes des proxies vers le groupe de proxies).

4. Attendez quelques secondes pour la mise à jour de la configuration et la distribution des hôtes entre les proxies du groupe de proxies. Observez le changement en actualisant la liste des hôtes dans *Administration* > *Proxies*.

Remarques importantes

- Les traps SNMP ne sont pas pris en charge par les proxys d'un groupe de proxys.
- Les contrôles 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 avoir la même configuration sur tous les proxys du groupe de proxys.
- Les vérifications de base de données nécessitent des autorisations étendues sur l'objet/serveur de base de données.
- Les hôtes VMware surveillés par un groupe de proxys seront répartis aléatoirement entre les proxys du groupe. Cela amène chaque proxy à mettre en cache toutes les données VMware, ce qui entraîne une charge supplémentaire sur vCenter.
- Les hôtes créés à partir des données d'auto-enregistrement provenant d'un proxy d'un groupe de proxys sont configurés pour être surveillés par ce groupe de proxys. Cependant, les hôtes créés à partir des données de découverte réseau provenant d'un proxy d'un groupe de proxys sont configurés pour être surveillés par ce proxy.

15 Chiffrement

Vue d'ensemble 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 sur des clés pré-partagées est pris en charge.

Le chiffrement peut être configuré pour les connexions :

- Entre Zabbix server, Zabbix proxy, Zabbix agent, Zabbix web service, les utilitaires zabbix_sender et zabbix_get
- Vers la base de données Zabbix depuis Zabbix frontend et server/proxy
- Entre Zabbix frontend et Zabbix server

Le chiffrement est facultatif et configurable pour chaque composant individuellement :

- 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 des clés pré-partagées, et d'autres encore peuvent continuer à utiliser des communications non chiffrées (comme auparavant).
- Le serveur (proxy) peut utiliser des configurations de chiffrement différentes pour différents hôtes.

Les programmes daemon Zabbix utilisent un seul port d'écoute pour les connexions entrantes chiffrées et non chiffrées. L'ajout du chiffrement ne nécessite pas l'ouverture 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 pré-partagées sont saisies dans le frontend 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 le frontend 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 des éléments et aux actions, en fonction de 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 supplémentaires sont ajoutées pour établir la connexion TLS.
  - Il peut être nécessaire d'augmenter les délais d'expiration ; 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 des connexions chiffrées.
- Le chiffrement n'est pas pris en charge par la découverte réseau. Les vérifications de l'agent Zabbix effectuées par la découverte réseau ne seront pas chiffrées et, si l'agent Zabbix est configuré pour rejeter les connexions non chiffrées, ces vérifications n'aboutiront pas.

Compilation de Zabbix avec prise en charge du chiffrement Pour prendre en charge le chiffrement, Zabbix doit être compilé et lié avec 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
- LibreSSL - testé avec les versions 2.7.4, 2.8.2 :

- LibreSSL 2.6.x n'est pas pris en charge
- LibreSSL est pris en charge comme remplacement compatible d'OpenSSL ; les nouvelles fonctions d'API `tls_*()` spécifiques à LibreSSL ne sont pas utilisées. Les composants Zabbix compilés avec LibreSSL ne pourront pas utiliser PSK, seuls les certificats pourront être utilisés.

Note:
Vous pouvez en savoir plus sur la configuration de SSL pour le frontend Zabbix en consultant 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 comme :

```
./configure --enable-server --enable-agent --with-mysql --enable-agent --enable-ipv6 --with-net-snmp --wit
```

Différents composants Zabbix peuvent être compilés avec différentes bibliothèques cryptographiques (par exemple, un serveur avec *OpenSSL*, un agent avec *GnuTLS*).

Attention:
Si vous prévoyez d'utiliser des clés pré-partagées (PSK), envisagez d'utiliser les bibliothèques *GnuTLS* ou *OpenSSL 1.1.0* (ou plus récentes) dans les composants Zabbix utilisant des PSK. Les bibliothèques *GnuTLS* et *OpenSSL 1.1.0* prennent en charge les suites cryptographiques PSK avec Perfect Forward Secrecy. Les anciennes versions de la bibliothèque *OpenSSL* (1.0.1, 1.0.2c) prennent également en charge les PSK, mais les suites cryptographiques PSK disponibles n'offrent pas la Perfect Forward Secrecy.

Gestion du chiffrement des connexions Les connexions dans Zabbix peuvent utiliser :

- aucun chiffrement (par défaut)
- le chiffrement basé sur des certificats RSA
- le chiffrement basé sur PSK

Il existe deux paramètres importants utilisés pour spécifier le chiffrement entre les composants Zabbix :

- TLSConnect - spécifie quel chiffrement utiliser pour les connexions sortantes (non chiffrées, PSK ou certificat)
- TLSAccept - spécifie quels types de connexions sont autorisés pour les connexions entrantes (non chiffrées, PSK ou certificat). Une ou plusieurs valeurs peuvent être spécifiées.

`TLSConnect` est utilisé dans les fichiers de configuration de Zabbix proxy (en mode actif, il spécifie uniquement les connexions vers le serveur) et de Zabbix agent (pour les vérifications actives). Dans le frontend Zabbix, l'équivalent de TLSConnect est le champ *Connections to host* dans l'onglet *Data collection* → *Hosts* → *<some host>* → *Encryption* et le champ *Connections to proxy* dans l'onglet *Administration* → *Proxies* → *<some proxy>* → *Encryption*. 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 de Zabbix proxy (en mode passif, il spécifie uniquement les connexions provenant du serveur) et de Zabbix agent (pour les vérifications passives). Dans le frontend Zabbix, l'équivalent de TLSAccept est le champ *Connections from host* dans l'onglet *Data collection* → *Hosts* → *<some host>* → *Encryption* et le champ *Connections from proxy* dans l'onglet *Administration* → *Proxies* → *<some proxy>* → *Encryption*.

Normalement, vous ne configurez qu'un seul type de chiffrement pour les connexions entrantes. Mais vous pouvez vouloir changer le type de chiffrement, par exemple passer d'une connexion non chiffrée à un chiffrement basé sur certificat avec un temps d'arrêt minimal et une possibilité de retour en arrière. Pour y parvenir :

- Définissez `TLSAccept=unencrypted,cert` dans le fichier de configuration de l'agent et redémarrez Zabbix agent
- Testez la connexion avec zabbix_get vers l'agent en utilisant un certificat. Si cela fonctionne, vous pouvez reconfigurer le chiffrement pour cet agent dans le frontend Zabbix, dans l'onglet *Data collection* → *Hosts* → *<some host>* → *Encryption*, en définissant *Connections to host* sur "Certificate".
- 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 supervisé par un proxy), les connexions vers cet agent seront alors chiffrées
- Si tout fonctionne comme prévu, vous pouvez définir `TLSAccept=cert` dans le fichier de configuration de l'agent et redémarrer Zabbix agent. L'agent n'acceptera alors plus que les connexions chiffrées basées sur certificat. Les connexions non chiffrées et basées sur PSK seront rejetées.

De manière similaire, cela fonctionne sur le serveur et le proxy. Si, dans le frontend Zabbix, dans la configuration de l'hôte, *Connections from host* est défini sur "Certificate", alors seules les connexions chiffrées basées sur certificat seront acceptées depuis l'agent (vérifications actives) et zabbix_sender (éléments trapper).

Le plus souvent, vous configurerez les connexions entrantes et sortantes pour utiliser le même type de chiffrement, ou aucun chiffrement. Mais techniquement, il est possible de le configurer de manière asymétrique, par exemple un chiffrement basé sur certificat pour les connexions entrantes et un chiffrement basé sur PSK pour les connexions sortantes.

La configuration du chiffrement pour chaque hôte est affichée dans le frontend Zabbix, dans *Data collection* → *Hosts*, dans la colonne *Agent encryption*. Par exemple :

Example	Connections to host	Allowed connections from host	Rejected connections from host
NONE	Non chiffré	Non chiffré	Chiffré, chiffré basé sur certificat et chiffré basé sur PSK
CERT NONE PSK CERT	Chiffré, basé sur certificat	Chiffré, basé sur certificat	Non chiffré et chiffré basé sur PSK
PSK NONE PSK CERT	Chiffré, basé sur PSK	Chiffré, basé sur PSK	Non chiffré et chiffré basé sur certificat
PSK NONE PSK CERT	Chiffré, basé sur PSK	Non chiffré et chiffré basé sur PSK	Chiffré basé sur certificat
CERT NONE PSK CERT	Chiffré, basé sur certificat	Non chiffré, chiffré basé sur PSK ou basé sur certificat	-

Attention:
Les connexions ne sont pas chiffrées par défaut. Le chiffrement doit être configuré individuellement pour chaque hôte et proxy.

zabbix_get et zabbix_sender avec chiffrement Voir les man-pages zabbix_get et zabbix_sender pour une utilisation avec chiffrement.

Suites de chiffrement Par défaut, les suites de chiffrement sont configurées 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 conformément à leurs politiques de sécurité. L'utilisation de cette fonctionnalité est facultative (les suites de chiffrement intégrées par défaut continuent de fonctionner).

Pour les bibliothèques cryptographiques compilées avec les paramètres par défaut, les règles intégrées de Zabbix aboutissent généralement aux suites de chiffrement suivantes (par ordre de priorité décroissant) :

Bibliothèque	Suites de chiffrement par certificat	Suites de chiffrement PSK
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

Bibliothèque	Suites de chiffrement par certificat	Suites de chiffrement PSK
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 cryptographiques configurées par l'utilisateur Les critères intégrés de sélection des suites cryptographiques peuvent être remplacés par des suites cryptographiques configurées par l'utilisateur.

> **Attention:**
> Les suites cryptographiques configurées par l'utilisateur sont une fonctionnalité destinée aux utilisateurs avancés qui comprennent les suites cryptographiques TLS, leur sécurité et les conséquences des erreurs, et qui sont à l'aise avec le dépannage TLS.

Les critères intégrés de sélection des suites cryptographiques peuvent être remplacés à l'aide des paramètres suivants :

Portée du remplacement	Paramètre	Valeur	Description
Sélection des suites cryptographiques pour les certificats	TLSCipherCert13	Chaînes de chiffrement OpenSSL 1.1.1 valides pour le protocole TLS 1.3 (leurs valeurs sont transmises à la fonction OpenSSL `SSL_CTX_set_ciphersuites()`).	Critères de sélection des suites cryptographiques basés sur les certificats pour TLS 1.3 OpenSSL 1.1.1 ou version plus récente uniquement.
	TLSCipherCert	Chaînes de chiffrement OpenSSL valides pour TLS 1.2 ou chaînes de priorité GnuTLS valides. Leurs valeurs sont transmises respectivement aux fonctions `SSL_CTX_set_cipher_list()` ou `gnutls_priority_init()`.	Critères de sélection des suites cryptographiques basés sur les certificats pour TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)
Sélection des suites cryptographiques pour PSK	TLSCipherPSK13	Chaînes de chiffrement OpenSSL 1.1.1 valides pour le protocole TLS 1.3 (leurs valeurs sont transmises à la fonction OpenSSL `SSL_CTX_set_ciphersuites()`).	Critères de sélection des suites cryptographiques basés sur PSK pour TLS 1.3 OpenSSL 1.1.1 ou version plus récente uniquement.
	TLSCipherPSK	Chaînes de chiffrement OpenSSL valides pour TLS 1.2 ou chaînes de priorité GnuTLS valides. Leurs valeurs sont transmises respectivement aux fonctions `SSL_CTX_set_cipher_list()` ou `gnutls_priority_init()`.	Critères de sélection des suites cryptographiques basés sur PSK pour TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)
Liste combinée des suites cryptographiques pour certificat et PSK	TLSCipherAll13	Chaînes de chiffrement OpenSSL 1.1.1 valides pour le protocole TLS 1.3 (leurs valeurs sont transmises à la fonction OpenSSL `SSL_CTX_set_ciphersuites()`).	Critères de sélection des suites cryptographiques pour TLS 1.3 OpenSSL 1.1.1 ou version plus récente uniquement.

Portée du remplacement	Paramètre	Valeur	Description
	TLSCipherAll	Chaînes de chiffrement OpenSSL valides pour TLS 1.2 ou chaînes de priorité GnuTLS valides. Leurs valeurs sont transmises respectivement aux fonctions `SSL_CTX_set_cipher_list()` ou `gnutls_priority_init()`.	Critères de sélection des suites cryptographiques pour TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)

Pour remplacer la sélection des suites cryptographiques dans les utilitaires zabbix_get et zabbix_sender, utilisez les paramètres de ligne de commande suivants :

- `--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, alors le serveur, le proxy ou l'agent ne démarrera pas et une erreur sera consignée dans le journal.

Il est important de comprendre dans quels cas 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 le paramètre `--tls-connect`)

Connexions entrantes

C'est un peu plus compliqué avec 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

Un certain schéma peut être observé dans les deux tableaux ci-dessus :

- TLSCipherAll et TLSCipherAll13 ne peuvent être spécifiés que si une liste combinée de suites de chiffrement basées sur des 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, le proxy si la bibliothèque cryptographique prend en charge PSK), agent configuré pour accepter à la fois des connexions entrantes basées sur des certificats et sur PSK
- dans les autres cas, TLSCipherCert* et/ou TLSCipherPSK* suffisent

Les tableaux suivants montrent les valeurs par défaut intégrées de `TLSCipher*`. Elles peuvent constituer un bon point de départ pour définir 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 anciennes versions 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 :

- Tester les chaînes de chiffrement et n'autoriser que les suites de chiffrement PFS
- Passer de AES128 à AES256

Test des chaînes de chiffrement et autorisation uniquement des suites cryptographiques PFS

Pour voir quelles suites cryptographiques 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 cryptographiques souhaitées. Il est peu pratique de redémarrer plusieurs fois le serveur, le proxy ou l'agent Zabbix uniquement pour ajuster les paramètres `TLSCipher*`. Des options plus pratiques consistent à utiliser zabbix_sender ou la commande `openssl`. Montrons 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 PSK pour cet exemple. Ajustez les chemins et les noms des fichiers de certificat et de PSK en fonction de votre environnement.

Si vous n'utilisez pas de certificats, mais uniquement une 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 cryptographiques 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
```

Ici, vous voyez les suites cryptographiques sélectionnées par défaut. Ces valeurs par défaut sont choisies pour garantir l'interopérabilité avec les agents Zabbix exécutés sur des systèmes avec des versions plus anciennes d'OpenSSL (à partir de 1.0.1).

Avec des systèmes plus récents, vous pouvez choisir de renforcer la sécurité en n'autorisant qu'un petit nombre de suites cryptographiques, par exemple uniquement des suites cryptographiques avec PFS (Perfect Forward Secrecy). Essayons d'autoriser uniquement des suites cryptographiques avec PFS à l'aide des paramètres `TLSCipher*`.

> **Attention:**
> Le résultat ne sera pas interopérable avec les systèmes utilisant OpenSSL 1.0.1 et 1.0.2, si une 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
```

et testez à nouveau :

```
$ 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 ne contiennent que des suites cryptographiques TLS 1.3 et des suites cryptographiques TLS 1.2 ECDHE-*, comme prévu.

2. TLSCipherAll et TLSCipherAll13 ne peuvent pas être testés avec zabbix_sender ; ils n'affectent pas 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 des suites cryptographiques 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 a un certificat configuré et si l'agent a également une PSK.

Si votre environnement Zabbix utilise uniquement un chiffrement basé sur PSK et aucun certificat, alors un seul suffit :

```
TLSCipherPSK=kECDHEPSK+AES128
```

Maintenant que vous comprenez comment cela fonctionne, vous pouvez tester la sélection des suites cryptographiques même en dehors de Zabbix, avec la commande `openssl`. Testons les trois valeurs de paramètre `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 `openssl ciphers` avec l'option `-V` pour 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
```

Passer de AES128 à AES256

Zabbix utilise AES128 par défaut pour les données. Supposons que vous utilisiez des certificats et que vous souhaitiez passer à AES256, sur OpenSSL 1.1.1.

Ceci peut être réalisé en ajoutant les paramètres respectifs suivants 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 soient utilisées, les paramètres `TLSCipherPSK*` sont également définis pour éviter leurs valeurs par défaut qui incluent des chiffrements moins sécurisés pour une interopérabilité plus large. 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
```

1 Par certificat

Aperçu

Zabbix peut utiliser des certificats RSA au format PEM, signés par une autorité de certification (CA) publique ou interne.

La vérification des certificats est effectuée par rapport à un certificat de 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 configuration 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 attentivement et testez vos extensions de 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 configurer les certificats sur les composants Zabbix.

Paramètre	Obligatoire	Description
TLSCAFile	oui	Chemin complet vers un fichier contenant les certificats de la ou des autorités de certification (CA) de niveau supérieur pour la vérification du certificat du pair. Si vous utilisez une chaîne de certificats comportant plusieurs éléments, placez d'abord les certificats de la ou des CA de niveau inférieur, suivis des certificats de la ou des CA de niveau supérieur. Les certificats de plusieurs CA peuvent être inclus dans un seul fichier.
TLSCRLFile	non	Chemin complet vers un fichier contenant les listes de révocation de certificats (CRL).
TLSCertFile	oui	Chemin complet vers un fichier contenant le certificat. Si vous utilisez une chaîne de certificats comportant plusieurs éléments, placez d'abord le certificat du serveur, du proxy ou de l'agent, suivi des certificats de la ou des CA de niveau inférieur, puis des certificats de la ou des CA de niveau supérieur.
TLSKeyFile	oui	Chemin complet vers un fichier contenant la clé privée. Assurez-vous que ce fichier est lisible uniquement par le Zabbix user en définissant des droits d'accès appropriés.
TLSServerCertIssuer	non	Émetteur autorisé du certificat du serveur.
TLSServerCertSubject	non	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 de configuration pour :

- Zabbix server
- Zabbix proxy
- Zabbix agent

Serveur Zabbix

1. Préparez le fichier de certificat de l'autorité de certification (CA).

Afin de vérifier les certificats des pairs, le serveur Zabbix doit avoir accès au fichier contenant les certificats racine CA auto-signés de plus haut niveau. Par exemple, si des certificats de deux autorités de certification racines 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 dans `/home/zabbix/zabbix_server.crt`.
Le premier certificat est le certificat du serveur Zabbix, suivi du certificat CA 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 ne pas parvenir à établir une connexion chiffrée si les extensions *X509v3 Subject Alternative Name* ou *Netscape Cert Type* sont utilisées. Pour plus d'informations, voir Limitations on using X.509 v3 certificate extensions.

3. Placez la clé privée du serveur Zabbix dans un fichier, par exemple dans `/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 de l'autorité de certification (CA) de niveau supérieur, le certificat/la chaîne de certificats du proxy Zabbix, ainsi que la clé privée, comme décrit dans la section Serveur Zabbix. Ensuite, modifiez 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 Restriction de l'émetteur et du sujet de certificat autorisés.

Les paramètres TLS dans le fichier de configuration final du proxy peuvent ressembler à ceci :

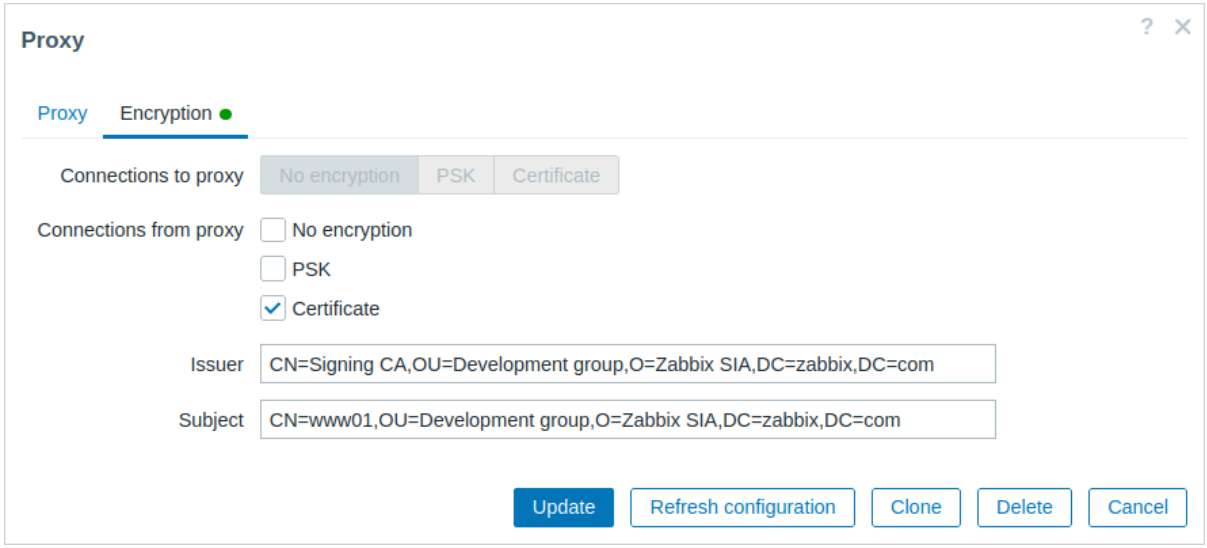
```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix server,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSCertFile=/home/zabbix/zabbix_proxy.crt
TLSKeyFile=/home/zabbix/zabbix_proxy.key
```

3. Configurez le chiffrement pour ce proxy dans le frontend Zabbix :

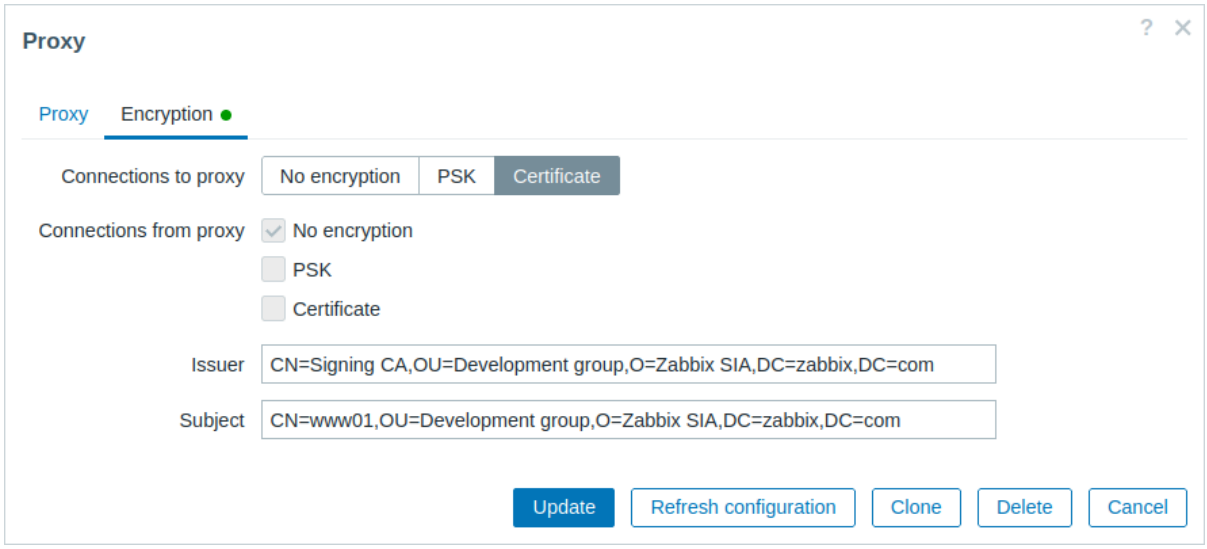
- Accédez à : *Administration* → *Proxies*.
- Sélectionnez le proxy et cliquez sur l'onglet *Encryption*.

Dans les exemples ci-dessous, les champs *Issuer* et *Subject* sont renseignés. Pour plus d'informations sur les raisons et la manière d'utiliser ces champs, consultez Restriction de l'émetteur et du sujet de certificat autorisés.

Pour un proxy actif :



Pour un proxy passif :



Agent Zabbix

1. Préparez les fichiers contenant les certificats CA de niveau supérieur, le certificat/la chaîne de certificats de l'agent Zabbix et la clé privée, comme décrit dans la section Serveur Zabbix. Ensuite, modifiez 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 l'agent actif : `TLSConnect=cert`
- Pour l'agent passif : `TLSAccept=cert`

Note:
Pour améliorer la sécurité de l'agent, vous pouvez définir les paramètres `TLSServerCertIssuer` et `TLSServerCertSubject`. Pour plus d'informations, voir Restriction de l'émetteur et du sujet de certificat autorisés.

Les paramètres TLS dans le fichier de configuration final de l'agent peuvent se présenter comme suit. Notez que l'exemple suppose que l'hôte est surveillé par un proxy ; il est donc spécifié comme sujet du certificat :

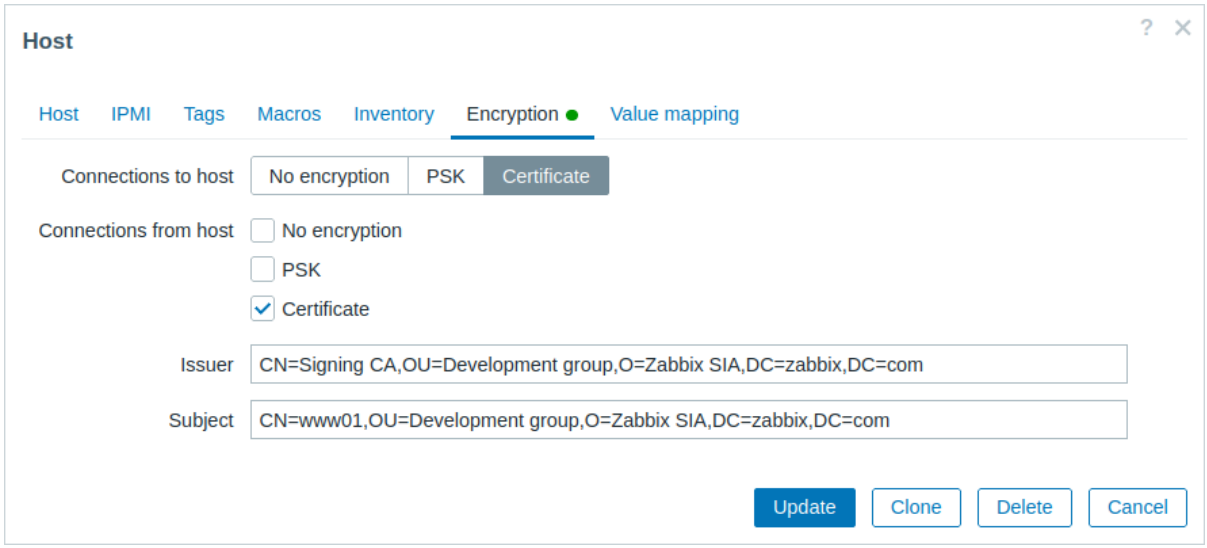
```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix proxy,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSCertFile=/home/zabbix/zabbix_agentd.crt
TLSKeyFile=/home/zabbix/zabbix_agentd.key
```

3. Configurez le chiffrement dans l'interface web Zabbix pour l'hôte surveillé par cet agent.

- Accédez à : *Collecte de données* → *Hôtes*.

- Sélectionnez l'hôte et cliquez sur l'onglet *Chiffrement*.

Dans l'exemple ci-dessous, les champs *Issuer* et *Subject* sont renseignés. Pour plus d'informations sur les raisons et la manière d'utiliser ces champs, voir Restriction de l'émetteur et du sujet de certificat autorisés.



Service web Zabbix

1. Préparez les fichiers contenant les certificats de l'autorité de certification racine, le certificat/la chaîne de certificats du service web Zabbix, ainsi que la clé privée, comme décrit dans la section Zabbix server. Ensuite, modifiez 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 dans le 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 Zabbix server pour se connecter au service web Zabbix configuré avec TLS en modifiant le paramètre `WebServiceURL` dans le fichier de configuration de Zabbix server :

```
WebServiceURL=https://example.com:443/report
```

Restriction de l'émetteur et du sujet de certificat autorisés

Lorsque deux composants Zabbix (par exemple, le serveur et l'agent) établissent une connexion TLS, ils valident mutuellement leurs certificats. Si le certificat du pair est signé par une AC de confiance (avec un certificat racine préconfiguré dans `TLSCAFile`), est valide, n'a pas expiré et satisfait aux autres vérifications, alors la communication entre les composants peut se poursuivre. 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 de petits environnements où les certificats sont signés par une AC interne dédiée et où le risque d'usurpation d'identité est faible, cela peut ne pas être suffisant dans des environnements plus vastes ou plus sensibles en matière de sécurité.

Si votre AC racine émet des certificats qui ne devraient pas être acceptés par Zabbix, ou si vous souhaitez réduire le risque d'usurpation d'identité, 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 a un émetteur ou un sujet différent. De même, un proxy passif n'acceptera pas les requêtes 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. Cependant, les caractères génériques (*) et les expressions régulières ne sont pas pris en charge.
- Les exigences suivantes de la RFC 4514 sont implémentées - caractères nécessitant un échappement (avec une barre oblique inverse '\', U+005C) :
  - n'importe où 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 croisillon ('#', 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 que 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 formatage des champs

Zabbix suit les recommandations de RFC 4514, qui spécifie un ordre « inversé » pour ces champs, en commençant par les champs de niveau le plus bas (CN), puis en passant aux 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 allant du niveau le plus élevé au 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 correspondra 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, exécutez 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 pourra être utilisée dans les fichiers de configuration et l'interface web de Zabbix :

```
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 liées à l'utilisation des extensions de certificat X.509 v3

Lors de la mise en œuvre de certificats X.509 v3 dans Zabbix, certaines extensions peuvent ne pas être entièrement prises en charge ou 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 ne peut valider que la valeur du champ *Subject* du certificat (voir Restriction de l'émetteur et du sujet 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 soit accepter, soit rejeter les 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 de la clé, 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 d'autorité de certification 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 incluant le certificat 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 racine et intermédiaire dans les chaînes de certificats dispose d'une CRL correspondante (même si elle est vide) incluse dans le `TLSCRLFile`.

2 Par clés pré-partagées (PSK)

Aperçu

Chaque clé pré-partagée (PSK) de Zabbix est en fait une paire de :

- chaîne d'identité PSK non secrète,
- valeur de chaîne PSK secrète.

La chaîne d'identité PSK est une chaîne UTF-8 non vide. Par exemple, "PSK ID 001 Zabbix agentd". C'est un nom unique par lequel ce composant PSK spécifique est désigné par les composants Zabbix. Ne placez pas d'informations sensibles dans la chaîne d'identité PSK - elles sont transmises non cryptées sur le réseau.

La valeur PSK est une chaîne de caractères hexadécimaux difficile à deviner, par exemple "e560cb0d918d26d31b4f642181f5f570ad89a39093

Limites de taille

Il existe des limites de taille pour l'identité et la valeur PSK dans Zabbix, dans certains cas, une bibliothèque de chiffrement 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 UTF-8 caractères	128-bit (PSK 16 octets, saisi sous la forme de 32 chiffres hexadécimaux)	2048-bit (PSK de 256 octets, saisi sous la forme de 512 chiffres hexadécimaux)
GnuTLS	128 octets (peut inclure des caractères UTF-8)	-	2048 bits (PSK de 256 octets, entré sous la 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, entré sous la forme de 512 chiffres hexadécimaux)
OpenSSL 1.1.1	127 octets (peut inclure des caractères UTF-8)	-	512 bits (PSK de 64 octets, saisis sous la forme de 128 chiffres hexadécimaux)
OpenSSL 1.1.1a et suivants	127 octets (peut inclure des caractères UTF-8)	-	2 048 bits (PSK de 256 octets, saisis sous la forme de 512 chiffres hexadécimaux)

Attention:
L'interface Zabbix permet de configurer une chaîne d'identité PSK de 128 caractères maximum et une PSK de 2 048 bits, quelles que soient les bibliothèques de chiffrement utilisées.
Si certains composants Zabbix prennent en charge des limites inférieures, il est de la responsabilité de l'utilisateur de configurer l'identité et la valeur PSK avec la 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 de PSK, le serveur recherche l'identité PSK et la valeur PSK configurées pour cet agent dans la base de données (en fait dans le cache de configuration). Lors de la réception d'une connexion, l'agent utilise l'identité PSK et la valeur PSK de son fichier de configuration. Si les deux parties ont la même chaîne d'identité PSK et la même valeur PSK, la connexion peut réussir.

Attention:
Chaque identité PSK doit être associée à une seule valeur. Il est de la responsabilité de l'utilisateur de s'assurer qu'il n'y a pas deux PSK avec la même chaîne d'identité mais des valeurs différentes. Ne pas le faire peut entraîner des erreurs imprévisibles ou des interruptions de communication entre les composants Zabbix utilisant des PSK avec cette chaîne d'identité PSK.

Génération de PSK

Par exemple, un PSK de 256 bits (32 octets) peut être généré 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 la commande "psktool" ci-dessus génère un fichier de base de données avec une identité PSK et son PSK associé. Zabbix ne prévoit qu'un PSK dans le fichier PSK. Par conséquent, la chaîne d'identité et les deux-points (':') doivent être supprimés du fichier.

Configuration de la PSK pour la communication serveur-agent (exemple)

Sur l'hôte de l'agent, écrivez la valeur PSK dans un fichier, par exemple `/home/zabbix/zabbix_agentd.psk`. Le fichier doit contenir la PSK dans la première ligne de texte, par exemple :

```
1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952
```

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_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 une PSK. L'identité PSK sera « PSK 001 ».

Redémarrez l'agent. Vous pouvez maintenant tester la connexion à l'aide de `zabbix_get`, par exemple :

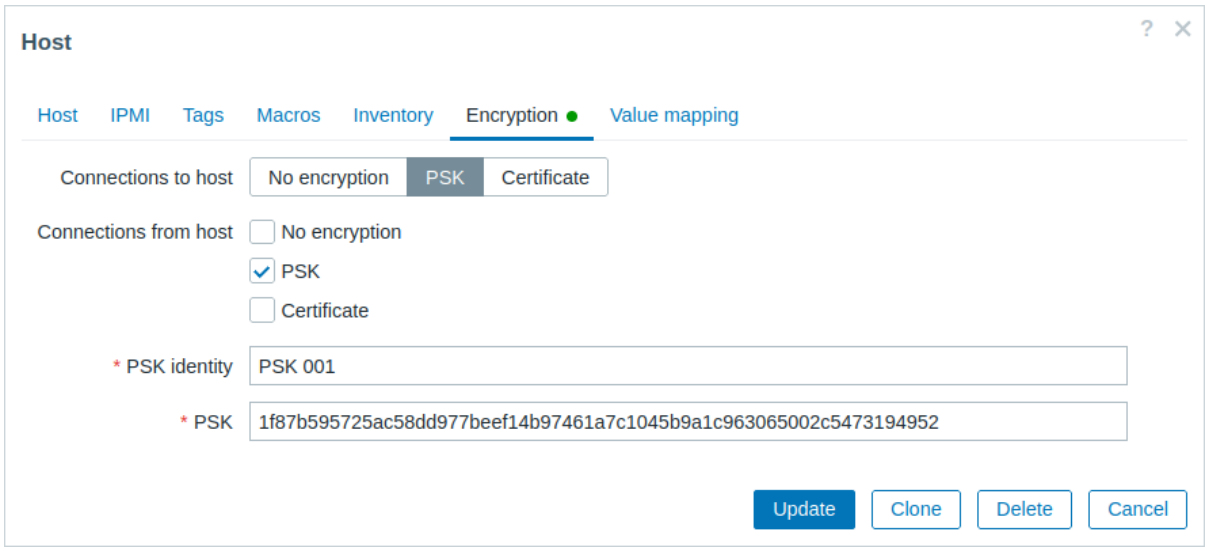
```
zabbix_get -s 127.0.0.1 -k "system.cpu.load[all,avg1]" --tls-connect=psk --tls-psk-identity="PSK 001" --tl
```

(Pour minimiser le temps d'arrêt, consultez comment modifier le type de connexion dans Gestion du chiffrement des connexions).

Configurez le chiffrement PSK pour cet agent dans le frontend Zabbix :

- Allez à : *Collecte de données* → *Hôtes*
- Sélectionnez l'hôte et cliquez sur l'onglet **Chiffrement**

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 la PSK. Vérifiez les fichiers journaux du serveur et de l'agent pour détecter d'éventuels messages d'erreur.

Configuration de PSK pour le serveur - communication d'un proxy actif (exemple)

Sur le proxy, écrivez la valeur 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

Définir les droits d'accès au fichier PSK - il ne doit être lisible que par l'utilisateur Zabbix.

Modifier les paramètres TLS dans le fichier de configuration de proxy `zabbix_proxy.conf`, par exemple, définir :

TLSConnect=psk TLSPSKFile=/home/zabbix/zabbix_proxy.psk TLSPSKIdentity=PSK 002

Le proxy se connecte au serveur en utilisant PSK. L'identité PSK sera "PSK 002".

(Pour minimiser les temps d'interruption, modifier le type de connexion comme décrit dans Gestion du chiffrement de la connexion).

Configurer PSK pour ce proxy dans l'interface Zabbix. Aller dans *Administration* → *Proxies*, sélectionner le proxy et aller dans l'onglet "Chiffrement". Dans "Connexions du proxy", cocher PSK. Coller dans le champ "Identité PSK" "PSK 002" et "e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9" dans le champ "PSK". Cliquez sur "Actualiser".

Redémarrer le proxy. Il commencera à utiliser des connexions au serveur chiffrées basées sur PSK. Vérifiez les fichiers journaux du serveur et du proxy à la recherche de messages d'erreur.

Pour un proxy passif, la procédure est similaire. La seule différence est - définir `TLSAccept=psk` dans le fichier de configuration du proxy et définir "Connexions du proxy" dans l'interface Zabbix à PSK.

3 Dépannage

Recommandations générales

- En cas de problème, commencez par comprendre quel composant agit en tant que client TLS et lequel agit en tant que serveur TLS.
  Le serveur Zabbix, les proxys et les agents, en fonction de leurinteraction, tous peuvent fonctionner en tant que serveurs et clients TLS.
  Par exemple , Le serveur Zabbix se connectant à l'agent pour une vérification passive, agit comme un client TLS. L'agent joue le rôle de serveur TLS.
  L'agent Zabbix, qui demande une liste de vérifications actives à partir du proxy, agit comme un client TLS. Le proxy joue le rôle de serveur TLS. Les utilitaires
  " zabbix_get" et `zabbix_sender` agissent toujours comme des clients TLS.
- Zabbix utilise une authentification mutuelle. Chaque côté vérifie son homologue et peut refuser la connexion.
  Par exemple, le serveur Zabbix qui se connecte à l'agent peut fermer la connexion immédiatement si le certificat de l'agent n'est pas valide. Et vice-versa - l'agent Zabbix acceptant une connexion à partir du serveur peut fermer la connexion si l'agent n'est pas approuvé par le serveur.
- Examinez les fichiers journaux des deux côtés - côté client TLS et côté serveur TLS.
  Le côté qui refuse la connexion peut inscrire une raison précise pour laquelle il l'a refusée. Un autre côté signale souvent une erreur plutôt générale (par exemple "Connexion fermée par un homologue", "la connexion a été interrompue de manière incorrecte").
- Parfois, un chiffrement mal configuré donne lieu à des messages d'erreur déroutants n'indiquant en aucun cas une cause réelle.
  Dans les sous-sections qui suivent, nous essaierons de fournir une liste (loin d'être exhaustive) de messages et de causes possibles pouvant aider au dépannage.
  Les toolkits (OpenSSL, GnuTLS) produisent souvent des messages d'erreur différents dans les mêmes situations problématiques.
  Parfois, les messages d'erreur dépendent même d'une combinaison particulière de kits d'outils cryptographiques de chaque côté.

1 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 les connexions non chiffrées

Dans le journal du serveur ou du proxy (avec *GnuTLS* 3.3.16)

```
Get value from agent failed: zbx_tls_connect(): gnutls_handshake() failed: \
    -110 The TLS connection was non-properly terminated.
```

Dans le journal du serveur ou du proxy (avec *OpenSSL* 1.0.2c)

```
Get value from agent failed: TCP connection successful, cannot establish TLS to [[127.0.0.1]:10050]: \
    Connection closed by peer. Check allowed connection types and access rights
```

Un côté se connecte avec le certificat mais l'autre côté n'accepte que PSK ou vice versa

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'utiliser le sender Zabbix compilé avec le support TLS pour envoyer des données au serveur/proxy Zabbix compilé sans TLS

Dans le journal côté connexion :

Linux :

...In zbx_tls_init_child() ...OpenSSL library (version OpenSSL 1.1.1 11 Sep 2018) initialized ... ...In zbx_tls_connect(): psk_identity:"PSK test sender" ...End of zbx_tls_connect():FAIL error:'connection closed by peer' ...send value error: TCP successful, cannot establish TLS to [[localhost]:10051]: connection closed by peer

Windows :

...OpenSSL library (version OpenSSL 1.1.1a 20 Nov 2018) initialized ... ...In zbx_tls_connect(): psk_identity:"PSK test sender" ...zbx_psk_client_cb() requested PSK identity "PSK test sender" ...End of zbx_tls_connect():FAIL error:'SSL_connect() I/O er-

ror: [0x00000000] The operation completed successfully.' ...send value error: TCP successful, cannot establish TLS to [[192.168.1.2]:10051]: SSL_connect() I/O error: [0x00000000] The operation completed successfully.

Dans le journal côté serveur/proxy :

...failed to accept an incoming connection: from 127.0.0.1: support for TLS was not compiled in

Un côté se connecte à PSK mais un autre utilise LibreSSL ou a été compilé sans support de chiffrement

LibreSSL ne supporte pas PSK.

Dans le journal côté connexion :

...TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_connect() I/O error: [0] Success

Dans journal de l'autre côté :

...failed to accept an incoming connection: from 192.168.1.2: support for PSK was not compiled in

Dans l'interface Web Zabbix :

Get value from agent failed: TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_connect() I/O error: [0] Success

Un côté se connecte à PSK mais un autre utilise OpenSSL avec le support PSK désactivé

Dans le journal côté connexion :

...TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_connect() set result code to SSL_ERROR_SSL: file ../ssl/record/rec_layer_s3.c line 1536: error:14094410:SSL routines:ssl3_read_bytes:sslv3 alert handshake failure: SSL alert number 40: TLS read fatal alert "handshake failure"

Dans le journal de l'autre côté :

...failed to accept an incoming connection: from 192.168.1.2: TLS handshake set result code to 1: file ssl/statem/statem_srvr.c line 1422: error:1417A0C1:SSL routines:tls_post_process_client_hello:no shared cipher: TLS write fatal alert "handshake failure"

2 Problèmes de certificat

OpenSSL utilisé avec les CRL et pour certaines autorités de certification dans la chaîne de certificats, sa CRL n'est pas incluse dans `TLSCRLFile`

Dans le journal du serveur TLS en cas de peer *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 peer *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
```

La CRL a expiré ou expire pendant le fonctionnement du serveur

OpenSSL, dans le journal du serveur :

- avant 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 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 "certificat révoqué". Lorsque la CRL expire, le message d'erreur devient "certificat expiré", ce qui est assez trompeur.

GnuTLS, dans le journal du serveur :

- avant et après expiration :

```
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, AC 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"'
```

Cela a été observé lorsque, par erreur, le certificat du serveur avait la même chaîne Issuer et Subject, bien qu'il ait été signé par une AC. Issuer et Subject sont identiques dans le certificat AC de niveau supérieur, mais ils ne peuvent pas être identiques dans un certificat de serveur. (La même règle s'applique aux certificats de proxy et d'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.

3 Problèmes de PSK

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 :

```
invalid PSK in file "/home/zabbix/zabbix_proxy.psk"
```

Une chaîne d'identité PSK supérieure à 128 octets est transmise à GnuTLS

Dans le journal côté client TLS :

```
gnutls_handshake() failed: -110 The TLS connection was non-properly terminated.
```

Dans le journal 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 côté 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 acceptant la connexion :

```
...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 la forme de 128 chiffres hexadécimaux).

Voir aussi : Limite de taille des valeurs

16 Interface web

Aperçu Pour un accès facile à Zabbix depuis n'importe où et depuis n'importe quelle plate-forme, une interface Web est fournie.

> **Note:**
> Si vous utilisez plus d'une instance frontend, assurez-vous que les paramètres régionaux et les bibliothèques (LDAP, SAML, etc.) sont installés et configurés de manière identique pour tous les frontends.

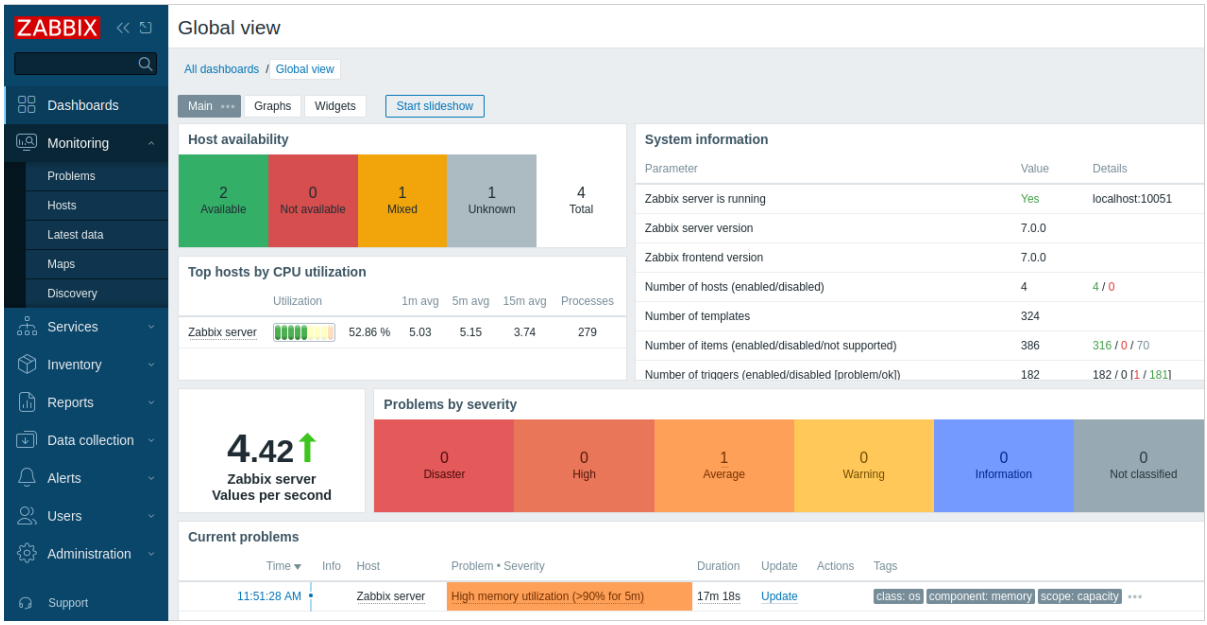
Aide Frontend Un lien d'aide ? est fourni dans les formulaires frontend Zabbix avec des liens directs vers les parties correspondantes de la documentation.

1 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.

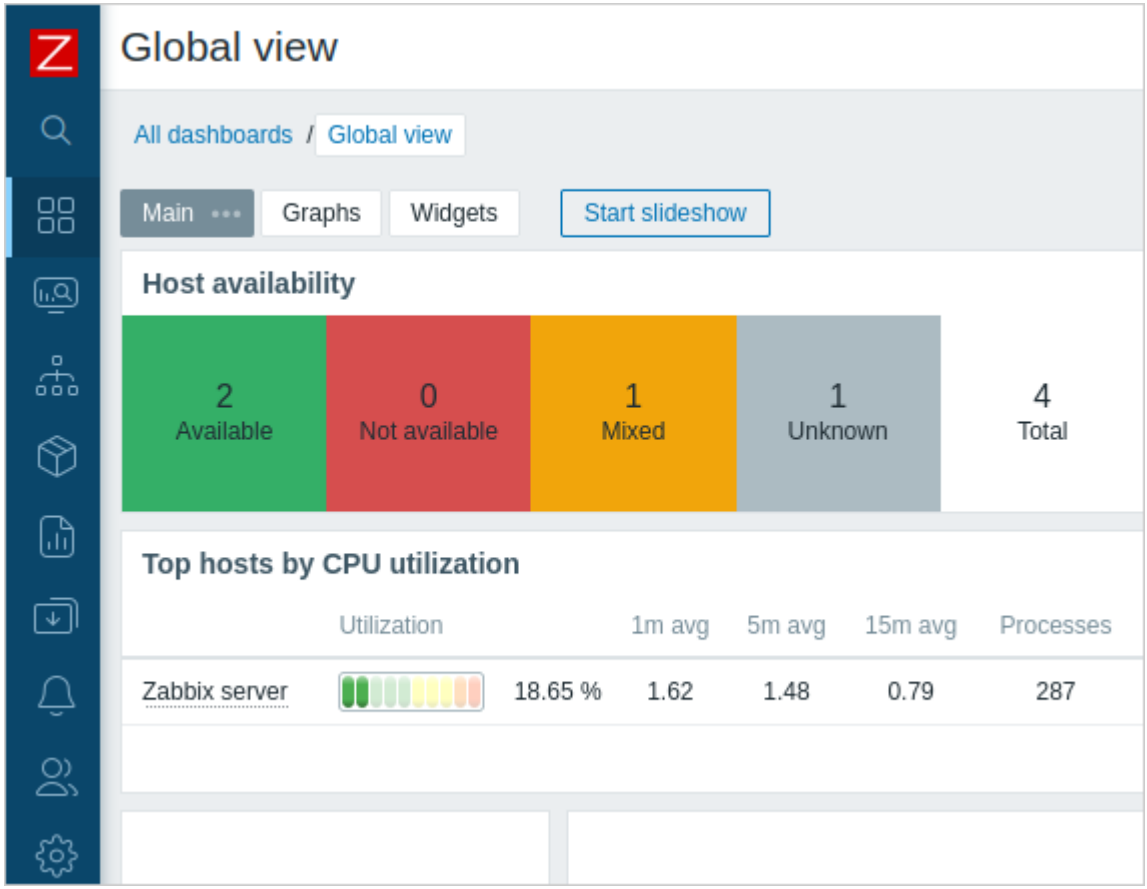


Utilisation du menu

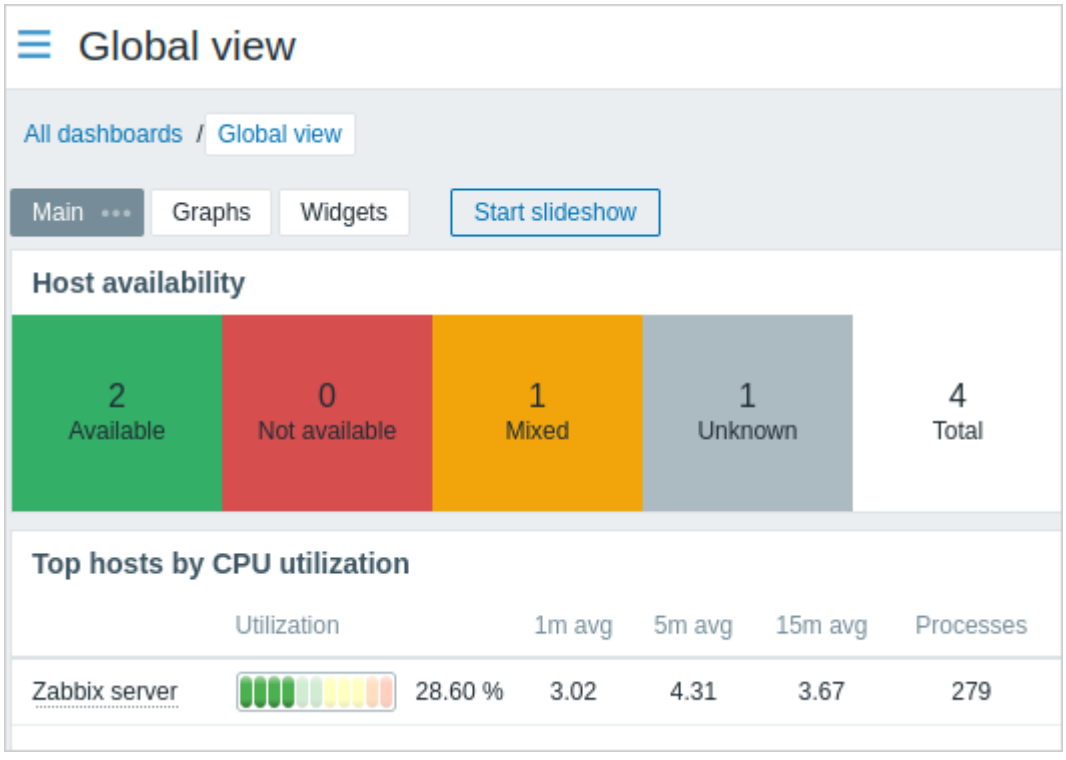
Une zone de recherche globale se trouve sous le logo Zabbix.

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 aux 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'extension. Si le curseur de la souris est à nouveau placé en dehors du menu complet, le menu se réduira de nouveau après deux secondes.

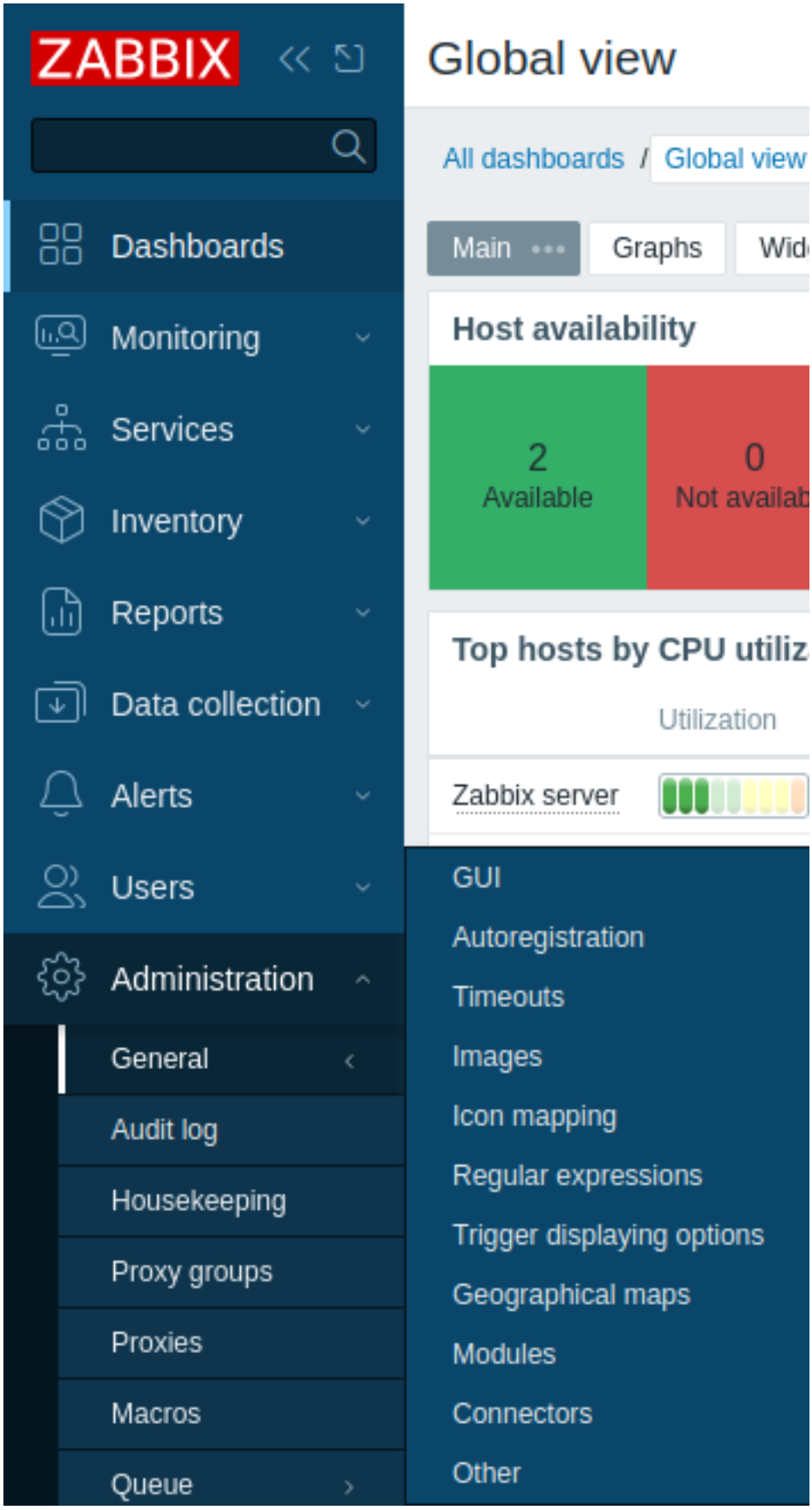
Vous pouvez également faire réapparaître complètement un menu réduit en appuyant sur la touche Tab. En appuyant plusieurs fois sur la touche Tab, vous pourrez placer le focus sur l'élément de menu suivant.

Menu masqué

Même lorsque le menu est complètement masqué, le menu complet reste accessible en un seul clic de souris, en cliquant sur l’icône burger  . Notez qu’il réapparaît par-dessus le contenu de la page ; pour déplacer le contenu de la page vers la droite, vous devez afficher de nouveau le menu en cliquant sur le bouton d’affichage de la barre latérale.

Niveaux de menu

Le menu comporte jusqu’à trois niveaux.



Menus contextuels

En plus du menu principal, Zabbix propose des menus contextuels pour les hôtes, éléments et événements, afin d'accéder rapidement aux entités fréquemment utilisées, telles que les dernières valeurs, un graphique simple, le formulaire de configuration, les scripts associés ou les liens externes.

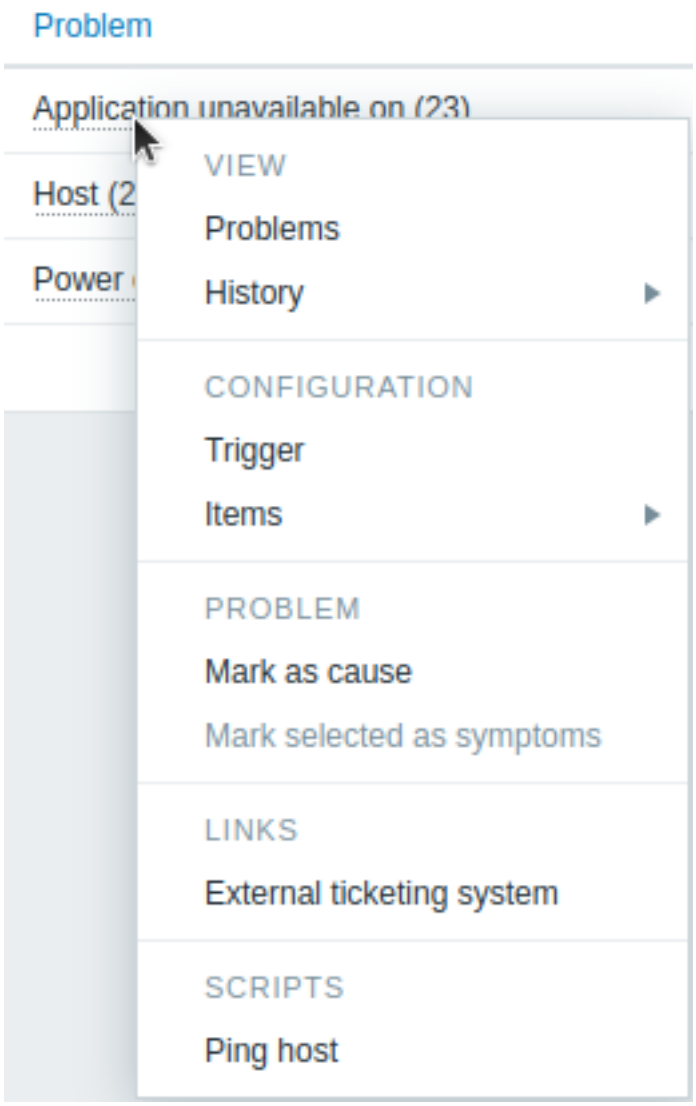
Les menus contextuels sont accessibles en cliquant sur le nom de l'hôte, de l'élément ou du problème/trigger dans les emplacements pris en charge.

1 Menu Événement

Aperçu

Le menu de l'événement contient des raccourcis vers des actions ou des sections de l'interface web qui sont fréquemment nécessaires pour un événement.

Le menu de l'événement peut être ouvert en cliquant sur le nom de l'événement.



Contenu

Le menu contextuel de l'événement comporte six sections : *Vue*, *Actions*, *Configuration*, *Problème*, *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 *Vue* contient des liens vers :

- **Problèmes** - ouvre la liste des problèmes non résolus du déclencheur sous-jacent ;
- **Historique** - mène à l'historique du graphique/de l'élément de *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 *Aperçu 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 de :

- **Déclencheur** qui a généré le 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 permettant de :

- **Marquer comme cause** - marquer le problème comme cause ;
- **Marquer la sélection comme symptômes** - marquer les problèmes sélectionnés comme symptômes de ce problème.

La section *Liens* contient des liens permettant de :

- accéder à une URL de déclencheur configurée ;
- accéder à des liens personnalisés configurés dans les Scripts globaux (avec la portée « Manual event action » et le type « URL ») ;
- accéder à un ticket externe configuré pour le problème (voir l'option *Include event menu entry* lors de la configuration des webhooks).

La section *Scripts* contient des liens permettant d'exécuter un script global (avec la portée *Manual event action*). Cette fonctionnalité peut être utile pour exécuter des scripts utilisés pour gérer des tickets de problème dans des systèmes externes.

Emplacements pris en charge

Le menu contextuel de l'événement est accessible en cliquant sur un problème ou un nom d'événement dans diverses sections de l'interface, par exemple :

- Les widgets des tableaux de bord, tels que le widget *Problems*, le widget *Trigger overview*, etc.
- Supervision → Problèmes
- Supervision → Problèmes → Détails de l'événement
- Rapports → Top 100 triggers (les scripts globaux et l'accès au ticket externe ne sont pas pris en charge à cet emplacement)

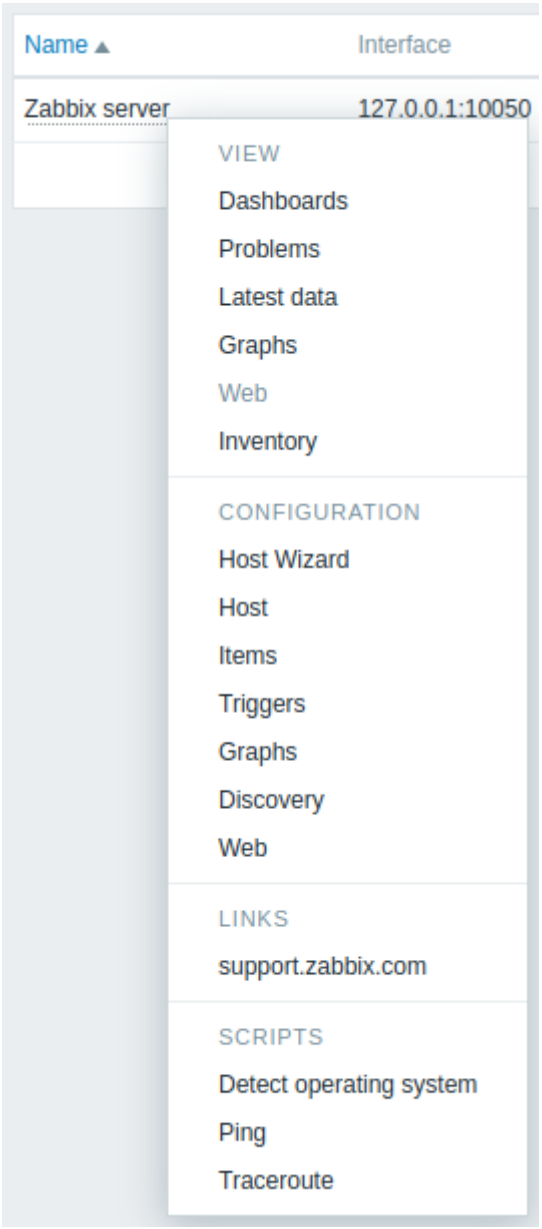
2 Menu Hôte

Aperçu

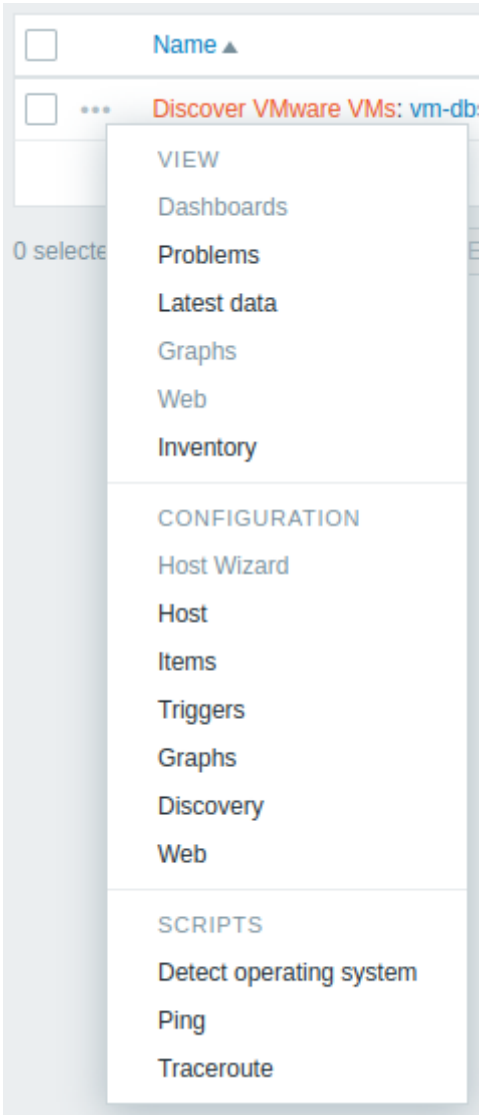
Le menu de l'hôte contient des raccourcis vers des actions ou des sections de l'interface web fréquemment nécessaires pour un hôte.

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 web, par exemple :

Monitoring → *Dernières données*



Collecte de données → *Hôtes*



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 hôte** - ouvre Host Wizard pour l'hôte actuel (désactivé pour les hôtes découverts).
- **Hôte** - formulaire de configuration de l'hôte actuel.
- **Éléments** - la liste des éléments de l'hôte actuel.
- **Déclencheurs** - la liste des déclencheurs de l'hôte actuel.
- **Graphiques** - les graphiques simples de l'hôte actuel.
- **Découverte** - la liste des règles de découverte de bas niveau de l'hôte actuel.

- **Web** - la 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 de :

- accéder à une URL de déclencheur configurée.
- accéder à des liens personnalisés configurés dans les scripts globaux (avec la portée *Manual host action* et le type 'URL').

La section *Scripts* permet d'exécuter des scripts globaux configurés pour l'hôte actuel. Pour être disponibles dans le menu de l'hôte, ces scripts doivent avoir leur portée définie sur *Manual host action*.

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 web, par exemple :

- Les widgets des tableaux de bord, 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.

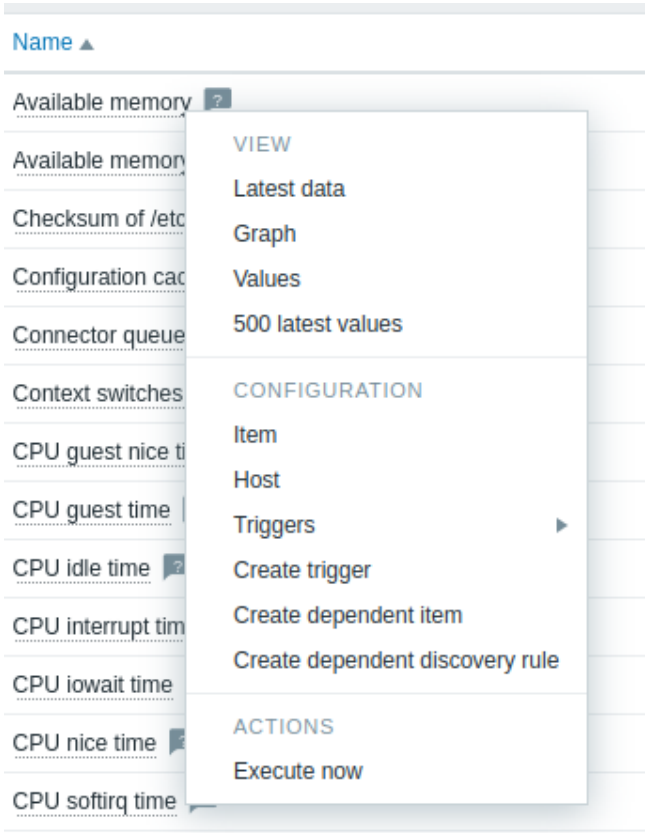
3 Menu des éléments

Aperçu

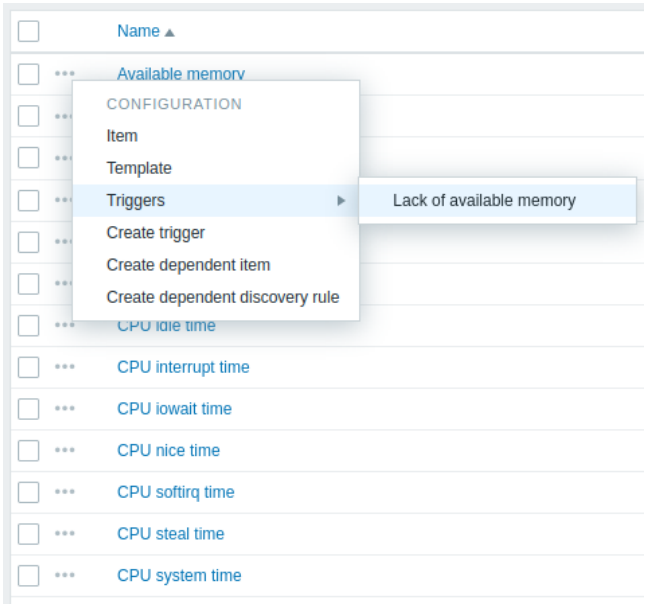
Le menu de l'élément contient des raccourcis vers des actions ou des sections de l'interface qui sont fréquemment nécessaires pour un élément.

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 → Dernières données



Collecte de données → Modèles → Éléments



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 pour l'élément actuel.

La section *Configuration* (disponible uniquement pour les utilisateurs de type *Admin* et *Super admin*) contient les options suivantes :

- **Élément** - ouvre le formulaire de configuration de l'élément de l'élément actuel ;
- **Modèle** - ouvre le formulaire de configuration du modèle du modèle auquel appartient l'élément (disponible uniquement lorsque le menu de l'élément est accessible depuis *Collecte de données → Modèles → Éléments*) ;
- **Hôte** - ouvre le formulaire de configuration de l'hôte de l'hôte auquel appartient l'élément ;
- **Déclencheurs** - au survol de la souris, ouvre une liste des déclencheurs de l'élément, le cas échéant ; cliquer sur un déclencheur ouvre son formulaire de configuration du déclencheur ;
- **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 un contrôle afin d'obtenir immédiatement une nouvelle valeur pour 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 web, par exemple :

- Monitoring → Dernières données
- Collecte de données → Hôtes → Éléments

- Collecte de données → Hôtes → Règles de découverte → Prototypes 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.

2 Sections de l'interface

Structure du menu Le menu de l'interface web Zabbix a la structure suivante :

- Tableaux de bord
- Surveillance
  - Problèmes
  - Hôtes
  - Dernières données
  - Cartes
  - Découverte
- 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 d'événements
  - Découverte
- Alertes
  - Actions
    - Actions sur les déclencheurs
    - Actions sur les services
    - Actions de découverte
    - Actions d'auto-enregistrement
    - Actions internes
  - Types de média
  - Scripts
- Utilisateurs
  - Groupes d'utilisateurs
  - Rôles utilisateur
  - Utilisateurs
  - Jetons API
  - Authentification
- Administration
  - Général
    - Interface graphique
    - Auto-enregistrement
    - Délais d'expiration
    - Images
    - Correspondance des icônes
    - Expressions régulières

        - Options d'affichage des déclencheurs
        - Cartes géographiques
        - Modules
        - Connecteurs
        - Autre
    - Journal d'audit
    - Nettoyage interne
    - Groupes de proxys
    - 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

1 Tableaux de bord

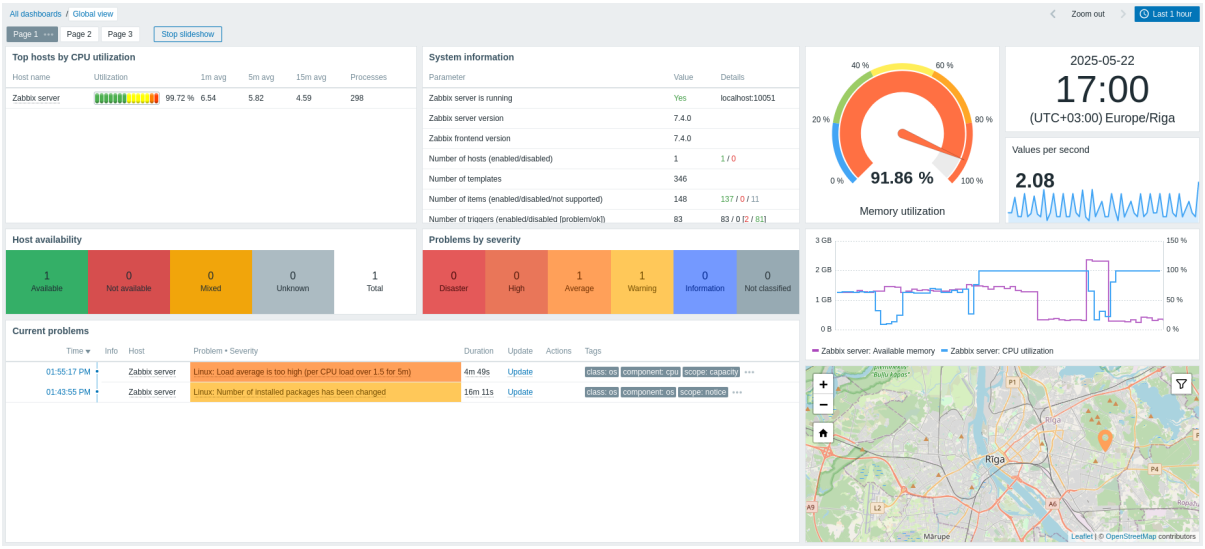
Aperçu

La section *Tableaux de bord* est conçue pour afficher des résumés de toutes les informations importantes dans un **tableau de bord**.

Bien qu'un seul tableau de bord puisse être affiché à la fois, il est possible de configurer plusieurs tableaux de bord. Chaque tableau de bord peut contenir une ou plusieurs pages pouvant être affichées en rotation sous forme de diaporama.

Une page de tableau de bord se compose de widgets, et chaque widget est conçu pour afficher des informations d'un certain type et provenant d'une certaine source, par exemple un résumé, une carte, un graphique, l'horloge, etc.

L'accès aux hôtes dans les widgets dépend des autorisations sur les hôtes.



Les pages et les widgets sont ajoutés au tableau de bord et modifiés en mode d'édition du tableau de bord. Les pages peuvent être consultées et affichées en rotation en mode d'affichage.

La période affichée dans les widgets de graphique est contrôlée par le sélecteur de période situé au-dessus des widgets. Son libellé affiche la période actuellement sélectionnée. Un clic sur ce libellé permet de développer ou de réduire le sélecteur de période.

Taille du tableau de bord

La largeur minimale d'un tableau de bord est de 1 200 pixels. Le tableau de bord ne sera pas réduit en dessous de cette largeur ; à la place, une barre de défilement horizontale s'affiche si la fenêtre du navigateur est plus petite.

La largeur maximale d'un tableau de bord correspond à la largeur de la fenêtre du navigateur. Les widgets du tableau de bord s'étirent horizontalement pour s'adapter à la fenêtre. En même temps, un widget de tableau de bord ne peut pas être étiré horizontalement au-delà des limites de la fenêtre.

Horizontalement, le tableau de bord est composé de 72 colonnes toujours de largeur égale, qui s'étirent/se réduisent dynamiquement (mais pas à moins de 1 200 pixels au total).

Verticalement, le tableau de bord peut contenir un maximum de 64 lignes ; chaque ligne a une hauteur fixe de 70 pixels.

Un widget peut donc mesurer jusqu'à 72 colonnes de large et 64 lignes de haut.

Affichage des tableaux de bord

Pour afficher tous les tableaux de bord configurés, cliquez sur *Tous les tableaux de bord* juste en dessous du titre de la section.



Les tableaux de bord sont affichés avec une étiquette de partage :

- *Mes* - indique un tableau de bord appartenant à l'utilisateur actuel
- *Partagé* - indique un tableau de bord public ou un tableau de bord privé partagé avec un utilisateur ou un groupe d'utilisateurs

Notez que les étiquettes de partage ne sont affichées que pour les tableaux de bord appartenant à l'utilisateur actuel. Les tableaux de bord appartenant à d'autres utilisateurs sont répertoriés sans étiquettes de partage.

Le filtre situé à droite au-dessus de la liste permet de filtrer les tableaux de bord par nom et par ceux créés par l'utilisateur actuel.

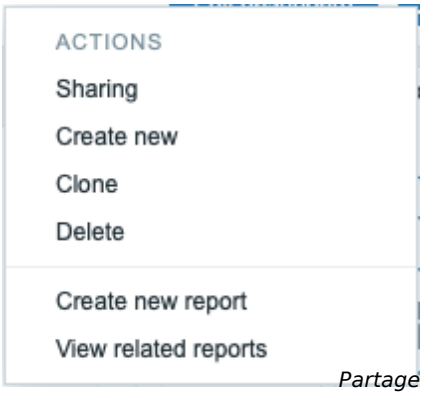
Pour supprimer un ou plusieurs tableaux de bord, cochez les cases des tableaux de bord correspondants et cliquez sur *Supprimer* sous la liste.

Affichage d'un tableau de bord

Pour afficher un seul tableau de bord, cliquez sur son nom dans la liste des tableaux de bord.

Lors de l'**affichage** d'un tableau de bord, les options suivantes sont disponibles :

Edit dashboard	Passer en mode **édition** du tableau de bord. Le mode édition s'ouvre également lors de la création d'un nouveau tableau de bord et lorsque vous cliquez sur le bouton de modification d'un widget.
	Ouvrir le menu d'actions (voir les descriptions des actions ci-dessous).

	*Partage* - modifier les préférences de partage du tableau de bord. *Créer* - créer un nouveau tableau de bord. *Cloner* - créer un nouveau tableau de bord en copiant les propriétés du tableau de bord existant. Il vous sera d'abord demandé de saisir les paramètres du tableau de bord. Ensuite, le nouveau tableau de bord s'ouvrira en mode édition avec tous les widgets du tableau de bord d'origine. *Supprimer* - supprimer le tableau de bord. *Créer un nouveau rapport* - ouvrir une fenêtre contextuelle contenant le formulaire de configuration du rapport. Désactivé si l'utilisateur n'a pas l'autorisation de gérer les rapports planifiés. *Afficher les rapports associés* - ouvrir une fenêtre contextuelle contenant une liste des rapports existants basés sur le tableau de bord actuel. Désactivé s'il n'existe aucun rapport associé ou si l'utilisateur n'a pas l'autorisation d'afficher les rapports planifiés.
	Afficher uniquement le contenu de la page (mode kiosque). Le mode kiosque est également accessible avec les paramètres d'URL suivants : `/zabbix.php?action=dashboard.view&kiosk=1`. Pour revenir au mode normal : `/zabbix.php?action=dashboard.view&kiosk=0`.

Partage

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 ne sont visibles que par leur propriétaire ainsi que par les utilisateurs/groupes d'utilisateurs ajoutés dans les préférences de partage.

Pour modifier le statut de partage d'un tableau de bord, cliquez sur l'option *Partage* dans le menu d'action lors de l'affichage d'un tableau de bord unique :

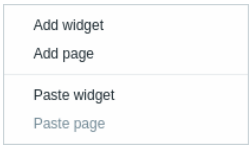
Parameter	Description
Type	Sélectionnez le type de tableau de bord : **Privé** - le tableau de bord est visible uniquement par les groupes d'utilisateurs et les utilisateurs sélectionnés. **Public** - le tableau de bord est visible par tous.
List of user group shares	Sélectionnez les groupes d'utilisateurs auxquels le tableau de bord est accessible. Vous pouvez autoriser un accès en lecture seule ou en lecture-écriture.
List of user shares	Sélectionnez les utilisateurs auxquels le tableau de bord est accessible. Vous pouvez autoriser un accès en lecture seule ou en lecture-écriture.

Consultez Permissions to dashboards pour savoir comment les options de partage limitent les actions disponibles pour les utilisateurs.

Modification d'un tableau de bord

Lors de la **modification** d'un tableau de bord, les options suivantes sont disponibles :

	Modifier les paramètres généraux du tableau de bord.

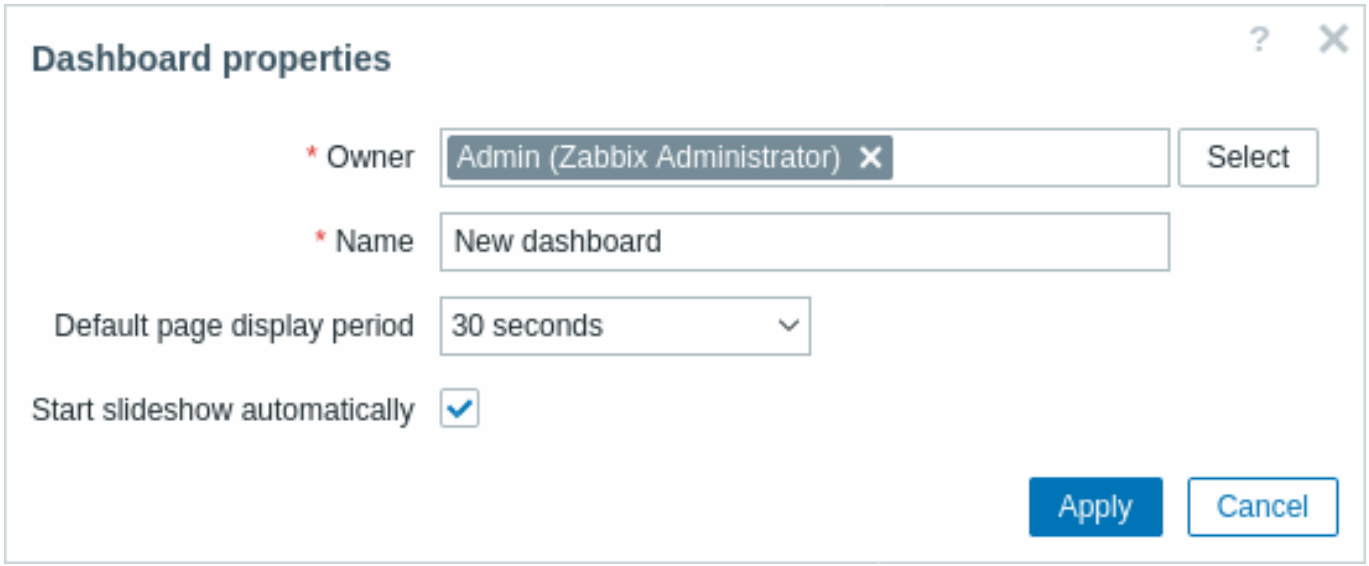
	Ajouter un nouveau widget. Cliquer sur le bouton fléché ouvre le menu d'action (voir les descriptions des actions ci-dessous).
	*Ajouter un widget* - ajouter un nouveau widget. *Ajouter une page* - ajouter une nouvelle page. *Coller un widget* - coller un widget copié. Cette option est grisée si aucun widget n'a été copié. Une seule entité (widget ou page) peut être copiée à la fois. *Coller une page* - coller une page copiée. Cette option est grisée si aucune page n'a été copiée.
	Enregistrer les modifications du tableau de bord.
	Annuler les modifications du tableau de bord.

Création d'un tableau de bord

Il est possible de créer un nouveau tableau de bord de deux façons :

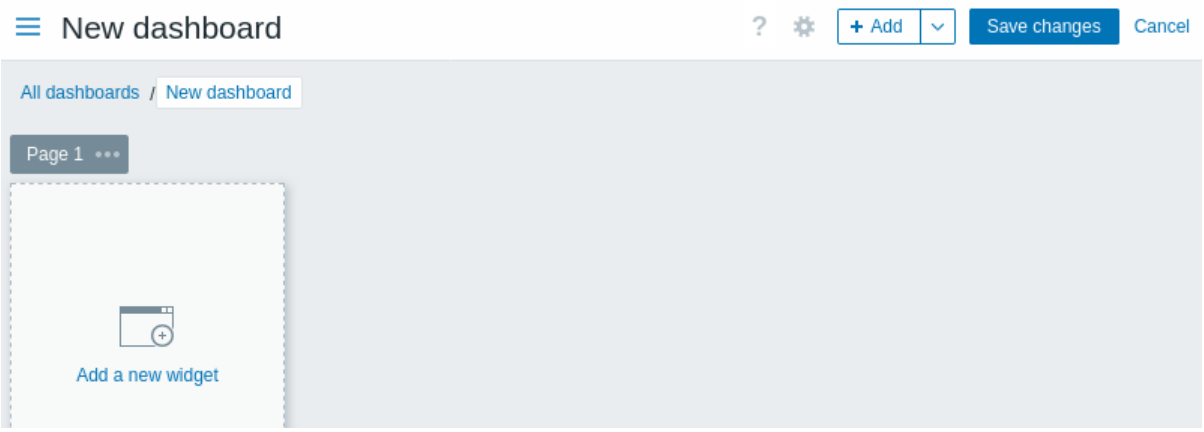
- Cliquez sur *Créer un tableau de bord* lors de l'affichage de tous les tableaux de bord
- Sélectionnez *Créer nouveau* dans le menu d'action lors de l'affichage d'un seul tableau de bord

Il vous sera d'abord demandé de saisir les paramètres généraux du tableau de bord :



Paramètre	Description
Propriétaire	Sélectionnez l'utilisateur système qui sera le propriétaire du tableau de bord.
Nom	Saisissez le nom du tableau de bord.
Période d'affichage par défaut de la page	Sélectionnez la durée pendant laquelle une page du tableau de bord est affichée avant de passer à la page suivante dans un diaporama.
Démarrer automatiquement le diaporama	Cochez cette case pour lancer automatiquement un diaporama lorsqu'il existe plus d'une page de tableau de bord.

Lorsque vous cliquez sur *Appliquer*, un tableau de bord vide s'ouvre :



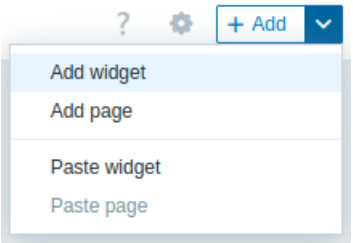
Pour remplir le tableau de bord, vous pouvez ajouter des widgets et des pages.

Cliquez sur le bouton *Enregistrer les modifications* pour enregistrer le tableau de bord. Si vous cliquez sur *Annuler*, le tableau de bord ne sera pas créé.

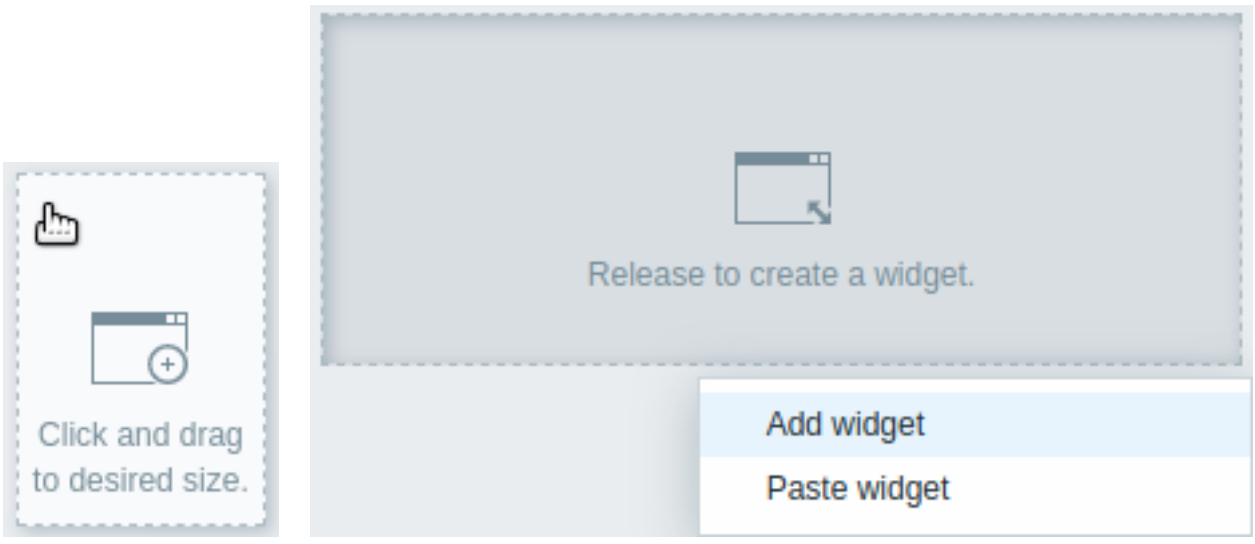
Ajout de widgets

Vous pouvez ajouter divers widgets (Journal des actions, Horloge, Statut de découverte, etc.) à un tableau de bord. Les widgets peuvent être ajoutés de deux façons :

- Cliquez sur le bouton *Add* ou sur la flèche à côté, puis sélectionnez *Add widget* dans le menu d'actions. Après avoir configuré le widget, il sera ajouté avec sa taille par défaut et placé après tout widget existant.

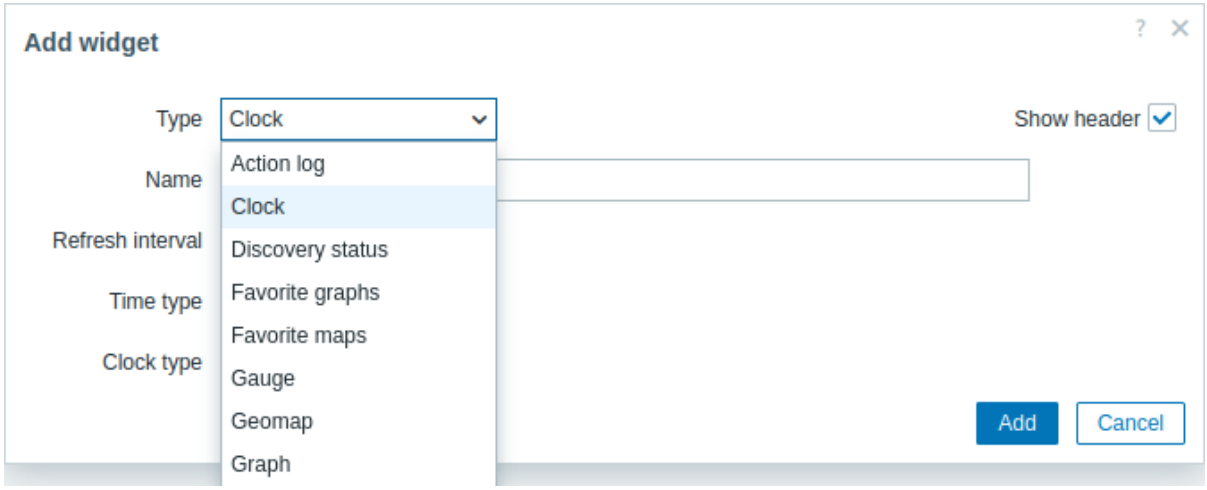


- Déplacez la souris vers un emplacement vide du tableau de bord. Remarquez qu'un espace réservé apparaît, puis cliquez dessus pour ouvrir le formulaire de configuration du widget. Après avoir configuré le widget, il sera ajouté avec sa taille par défaut ou redimensionné pour s'adapter à l'espace disponible. Vous pouvez également faire glisser l'espace réservé pour définir la taille du widget, puis le relâcher pour ouvrir le formulaire de configuration du widget (si un widget est copié dans le presse-papiers, il vous sera d'abord demandé de choisir entre *Add widget* ou *Paste widget*).



Dans le formulaire de configuration du widget :

1. Sélectionnez le type de widget.
2. Configurez les paramètres du widget.
3. Cliquez sur *Add* pour ajouter le widget au tableau de bord.



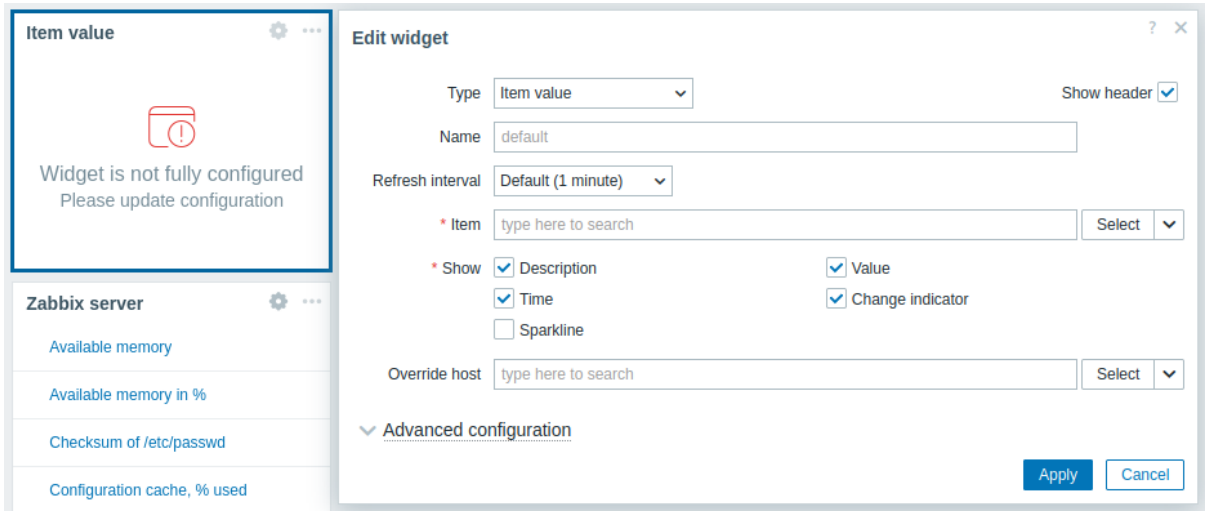
Modification des widgets

Les widgets peuvent être redimensionnés et repositionnés dans le mode d'édition du tableau de bord en faisant glisser leur barre de titre. Vous pouvez également repositionner le formulaire de configuration du widget en faisant glisser sa barre de titre.

Chaque widget comprend des contrôles dans son coin supérieur droit :

- - modifier le widget
- - ouvrir le menu du widget

Pendant la modification, le widget est mis en surbrillance et les changements sont prévisualisés en temps réel. Si les paramètres obligatoires du widget ne sont pas configurés, le widget passe à l'état *Widget is not fully configured*.



Après la modification, cliquez sur *Apply* dans le widget, puis sur *Save changes* dans le coin supérieur droit du tableau de bord pour appliquer vos modifications.

Copier/coller des widgets

Vous pouvez copier et coller des widgets de tableau de bord afin de créer rapidement de nouveaux widgets avec la même configuration. Les widgets peuvent être copiés-collés au sein du même tableau de bord ou entre des tableaux de bord ouverts dans des onglets différents.

Pour copier un widget, utilisez le menu du widget. Pour coller un widget copié (disponible en mode d'édition du tableau de bord) :

- Cliquez sur la flèche à côté 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 également coller un widget copié par-dessus un widget existant à l'aide de l'option *Coller* dans son menu du widget.

Créer un diaporama

Un diaporama s'exécute automatiquement si le tableau de bord contient au moins deux pages (voir Ajout de pages) et si l'une des conditions suivantes est remplie :

- L'option *Démarrer automatiquement le diaporama* est cochée dans les propriétés du tableau de bord
- L'URL du tableau de bord contient un paramètre `slideshow=1`

Les pages défilent selon les intervalles définis dans les propriétés du tableau de bord et des pages individuelles. Cliquez sur :

- *Arrêter le diaporama* - pour arrêter le diaporama
- *Démarrer le diaporama* - pour démarrer le diaporama



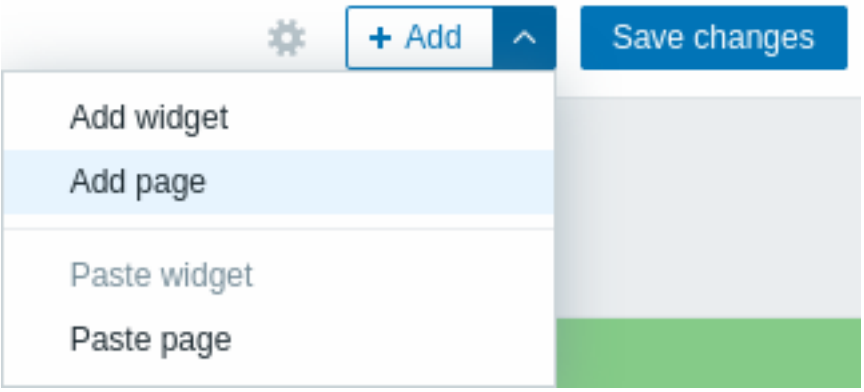
Les commandes liées au diaporama sont également disponibles en mode kiosque (où seul le contenu de la page est affiché) :

-  - arrêter le diaporama
-  - démarrer le diaporama
-  - revenir d'une page
-  - passer à la page suivante

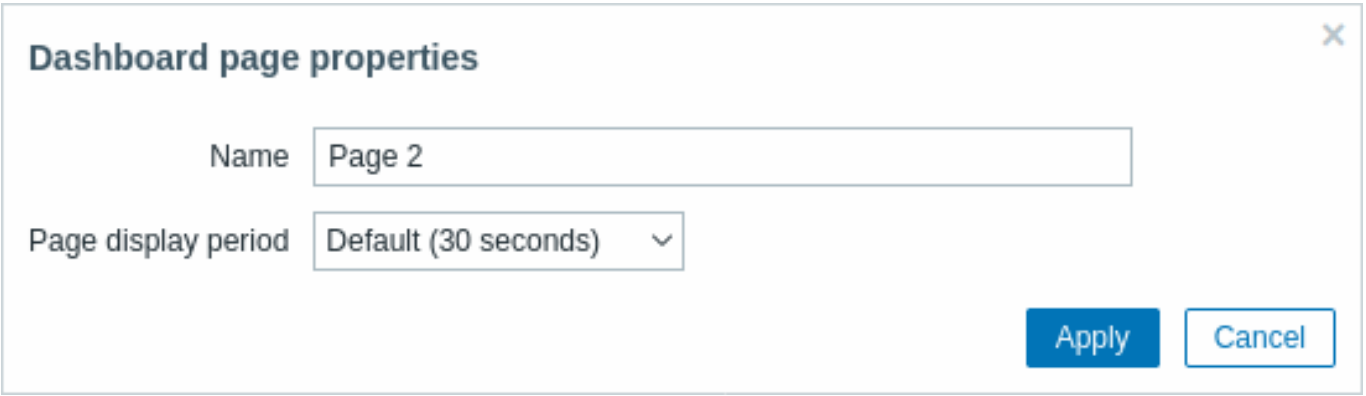
Ajout de pages

Pour ajouter une nouvelle page à un tableau de bord :

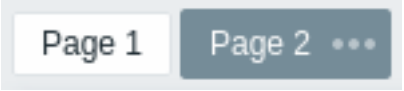
- Assurez-vous que le tableau de bord est en mode édition
- Cliquez sur la flèche à côté du bouton *Ajouter* et sélectionnez l'option *Ajouter une page*



- Renseignez les paramètres généraux de la page et cliquez sur *Appliquer*. Si vous laissez le nom vide, la page sera ajoutée avec un nom au format `Page N`, où « N » est le numéro incrémental de la page. La période d'affichage de la page permet de personnaliser la durée pendant laquelle une page est affichée dans un diaporama.



Une nouvelle page sera ajoutée, indiquée par un nouvel onglet (*Page 2*).



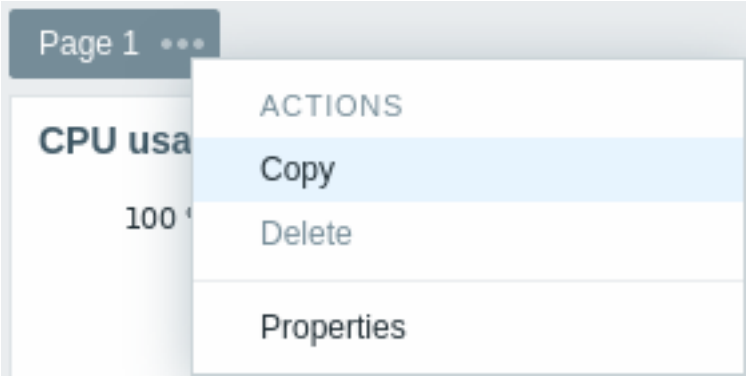
Les pages peuvent être réorganisées par glisser-déposer des onglets de page. La réorganisation conserve le nom d'origine des pages. Il est toujours possible d'accéder à chaque page en cliquant sur son onglet.

Lorsqu'une nouvelle page est ajoutée, elle est vide. Vous pouvez y ajouter des widgets comme décrit ci-dessus.

Copier/coller des pages

Les pages de tableau de bord peuvent être copiées et collées, ce qui permet de créer une nouvelle page avec les propriétés d'une page existante. Elles peuvent être collées à partir du même tableau de bord ou d'un autre tableau de bord.

Pour coller une page existante dans le tableau de bord, copiez-la d'abord à l'aide du menu de la page :

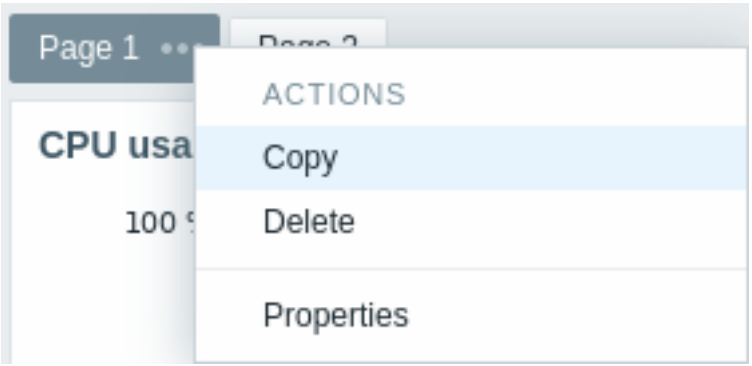


Pour coller la page copiée :

- Assurez-vous que le tableau de bord est en mode édition
- Cliquez sur la flèche à côté du bouton *Ajouter* et sélectionnez l'option *Coller la page*

Menu de la page

Le menu de la page peut être ouvert en cliquant sur les trois points à côté du nom de la page :

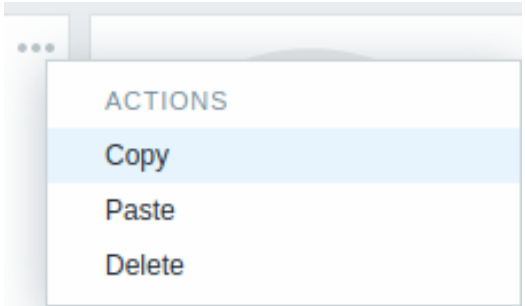


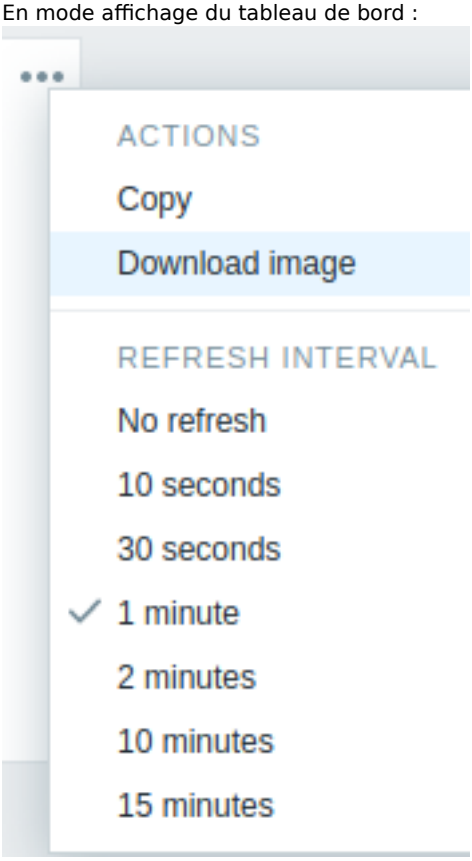
Il contient les options suivantes :

- *Copier* - copier la page
- *Supprimer* - supprimer la page (les pages ne peuvent être supprimées qu'en mode d'édition du tableau de bord)
- *Propriétés* - personnaliser les paramètres de la page (le nom et la période d'affichage de la page dans un diaporama)

Menu du widget

Le menu du widget contient différentes options selon que le tableau de bord est en mode édition ou en mode affichage :

Menu du widget	Options
En mode édition du tableau de bord : 	*Copier* - copier le widget. *Coller* - coller un widget copié sur ce widget. Cette option est grisée si aucun widget n'a été copié. *Supprimer* - supprimer le widget.

Menu du widget	Options
En mode affichage du tableau de bord :  	*Copier* - copier le widget. *Télécharger l'image* - télécharger le widget sous forme d'image PNG. Cette option est disponible pour les widgets prenant en charge cette fonctionnalité (voir les pages de chaque widget). Selon les paramètres de votre navigateur, l'image est soit automatiquement enregistrée dans le dossier Téléchargements, soit une boîte de dialogue Enregistrer s'ouvre, vous permettant de choisir l'emplacement. *Intervalle de rafraîchissement* - sélectionner la fréquence de rafraîchissement du contenu du widget

Autorisations relatives aux tableaux de bord

Les autorisations relatives aux tableaux de bord pour les utilisateurs standard et les utilisateurs de type « Admin » (à condition que l'accès aux tableaux de bord et l'option *Créer et modifier des tableaux de bord* soient activés pour le rôle utilisateur qui leur est attribué) sont limitées de la manière décrite ci-dessous.

- Ils peuvent voir et cloner un tableau de bord public même si aucun utilisateur ou groupe d'utilisateurs n'est ajouté dans les préférences de partage du tableau de bord.
- Ils peuvent voir et cloner un tableau de bord privé s'ils disposent d'au moins des droits *Lecture* sur celui-ci, définis au moyen des préférences de partage.
- Ils peuvent modifier et supprimer un tableau de bord uniquement s'ils disposent des droits *Lecture-écriture* sur celui-ci, définis au moyen des préférences de partage.
- Ils ne peuvent pas changer le propriétaire du tableau de bord.

1 Widgets de 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.

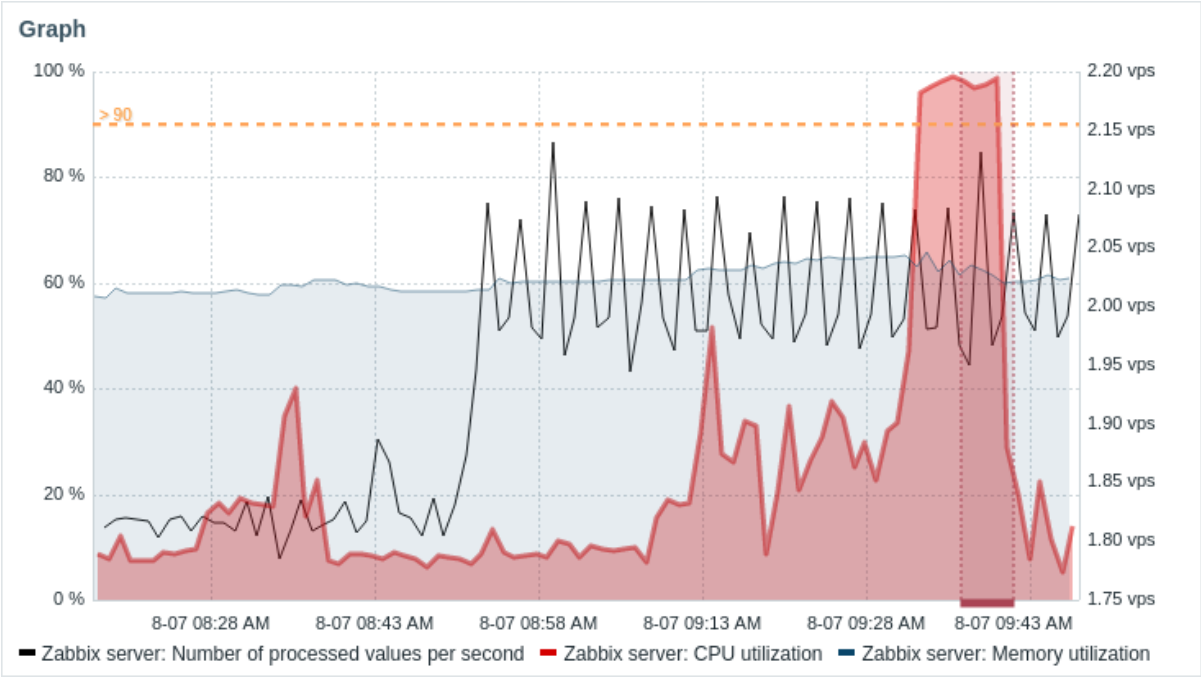
Pour plus de détails sur la configuration des widgets, consultez Paramètres des widgets de tableau de bord.

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.

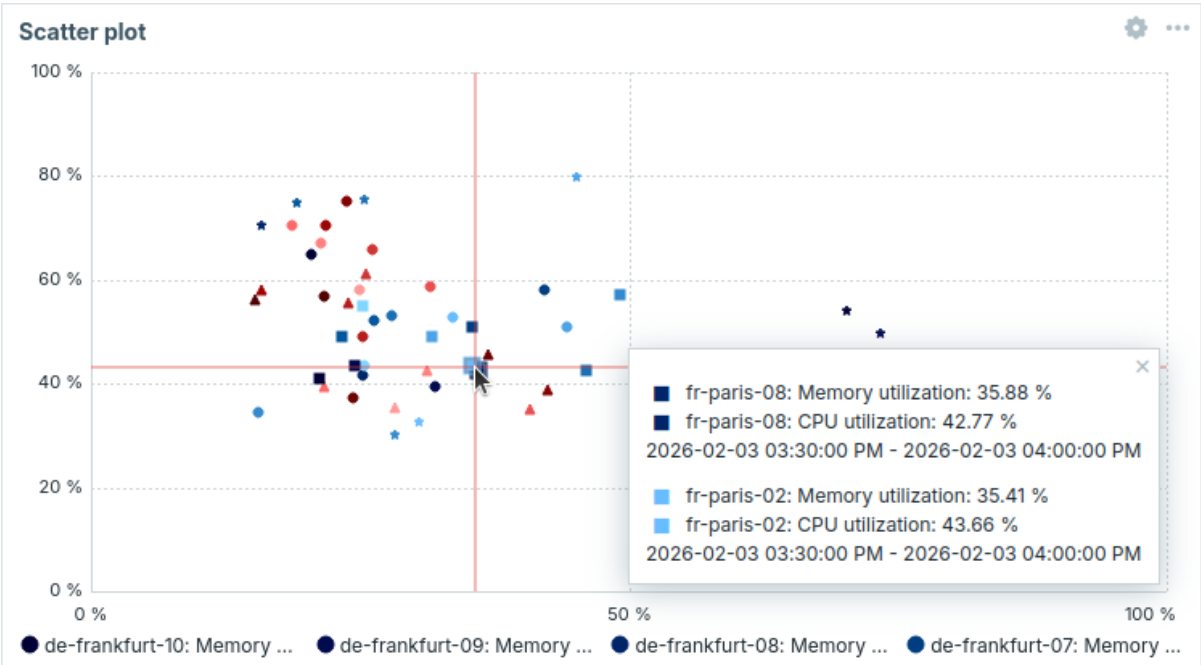


Voir aussi :

- 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.

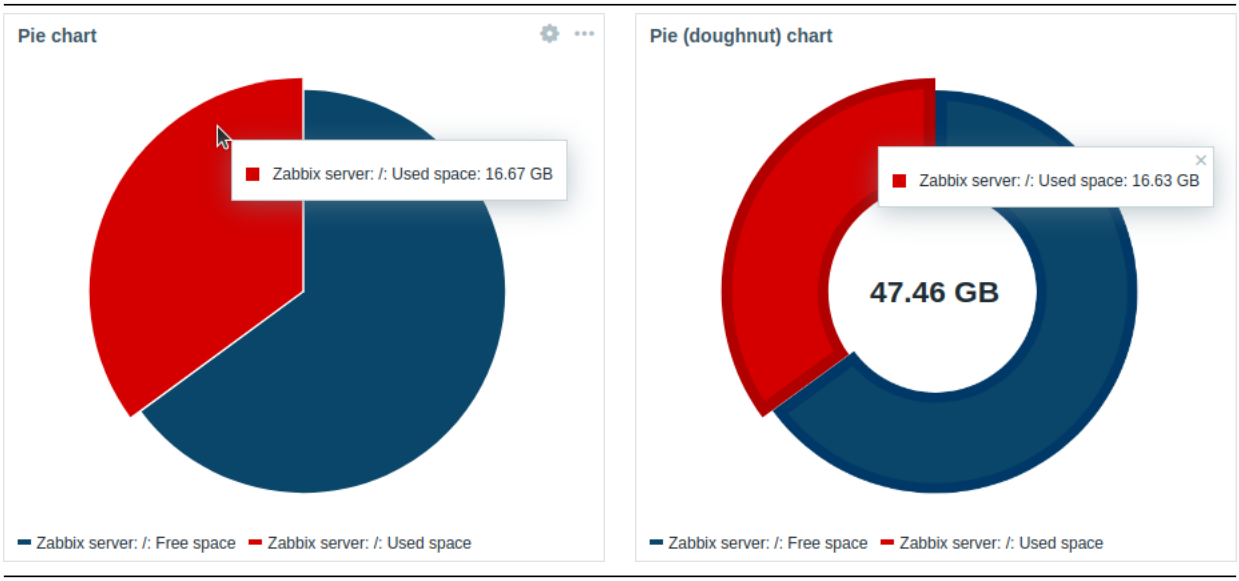
Nuage de points

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

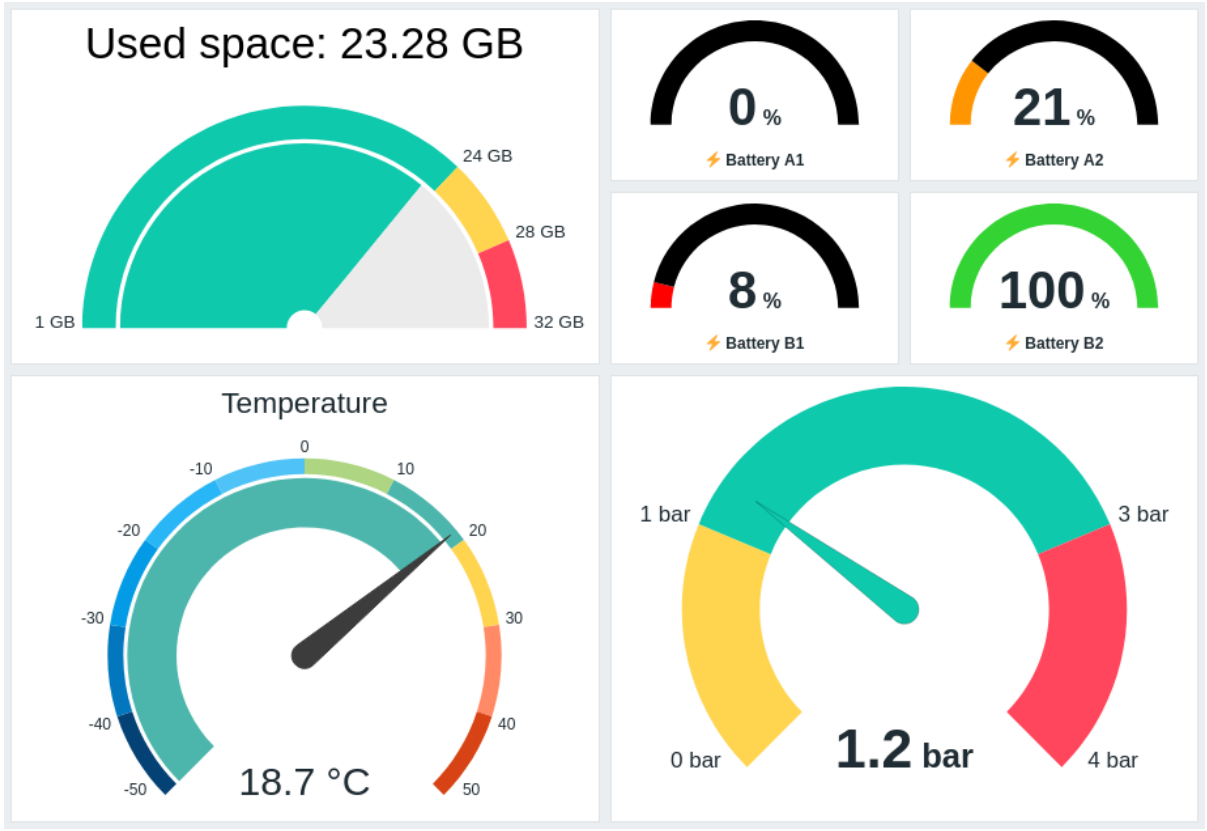
Le widget Camembert affiche des données numériques d'éléments sous la forme d'un graphique vectoriel en secteurs ou en anneau. Ce widget peut vous aider à visualiser comment les éléments ou les hôtes contribuent à l'ensemble du jeu de données.



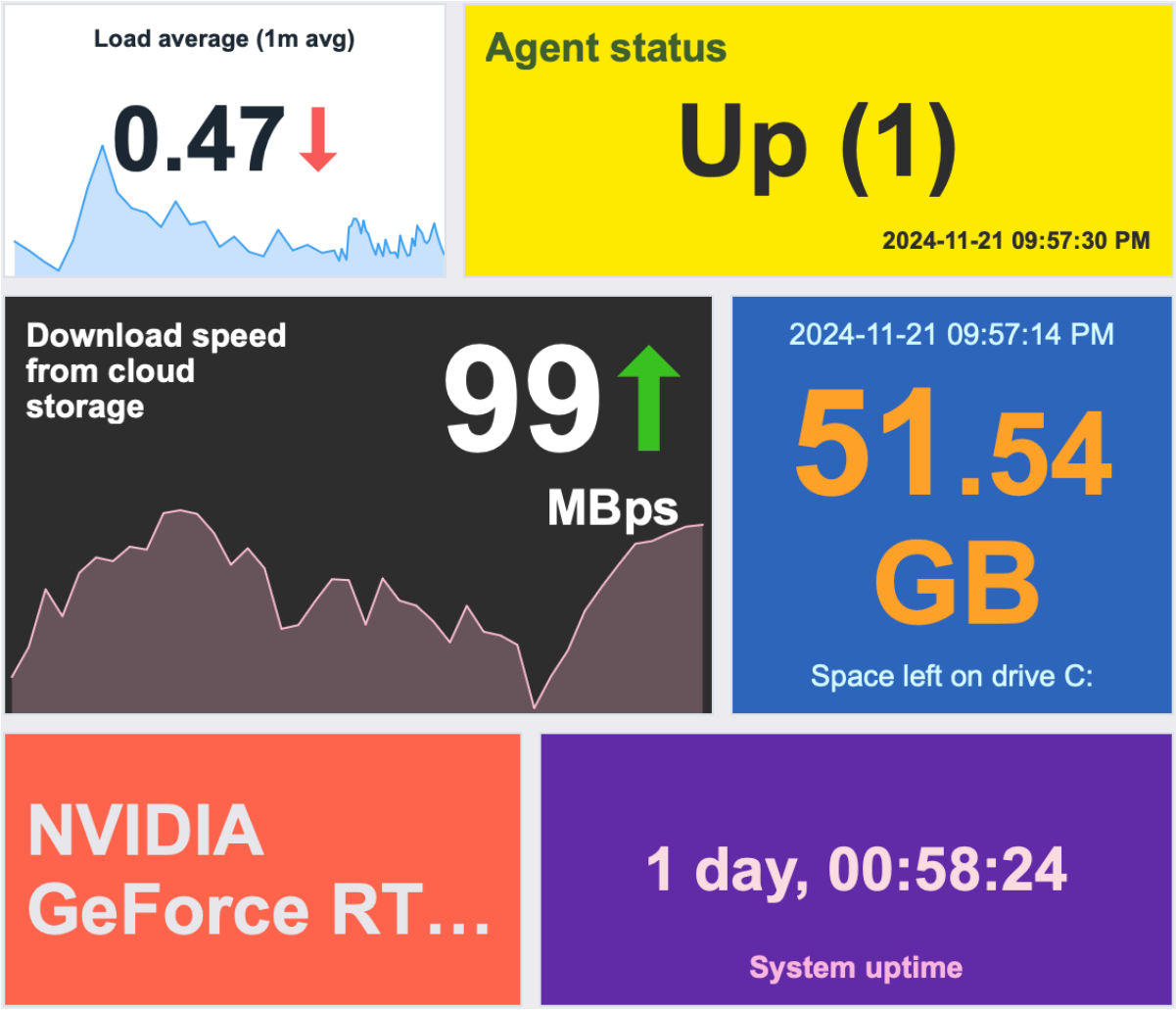
Jauge/Valeur d'élément

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 :

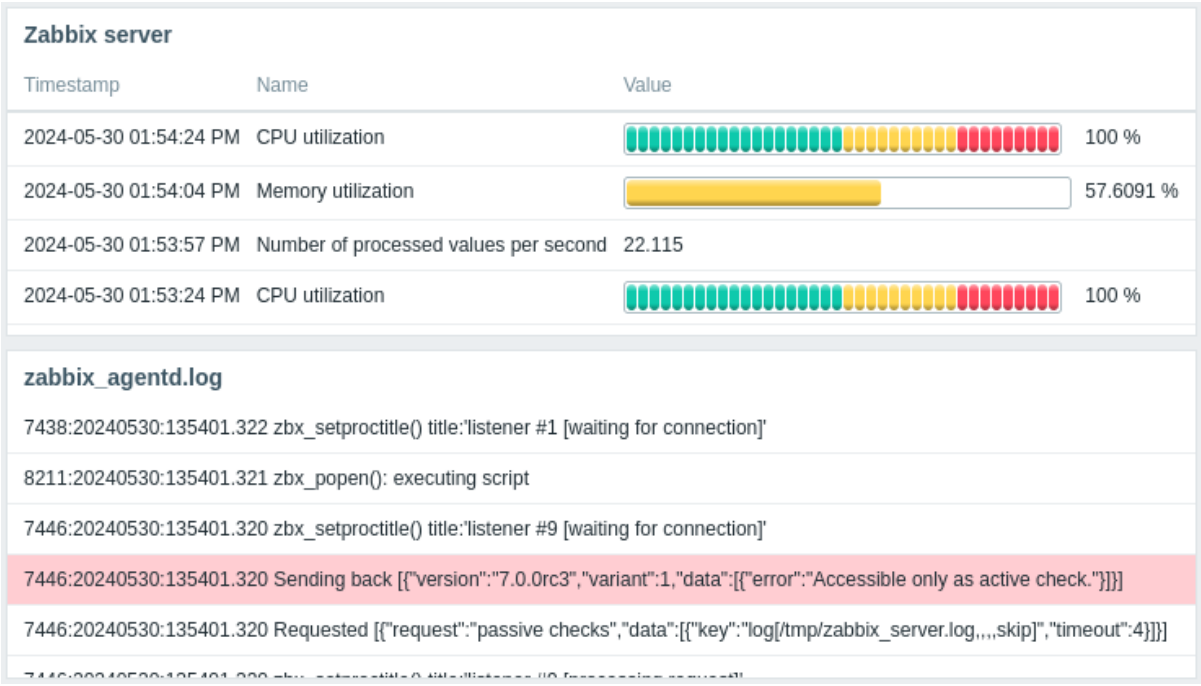


Widget Valeur d'élément :



Historique des éléments

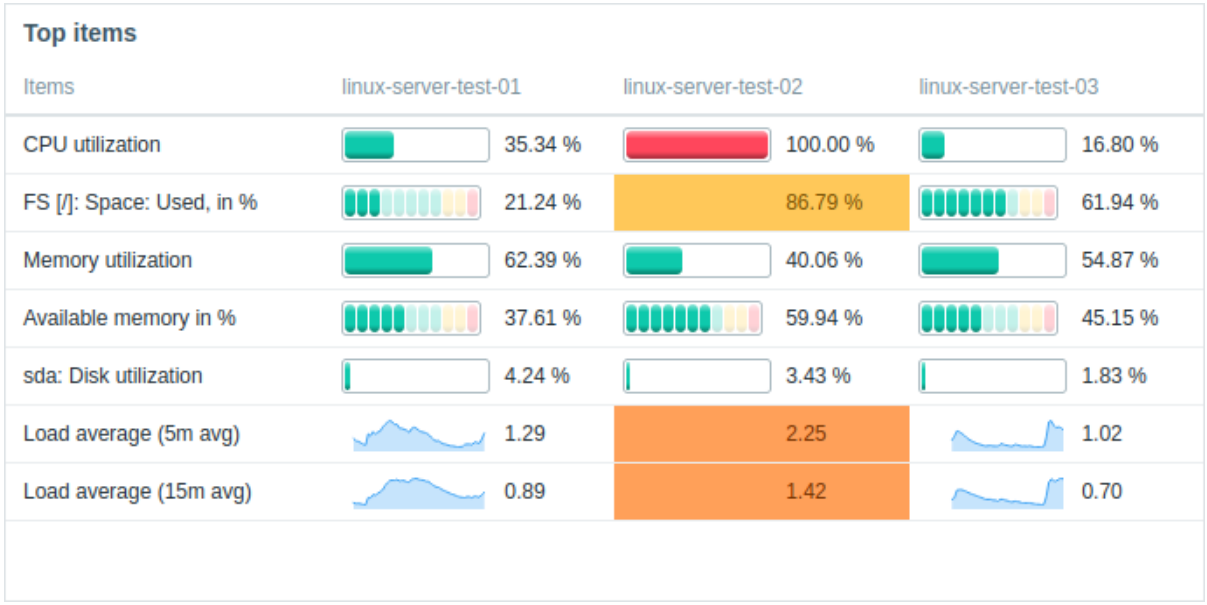
Le widget Item history affiche les données les plus récentes pour tous les types d'éléments (numériques, texte, etc.) sous forme de tableau. Il peut afficher des barres de progression et des images (utile pour les éléments de navigateur) et peut mettre en évidence des valeurs (utile pour la surveillance 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.

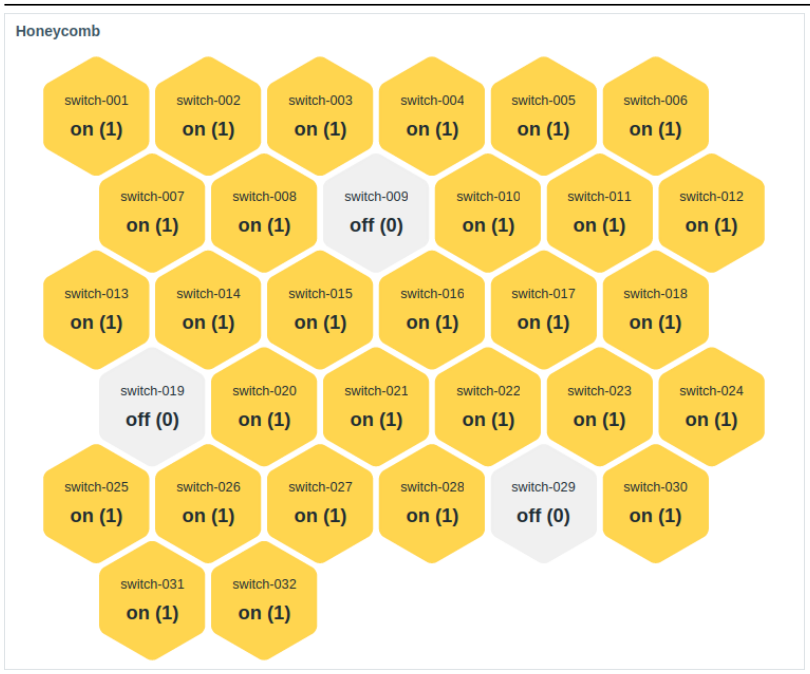


Voir aussi :

- 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.



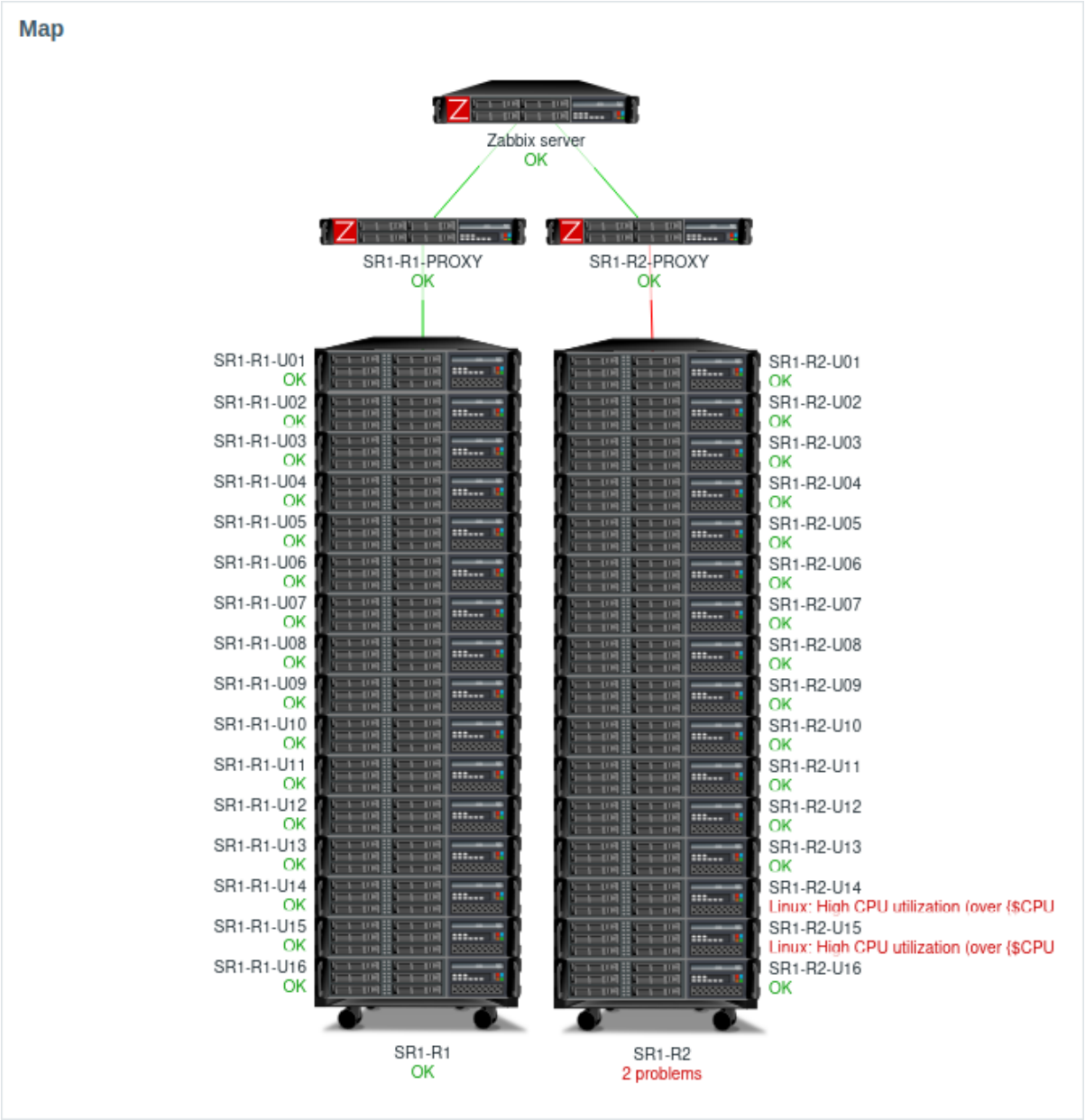


Cartes

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.

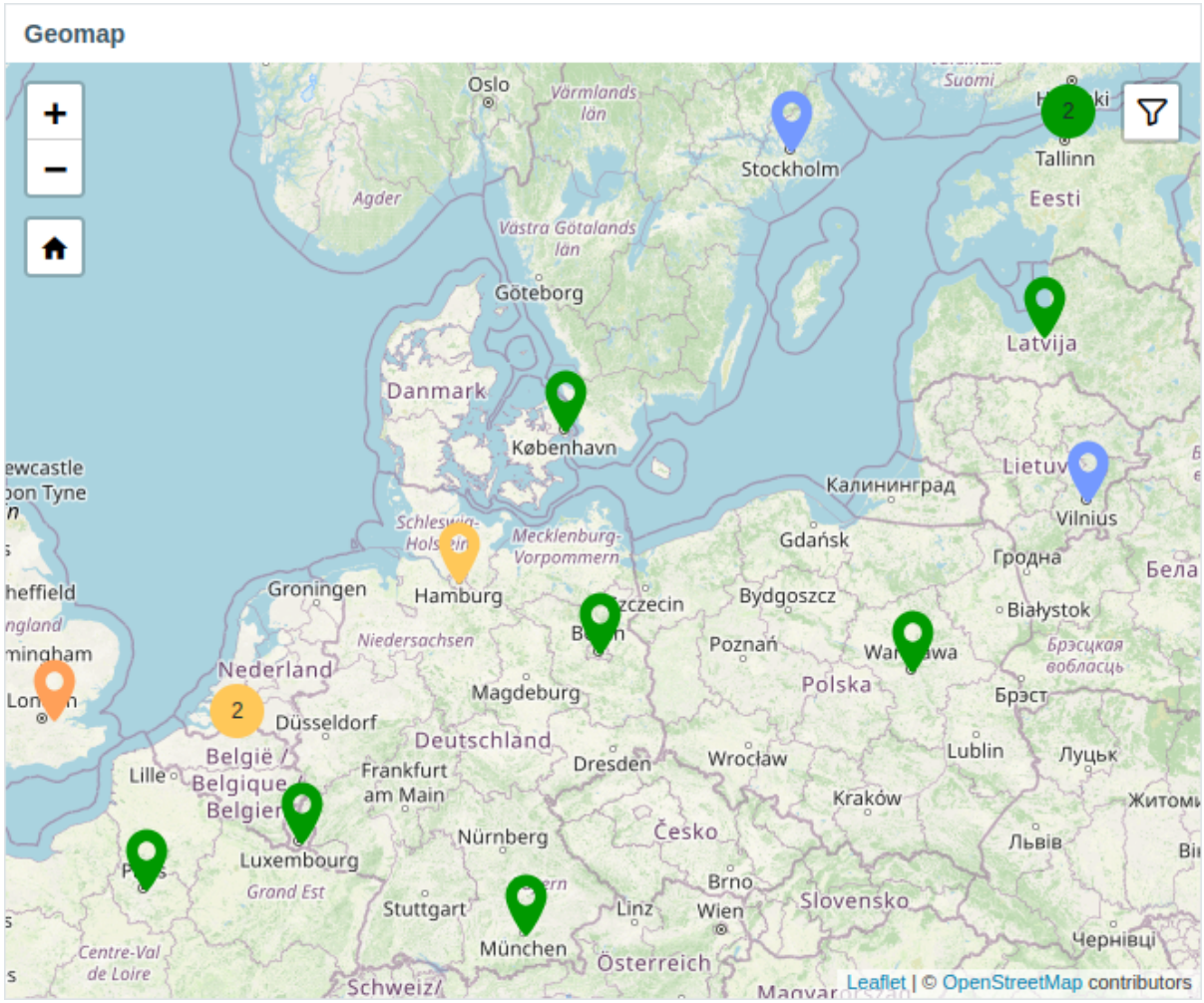


Voir aussi :

- 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.

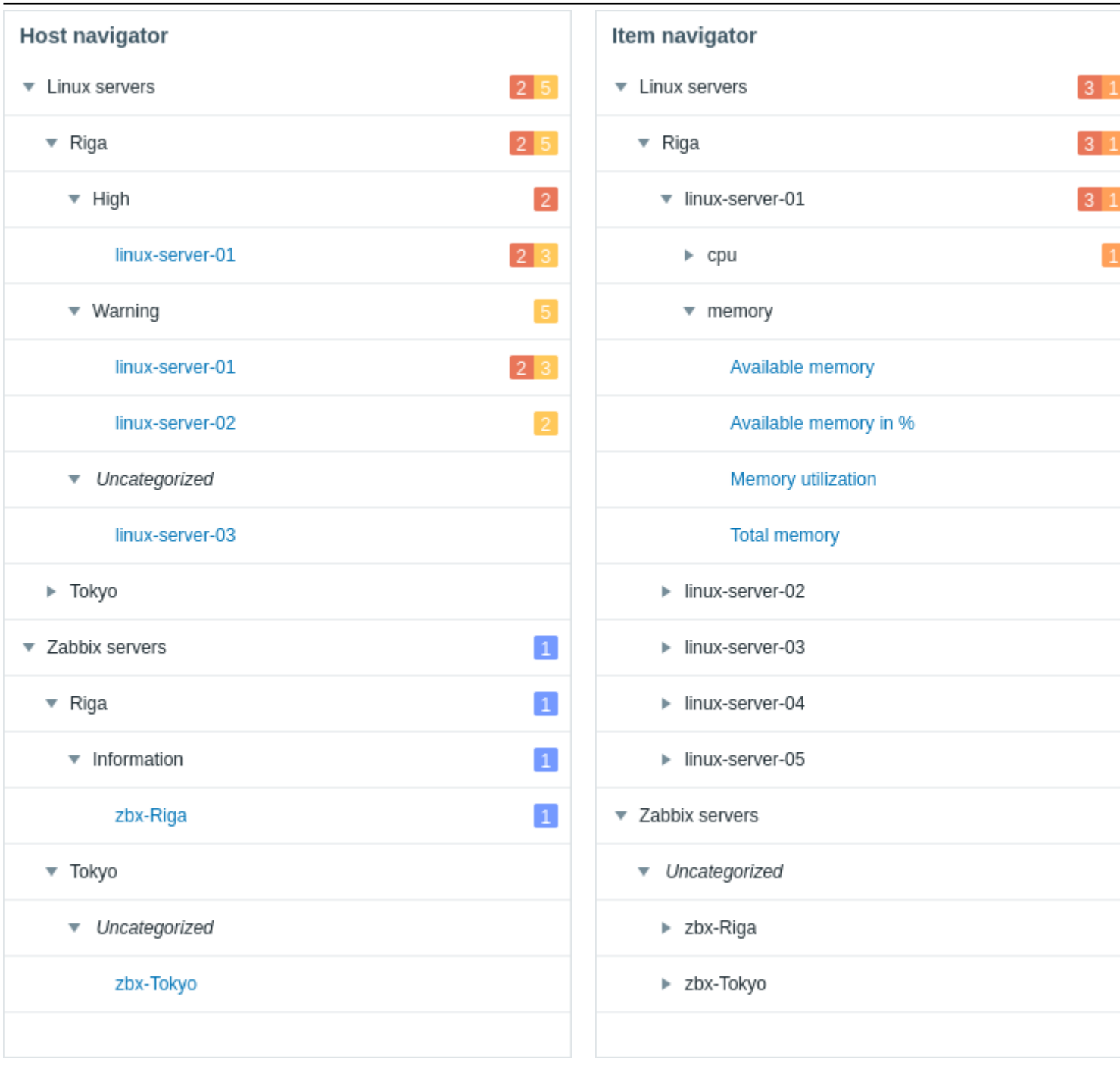


Détails de l’hôte/de l’élément

Ces widgets fournissent des listes et des vues détaillées des hôtes ou des éléments.

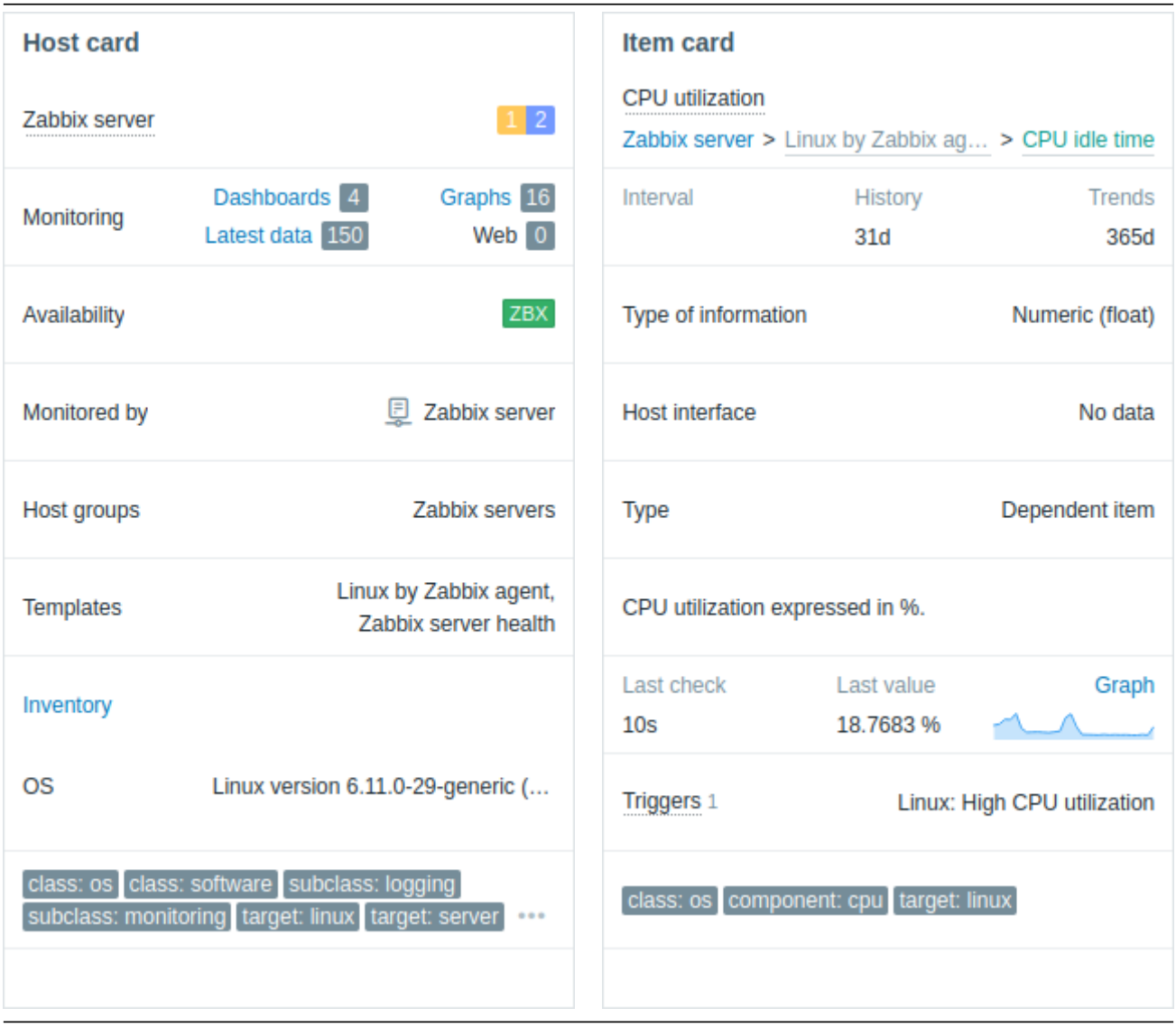
Navigateur d’hôte/d’élément

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

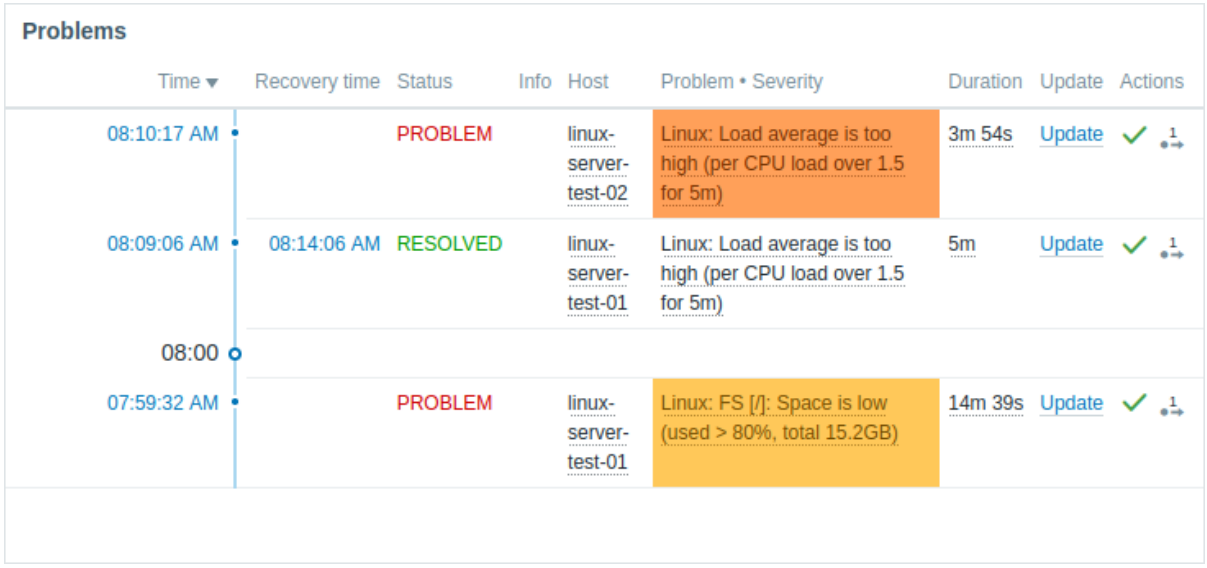
Le widget Disponibilité de l'hôte affiche la disponibilité des hôtes dans les groupes d'hôtes sélectionnés, vous permettant de surveiller le nombre d'hôtes qui sont opérationnels ou indisponibles.

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.

Web monitoring

Host group ▲	Ok	Failed	Unknown
Applications	1	1	1
Applications/External	1		
Applications/Internal			1
Applications/Test		1	

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.

Discovery status

Discovery rule	Up	Down
DC New York	120	5
DC Riga	95	8
Local network	15	3

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

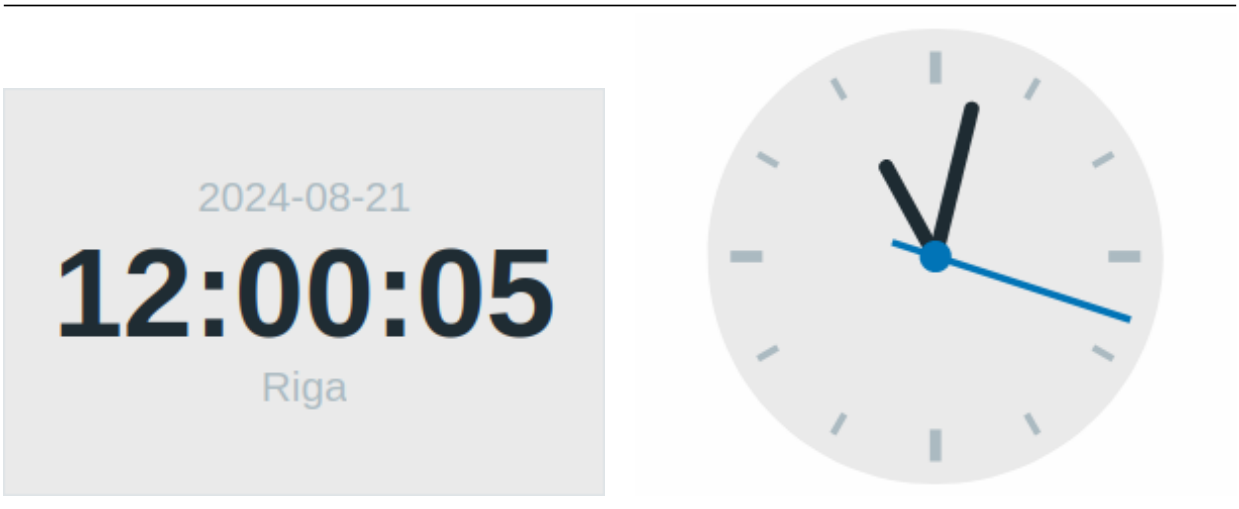
Le widget Informations système affiche un résumé des principales données du serveur Zabbix et du système, ou des détails sur les nœuds de haute disponibilité.

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	

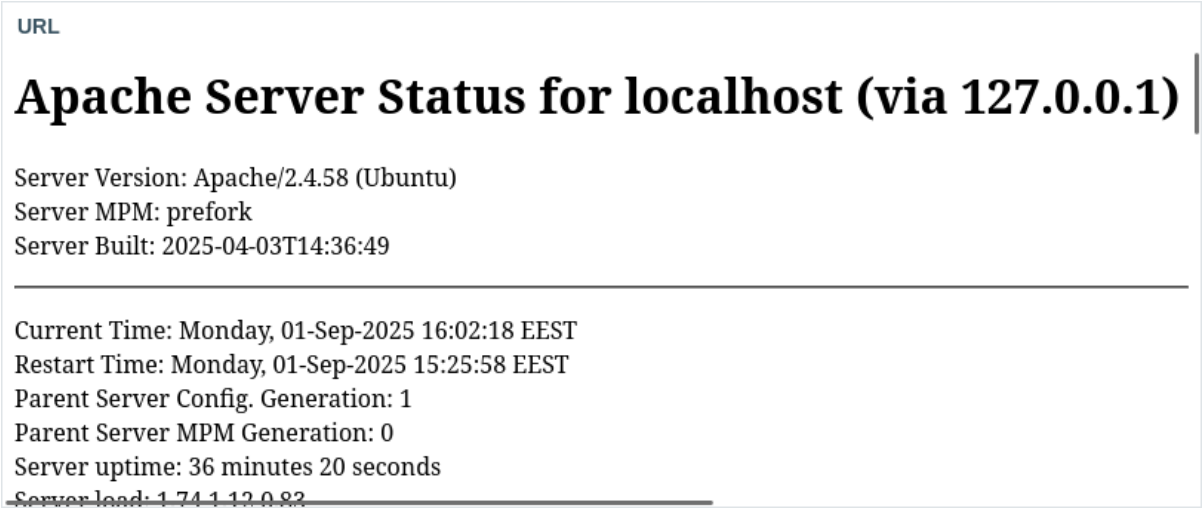
Horloge

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.



2 Paramètres du widget de 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 *Pas d'actualisation* à *15 minutes* selon le type de widget. Par exemple : - *Pas d'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 et 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 à trois points d'un widget et sélectionnez l'intervalle d'actualisation souhaité dans la liste déroulante. Notez que l'intervalle d'actualisation propre à un utilisateur a priorité 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. Lorsqu'elle n'est pas cochée, l'en-tête est masqué pour économiser de l'espace et devient visible uniquement au survol du widget par la souris (en modes affichage et édition). L'en-tête est également semi-visible lors du déplacement d'un widget 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 :

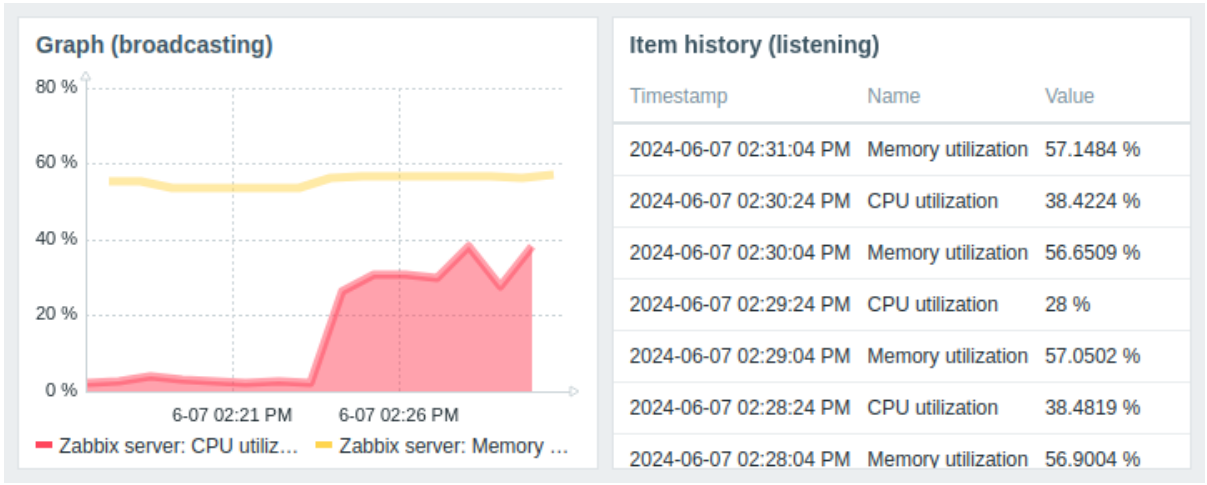
- Le widget *Journal des actions* peut uniquement récupérer les données de période de temps des widgets *Graphique*, *Graphique (classique)* et *Prototype de graphique*.
- Le widget *Geomap* peut diffuser des données d'hôte aux widgets qui les reçoivent (*Honeycomb*, *Top items*, etc.) et peut également 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.

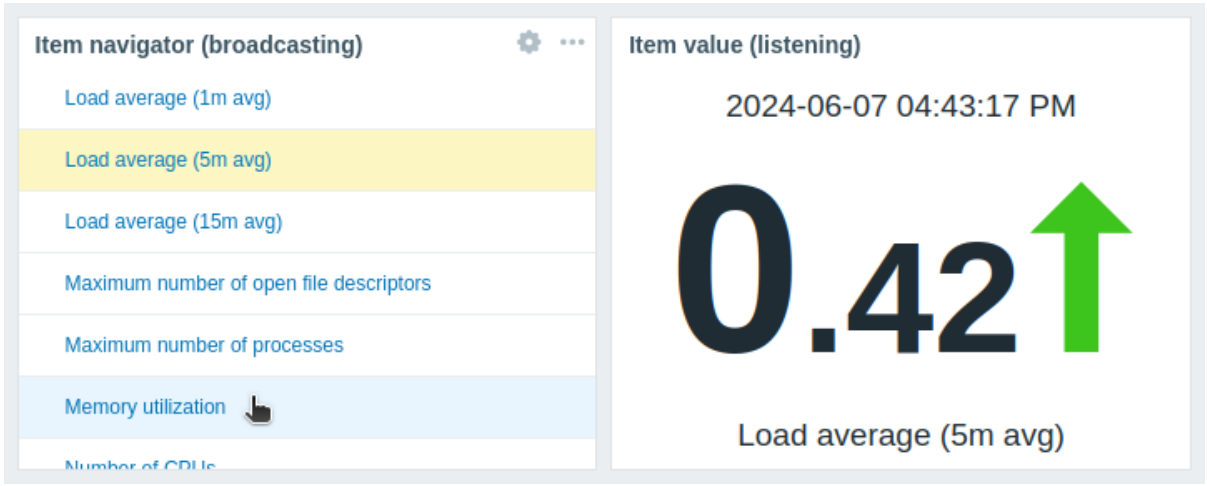
Widget	Diffuse	Reçoit
Journal des actions	-	Période de temps
Horloge	-	-
Statut de découverte	-	-
Graphiques favoris	-	-
Cartes favorites	-	-
Jauge	-	Hôtes, Éléments
Geomap	Hôtes	Groupes d'hôtes, Hôtes
Graphique	Période de temps, Éléments	Hôtes, Éléments, Période de temps
Graphique (classique)	Période 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.

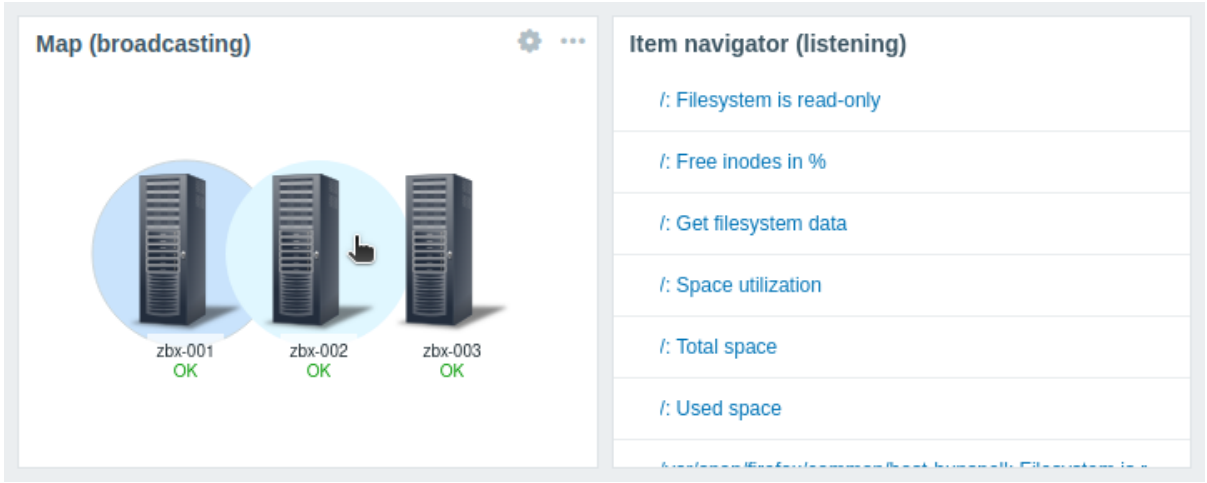
Les widgets qui prennent en charge la diffusion commencent à le faire automatiquement dès leur création. Par exemple, le widget *Graph* diffuse immédiatement les données de période de temps aux widgets à l'écoute.



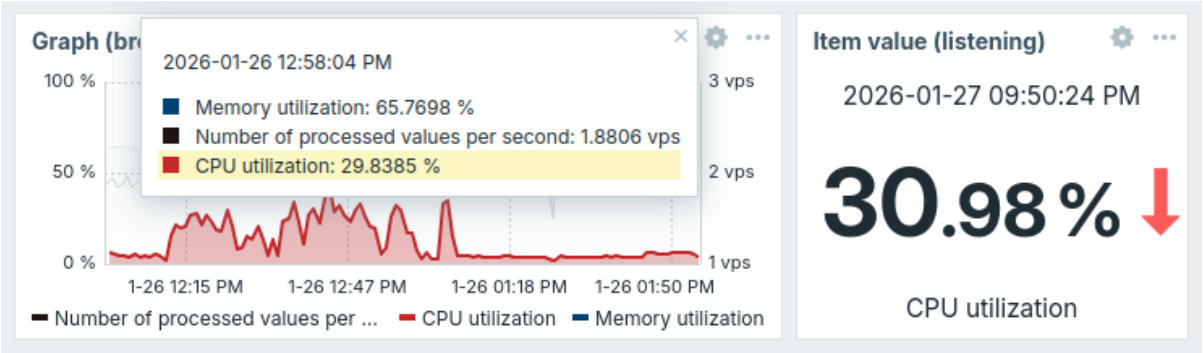
Les widgets qui prennent en charge la sélection d'entités diffusent automatiquement les données de la première entité disponible. Par exemple, le widget *Item navigator* diffuse les données du premier élément de sa liste d'éléments. Lorsqu'un autre élément est sélectionné, il diffuse les données de cet 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.

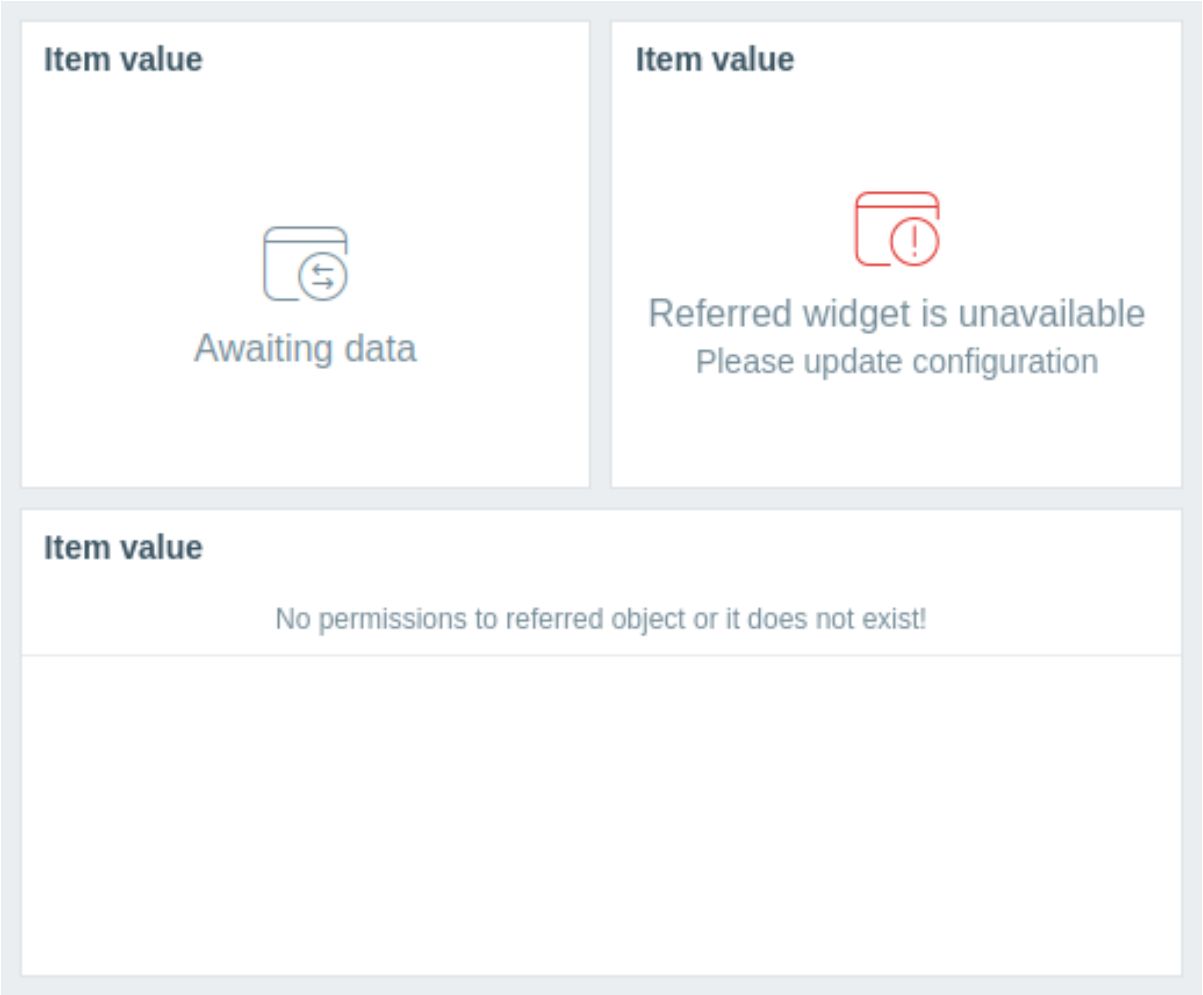


Les widgets tels que *Graph* et *Scatter plot* peuvent diffuser des données d'élément lorsqu'un élément est sélectionné dans leurs infobulles :



Les widgets ont également des comportements spécifiques lorsqu'ils **reçoivent** des données d'autres widgets :

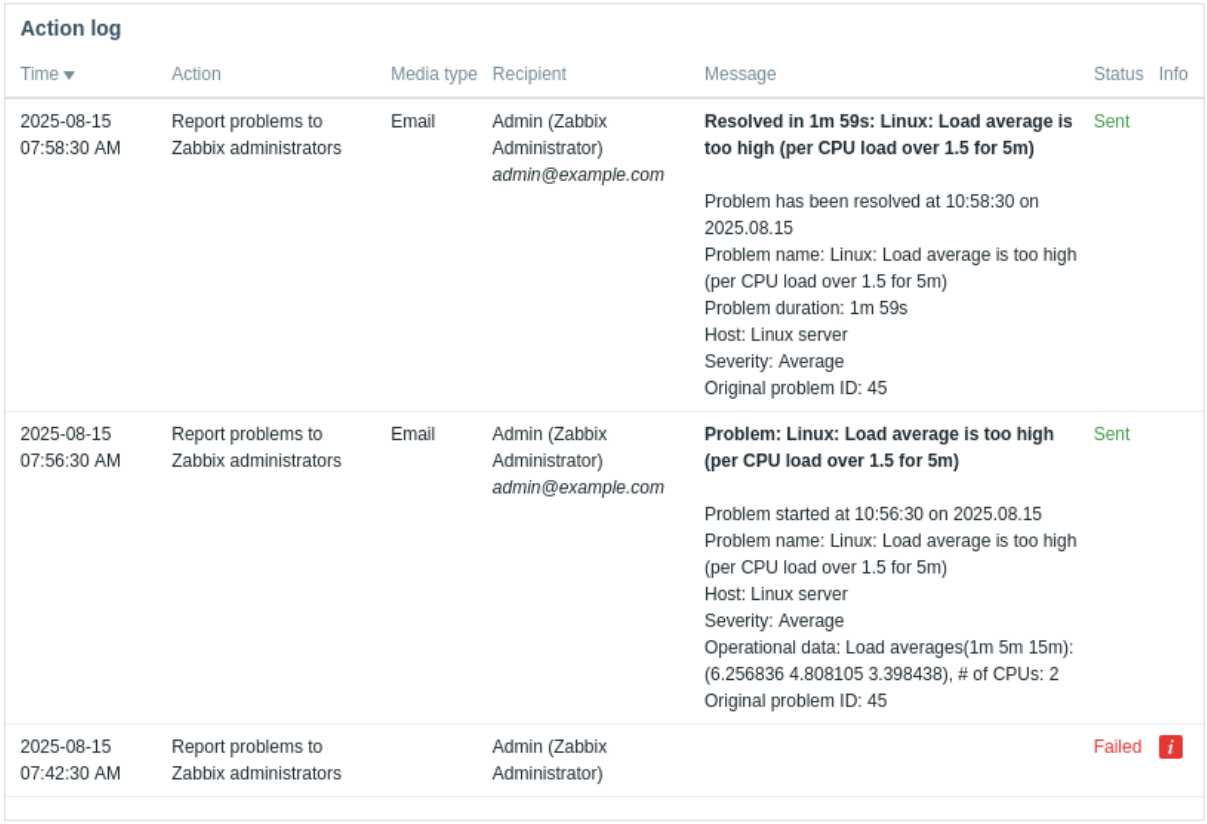
- Si le widget source de données ne diffuse pas de données, le widget à l'écoute passe à l'état *Awaiting data*.
- Si le widget source de données a été supprimé, remplacé par un widget incompatible ou déplacé vers une autre page du tableau de bord, le widget à l'écoute passe à l'état *Referred widget 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!"*



1 Journal des actions

Aperçu

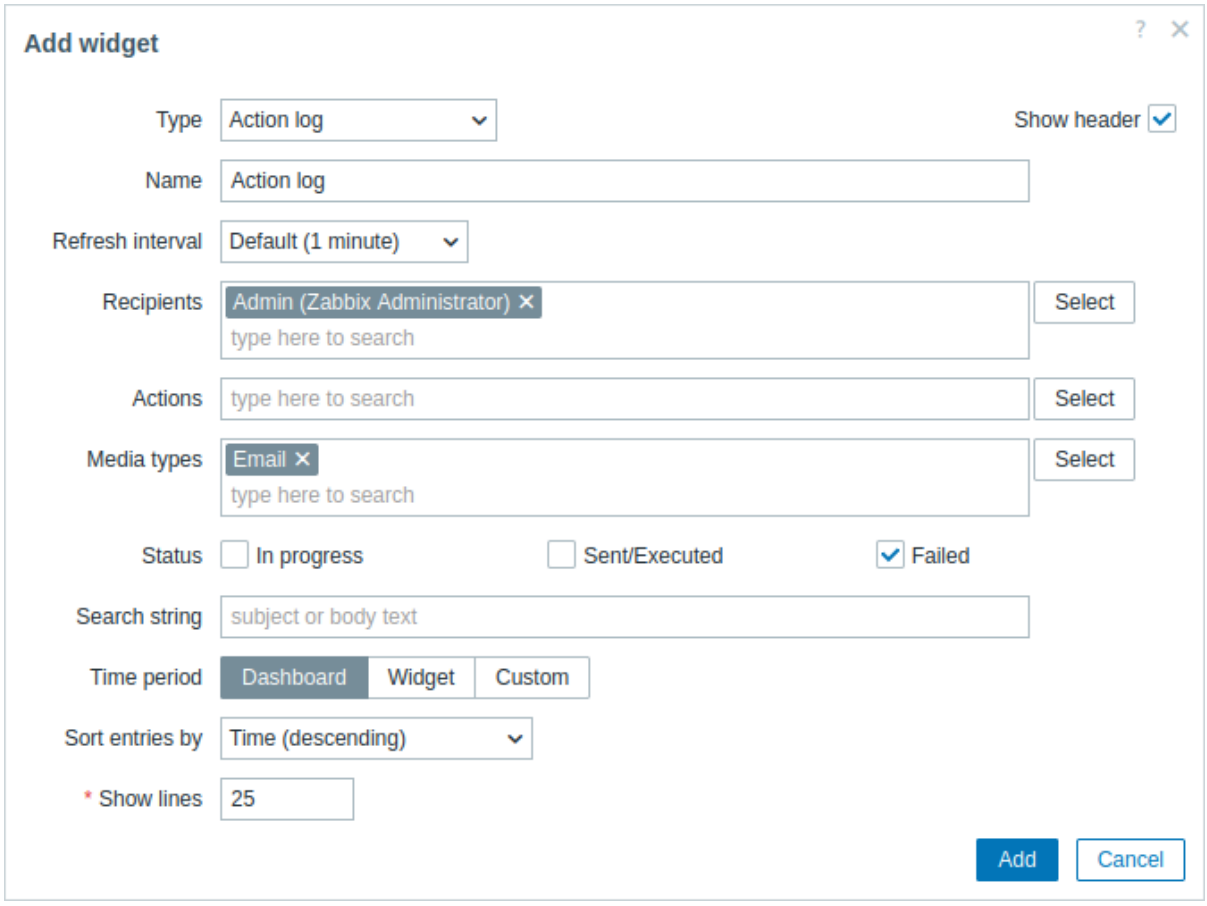
Le widget *Journal des actions* affiche les détails des opérations (notifications, commandes distantes) exécutées dans le cadre d'une action. Le journal détaillé facilite l'audit, le dépannage et le suivi de l'exécution de chaque action.



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 :



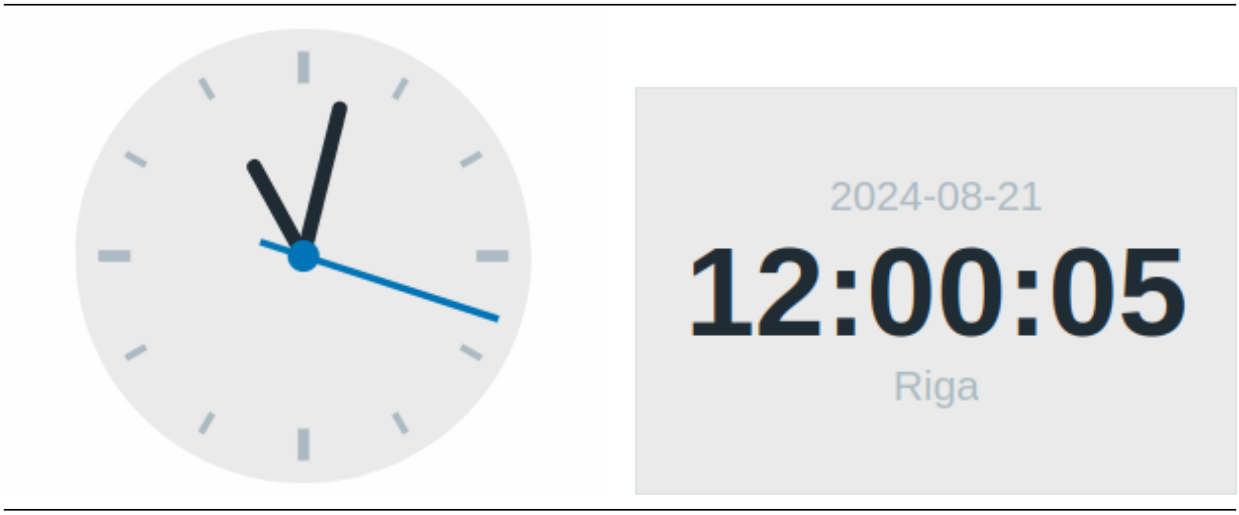
En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

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.

2 Horloge

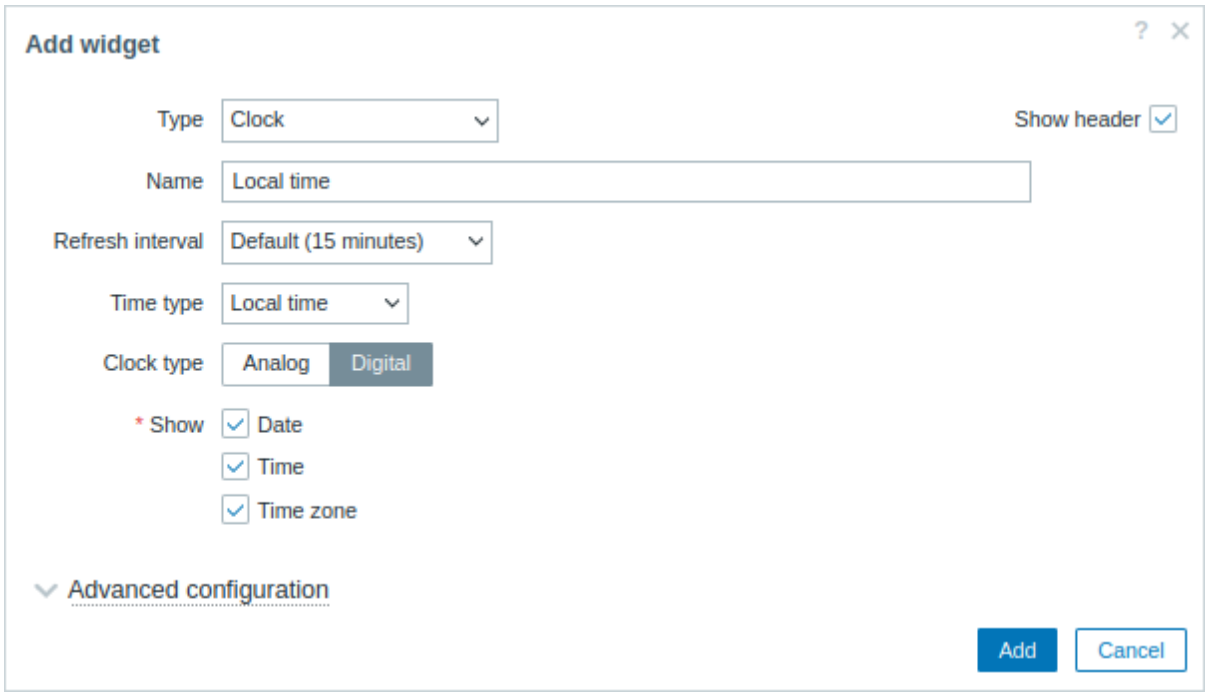
Vue d'ensemble

Le widget *Clock* affiche l'heure locale, du serveur ou de l'hôte au format analogique ou numérique.



Configuration

Pour configurer, sélectionnez *Horloge* comme type :



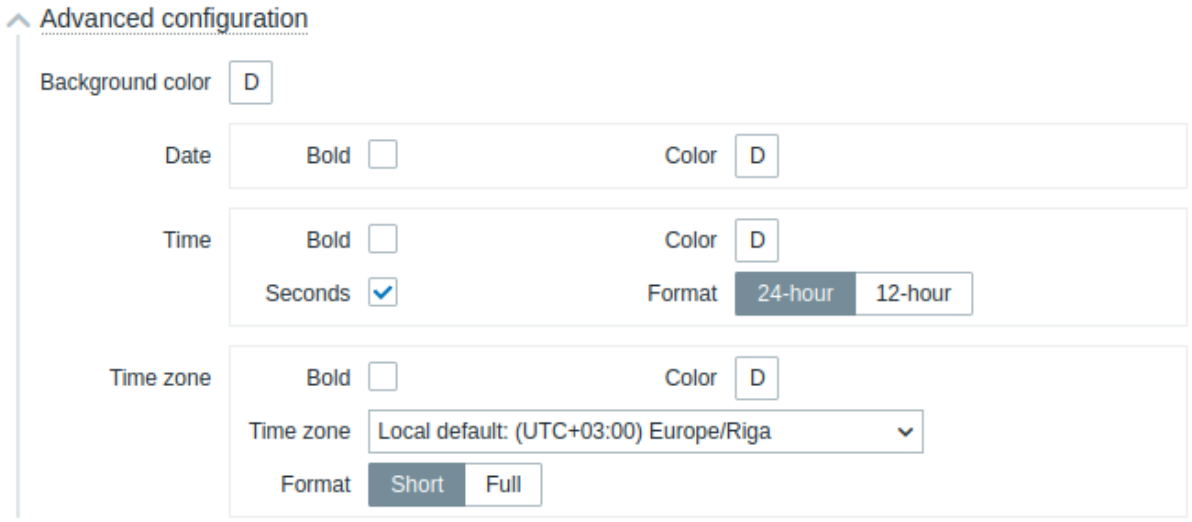
En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

Type d'heure	Sélectionnez la source de l'heure : **Local** - heure système de l'appareil exécutant le navigateur ; **Serveur** - fuseau horaire défini globalement ou pour l'utilisateur Zabbix ; **Hôte** - heure récupérée à partir d'un élément d'hôte.
Élément	Sélectionnez l'élément pour l'heure de l'hôte (utilisez l'élément system.localtime[local]). Ce paramètre est disponible si *Type d'heure* est défini sur « Heure de l'hôte ».
Type d'horloge	Sélectionnez le type d'horloge (analogique ou numérique).
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 l'échelle d'affichage. Ce paramètre est disponible si *Type d'horloge* est défini sur « Numérique ».
Configuration avancée	Cliquez sur l'intitulé *Configuration avancée* pour afficher les options de configuration avancée de l'horloge numérique. Cette section est disponible si *Type d'horloge* est défini sur « Numérique ».

Configuration avancée

Des 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).

De plus, la configuration avancée permet de modifier la couleur de fond de l'ensemble du widget.

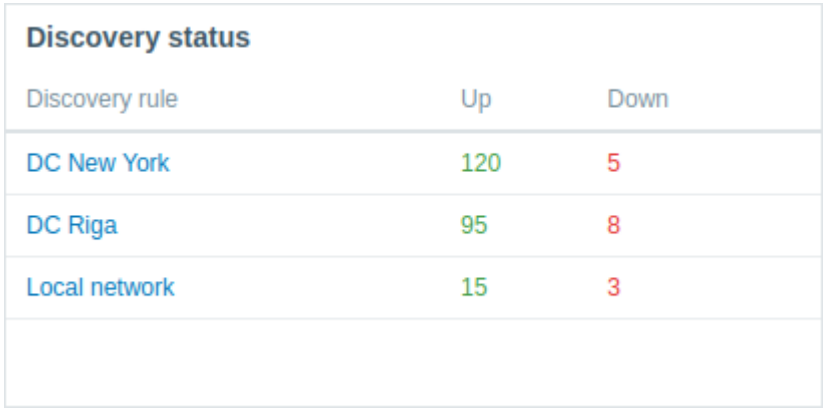


Couleur de fond	Sélectionnez la couleur de fond à l'aide du sélecteur de couleurs. `D` correspond à la couleur par défaut (dépend du thème de l'interface). Pour revenir à la valeur par défaut, cliquez sur le bouton *Utiliser la valeur par défaut* dans le sélecteur de couleurs.
Date	
Gras	Cochez la case pour afficher la date en gras.
Couleur	Sélectionnez la couleur de la date 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 *Utiliser la valeur par défaut* dans le sélecteur de couleurs.
Heure	
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 *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	
Gras	Cochez la case pour afficher le fuseau horaire en gras.
Couleur	Sélectionnez la couleur du fuseau horaire 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 *Utiliser la valeur par défaut* dans le sélecteur de couleurs.
Fuseau horaire	Sélectionnez le fuseau horaire.
Format	Sélectionnez l'affichage du fuseau horaire au format court (par ex. `New York`) ou au format complet (par ex. `(UTC-04:00) America/New York`).

3 État de la découverte

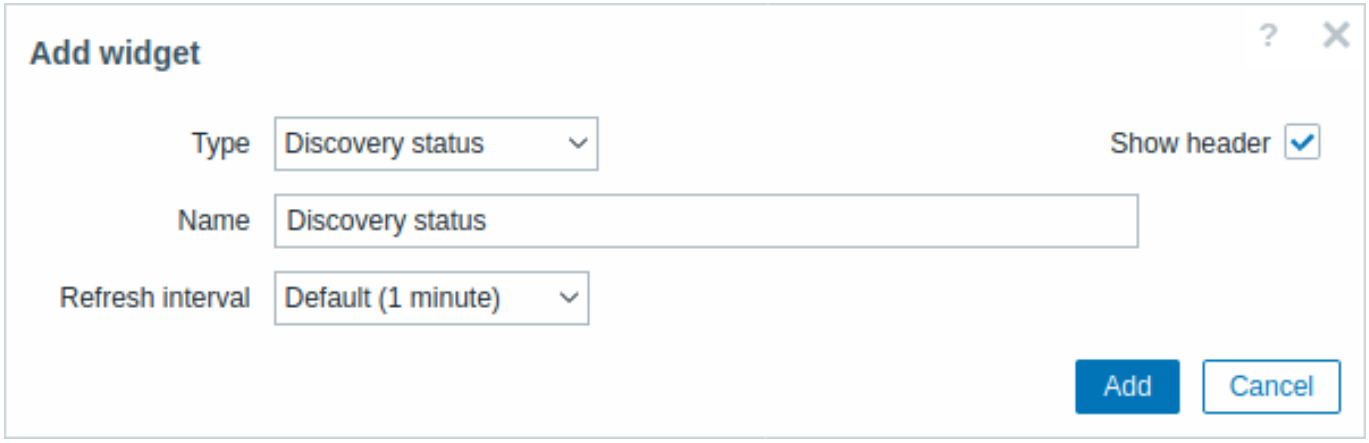
Vue d'ensemble

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

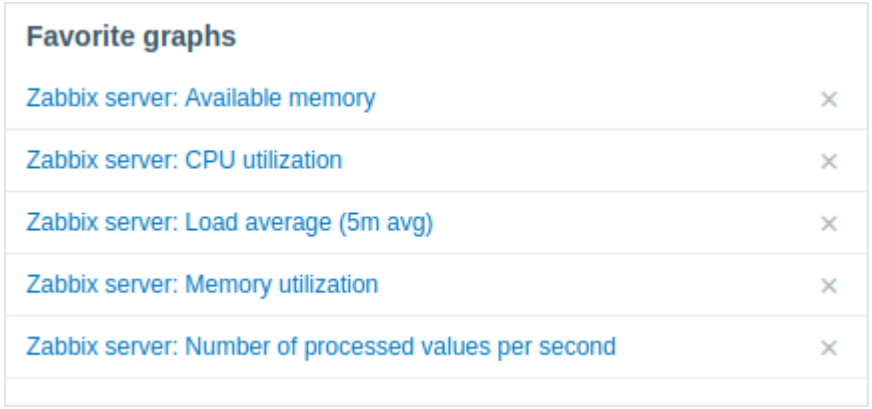
Tous les paramètres de configuration de ce widget sont communs à tous les widgets.



4 Graphiques favoris

Aperçu

Le widget *Graphiques favoris* affiche une liste alphabétique de raccourcis vers les graphiques marqués comme favoris par l'utilisateur actuel.



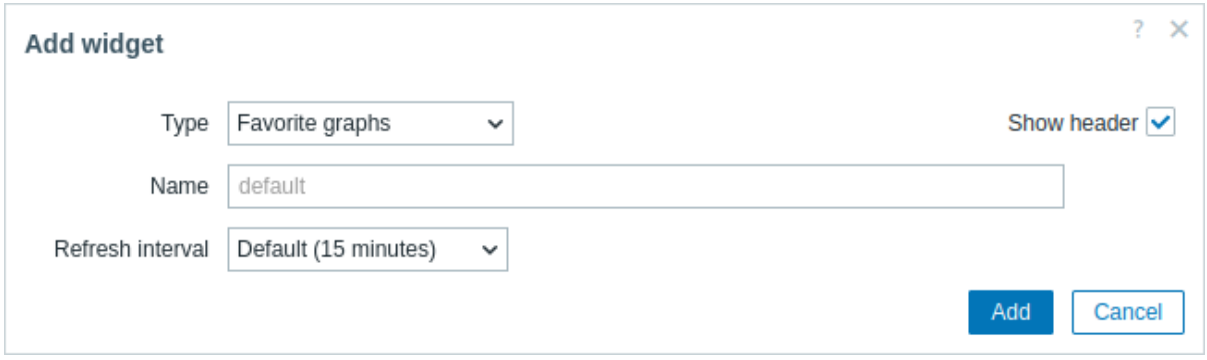
Un clic sur x supprime le graphique de la liste et des favoris.

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

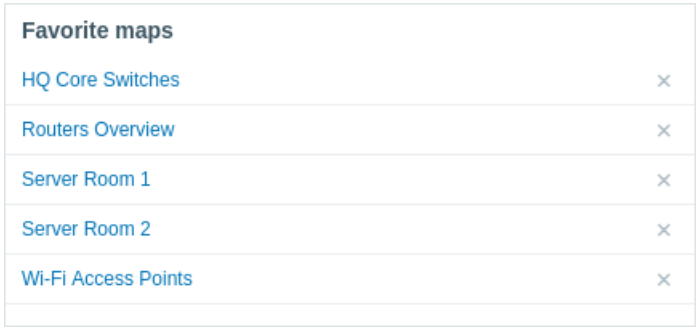
Tous les paramètres de configuration de ce widget sont communs à tous les widgets.



5 Cartes favorites

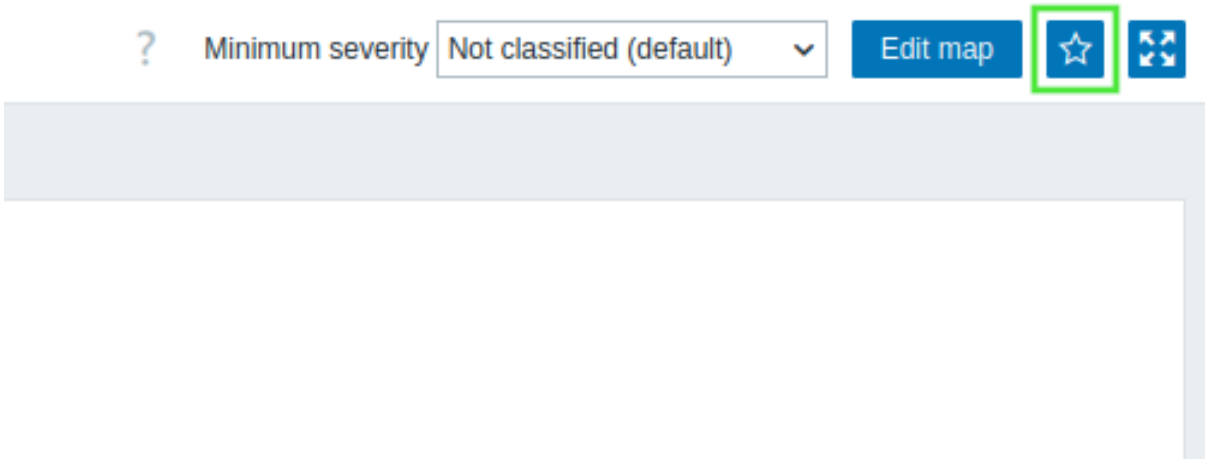
Aperçu

Le widget *Cartes favorites* affiche une liste de raccourcis vers les cartes marquées comme favorites par l'utilisateur actuel, triées par ordre alphabétique.



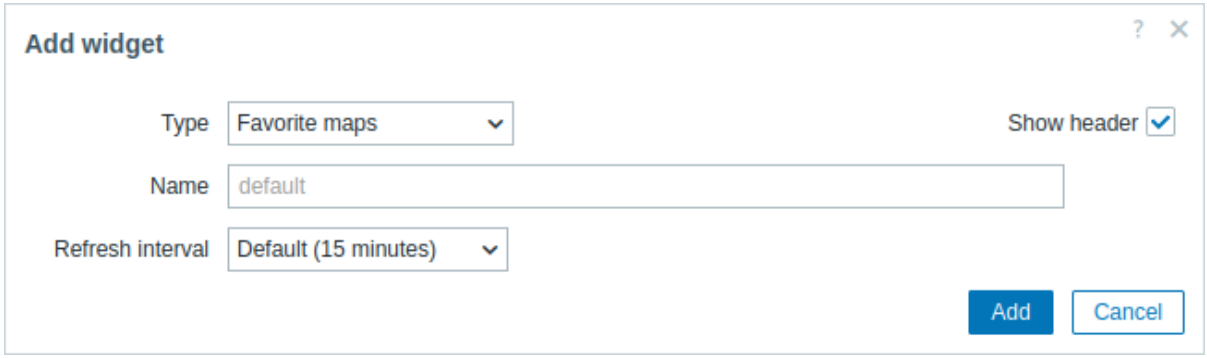
Un clic sur x supprime la carte des favoris.

Les cartes sont marquées comme favorites en cliquant sur le bouton  *Ajouter aux favoris* lors de l'affichage des cartes dans *Monitoring* > *Maps*.



Configuration

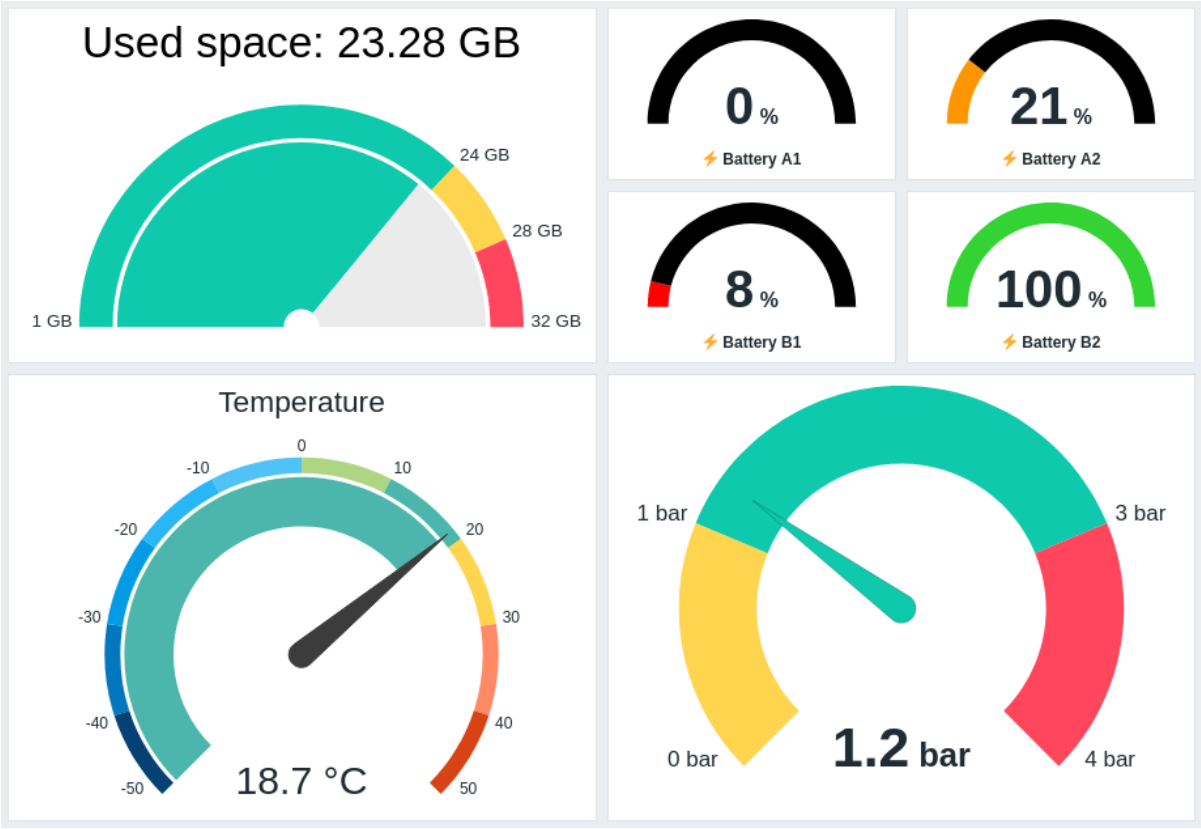
Tous les paramètres de configuration de ce widget sont communs à tous les widgets.



6 Jauge

Vue d'ensemble

Le widget *Gauge* affiche la valeur d'un seul élément numérique sous forme de jauge. Il est utile pour surveiller des métriques importantes, visualiser des seuils et détecter des changements soudains.



Vous pouvez configurer le widget pour afficher les éléments suivants :

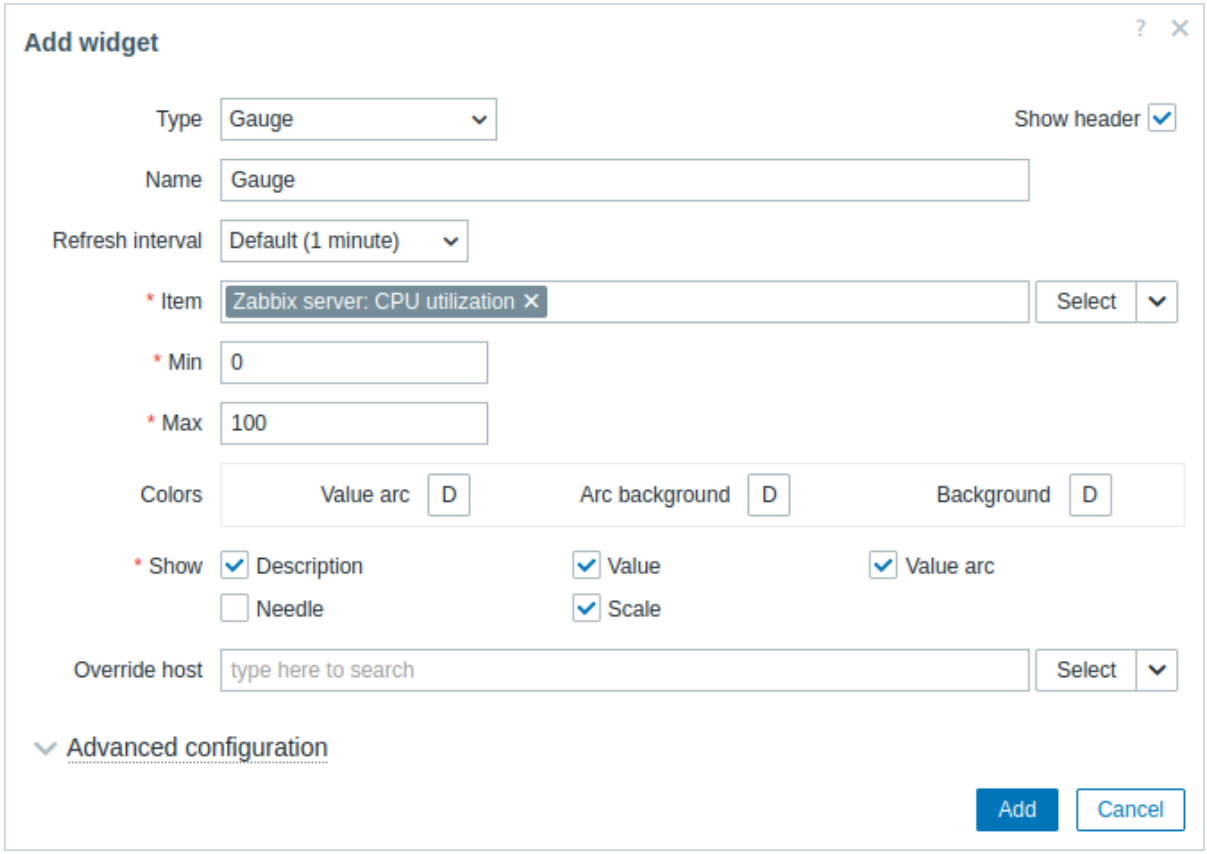
- Description de l'élément (Espace utilisé : {ITEM.LASTVALUE}, Température)
- Valeur de l'élément et unités (21 %, 18.7 °C, 1.2 bar)
- Échelle de la jauge (1GB/24GB/28GB/32GB, -50/-40/-30/etc.)
- Arc de la jauge (arc de valeur de la jauge et arc des seuils de la jauge)
- Aiguille de la jauge

Un clic sur le widget ouvre un graphique simple pour l'élément.

Les informations affichées dans le widget *Gauge* peuvent être téléchargées sous forme d'image PNG en sélectionnant l'option *Download image* dans le menu du widget.

Configuration

Pour configurer, sélectionnez *Gauge* comme type :



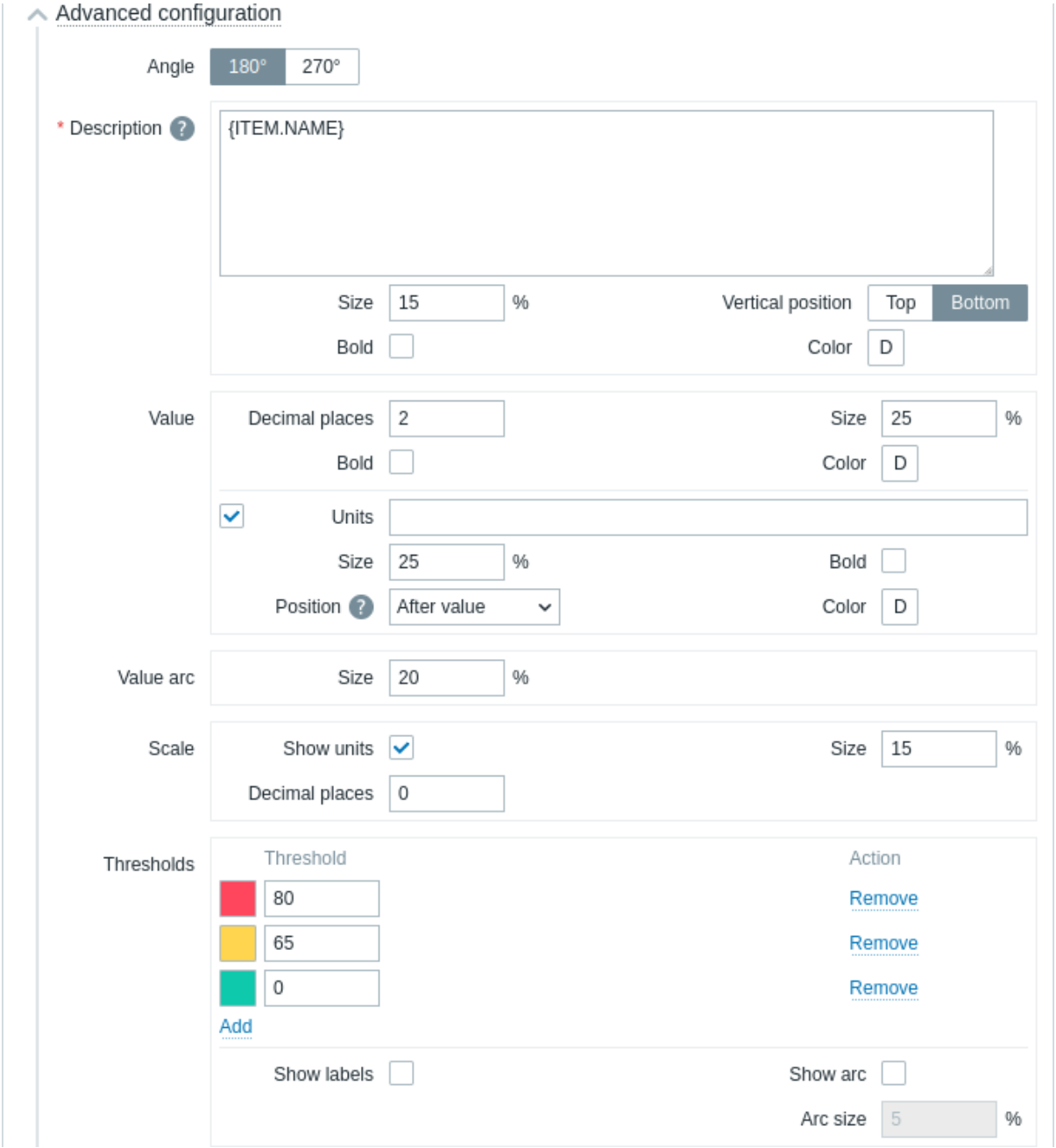
En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

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 s'affiche. 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 web. 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 correspondant de la jauge : 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 avec 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'intitulé *Advanced configuration* pour afficher les options de configuration avancée. C'est également ici que vous pourrez ajuster les éléments de la jauge sélectionnés dans le champ *Show*.

Configuration avancée

Les options de configuration avancée sont disponibles dans la section repliable *Configuration avancée* :



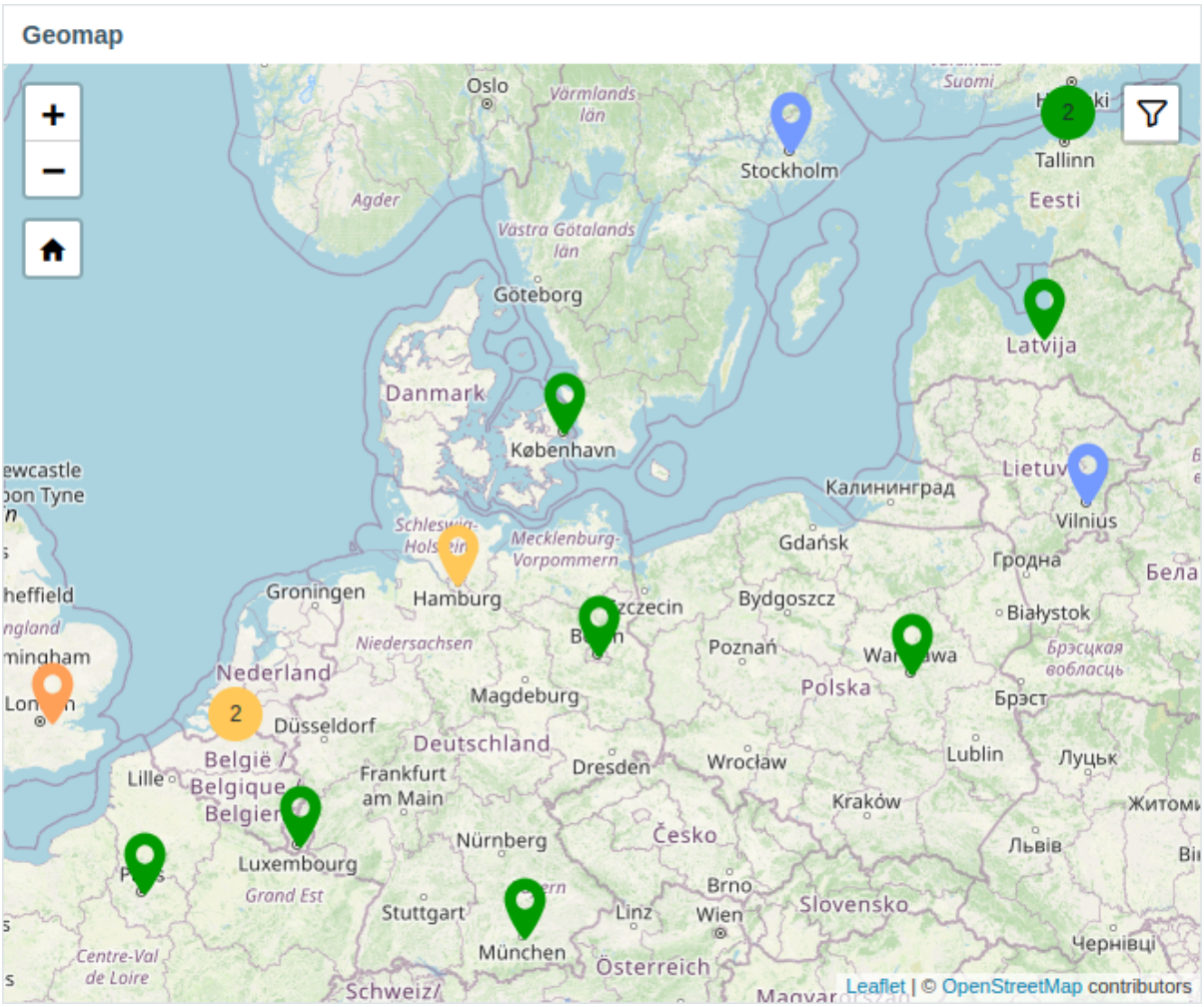
Angle	Sélectionnez l'angle de la jauge (180° ou 270°).
Description	
Description	Saisissez la description de l'élément. Cette description peut remplacer le nom d'élément par défaut. Les descriptions sur plusieurs lignes sont prises en charge. Une combinaison de texte et de macros prises en charge est 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).

7 Géocarte

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.

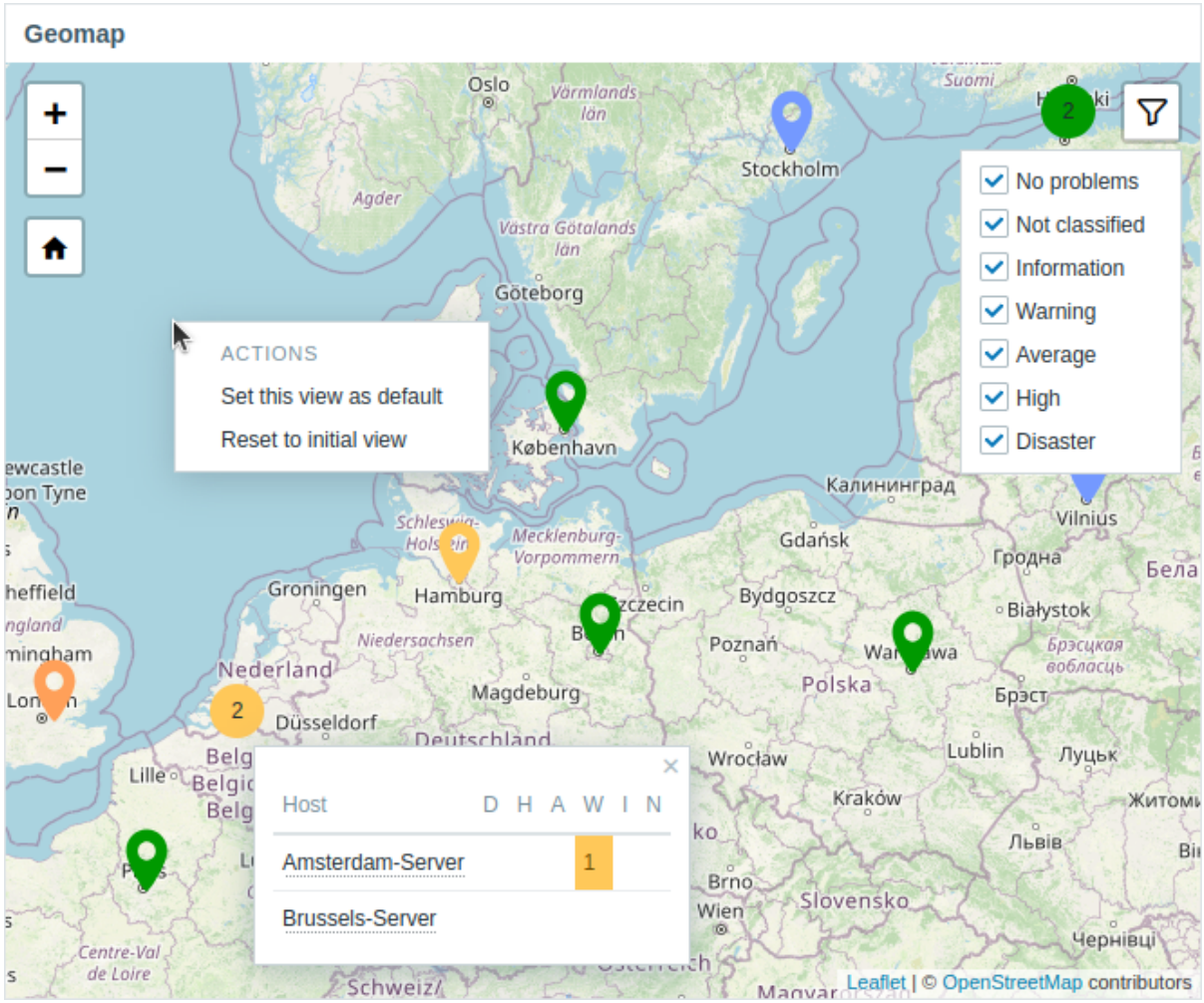


Note:
Zabbix propose plusieurs fournisseurs de tuiles cartographiques prédéfinis et vous permet d'en ajouter des personnalisés dans *Administration* > *General* > *Geographical maps*.

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.

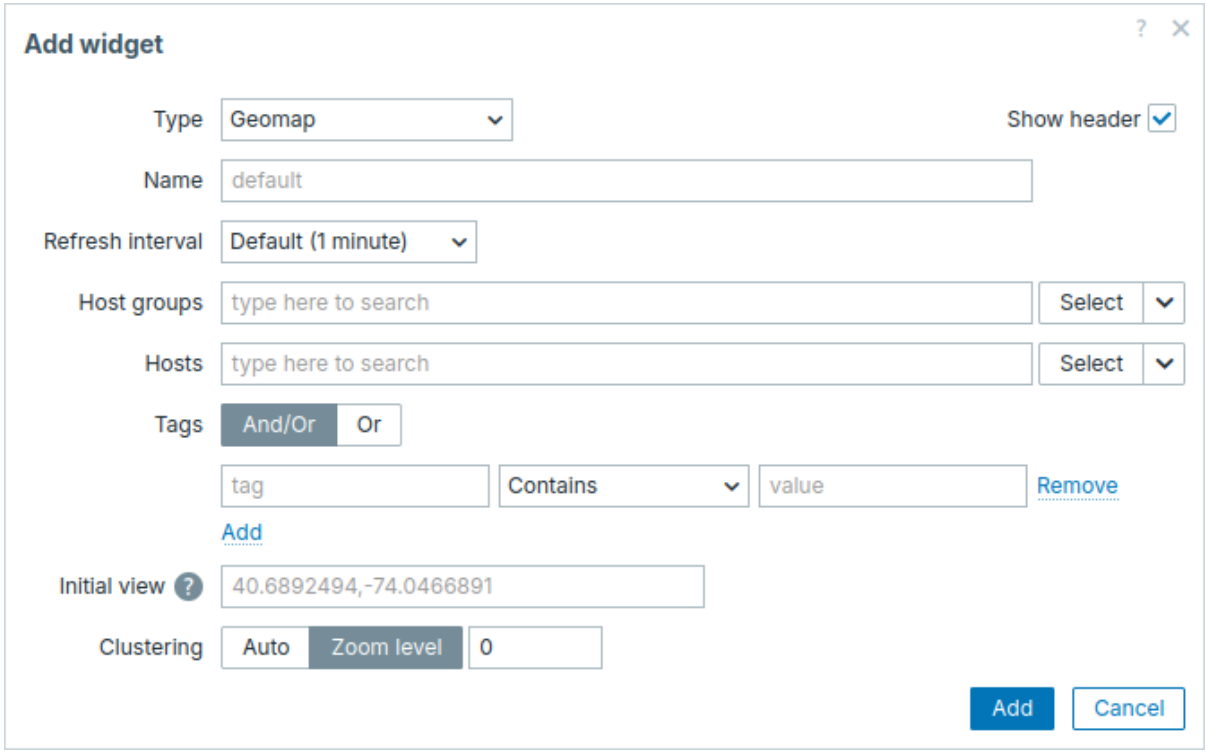
Utilisez les options suivantes pour interagir avec le widget :

- 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.



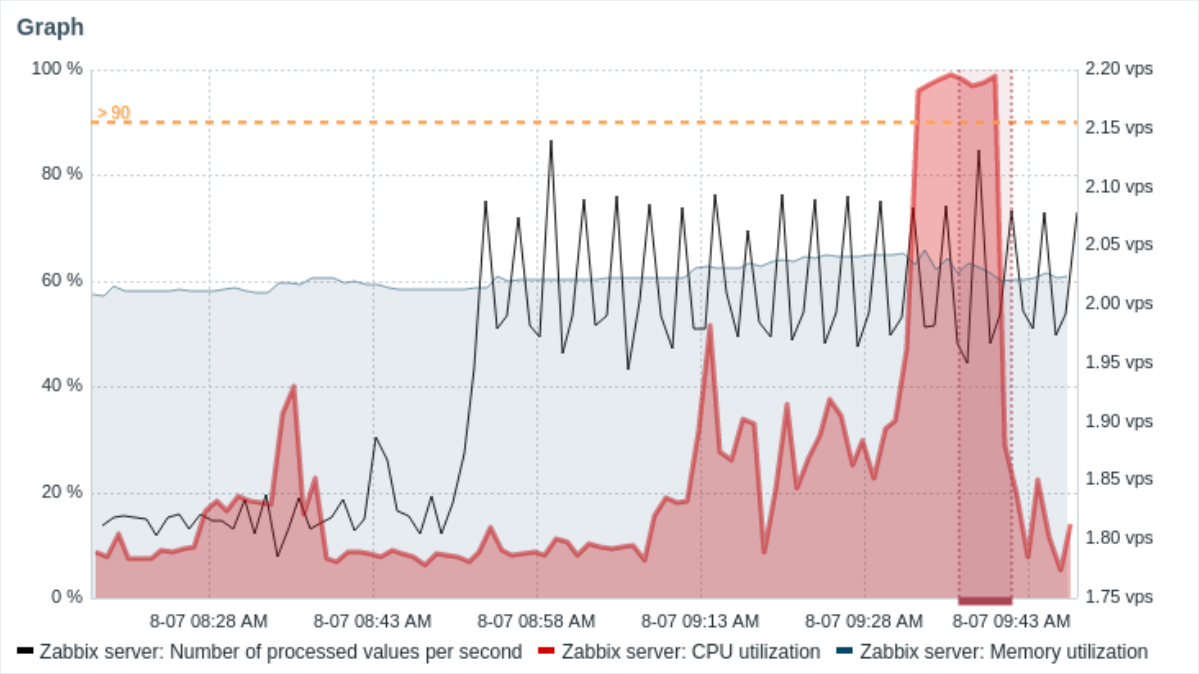
En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

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 s'affiche. Si rien n'est sélectionné dans les champs *Groupes d'hôtes* et *Hôtes*, tous les hôtes avec 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 s'affiche. Si rien n'est sélectionné dans les champs *Groupes d'hôtes* et *Hôtes*, tous les hôtes avec 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 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 : **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.
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 maximal 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.

8 Graphique

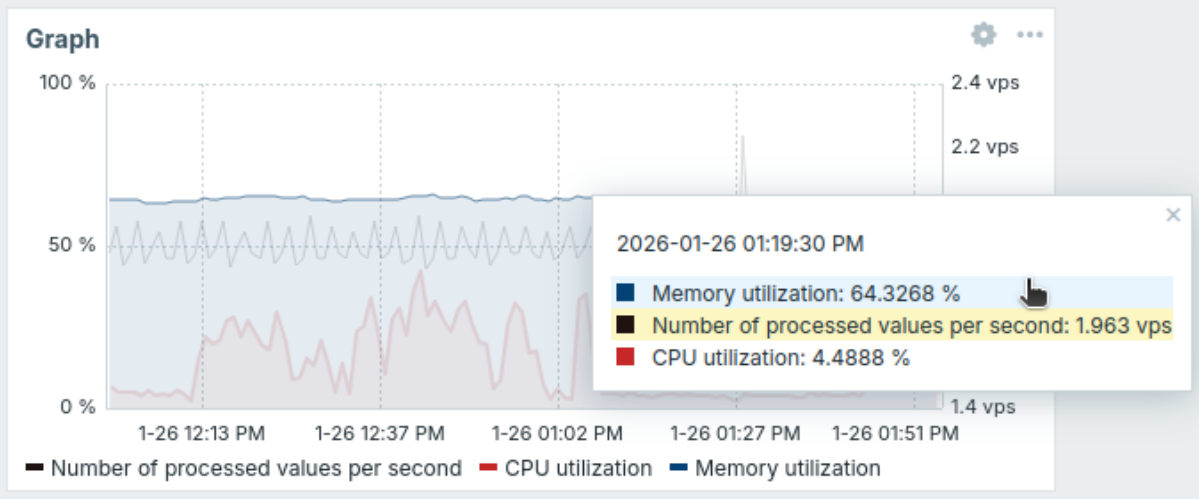
Vue d'ensemble

Le widget *Graph* 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.



En survolant le graphique, une infobulle s'affiche avec une liste des éléments triés par valeur dans l'ordre décroissant.

En survolant un élément dans l'infobulle, son graphique est mis en évidence (les autres sont estompés) ; en sélectionnant l'élément, ses données sont diffusées vers d'autres widgets.

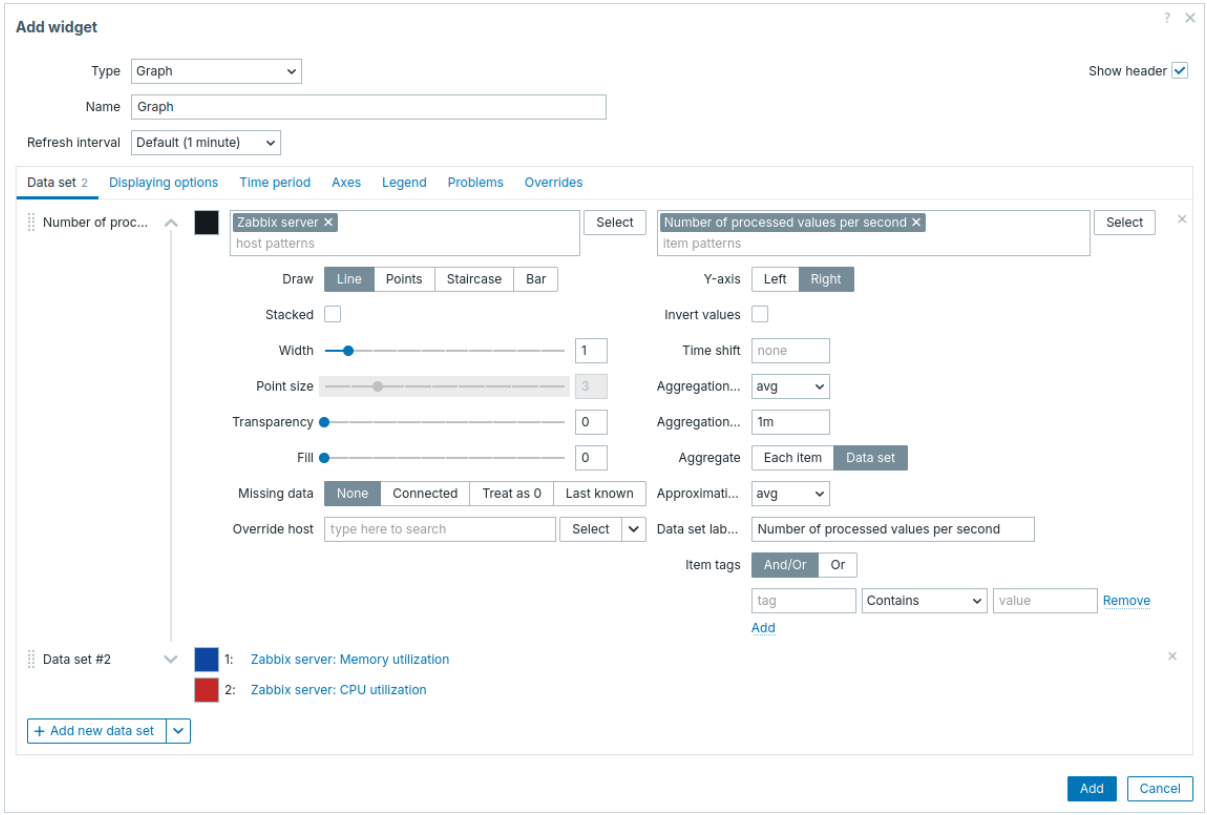


L'ancien widget Graph (classic) est également disponible.

Les informations affichées dans le widget *Graph* peuvent être téléchargées sous forme d'image PNG en sélectionnant l'option *Download image* dans le menu du widget.

Configuration

Pour configurer, sélectionnez *Graph* comme type :



Jeu de données

L'onglet **Jeu de données** permet de sélectionner les données du graphique 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 seule couleur de base 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'hôte et d'élément ; les données des éléments correspondant à ces modèles seront affichées sur le graphique ; jusqu'à 50 éléments peuvent être affichés. Des modèles génériques peuvent être utilisés pour la sélection (par exemple, ∗ renverra des résultats correspondant à zéro ou plusieurs caractères). Pour spécifier un modèle générique, saisissez la chaîne manuellement 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'hôte et d'élément est obligatoire pour les jeux de données « Modèles d'éléments ». Voir aussi : Détails de configuration du jeu de données. Pour le jeu de données **Liste d'éléments** : Sélectionnez les éléments du 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 jeux de données « Liste d'éléments ». Voir aussi : Détails de 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'hôte n'est pas disponible, et le paramètre permettant de spécifier une liste d'éléments permet de sélectionner uniquement les 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 *Barre*. Notez que s'il n'y a qu'un seul point de données dans le graphique en ligne/en escalier, il est dessiné comme un point quel que soit le type de tracé. La taille du point est calculée à partir de la largeur de ligne, mais elle ne peut pas être inférieure à 3 pixels, même si la largeur de ligne est inférieure.
Empilé	Cochez la case pour afficher les données empilées (zones remplies affichées). Cette option est désactivée lorsque le type de tracé *Points* est sélectionné.
Largeur	Définissez la largeur de ligne. Cette option est disponible lorsque le type de tracé *Ligne* ou *Escalier* est sélectionné.
Taille du point	Définissez la taille du point. 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 reste vide ; **Connecté** - les deux valeurs de bord sont reliées ; **Traiter comme 0** - les données manquantes sont affichées comme des valeurs 0 ; **Dernière valeur connue** - les données manquantes sont affichées avec la même valeur que la dernière valeur connue ; non applicable aux types de tracé *Points* et *Barre*.
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.
Axe Y	Sélectionnez le côté du graphique où l'axe Y sera affiché.
Décalage temporel	Spécifiez un décalage temporel si nécessaire. Vous pouvez utiliser des suffixes de temps dans ce champ. Les valeurs négatives sont autorisées.

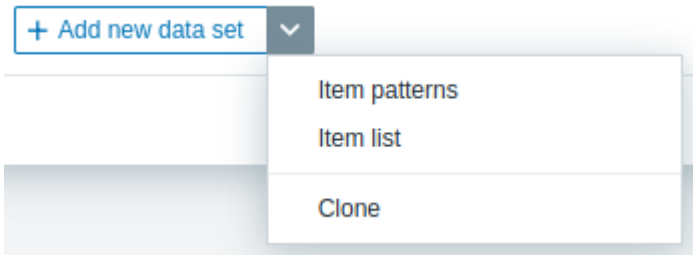
Fonction d'agrégation	Spécifiez la fonction d'agrégation à utiliser : **min** - afficher la plus petite valeur ; **max** - afficher la plus grande valeur ; **avg** - afficher la valeur moyenne ; **sum** - afficher la somme des valeurs ; **count** - afficher le nombre de valeurs ; **first** - afficher la première valeur ; **last** - afficher la dernière valeur ; **none** - afficher toutes les valeurs (sans agrégation). L'agrégation permet d'afficher une valeur agrégée pour l'intervalle choisi (5 minutes, une heure, un jour), au lieu de toutes les valeurs. Voir aussi : Agrégation dans les graphiques.
Intervalle d'agrégation	Spécifiez l'intervalle d'agrégation des valeurs. Vous pouvez utiliser des suffixes de temps dans ce champ. Une valeur numérique sans suffixe sera considérée comme des secondes. Notez que si le widget est configuré pour afficher des données historiques basées sur les tendances (*Sélection des données d'historique* est définie sur *Tendances* ou *Auto*), il est recommandé d'utiliser un intervalle d'agrégation multiple de 1 heure (par ex. 3600, 60m, 1h, 3h, etc.). Les tendances stockent des valeurs agrégées par heure, donc l'utilisation d'un intervalle d'agrégation qui n'est pas un multiple de 1 heure (par ex. 100s, 7min, 15min, 90min, etc.) peut conduire à des résultats difficiles à interpréter.
Agréger	Indiquez s'il faut agréger : **Chaque élément** - chaque élément du jeu de données sera agrégé et affiché séparément ; **Jeu de données** - tous les éléments du jeu de données seront agrégés et affichés comme une seule valeur.
Approximation	Indiquez quelle valeur afficher lorsque plus d'une valeur existe 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 sur une longue période avec un intervalle de mise à jour fréquent (par exemple, une année de valeurs collectées toutes les 10 minutes).
Libellé du jeu de données	Spécifiez le libellé du jeu de données affiché dans la configuration *Jeu de données* du graphique et dans la *Légende* du graphique (pour les jeux de données agrégés). Tous les jeux de données sont numérotés, y compris ceux ayant un *Libellé du jeu de données* spécifié. Si aucun libellé n'est spécifié, le jeu de données sera libellé automatiquement selon son numéro (par ex. « Jeu de données n°2 », « Jeu de données n°3 », etc.). La numérotation des jeux de données est recalculée après réorganisation/glisser-déposer des jeux de données. Les libellés de jeu de données trop longs seront raccourcis pour s'adapter à l'emplacement d'affichage (par ex. « Nombre de proc... »).

Tags d'élément	Spécifiez des tags pour filtrer les éléments 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 : **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). 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.

Détails de la configuration des jeux de données

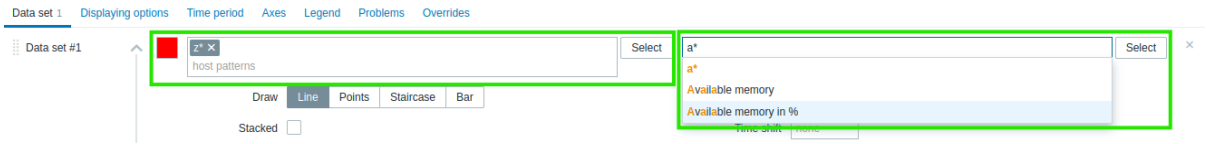
Les jeux de données existants sont affichés dans une liste. Vous pouvez :

- Cliquer sur l'icône de déplacement et faire glisser un jeu de données vers un nouvel emplacement dans la liste.
- Cliquer sur l'icône de développement pour afficher les détails du jeu de données. Lorsqu'elle est développée, cette 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 uni, ou passer à l'onglet *Palette* et sélectionner une ligne de couleurs prédéfinies. La couleur choisie s'applique directement aux jeux de données *Item list* ou sert de base aux nuances générées dans *Item patterns*. Utilisez Tab pour passer d'un contrôle de la boîte de dialogue à l'autre, les touches fléchées pour parcourir les échantillons ou les lignes de la palette, Entrée pour sélectionner et Échap pour annuler.
- Cliquer sur le bouton *Add new data set* pour ajouter un jeu de données vide permettant de sélectionner des motifs d'hôte et d'élément. Si vous cliquez sur l'icône pointant vers le bas à côté du bouton *Add new data set*, un menu déroulant apparaît, vous permettant d'ajouter un nouveau jeu de données *Item patterns* ou *Item list*, ou de *Clone* le jeu de données actuellement ouvert. Si tous les jeux de données sont réduits, l'option *Clone* n'est pas disponible.

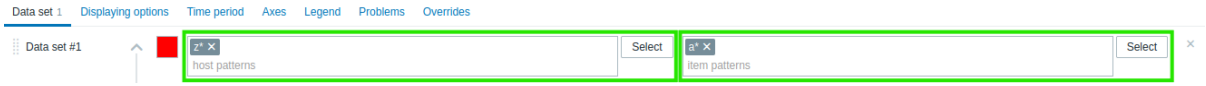


L'ensemble de données **Modèles d'éléments** contient les champs *Modèles d'hôtes* et *Modèles d'éléments*, qui reconnaissent tous deux les noms complets ou les modèles contenant un symbole générique (*). Cette fonctionnalité vous permet de sélectionner tous les noms d'hôtes et noms d'éléments contenant 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.

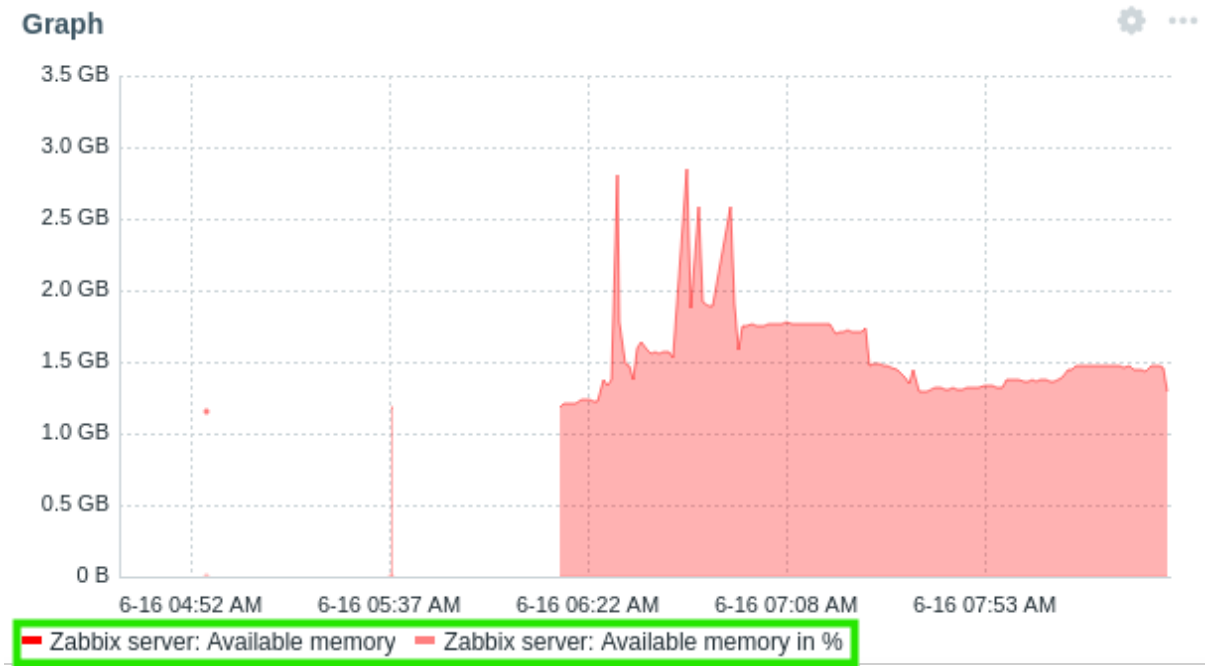
Par exemple, après avoir saisi le modèle **z*** dans le champ *Modèles d'hôtes*, la liste déroulante affiche tous les noms d'hôtes contenant ce modèle : **z*, Zabbix server, Zabbix proxy**. Après avoir appuyé sur *Entrée*, ce modèle est accepté et affiché comme **z***. De même, après avoir saisi le modèle **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 %**.



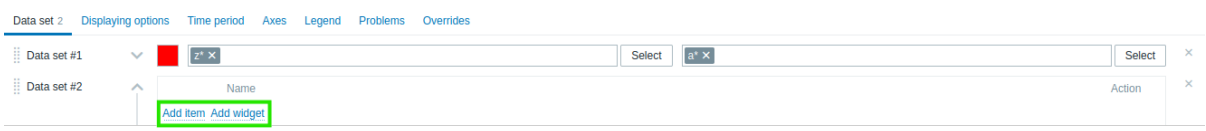
Après avoir appuyé sur *Entrée*, le modèle est accepté et affiché comme **a***.



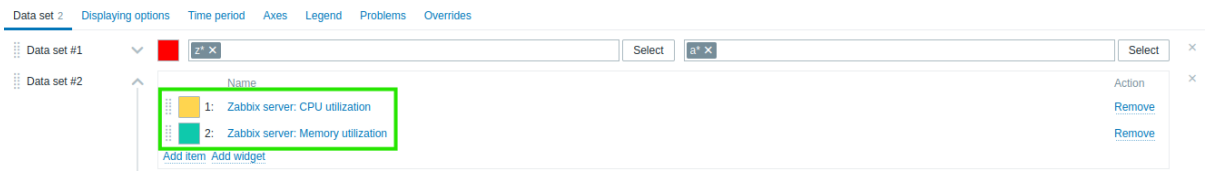
Le graphique affiche alors tous les éléments appartenant au(x) nom(s) d'hôte sélectionné(s).



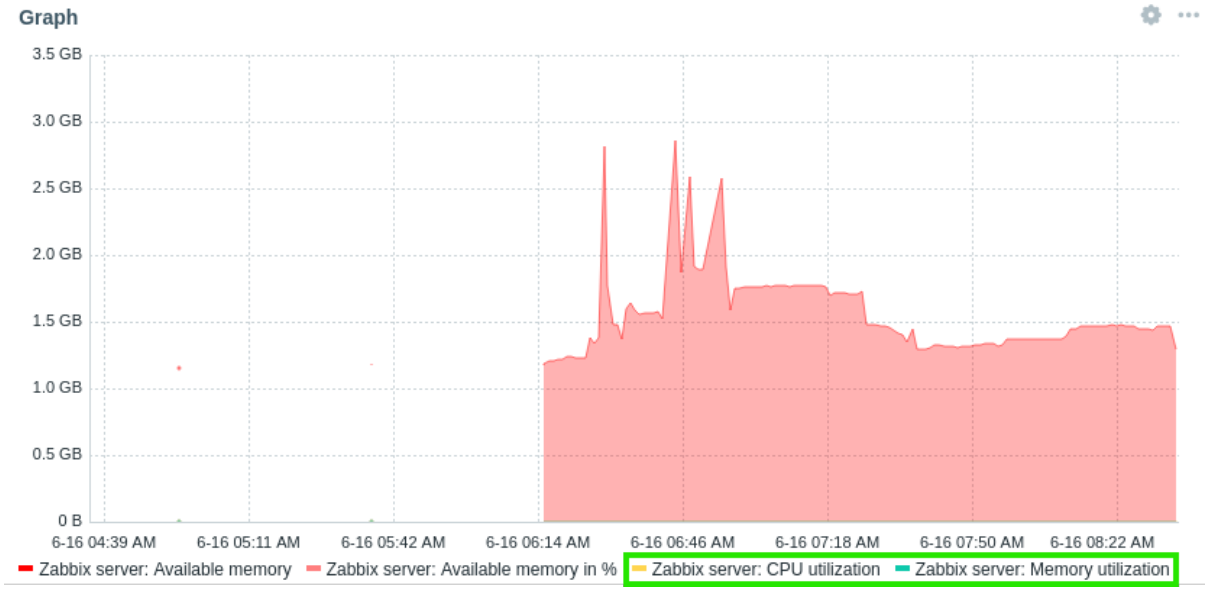
L'ensemble de données **Liste des éléments** contient le bouton *Ajouter un élément*, qui vous permet d'ajouter des éléments à afficher sur le graphique. Vous pouvez également ajouter des widgets compatibles comme source de données pour les éléments en cliquant sur le bouton *Ajouter un widget*.



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.

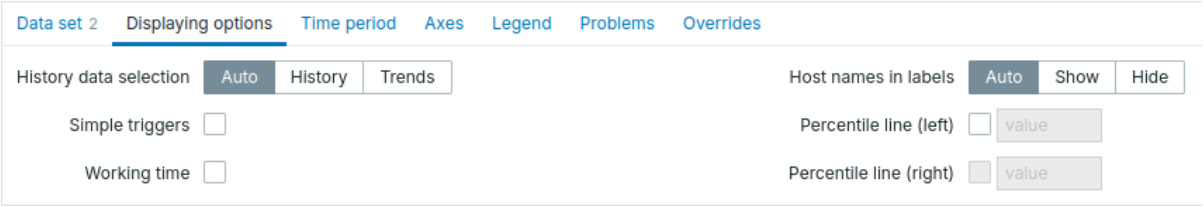


Après avoir sélectionné un ou plusieurs éléments, ils seront affichés dans la liste des éléments de l'ensemble de données ainsi que dans le graphique.



Options d'affichage

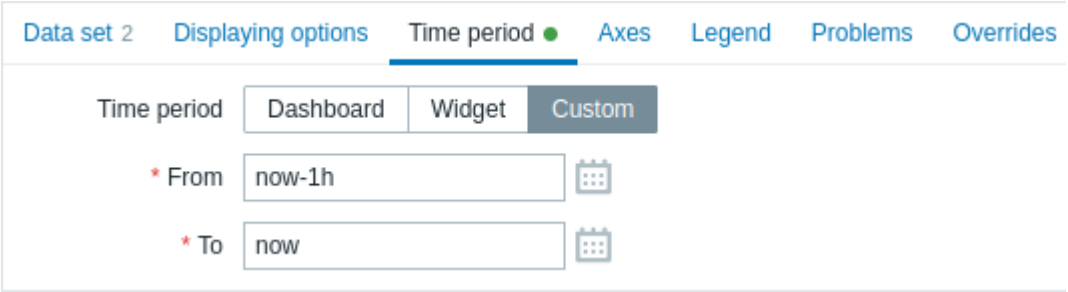
L'onglet **Options d'affichage** permet de définir la sélection des données d'historique :



Sélection des données d'historique	Définissez la source des données du graphique : **Auto** - les données sont sélectionnées selon l'algorithme des graphiques classiques (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 seront tracés sous forme de lignes en pointillés en utilisant la couleur de sévérité du déclencheur. Un déclencheur simple est un déclencheur avec une seule fonction (uniquement `last`, `max`, `min`, `avg`) pour un seul élément dans l'expression. Un maximum de trois déclencheurs peut être tracé. Notez que le déclencheur doit se trouver dans la plage affichée pour être visible.
Temps de travail	Cochez la case pour afficher le temps de travail sur le graphique. 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 frontal par défaut *Original blue*).
Noms d'hôte dans les étiquettes	Choisissez s'il faut afficher les noms d'hôte dans la légende du graphique et dans l'infobulle qui apparaît lors du survol des données du graphique : **Auto** - les noms d'hôte sont affichés uniquement si plusieurs hôtes sont présents dans les jeux de données (par défaut). **Afficher** - les noms d'hôte sont affichés. **Masquer** - les noms d'hôte 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. Si, par exemple, un percentile de 95 % est défini, la ligne de percentile se situera au niveau sous lequel se trouvent 95 pour cent 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. Si, par exemple, un percentile de 95 % est défini, la ligne de percentile se situera au niveau sous lequel se trouvent 95 pour cent des valeurs.

Période de temps

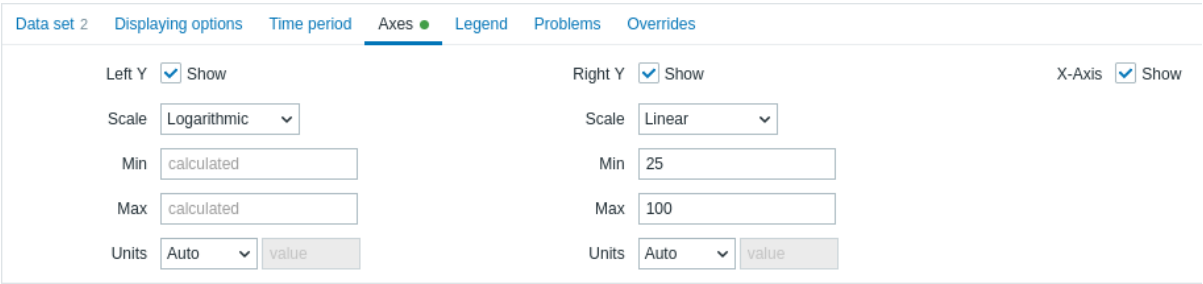
L'onglet **Période de temps** permet de définir une période pour laquelle afficher les données dans le graphique :



Période de temps	Sélectionnez la source de données pour la période de temps : **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 de temps* 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 *Période de temps* est défini sur "Widget".
De	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 *Période de temps* est défini sur "Personnalisée".
À	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 *Période de temps* est défini sur "Personnalisée".

Axes

L'onglet **Axes** permet de personnaliser l'affichage des axes :

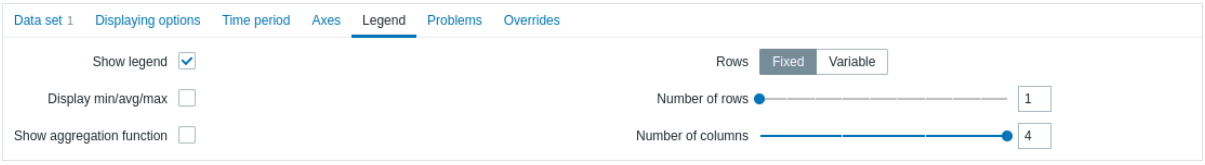


Y gauche	Cochez cette case pour rendre l'axe Y gauche visible. La case à cocher peut être désactivée si elle est décochée soit dans l'onglet *Data set*, soit dans l'onglet *Overrides*.
Y droite	Cochez cette case pour rendre l'axe Y droite visible. La case à cocher peut être désactivée si elle est décochée soit dans l'onglet *Data set*, soit dans l'onglet *Overrides*.
Axe X	Décochez cette case pour masquer l'axe X (coché 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 ex., 10, 20, 30), adaptée aux données qui évoluent régulièrement ou couvrent une plage faible à modérée ; **Logarithmique** - les valeurs de l'axe augmentent de façon exponentielle (par ex., 10, 100, 1000), adaptée 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 : **Auto** - les valeurs de l'axe sont affichées en utilisant l'unité du premier élément du jeu de données ; **Static** - les valeurs de l'axe sont affichées en utilisant l'unité spécifiée dans le champ de saisie *value* ; si le champ est laissé vide, seules les valeurs numériques sont affichées.

Légende

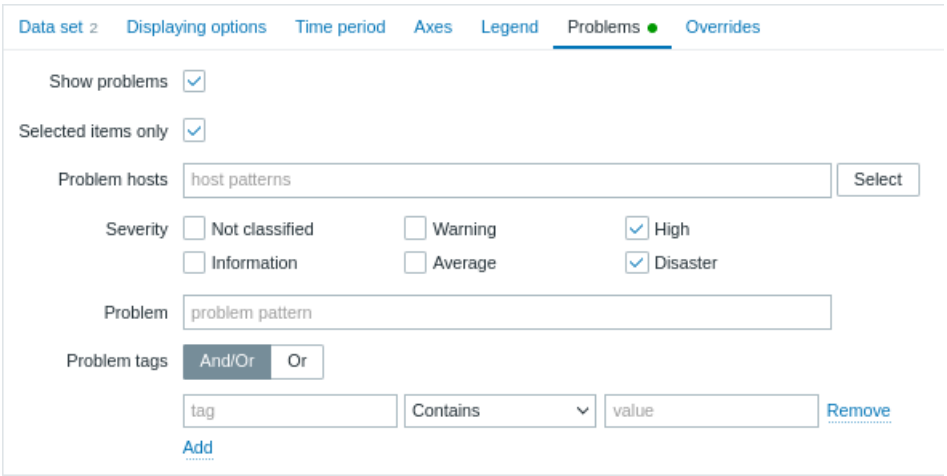
L'onglet **Légende** permet de personnaliser la légende du graphique :



Afficher la légende	Décochez cette case pour masquer la légende sur le graphique (cochée par défaut).
Afficher 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 la quantité 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 coché.

Problèmes

L'onglet **Problèmes** permet de personnaliser l'affichage des problèmes :

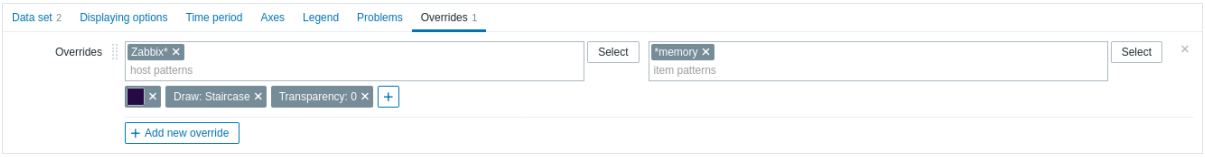


Afficher les problèmes	Cochez cette case pour activer l'affichage des problèmes sur le graphique (non cochée, c'est-à-dire désactivée par défaut).
Éléments sélectionnés uniquement	Cochez cette case pour inclure uniquement les problèmes des éléments sélectionnés à afficher sur le graphique.

Hôtes des problèmes	Sélectionnez les hôtes des problèmes à afficher sur le graphique. Des motifs génériques peuvent être utilisés (par exemple, ∗ renverra des résultats correspondant à zéro ou plusieurs caractères). Pour spécifier un motif générique, saisissez simplement la chaîne manuellement et appuyez sur *Entrée*. Pendant la saisie, notez 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.
Sévérité	Cochez les niveaux de sévérité des problèmes pour filtrer les problèmes à afficher sur le graphique. Si aucun niveau de sévérité n'est coché, tous les problèmes seront affichés.
Problème	Indiquez le nom du problème à afficher sur le graphique.
Balises de problème	Indiquez les balises de problème afin de limiter le nombre de problèmes affichés dans le widget. Il est possible d'inclure ou d'exclure des balises et des valeurs de balise spécifiques. Plusieurs conditions peuvent être définies. La correspondance des noms de balise est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition : **Existe** - inclure les noms de balise spécifiés ; **Égal à** - inclure les noms de balise et les valeurs spécifiés (sensible à la casse) ; **Contient** - inclure les noms de balise dont les valeurs contiennent la chaîne saisie (correspondance de sous-chaîne, insensible à la casse) ; **N'existe pas** - exclure les noms de balise spécifiés ; **N'est pas égal à** - exclure les noms de balise et les valeurs spécifiés (sensible à la casse) ; **Ne contient pas** - exclure les noms de balise 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 de balise seront regroupées par la condition *Ou* ; **Ou** - il suffit qu'une seule condition soit remplie.

Remplacements

L'onglet **Remplacements** permet d'ajouter des remplacements personnalisés pour les jeux de données :



Les remplacements sont utiles lorsque plusieurs éléments sont sélectionnés pour un jeu de données à l'aide du caractère générique ∗ et que vous souhaitez modifier la façon 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 :

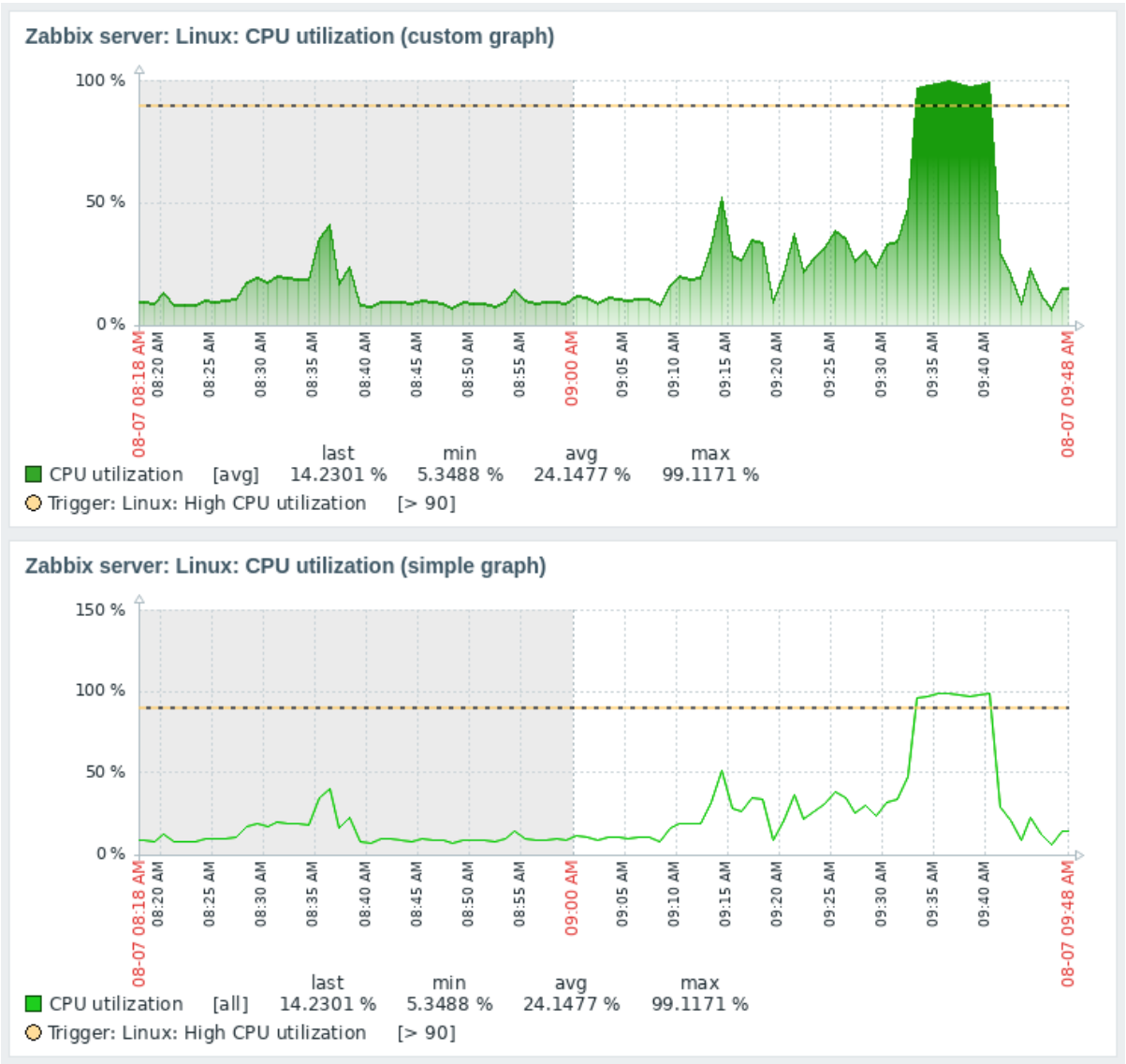
- Cliquez sur le bouton 

- Sélectionnez les hôtes et les éléments pour le remplacement. Vous pouvez également saisir des motifs d'hôte et d'élément. Des motifs génériques peuvent être utilisés (par exemple, ∗ renverra les résultats correspondant à zéro, un ou plusieurs caractères). Pour spécifier un motif 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 ex. item2, item3). Les paramètres de motif d'hôte et de motif d'élément sont obligatoires. Le paramètre permettant de spécifier des motifs d'hôte 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 la description des paramètres, voir l'onglet *Jeu de données* ci-dessus.

9 Graphique (classique)

Aperçu

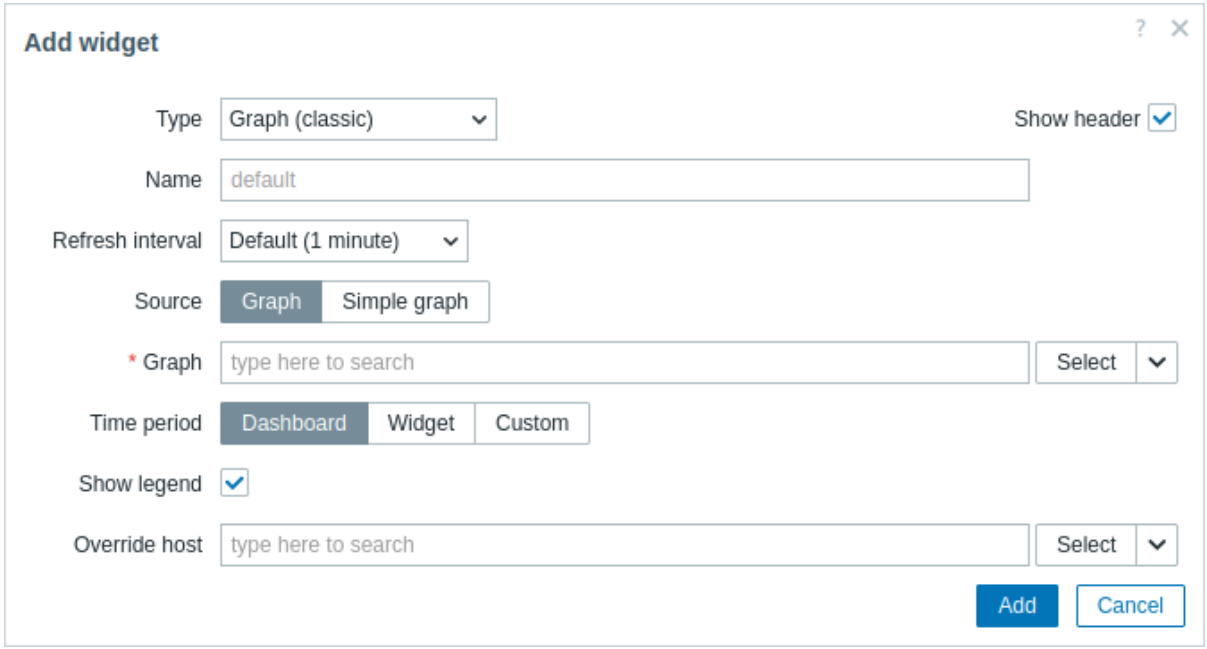
Le widget *Graph (classic)* affiche les données numériques des éléments sous forme d'image, sous la forme d'un graphique personnalisé ou d'un graphique simple. Il prend en charge la sélection de la période, les lignes de déclencheur et l'affichage facultatif de la légende.



Les informations affichées dans le widget *Graph (classic)* peuvent être téléchargées sous forme d'image PNG en sélectionnant l'option *Download image* dans le menu du widget.

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)

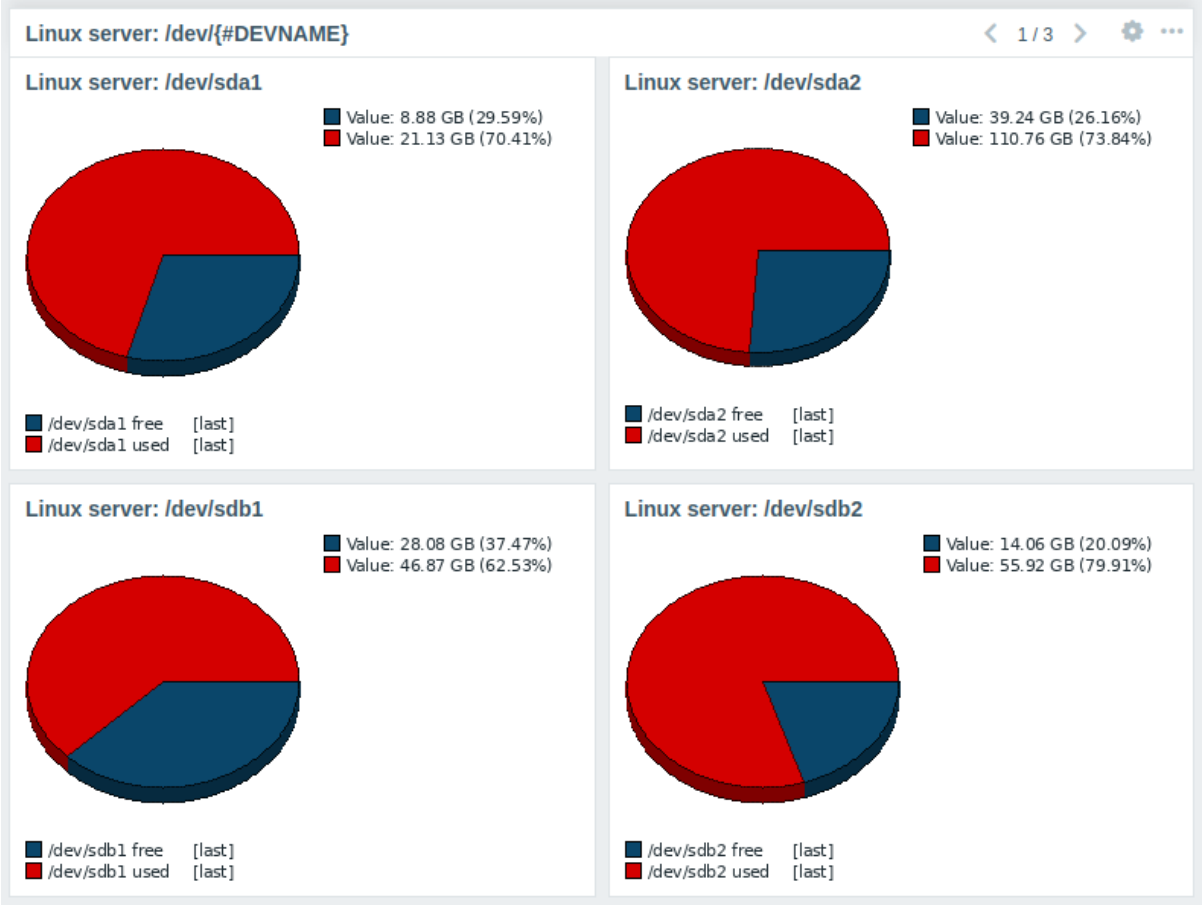
Si la hauteur du graphique dans le widget est insuffisante, la légende peut ne pas être affichée ou n'être affichée que partiellement. Les déclencheurs et les percentiles sont masqués en premier, suivis de la légende des éléments. Pour afficher la légende complète, augmentez la taille verticale du widget.

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.

10 Prototype de graphique

Aperçu

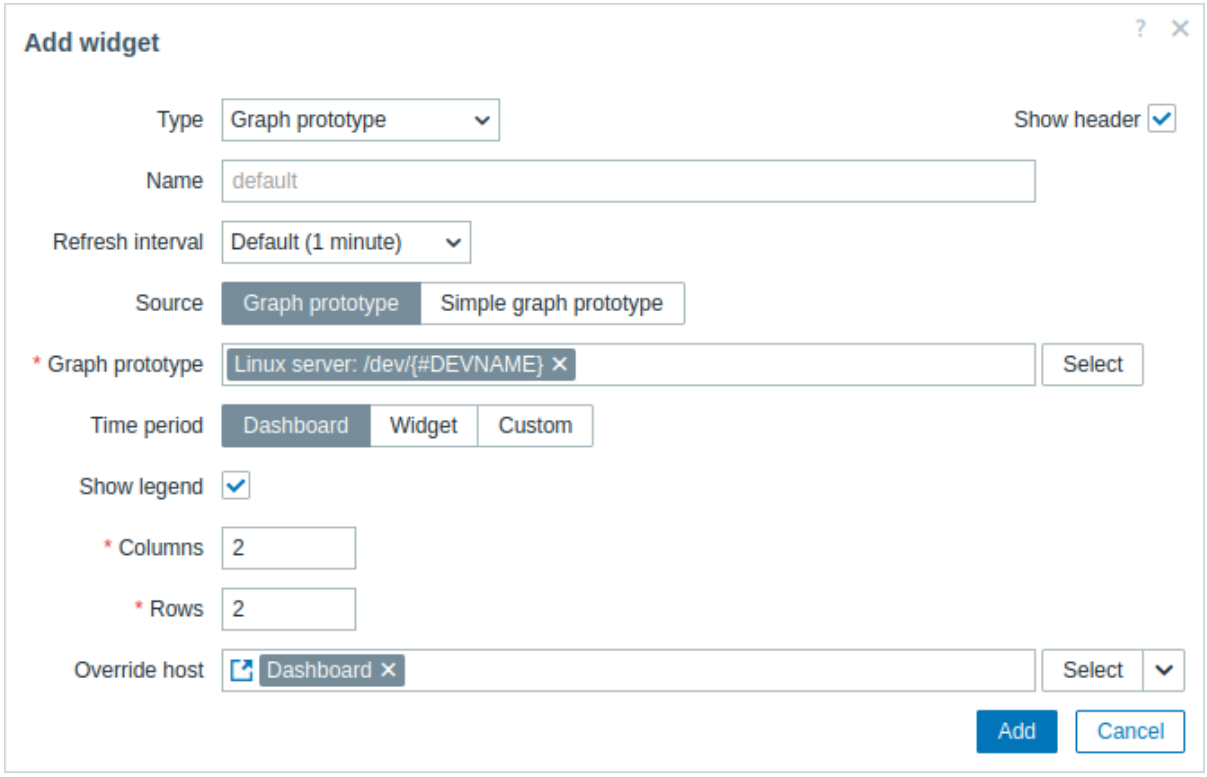
Le widget *Prototype de graphique* affiche une grille de graphiques personnalisés créés automatiquement, basée sur des prototypes de graphiques ou des prototypes d'éléments de règle de découverte de bas niveau.



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, sélectionnez *Prototype de graphique* 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 : **Prototype de graphique** ou **Prototype de graphique simple**.
Prototype de graphique	Sélectionnez un prototype de graphique pour afficher les graphiques découverts par le prototype de graphique. Ce paramètre est disponible si *Source* est défini sur « Prototype de graphique ».
Prototype d'élément	Sélectionnez un prototype d'élément pour afficher des graphiques simples pour les éléments découverts par le prototype d'élément. Ce paramètre est disponible si *Source* est défini sur « Prototype de graphique simple ».
Période	Définissez une période pour laquelle afficher les données dans les graphiques. 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 ».
Afficher la légende	Décochez cette case pour masquer la légende sur les graphiques (cochée par défaut).
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.
Colonnes	Saisissez le nombre de colonnes de graphiques à afficher dans un widget de prototype de graphique.
Lignes	Saisissez le nombre de lignes de graphiques à afficher dans un widget de prototype de graphique.

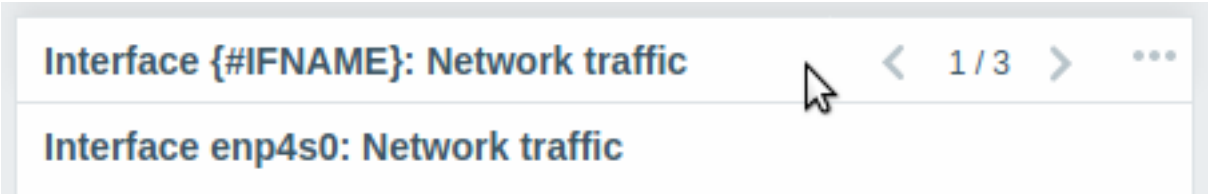
La légende du graphique se compose de trois sections :

- 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.

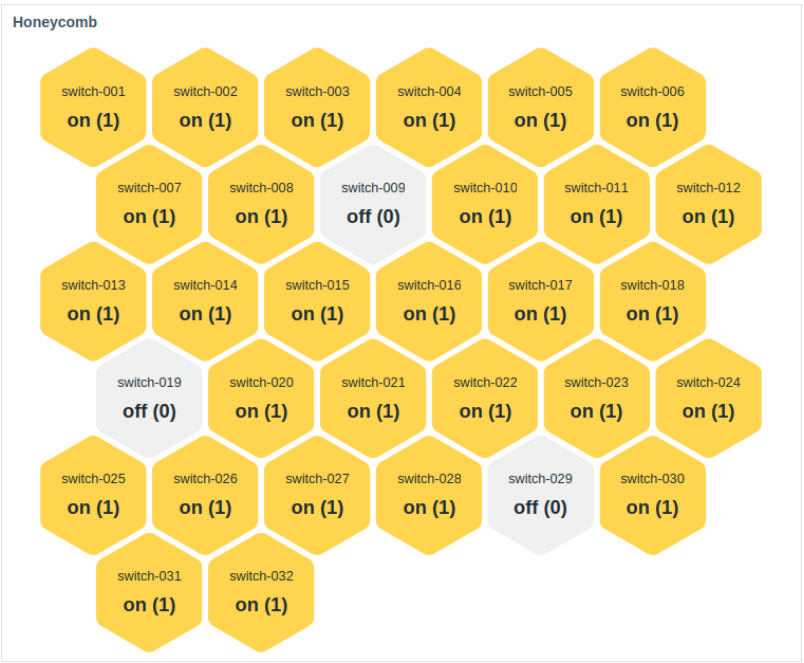
Bien que les paramètres *Colonnes* et *Lignes* permettent d'intégrer plus d'un graphique dans le widget, il peut tout de même y avoir plus de graphiques découverts que de colonnes/lignes dans le widget. Dans ce cas, la pagination devient disponible dans le widget, et un en-tête escamotable permet de passer d'une page à l'autre à l'aide des flèches gauche et droite :

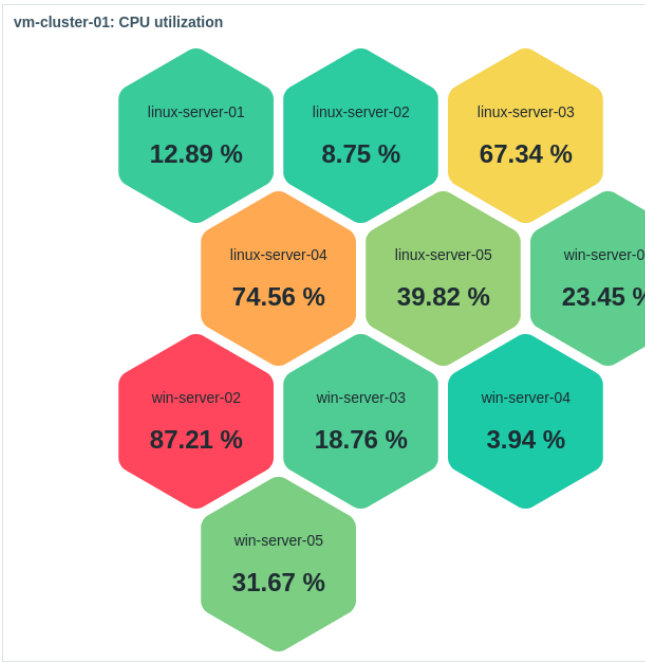


11 Honeycomb

Vue d'ensemble

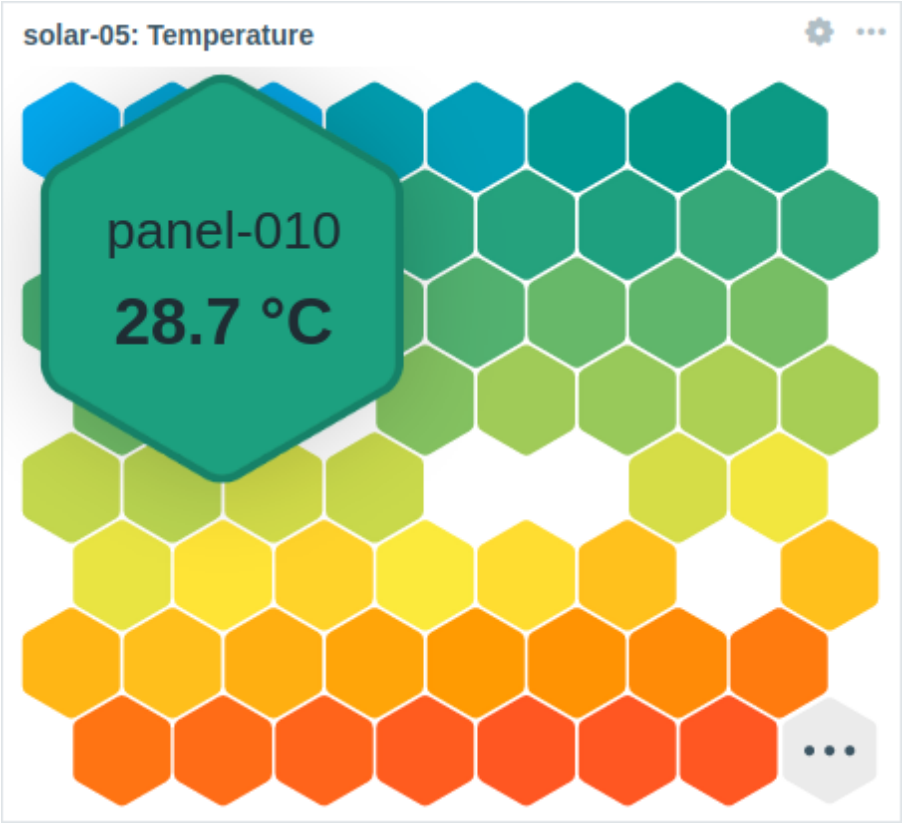
Le widget *Honeycomb* affiche les hôtes ou les éléments sous forme de cellules hexagonales dans une grille. Cette disposition permet de détecter facilement les hôtes/éléments en problème, de repérer des groupes de problèmes, de comparer des groupes d'un seul coup d'œil, et plus encore.





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 de cellules, leur taille et leur positionnement s'ajustant dynamiquement.

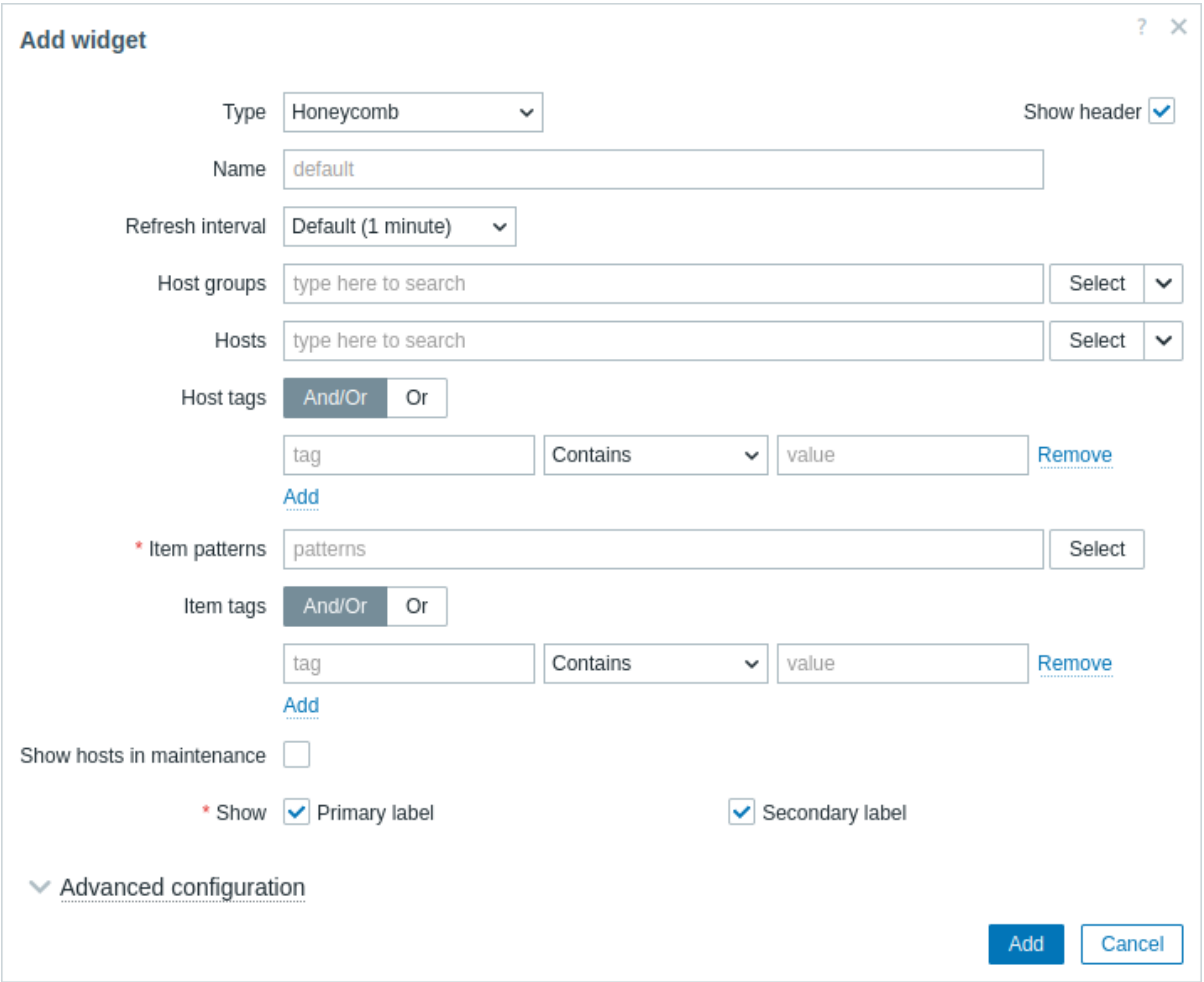
Au survol de la souris, la cellule ciblée s'agrandit pour améliorer la visibilité. Un clic sur une cellule met sa bordure en surbrillance jusqu'à ce qu'une autre cellule soit sélectionnée.



Les informations affichées dans le widget *Honeycomb* peuvent être téléchargées sous forme d'image PNG en sélectionnant l'option *Download image* dans le menu du widget.

Configuration

Pour configurer, sélectionnez *Honeycomb* comme type :



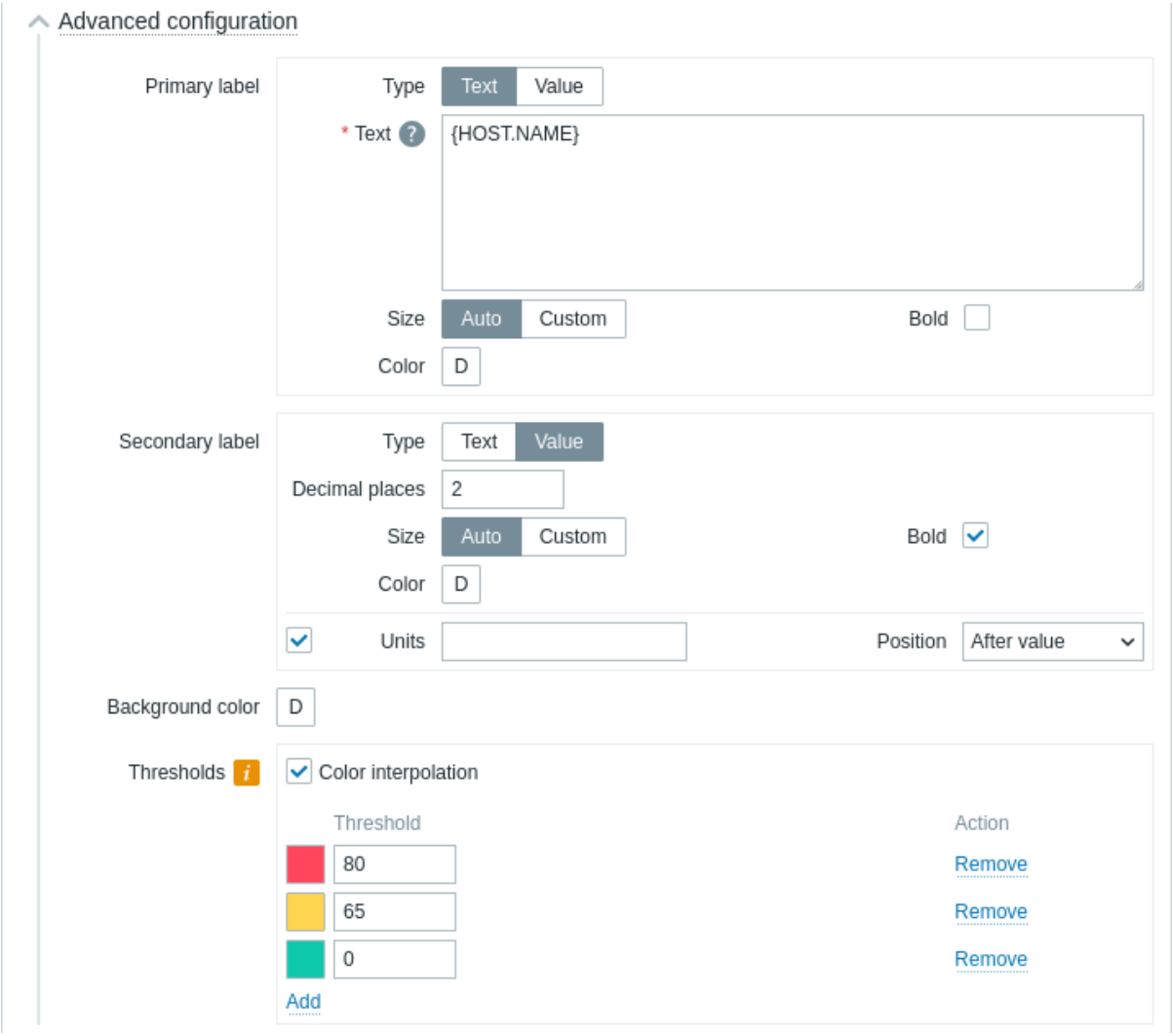
En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

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 des 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 *Item pattern* 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 des hôtes correspondants s'affiche. Si aucun hôte n'est sélectionné, le widget affichera tous les hôtes avec des éléments correspondant au *Item pattern* 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 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). 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 *Enter*. 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 des 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 respectif de la cellule honeycomb - libellé principal, libellé secondaire. Au moins un élément doit être sélectionné.
Advanced configuration	Cliquez sur le libellé *Advanced configuration* pour afficher les options de configuration avancée.

Configuration avancée

Des 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'abeilles.



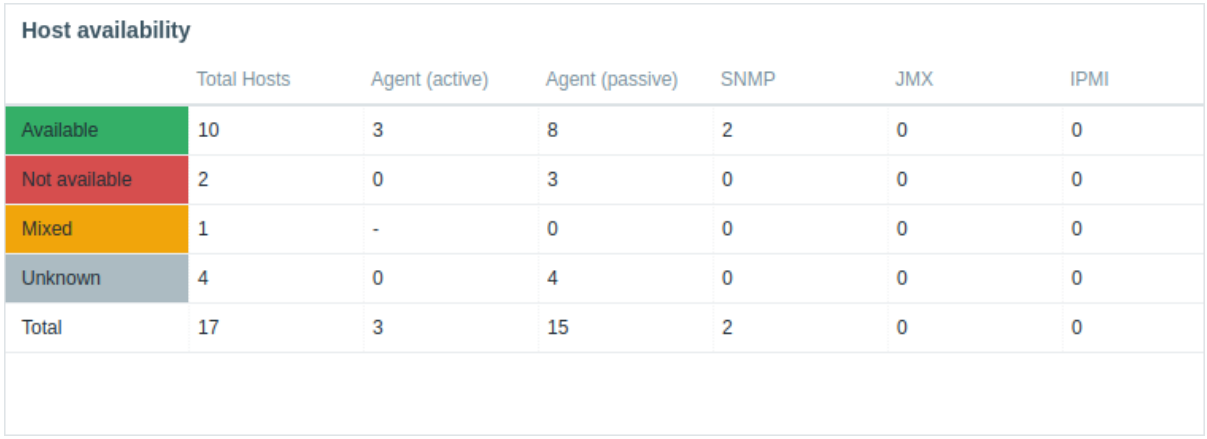
Libellé principal/secondaire	
Type	Sélectionnez le type de libellé : **Texte** - le libellé affichera le texte spécifié dans le paramètre *Texte* ; **Valeur** - le libellé affichera la valeur de l'élément avec le nombre de décimales spécifié dans le paramètre *Décimales*.
Texte	Saisissez le texte du libellé. 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'abeilles sont triées par ordre alphabétique selon le nom d'hôte, puis, pour chaque hôte, selon le nom de l'élément. Ce paramètre est disponible si *Type* est défini sur « Texte ».

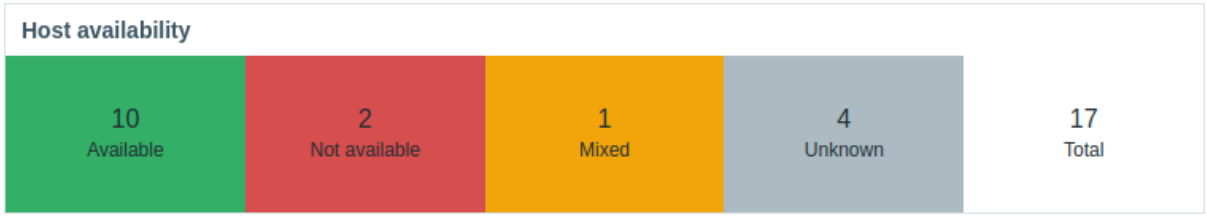
Décimales	Saisissez le nombre de décimales à afficher avec la valeur. Ce paramètre est disponible si *Type* est défini sur « Valeur » et n'affecte que les éléments qui renvoient des données numériques (float).
Taille	Sélectionnez la taille du libellé : **Auto** - utiliser une taille de libellé ajustée automatiquement ; **Personnalisée** - saisir une taille de libellé personnalisée (en pourcentage, par rapport à la taille de la cellule en nid d'abeilles). Notez que les libellés qui ne tiennent pas dans la taille de la cellule en nid d'abeilles sont tronqués.
Gras	Cochez la case pour afficher les unités de l'élément en gras.
Couleur	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 *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 « Texte ».
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 « Texte ».
Couleur d'arrière-plan	
Couleur d'arrière-plan	Sélectionnez la couleur d'arrière-plan des cellules en nid d'abeilles 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 *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'abeilles. 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.

12 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.



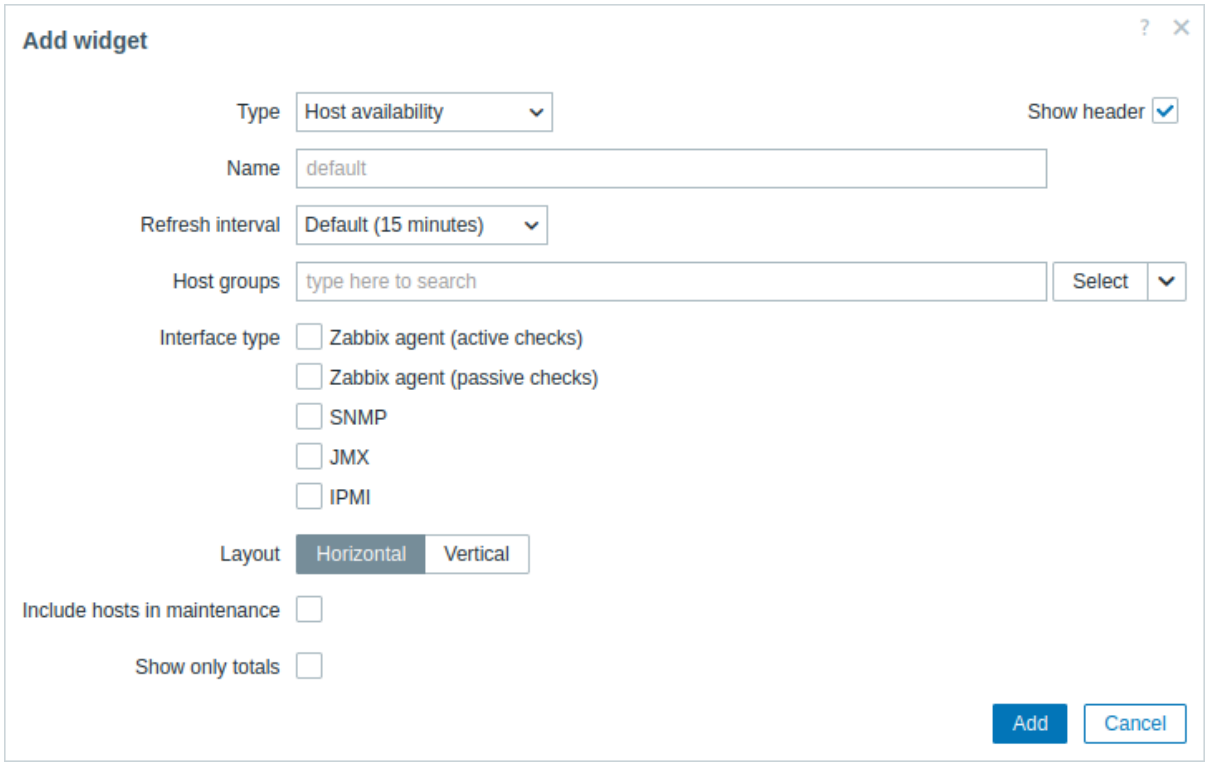


La disponibilité des hôtes est comptabilisée comme suit :

- *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 :



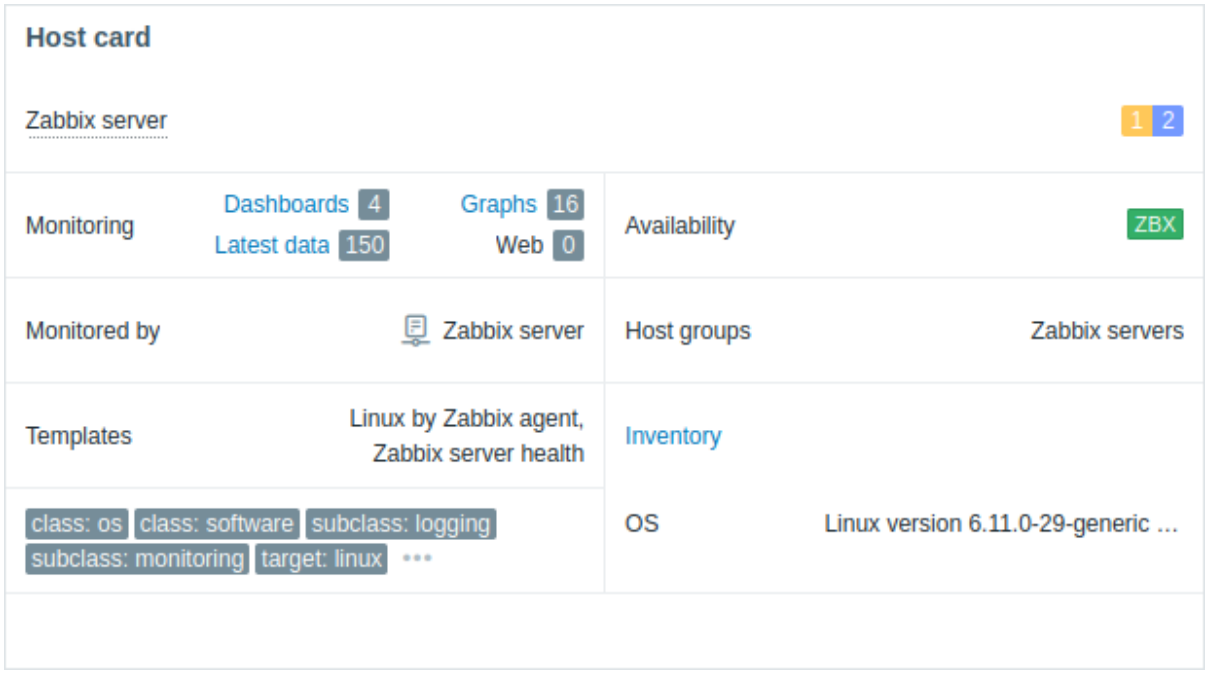
En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

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.

13 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.).

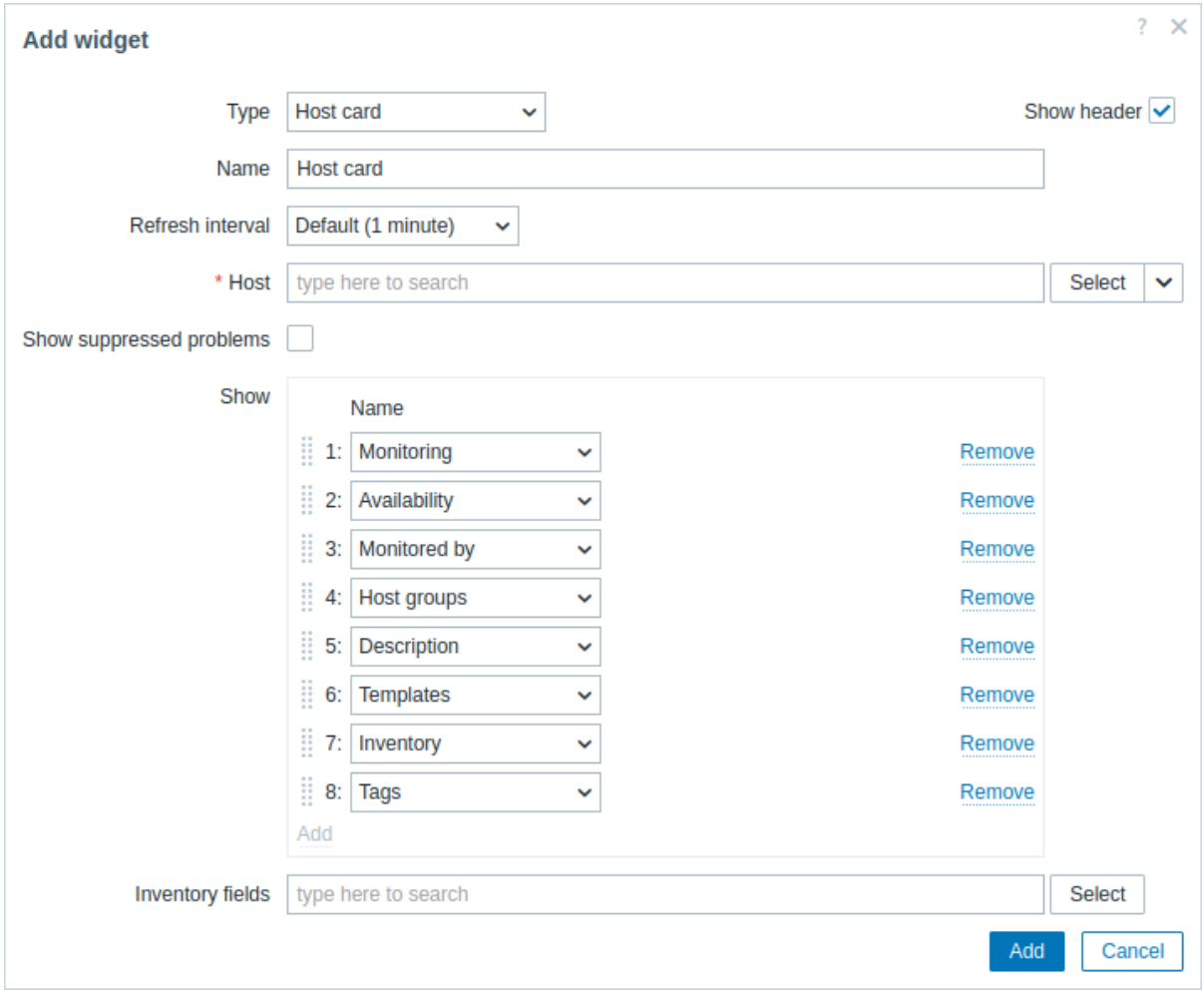


L’hôte peut être défini directement dans la configuration du widget ou sélectionné à partir d’un widget compatible ou du sélecteur d’hôte sur le tableau de bord.

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 :

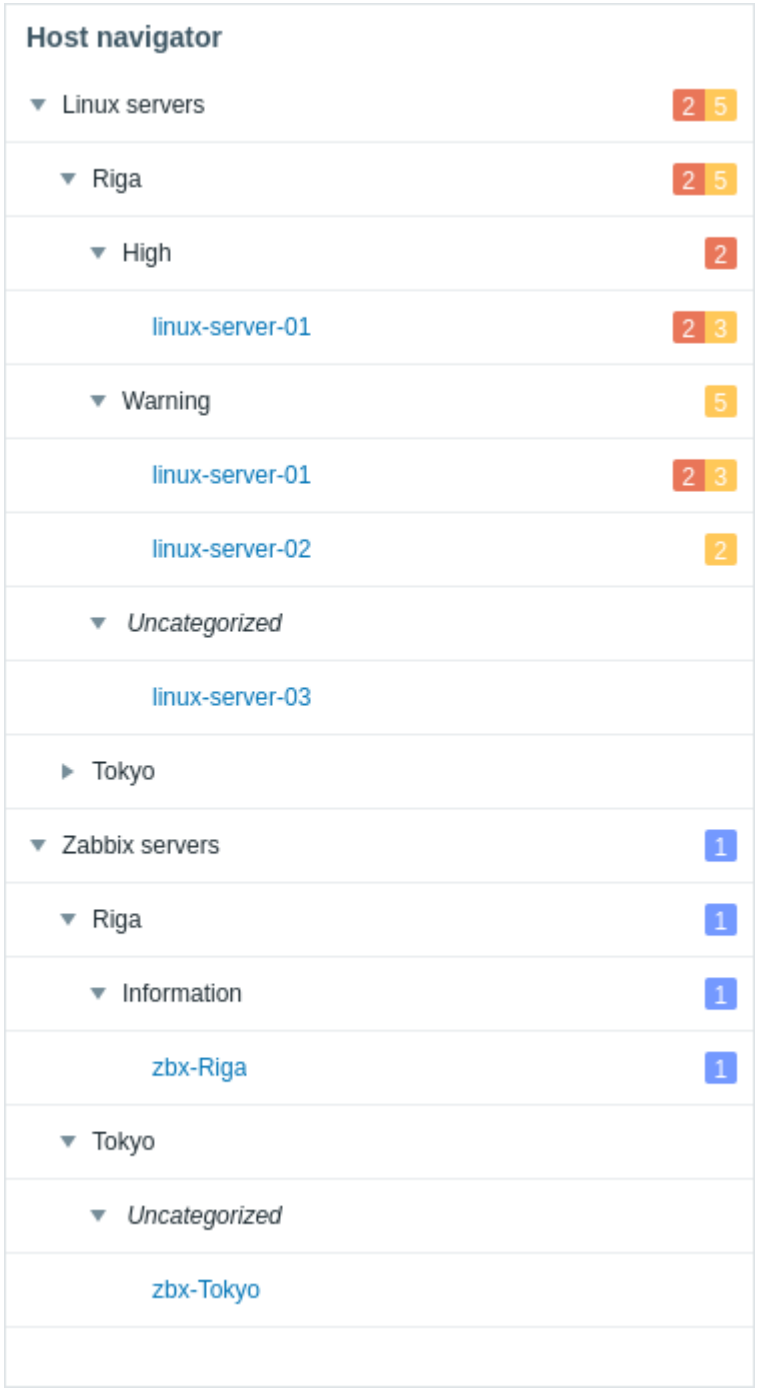
Hôte	Sélectionnez l'hôte. Vous pouvez également sélectionner un widget compatible ou le tableau de bord comme source de données pour les hôtes. Des suggestions correspondantes apparaissent au fur et à mesure de la saisie. La section supérieure du widget affiche toujours le **nom de l'hôte** (un clic 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*.

14 Navigateur d'hôtes

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.

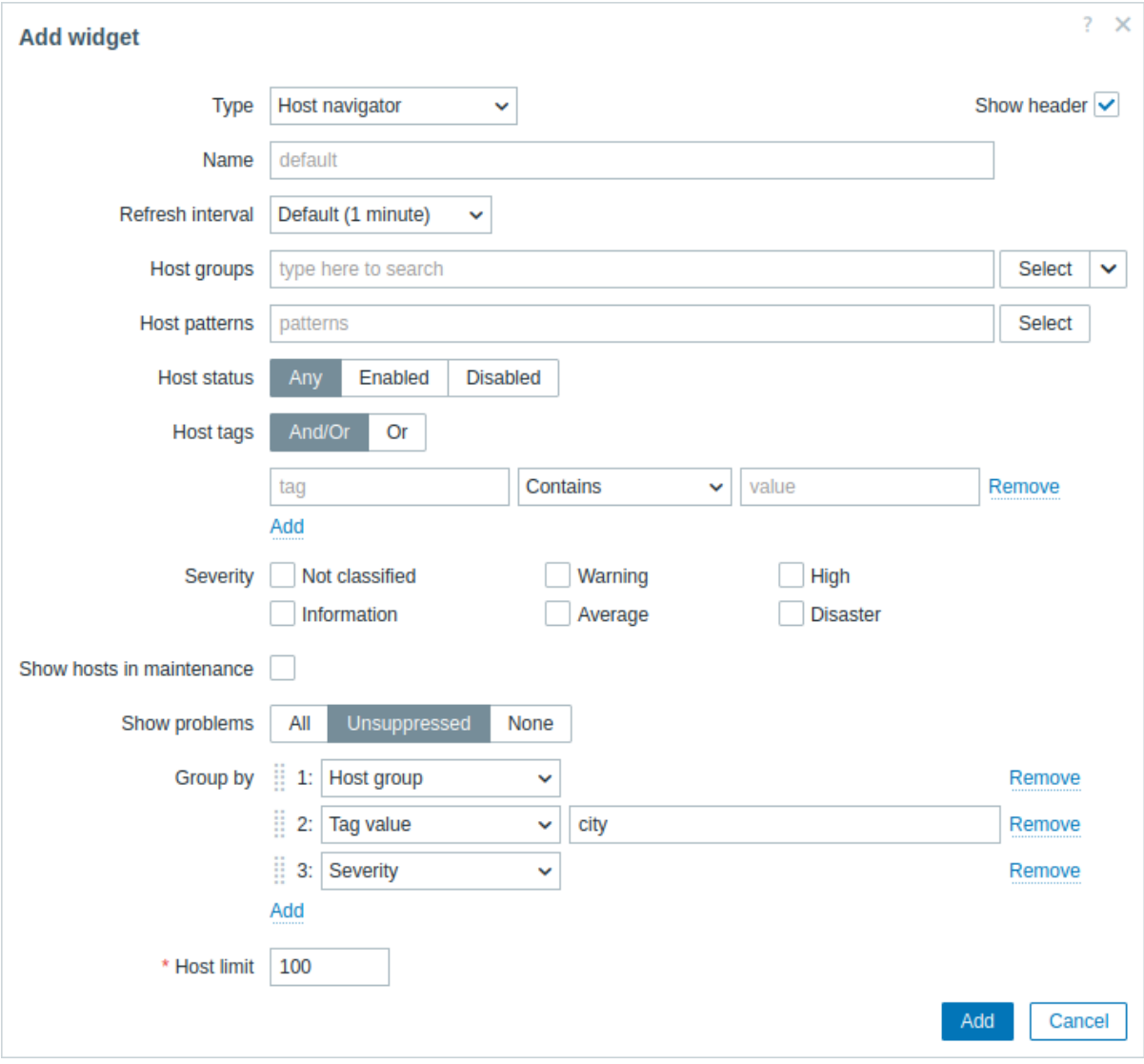
Pour les groupes, les problèmes et les hôtes en maintenance, des détails supplémentaires sont accessibles via des infobulles au survol de la souris.

Le widget est particulièrement utile pour contrôler ce que les autres widgets affichent en fonction de l'hôte sélectionné.



Configuration

Pour configurer, sélectionnez *Host navigator* comme type :



En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

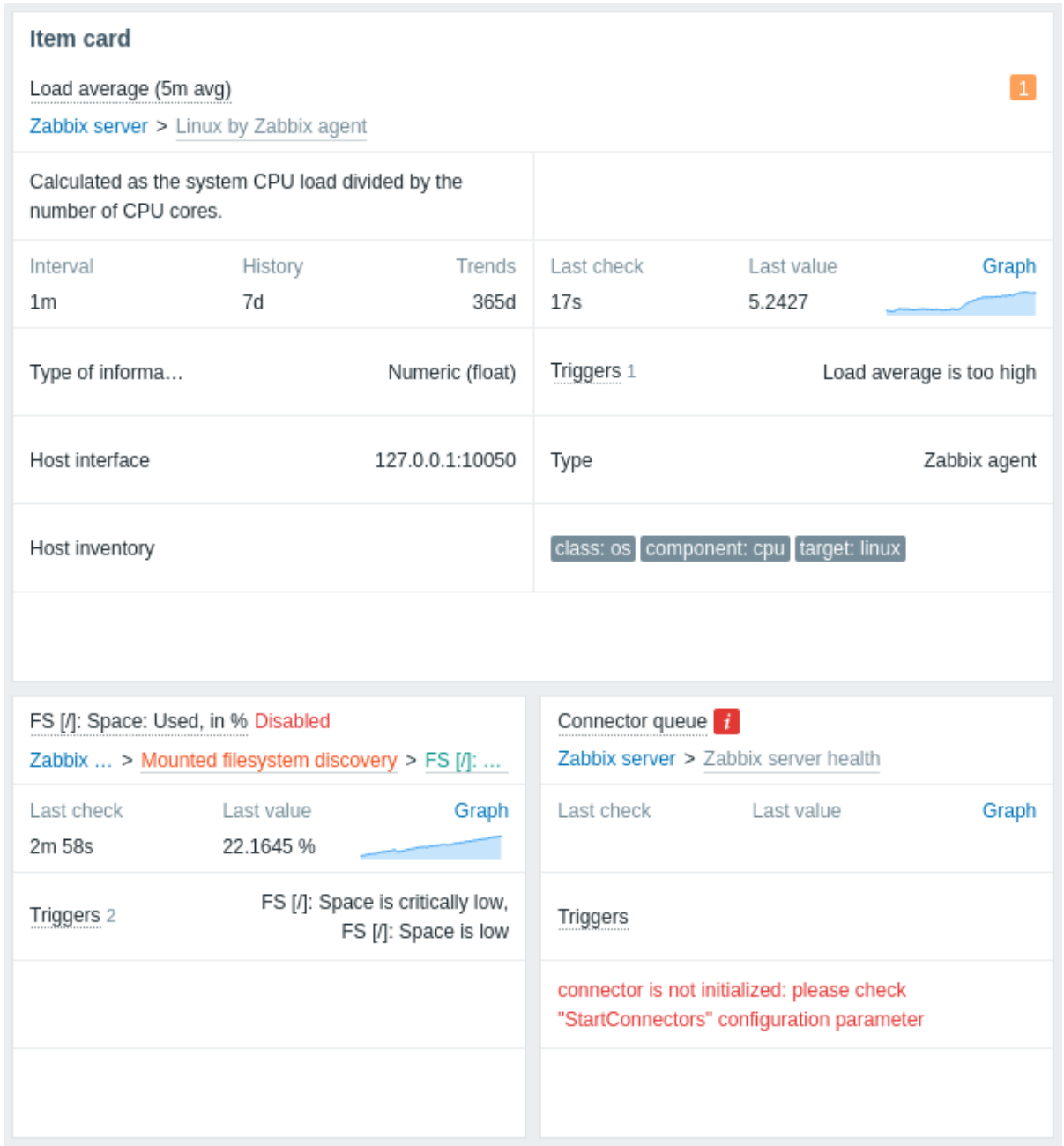
Host groups	Sélectionnez des groupes d'hôtes. Vous pouvez également sélectionner un widget compatible comme source de données pour les groupes d'hôtes. Ce champ prend en charge l'autocomplétion ; ainsi, lorsque vous commencez à saisir le nom d'un groupe, une liste déroulante de groupes correspondants est proposée. 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 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'hôte ou sélectionnez des hôtes existants comme motifs d'hôte. Les hôtes correspondant aux motifs spécifiés seront affichés dans le navigateur d'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 est proposée. Si aucun hôte n'est sélectionné, le widget affichera tous les hôtes. Des motifs génériques peuvent être utilisés pour la sélection (par exemple, ∗ renverra les hôtes correspondant à zéro, un ou plusieurs caractères ; `Zabbix*` renverra 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, une liste déroulante affiche les hôtes correspondants, limitée à ceux 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 selon leur état (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 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 : **And/Or** - toutes les conditions doivent être remplies, les conditions ayant le même nom de tag étant 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.
Severity	Marquez les sévérités de problème pour filtrer les hôtes avec problèmes à afficher dans le widget. Si aucune sévérité n'est marquée, tous les hôtes avec tous les problèmes seront affichés.
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 libellé *Show data in maintenance* lors de la configuration du widget sur un tableau de bord de modèle.
Show problems	

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 la sévérité de leurs problèmes. Si *Show problems* est configuré pour afficher les problèmes, ils sont affichés comme suit : - pour chaque groupe de sévérité, seul le nombre de problèmes correspondant est affiché ; - pour chaque hôte, tous ses nombres de problèmes sont affichés. Notez que les hôtes seront regroupés uniquement selon les sévérités marquées dans le paramètre *Severity* ; si aucune sévérité n'est marquée, tous les hôtes seront regroupés selon toutes les sévérités. 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, si vous spécifiez plusieurs noms de tag (1 : color, 2 : city), les hôtes seront regroupés par couleur (red, blue, 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 et que l'hôte appartient à plusieurs groupes d'hôtes). En cliquant sur ces hôtes, vous les sélectionnez et les mettez en surbrillance 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 ils doivent tous être uniques. Si aucun attribut de regroupement n'est spécifié, les hôtes ne seront pas regroupés.
Host limit	Saisissez le nombre maximal d'hôtes à afficher. Les valeurs possibles vont de 1 à 9999. Lorsque davantage d'hôtes sont disponibles à l'affichage que 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 tag (plus de 200), seules les 100 premières valeurs de tag 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.

15 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.).

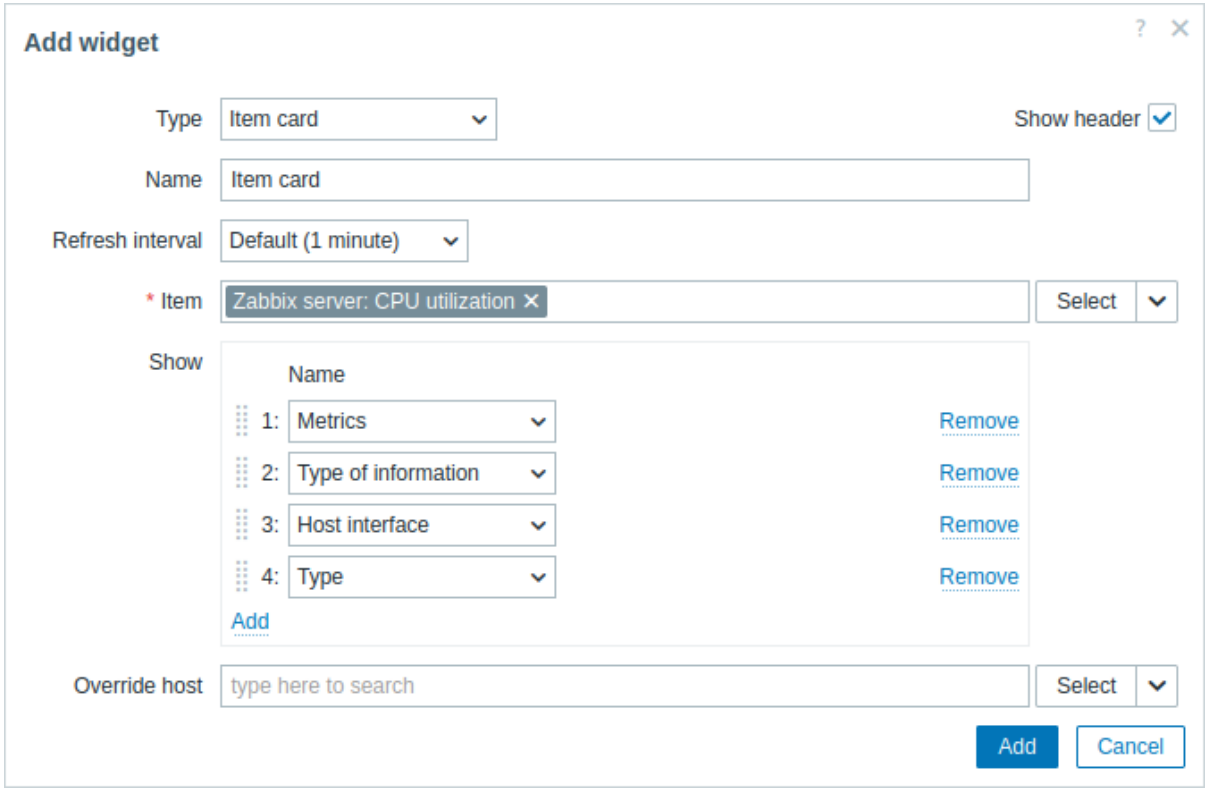


L'élément peut être défini directement dans la configuration du widget ou sélectionné à partir d'un widget compatible sur le tableau de bord.

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 :



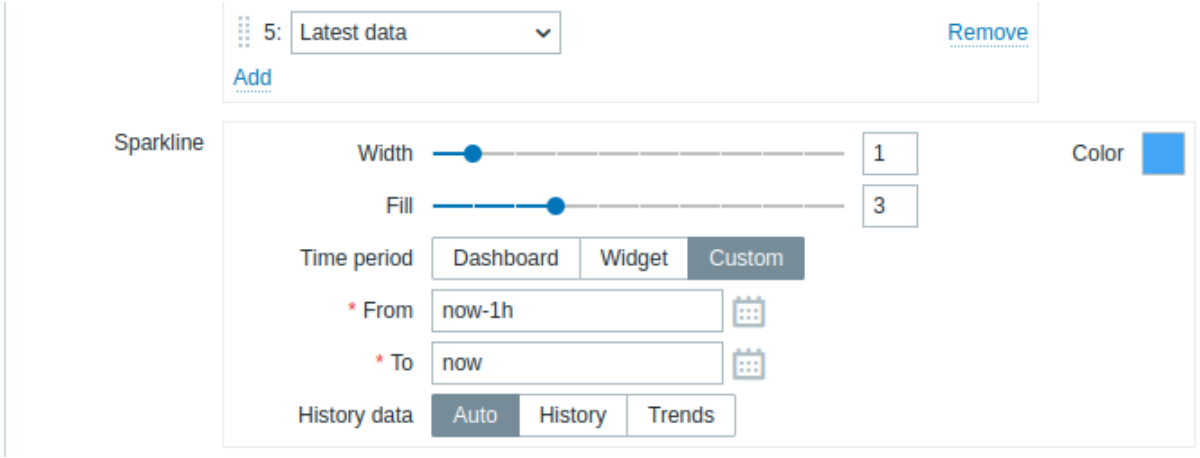
En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

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.



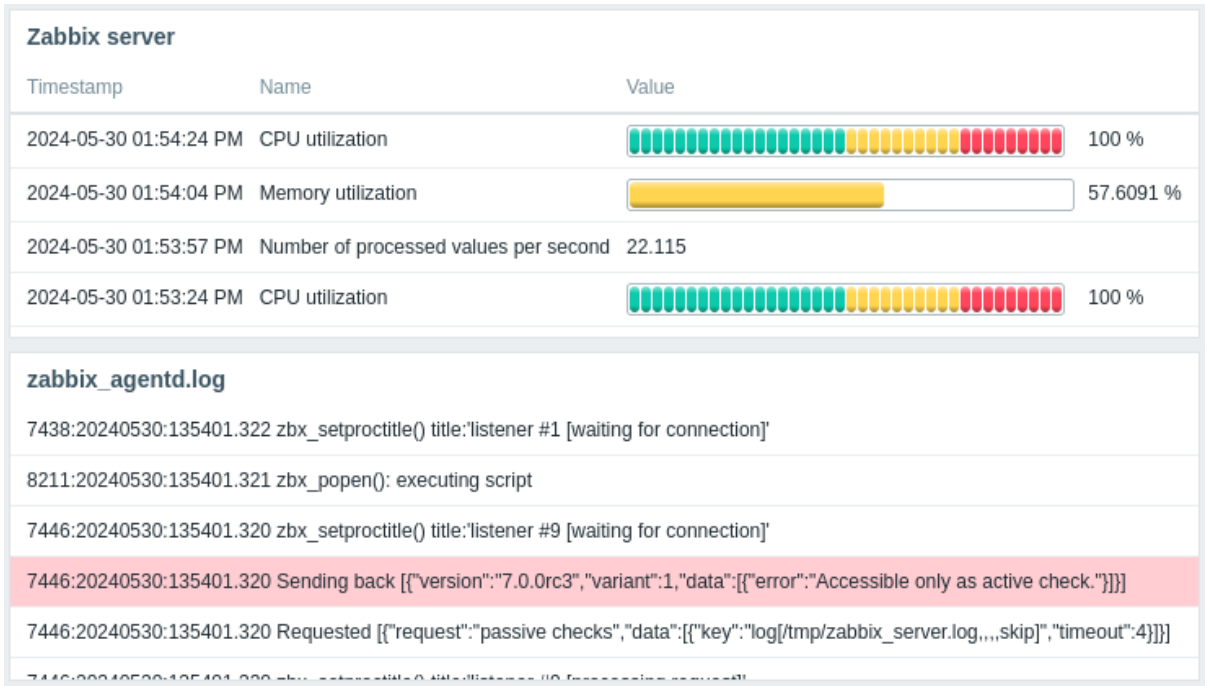
Largeur	Définissez l'épaisseur de la ligne du graphique à l'aide du curseur ou en saisissant manuellement une valeur comprise entre 0 et 10.
Couleur	Sélectionnez la couleur de la ligne et du remplissage.
Remplissage	Définissez le niveau de transparence de la couleur de remplissage à l'aide du curseur ou en saisissant manuellement une valeur comprise entre 0 et 10.

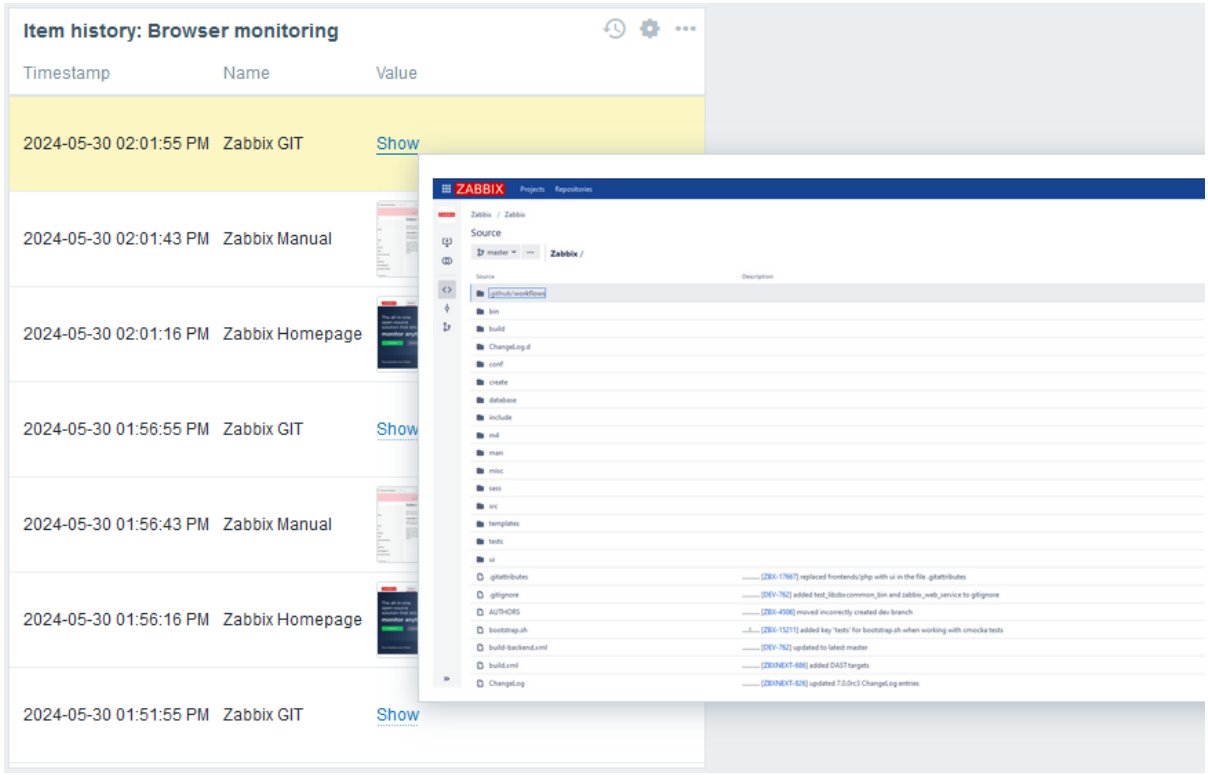
Période	Sélectionnez la source de données pour la période : **Tableau de bord** - utiliser le sélecteur de période du tableau de bord ; **Widget** - utiliser un widget compatible (défini dans le paramètre *Widget*) ; **Personnalisée** - utiliser une période personnalisée définie dans les paramètres *De* 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.

16 Historique des éléments

Aperçu

Le widget *Historique des éléments* 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).

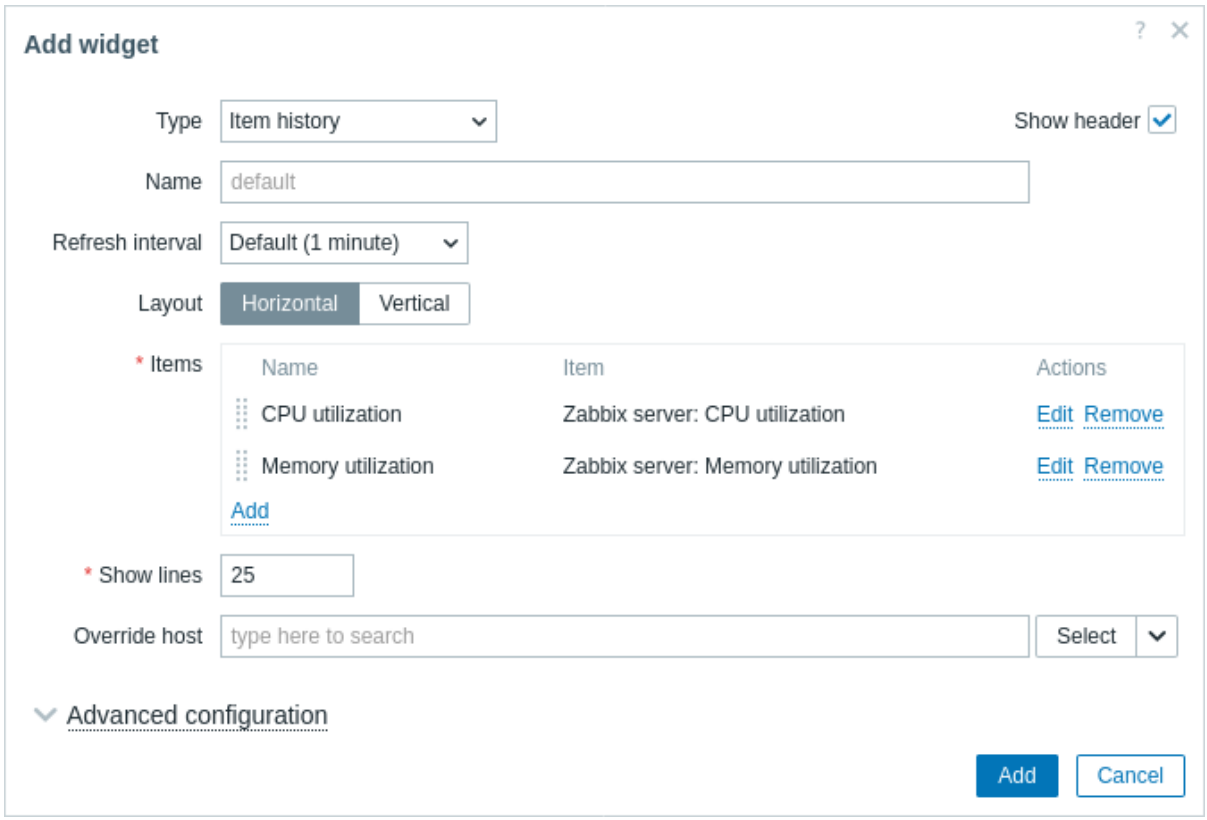




Jusqu'à 1000 enregistrements peuvent être affichés.

Configuration

Pour configurer, sélectionnez *Historique des éléments* comme type :



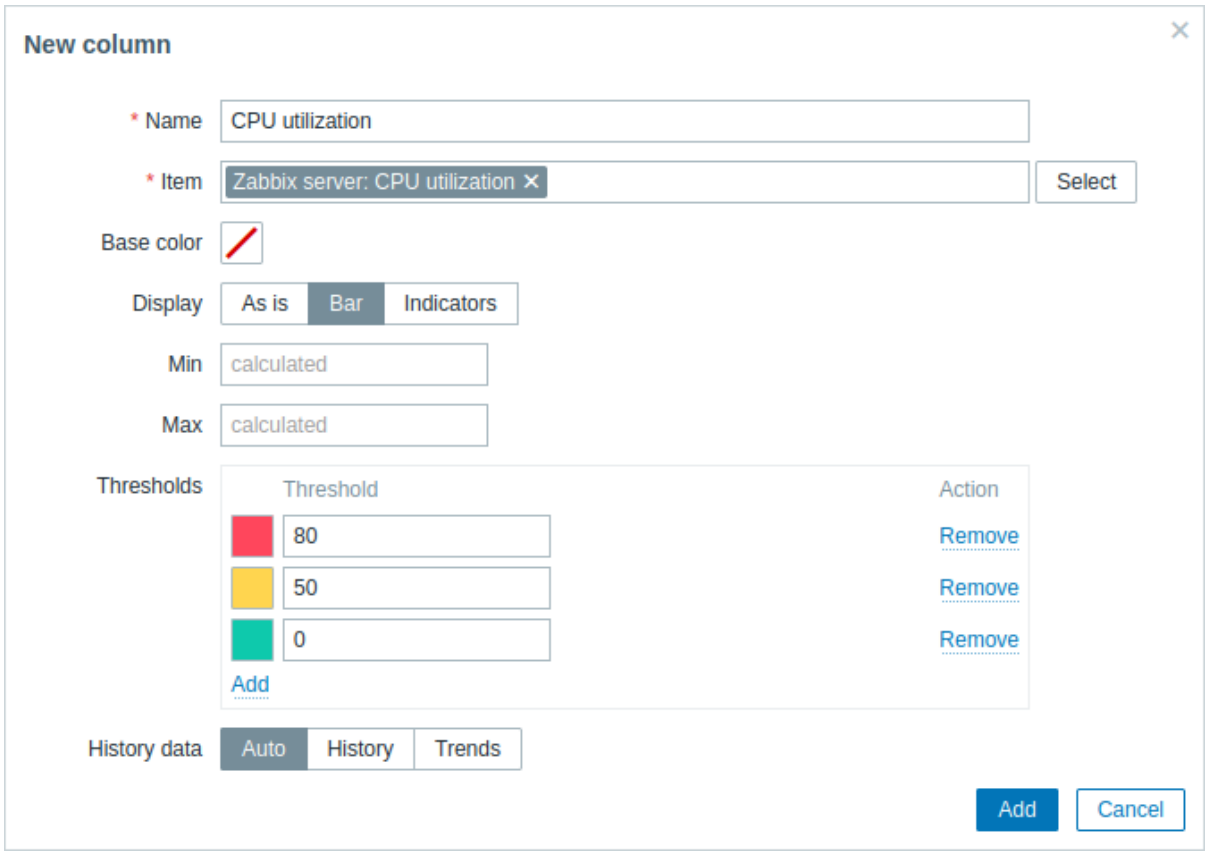
En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

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 :

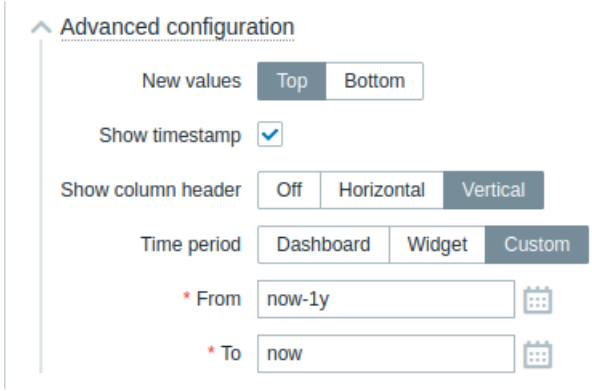
Mise en évidence	Cliquez sur *Ajouter* pour ajouter une mise en évidence, sélectionnez une couleur de mise en évidence dans le sélecteur de couleurs et spécifiez une expression régulière. La couleur sélectionnée sera utilisée comme couleur d'arrière-plan pour les valeurs d'élément dont le texte correspond à l'expression régulière spécifiée.
Affichage	Sélectionnez comment la valeur de l'élément doit être affichée : **Tel quel** - exactement comme reçue. Les sauts de ligne sont conservés si la valeur contient des retours à la ligne. Le renvoi à la ligne est appliqué si *Disposition* est défini sur « Vertical » et que plus d'une colonne est configurée ; **HTML** - comme texte mis en forme en HTML ; **Ligne unique** - sur une seule ligne, tronquée à une longueur spécifiée (1 à 500 caractères). Le survol ou le clic sur la valeur tronquée ouvre une fenêtre contextuelle contenant la valeur complète.
Utiliser une police à chasse fixe	Cochez cette case pour afficher la valeur de l'élément avec une police à chasse fixe (décochée 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* :

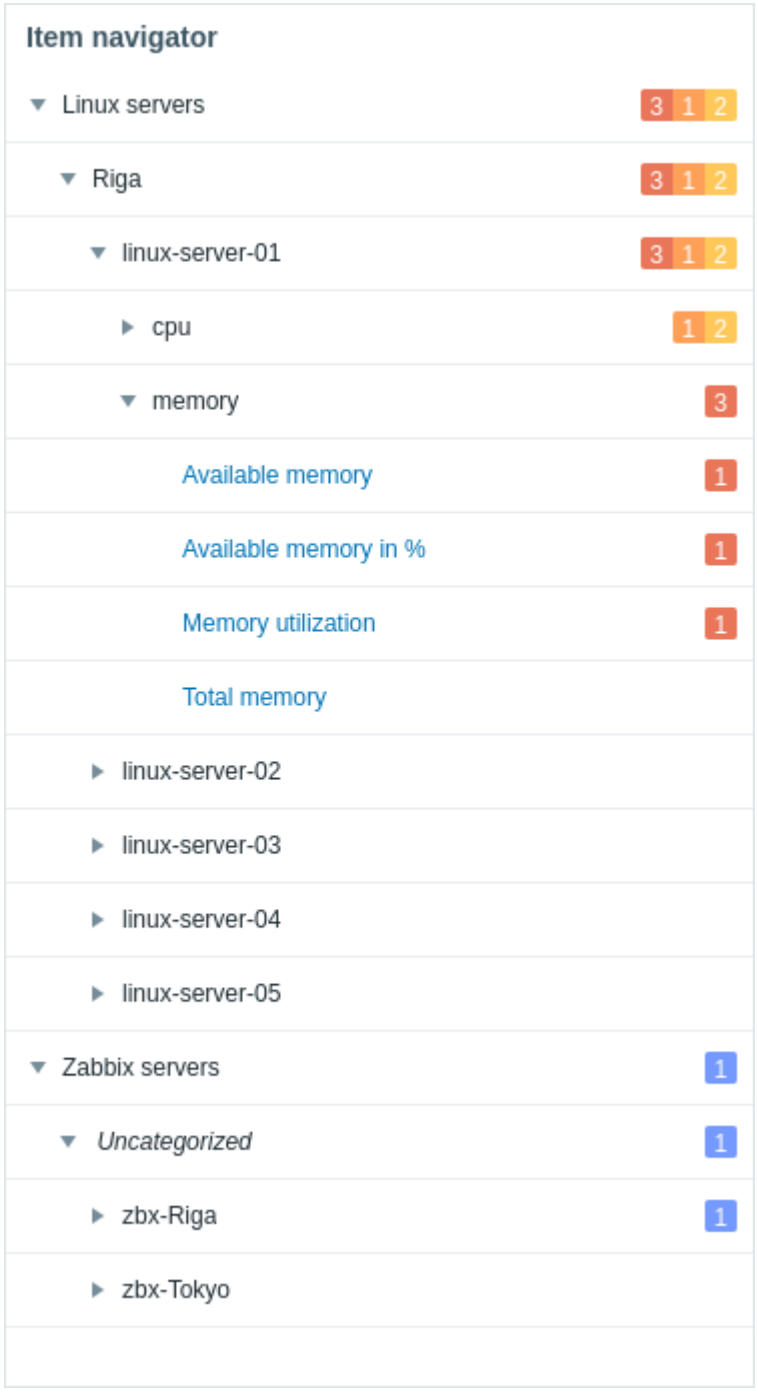


Nouvelles valeurs	Sélectionnez où les nouvelles valeurs d'élément doivent être ajoutées : **En haut** - en haut des colonnes ; **En bas** - en bas des colonnes.
Afficher l'horodatage	Cochez cette case pour afficher la colonne d'horodatage (décochée par défaut).
Afficher l'en-tête de colonne	Sélectionnez 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".

17 Navigateur d'éléments

Aperçu

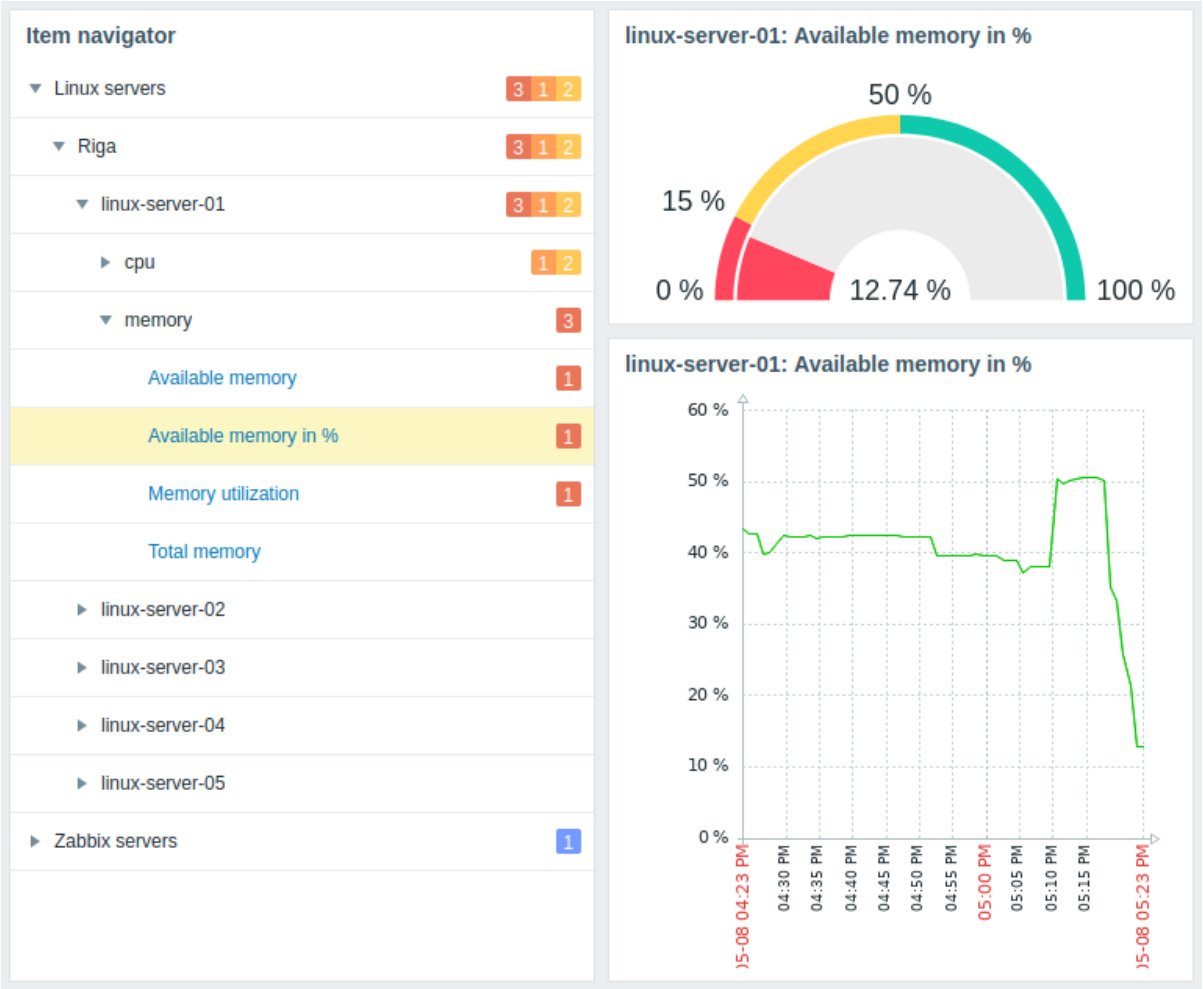
Le widget *Navigateur d'éléments* affiche une liste d'éléments selon diverses options de filtrage et de regroupement.



Les groupes selon lesquels les éléments sont organisés peuvent être développés ou réduits.

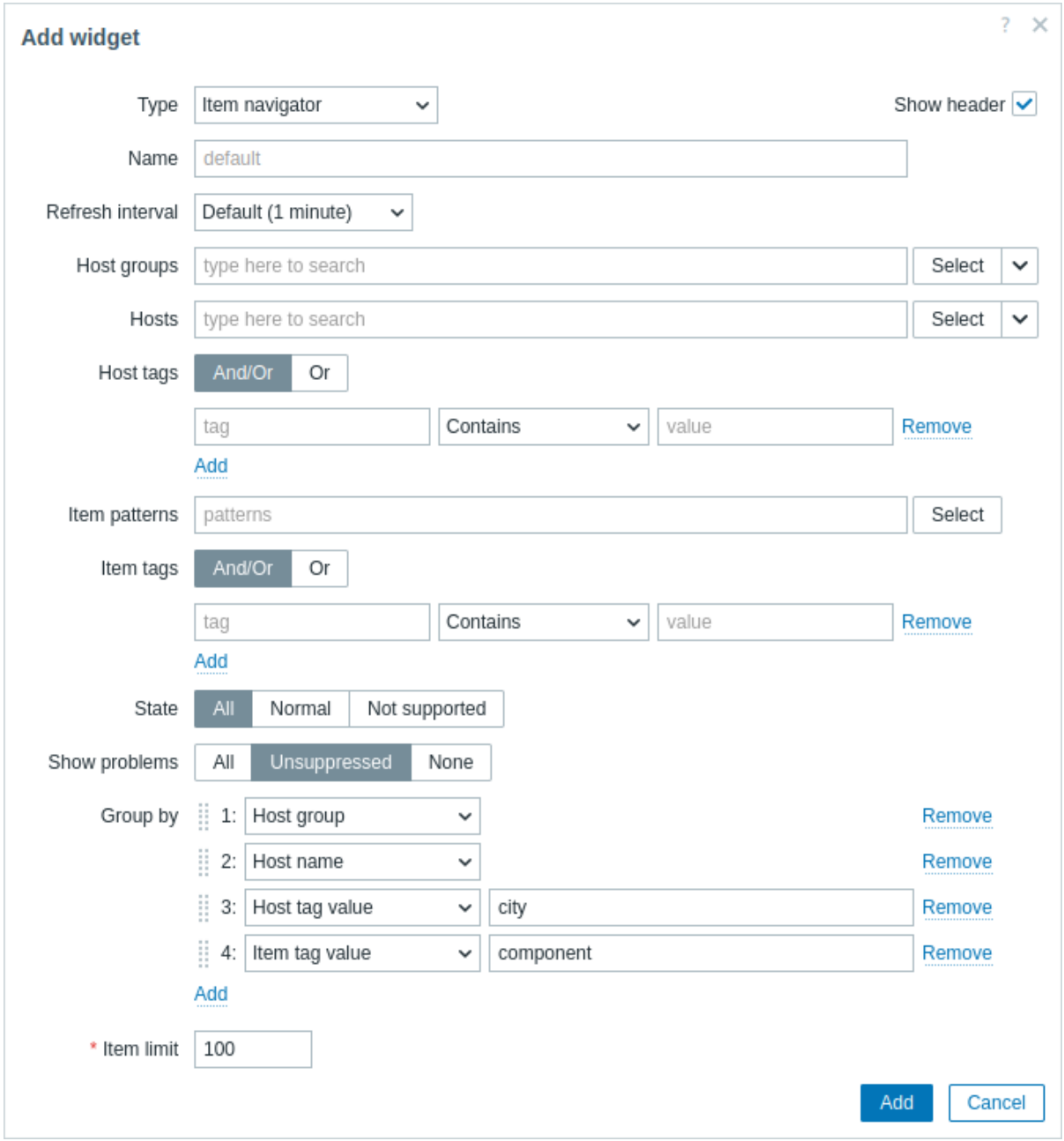
Pour les groupes et les problèmes, des détails supplémentaires sont accessibles via des infobulles au survol de la souris.

Le widget est particulièrement utile pour contrôler ce que les autres widgets affichent en fonction de l'élément sélectionné.



Configuration

Pour configurer, sélectionnez *Navigateur d'éléments* 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 de groupes correspondants est proposée. 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 les éléments appartenant à 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.
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 prend en charge l'autocomplétion ; ainsi, lorsque vous commencez à saisir le nom d'un hôte, une liste déroulante d'hôtes correspondants est proposée. Si aucun hôte n'est sélectionné, le widget affichera 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.

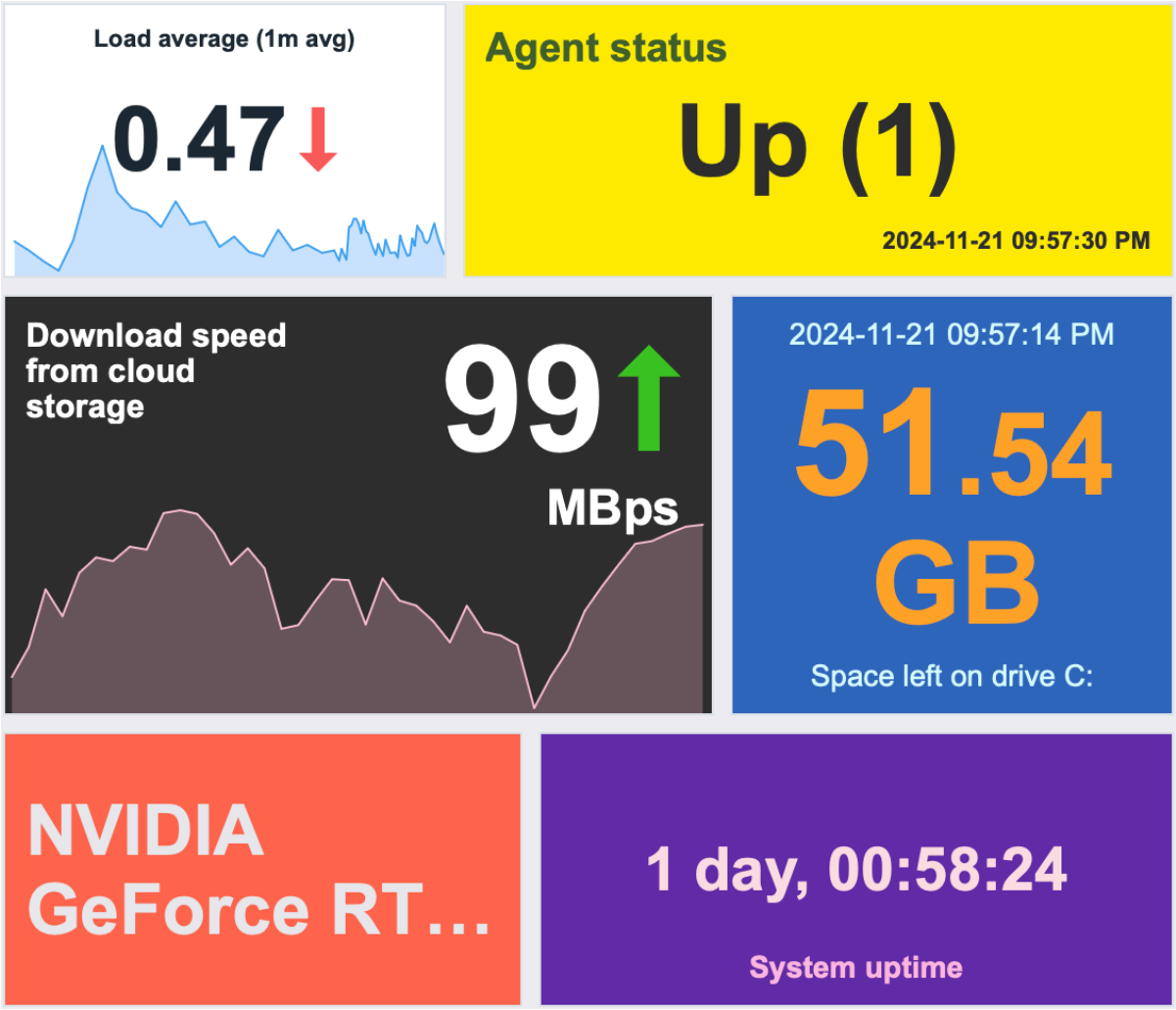
Tags d'hôte	Spécifiez des tags pour filtrer les éléments affichés dans le widget. Il est possible d'inclure comme d'exclure des tags et des valeurs de tag spécifiques. Plusieurs conditions peuvent être définies. La correspondance du nom du tag est toujours sensible à la casse. Plusieurs opérateurs sont 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.
Modèles d'élément	Saisissez des modèles d'élément ou sélectionnez des éléments existants comme modèles d'élément. Les éléments correspondant aux modèles spécifiés seront affichés dans le navigateur d'éléments. 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 *Hôtes* sélectionnés ou aux hôtes des *Groupes d'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 ex. 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.
Tags d'élément	Spécifiez des tags pour filtrer les éléments 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 du nom du 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.
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 sévérité du problème, qui peut être personnalisée dans *Administration* > *Général* > *Options d'affichage des déclencheurs*. Les sévérités des problèmes peuvent être modifiées 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 par groupes d'hôtes de leurs hôtes ; **Nom d'hôte** - regrouper les éléments par leurs hôtes ; **Valeur du tag d'hôte** - saisissez un nom de tag pour regrouper les éléments par les valeurs de ce tag d'hôte (par exemple, saisissez « city » pour regrouper les éléments par les valeurs « Riga », « Tokyo », etc.) ; **Valeur du tag d'élément** - saisissez un nom de tag pour regrouper les éléments par les valeurs de ce tag d'élément (par exemple, saisissez « component » pour regrouper les éléments par 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 de tag d'hôte (1 : color, 2 : city) entraînera un regroupement des 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 et que l'hôte de l'élément appartient à plusieurs groupes d'hôtes). En cliquant sur de tels éléments, vous les sélectionnez et les mettez en surbrillance dans tous les groupes. Les éléments qui ne correspondent pas aux attributs de regroupement configurés sont affichés dans le groupe *Non classé*. Si *Afficher les problèmes* est configuré pour afficher les problèmes, ils 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 ils doivent tous 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 davantage d'éléments sont disponibles à l'affichage que la limite définie, un message correspondant est affiché sous les éléments affichés (par exemple, « 100 éléments affichés sur 100+ »). 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 par leurs hôtes (contenant chacun 200 éléments), seul le premier hôte avec ses 100 éléments sera affiché dans le widget. Ce paramètre n'est pas affecté par le paramètre *Limit for search and filter results* dans *Administration → Général → GUI*.

18 Valeur d'élément

Vue d'ensemble

Le widget *Valeur de l'élément* affiche la valeur d'un seul élément numérique ou de type chaîne. Il est utile pour surveiller des métriques importantes, visualiser des seuils et détecter des changements soudains dans les données.



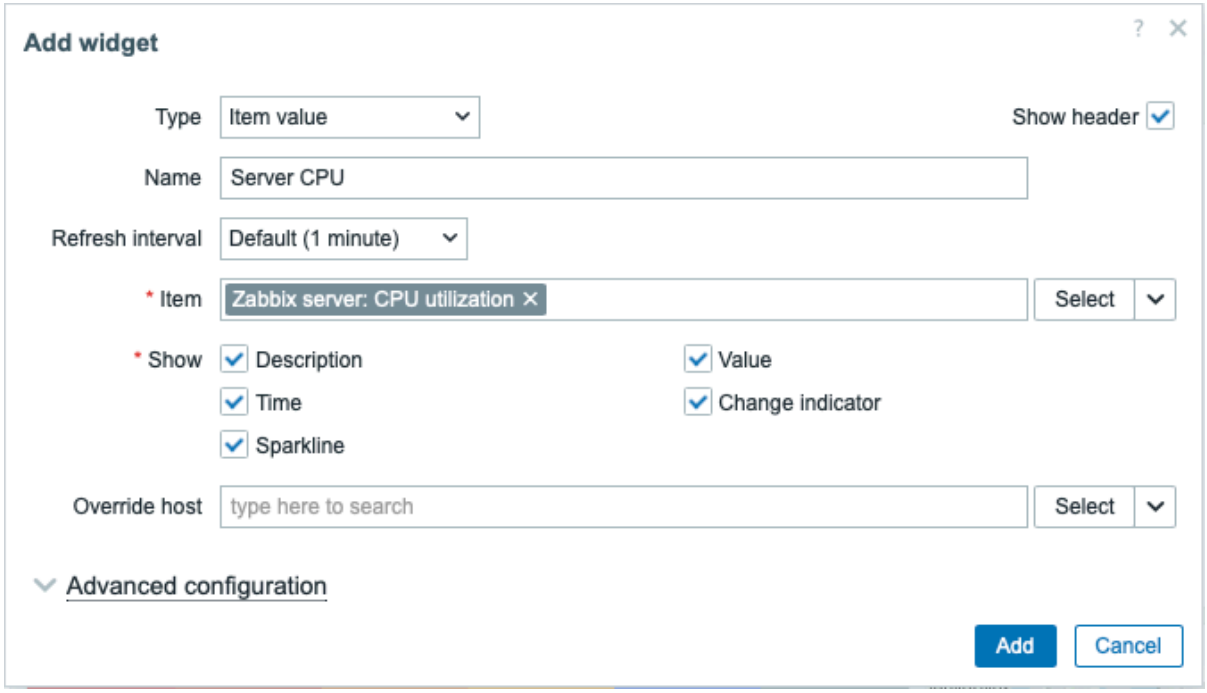
Vous pouvez configurer le widget pour afficher les éléments suivants :

- Horodatage de la métrique (2024-11-21 09:57:30)
- Description de l'élément (Load average (1m), Agent status)
- Valeur de l'élément, indicateur de changement et unités (0.47 ↓, 99 ↑ Mbps)
- Couleur d'arrière-plan
- Graphique sparkline pour les valeurs de la période spécifiée

Un clic sur le widget ouvre un graphique ad hoc pour les éléments numériques ou les dernières données pour les éléments de type 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 les éléments. 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'intitulé *Configuration avancée* pour afficher les options de configuration avancée.

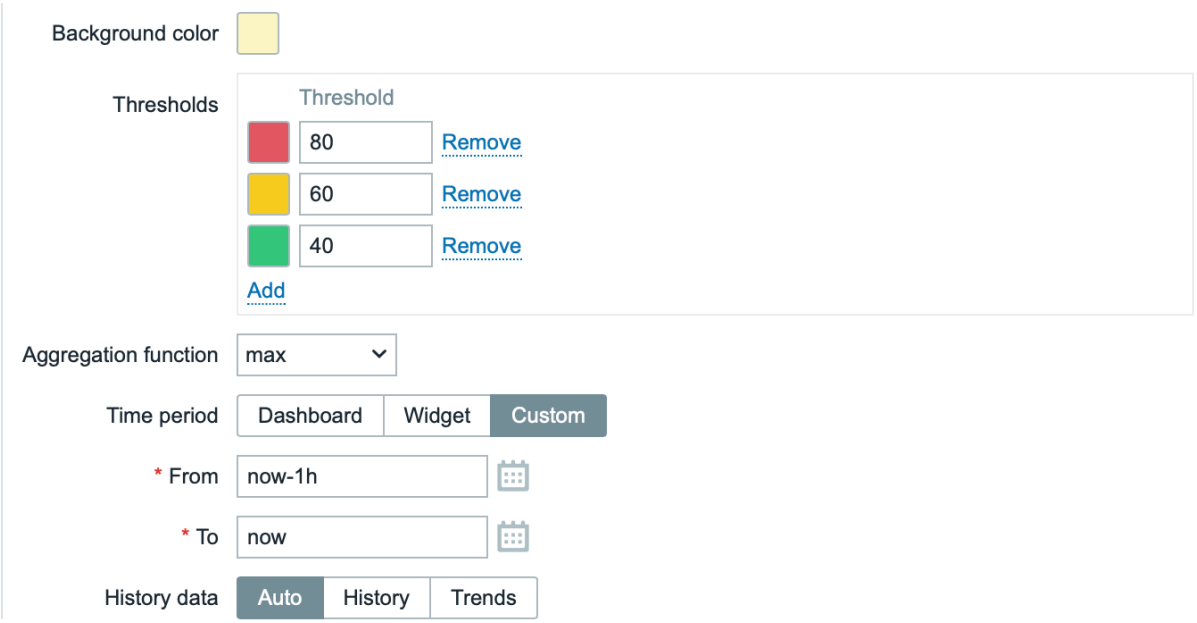
Configuration avancée

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) pour l'ensemble du widget ainsi qu'une fonction d'agrégation pour l'affichage des valeurs.

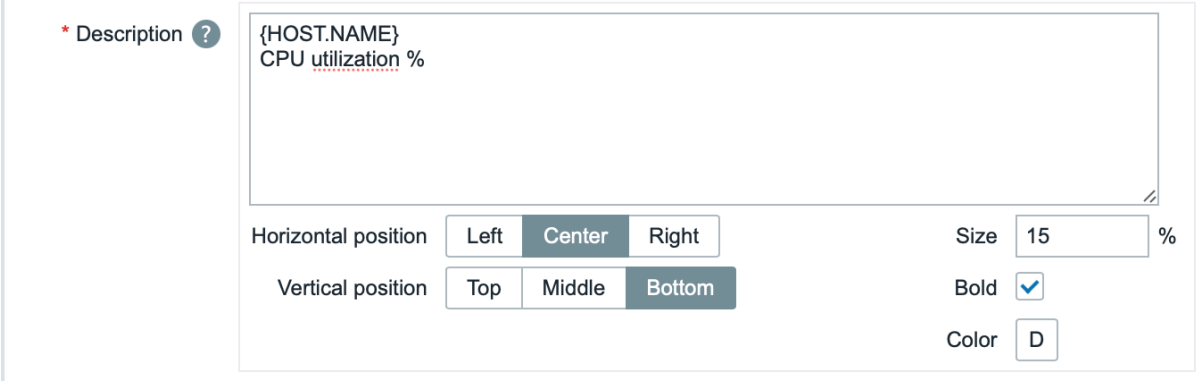


Couleur d'arrière-plan	Sélectionnez la couleur d'arrière-plan pour l'ensemble du widget à l'aide du sélecteur de couleurs. D correspond à la couleur par défaut (dépend du thème de l'interface). Pour revenir à la valeur par défaut, cliquez sur le bouton *Utiliser la valeur par défaut* dans le sélecteur de couleurs.
Seuils	Configurez la couleur d'arrière-plan dynamique pour 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. Une fois que la valeur de l'élément est égale ou supérieure à la valeur du seuil, la couleur d'arrière-plan changera. La liste sera triée par ordre croissant lors de l'enregistrement. Notez que la couleur d'arrière-plan dynamique ne s'affichera correctement que pour les éléments numériques.
Fonction d'agrégation	Indiquez quelle fonction d'agrégation utiliser : **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 ; **not used** - afficher la valeur la plus récente (sans agrégation). L'agrégation permet d'afficher une valeur agrégée pour l'intervalle choisi (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 l'agrégation des valeurs. 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 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. Ce paramètre n'est pas disponible si *Fonction d'agrégation* est définie sur "not used".
Widget	Saisissez ou sélectionnez un widget compatible comme source de données pour la période. Ce paramètre est disponible si *Période* est définie 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 *Période* est définie 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 *Période* est définie sur "Custom".
Données d'historique	Prendre les données depuis l'historique ou les tendances : **Auto** - sélection automatique ; **History** - prendre les données d'historique ; **Trends** - prendre les données de tendance. Ce paramètre s'applique uniquement aux données numériques. Les données non numériques seront toujours prises depuis l'historique.

Description

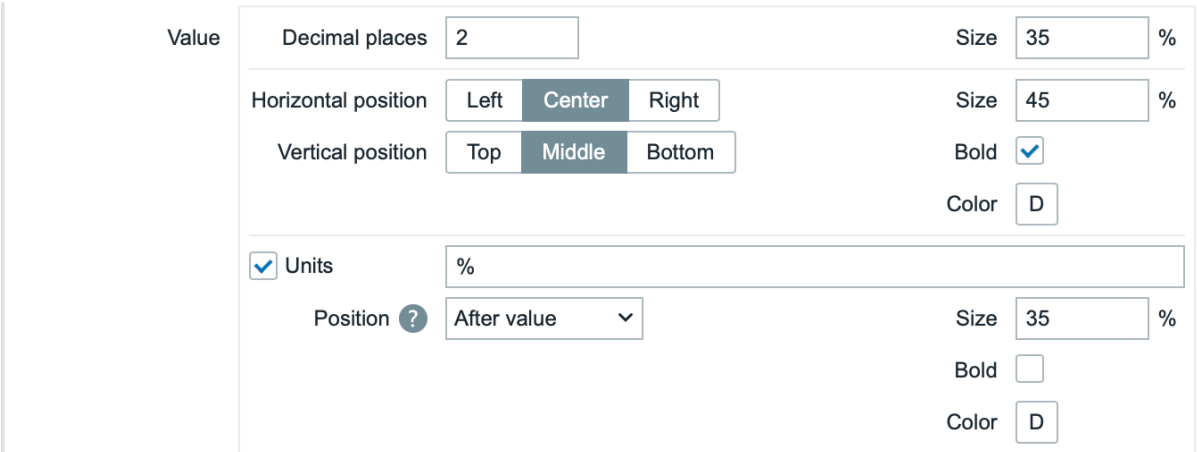
Ces paramètres déterminent comment la description de l'élément doit être affichée.



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

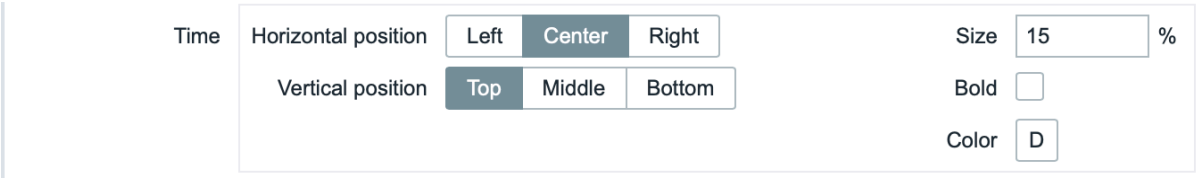
Ces paramètres déterminent comment la valeur de l'élément doit être affichée.



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 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 est en hausse (pour les éléments numériques)
- ↓ - la valeur de l'élément est en baisse (pour les éléments numériques)
- ↕ - la valeur de l'élément a changé (pour les éléments de type chaîne et les éléments avec correspondance de valeurs)

Remarque :

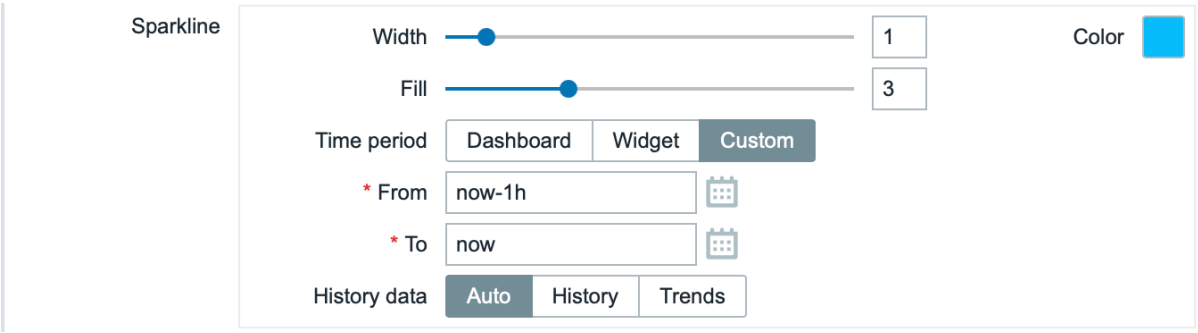
- Les indicateurs de hausse (↑) et de baisse (↓) ne sont pas affichés s'il n'y a qu'une seule valeur.
- Les valeurs sont toujours comparées dans le même périmètre ; par exemple :
  - La dernière valeur est comparée à la valeur précédente.
  - Une valeur mensuelle est comparée au mois précédent.
  - Pour 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 désigne 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.

La taille verticale de l'indicateur de changement est égale à la taille de la valeur (partie entière de la valeur pour les éléments numériques).

Mini-graphique

Ces paramètres déterminent comment le mini-graphique doit être affiché.

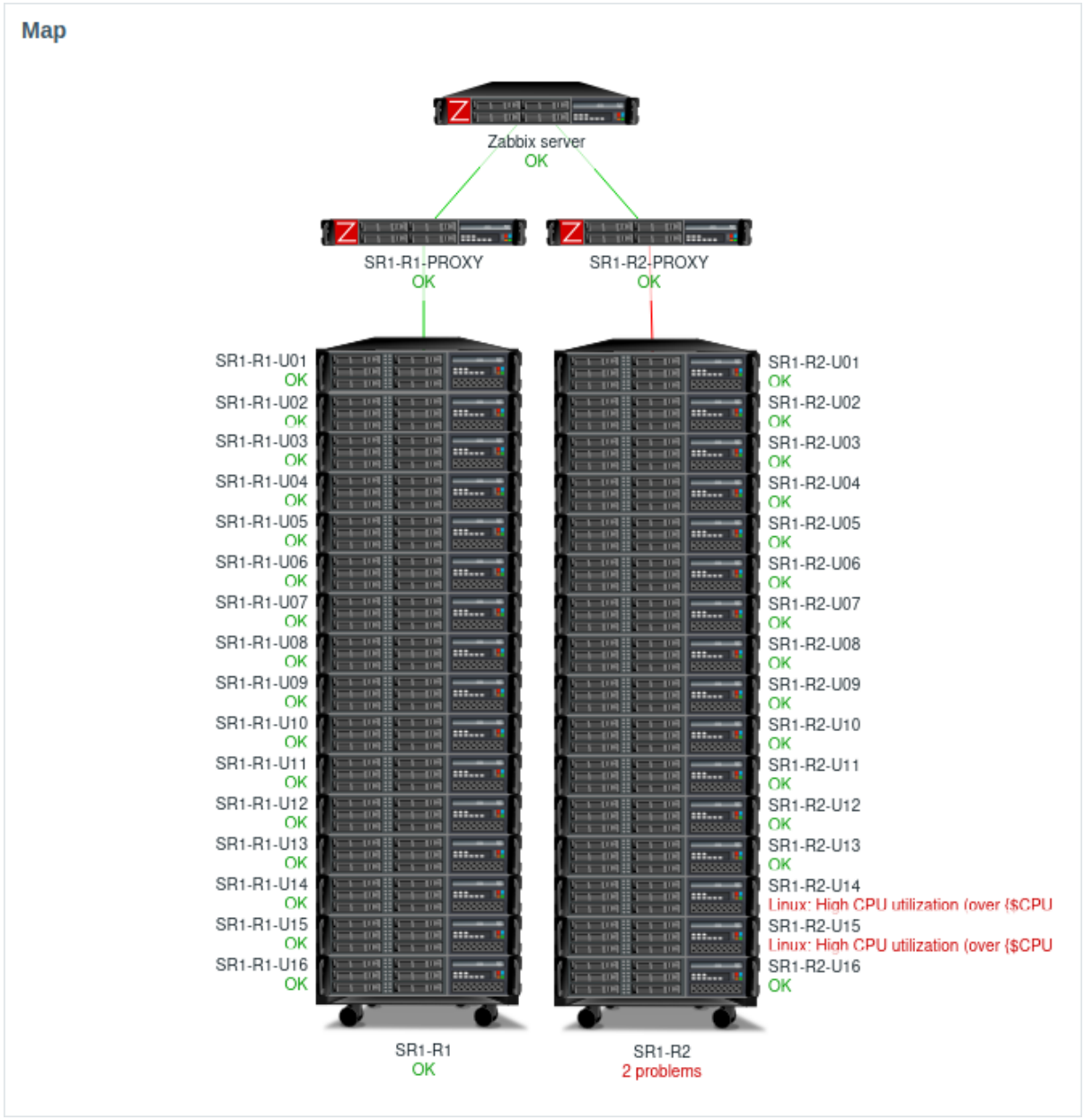


Largeur	Définissez l'épaisseur de la ligne du graphique à l'aide du curseur ou en saisissant manuellement une valeur comprise entre 0 et 10.
Couleur	Sélectionnez la couleur de la ligne et du remplissage.
Remplissage	Définissez le niveau de transparence de la couleur de remplissage à l'aide du curseur 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.

19 Carte

Vue d'ensemble

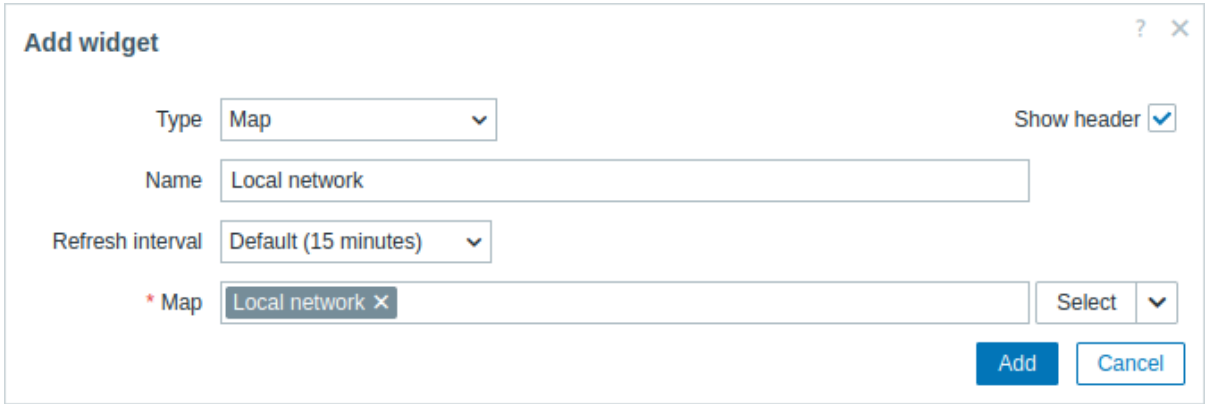
Le widget *Map* peut afficher une carte du réseau ou une visualisation similaire, vous offrant une vue d'ensemble dynamique de votre réseau.



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 :



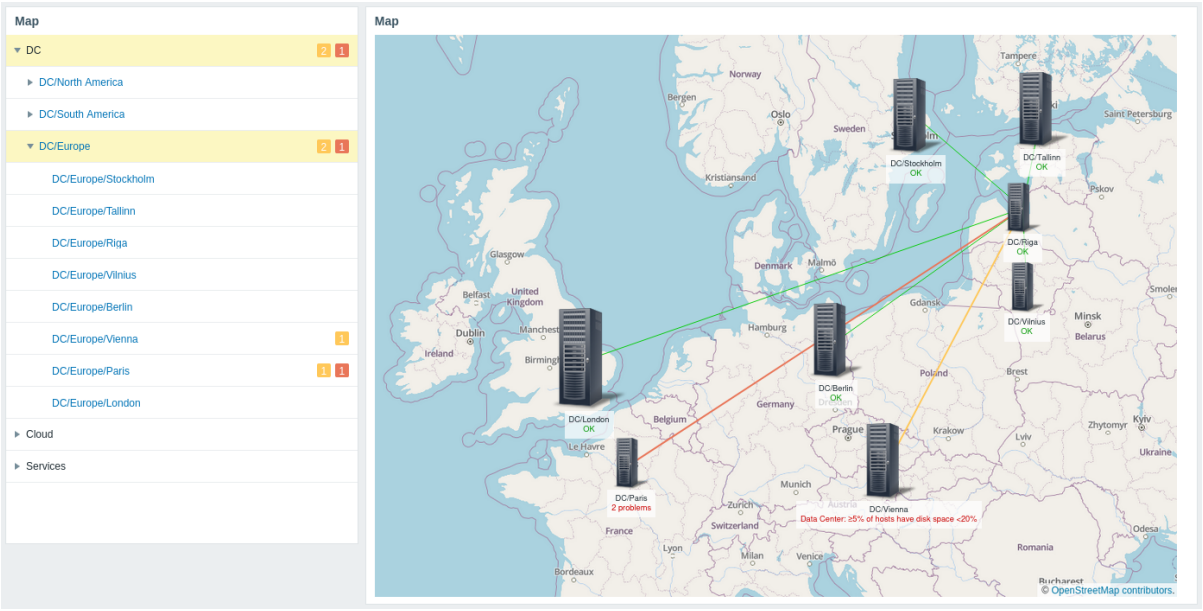
En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

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.

20 Arborescence de navigation des cartes

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.

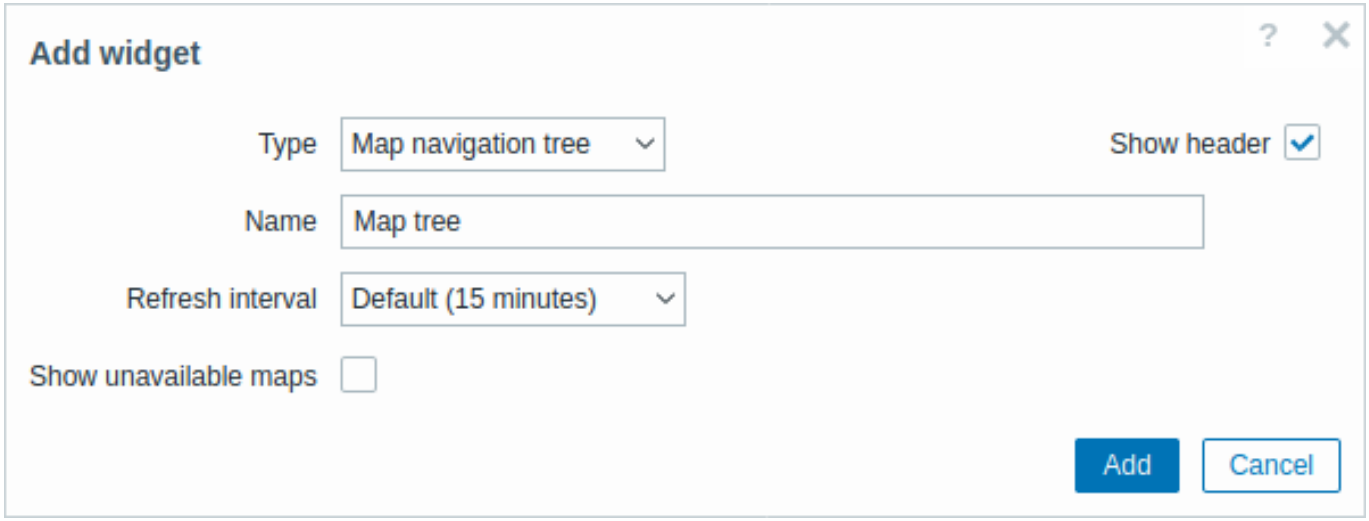


Vous pouvez l'associer au widget Map pour afficher dynamiquement la carte sélectionnée dans l'arborescence.

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 :

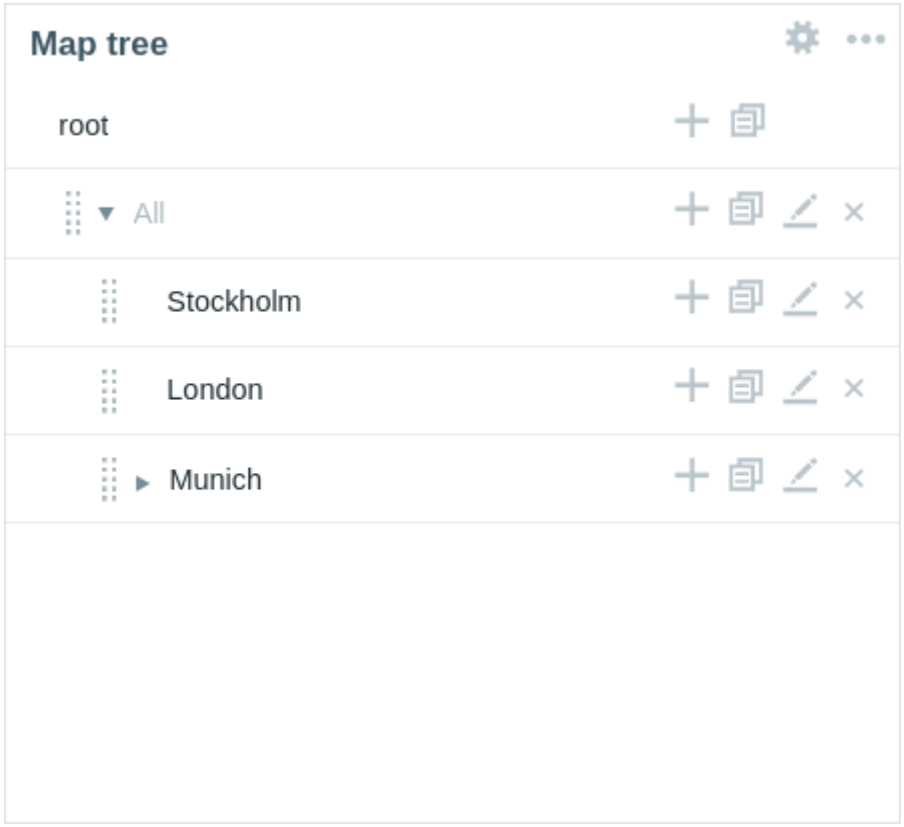


En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

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.

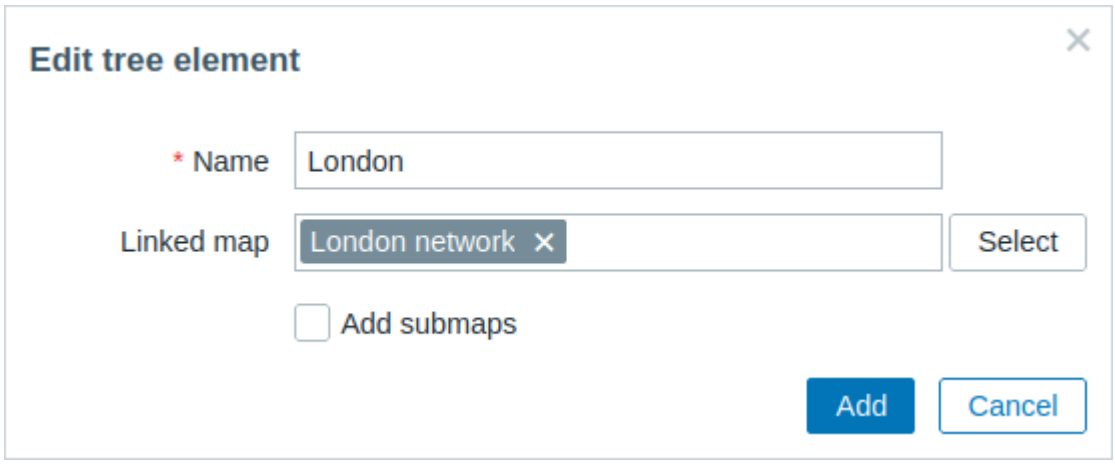
Les éléments de l'arborescence de navigation sont affichés dans une liste. Vous pouvez :

- 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.



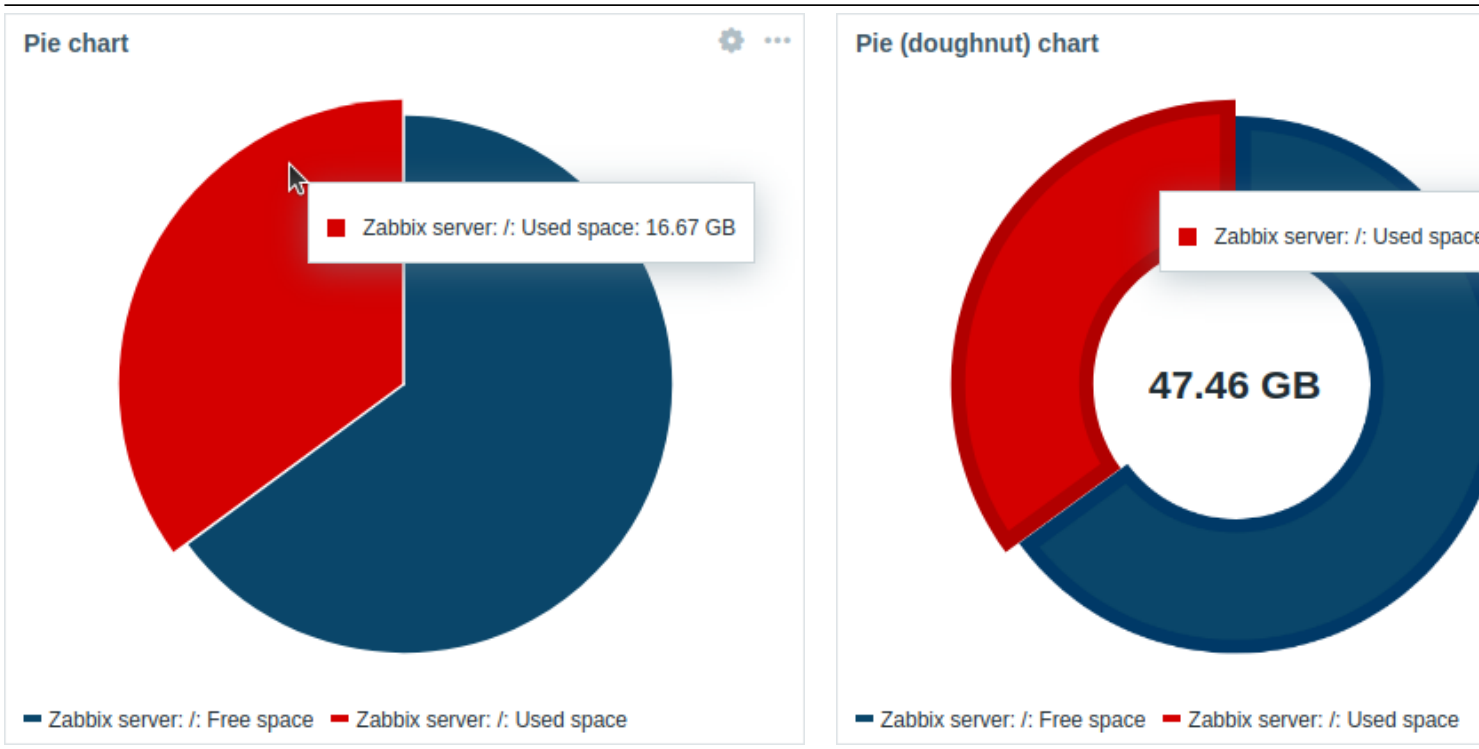
The following navigation tree element configuration parameters are available:

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.

21 Diagramme circulaire

Vue d'ensemble

Le widget *Camembert* affiche les données numériques des éléments sous forme de 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.

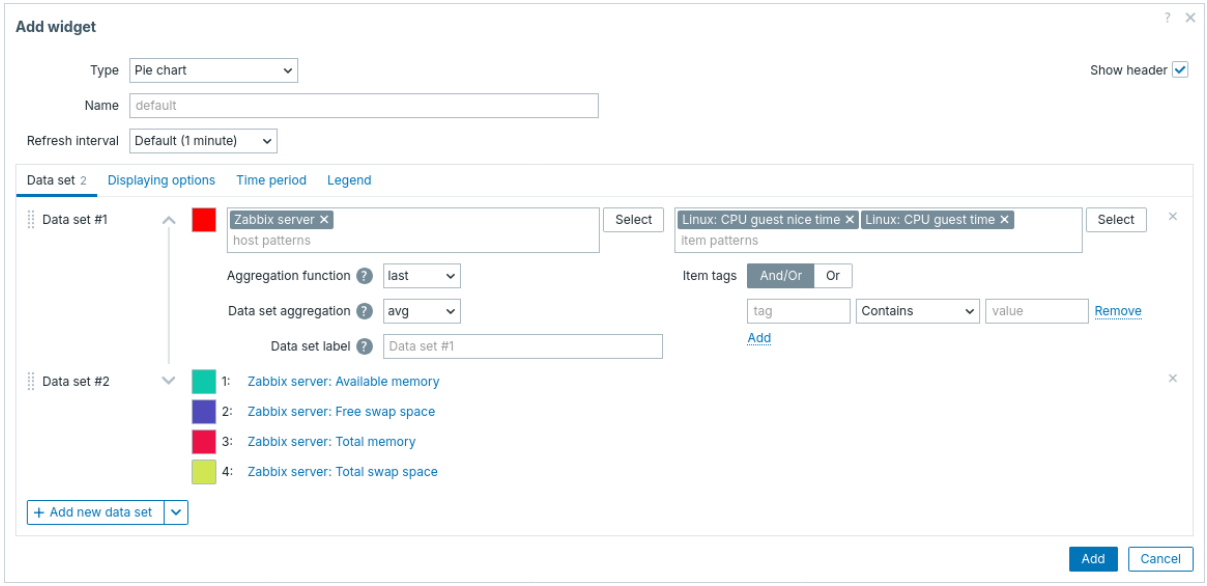


Au survol de la souris, le secteur survolé s'agrandit et affiche sa légende ; un clic sur le secteur le maintient agrandi et sa légende reste visible jusqu'à sa fermeture.

Les informations affichées dans le widget *Camembert* peuvent être téléchargées sous forme d'image PNG en sélectionnant l'option *Télécharger l'image* dans le menu du widget.

Configuration

Pour configurer, sélectionnez *Pie chart* comme type :



Jeu de données

L'onglet **Jeu de données** permet de sélectionner les données du graphique en secteurs en ajoutant des jeux de données. Deux types de jeux de données peuvent être ajoutés :

- *Motifs d'élément* - les données des éléments correspondants sont affichées. Vous pouvez choisir une seule couleur de base 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 *Motifs d'élément* est ajouté.

Jeu de données	Pour le jeu de données **Motifs d'élément** : Sélectionnez ou saisissez des motifs d'hôte et d'élément ; les données des éléments correspondant aux motifs saisis seront affichées dans le graphique en secteurs ; jusqu'à 50 éléments peuvent être affichés. Des motifs génériques peuvent être utilisés pour la sélection (par exemple, ∗ renverra les résultats correspondant à zéro ou plusieurs caractères). Pour spécifier un motif générique, saisissez la chaîne manuellement 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 des motifs d'hôte et d'élément est obligatoire pour les jeux de données « Item patterns ». Voir aussi : Détails de configuration du jeu de données. Pour le jeu de données **Liste d'éléments** : Sélectionnez les éléments pour le graphique en secteurs en cliquant sur le bouton *Ajouter un élément*. La liste déroulante *Type* après le nom de l'élément permet de sélectionner le type d'affichage pour chaque élément : **Normal** - la valeur de l'élément est représentée proportionnellement dans le graphique en secteurs (par défaut) ; **Total** - la valeur de l'élément occupe tout le graphique en secteurs. Notez qu'un seul élément « Total » peut exister par graphique en secteurs, et il sera placé en premier dans la légende du graphique en secteurs. 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 qu'aux éléments individuels. La spécification d'éléments ou de widgets est obligatoire pour les jeux de données « Item list ». Voir aussi : Détails de configuration du jeu de données. Notez que seuls les types d'élément numériques sont autorisés. Lors de la configuration du widget sur un tableau de bord de modèle, le paramètre permettant de spécifier des motifs d'hôte n'est pas disponible, et le paramètre permettant de spécifier une liste d'éléments 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 : **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 (5 minutes, une heure, un jour) sélectionné dans l'onglet *Période* ou utilisé pour l'ensemble du tableau de bord.
Agrégation du jeu de données	Spécifiez la fonction d'agrégation à utiliser pour l'ensemble du jeu de données : **not used** - 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 (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 en secteurs (pour les jeux de données agrégés). Tous les jeux de données sont numérotés, y compris ceux pour lesquels un *Libellé du jeu de données* est spécifié. Si aucun libellé n'est spécifié, le jeu de données sera automatiquement libellé selon son numéro (par exemple « Jeu de données n°2 », « Jeu de données n°3 », etc.). La numérotation des jeux de données est recalculée après réorganisation/glisser-déposer des jeux de données. Les libellés de jeu de données trop longs seront raccourcis pour s'adapter à l'emplacement d'affichage (par exemple « Nombre de proc... »).
Tags d'élément	Spécifiez des tags pour filtrer les éléments 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 : **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). 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.

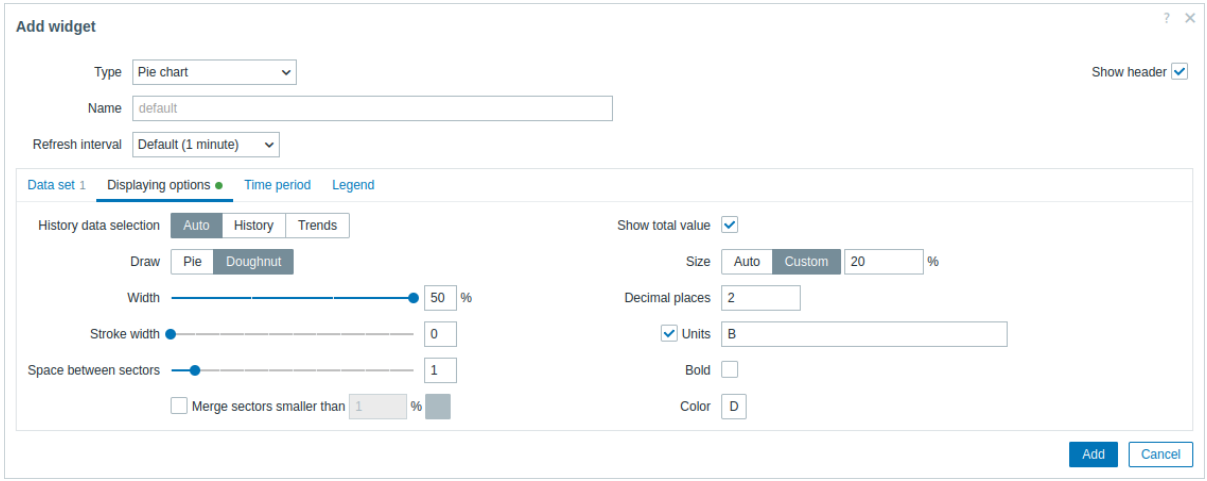
Détails de configuration des jeux de données

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

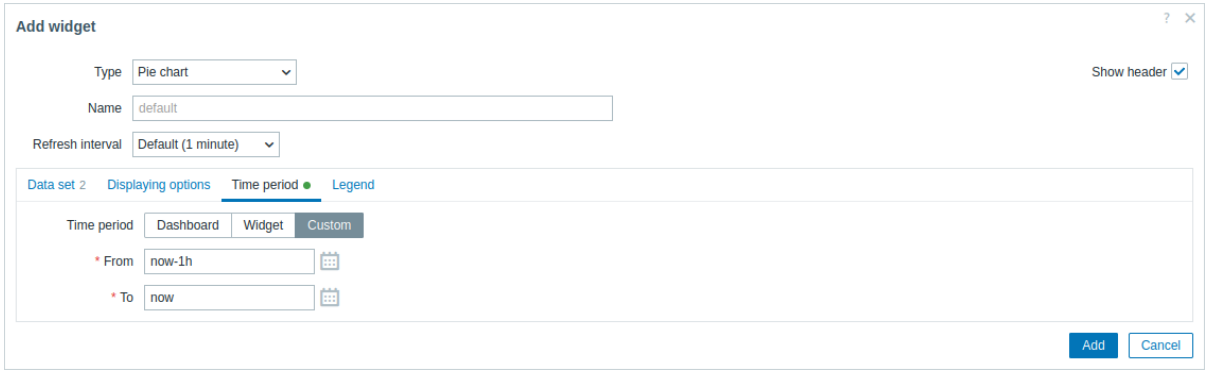
L'onglet **Options d'affichage** permet de définir les options de sélection des données d'historique et de visualisation pour le graphique en secteurs :



Sélection des données d'historique	Sélectionnez la source des données : **Auto** - les données sont extraites selon l'algorithme classique (par défaut) ; **History** - données issues de l'historique ; **Trends** - données issues des tendances.
Dessiner	Sélectionnez le style de visualisation du graphique en secteurs : **Pie** - un secteur complet (les secteurs occupent 100 % du rayon) ; **Doughnut** - un graphique en secteurs avec un cercle vide au milieu (les secteurs utilisent 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 plus petits que N %	Cochez la case pour fusionner les secteurs plus petits que N %. Si cette option est activée, sélectionnez la couleur des secteurs fusionnés ainsi que le seuil en pourcentage (N) pour la fusion des petits secteurs.
Style de dessin : ***Doughnut***	
Largeur	Sélectionnez la largeur du doughnut : 20, 30, 40 ou 50 % (par défaut) du rayon.
Largeur du contour	Sélectionnez la largeur de la bordure du secteur du doughnut (0-10).
Afficher la valeur totale	Cochez la case pour afficher la valeur totale au milieu du graphique en doughnut.
Taille	Sélectionnez l'option de taille pour la valeur totale : **Auto** - la taille du texte est ajustée automatiquement pour s'adapter de manière lisible au centre du doughnut ; **Custom** - spécifiez la taille du texte en pourcentage de hauteur par rapport à la hauteur totale du widget.
Décimales	Indiquez le nombre de décimales pour la valeur totale (0-6).
Unités	Indiquez les unités pour la valeur totale.
Gras	Cochez la case pour afficher la valeur totale en gras.
Couleur	Sélectionnez la couleur de la valeur totale.

Période de temps

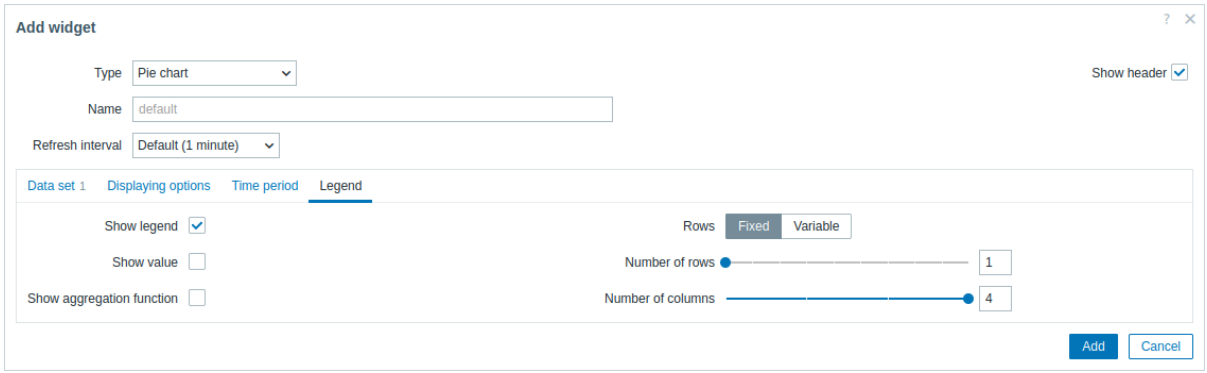
L'onglet **Période de temps** permet de définir une période de temps personnalisée pour les paramètres d'agrégation du graphique en secteurs :



Période de temps	Sélectionnez la source de données pour la période de temps : **Tableau de bord** - utiliser le sélecteur de période de temps du tableau de bord ; **Widget** - utiliser un widget compatible (défini dans le paramètre *Widget*) ; **Personnalisée** - utiliser une période de temps personnalisée spécifiée 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 de temps* 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 (*Graphique*, *Graphique (classique)*, *Prototype de graphique*) comme source de données pour la période de temps. Ce paramètre est disponible si *Période de temps* est défini sur "Widget".
De	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 *Période de temps* est défini sur "Personnalisée".
À	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 *Période de temps* est défini sur "Personnalisée".

Légende

L'onglet **Légende** permet de personnaliser la légende du graphique en secteurs :

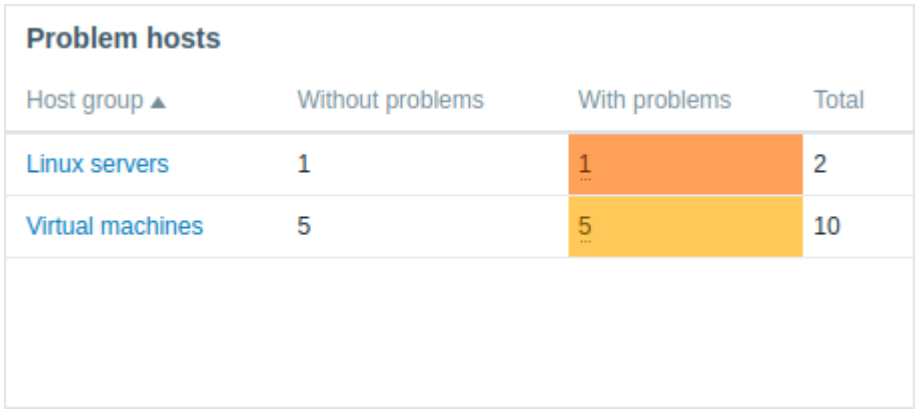


Afficher la légende	Décochez cette case pour masquer la légende sur le graphique en secteurs (cochée par défaut).
Afficher la valeur	Cochez cette case pour afficher la valeur de l'élément dans la légende.
Afficher la fonction d'agrégation	Cochez cette case pour afficher la fonction d'agrégation dans 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* est décoché.

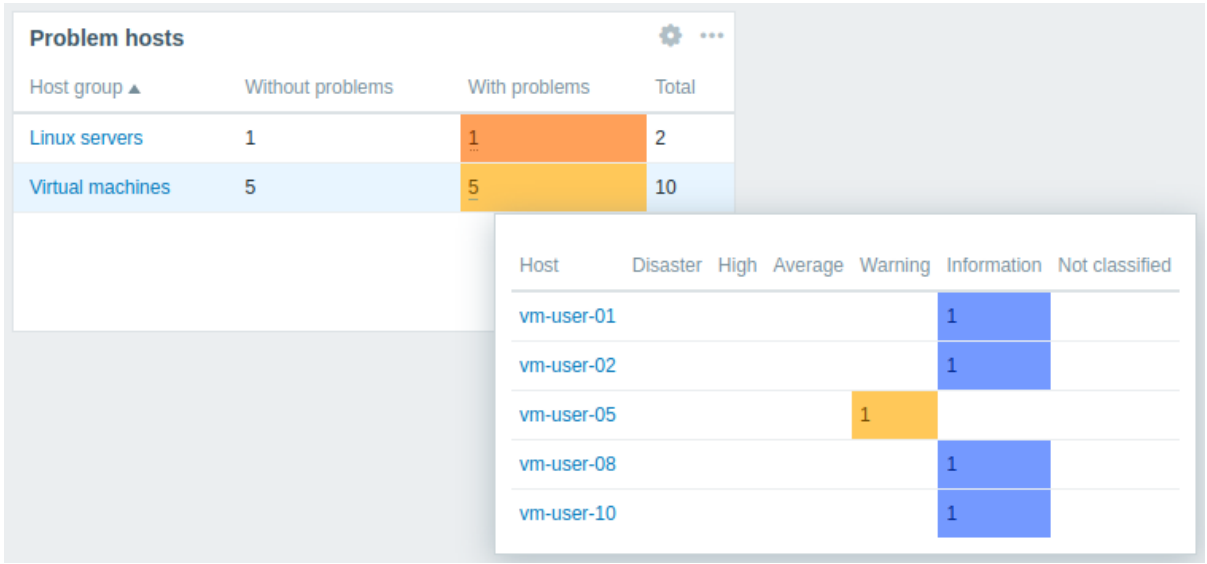
22 Hôtes en problème

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.).

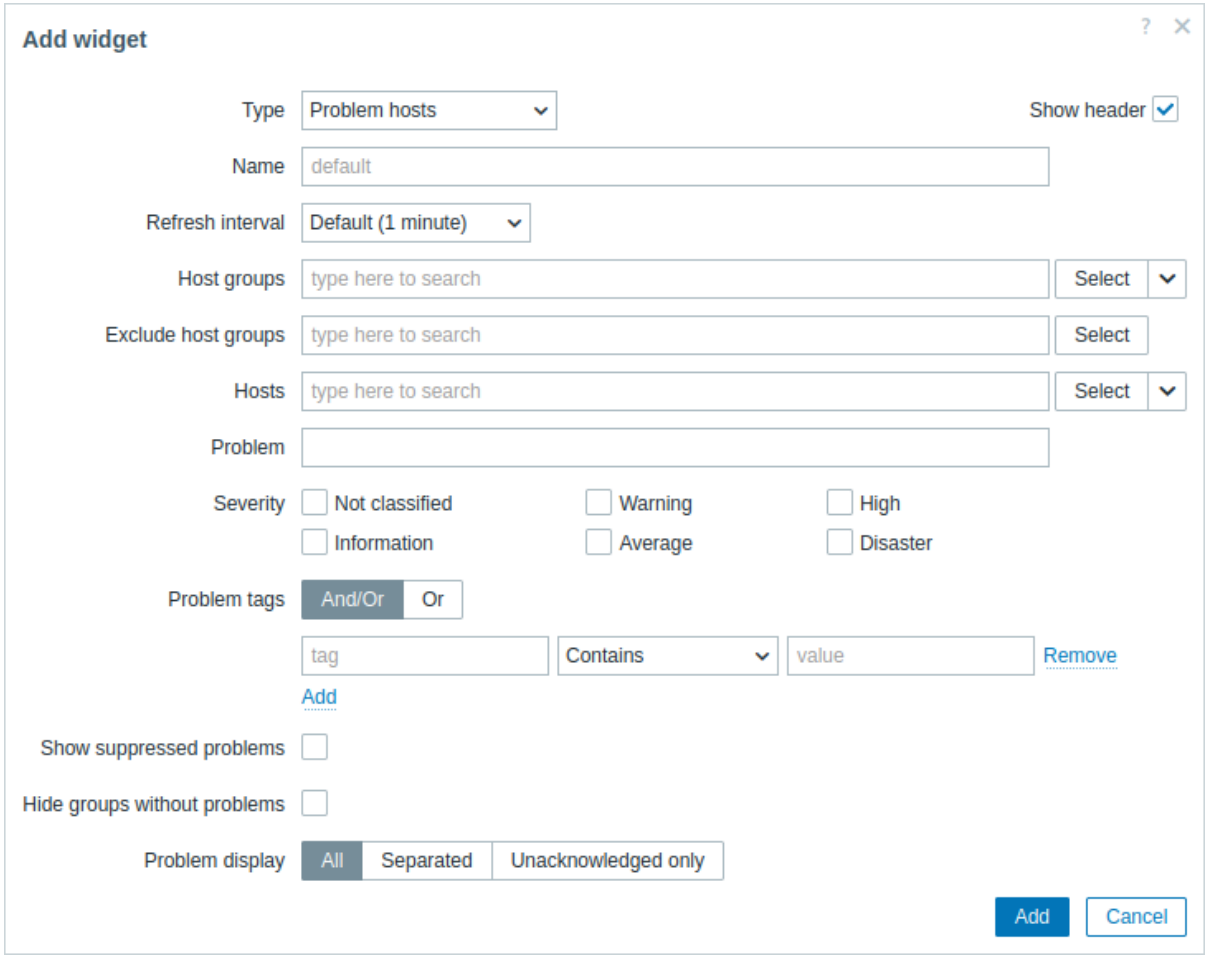


En survolant un nombre de problèmes, la liste des hôtes présentant des problèmes s'affiche :



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 :

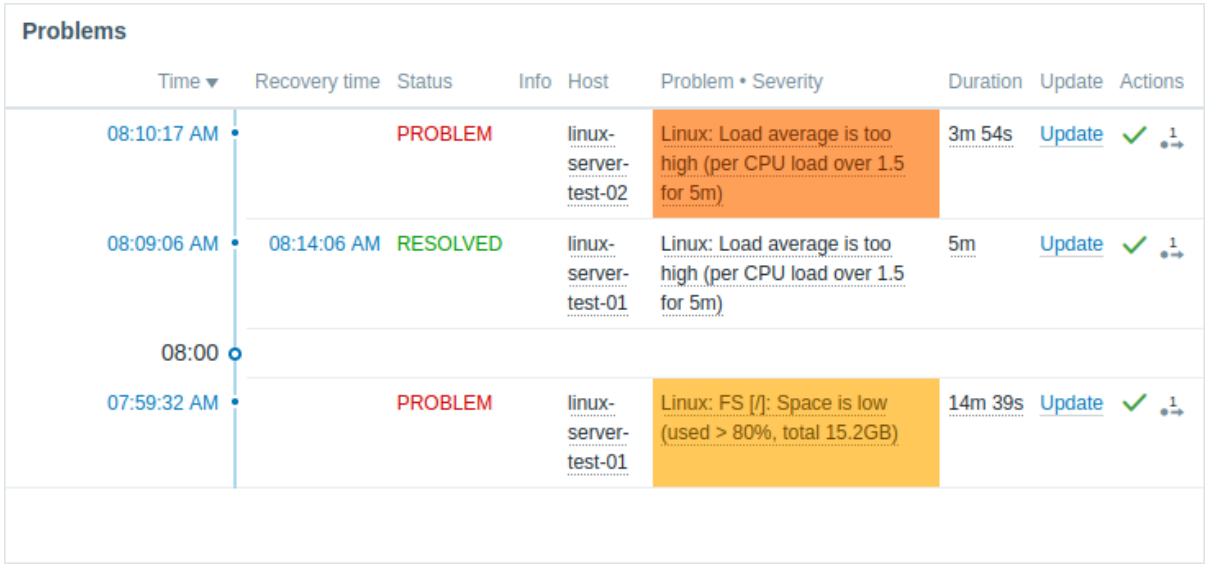
Groupes d'hôtes	Sélectionnez les groupes d'hôtes à afficher dans le widget. Vous pouvez également sélectionner un widget compatible comme source de données pour les groupes d'hôtes. Ce champ prend en charge l'autocomplétion ; ainsi, lorsque vous commencez à saisir le nom d'un groupe, une liste déroulante 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.

23 Problèmes

Vue d'ensemble

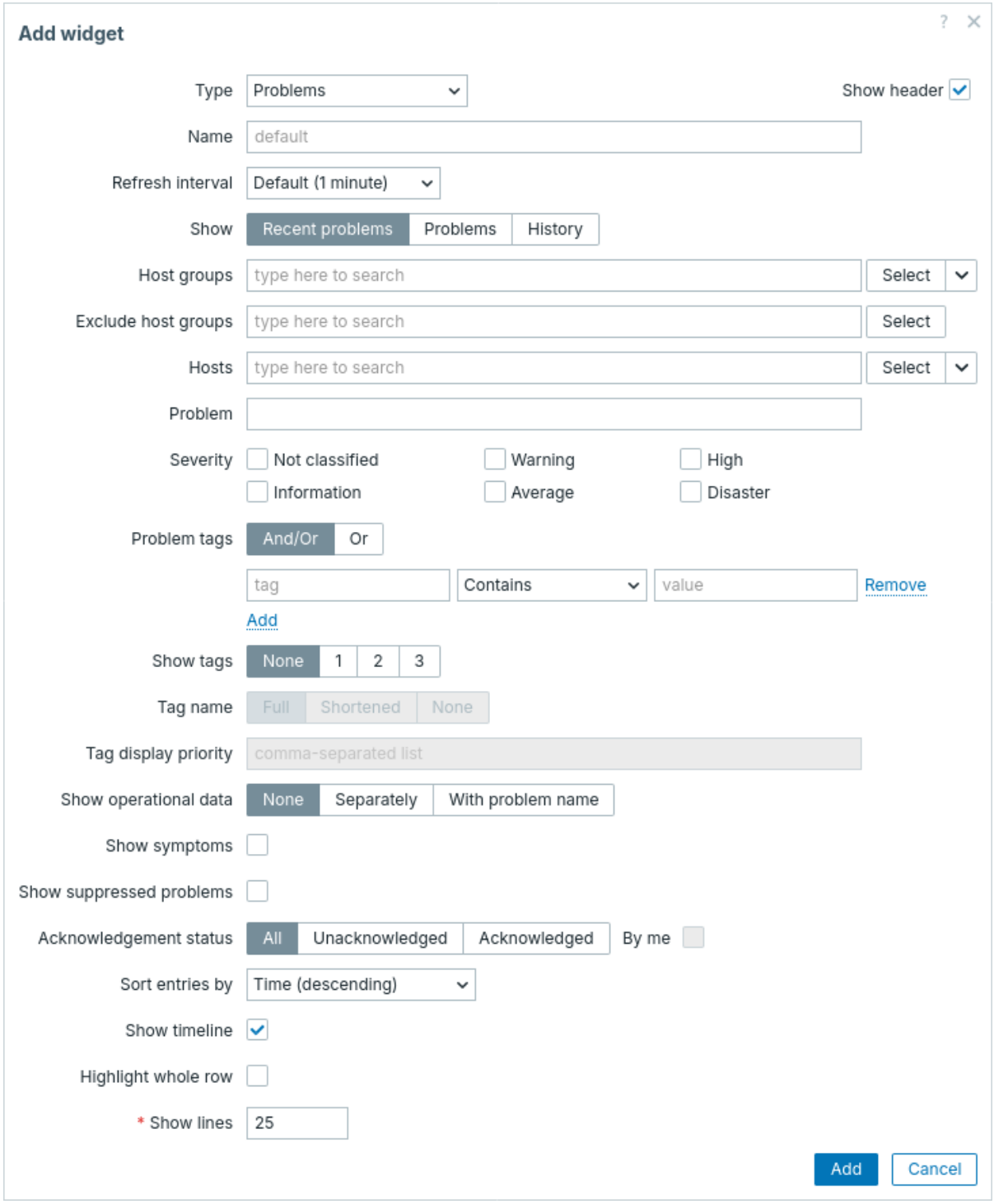
Le widget *Problems* affiche les problèmes actuels filtrés selon différents paramètres (groupes d'hôtes, hôtes, noms de problèmes, etc.), vous offrant une vue claire de ce qui nécessite votre attention.



Il affiche les mêmes données que *Monitoring* > *Problems* et peut afficher jusqu'à 1000 enregistrements.

Configuration

Pour configurer le widget, sélectionnez *Problems* comme type :



Vous pouvez limiter de différentes manières le nombre de problèmes affichés dans le widget : par statut du problème, nom du problème, sévérité, groupe d'hôtes, hôte, tag d'événement, statut d'acquittement, etc.

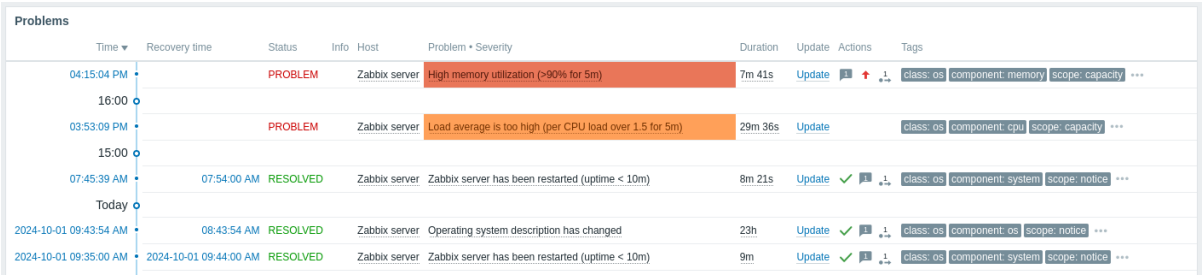
En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

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 prend en charge l'autocomplétion : 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 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 prend en charge l'autocomplétion : 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 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 prend en charge l'autocomplétion : 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 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 niveaux de sévérité des problèmes pour filtrer les problèmes à afficher dans le widget. Si aucun niveau de sévérité 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 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. Lors du filtrage, les tags spécifiés ici seront affichés en premier avec le problème, sauf si cela est remplacé 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, passez la souris sur l'icône à trois points.
Tag name	Sélectionnez le mode d'affichage du nom du tag : **Full** - les noms et les valeurs des tags sont affichés en entier ; **Shortened** - les noms des tags sont abrégés à 3 caractères, mais les valeurs des tags sont affichées en entier ; **None** - seules les valeurs des tags sont affichées, sans les noms.
Tag display priority	Saisissez la priorité d'affichage des tags pour un problème, sous la forme d'une liste de tags séparés par des virgules. Seuls les noms de tag doivent être utilisés, sans valeurs. Exemple : `Services,Applications,Application` Les tags de cette liste seront toujours affichés en premier, 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 distincte ; **With problem name** - ajoute les données opérationnelles au nom du problème, en utilisant des parenthèses pour les données opérationnelles.
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 individuelle.
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** (ordre décroissant ou croissant) ; **Severity** (ordre décroissant ou croissant) ; **Problem name** (ordre décroissant ou croissant) ; **Host** (ordre décroissant ou croissant). Le tri des entrées par **Host** (ordre décroissant ou croissant) 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 toute la ligne pour les problèmes non résolus. La couleur de sévérité 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 problème à afficher.

Utilisation du widget



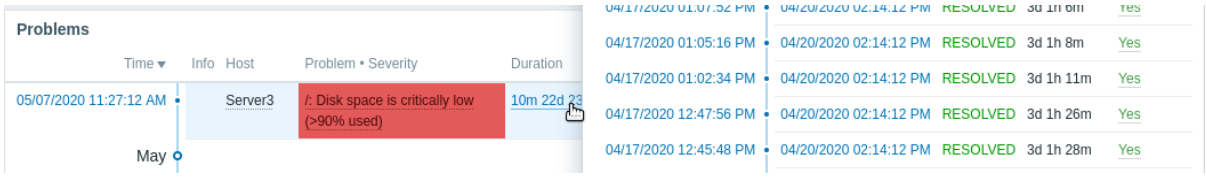
Le widget Problèmes offre un accès rapide à des informations supplémentaires :

- Cliquez sur la date du problème (dans la colonne Heure) ou sur la date de rétablissement (dans la colonne Heure de rétablissement) 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.
- Appuyer sur Mettre à jour 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 comprend la liste des événements de problème pour ce déclencheur et, si elle est définie, la description du déclencheur ainsi qu'une URL cliquable.



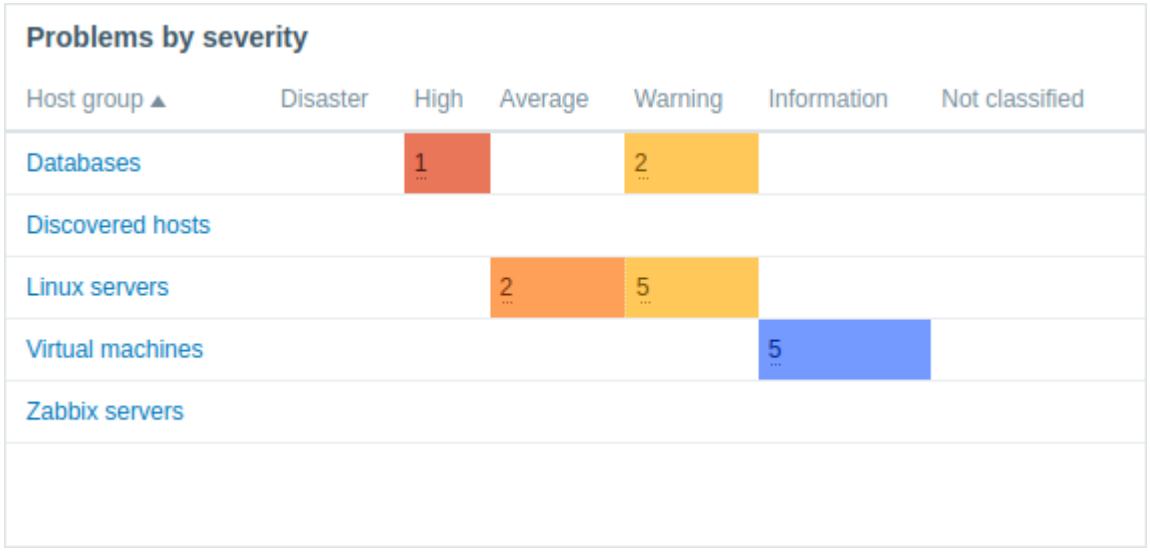
Pour afficher la fenêtre contextuelle de l'événement de problème :

- Passez le pointeur de la souris sur la durée du problème dans la colonne *Durée* du widget *Problèmes*. La fenêtre contextuelle disparaît dès que vous éloignez le pointeur de la durée.
- Cliquez sur la durée dans la colonne *Durée* du widget *Problèmes*. La fenêtre contextuelle ne disparaît que si vous cliquez de nouveau sur la durée.

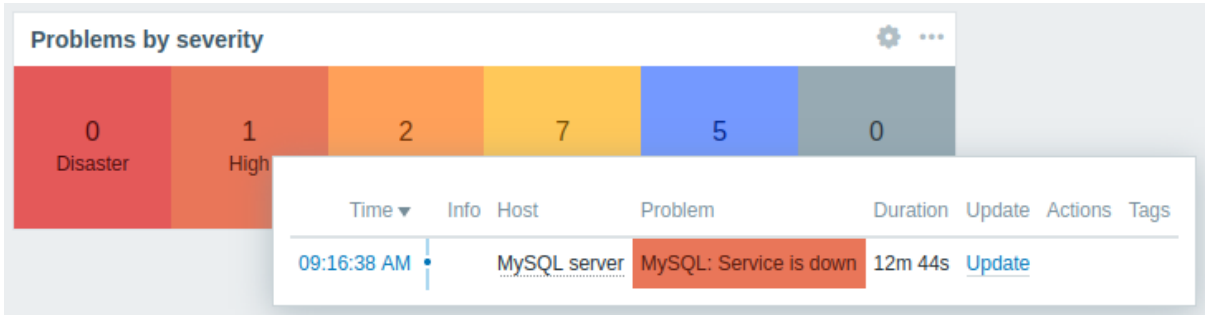
24 Problèmes par gravité

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.).

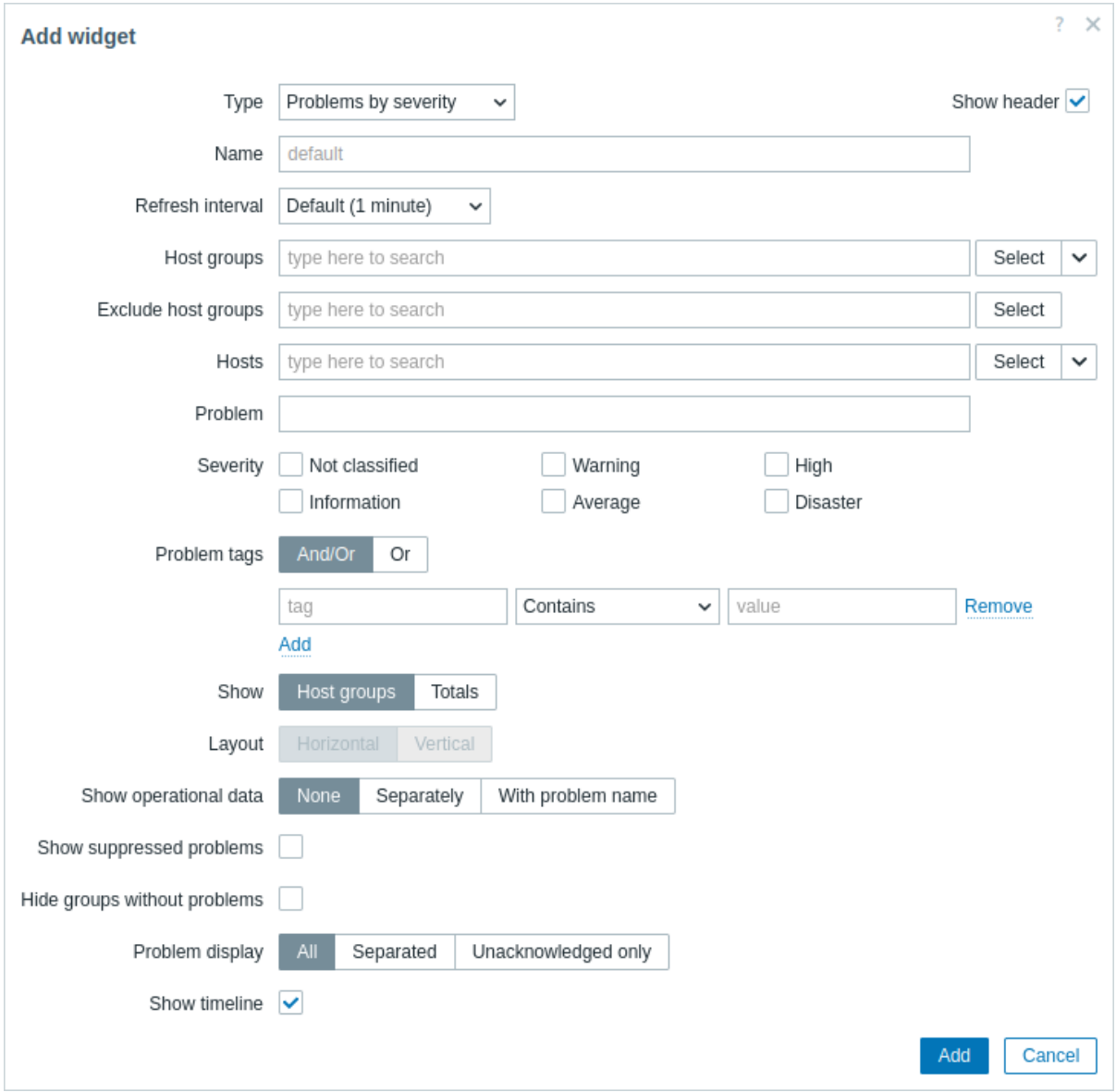


En survolant un nombre de problèmes, la liste des problèmes associés s'affiche :



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 :

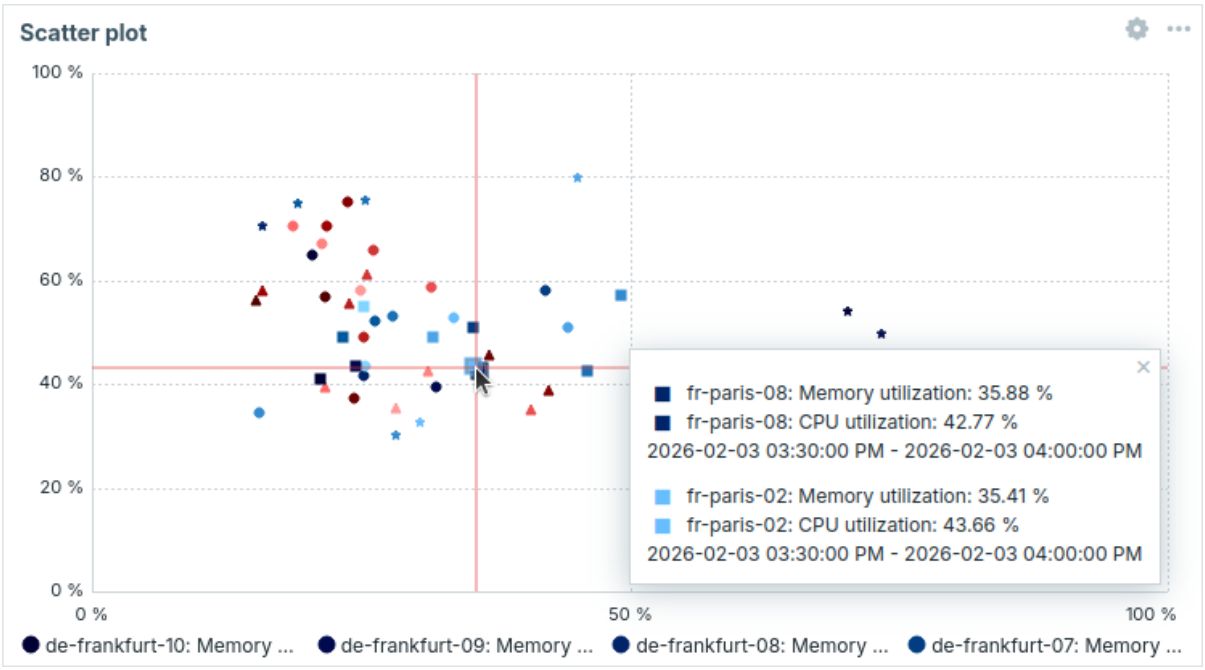
Host groups	Sélectionnez les groupes d'hôtes à afficher dans le widget. Vous pouvez également sélectionner un widget compatible comme source de données pour les groupes d'hôtes. Ce champ prend en charge l'autocomplétion ; ainsi, lorsque vous commencez à saisir le nom d'un groupe, une liste déroulante 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.
Exclude host groups	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.

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 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.
Problem	Vous pouvez limiter le nombre d'hôtes en problème affichés selon 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 sévérités de problème pour filtrer les problèmes à afficher dans le widget. Si aucune sévérité 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 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 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). 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.
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 sévérité 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 (voir la description de *Operational data* dans *Monitoring* > *Problems*) comme suit : **None** - les données opérationnelles ne seront pas affichées **Separately** - les données opérationnelles seront affichées sur une ligne distincte **With problem name** - les données opérationnelles seront affichées avec le nom du problème.
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 complet de problèmes sera affiché ; **Separated** - le nombre de problèmes non acquittés sera affiché séparément comme une partie du nombre total de problèmes ; **Unacknowledged only** - seul le nombre de problèmes non acquittés sera affiché.
Show timeline	Cochez la case pour afficher une chronologie visuelle.

25 Nuage de points

Aperçu

Le widget *Nuage de points* affiche la relation entre deux métriques en traçant des points de données individuels le long des axes X et Y. Cela permet de révéler des motifs, des groupes, des corrélations et des valeurs aberrantes dans l'ensemble de données.



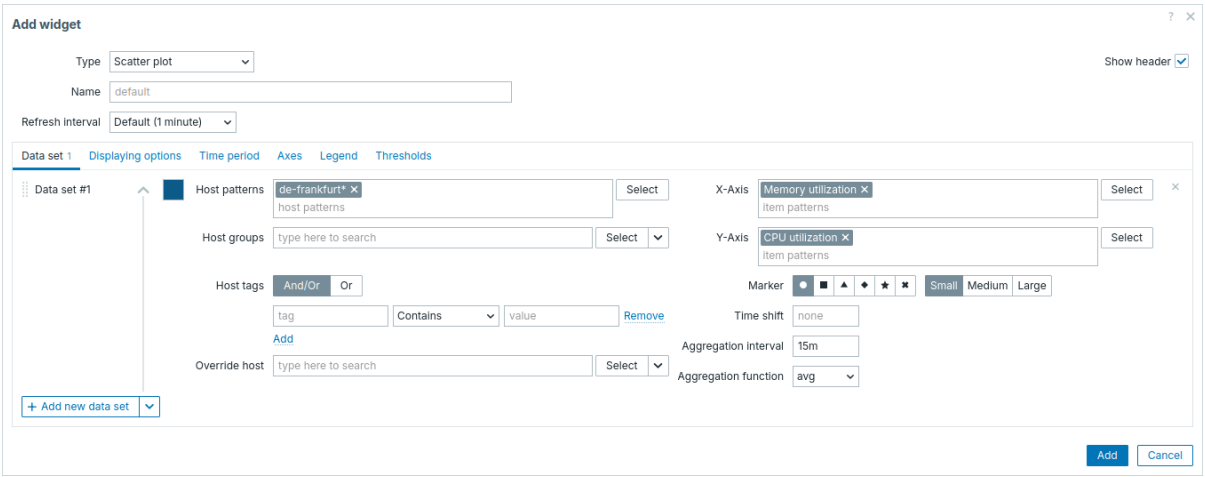
En survolant un point de données, une infobulle s'affiche avec une liste des éléments de ce point de données et d'autres points situés dans un rayon de 6 px ; triés d'abord selon les valeurs de l'axe X du nuage de points par ordre croissant, puis selon les valeurs de l'axe Y par ordre décroissant.

La sélection d'un élément dans l'infobulle diffuse ses données vers d'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 du widget.

Configuration

Pour configurer, sélectionnez *Scatter plot* comme type :



Jeu de données

L'onglet **Jeu de données** permet de sélectionner les données du nuage de points en ajoutant des jeux de données. Deux types de jeux de données peuvent être ajoutés :

- *Motifs d'éléments* - les hôtes avec des éléments correspondants sont affichés comme points de données. Vous pouvez choisir une seule couleur de base ou sélectionner une ligne de palette pour attribuer des couleurs distinctes à chaque hôte avec les éléments correspondants.
- *Liste d'éléments* - les éléments sélectionnés sont affichés comme points de données, où chaque point représente une paire d'éléments (X et Y). Vous pouvez choisir individuellement la couleur de chaque jeu 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 les deux axes.

Par défaut, un jeu de données *Motifs d'éléments* est ajouté.

Jeu de données	Pour le jeu de données **Motifs d'éléments** : Sélectionnez ou saisissez des motifs d'hôte et d'élément ; les données des éléments correspondant à ces motifs seront affichées sur le nuage de points ; jusqu'à 100 éléments peuvent être affichés (50 sur chaque axe pour un total de 50 hôtes). Des motifs génériques peuvent être utilisés pour la sélection (par exemple, ∗ renverra les résultats correspondant à zéro ou plusieurs caractères). Pour spécifier un motif générique, saisissez la chaîne manuellement 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 des motifs d'hôte et d'élément est obligatoire pour les jeux de données « Item patterns ». Voir aussi : Détails de configuration du jeu de données. Pour le jeu de données **Liste d'éléments** : Sélectionnez les éléments pour le 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 jeux de données « Item list ». Voir aussi : Détails de configuration du jeu de données. Notez que seuls les types d'élément numériques sont autorisés. Lors de la configuration du widget sur un tableau de bord de modèle, le paramètre permettant de spécifier des motifs d'hôte n'est pas disponible, et le paramètre permettant de spécifier une liste d'éléments permet de sélectionner uniquement les 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 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 est disponible uniquement pour les jeux de données *Motifs d'éléments* et n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.

Tags d'hôte	Spécifiez des tags pour filtrer les 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 dont les valeurs 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 dont les valeurs 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 est disponible uniquement pour les jeux de données *Motifs 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'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.
Marqueur	Choisissez le type et la taille du marqueur pour les points de données d'hôte dans le jeu de données.
Décalage temporel	Spécifiez un décalage temporel si nécessaire. Vous pouvez utiliser des suffixes de temps dans ce champ. Les valeurs négatives sont autorisées.
Fonction d'agrégation	Spécifiez la fonction d'agrégation à utiliser : **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 les deux axes) ; **first** - afficher la première valeur ; **last** - afficher la dernière valeur ; **none** - afficher toutes les valeurs (sans agrégation). L'agrégation permet d'afficher une valeur agrégée pour l'intervalle choisi (5 minutes, une heure, un jour), au lieu de toutes les valeurs. Voir aussi : Agrégation dans les graphiques.
Intervalle d'agrégation	Spécifiez l'intervalle d'agrégation des valeurs. Vous pouvez utiliser des suffixes de temps dans ce champ. Une valeur numérique sans suffixe sera considérée comme des secondes. Notez que si le widget est configuré pour afficher des données historiques basées sur les tendances (*Sélection des données d'historique* est définie sur *Tendances* ou *Auto*), il est recommandé d'utiliser un intervalle d'agrégation multiple de 1 heure (par exemple, 3600, 60m, 1h, 3h, etc.). Les tendances stockent des valeurs agrégées par heure ; ainsi, l'utilisation d'un intervalle d'agrégation qui n'est pas un multiple de 1 heure (par exemple, 100s, 7min, 15min, 90min, etc.) peut conduire à des résultats difficiles à interpréter. La spécification d'un intervalle d'agrégation est obligatoire. L'intervalle d'agrégation détermine l'intervalle de temps et les coordonnées d'un point de données ; par exemple, si les éléments de l'axe X ont 4 valeurs et les éléments de l'axe Y ont 1 valeur, les coordonnées reflètent l'agrégation de 4 sur X et de 1 sur Y dans l'intervalle spécifié.

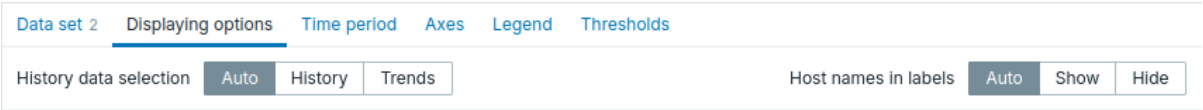
Détails de configuration des ensembles de données

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

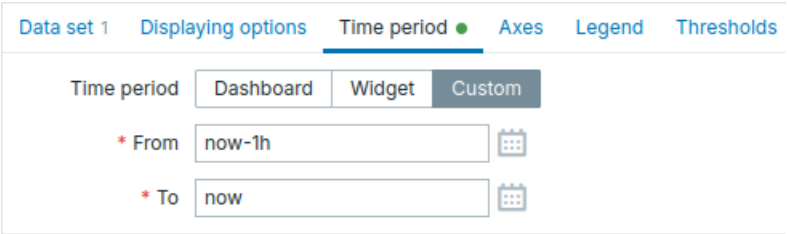
L'onglet **Options d'affichage** permet de définir la sélection des données d'historique :



Sélection des données d'historique	Définissez la source des données du nuage de points : **Auto** - les données sont extraites selon l'algorithme des graphiques classiques (par défaut). **Historique** - données provenant de l'historique. **Tendances** - données provenant des tendances.
Noms d'hôte dans les libellés	Choisissez s'il faut afficher les noms d'hôte dans la légende du graphique et dans l'infobulle qui apparaît au survol des données du graphique : **Auto** - les noms d'hôte ne sont affichés que si plusieurs hôtes sont présents dans les jeux de données (par défaut). **Afficher** - les noms d'hôte sont affichés. **Masquer** - les noms d'hôte sont masqués. Ce paramètre n'est pas disponible lors de la configuration du widget sur un tableau de bord de modèle.

Période de temps

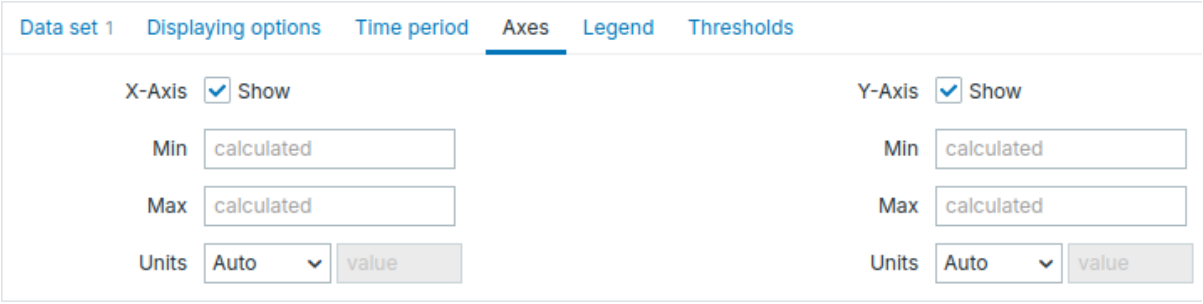
L'onglet **Période de temps** permet de définir une période pour laquelle afficher les données dans le nuage de points :



Période de temps	Sélectionnez la source de données pour la période de temps : **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 de temps* 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 *Période de temps* est défini sur "Widget".
De	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 *Période de temps* est défini sur "Personnalisée".
À	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 *Période de temps* est défini sur "Personnalisée".

Axes

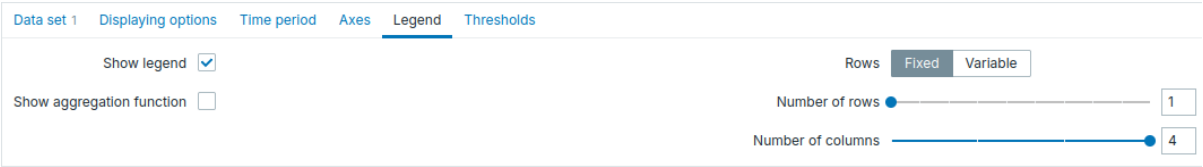
L'onglet **Axes** permet de personnaliser l'affichage des axes :



Axe X/Axe Y	Décochez cette case pour masquer l'axe X/l'axe Y (cochée 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 nuage de points dans la liste déroulante : **Auto** - les valeurs des axes sont affichées en utilisant l'unité du premier élément de l'ensemble de données ; **Statique** - les valeurs des axes sont affichées en utilisant l'unité spécifiée dans le champ de saisie *value* ; si le champ est laissé vide, seules les valeurs numériques sont affichées.

Légende

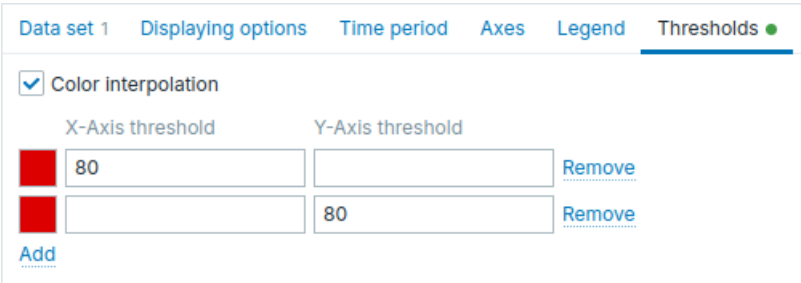
L'onglet **Légende** permet de personnaliser la légende du nuage de points :



Afficher la légende	Décochez cette case pour masquer la légende sur le nuage de points (cochée par défaut).
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 min/moy/max* n'est pas coché.

Seuils

L'onglet **Seuils** permet de configurer des seuils qui déterminent les couleurs des marqueurs d'hôte :



Interpolation des couleurs	Cochez la case pour activer une transition fluide entre les couleurs de seuil pour les marqueurs d'hôte. Ce paramètre s'applique si deux seuils ou plus 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 correspondances de valeurs sont prises en charge.

26 Rapport SLA

Aperçu

Le widget *Rapport SLA* affiche les rapports SLA, vous aidant à surveiller les performances des services par rapport aux objectifs définis.

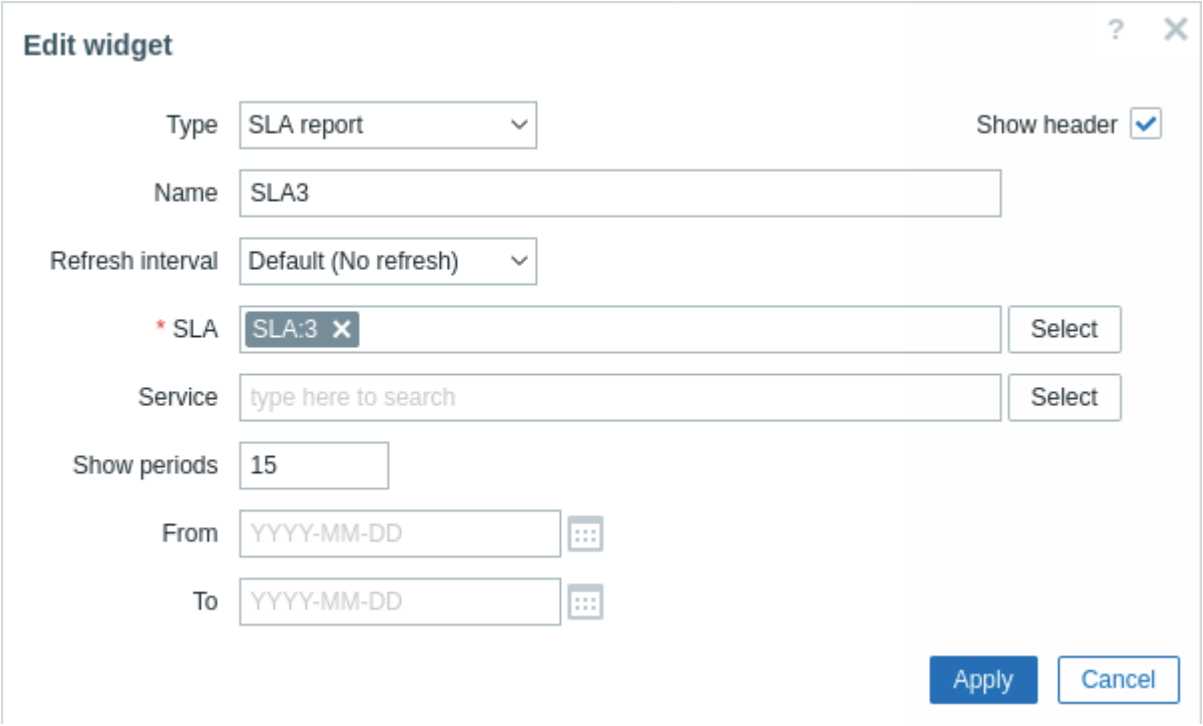
SLA report

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

Il affiche les mêmes données que *Services* > *Rapport SLA*.

Configuration

Pour configurer, sélectionnez *Rapport de SLA* comme type :



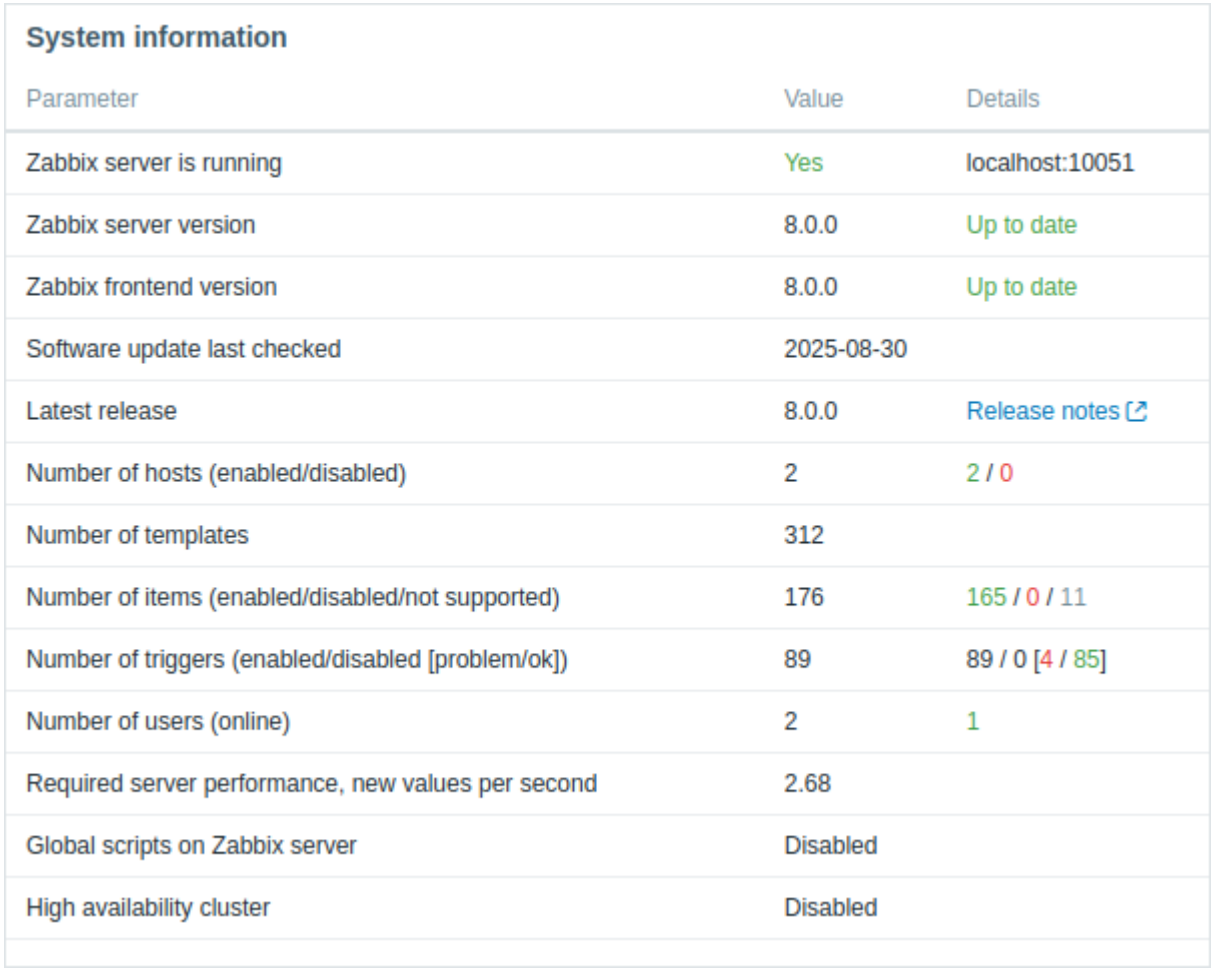
En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

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.

27 Informations système

Vue d'ensemble

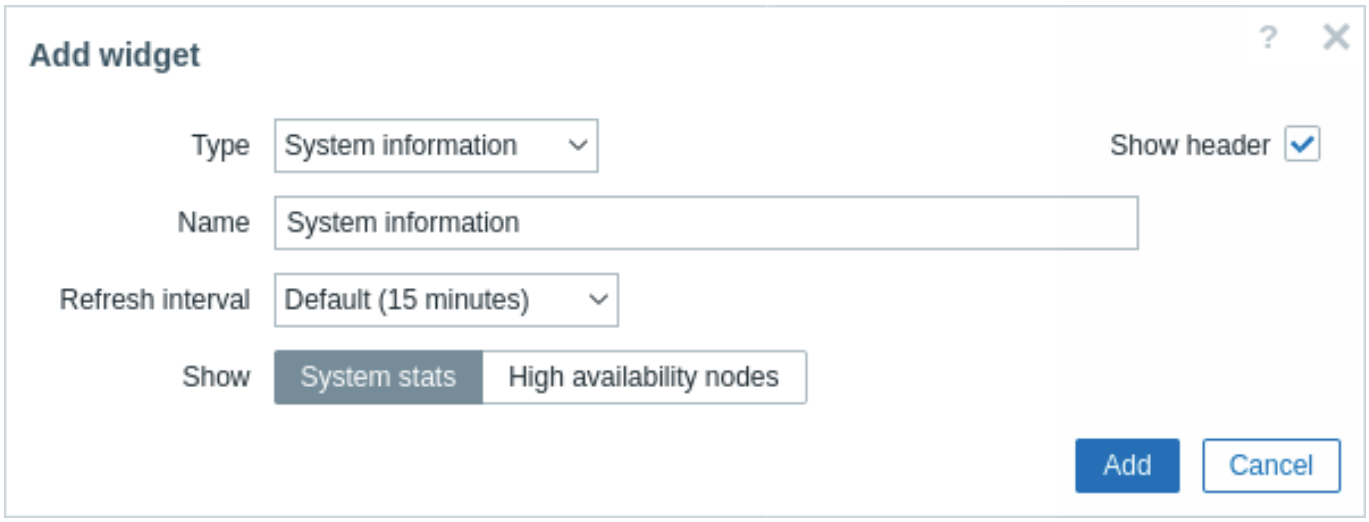
Le widget *Informations système* affiche un résumé des principales données du serveur Zabbix et du système, ou des détails sur les nœuds de haute disponibilité.



Il affiche les mêmes données que *Rapports* > *Informations système*.

Configuration

Pour configurer, sélectionnez *Informations système* comme type :



En plus des paramètres communs à tous les widgets, vous pouvez définir les options spécifiques suivantes :

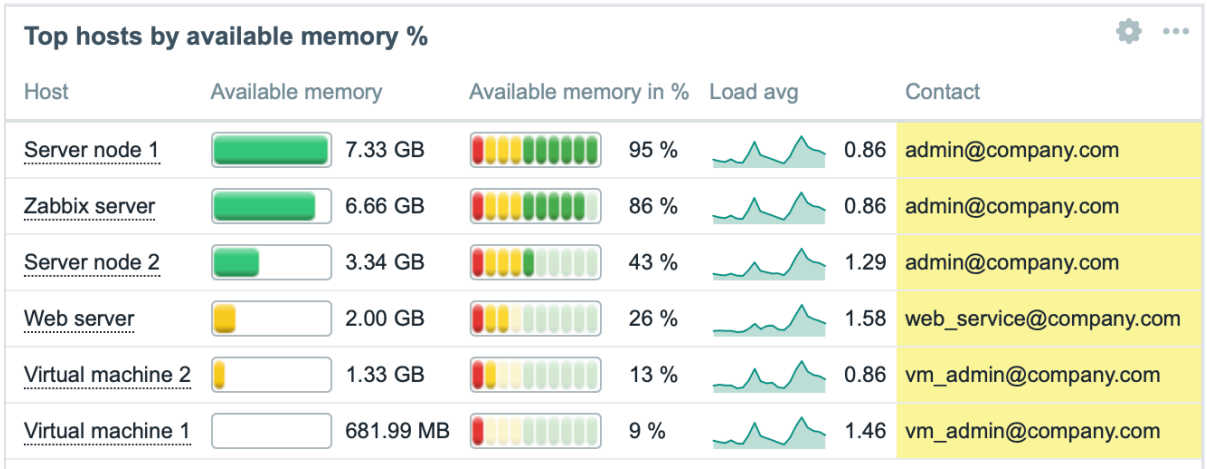
Afficher	Sélectionnez ce qu'il faut afficher : **Statistiques système** - affiche un résumé des principales données du serveur Zabbix et du système ; **Nœuds de haute disponibilité** - affiche l'état des nœuds de haute disponibilité (si le cluster de haute disponibilité est activé).

Afficher les détails de la vérification des mises à jour logicielles	Cochez la case pour afficher les détails de la vérification des mises à jour logicielles de Zabbix. Cette option n'est disponible que si la vérification des mises à jour logicielles est activée dans la configuration du serveur Zabbix et que « Statistiques système » est sélectionné dans le champ *Afficher*.

28 Hôtes principaux

Vue d'ensemble

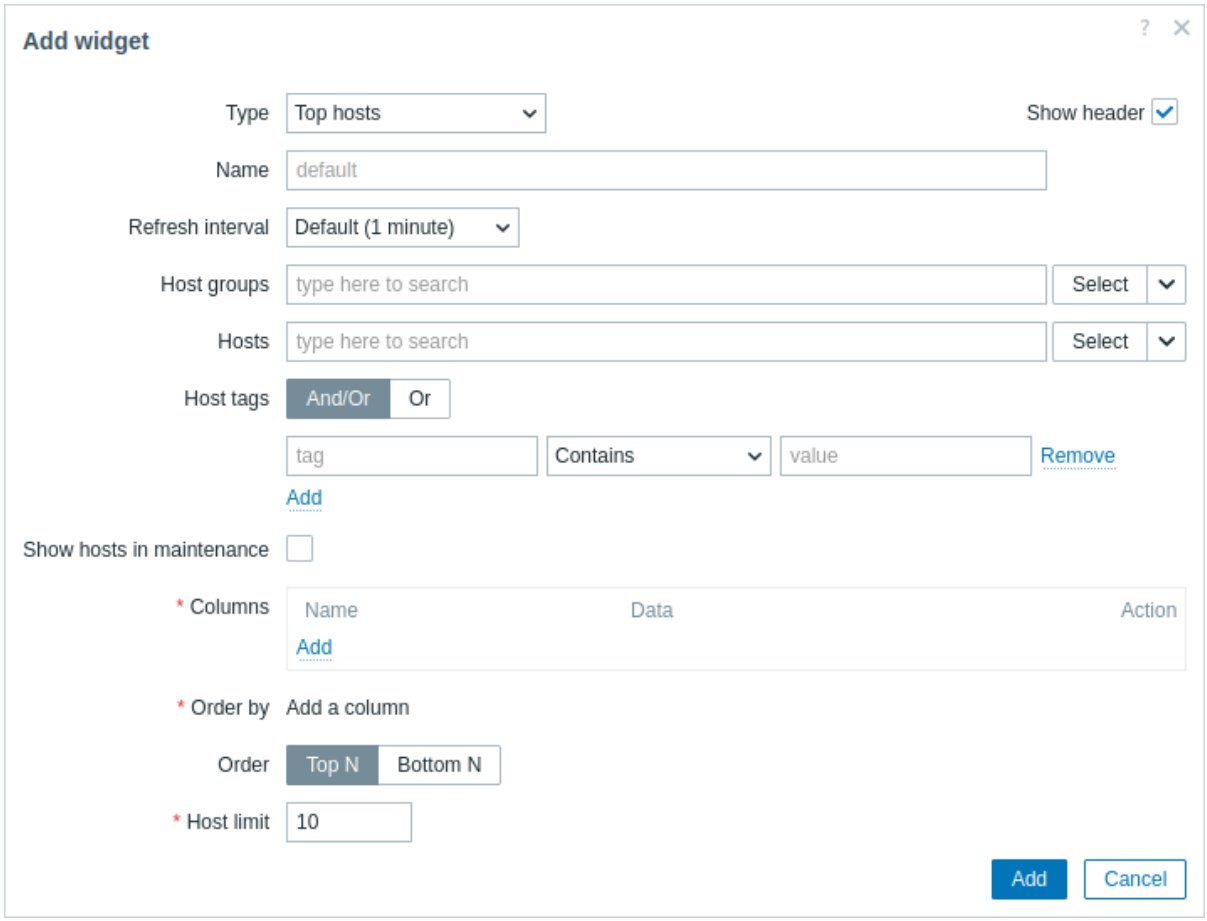
Le widget *Top hosts* affiche les valeurs les plus élevées/les plus basses pour les éléments sélectionnés de plusieurs hôtes.



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 :

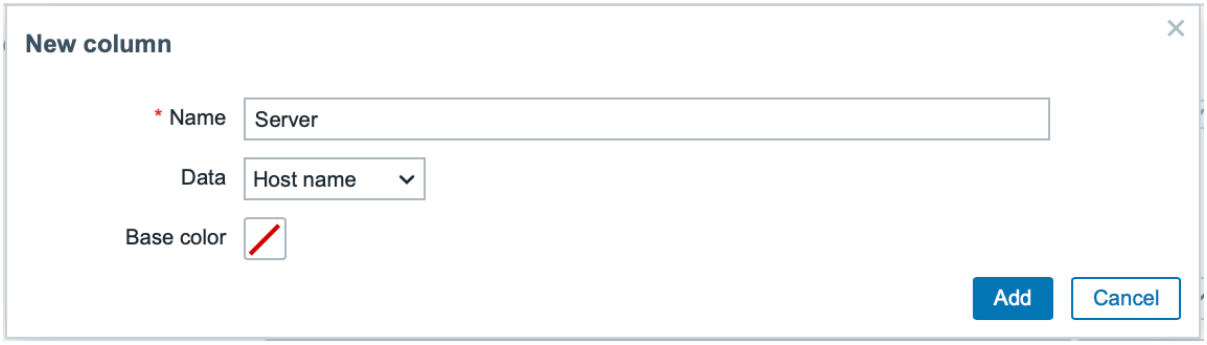
Groupes d'hôtes	Sélectionnez les groupes d'hôtes à afficher dans le widget. Vous pouvez également sélectionner un widget compatible comme source de données pour les groupes d'hôtes. Ce champ prend en charge l'autocomplétion ; ainsi, lorsque vous commencez à saisir le nom d'un groupe, une liste déroulante 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.

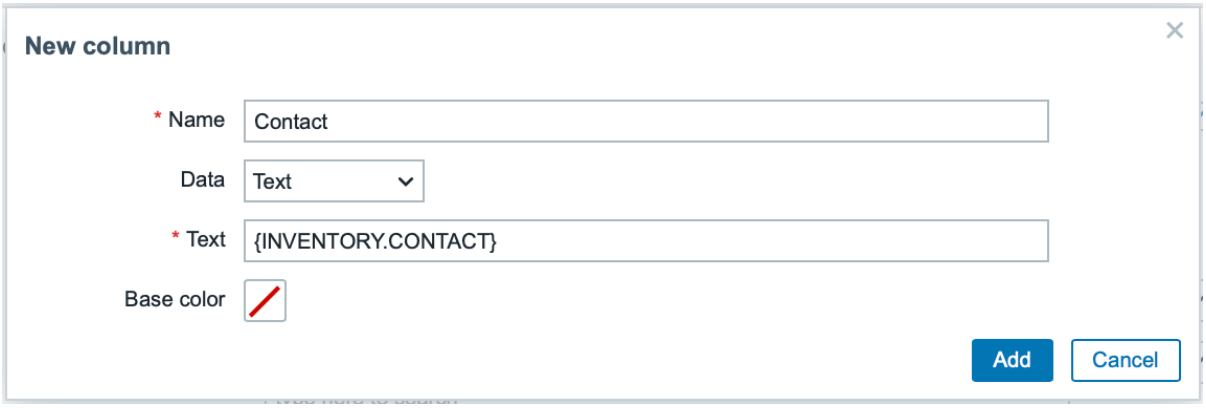


Paramètres pris en charge :

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.

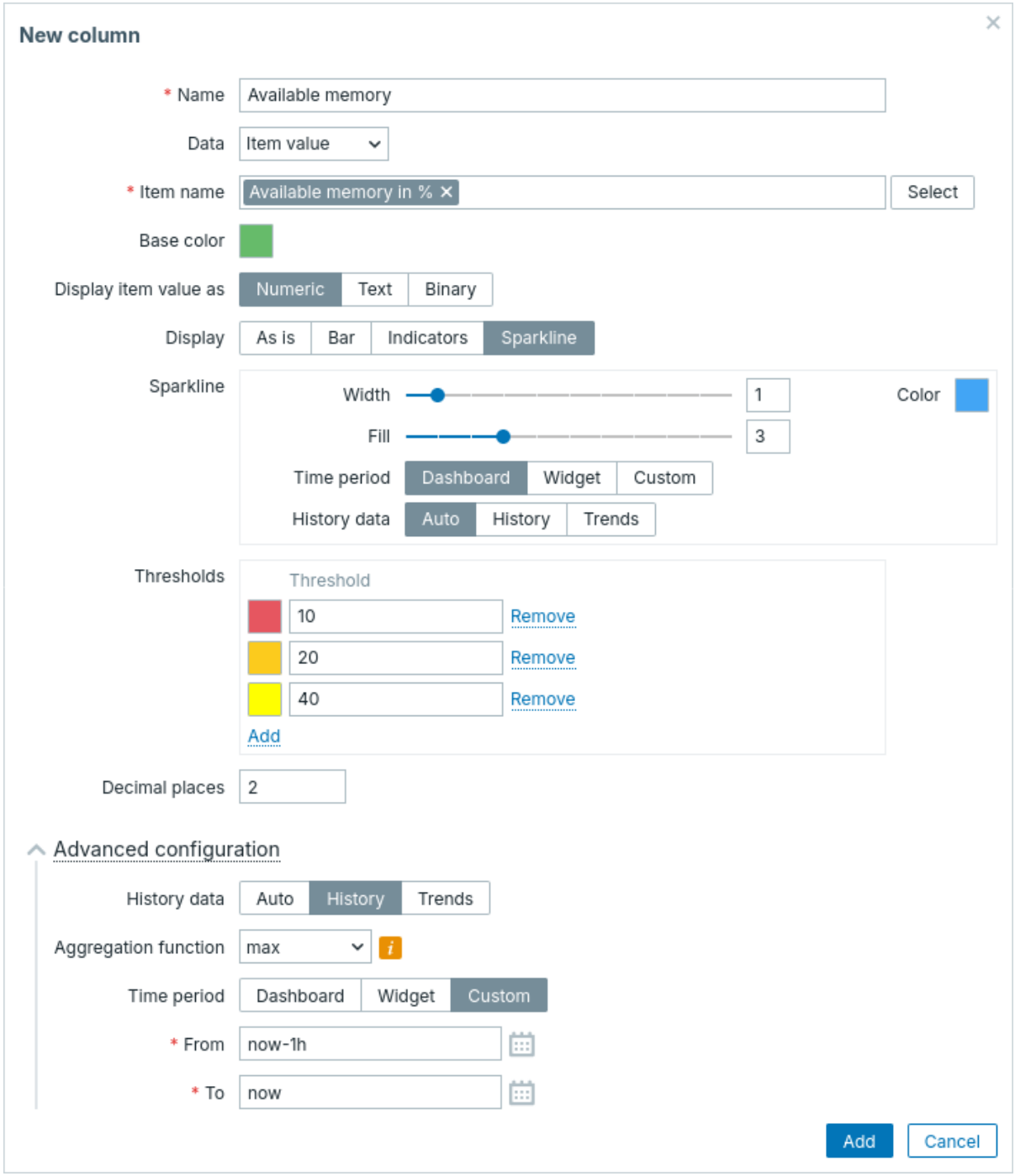


Paramètres pris en charge :

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 d'élément

La colonne de valeur d'élément est utilisée pour 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 faire correspondre 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 est la première par ordre alphabétique (par exemple, *CPU utilization* avec la clé `proc.cpu.util` au lieu de `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/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 quels paramètres supplémentaires seront disponibles. Reportez-vous à la liste des paramètres de chaque format ci-dessous.
Advanced configuration	

History data	Prendre les données depuis l'historique ou les tendances : **Auto** - sélection automatique ; **History** - prendre les données de l'historique ; **Trends** - prendre les données des tendances. Ce paramètre n'est disponible que pour les valeurs d'élément numériques. Pour les valeurs texte et binaires, les données seront toujours prises depuis l'historique.
Aggregation function	Spécifiez la fonction d'agrégation à utiliser : **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 ; **not used** - afficher la valeur la plus récente (sans agrégation). L'agrégation permet d'afficher une valeur agrégée pour l'intervalle choisi (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.
Time period	Spécifiez la période à utiliser pour l'agrégation des valeurs. 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. 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 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".

Paramètres de valeur d'élément numérique :

Display	Définissez comment la valeur doit être affichée : **As is** - comme texte normal ; **Bar** - comme barre pleine remplie de couleur ; **Indicators** - comme barre segmentée remplie de couleur ; **Sparkline** - mini graphique linéaire.
Thresholds	Spécifiez les valeurs de seuil 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.
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 des valeurs à inclure dans le graphique sparkline. 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".
History data	Prendre les données depuis l'historique ou les tendances : **Auto** - sélection automatique ; **History** - prendre les données de l'historique ; **Trends** - prendre les données des tendances.

Paramètres de valeur d'élément texte :

Highlights	Spécifiez les expressions régulières dont la correspondance doit entraîner un changement de la couleur d'arrière-plan/de remplissage.

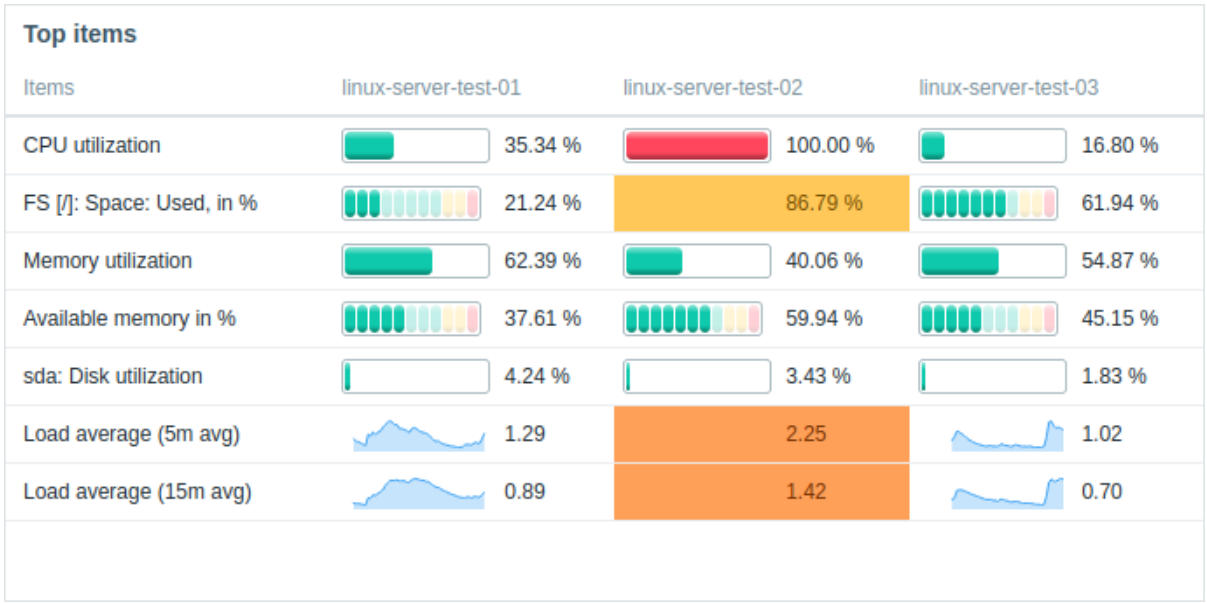
Paramètres de valeur d'élément binaire :

Show thumbnail	Indiquez s'il faut créer et afficher une miniature pour l'image contenant des données binaires ou afficher un lien hypertexte *Show* menant à l'image en taille réelle dans la colonne de valeur.

29 Éléments principaux

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.

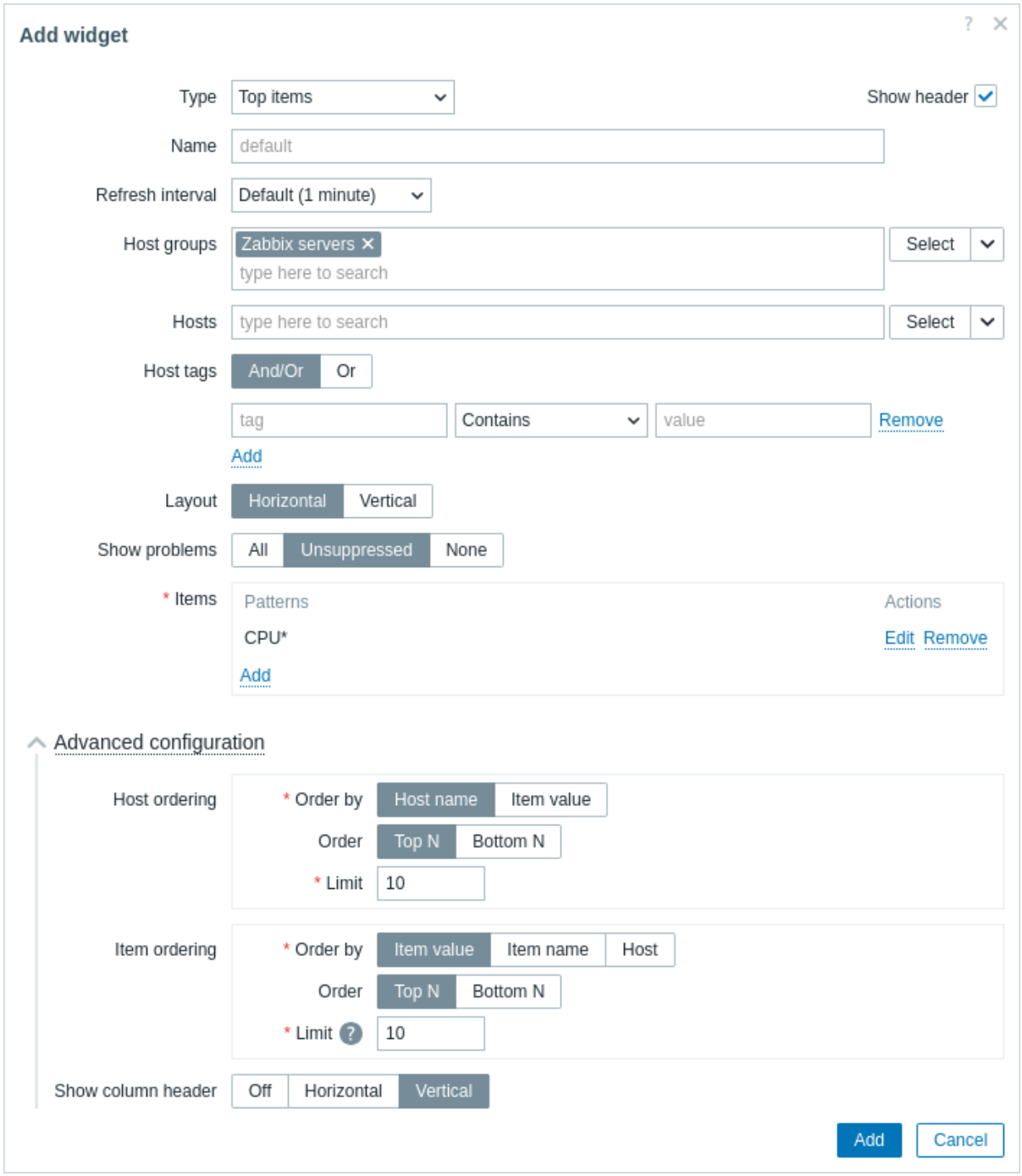


Par défaut, le widget affiche les valeurs des dernières 24 heures. Cette période peut être modifiée dans *Administration* > *General* > *GUI* (*Max history display period*).

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 :

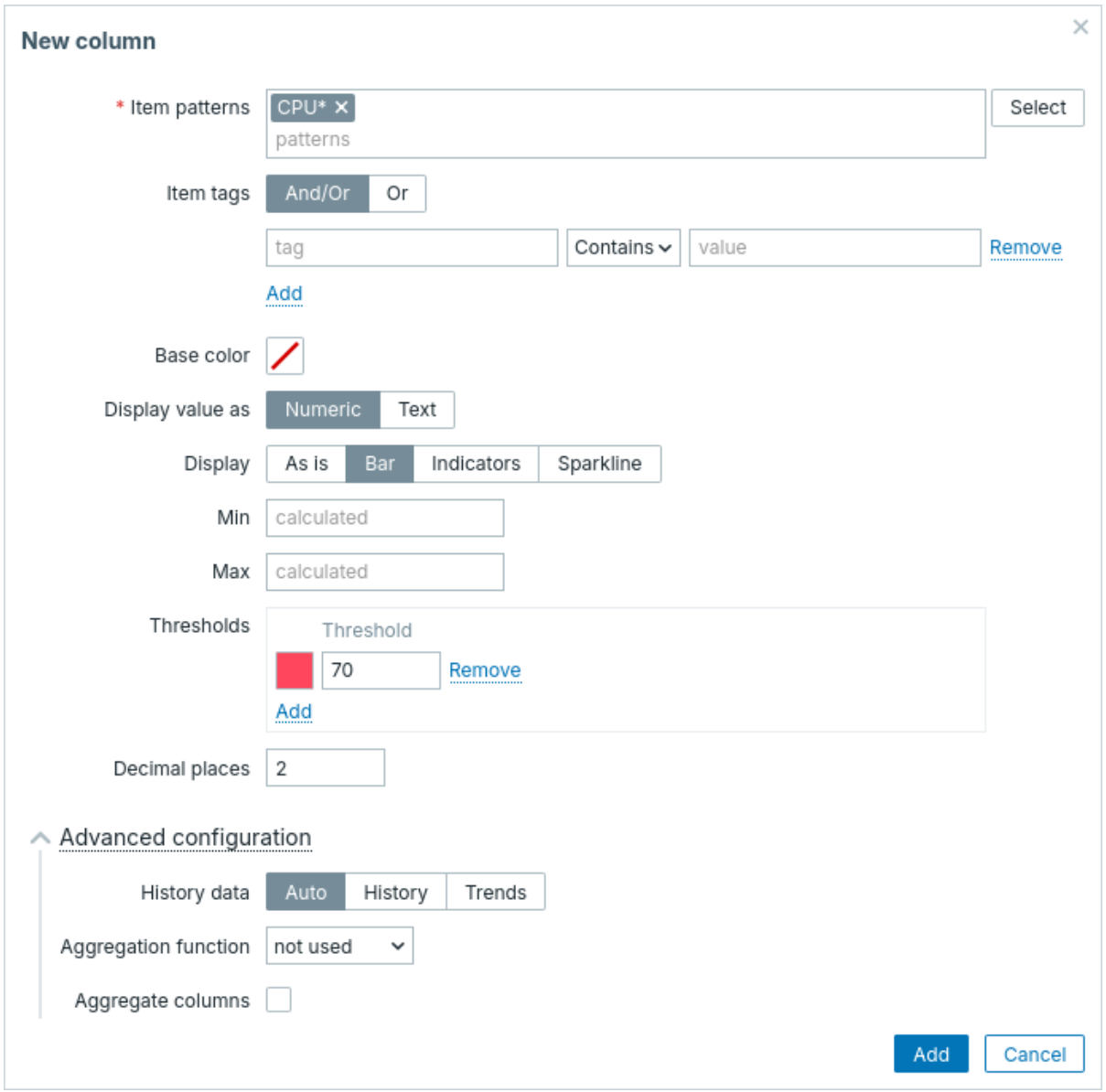
Host groups	Sélectionnez des groupes d'hôtes. Vous pouvez également sélectionner un widget compatible comme source de données pour les groupes d'hôtes. Ce champ prend en charge l'autocomplétion ; en commençant à saisir le nom d'un groupe, une liste déroulante des groupes correspondants sera proposée. 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 ; en commençant à saisir le nom d'un hôte, une liste déroulante des hôtes correspondants sera proposée. 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 la quantité de données d'hôte 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 : **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.
Layout	Sélectionnez l'option de disposition : **Horizontal** - les noms d'hôte seront affichés à gauche ; **Vertical** - les noms d'hôte seront affichés en haut.
Show problems	Filtrez les problèmes à afficher en fonction de leur état (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 sévérité du problème, qui peut être personnalisée dans *Administration > General > Trigger displaying options*. Les sévérités des problèmes peuvent être modifiées lors de la mise à jour des problèmes.
Items	Ajoutez des modèles d'élément ou des éléments spécifiques à afficher (voir Configuration des colonnes).
Advanced configuration	
Host ordering	Sélectionnez les options de tri pour la colonne/ligne des hôtes.
Order by	Triez les hôtes par : **Host name** - les hôtes seront triés par nom d'hôte ; **Item value** - les hôtes seront triés selon la valeur des éléments sélectionnés.
Order	Sélectionnez s'il faut 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*.
Limit	Saisissez la limite des hôtes affichables (plage 1-1000 ; valeur par défaut 10). Cette valeur deviendra la valeur de N dans le champ *Order*.
Item ordering	Sélectionnez les options de tri pour la colonne/ligne des éléments.

Order by	Triez les éléments par : **Item value** - les éléments seront triés selon leur valeur ; **Item name** - les éléments seront triés par nom d'élément ; **Host** - les éléments seront triés selon le modèle d'hôte sélectionné.
Order	Sélectionnez s'il faut 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*.
Limit	Saisissez la limite des éléments affichables (plage 1-1000 ; valeur par défaut 10). Cette valeur deviendra la valeur de N dans le champ *Order*.
Show column header	Sélectionnez les options d'affichage de l'en-tête de colonne : **Off** - ne pas afficher l'en-tête de colonne ; **Horizontal** - afficher le texte horizontalement dans l'en-tête ; **Vertical** - afficher 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 comment la valeur doit être affichée : **Tel quel** - comme du texte normal ; **Barre** - sous forme de barre pleine colorée ; **Indicateurs** - sous forme de barre segmentée colorée ; **Sparkline** - mini graphique linéaire.
Min	Valeur minimale pour l'affichage en barre/indicateurs.
Max	Valeur maximale pour l'affichage en barre/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.
Voir *Configuration avancée* pour la description des champs liés à la période et à la sélection des données d'historique.	
Seuils	Spécifiez les valeurs seuils auxquelles 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 combien de décimales doivent être affichées avec la valeur.

Données d'historique	Prendre les données depuis l'historique ou les tendances : **Auto** - sélection automatique ; **Historique** - prendre les données d'historique ; **Tendances** - prendre les données de tendance. Ce paramètre est disponible uniquement pour les valeurs numériques d'élément. Pour les valeurs texte, les données seront toujours prises depuis l'historique.
Fonction d'agrégation	Indiquez la fonction d'agrégation à utiliser : **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 ; **not used** - afficher la valeur la plus récente (sans agrégation). L'agrégation permet d'afficher une valeur agrégée pour l'intervalle choisi (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, 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 défini sur "not used".
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".
Agréger les colonnes	Si cette option est cochée, tous les éléments correspondant aux modèles d'élément de la colonne sont regroupés dans une seule colonne/ligne combinée. Les valeurs des éléments correspondants seront agrégées à l'aide de la *Fonction d'agrégation de colonne* sélectionnée et affichées sous le *Nom de colonne combinée*. Par défaut : désactivé (non coché). Ce paramètre est affiché uniquement lorsque *Afficher la valeur comme* est défini sur **Numérique**.
Fonction d'agrégation de colonne	Choisissez comment les valeurs des modèles d'élément regroupés sont combinées. Options disponibles : **min**, **max**, **avg**, **count**, **sum** (par défaut). Ce paramètre est affiché uniquement lorsque *Agréger les colonnes* est coché.
Nom de colonne combinée	Saisissez le nom d'affichage de la colonne agrégée. Ce champ est obligatoire lorsque *Agréger les colonnes* est coché et devient l'en-tête de colonne pour les valeurs combinées.

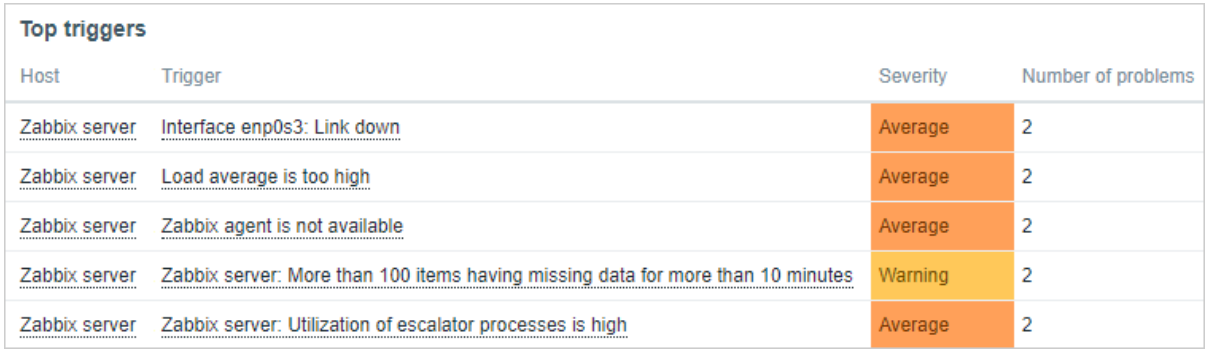
Paramètres de colonne pour les valeurs texte :

Mises en évidence	Indiquez les expressions régulières dont la correspondance doit entraîner le changement de la couleur d'arrière-plan/de remplissage.

30 principaux déclencheurs

Vue d'ensemble

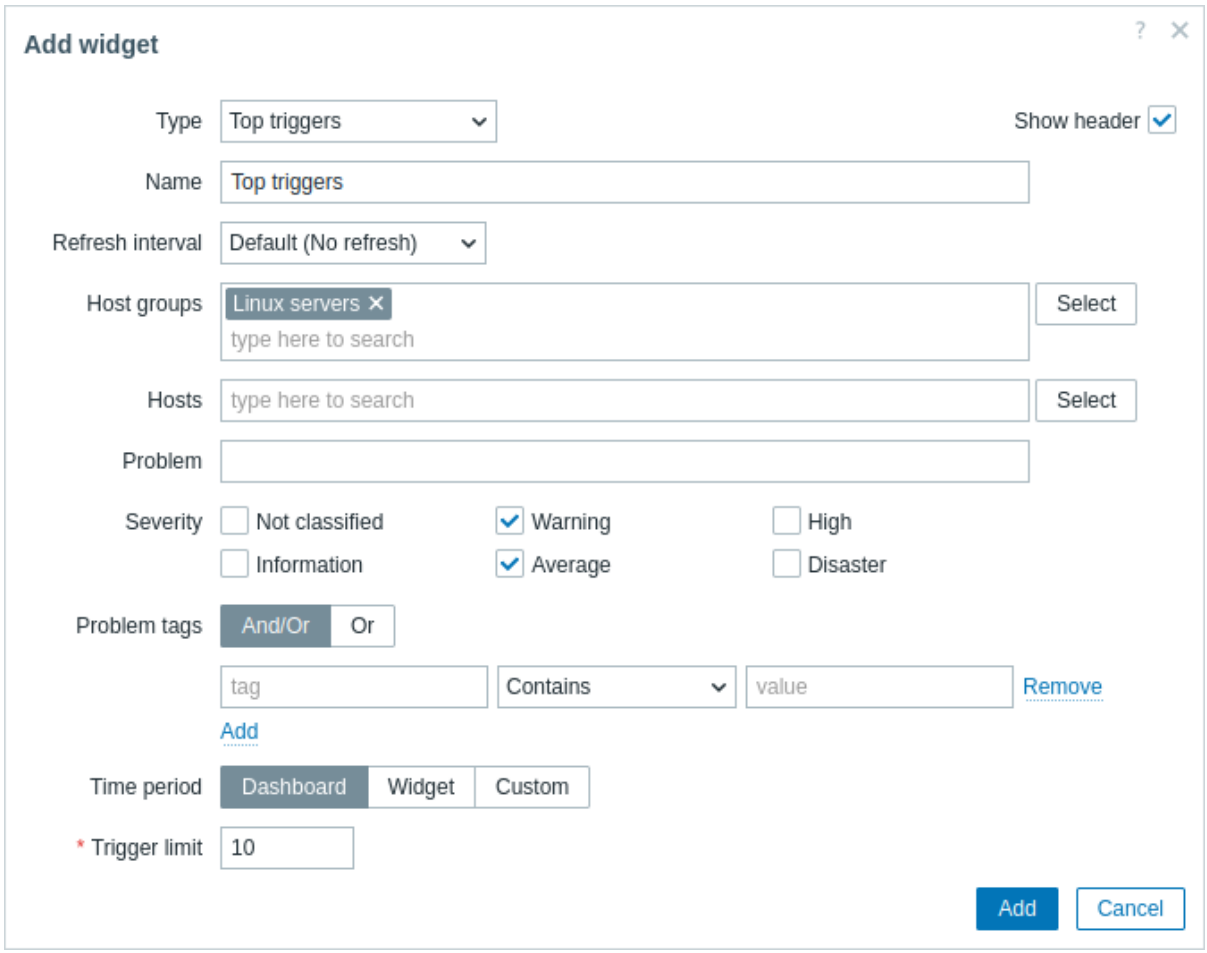
Le widget *Top triggers* affiche les déclencheurs ayant le plus grand nombre de problèmes.



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.

31 Aperçu des déclencheurs

Aperçu

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.



Le widget utilise plusieurs éléments visuels pour transmettre les détails des déclencheurs :

- Les couleurs des blocs pour les déclencheurs en état de PROBLÈME correspondent aux sévérités des problèmes.
- Les blocs clignotants indiquent des changements récents de l'état des déclencheurs (au cours des 2 dernières minutes).
- Les flèches grises indiquent des déclencheurs avec des dépendances (survolez-les pour voir les détails).
- Les icônes de case à cocher indiquent les problèmes acquittés ; tous les problèmes actifs ou résolus du déclencheur doivent être acquittés pour que cette icône apparaisse.

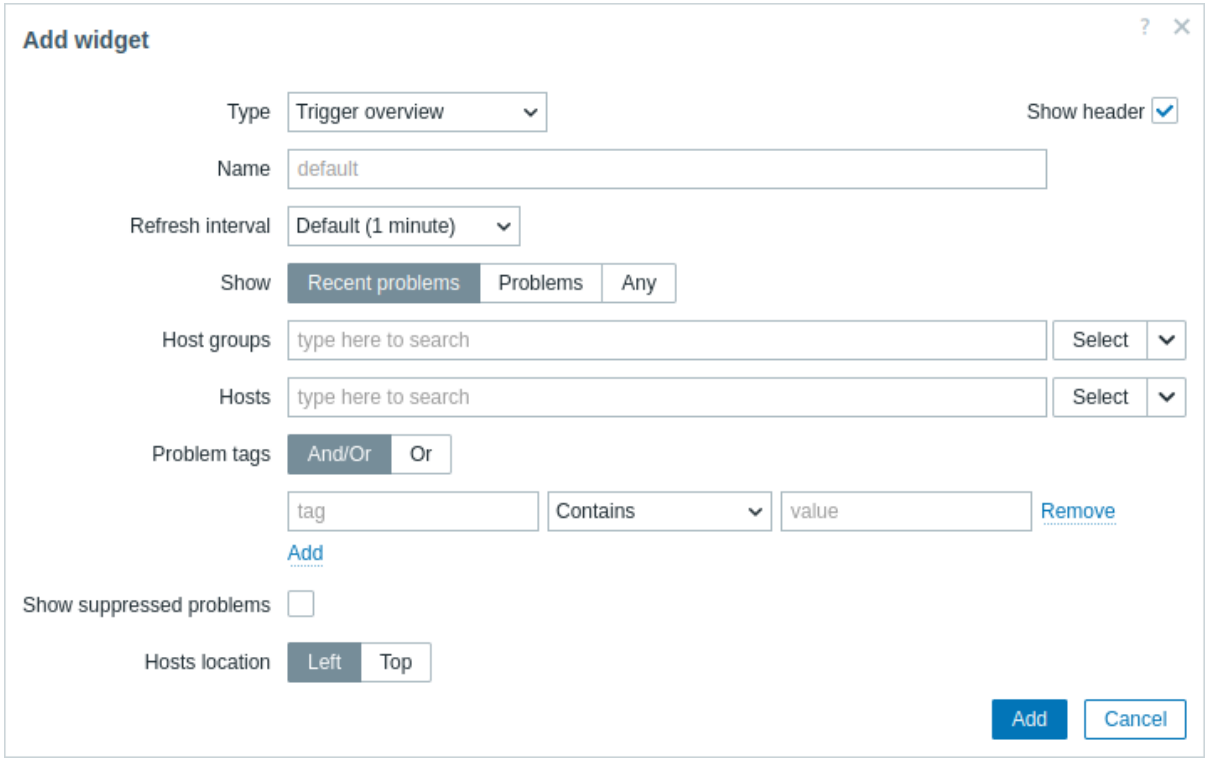
Note:
Les couleurs des sévérités des déclencheurs peuvent être ajustées dans *Administration* > *General*. Les sévérités des déclencheurs peuvent être modifiées lors de la mise à jour des problèmes.

Cliquer sur un bloc fournit des liens vers les informations liées au déclencheur (problèmes, configuration du déclencheur, etc.).

Par défaut, jusqu'à 50 enregistrements sont affichés (configurable dans *Administration* > *General* > *GUI* sous *Max number of columns and rows in overview tables*). S'il existe davantage de déclencheurs, un message vous invite à affiner les critères de filtrage. Cette limite est appliquée avant tout filtrage supplémentaire (par exemple, par balises).

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 :

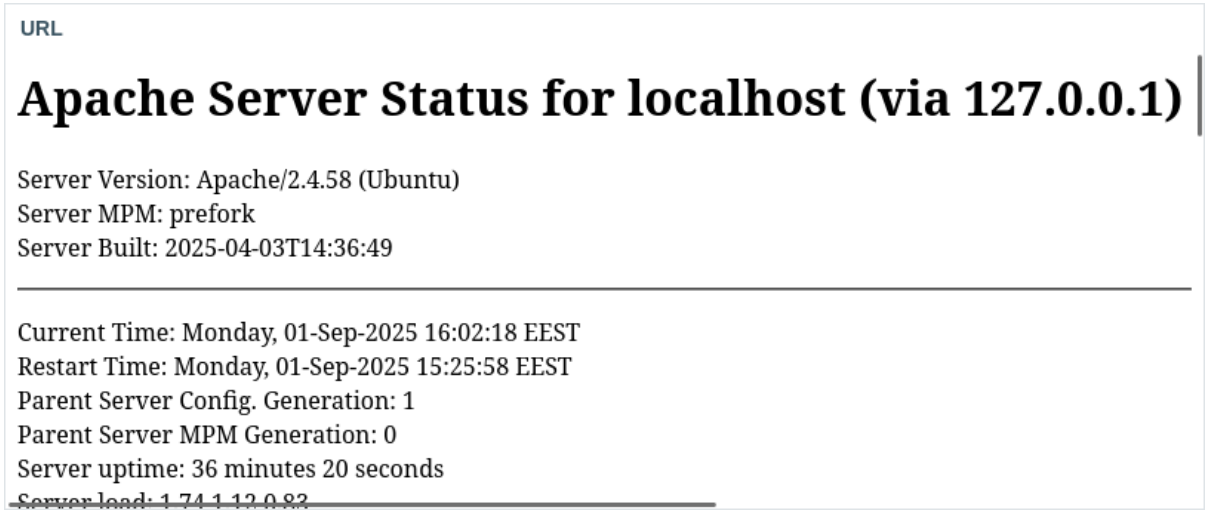
Show	Filtrer les déclencheurs par état du déclencheur : **Recent problems** - *(par défaut)* afficher les déclencheurs qui ont récemment été ou sont encore à l'état PROBLEM (résolus et non résolus) ; **Problems** - afficher les déclencheurs qui sont à l'état PROBLEM (non résolus) ; **Any** - afficher 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.

32 URL

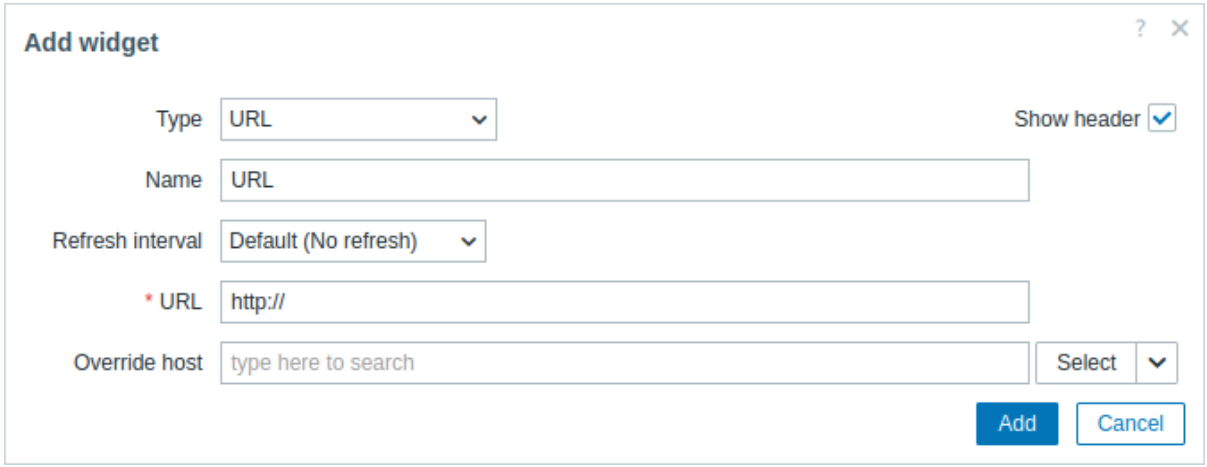
Aperçu

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 :

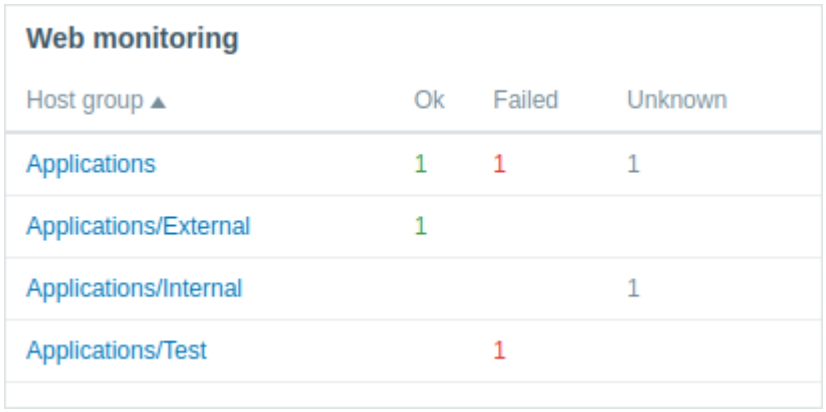
URL	Saisissez l'URL à afficher (jusqu'à 2048 caractères). Les URL externes doivent commencer par `http://` ou `https://`. Les URL internes prennent en charge les chemins relatifs (par exemple, `zabbix.php?action=report.status`). Les macros {HOST.*} sont prises en charge.
Override host	Sélectionnez un 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 tableau de bord de modèle.

Attention:
Les navigateurs peuvent ne pas charger une page HTTP configurée dans le widget si l'interface Zabbix est accessible via HTTPS. De plus, il peut être nécessaire de modifier l'en-tête de réponse HTTP X-Frame-Options pour afficher du contenu externe à l'interface Zabbix.

33 Surveillance Web

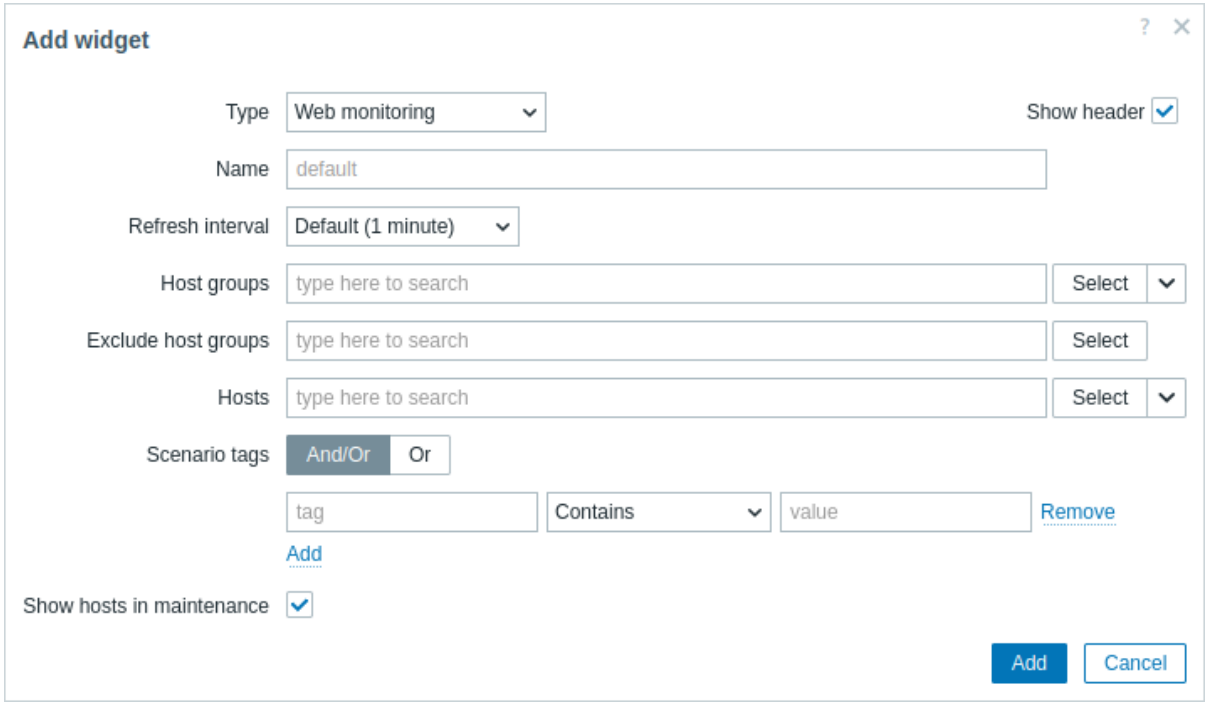
Aperçu

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.

2 Supervision

Aperçu

Le menu Surveillance regroupe les aspects concernant l'affichage des données. Quelques soient les informations pour lesquelles Zabbix est configuré (collecte, visualisation et utilisation), elles seront affichées dans les différentes sections du menu Surveillance.

Boutons du mode d'affichage

Les boutons suivants, situés dans le coin supérieur droit, sont communs à chaque section :

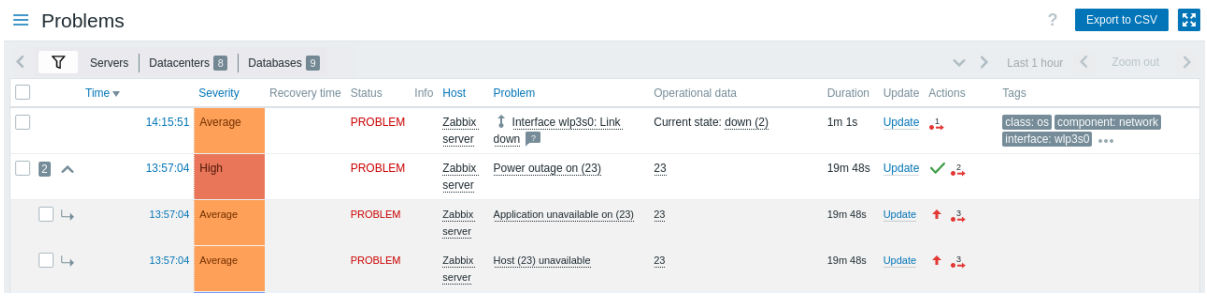
	Afficher la page en mode kiosque. Dans ce mode, seul le contenu de la page est affiché. Pour quitter le mode kiosque, déplacez le curseur de la souris jusqu'à ce que le bouton de sortie  apparaisse, puis cliquez dessus. Vous reviendrez au mode normal.

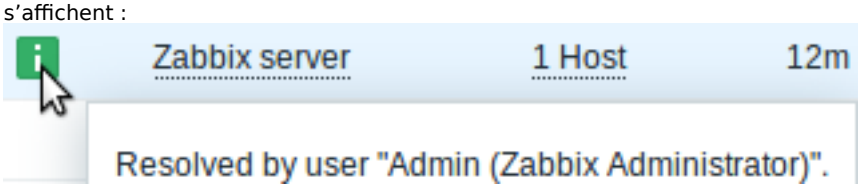
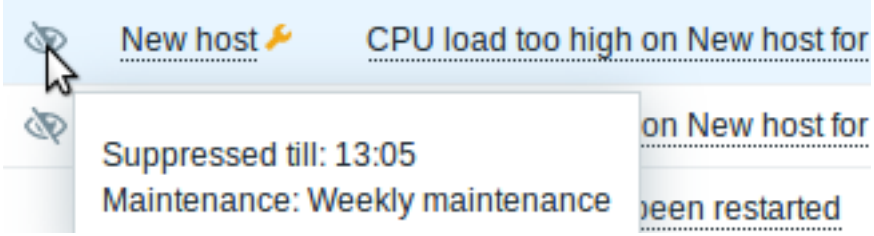
1 Problèmes

Aperçu

Dans *Surveillance* > *Problèmes*, vous pouvez voir les problèmes que vous avez actuellement. Les problèmes sont les déclencheurs qui sont à l'état « Problème ».

Par défaut, tous les nouveaux problèmes sont classés comme problèmes de cause. Il est possible de reclasser manuellement certains problèmes comme problème symptomatique d'un problème de cause. Pour plus de détails, voir événements de cause et de symptôme.



Colonne	Description
Case à cocher	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 symptomatiques pour le problème de cause ;  - développer pour afficher les événements symptomatiques ;  - réduire pour masquer les événements symptomatiques ;  - il s'agit d'un événement symptomatique.
Heure	L'heure de début du problème est affichée.
Sévérité	La sévérité du problème est affichée. La sévérité du problème est initialement basée sur la sévérité du déclencheur de problème sous-jacent, cependant, une fois l'événement survenu, elle peut être mise à jour à l'aide de l'écran *Mettre à jour le problème* screen. La couleur de la sévérité du problème est utilisée comme arrière-plan de cellule pendant la durée du problème.
Heure de résolution	L'heure de résolution du problème est affichée.
Statut	Le statut du problème est affiché : **Problème** - problème non résolu **Résolu** - 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* > *Général* > *Options d'affichage des déclencheurs*.
Info	Une icône d'information verte est affichée si un problème est fermé par corrélation globale ou manuellement lors de la mise à jour du problème. En passant la souris sur l'icône, plus de détails s'affichent :  L'icône suivante est affichée si un problème supprimé est affiché (voir l'option *Afficher les problèmes supprimés* dans le filtre). En passant la souris sur l'icône, plus de détails s'affichent : 
Hôte	L'hôte du problème est affiché. Cliquer sur le nom de l'hôte ouvre le menu de l'hôte.
Problème	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 survient et les valeurs résolues ne sont plus mises à jour ensuite. *Notez* qu'il est possible d'ajouter au nom du problème des données opérationnelles affichant certaines des dernières valeurs d'élément. Cliquer sur le nom du problème ouvre le menu de l'événement. Survoler l'icône  après le nom du problème affiche la description du déclencheur (pour les problèmes qui en ont une).
Données opérationnelles	Les données opérationnelles sont affichées et contiennent les dernières valeurs d'élément. Les données opérationnelles peuvent être une combinaison de texte et de macros de valeur d'élément si elles sont configurées au niveau du déclencheur. Si aucune donnée opérationnelle n'est configurée au niveau du déclencheur, les dernières valeurs de tous les éléments de l'expression sont affichées. Cette colonne n'est affichée que si *Séparément* est sélectionné pour *Afficher les données opérationnelles* dans le filtre.
Durée	La durée du problème est affichée. Voir aussi : Durée négative du problème

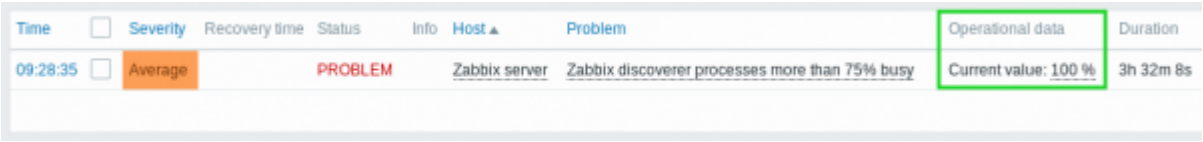
Colonne	Description
Mise à jour	Cliquez sur le lien *Mise à jour* pour accéder à l'écran de mise à jour du problème, où diverses actions peuvent être effectuées sur le problème, notamment l'ajout de commentaires et l'acquittement du problème.
Actions	L'historique des activités concernant le problème est affiché à l'aide d'icônes symboliques :  - le problème a été acquitté. Cette icône est toujours affichée en premier.  - des commentaires ont été ajoutés. Le nombre de commentaires est également affiché.  - la sévérité du problème a été augmentée (par ex. Information > Avertissement).  - la sévérité du problème a été diminuée (par ex. Avertissement > Information).  - la sévérité du problème a été modifiée, mais est revenue à son niveau d'origine (par ex. Avertissement > Information > Avertissement).  - le problème a été supprimé.  - le problème n'est plus supprimé.  - 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. En passant la souris sur les icônes, des fenêtres contextuelles avec des détails sur l'activité sont affichées. Voir affichage des détails pour en savoir plus sur les icônes utilisées dans la fenêtre contextuelle pour les actions effectuées.
Balises	Les balises sont affichées (le cas échéant). En outre, les balises d'un système de ticketing externe peuvent également être affichées (voir l'option *Traiter les balises* lors de la configuration des webhooks).

Données opérationnelles des problèmes

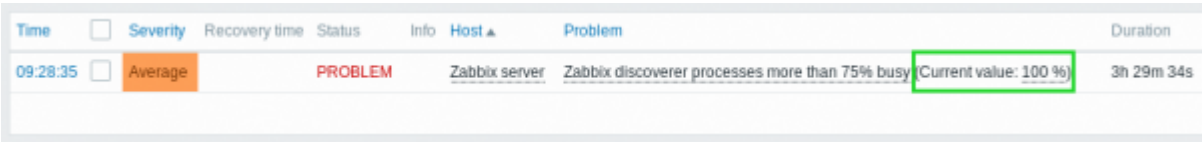
Il est possible d'afficher des données opérationnelles pour les problèmes en cours, c'est-à-dire les dernières valeurs des éléments, par opposition aux valeurs des éléments au moment du problème.

L'affichage des données opérationnelles peut être configuré dans le filtre de *Monitoring* > *Problems* ou dans la configuration du widget de tableau de bord correspondant, en sélectionnant l'une des trois options :

- *None* - aucune donnée opérationnelle n'est affichée
- *Separately* - les données opérationnelles sont affichées dans une colonne distincte



- *With problem name* - les données opérationnelles sont ajoutées au nom du problème, entre parenthèses. Les données opérationnelles ne sont ajoutées au nom du problème que si le champ *Operational data* n'est pas vide dans la configuration du déclencheur.



Le contenu des données opérationnelles peut être configuré pour chaque déclencheur, dans le champ *Operational data*. Ce champ accepte une chaîne arbitraire avec des macros, notamment la macro `{ITEM.LASTVALUE<1-9>}`.

`{ITEM.LASTVALUE<1-9>}` dans ce champ sera toujours résolue en dernières valeurs des éléments dans l'expression du déclencheur. `{ITEM.VALUE<1-9>}` dans ce champ sera résolue en valeurs des éléments au moment du changement d'état du

déclencheur (c'est-à-dire lors du passage à **Problem**, du passage à **OK**, de la fermeture manuelle par l'utilisateur ou de la fermeture par corrélation).

Notez que la fermeture manuelle du problème ne produit pas de nouvelle valeur, de sorte que la valeur résolue de {ITEM.LASTVALUE<1-9>} ou {ITEM.VALUE<1-9>} affichera toujours la valeur du moment du problème.

{ITEM.LASTVALUE<1-9>} ou {ITEM.VALUE<1-9>} sera résolue en *UNKNOWN* si la dernière valeur d'historique a été collectée il y a plus longtemps que la durée *Max history display period* (voir *Administration > General*).

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 situés sous la liste offrent quelques 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 devant les problèmes concernés, puis cliquez sur le bouton *Mise à jour en masse*.

Boutons

Le bouton à droite offre l'option suivante :

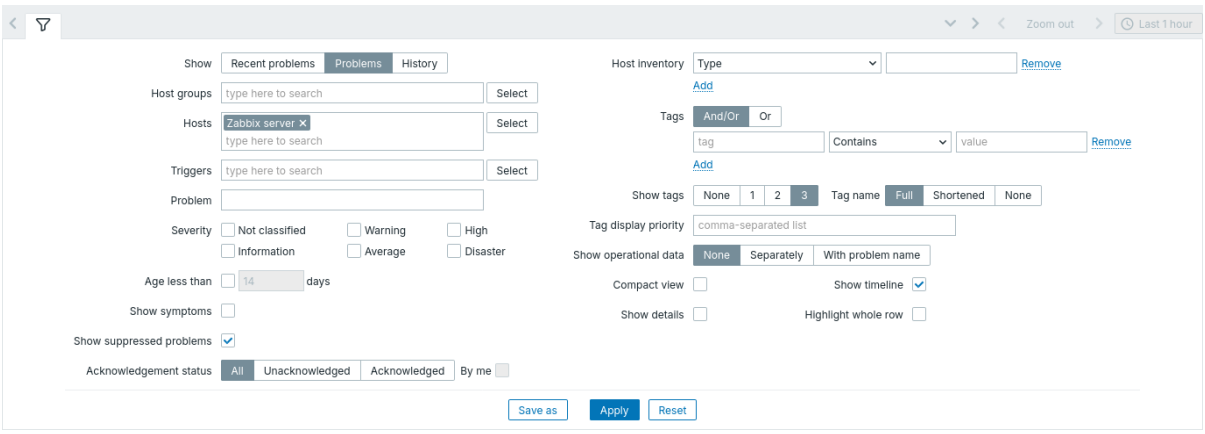
Export to CSV	Exporter le contenu de toutes les pages vers un fichier CSV.

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 n'afficher que les problèmes 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. Les paramètres de filtre favoris peuvent être enregistrés sous forme d'onglets, puis rapidement accessibles en cliquant sur les onglets au-dessus du filtre.



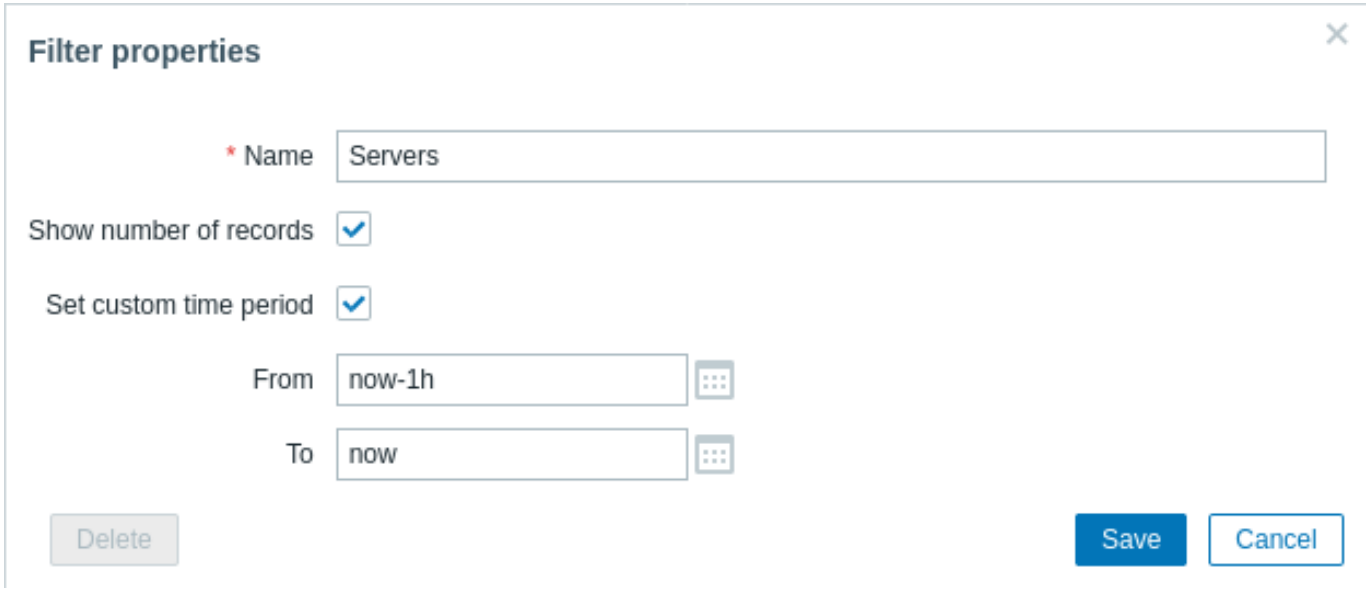
Paramètre	Description
Afficher	Filtrer par statut du problème : **Problèmes récents** - les problèmes non résolus et récemment résolus sont affichés (par défaut) **Problèmes** - les problèmes non résolus sont affichés **Historique** - l'historique de tous les événements est affiché
Groupes d'hôtes	Filtrer par un ou plusieurs groupes d'hôtes. La spécification d'un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués.
Hôtes	Filtrer par un ou plusieurs hôtes.
Déclencheurs	Filtrer par un ou plusieurs déclencheurs.
Problème	Filtrer par nom de problème.
Sévérité	Filtrer par sévérité du déclencheur (problème).
Âge inférieur à	Filtrer selon l'ancienneté du problème.
Afficher les symptômes	Cochez la case pour afficher sur sa propre ligne les problèmes classés comme symptômes.
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 ou d'une suppression de problème individuelle.
Statut d'acquittement	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.
Inventaire de l'hôte	Filtrer par type et valeur d'inventaire.
Tags	Filtrer par nom et valeur de tag d'événement. Il est possible d'inclure comme 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 : **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) Il existe deux types de calcul 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 Lors du filtrage, les tags spécifiés ici seront affichés en premier avec le problème, sauf si cela est remplacé par la liste *Priorité d'affichage des tags* (voir ci-dessous).
Afficher les tags	Sélectionnez le nombre de tags affichés : **Aucun** - aucune colonne *Tags* dans *Surveillance* > *Problèmes* **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.
Nom du tag	Sélectionnez le mode d'affichage du nom du tag : **Complet** - les noms et valeurs de tag sont affichés en entier **Abrégé** - les noms de tag sont abrégés à 3 caractères ; les valeurs de tag sont affichées en entier **Aucun** - seules les valeurs de tag sont affichées ; aucun nom
Priorité d'affichage des tags	Saisissez la priorité d'affichage des tags pour un problème sous la forme d'une liste de tags séparés par des virgules (par exemple : `customer,scope,component`). Seuls les noms de tag doivent être utilisés, sans valeurs. Les tags de cette liste seront toujours affichés en premier, remplaçant l'ordre naturel alphabétique.
Afficher les données opérationnelles	Sélectionnez le mode d'affichage des données opérationnelles : **Aucune** - aucune donnée opérationnelle n'est affichée **Séparément** - les données opérationnelles sont affichées dans une colonne distincte **Avec le nom du problème** - ajouter les données opérationnelles au nom du problème, en utilisant des parenthèses pour les données opérationnelles
Vue compacte	Cochez la case pour activer la vue compacte.
Afficher les détails	Cochez la case pour afficher les expressions de déclencheur sous-jacentes des problèmes. Désactivé si la case *Vue compacte* est cochée.
Afficher la chronologie	Cochez la case pour afficher la chronologie visuelle et le regroupement. Désactivé si la case *Vue compacte* est cochée.

Paramètre	Description
Surligner toute la ligne	Cochez la case pour surligner toute la ligne des problèmes non résolus. La couleur de sévérité du problème est utilisée pour le surlignage. *Surligner toute la ligne* n'est pas disponible dans les thèmes à contraste élevé.

Onglets pour les filtres favoris

Les ensembles de paramètres de filtre fréquemment utilisés peuvent être enregistrés dans des onglets.

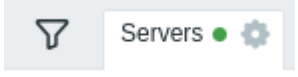
Pour enregistrer un nouvel ensemble de paramètres de filtre, ouvrez l'onglet principal et configurez les paramètres du filtre, puis appuyez sur le bouton *Enregistrer sous*. Dans une nouvelle fenêtre contextuelle, définissez les *Propriétés du filtre*.



Paramètre	Description
Nom	Nom du filtre affiché dans la liste des onglets.
Afficher le nombre d'enregistrements	Cochez cette option si vous souhaitez que le nombre de problèmes soit affiché à côté du nom de l'onglet.
Remplacer le sélecteur de période	Cochez cette option pour définir une période par défaut spécifique pour cet ensemble de filtres. Si cette option est définie, vous ne pourrez modifier la période pour cet onglet qu'en mettant à jour les paramètres du filtre. Pour les onglets sans période personnalisée, l'intervalle de temps peut être modifié en appuyant sur le bouton du sélecteur de temps dans le coin supérieur droit (le nom du bouton dépend de l'intervalle sélectionné : Cette semaine, 30 dernières minutes, Hier, etc.). Cette option n'est disponible que pour les filtres dans *Monitoring* > *Problems*.
De/À	Début et fin de la période en syntaxe temporelle absolue (`Y-m-d H:i:s`) ou relative (`now-1d`). Disponible si *Définir une période personnalisée* est coché.

Une fois enregistré, le filtre est créé comme onglet de filtre nommé et activé immédiatement.

Pour modifier les propriétés de filtre d'un filtre existant, appuyez sur le symbole d'engrenage à côté du nom de l'onglet actif.



Remarques :

- Pour masquer la zone de filtre, cliquez sur le nom de l'onglet actuel. Cliquez de 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, appuyez sur *Entrée* pour ouvrir.
- Les boutons gauche/droite au-dessus du filtre peuvent être utilisés pour passer d'un filtre enregistré à l'autre. Sinon, le bouton pointant vers le bas ouvre un menu déroulant contenant tous les filtres enregistrés, et vous pouvez cliquer sur celui dont vous avez besoin.
- Les onglets de filtre peuvent être réorganisés par glisser-déposer.

- 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 *Mettre à jour*, qui s'affiche à la place du bouton *Enregistrer sous*.
- Les paramètres actuels du filtre sont mémorisés dans le profil utilisateur. Lorsque l'utilisateur ouvre de nouveau la page, les paramètres du filtre restent identiques.

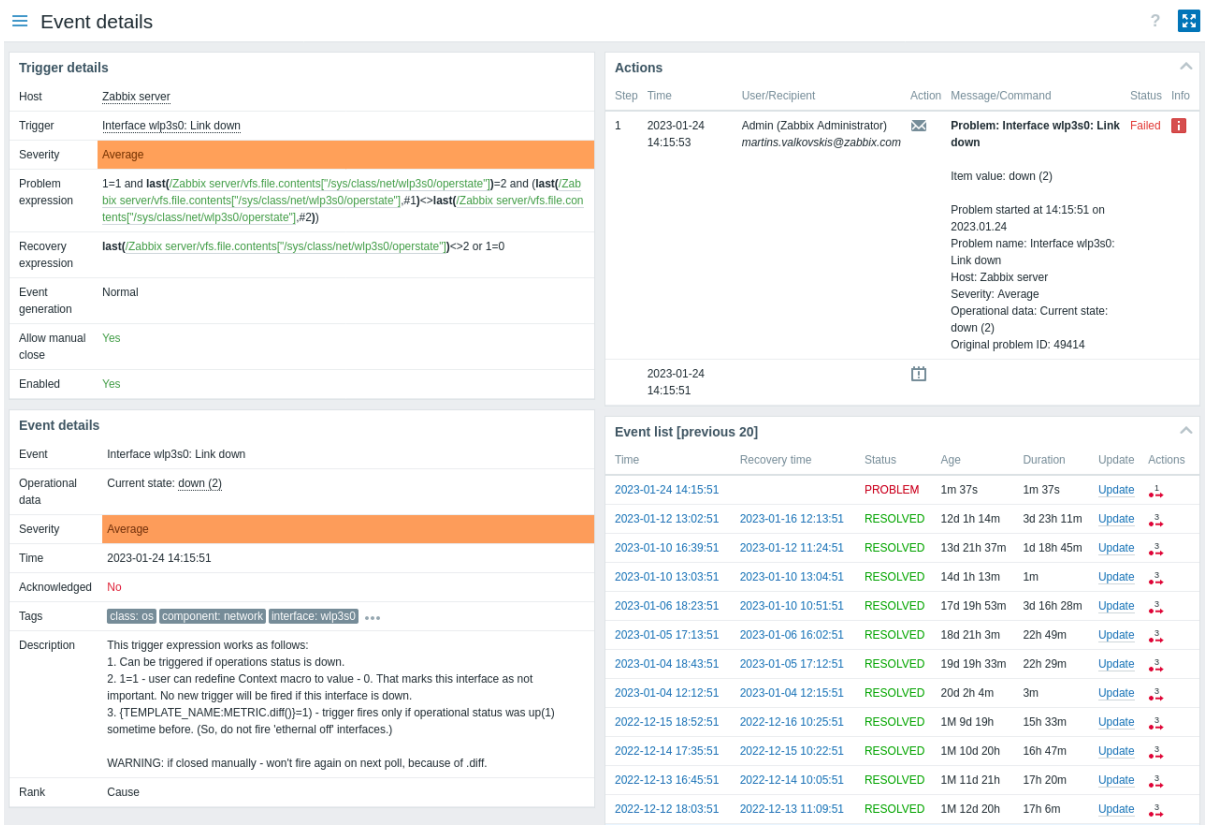
Note:
Pour partager des filtres, copiez l'URL d'un filtre actif et envoyez-la à d'autres personnes. Après avoir ouvert cette URL, les autres utilisateurs pourront enregistrer cet ensemble de paramètres comme filtre permanent dans leur compte Zabbix.
Voir aussi : Paramètres de page.

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 et de résolution du problème dans *Surveillance* > *Problèmes* sont des liens. Cliquer dessus ouvre plus de détails sur l'événement.



Notez que la sévérité du problème peut différer entre 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 *Mettre à jour le problème*.

Dans la liste des actions, les icônes suivantes sont utilisées pour indiquer le type d'activité :

- - événement de problème généré.

-  - un message a été envoyé.
-  - événement de problème acquitté.
-  - événement de problème non acquitté.
-  - un commentaire a été ajouté.
-  - la sévérité du problème a été augmentée (par ex. Information > Avertissement).
-  - la sévérité du problème a été diminuée (par ex. Avertissement > Information).
-  - la sévérité du problème a été modifiée, puis est revenue à son niveau d'origine (par ex. Avertissement > Information > Avertissement).
-  - une commande distante a été exécutée.
-  - l'événement de problème a été résolu.
-  - le problème a été fermé manuellement.
-  - le problème a été supprimé.
-  - le problème n'est plus supprimé.
-  - le problème a été converti en problème symptôme.
-  - le problème a été converti en problème cause.

1 Problèmes de cause et de symptôme

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.

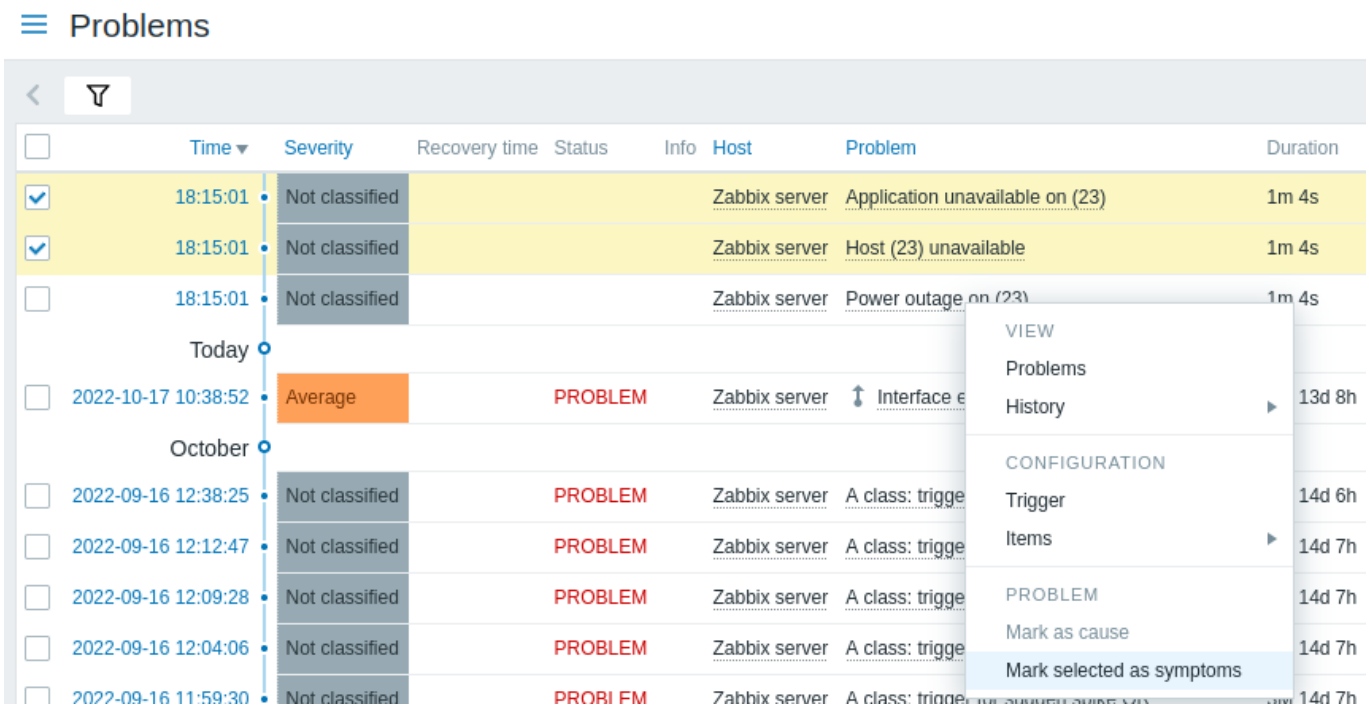
Seuls les problèmes de cause sont comptabilisés dans les totaux des problèmes sur les cartes, dans les widgets du tableau de bord tels que *Problèmes par gravité* ou *Hôtes avec problèmes*, etc. Cependant, le classement des problèmes n'affecte pas les services.

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-le d'abord dans la liste des problèmes. Un ou plusieurs problèmes peuvent être sélectionnés.

Ensuite, accédez au problème cause et, dans son menu contextuel, cliquez sur l'option *Marquer la sélection comme symptômes*.



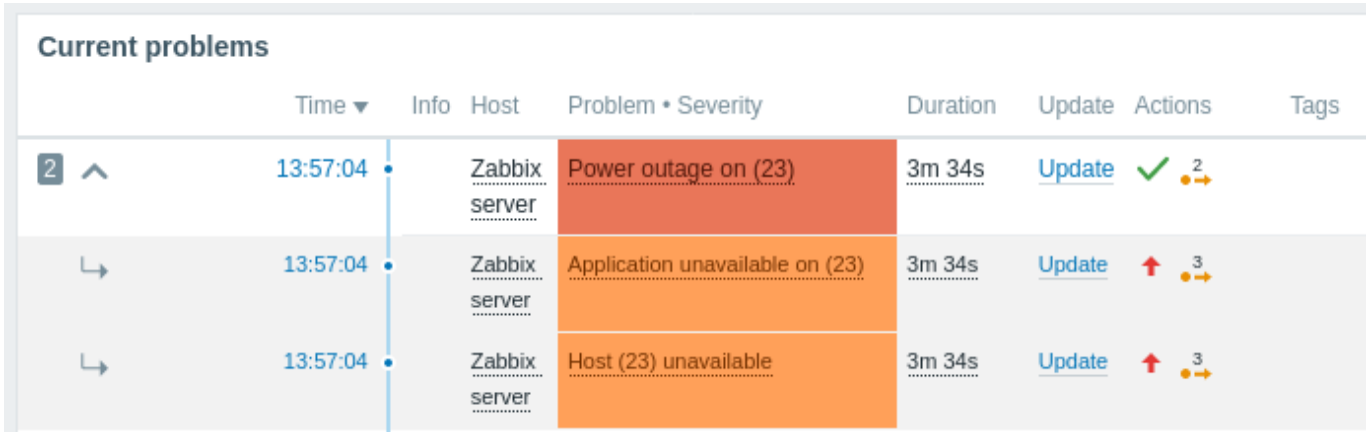
Après cela, les problèmes sélectionnés seront mis à jour par le serveur en tant que problèmes symptômes du problème cause.

Pendant la mise à jour du statut du problème, celui-ci est affiché de l'une des deux manières suivantes :

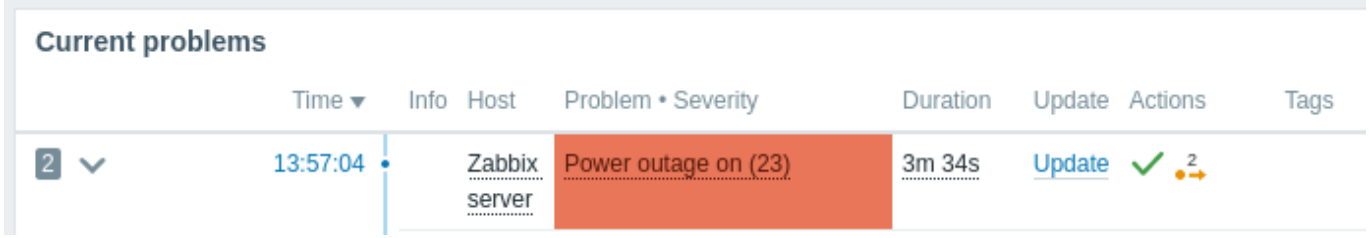
- Un statut clignotant « UPDATING » s'affiche dans la colonne Statut ;
- Une icône clignotante ou dans la colonne Info (cela s'applique si seul *Problems* est sélectionné dans le filtre et que, par conséquent, la colonne Statut n'est pas affichée).

Affichage

Les problèmes symptômes sont affichés sous le problème cause et marqués en conséquence dans *Monitoring* -> *Problems* (ainsi que dans le widget de tableau de bord *Problems*) - avec une icône, une police plus petite et un arrière-plan différent.



En vue réduite, seul le problème cause est visible ; l'existence de problèmes symptômes est indiquée par le nombre au début de la ligne et l'icône permettant de développer la vue.



Il est également possible d'afficher en plus les problèmes symptômes avec une police normale et sur leur propre ligne. Pour cela, sélectionnez *Show symptoms* dans les paramètres du filtre ou la configuration du widget.

Rétablir en tant que problème de cause

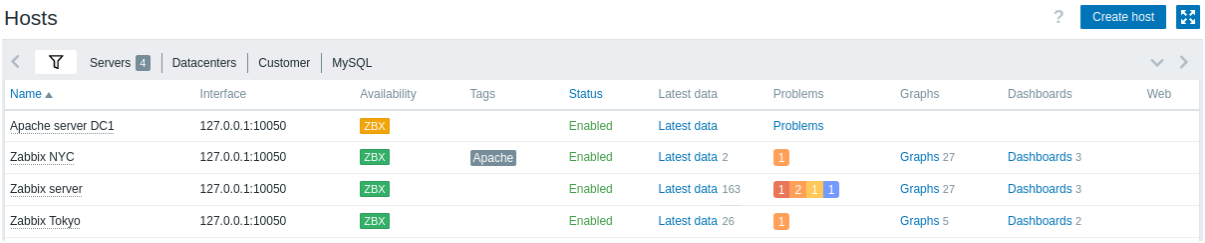
Un problème symptôme peut être rétabli en tant que problème de cause. Pour ce faire, vous pouvez :

- cliquer sur l'option *Mark as cause* dans le menu contextuel du problème symptôme ;
- cocher l'option *Convert to cause* dans l'écran de mise à jour du problème et cliquer sur *Update* (cette option fonctionne également si plusieurs problèmes sont sélectionnés).

2 Hôtes

Vue d'ensemble

La section *Surveillance → 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), ainsi que des liens permettant d'accéder facilement aux dernières données de l'hôte, à l'historique des problèmes, aux graphiques, aux tableaux de bord et aux scénarios web.



Colonne	Description
Nom	Le nom visible de l'hôte. Un clic 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 colonne pour trier les hôtes par nom dans l'ordre croissant ou décroissant.
Interface	L'interface principale de l'hôte est affichée.

Colonne	Description
Disponibilité	La disponibilité de l'hôte par interface configurée est affichée. Les icônes de disponibilité représentent l'état actuel des interfaces de l'hôte sur le serveur Zabbix. Par conséquent, si vous désactivez un hôte dans l'interface web, 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. En survolant l'icône, une fenêtre contextuelle affiche une 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 contrôles actifs. Si un hôte n'a que des contrôles actifs, 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 unique d'hôte 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 des interfaces, voir État d'interface inconnu et Paramètres d'interface d'hôte inaccessible/non disponible. 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 est disponible ou inconnue ; **Rouge** - toutes les interfaces ne sont pas disponibles ; **Gris** - au moins une interface est inconnue, mais aucune n'est non disponible. **Disponibilité des contrôles actifs.** Si au moins un contrôle actif est activé sur l'hôte, les contrôles actifs affectent également la disponibilité totale de l'interface Agent comme décrit ci-dessus. Pour déterminer la disponibilité des contrôles actifs, des messages heartbeat sont envoyés dans le thread de contrôle actif de l'agent. La fréquence des messages heartbeat est contrôlée par le paramètre `HeartbeatFrequency` dans la configuration de Zabbix agent ou agent 2 (60 secondes par défaut, plage 0-3600). Les contrôles actifs sont considérés comme non disponibles lorsque le heartbeat du contrôle actif 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 contrôle actif ; la disponibilité de leurs hôtes reste donc inconnue.
Tags	Les tags de l'hôte et de tous les modèles liés, avec macros non résolues.
État	État de l'hôte - *Activé* ou *Désactivé*. Cliquez sur l'en-tête de colonne pour trier les hôtes par état dans l'ordre croissant ou décroissant.
Dernières données	Un clic sur le lien ouvre la page *Surveillance* **-** *Dernières données* avec toutes les dernières données collectées depuis l'hôte. Le nombre d'éléments avec dernières données est affiché en gris.
Problèmes	Le nombre de problèmes ouverts de l'hôte triés par gravité. La couleur du carré indique la gravité du problème. Le nombre dans le carré correspond au nombre de problèmes pour la gravité donnée. Un clic sur l'icône ouvre la page *Surveillance* **-** *Problèmes* pour l'hôte actuel. Si un hôte n'a aucun problème, un lien vers la section *Problèmes* pour cet hôte est affiché sous forme de texte. Utilisez le filtre pour sélectionner si les problèmes supprimés doivent être inclus (non inclus par défaut).
Graphiques	Un clic sur le lien affiche 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.
Tableaux de bord	Un clic sur le lien affiche 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.

Colonne	Description
Web	Un clic sur le lien affiche 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.

Boutons

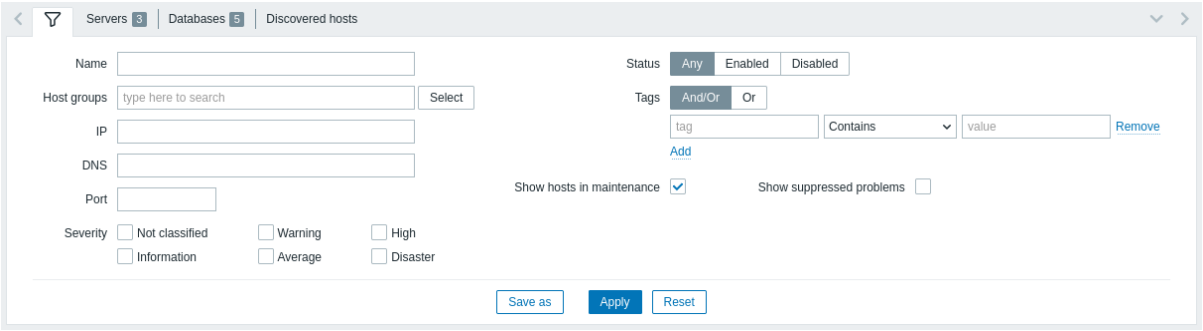
Créer un hôte permet de créer un nouvel hôte. Ce bouton est disponible uniquement pour les utilisateurs Admin et Super Admin.

Les boutons de mode d'affichage communs à toutes les sections sont décrits sur la page Surveillance.

Utilisation du filtre

Vous pouvez utiliser le filtre pour n'afficher que 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, sévérité du problème, statut (activé/désactivé/tous) ; vous pouvez également choisir d'afficher les problèmes supprimés et les hôtes actuellement en maintenance.



Paramètre	Description
Nom	Filtrer par nom visible de l'hôte.
Groupes d'hôtes	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.
Port	Filtrer par numéro de port.
Sévérité	Filtrer par sévérité du problème. Par défaut, les problèmes de toutes les sévérités sont affichés. Les problèmes sont affichés s'ils ne sont pas supprimés.
Statut	Filtrer par statut de l'hôte.
Tags	Filtrer par nom et valeur de tag d'hôte. Les hôtes peuvent être filtrés par tags au niveau de l'hôte ainsi que par tags de tous les modèles liés, y compris les modèles imbriqués. Il est possible d'inclure comme 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 : **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) Il existe deux types de calcul 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	Cochez la case pour afficher les hôtes en maintenance (affichés par défaut).

Paramètre	Description
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 ou d'une suppression de problème.

Enregistrement du filtre

Les paramètres de filtre favoris peuvent être enregistrés sous forme d'onglets, puis rapidement consultés en cliquant sur l'onglet correspondant au-dessus du filtre.

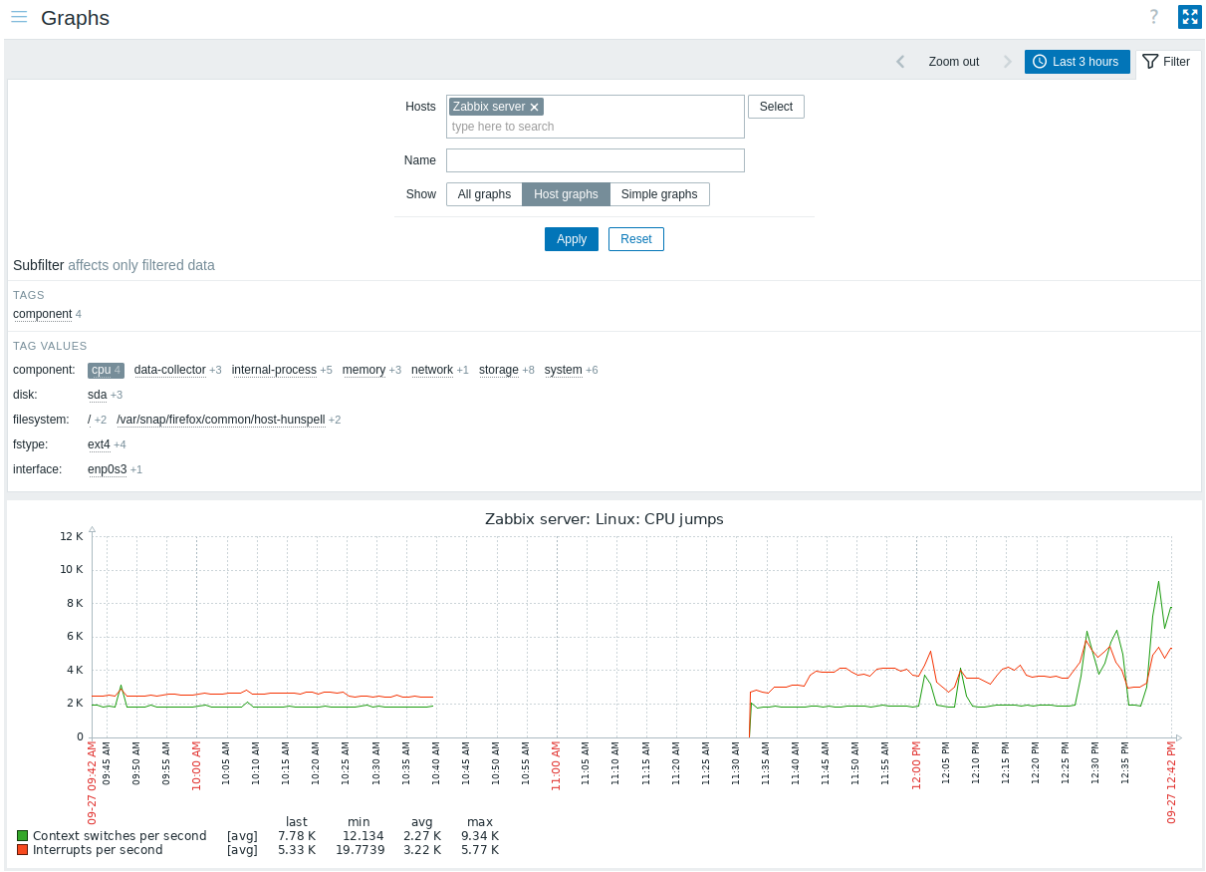
Voir plus de détails sur l'enregistrement des filtres.

1 Graphiques

Aperçu

Les graphiques d'hôte sont accessibles depuis *Monitoring* → *Hosts* en cliquant sur Graphs 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.

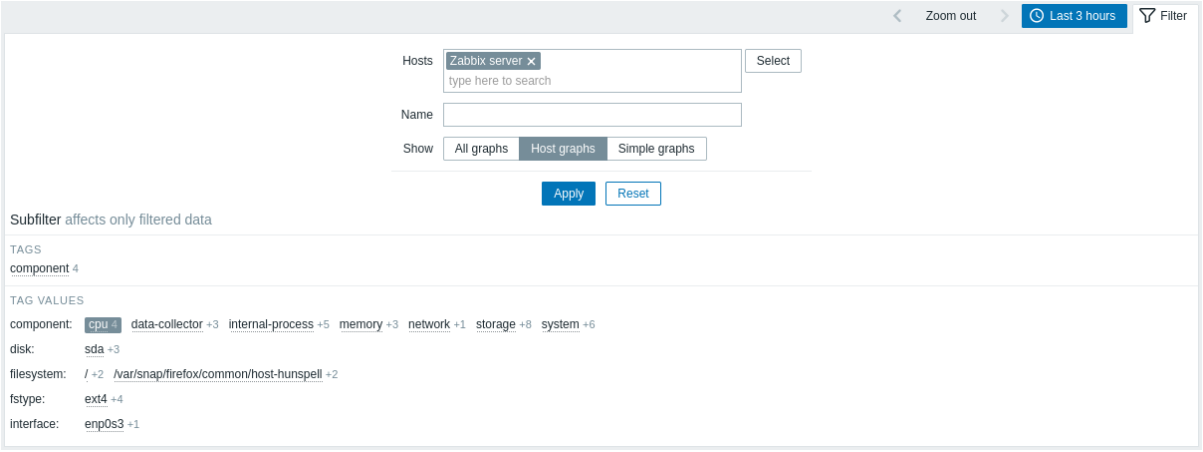
Sélecteur de période

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 permet de spécifier l'hôte, le nom du graphique et l'option *Afficher* (tous/graphiques d'hôte/graphiques simples).



Si aucun hôte n'est sélectionné dans le filtre, aucun graphique n'est affiché.

Utilisation du sous-filtre

Le sous-filtre permet de modifier davantage le filtrage du filtre principal.

Il contient des liens cliquables pour un accès rapide aux graphiques associés. Les graphiques sont associés par une entité commune — nom de tag ou valeur. Lorsqu'un nom/une valeur de tag est cliqué, il est mis en évidence avec un arrière-plan gris, et les graphiques sont immédiatement filtrés (il n'est pas nécessaire de cliquer sur *Appliquer* dans le filtre principal). Le fait de cliquer sur un autre nom/une autre valeur de tag l'ajoute aux résultats filtrés. Le fait de cliquer à nouveau sur le nom/la valeur de tag supprime le filtrage.

Les sous-filtres sont générés sur la base des données filtrées, limitées à 1000 enregistrements. Si vous souhaitez voir davantage d'enregistrements dans le sous-filtre, vous devez augmenter la valeur du paramètre *Limit for search and filter results* (dans *Administration -> General -> GUI*).

Contrairement au filtre principal, le sous-filtre est mis à jour avec chaque requête d'actualisation du tableau afin de toujours obtenir des informations à jour sur les options de filtrage disponibles et leurs compteurs.

Le nombre d'entités affichées horizontalement est limité à 100. S'il y en a davantage, une icône à trois points est affichée à la fin ; elle n'est pas cliquable. Les listes verticales (comme les tags avec leurs valeurs) sont limitées à 20 entrées. S'il y en a davantage, une icône à trois points est affichée ; elle n'est pas cliquable.

Un nombre à côté de chaque entité cliquable indique le nombre de graphiques qu'elle contient dans les résultats du filtre principal.

Une fois qu'une entité est sélectionnée, les nombres affichés avec les autres entités disponibles sont précédés d'un signe plus indiquant combien de graphiques peuvent être ajoutés à la sélection actuelle.

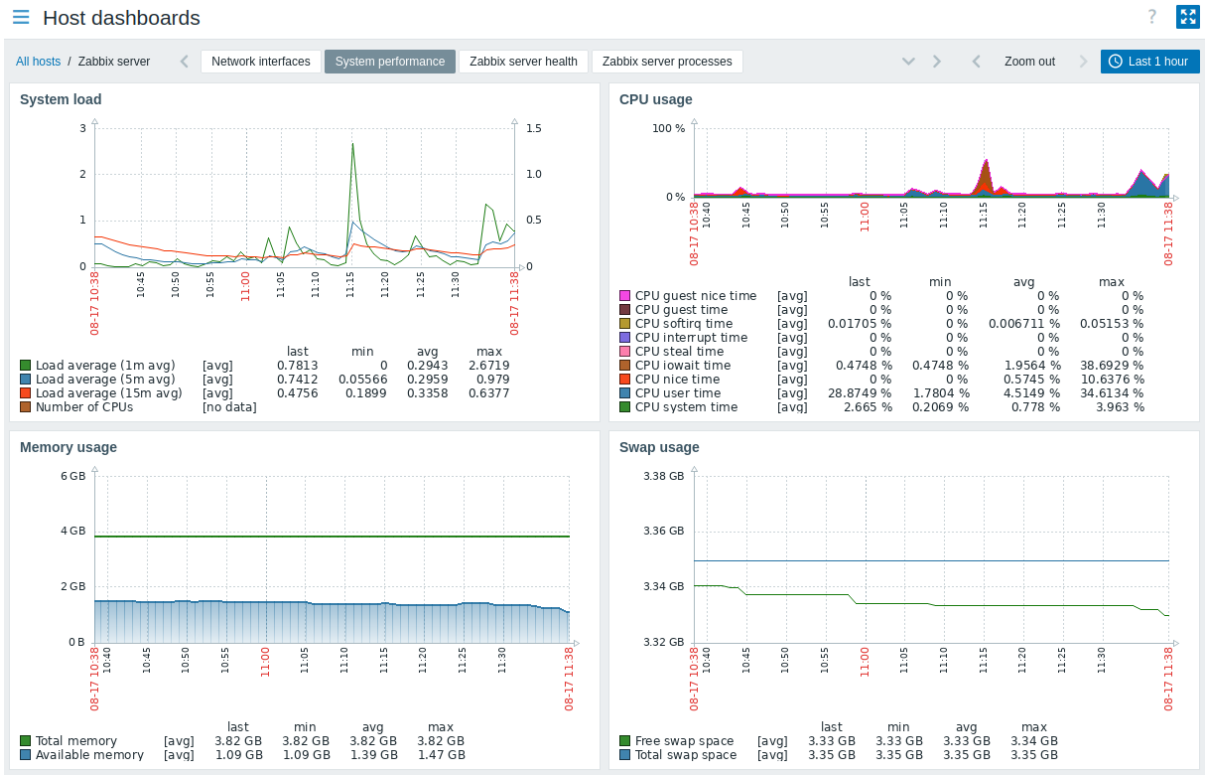
Boutons

Les boutons du mode d'affichage, communs à toutes les sections, sont décrits sur la page Monitoring.

2 Tableaux de bord d'hôte

Vue d'ensemble

Les tableaux de bord d'hôte ressemblent aux tableaux de bord globaux ; cependant, les tableaux de bord d'hôte n'ont pas de propriétaire et affichent des données uniquement pour l'hôte sélectionné.



Lors de l'affichage des tableaux de bord d'hôte, vous pouvez passer d'un tableau de bord configuré à un autre en cliquant sur :

- les onglets du tableau de bord ;
- les boutons fléchés sous l'en-tête ;
- le bouton fléché sous l'en-tête, qui affichera la liste complète des tableaux de bord d'hôte disponibles.

Pour passer à la section *Monitoring* → *Hosts*, cliquez sur le lien de navigation *All hosts* 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.

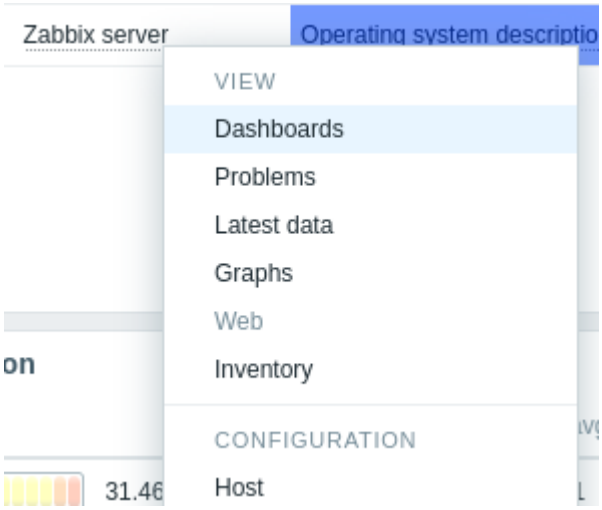
Les widgets des tableaux de bord d'hôte ne peuvent également être configurés qu'au niveau du template, à l'exception de la modification de l'intervalle d'actualisation. De plus, les widgets des tableaux de bord d'hôte ne peuvent être copiés vers d'autres tableaux de bord d'hôte qu'au sein du même template. 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 sont accessibles :

- après avoir recherché un nom d'hôte dans la recherche globale (cliquez sur le lien *Dashboards* fourni dans les résultats de recherche) ;
- après avoir cliqué sur un nom d'hôte dans *Inventaire* → *Hôtes* (cliquez sur le lien *Dashboards* fourni dans la vue d'ensemble de l'hôte) ;
- depuis le menu de l'hôte en cliquant sur *Dashboards*.



Notez que les tableaux de bord d'hôte *ne peuvent pas* être accessibles directement dans la section *Dashboards*, qui est réservée aux tableaux de bord globaux.

3 Scénarios web

Vue d'ensemble

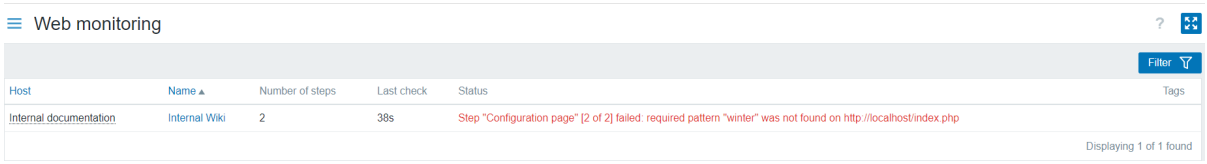
Les informations sur les scénarios web d'un hôte sont accessibles depuis *Monitoring* → *Hosts* en cliquant sur Web pour l'hôte concerné.



Un clic sur le nom de l'hôte ouvre le menu de l'hôte. Les données des hôtes désactivés sont également accessibles. Le nom d'un hôte désactivé est affiché en rouge.

Le statut d'un scénario web peut être :

- OK (vert) - toutes les étapes ont réussi et Zabbix a collecté des données pour le scénario.
- Error (rouge) - une ou plusieurs étapes ont échoué (consultez le message d'erreur pour obtenir des indications sur les paramètres à corriger).
- Unknown (gris) - aucun statut d'étape n'est encore disponible ou aucune donnée n'a été collectée.



Le nombre maximal de scénarios affichés par page dépend du paramètre *Rows per page* du profil utilisateur 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 pour les 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 d'un 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 *Surveillance* → *Hôtes*, sélectionnez cet hôte ou ce groupe dans le filtre. Vous pouvez également filtrer les scénarios en fonction des tags.

Boutons

Les boutons de mode d'affichage communs à toutes les sections sont décrits sur la page Surveillance.

3 Dernières données

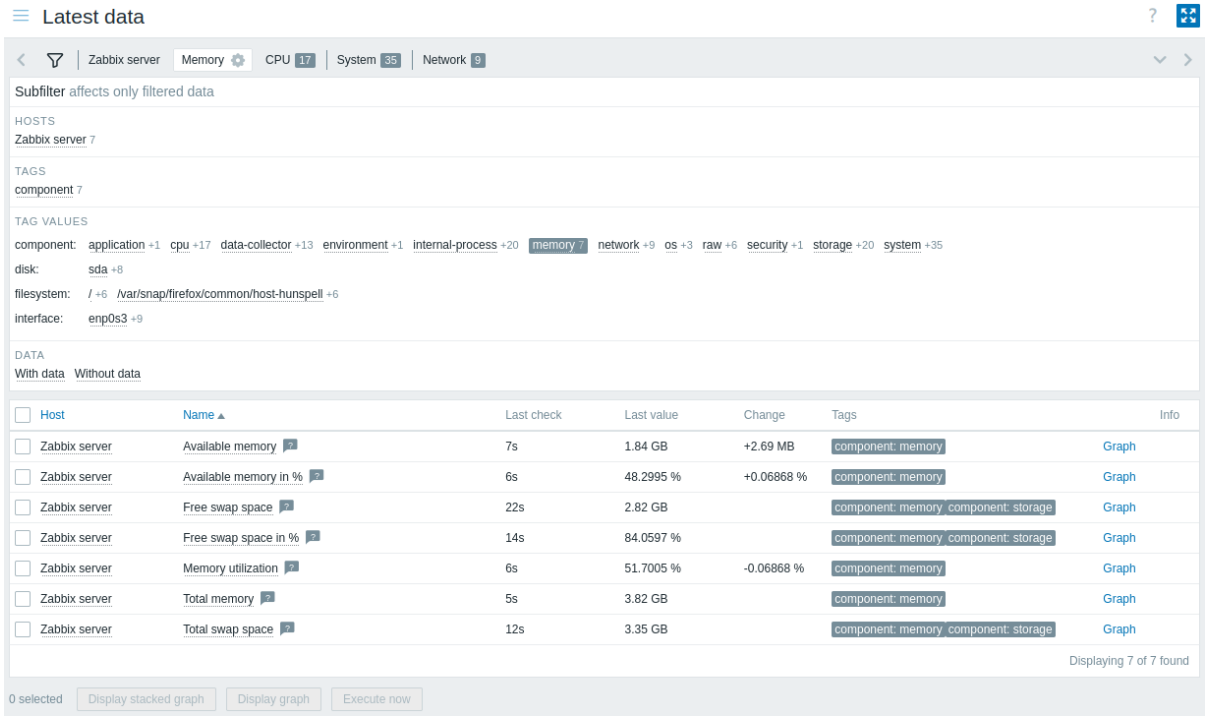
Vue d'ensemble

La section *Surveillance* → *Dernières données* 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 sont affichés que si le filtre est défini et qu'il y a des résultats à afficher.



Colonne	Description
Hôte	Nom de l'hôte auquel l'élément appartient. Cliquer sur le nom ouvre 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, son nom s'affiche en rouge. Notez que les données des hôtes désactivés (y compris les graphiques et les listes de valeurs d'éléments) sont accessibles dans la section *Dernières données*.
Nom	Nom de l'élément. Cliquer sur le nom ouvre le menu de l'élément. Une icône en forme de point d'interrogation s'affiche à côté du nom de l'élément pour tous les éléments ayant une description. Survolez l'icône pour afficher une infobulle contenant la description de l'élément.
Dernière vérification	Temps écoulé depuis la dernière vérification de l'élément.
Dernière valeur	Valeur la plus récente de l'élément. Les valeurs sont affichées avec la conversion d'unité et le mappage de valeurs appliqués. Survolez la valeur pour afficher les données brutes (tronquées à 8192 caractères). Les valeurs binaires sont affichées sous forme d'espace réservé (*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 dernières données ; pour l'étendre, mettez à jour la valeur du paramètre *Max history display period* dans *Administration → General → GUI*.
Changement	Différence entre la valeur précédente et la valeur la plus récente. Pour les éléments ayant une fréquence de mise à jour de 1 jour ou plus, la variation ne sera jamais affichée (avec le paramétrage 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. Les tags dans la liste des éléments sont cliquables. Cliquer sur un tag l'active dans le sous-filtre, ce qui fait que la liste des éléments n'affiche que les éléments contenant ce tag (ainsi que tous les autres tags précédemment sélectionnés dans le sous-filtre). Notez qu'une fois les éléments filtrés de cette manière, les tags dans la liste des éléments ne sont plus cliquables. Toute modification supplémentaire basée sur les tags (par exemple, pour supprimer des tags ou spécifier d'autres filtres) doit être effectuée dans le sous-filtre.
Graphique/Historique	Lien vers le graphique simple/historique des valeurs de l'élément.

Colonne	Description
Info	Informations supplémentaires sur l'élément. Si un élément comporte des erreurs (par exemple, s'il est devenu non pris en charge), une icône d'information s'affiche. Survolez l'icône pour plus de détails.

Boutons

Les boutons du mode d'affichage, communs à toutes les sections, sont décrits sur la page Monitoring.

Actions de masse

Les boutons situés sous la liste permettent d'effectuer des actions de masse sur un ou plusieurs éléments sélectionnés :

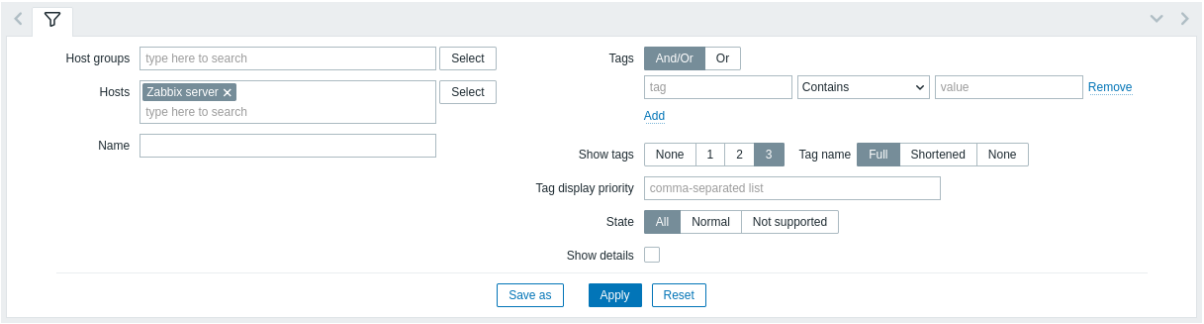
- *Afficher un graphique empilé* - affiche un graphique ad hoc empilé.
- *Afficher un graphique* - affiche un graphique ad hoc simple.
- *Exécuter maintenant* - exécute immédiatement une vérification pour obtenir de nouvelles valeurs d'élément. Pris en charge uniquement pour les vérifications **passives** (voir plus de détails). Cette option est disponible uniquement pour les hôtes disposant d'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 *Invoquer "Exécuter maintenant" sur les hôtes en lecture seule*.

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 de filtre est située au-dessus de la liste des éléments et du sous-filtre. Cliquez dessus pour développer le filtre.



Le filtre permet de restreindre la liste des éléments par groupe d'hôtes, hôte, nom d'élément, tag, état et autres paramètres. La spécification d'un groupe d'hôtes parent dans le filtre sélectionne implicitement tous les groupes d'hôtes imbriqués. Voir *Monitoring → Problems* pour plus de détails sur le filtrage par tags.

L'option de filtre *Show details* permet d'étendre les informations affichées pour les éléments. Cochez-la pour afficher des détails tels que l'intervalle de mise à jour de l'élément, les paramètres d'historique et de tendances, le type d'élément et les erreurs d'élément (ok/non pris en charge).

Enregistrement du filtre

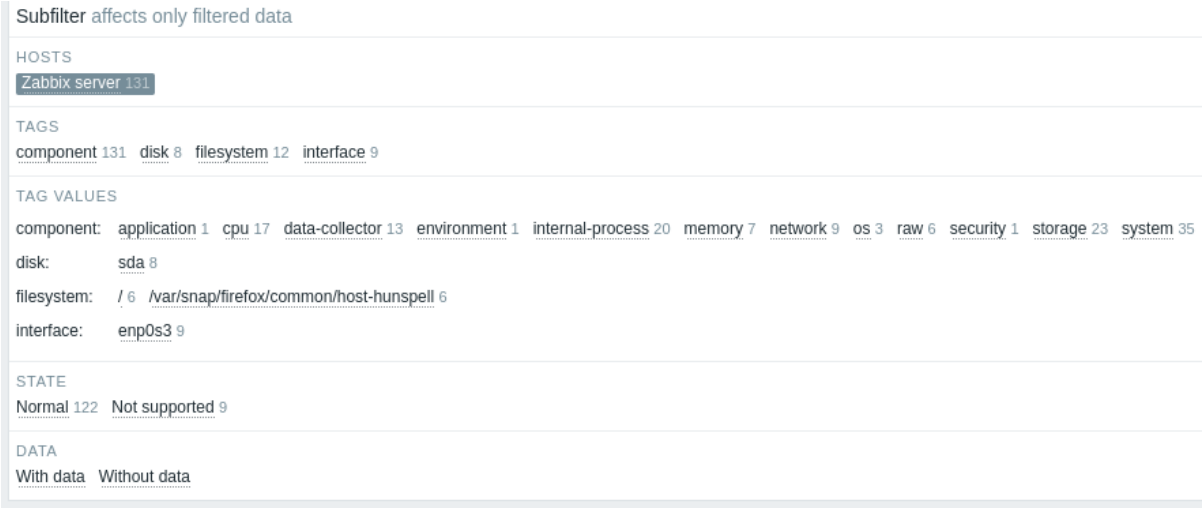
Les paramètres de filtre favoris peuvent être enregistrés sous forme d'onglets, puis rapidement consultés en cliquant sur l'onglet correspondant au-dessus du filtre.

Voir plus de détails sur l'enregistrement des filtres.

Utilisation du sous-filtre

Le sous-filtre permet de modifier davantage le filtrage du filtre principal.

Il contient des liens cliquables pour un accès rapide aux éléments associés. Les éléments sont associés par une entité commune : hôte, nom ou valeur de tag, état de l'élément ou statut des données. Lorsqu'une entité est cliquée, elle est mise en évidence avec un arrière-plan gris, et les éléments sont immédiatement filtrés (il n'est pas nécessaire de cliquer sur *Appliquer* dans le filtre principal). Cliquer sur une autre entité l'ajoute aux résultats filtrés. Cliquer de nouveau sur l'entité supprime le filtrage.



Les sous-filtres sont générés à partir des données filtrées, qui sont limitées à 1000 enregistrements. S'il y a 20 hôtes ayant chacun 100 éléments (soit 2000 enregistrements au total), seule la moitié des hôtes sera visible dans le sous-filtre. Si vous souhaitez voir davantage d'enregistrements dans le sous-filtre, vous devez augmenter la valeur du paramètre *Limit for search and filter results* (dans *Administration -> General -> GUI*).

Contrairement au filtre principal, le sous-filtre est mis à jour à chaque requête d'actualisation du tableau afin de toujours disposer d'informations à jour sur les options de filtrage disponibles et leurs compteurs.

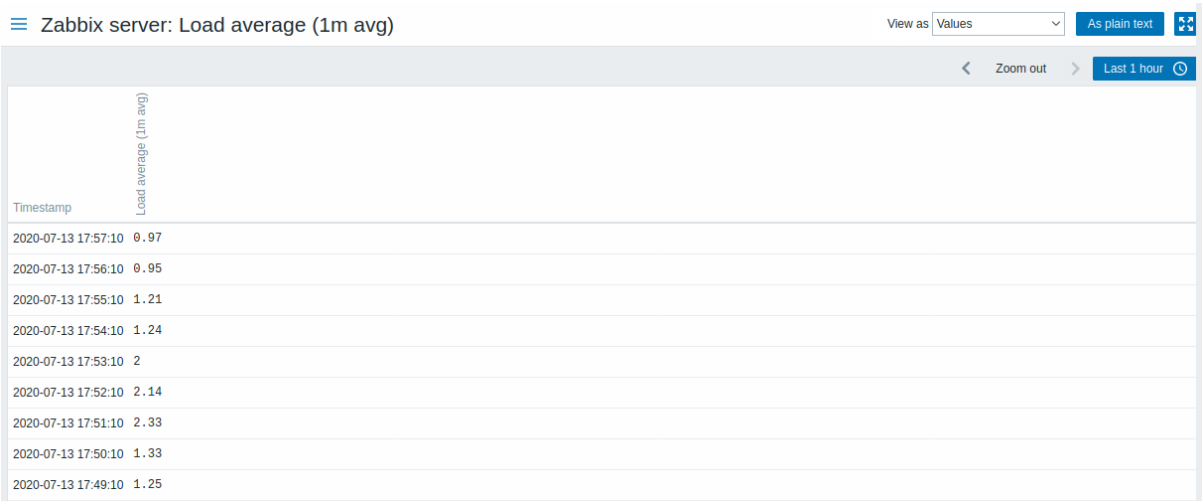
Pour chaque groupe d'entités (hôtes, tags, valeurs de tag, etc.), jusqu'à 10 lignes d'entités sont affichées. S'il y a davantage d'entités, cette liste peut être développée pour afficher un maximum de 1000 entrées (la valeur de SUBFILTER_VALUES_PER_GROUP dans les définitions du frontend) en cliquant sur l'icône à trois points ••• à la fin de la liste. Pour les *valeurs de tag*, la liste peut être développée pour afficher un maximum de 200 noms de tag avec leurs valeurs correspondantes. Notez qu'une fois entièrement développée, la liste ne peut plus être réduite.

Un nombre à côté de chaque entité cliquable indique le nombre d'éléments regroupés dans celle-ci (sur la base des résultats du filtre principal). Lorsqu'une entité est cliquée, les nombres des 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.

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 permet également 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*, défini dans *Administration → General → GUI*.

4 Cartes

Aperçu

Dans la section *Surveillance→ Cartes*, vous pouvez configurer, gérer et afficher les cartes du réseau.

Lorsque vous ouvrez cette section, vous verrez soit la dernière carte à laquelle vous avez accédé, soit une 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 :

Colonne	Description
Nom	Nom de la carte. Cliquez sur le nom pour afficher la carte.
Largeur	La largeur de la carte est affichée.
Hauteur	La hauteur de la carte est affichée.
Actions	Deux actions sont disponibles : **Propriétés** - définir les propriétés générales de la carte **Modifier** - accéder à la grille pour ajouter des éléments de carte

Pour configurer une nouvelle carte, cliquez sur le bouton *Créer une carte* dans le coin supérieur droit. Pour importer une carte à partir d'un fichier YAML, XML ou JSON, cliquez sur le bouton *Importer* dans le coin supérieur droit. L'utilisateur qui importe la carte sera défini comme son propriétaire.

Deux boutons sous la liste offrent des options de modification en masse :

- *Exporter* - exporter les cartes vers un fichier YAML, XML ou JSON
- *Supprimer* - supprimer les cartes

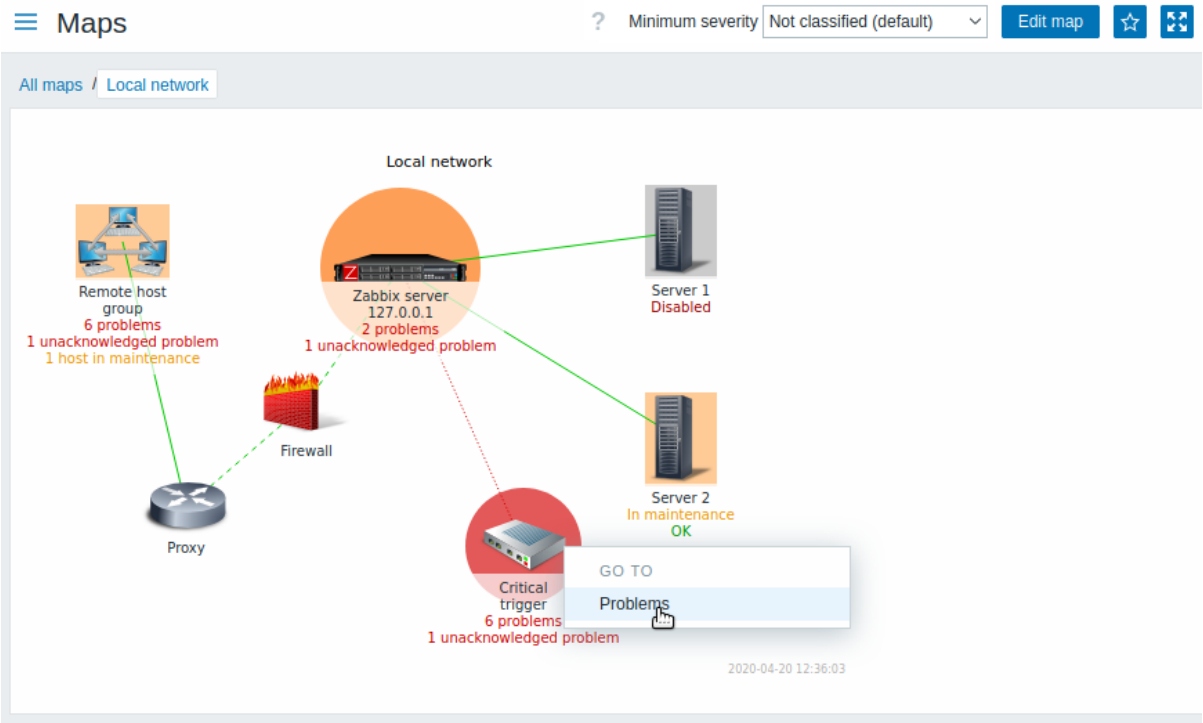
Pour utiliser ces options, cochez les cases devant les cartes concernées, puis cliquez sur le bouton requis.

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 visualiser une carte, cliquez sur son nom dans la liste de toutes les cartes.



Vous pouvez utiliser le menu déroulant dans la barre de titre de la carte pour sélectionner le plus bas niveau de sévérité des déclencheurs de problème à afficher. La sévérité marquée par *default* 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 sévérité de la carte de niveau supérieur (sauf s'elle est * Non classé *, dans ce cas, elle ne sera pas transmis 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 au niveau de gravité sélectionné ou supérieur seront affichés avec l'élément. Si tous les problèmes sont acquittés, 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 a priorité sur la mise en évidence de la gravité du problème, si l'élément de carte est un hôte.
- un hôte désactivé (non supervisé) 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.

Marqueurs de changements récents

Des triangles rouges pointant vers l'intérieur autour d'un élément indiquent un changement récent de l'état du déclencheur — c'est-à-dire un changement survenu au cours des 30 dernières minutes. Ces triangles sont affichés si la case à cocher *Mark elements on trigger status change* est cochée dans la configuration de la carte.

Liens

Cliquer sur un élément de carte ouvre un menu contenant 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 Supervision.

Résumé lisible dans les cartes

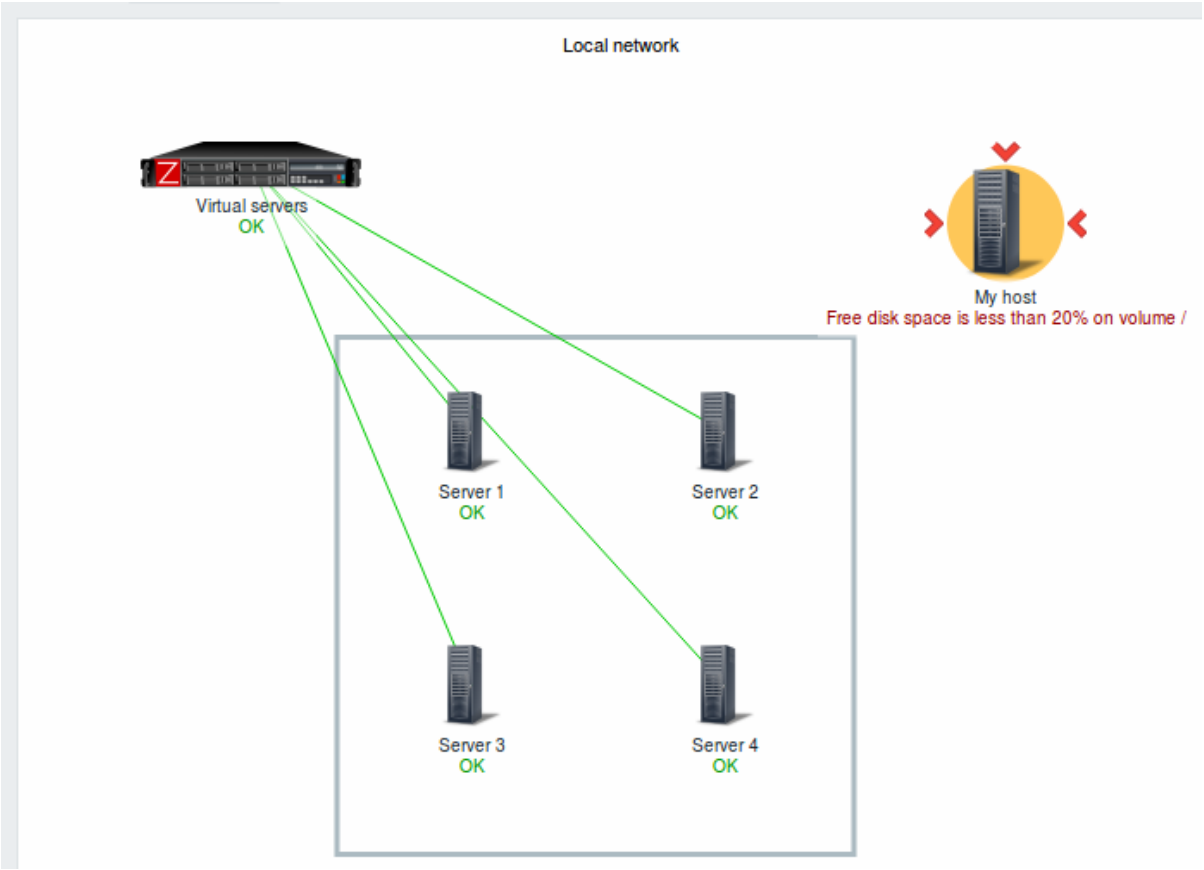
Une propriété "aria-label" cachée est disponible permettant aux informations de la carte d'être lues avec un lecteur d'écran. La description générale de la carte et une description d'élément individuel est disponible, au format suivant :

- pour la description de la carte : `<Nom de la carte>, <* de * éléments en état de problème>, <* problèmes au total>.`
- pour décrire un élément avec un problème : `<Type d'élément>, Statut <Statut de l'élément>, <Nom de l'élément>, <Description du problème>.`
- pour décrire un élément avec plusieurs problèmes : `<Type d'élément>, Statut <Statut de l'élément>, <Nom de l'élément>, <* problèmes>.`
- pour décrire un élément sans problème : `<Type d'élément>, Statut <Statut de l'élément>, <Nom de l'élément>.`

Par exemple, cette description est disponible :

`'Réseau local, 1 élément sur 6 en état de problème, 1 problème au total. Hôte, Problème d'état, Mon hôte,`

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`

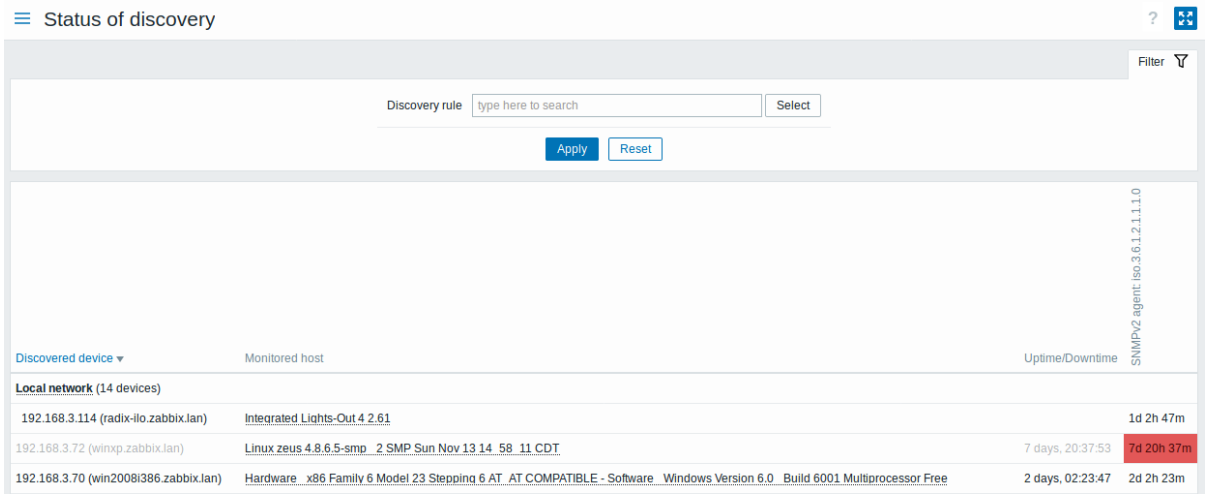
ouvrira la carte portant ce nom (Local network).

Si `sysmapid` (ID de carte) et `mapname` (nom de carte) sont spécifiés, `mapname` a une priorité plus élevée.

5 Découverte

Vue d'ensemble

Dans la section *Surveillance* → *Découverte*, les résultats de la découverte réseau sont affichés. Les appareils découverts sont triés par règle de découverte.



Données affichées :

Colonne	Description
Appareil découvert	Les appareils découverts sont répertoriés, regroupés par règle de découverte. En cliquant sur la règle de découverte, le menu de la règle s'ouvre et contient le lien vers le formulaire de configuration de la règle de découverte.
Hôte surveillé	Si un appareil est déjà surveillé, le nom de l'hôte sera indiqué dans cette colonne. En cliquant sur le nom de l'hôte, le menu de l'hôte s'ouvre.
Temps de fonctionnement/Temps d'arrêt	La durée pendant laquelle l'appareil est découvert ou perdu après une découverte précédente est affichée dans cette colonne.
Vérification de découverte	L'état du service individuel (vérification de découverte) pour chaque appareil découvert est affiché. Une cellule rouge indique que le service est indisponible. La durée de fonctionnement ou d'arrêt du service est incluse dans la cellule. Cette colonne n'est affichée que si le service a été trouvé sur au moins un appareil découvert.

Boutons

Les boutons de mode d'affichage communs à toutes les sections sont décrits sur la page Surveillance.

Utilisation d'un 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 des macros non résolues.

Si 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 répertoriées pour la sélection. Plusieurs règles de découverte peuvent être sélectionnées.

3 Services

Aperçu

Le menu Services concerne les fonctions surveillance du service de Zabbix.

1 Services

Aperçu

Dans cette section, vous pouvez voir un état global de l'ensemble des services qui ont été 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 un aperçu de la fonctionnalité de supervision des services.)

Les principales catégories d'état de service sont *OK* ou *Problème*, l'état *Problème* étant exprimé par le nom et la couleur correspondants du niveau de gravité du problème.

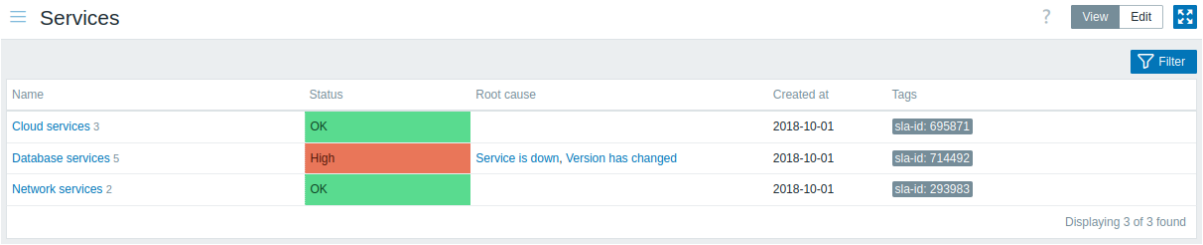
Alors que le mode d'affichage permet de superviser les services avec leur état et d'autres détails, vous pouvez également configurer la hiérarchie des services dans cette section (ajouter/modifier des services et des services enfants) en passant en mode édition.

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/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 s'affiche.

Données affichées :

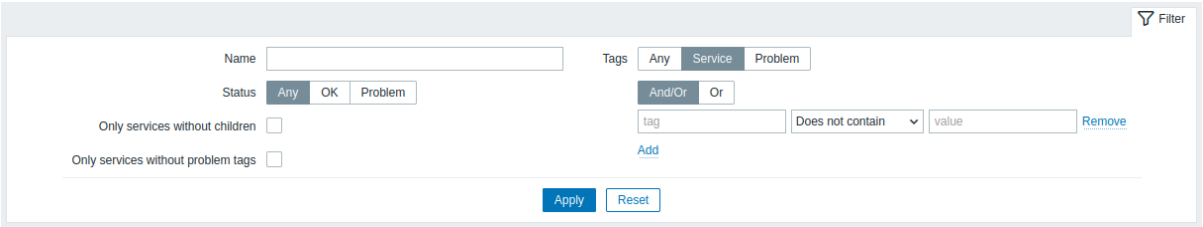
Paramètre	Description
Nom	Nom du service. Le nom du service est un lien vers les détails du service. Le nombre après le nom indique combien de services enfants le service possède.
Statut	État du service : **OK** - aucun problème **<couleur du problème et gravité>** - indique un problème et sa gravité. En cas de problèmes multiples, la couleur et la gravité du problème le plus critique sont affichées.
Cause racine	Les problèmes sous-jacents qui affectent directement ou indirectement l'état du service sont répertoriés. Les mêmes problèmes sont répertoriés comme renvoyés par la macro {SERVICE.ROOTCAUSE}. Cliquez sur le nom du problème pour voir plus de détails à son sujet dans *Surveillance → Problèmes*. Les problèmes qui n'affectent pas l'état du service ne figurent pas dans la liste.
Créé le	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.



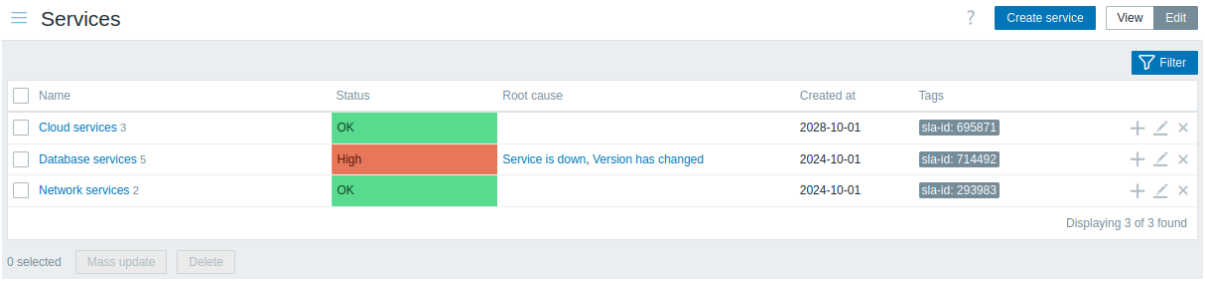
Paramètre	Description
Nom	Filtrer par nom de service.
Statut	Filtrer par état du service.

Paramètre	Description
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.
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 ces options supplémentaires :

-  - ajouter un service enfant à ce service
-  - modifier ce service
-  - supprimer ce service

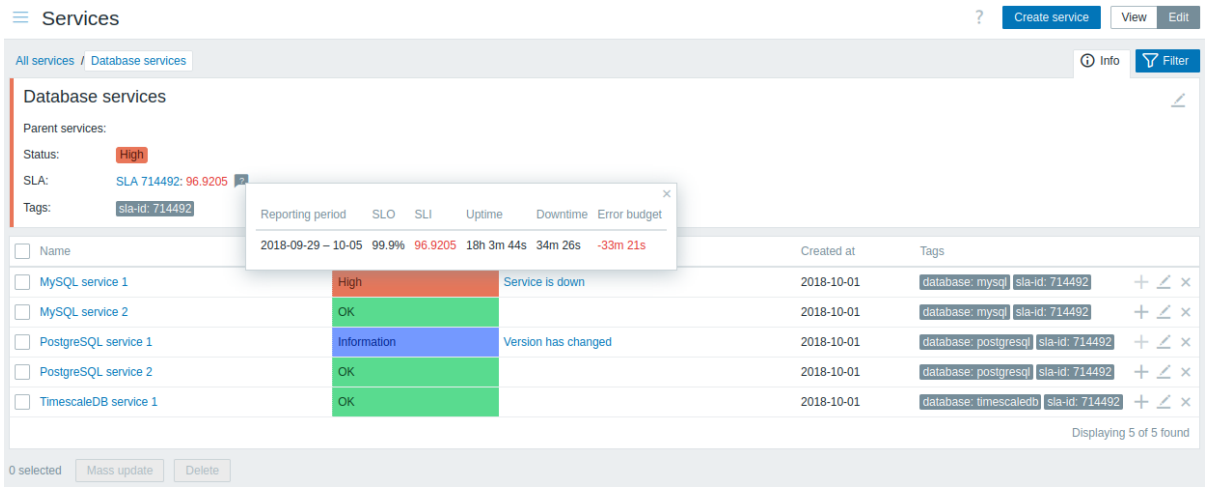


Pour configurer un nouveau service, cliquez sur le bouton *Create service* dans le coin supérieur droit.

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 comprennent la boîte d'informations et la liste des services enfants.



Pour accéder à la boîte d'informations, cliquez sur l'onglet *Info*. La boîte d'informations contient les éléments suivants :

- Noms des services parents (le cas échéant)
- Statut 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 boîte 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 du service

La boîte d'informations contient également un lien vers la configuration du service.

Pour utiliser le filtre des services enfants, cliquez sur l'onglet *Filter*.

En mode édition, la liste des services enfants est complétée par des options d'édition supplémentaires :

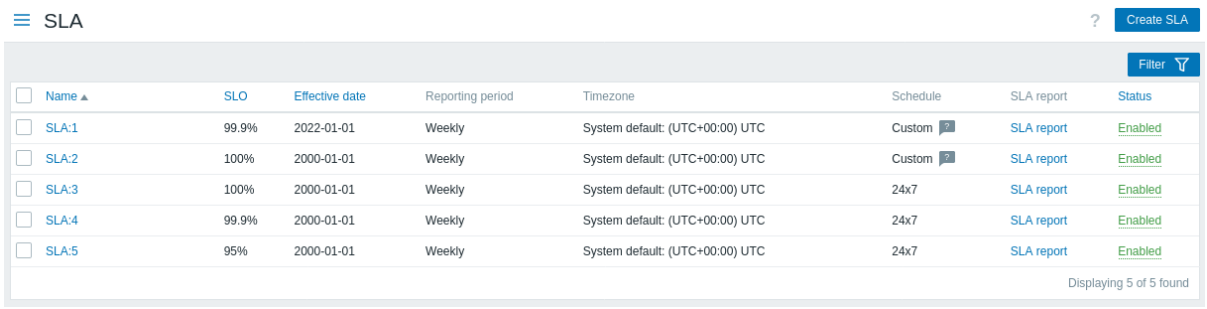
-  - ajouter un service enfant à ce service
-  - modifier ce service
-  - supprimer ce service

2 SLA

Aperçu

Cette section permet de visualiser et de configurer les SLA.

SLAs



Une liste des SLA configurés s'affiche. *Notez* que seuls les SLA liés aux services accessibles à l'utilisateur seront affichés (en lecture seule, sauf si *Gérer les SLA* est activé pour le rôle d'utilisateur).

Données affichées :

Paramètre	Description
Nom	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.
Date d'effet	La date de début du calcul du SLA est affichée.

Paramètre	Description
Période de rapport	La période utilisée dans le rapport SLA est affichée - *quotidiennement*, *hebdomadairement*, *mensuellement*, *trimestriellement* ou *annuellement*.
Fuseau horaire	Le fuseau horaire SLA est affiché.
Calendrier	Le calendrier SLA est affiché - 24x7 ou personnalisé.
Rapport SLA	Cliquez sur le lien pour voir le rapport SLA pour ce SLA.
Statut	Le statut SLA est affiché - activé ou désactivé.

3 Rapport SLA

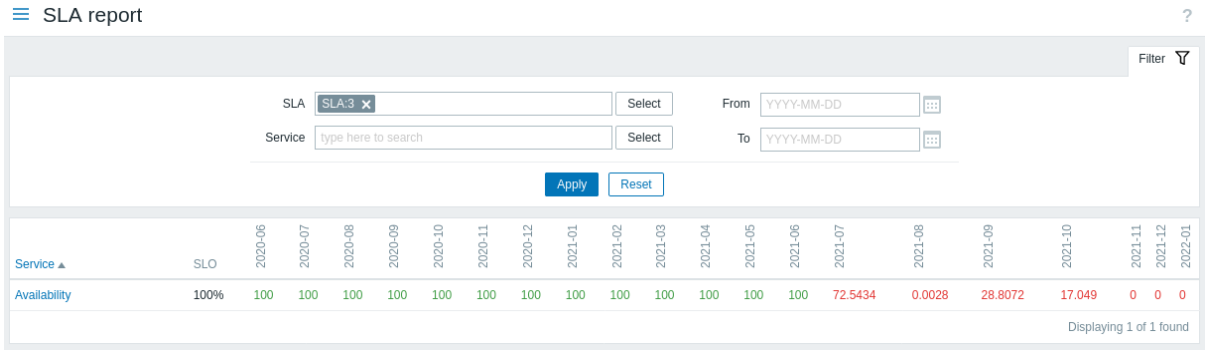
Aperçu

Cette section permet de consulter les rapports SLA, selon les critères sélectionnés dans le filtre.

Les rapports SLA peuvent également être affichés sous forme de widget de tableau de bord.

Rapport

Le filtre 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.

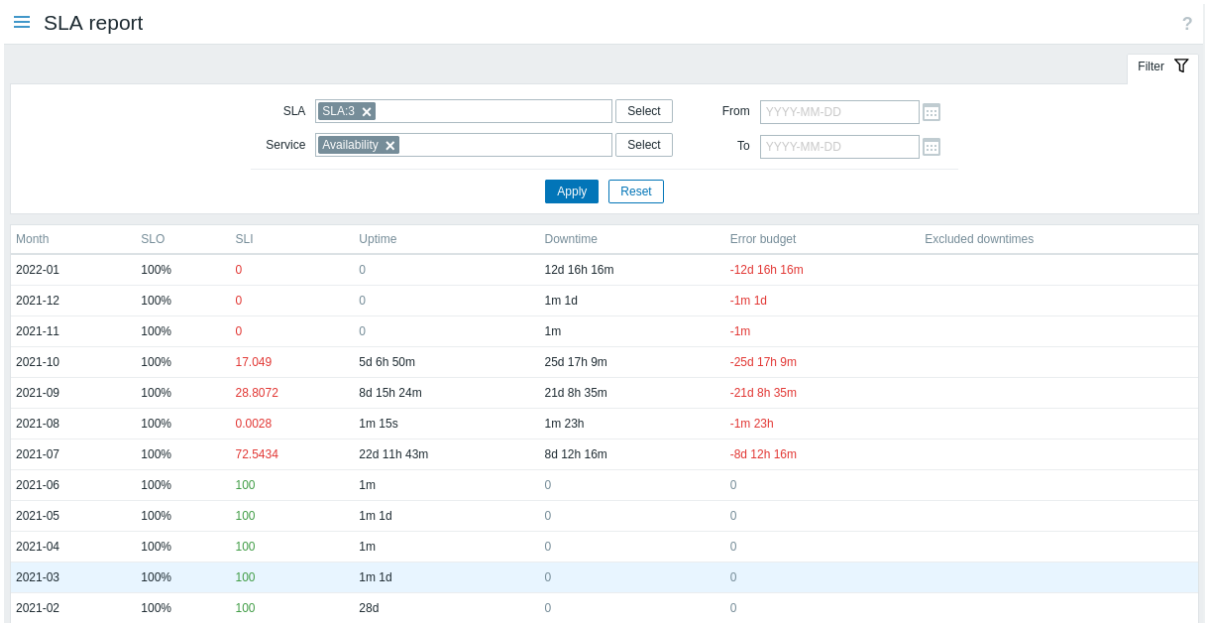


Chaque colonne (période) affiche le SLI pour cette période. Les SLI qui ne respectent pas le SLO défini sont surlignés en rouge.

20 périodes sont affichées dans le rapport. Un maximum de 100 périodes peuvent être affichées, si la date *De* et la date *À* sont 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 rapport du SLA.

4 Inventaire

Aperçu

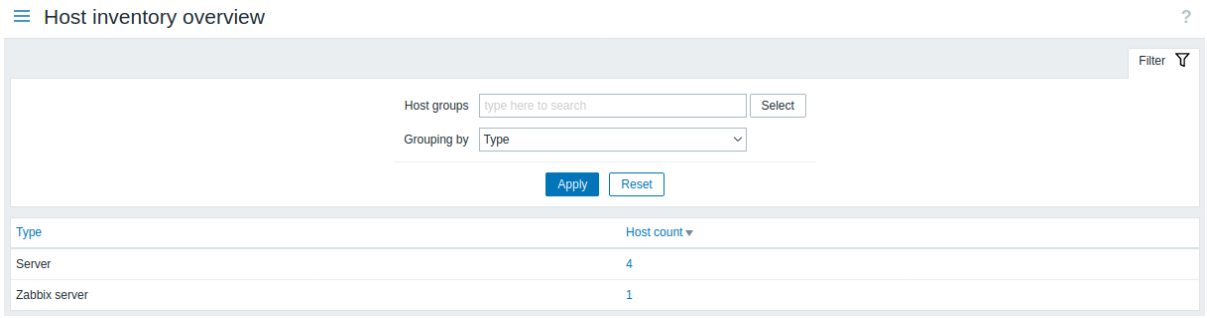
Le menu Inventaire comprend des sections offrant un aperçu des données d'inventaire de l'hôte selon un paramètre choisi, ainsi que la possibilité d'afficher les détails de l'inventaire de l'hôte.

1 Aperçu

Aperçu

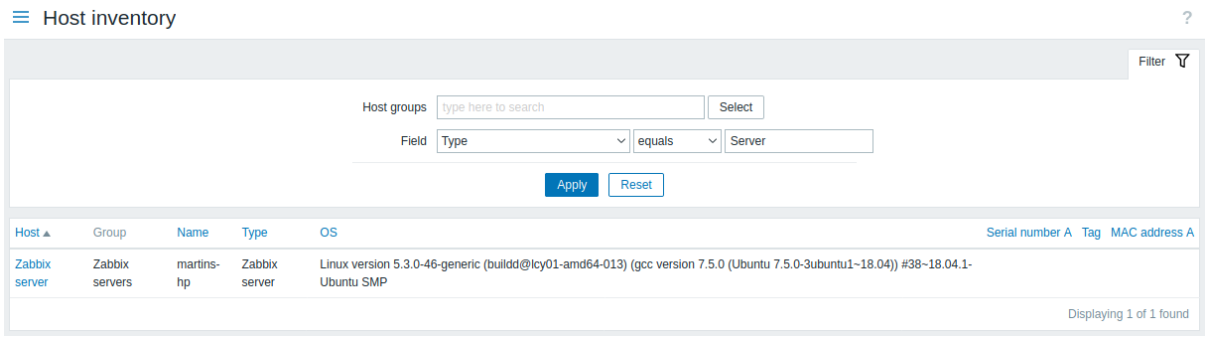
La section *Inventaire* → *Aperçu* fournit des moyens d'avoir un aperçu des données d'inventaire d'hôte.

Pour qu'un aperçu puisse être affiché, choisissez un groupe d'hôtes (ou aucun) et le champ d'inventaire dont vous souhaitez afficher des données. Le nombre d'hôtes correspondant à chaque entrée du champ choisi sera affiché.



L'intégralité de l'aperçu dépend de la quantité d'informations d'inventaire conservées concernant les hôtes.

Les nombres dans la colonne *Nombre d'hôtes* sont des liens, ils mènent à un filtre basé sur ces hôtes dans le tableau *Inventaire d'hôte*.

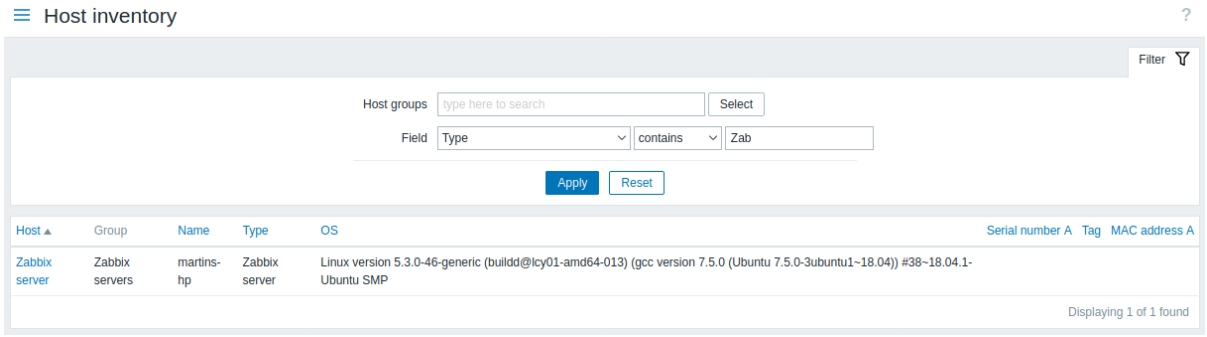


2 Hôtes

Aperçu

Dans la section *Inventaire* → *Hôtes*, 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 de n'afficher que les hôtes qui vous intéressent.

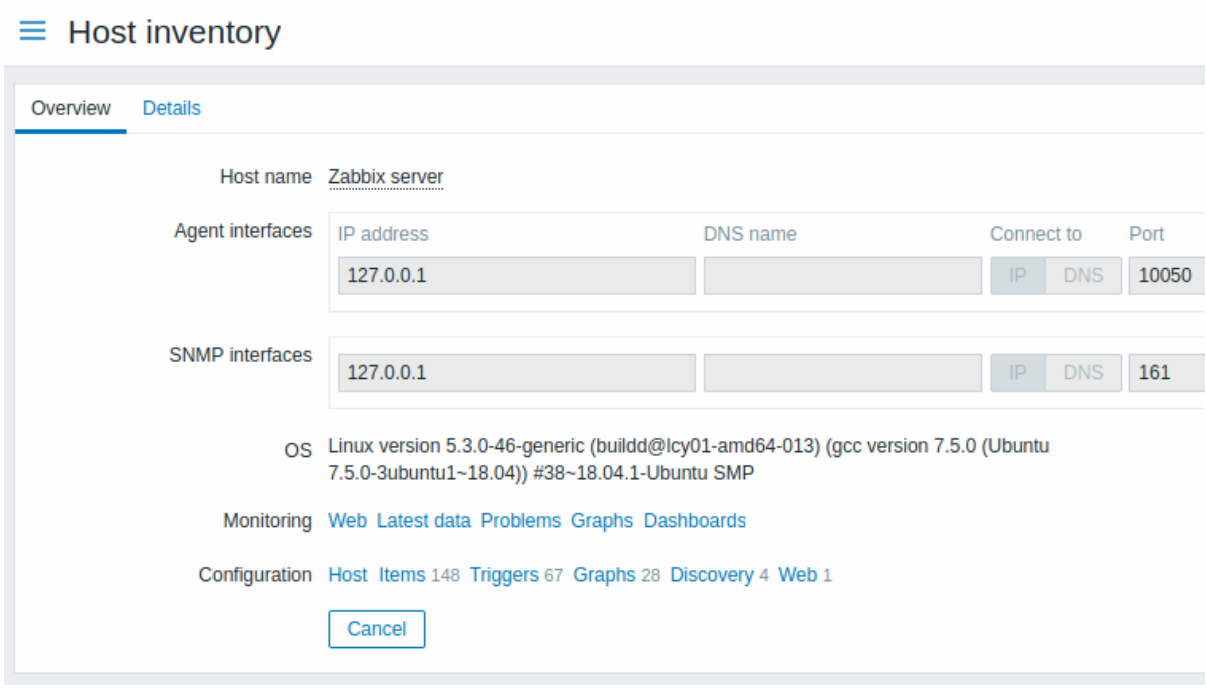


Pour afficher tous les inventaires d'hôtes, appuyez sur le bouton « Réinitialiser ».

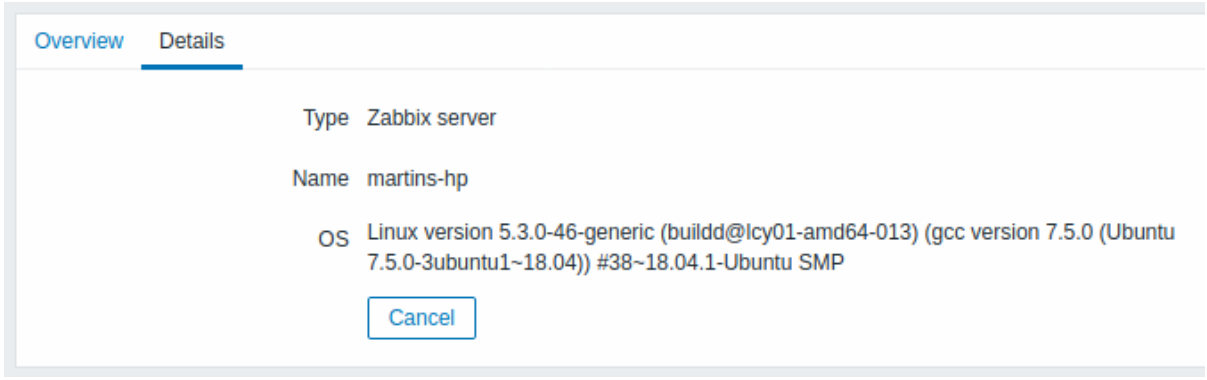
Bien que seuls certains champs d'inventaire clés soient affichés dans le tableau, vous pouvez également consulter toutes les informations d'inventaire disponibles pour cet hôte. Pour ce faire, cliquez sur le nom de l'hôte dans la première colonne.

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.

5 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.

1 Information système

Aperçu

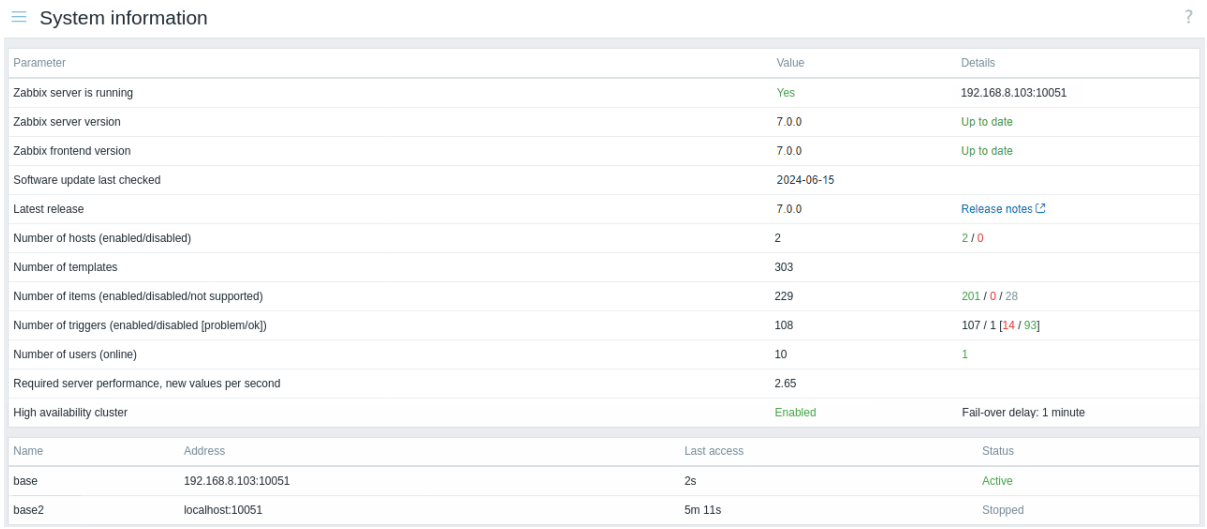
Dans *Rapports* → *Informations système*, un résumé des principales données du serveur Zabbix et du système est affiché. Les données système sont collectées à l'aide des éléments internes.

Notez que dans une configuration de haute disponibilité, il est possible de rediriger la source des informations système (instance du serveur). Pour ce faire, modifiez le fichier *zabbix.conf.php* : décommentez et définissez `$ZBX_SERVER` ou à la fois `$ZBX_SERVER` et `$ZBX_SERVER_PORT` vers un serveur autre que celui affiché comme actif. Notez que si vous définissez uniquement `$ZBX_SERVER`, une valeur par défaut (10051) sera utilisée pour `$ZBX_SERVER_PORT`.

Lorsque la configuration de haute disponibilité est activée, un bloc distinct s'affiche sous les statistiques système avec les détails des nœuds de haute disponibilité. Ce bloc n'est visible que par les utilisateurs Zabbix *Super Admin*.

Les *Informations système* sont également disponibles sous forme de widget de tableau de bord.

Statistiques système



Données affichées :

Paramètre	Valeur	Détails
Le serveur Zabbix est en cours d'exécution	Statut 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.
Version du serveur Zabbix	Le numéro de version actuel du serveur est affiché. *Remarque :* Il n'est affiché que lorsque le serveur Zabbix est en cours d'exécution.	Le statut 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 de prise en charge complète de cette version a expiré. Ces informations ne sont disponibles que 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 n'existe concernant la version actuelle.
Version de l'interface Zabbix	Le numéro de version de l'interface Zabbix est affiché.	Le statut 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 de prise en charge complète de cette version a expiré. Ces informations ne sont disponibles que 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 n'existe concernant la version actuelle.

Paramètre	Valeur	Détails
Dernière vérification des mises à jour logicielles	La date de la dernière vérification des mises à jour logicielles de Zabbix est affichée. Ces informations ne sont disponibles que si la vérification des mises à jour logicielles est activée dans la configuration du serveur Zabbix.	
Dernière version	Le numéro d'une version plus récente (si disponible) pour la version actuelle de Zabbix est affiché. Ces informations ne sont disponibles que 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 n'existe concernant la version actuelle.	Un lien vers les notes de version de la dernière version disponible de Zabbix 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 de données	Le nombre total d'éléments de données est affiché.	Nombre d'éléments de données au niveau de l'hôte surveillés / désactivés / non pris en charge. Les éléments de données sur des hôtes désactivés sont comptés comme désactivés.
Nombre de déclencheurs	Le nombre total de déclencheurs est affiché.	Nombre de déclencheurs au niveau de l'hôte activés / désactivés ; répartition des déclencheurs activés selon les états « Problème » / « OK ». Les déclencheurs répertoriés dans l'état « OK » incluent les déclencheurs ayant le statut « Inconnu ». Les déclencheurs dépendant d'éléments de données désactivés ou affectés à des hôtes désactivés sont comptés comme désactivés.
Nombre d'utilisateurs	Le nombre total d'utilisateurs configurés est affiché.	Nombre d'utilisateurs en ligne.
Performances serveur requises, nouvelles valeurs par seconde	Le nombre attendu de nouvelles valeurs traitées par le serveur Zabbix par seconde est affiché.	*Performances serveur requises* est une estimation et peut servir de référence. Pour obtenir des chiffres précis sur les 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 comptés comme une valeur par intervalle de mise à jour de l'élément. Les valeurs d'intervalle régulier sont comptées ; les valeurs d'intervalle flexible et d'intervalle planifié ne le sont pas. Le calcul n'est pas ajusté pendant une période de maintenance « nodata ». Les éléments trapper ne sont pas comptés.
Scripts globaux sur le serveur Zabbix	**Désactivé** sera affiché dans ce champ si les scripts globaux sont désactivés sur le serveur Zabbix en définissant EnableGlobalScripts=0 dans la configuration du serveur.	
Cluster de haute disponibilité	Statut du cluster de haute disponibilité pour le serveur Zabbix : **Désactivé** - serveur autonome **Activé** - au moins un nœud de haute disponibilité existe	Si activé, le délai de basculement est affiché.

Informations système affichera également un message d'erreur dans les conditions suivantes :

- La base de données utilisée ne possède pas le jeu de caractères ou le classement 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 admin*).
- Le housekeeping pour TimescaleDB est configuré de manière incorrecte (les tables d'historique ou de tendances contiennent des chunks compressés, mais les options *Remplacer la période d'historique des éléments* ou *Remplacer la période de tendance des éléments* sont désactivées).

Nœuds de haute disponibilité

Si le cluster de haute disponibilité est activé, un autre bloc de données s'affiche alors avec l'état de chaque nœud de haute disponibilité.



Name	Address	Last access	Status
node-active	192.168.1.13:10051	12s	Active
node6	192.168.1.10:10053	1h 2m 40s	Unavailable
node7	192.168.1.11:10053	3m 40s	Unavailable
node4	192.168.1.8:10052	1h 34m 29s	Stopped
node5	192.168.1.9:10053	1h 9m 51s	Stopped
node8	192.168.1.12:10051	21m 16s	Stopped
node1	192.168.1.5:10051	17s	Standby
node2	192.168.1.6:10051	16s	Standby
node3	192.168.1.7:10052	16 2021-10-20 17:58:47	Standby

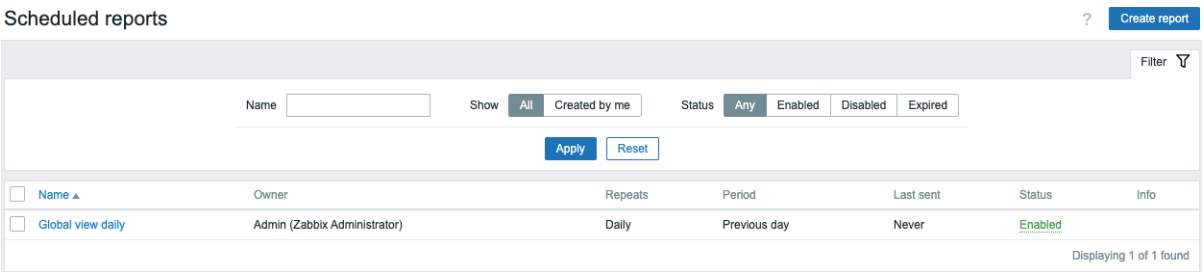
Données affichées :

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 du nœud. En survolant la cellule, l'horodatage du dernier accès s'affiche au format long.
Status	État du nœud. Les lignes du tableau sont triées selon ces états dans cet ordre de priorité : **Active** - le nœud est opérationnel et fonctionne **Unavailable** - le nœud n'a pas été vu depuis plus longtemps que le délai de basculement (vous pouvez vouloir en rechercher la raison) **Stopped** - le nœud a été arrêté ou n'a pas pu démarrer (vous pouvez vouloir le démarrer ou le supprimer) **Standby** - le nœud est opérationnel et en attente

2 Rapports planifiés

Aperçu

Dans *Rapports → Rapports planifiés*, les utilisateurs disposant de permissions 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'ouverture 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 :

Column	Description
Name	Nom du rapport. Un clic dessus ouvre le formulaire de configuration du rapport.
Owner	Utilisateur ayant créé le rapport.

Column	Description
Repeats	Fréquence de génération du rapport (quotidienne/hebdomadaire/mensuelle/annuelle).
Period	Période pour laquelle le rapport est préparé (jour, semaine, mois ou année précédente).
Last sent	Date et heure d'envoi du rapport le plus récent.
Status	Statut actuel du rapport (activé/désactivé/expiré). Les utilisateurs disposant de permissions suffisantes peuvent modifier le statut en cliquant dessus - de « Activé » à « Désactivé » (et inversement) ; de « Expiré » à « Désactivé » (et inversement). Pour les utilisateurs ne disposant pas de droits suffisants, le statut n'est pas cliquable.
Info	Affiche des icônes informatives : Une icône rouge indique que la génération du rapport a échoué ; en la survolant, une infobulle affichera les informations sur 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 qu'un rapport a expiré ; en la survolant, une infobulle affichera 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 :

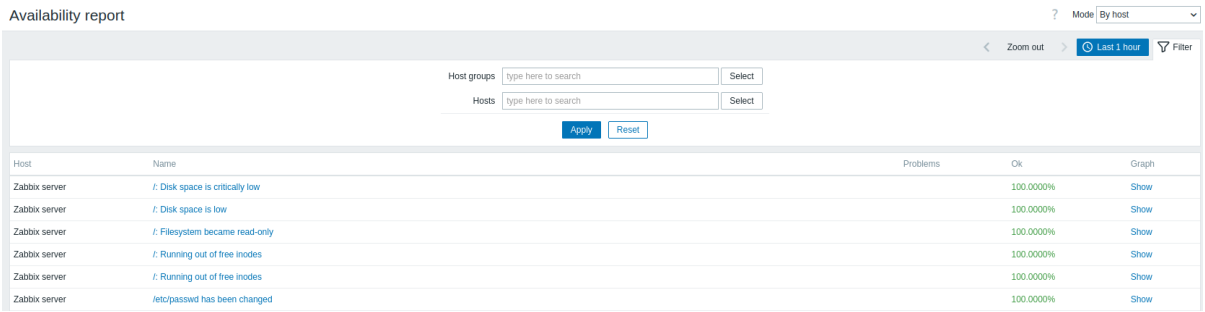
- Cochez les cases des rapports à mettre à jour dans la liste
- Cliquez sur le bouton requis sous la liste pour appliquer les modifications (*Activer*, *Désactiver* ou *Supprimer*)

3 Rapport de disponibilité

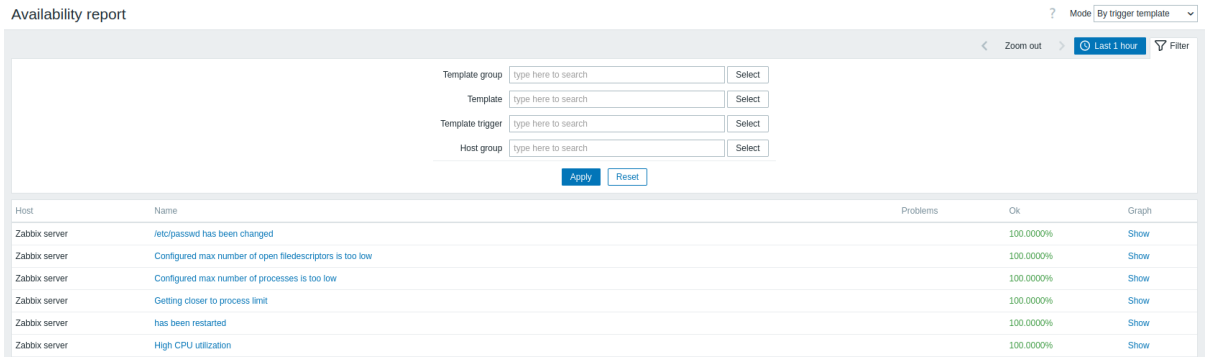
Aperçu

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 les macros non résolues.

Le filtre se trouve sous le nom de la section *Rapport de disponibilité*. Il peut être ouvert et replié 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.
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.

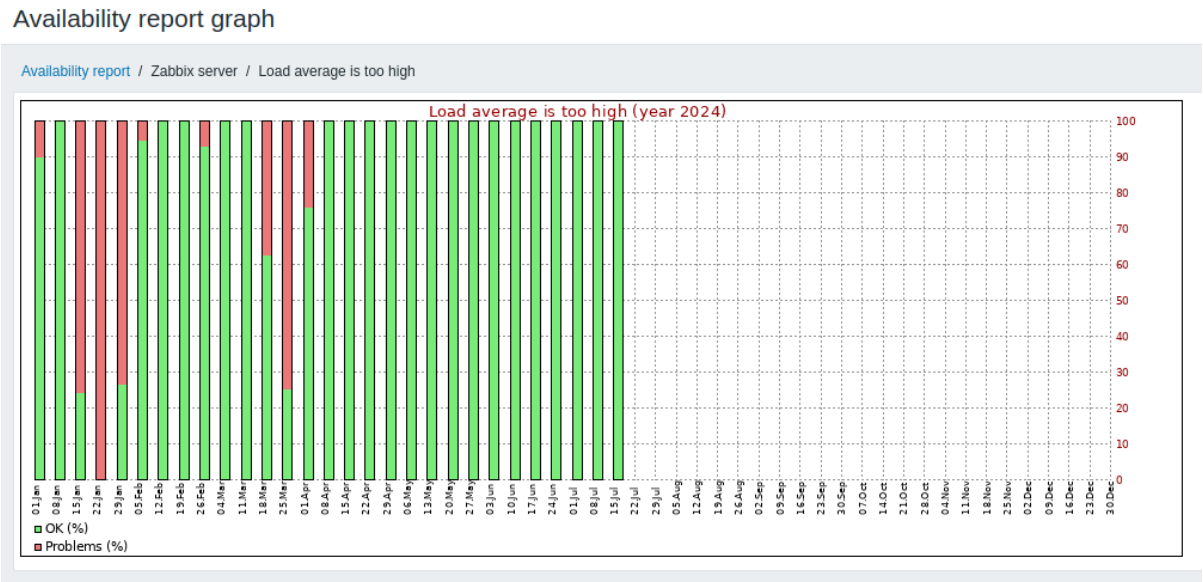
Filtrage par hôte

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 permet de sélectionner des périodes couramment utilisées en un clic. 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 en barres, où chaque barre représente une semaine écoulée de l'année en cours.



Le vert d'une barre correspond au temps OK et le rouge au temps de problème.

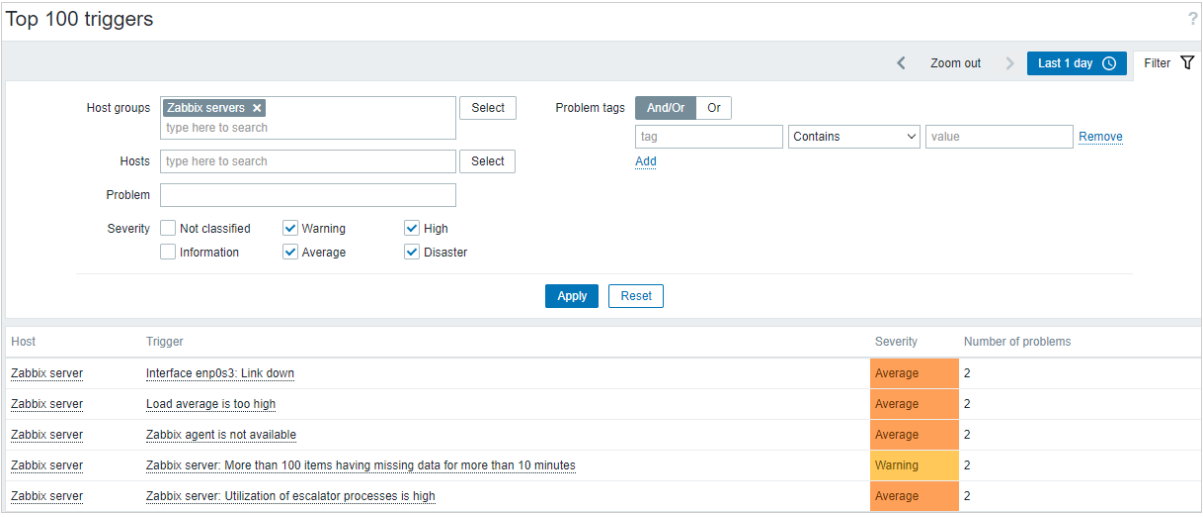
Effet des périodes de maintenance

La maintenance n'exclut pas automatiquement le temps du rapport de disponibilité. Seule la maintenance configurée comme *Aucune collecte de données* arrête l'interrogation (ainsi, aucun problème n'est généré) et laisse donc inchangé le pourcentage de disponibilité pour les déclencheurs concernés.

4 Top 100 déclencheurs

Vue d'ensemble

Dans *Rapports* → *Top 100 des déclencheurs*, vous pouvez voir les déclencheurs ayant le plus grand nombre de problèmes détectés pendant la période sélectionnée.



Les entrées des colonnes de l'hôte et du déclencheur sont toutes deux des liens offrant quelques options utiles :

- pour l'hôte - cliquer sur le nom de l'hôte affiche le menu de l'hôte
- pour le déclencheur - cliquer sur le nom du déclencheur affiche des liens vers les derniers événements, le graphique simple pour 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, balises ou sévérité du déclencheur. La spécification d'un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués. Pour de meilleures 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

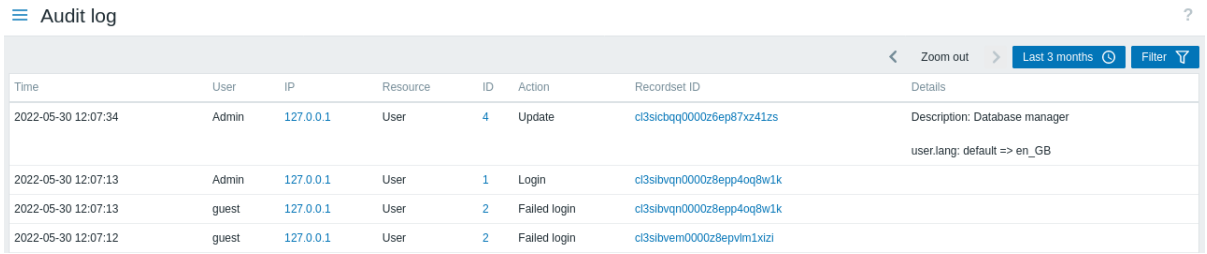
Le sélecteur de période à côté du filtre permet de sélectionner les périodes fréquemment requises en un seul clic de souris. Pour plus d'informations, voir Sélecteurs de période et d'hôte.

5 Journal d'audit

Aperçu

Dans la section *Rapports* → *Journal d'audit*, il est possible de consulter les enregistrements de l'activité des utilisateurs et du système.

Note:
Pour que les enregistrements d'audit soient collectés et affichés, la case *Activer la journalisation d'audit* doit être cochée dans la section *Administration* → *Journal d'audit*. Sans ce paramètre activé, l'historique des activités ne sera pas enregistré dans la base de données et ne sera pas affiché dans le journal d'audit.



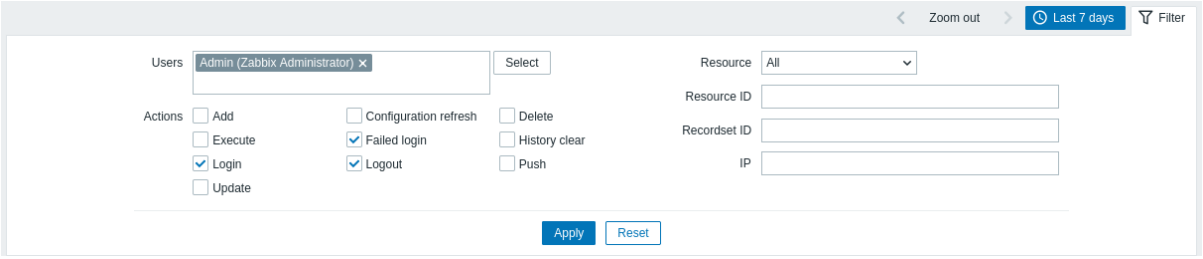
Le journal d'audit affiche les données suivantes :

Colonne	Description
Heure	Horodatage de l'enregistrement d'audit.
Utilisateur	Utilisateur ayant effectué l'activité.
IP	Adresse IP depuis laquelle l'activité a été initiée. Cliquer sur l'hyperlien filtrera les enregistrements du journal d'audit selon cette adresse IP.
Ressource	Type de la ressource concernée (*Jeton d'API*, *Action*, *Authentification*, *Auto-enregistrement*, etc.).
ID	ID de la ressource concernée. Cliquer sur l'hyperlien filtrera les enregistrements du journal d'audit selon cet ID de ressource.
Action	Type d'activité (*Ajouter*, *Actualisation de la configuration*, *Supprimer*, *Exécuter*, *Échec de connexion*, *Effacer l'historique*, *Connexion*, *Déconnexion*, *Push*, *Mettre à jour*).
ID d'ensemble d'enregistrements	ID partagé par 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 *ID d'ensemble d'enregistrements*. Cliquer sur l'hyperlien filtrera les enregistrements du journal d'audit selon cet *ID d'ensemble d'enregistrements*.
Détails	Description de la ressource et informations détaillées sur l'activité effectuée. Si un enregistrement contient plus de deux lignes, un lien supplémentaire *Détails* sera affiché. Cliquez sur ce lien pour afficher la liste complète des modifications.

Note:
Lorsqu'un élément trapper ou un élément agent HTTP (avec le trapping activé) a reçu des données, une entrée dans le journal d'audit sera ajoutée uniquement si les données ont été envoyées à l'aide de la méthode d'API `history.push`, et non avec l'utilitaire Zabbix sender.

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.

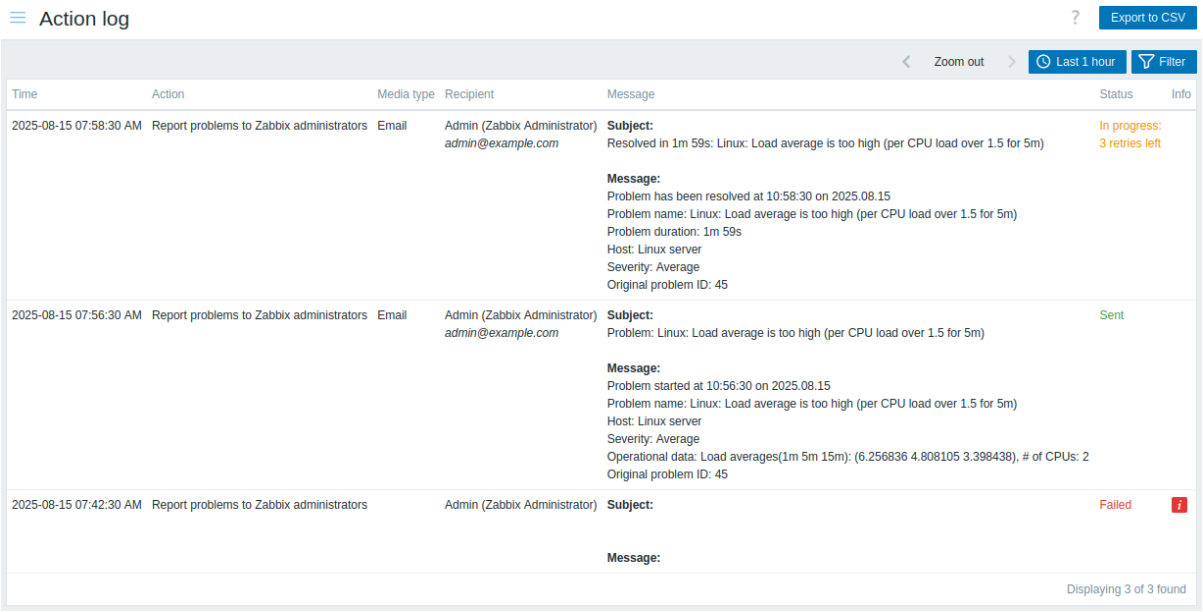
Sélecteur de période

Le sélecteur de période situé à côté du filtre permet de sélectionner les périodes fréquemment requises en un clic de souris. Pour plus d'informations, voir Sélecteurs de période et d'hôte.

6 Journal des actions

Aperçu

Dans *Rapports* → *Journal des actions*, vous pouvez afficher les détails des opérations (notifications, commandes distantes) exécutées dans le cadre d'une action.



Données affichées :

Colonne	Description
Heure	Horodatage de l'opération.
Action	Nom de l'action à l'origine des opérations.
Type de média	Type de média (par ex. E-mail, Jira, etc.) utilisé pour l'envoi d'une notification. Pour les opérations ayant exécuté des commandes distantes, cette colonne sera vide.
Destinataire	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 (adresse e-mail, nom d'utilisateur, etc.). Pour les opérations ayant exécuté des commandes distantes, cette colonne sera vide.
Message	Contenu du message, de la commande distante ou nom du script global. Une commande distante est séparée de l'hôte cible par le symbole deux-points : `<host>:<command>`. Par exemple, si la commande distante a été exécutée sur le serveur Zabbix, les informations auront le format suivant : `Zabbix server:<command>`.

Colonne	Description
Statut	Statut de l'opération : *En cours* - l'opération d'envoi d'une notification est en cours (le nombre restant de tentatives d'envoi par le serveur est également affiché) *Envoyée* - la notification a été envoyée *Exécutée* - la commande distante a été exécutée *Échec* - l'opération n'a pas été menée à bien
Info	Informations d'erreur (le cas échéant) concernant l'exécution de l'opération.

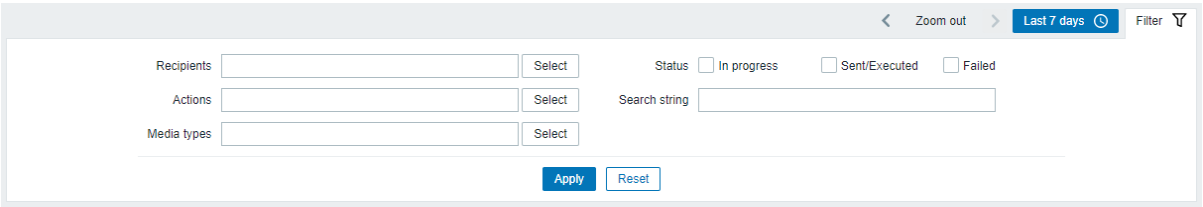
Boutons

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 du *Journal des actions*. Il peut être ouvert et réduit en cliquant sur l'onglet *Filtre* dans le coin supérieur droit de la page.



Vous pouvez utiliser le filtre pour affiner les enregistrements par destinataires des notifications, actions, types de média, statut, ou par contenu du message/de la commande distante (*Chaîne de recherche*). Pour de meilleures performances de recherche, les données sont recherchées avec les macros non résolues.

Sélecteur de période

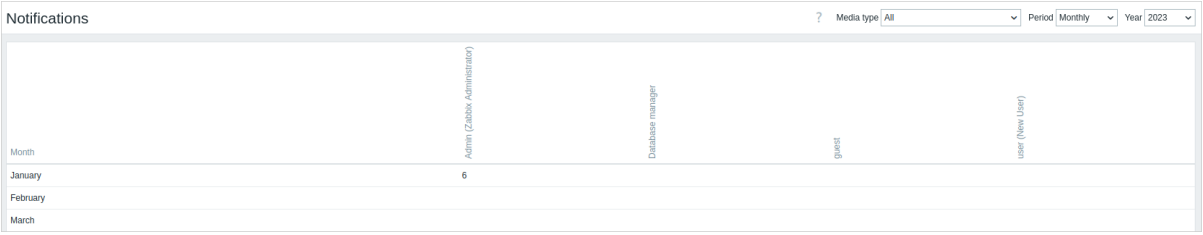
Le sélecteur de période situé à côté du filtre permet de sélectionner des périodes fréquemment requises en un clic de souris. Pour plus d'informations, voir Sélecteurs de période et d'hôte.

7 Notifications

Vue d'ensemble

Dans la section *Rapports* → *Notifications*, un rapport sur le nombre de notifications envoyées à chaque utilisateur est affiché.

À l'aide 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 des notifications envoyées.



Chaque colonne affiche les totaux pour un utilisateur du système.

6 Collecte de données

Aperçu

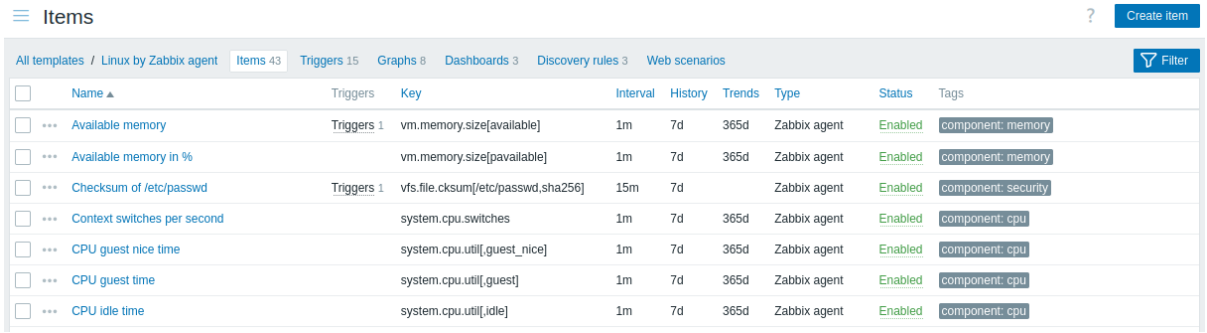
Ce menu comporte des sections liées à la configuration de la collecte de données.

1 Éléments

Aperçu

La liste des éléments d'un modèle peut être consultée depuis *Collecte de données* → *Modèles* en cliquant sur *Éléments* pour le modèle correspondant.

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.
Modèle	Modèle auquel l'élément appartient. Un clic 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. Un clic 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 s'affiche avant le nom de l'élément sous forme de lien gris. Un clic sur le lien du modèle ouvre la liste des éléments à ce niveau du modèle.
Déclencheurs	En passant la souris sur Déclencheurs, une infobulle s'affiche avec les déclencheurs associés à l'élément. Le nombre de déclencheurs s'affiche 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 de l'élément est affiché (agent Zabbix, agent SNMP, vérification simple, etc.).
Statut	Le statut de l'élément est affiché : *Activé* ou *Désactivé*. En cliquant sur le statut, vous pouvez le modifier, d'Activé à Désactivé (et inversement).
Tags	Les tags de l'élément sont affichés. Jusqu'à trois tags (paires nom:valeur) peuvent être affichés. S'il y a davantage de tags, un lien "..." s'affiche et permet de voir tous les tags au survol de la souris.

Pour configurer un nouvel élément, cliquez sur le bouton *Créer un élément* dans le coin supérieur droit.

Options de modification en masse

Les boutons sous la liste offrent quelques options de modification en masse :

- *Activer* - modifie le statut de l'élément en *Activé*.
- *Désactiver* - modifie le statut de l'élément en *Désactivé*.

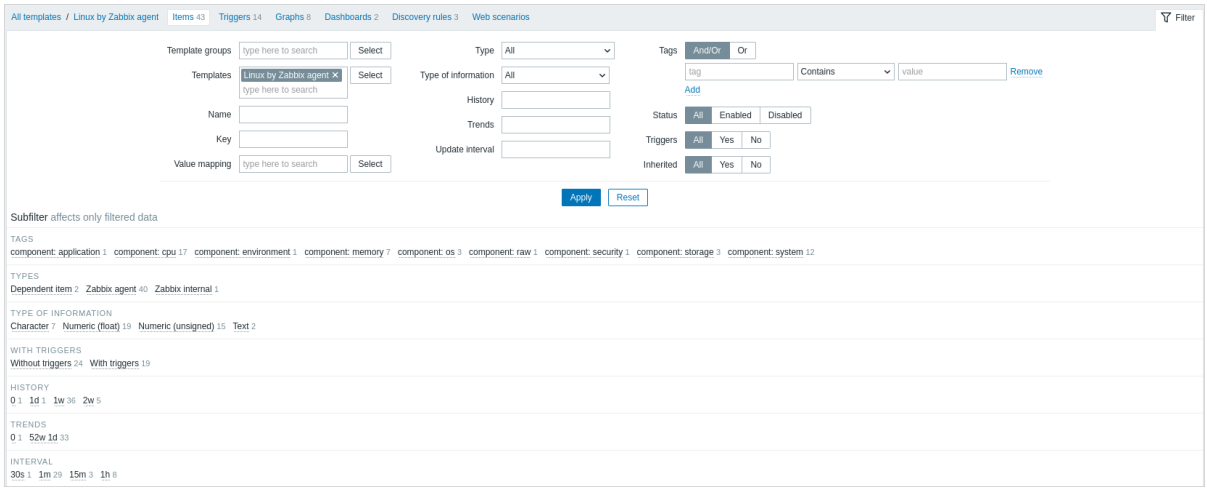
- *Copier* - copie les éléments vers d'autres hôtes ou modèles.
- *Mise à jour en masse* - met à jour plusieurs propriétés pour plusieurs éléments à la fois.
- *Supprimer* - supprime les éléments.

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 ceux 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 pourrez spécifier les critères de filtrage souhaités.

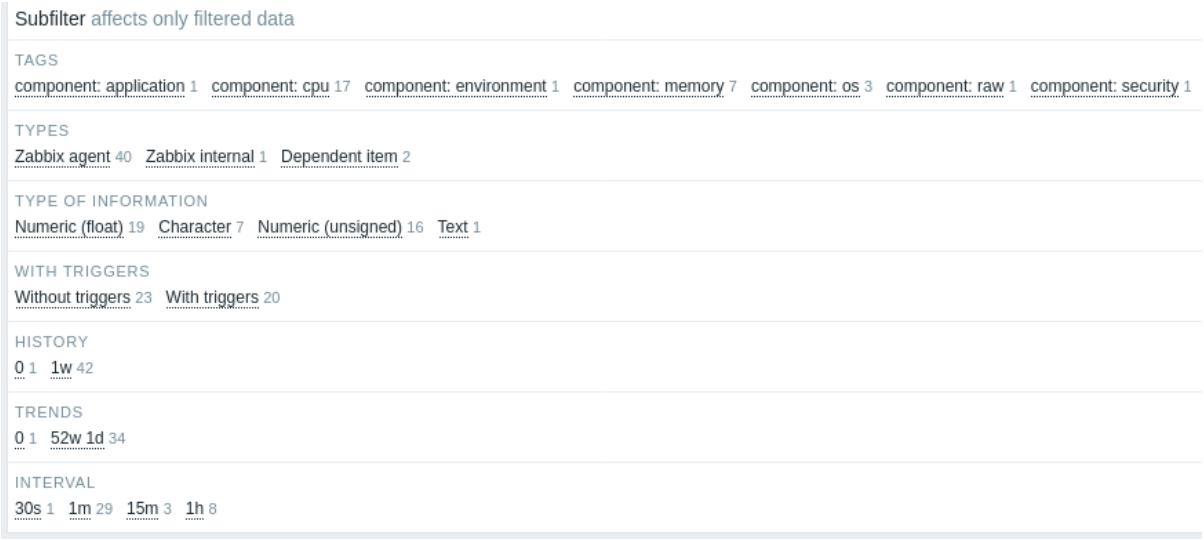


Paramètre	Description
Groupes de modèles	Filtrer par un ou plusieurs groupes de modèles. La spécification d'un groupe de modèles parent sélectionne implicitement tous les groupes imbriqués.
Modèles	Filtrer par un ou plusieurs modèles.
Nom	Filtrer par nom d'élément.
Clé	Filtrer par clé d'élément.
Correspondance de valeurs	Filtrer par la table de correspondance de valeurs utilisée. Ce paramètre n'est pas affiché si l'option *Modèles* est vide.
Type	Filtrer par type d'élément (agent Zabbix, agent SNMP, etc.).
Type d'information	Filtrer par type d'information (numérique non signé, flottant, etc.).
Historique	Filtrer selon la durée de conservation de l'historique des éléments.
Tendances	Filtrer selon la durée de conservation des tendances des éléments.
Intervalle de mise à jour	Filtrer par 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 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
Statut	Filtrer par statut d'élément - *Activé* ou *Désactivé*.
Déclencheurs	Filtrer les éléments avec (ou sans) déclencheurs.
Hérité	Filtrer les éléments hérités (ou non hérités) des modèles liés.

Utilisation du sous-filtre

Le sous-filtre permet de modifier davantage le filtrage défini dans le filtre principal.

Il contient des liens cliquables pour un accès rapide 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 cliquée, elle est mise en évidence avec un arrière-plan gris, et les éléments sont immédiatement filtrés (il n'est pas nécessaire de cliquer sur *Appliquer* dans le filtre principal). Cliquer sur une autre entité l'ajoute aux résultats filtrés. Cliquer de nouveau sur l'entité supprime le filtrage.



Les sous-filtres sont générés à partir des données filtrées, qui sont limitées à 1000 enregistrements. Si vous souhaitez voir davantage d'enregistrements dans le sous-filtre, vous devez augmenter la valeur du paramètre *Limit for search and filter results* (dans *Administration -> General -> GUI*).

Contrairement au filtre principal, le sous-filtre est mis à jour à chaque requête de rafraîchissement du tableau afin de toujours disposer d'informations à jour sur les options de filtrage disponibles et leurs compteurs.

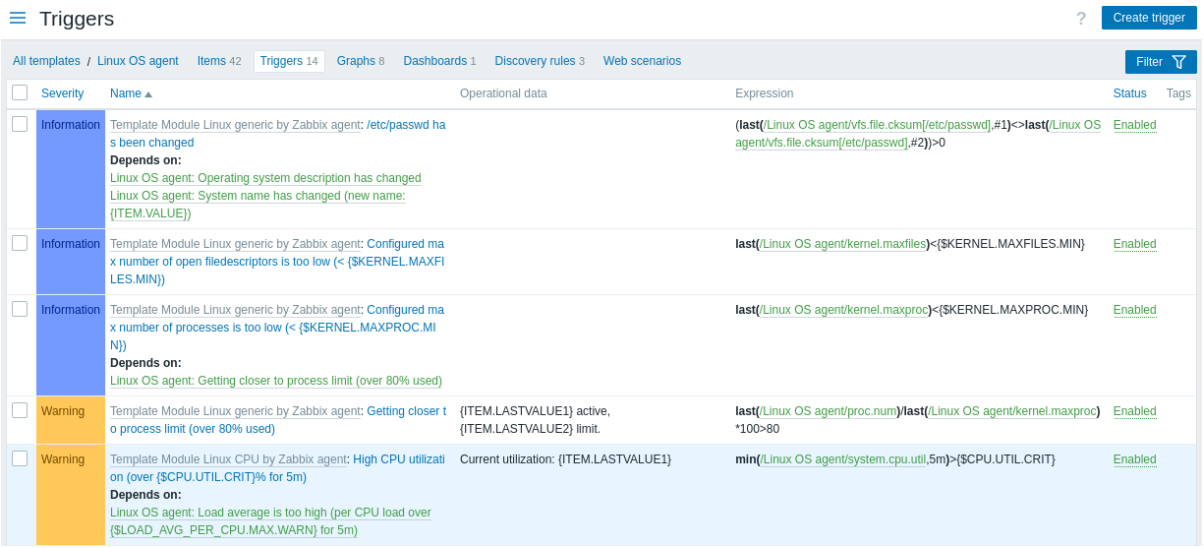
Le nombre d'entités affichées est limité à 100 horizontalement. S'il y en a davantage, une icône à trois points s'affiche à la fin ; elle n'est pas cliquable.

Un nombre à côté de chaque entité cliquable indique le nombre d'éléments regroupés dans celle-ci (sur la base des résultats du filtre principal). Lorsqu'une entité est cliquée, les nombres des autres entités disponibles s'affichent 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.

2 Déclencheurs

Aperçu

La liste des déclencheurs d'un modèle peut être consultée depuis *Collecte de données* → *Modèles* en cliquant sur *Déclencheurs* pour le modèle correspondant.



Données affichées :

Colonne	Description
Sévérité	La sévérité du déclencheur est affichée à la fois par son nom et par la couleur d'arrière-plan de la cellule.
Modèle	Modèle auquel appartient le déclencheur. Un clic sur le nom du modèle ouvre le formulaire de configuration du modèle. Cette colonne n'est affichée que 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. Un clic 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. Un clic 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 *Surveillance → Problèmes*.
Expression	L'expression du déclencheur est affichée. La partie modèle-élément de l'expression est affichée sous forme de lien, menant au formulaire de configuration de l'élément.
Statut	Le statut du déclencheur est affiché — *Activé* ou *Désactivé*. En cliquant sur le statut, vous pouvez le modifier — de Activé à Désactivé (et inversement).
Tags	Si un déclencheur contient des tags, le nom et la valeur du tag sont affichés dans cette colonne.

Pour configurer un nouveau déclencheur, cliquez sur le bouton *Créer un déclencheur* dans le coin supérieur droit.

Options de modification en masse

Les boutons situés sous la liste offrent plusieurs options de modification en masse :

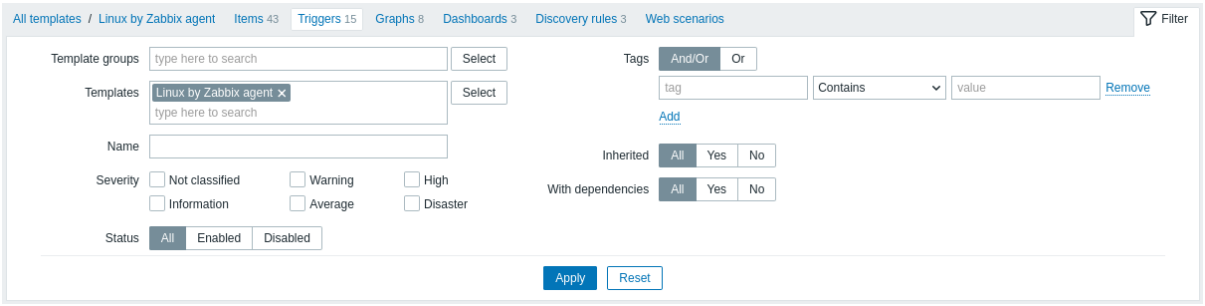
- *Activer* - modifier l'état du déclencheur en *Activé*
- *Désactiver* - modifier l'état du déclencheur en *Désactivé*
- *Copier* - copier les déclencheurs vers d'autres hôtes ou modèles
- *Mise à jour en masse* - mettre à jour plusieurs propriétés pour plusieurs déclencheurs à la fois
- *Supprimer* - supprimer les déclencheurs

Pour utiliser ces options, cochez les cases devant les déclencheurs correspondants, 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 pourrez spécifier les critères de filtrage souhaités.



Paramètre	Description
Groupes de modèles	Filtrer par un ou plusieurs groupes de modèles. La spécification d'un groupe de modèles parent sélectionne implicitement tous les groupes imbriqués.
Modèles	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.
Nom	Filtrer par nom de déclencheur.
Sévérité	Sélectionnez cette option pour filtrer par une ou plusieurs sévérités de déclencheur.
Statut	Filtrer par statut d'activation du déclencheur (Activé/Désactivé).

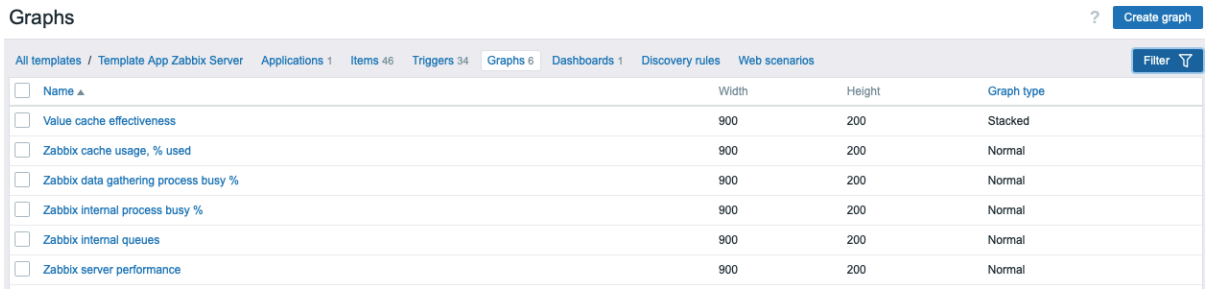
Paramètre	Description
Tags	Filtrer par nom et valeur de tag de déclencheur. Il est possible d'inclure comme d'exclure des tags et des valeurs de tag spécifiques. Plusieurs conditions peuvent être définies. La correspondance du nom de tag est toujours sensible à la casse. Plusieurs opérateurs sont disponibles pour chaque condition : **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) Il existe deux types de calcul 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 Les macros et les fonctions de macro sont prises en charge dans les champs de nom et de valeur de tag.
Hérité	Filtrer les déclencheurs hérités (ou non hérités) à partir des modèles liés.
Avec dépendances	Filtrer les déclencheurs avec (ou sans) dépendances.

3 Graphiques

Aperçu

La liste des graphiques personnalisés d'un modèle peut être consultée depuis *Collecte de données* → *Modèles* en cliquant sur *Graphiques* pour le modèle correspondant.

Une liste des graphiques existants s'affiche.



Données affichées :

Colonne	Description
Modèle	Modèle auquel le graphique appartient. Un clic sur le nom du modèle ouvre le formulaire de configuration du modèle. Cette colonne n'est affichée que si plusieurs modèles ou aucun modèle ne sont sélectionnés dans le filtre.
Nom	Nom du graphique personnalisé, affiché comme un lien bleu vers les détails du graphique. Un clic sur le lien du nom du graphique ouvre le formulaire de configuration du graphique. Si le graphique est hérité d'un autre modèle, le nom du modèle est affiché avant le nom du graphique sous forme de lien gris. Un clic sur le lien du modèle ouvre la liste des graphiques à ce niveau du modèle.
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*, *Secteur* ou *Éclaté*.

Pour configurer un nouveau graphique, cliquez sur le bouton *Créer un graphique* dans le coin supérieur droit.

Options d'édition en masse

Les boutons situés sous la liste offrent des options d'édition 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 respectifs, puis cliquez sur le bouton requis.

Utilisation du filtre

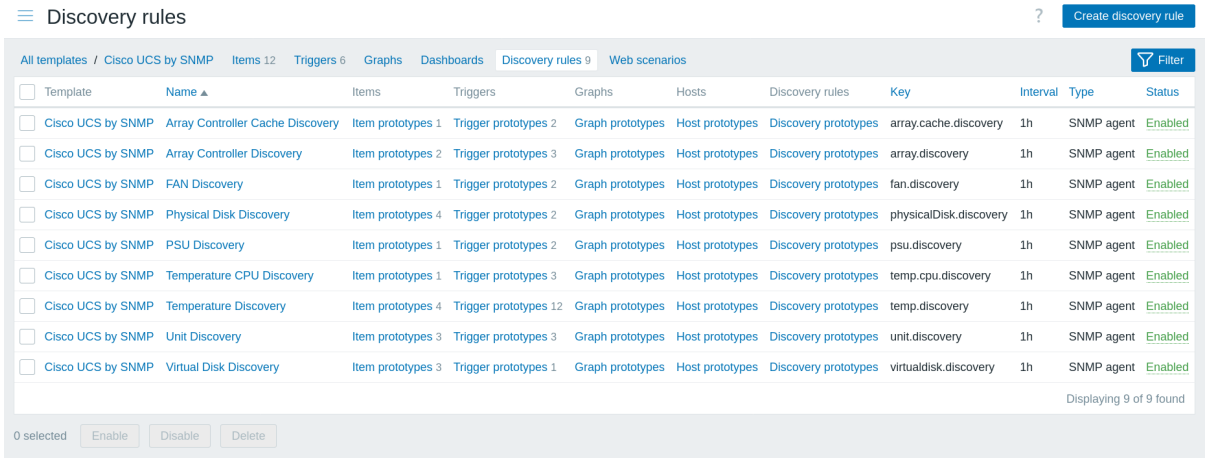
Vous pouvez filtrer les graphiques par groupe de modèles et par modèle. Pour de meilleures performances de recherche, les données sont recherchées avec les macros non résolues.

4 Règles de découverte

Aperçu

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 concerné.

Une liste des règles de découverte de bas niveau existantes s'affiche. Il est également possible de voir 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	Le modèle auquel appartient la règle de découverte. Un clic sur le nom du modèle ouvre le formulaire de configuration du modèle.
Nom	Nom de la règle, affiché sous forme de lien bleu. Un clic 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. Un clic 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 d'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.).
Statut	Le statut de la règle de découverte est affiché - *Activé* ou *Désactivé*. En cliquant sur le statut, vous pouvez le modifier - de Activé à Désactivé (et inversement).

Pour configurer une nouvelle règle de découverte de bas niveau, cliquez sur le bouton *Créer une règle de découverte* dans le coin supérieur droit.

Options d'édition en masse

Les boutons situés sous la liste offrent des options d'édition en masse :

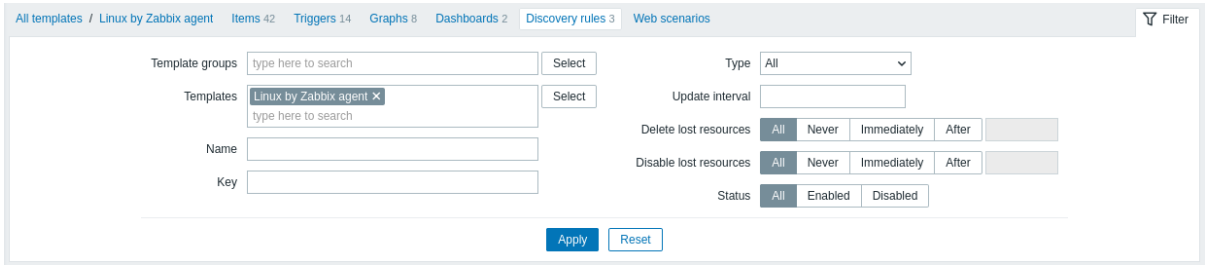
- *Activer* - changer l'état de la règle de découverte de bas niveau en *Activé*
- *Désactiver* - changer l'état de la règle de découverte de bas niveau en *Désactivé*
- *Supprimer* - supprimer les règles de découverte de bas niveau

Pour utiliser ces options, cochez les cases devant les règles de découverte respectives, 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.

L'icône *Filtre* est disponible dans le coin supérieur droit. En cliquant dessus, vous ouvrirez un filtre dans lequel vous pourrez 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.



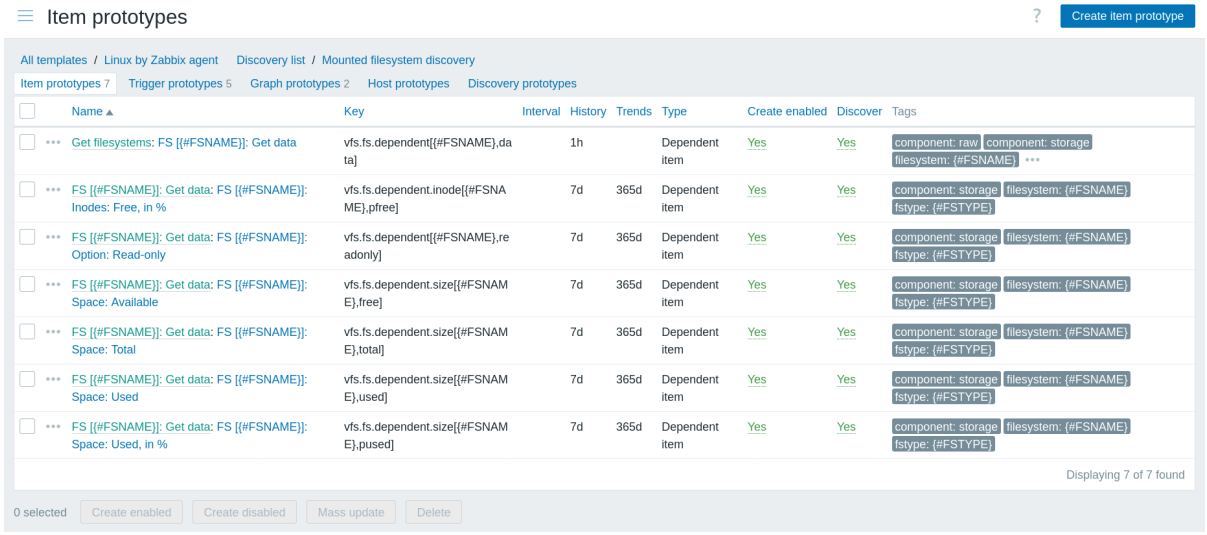
Paramètre	Description
Groupes de modèles	Filtrer par un ou plusieurs groupes de modèles. La spécification d'un groupe de modèles parent sélectionne implicitement tous les groupes imbriqués.
Modèles	Filtrer par un ou plusieurs modèles.
Nom	Filtrer par nom de règle de découverte.
Clé	Filtrer par clé d'élément de découverte.
Type	Filtrer par type d'élément de découverte.
Intervalle de mise à jour	Filtrer par intervalle de mise à jour. Non disponible pour les éléments Zabbix trapper et les éléments dépendants.
Supprimer les ressources perdues	Filtrer par période de *Suppression des ressources perdues*.
Désactiver les ressources perdues	Filtrer par période de *Désactivation des ressources perdues*.
Statut	Filtrer par statut de la règle de découverte (Tous/Activé/Désactivé).

1 Prototypes d'éléments

Aperçu

Dans cette section, les prototypes d'élément 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'élément serviront de base à la création de véritables éléments d'hôte lors de la découverte de bas niveau.



Données affichées :

Column	Description
Name	Le nom du prototype d'élément est affiché sous forme de lien bleu. Un clic 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. Un clic sur le lien du modèle ouvre la liste des prototypes d'élément au niveau du modèle lié. Si le prototype d'élément est du type « Dependent item », le nom de l'élément maître est affiché avant le nom du prototype d'élément en vert (comme dans le cas des prototypes d'élément 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 balises du prototype d'élément sont affichées.

Pour configurer un nouveau prototype d'élément, cliquez sur le bouton *Create item prototype* dans le coin supérieur droit.

Options de modification en masse

Les boutons situés sous la liste proposent plusieurs options de modification en masse :

- *Créer activés* - créer ces éléments en tant qu'*Activés*
- *Créer désactivés* - créer ces éléments en tant que *Désactivés*
- *Mise à jour en masse* - mettre à jour en masse ces prototypes d'éléments
- *Supprimer* - supprimer ces prototypes d'éléments

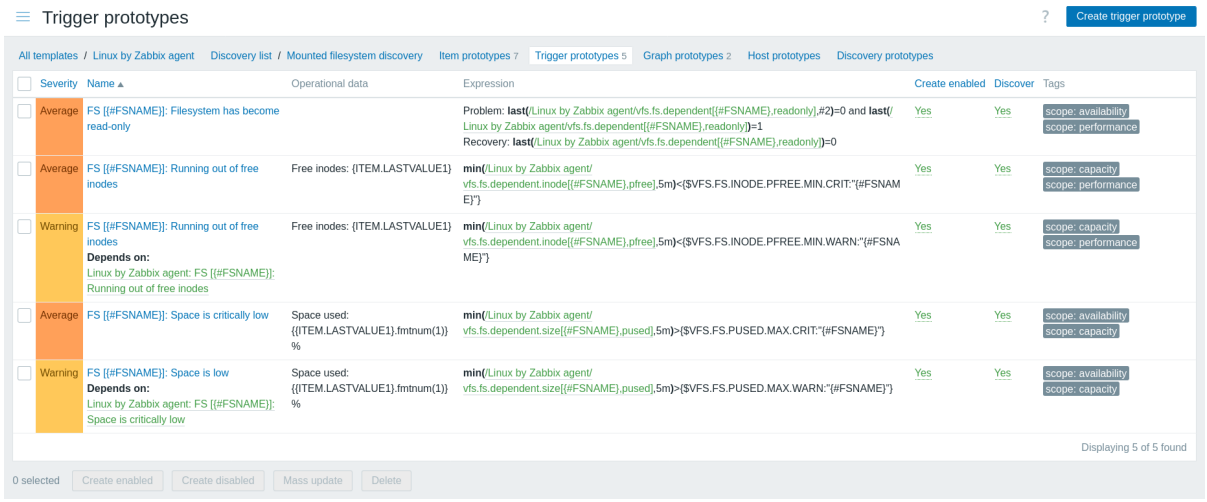
Pour utiliser ces options, cochez les cases devant les prototypes d'éléments correspondants, puis cliquez sur le bouton requis.

2 Prototypes de déclencheur

Aperçu

Dans cette section, les prototypes de déclencheur 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 déclencheur serviront de base à la création de véritables déclencheurs d'hôte lors de la découverte de bas niveau.



Données affichées :

Column	Description
Severity	La gravité du déclencheur est affichée à la fois par son nom et par la couleur d'arrière-plan de la cellule.
Name	Nom du prototype de déclencheur, affiché sous forme de lien bleu. Un clic 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. Un clic sur le lien du modèle ouvre la liste des prototypes de déclencheur au niveau du modèle lié.
Expression	L'expression du déclencheur est affichée. La partie modèle-élément de l'expression est affichée sous forme de lien, menant au formulaire de configuration de l'élément.
Operational data	Le format des données opérationnelles du déclencheur est affiché, contenant des chaînes arbitraires et des macros qui seront résolues dynamiquement dans *Monitoring → Problems*.
Create enabled	Créer le déclencheur sur la base de ce prototype comme suit : **Yes** - activé **No** - désactivé. Vous pouvez basculer entre « Yes » et « No » en cliquant dessus.
Discover	Découvrir le déclencheur 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 de déclencheur sont affichés.

Pour configurer un nouveau prototype de déclencheur, cliquez sur le bouton *Create trigger prototype* dans le coin supérieur droit.

Options de modification en masse

Les boutons situés sous la liste offrent plusieurs options de modification en masse :

- *Créer activés* - créer ces déclencheurs à l'état *Activé*
- *Créer désactivés* - créer ces déclencheurs à l'état *Désactivé*
- *Mise à jour en masse* - mettre à jour en masse ces prototypes de déclencheur
- *Supprimer* - supprimer ces prototypes de déclencheur

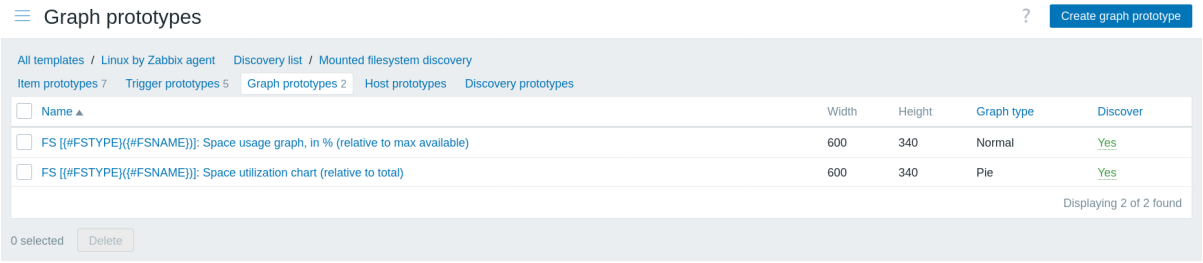
Pour utiliser ces options, cochez les cases devant les prototypes de déclencheur correspondants, puis cliquez sur le bouton requis.

3 Prototypes de graphiques

Aperçu

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 véritables graphiques d'hôte lors de la découverte de bas niveau.



Données affichées :

Colonne	Description
Nom	Nom du prototype de graphique, affiché sous forme de lien bleu. Un clic 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. Un clic sur le lien du modèle ouvre la liste des prototypes de graphiques au niveau du modèle lié.
Largeur	La largeur du prototype de graphique est affichée.
Hauteur	La hauteur du prototype de graphique est affichée.
Type	Le type du prototype de graphique est affiché : *Normal*, *Empilé*, *Secteurs* ou *Secteurs éclatés*.
Découvrir	Découvrir le graphique sur la base de ce prototype : **Oui** - découvrir **Non** - ne pas découvrir. Vous pouvez basculer entre « Oui » et « Non » en cliquant dessus.

Pour configurer un nouveau prototype de graphique, cliquez sur le bouton *Créer un prototype de graphique* dans le coin supérieur droit.

Options de modification en masse

Les boutons situés sous la liste offrent quelques options de modification en masse :

- *Supprimer* - supprimer ces prototypes de graphiques

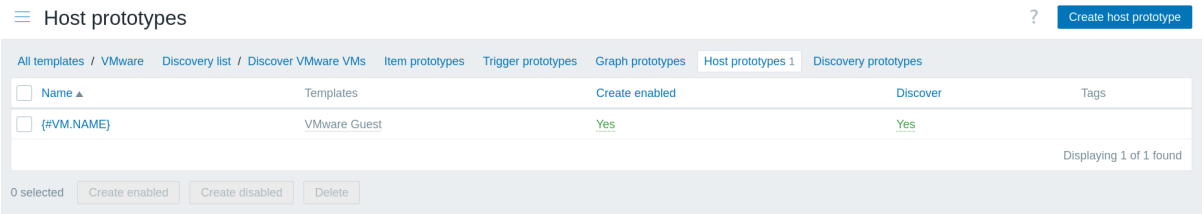
Pour utiliser ces options, cochez les cases devant les prototypes de graphiques correspondants, puis cliquez sur le bouton requis.

4 Prototypes d'hôtes

Aperçu

Dans cette section, les prototypes d'hôte 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'hôte serviront de base à la création de véritables hôtes lors de la découverte de bas niveau.



Données affichées :

Column	Description
Name	Nom du prototype d'hôte, affiché sous forme de lien bleu. Un clic 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. Un clic sur le lien du modèle ouvre la liste des prototypes d'hôte 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 le coin supérieur droit.

Options d'édition en masse

Les boutons situés sous la liste offrent des options d'édition en masse :

- *Créer activé* - créer ces hôtes comme *Activé*
- *Créer désactivé* - créer ces hôtes comme *Désactivé*
- *Supprimer* - supprimer ces prototypes d'hôtes

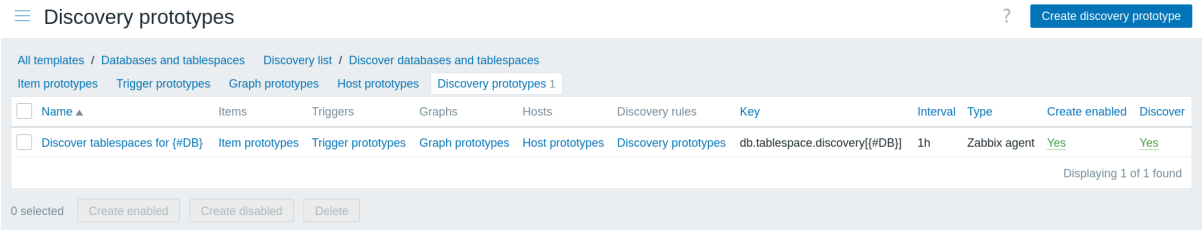
Pour utiliser ces options, cochez les cases devant les prototypes hôtes respectifs, puis cliquez sur le bouton requis.

5 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 véritables prototypes de découverte de la règle de découverte parente.



Données affichées :

Column	Description
Name	Nom du prototype de découverte, affiché sous forme de lien bleu. Un clic 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. Un clic 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 d'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*). En cliquant sur l'état, vous pouvez 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*). En cliquant sur l'état, vous pouvez le modifier, de Yes à No (et inversement).

Pour configurer un nouveau prototype de découverte, cliquez sur le bouton *Create discovery prototype* dans le coin supérieur droit.

Options de modification en masse

Les boutons situés sous la liste offrent quelques 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

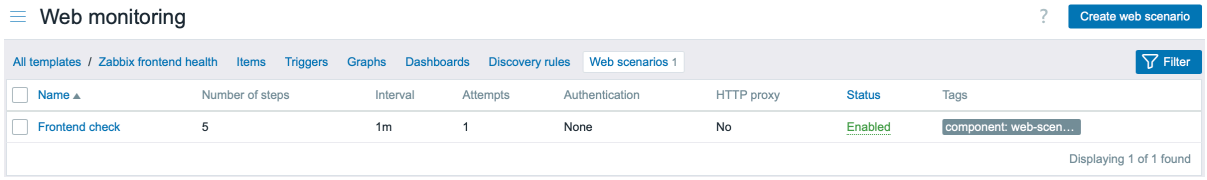
Pour utiliser ces options, cochez les cases devant les prototypes de découverte correspondants, puis cliquez sur le bouton requis.

5 Scénarios Web

Aperçu

La liste des scénarios web d'un modèle est accessible depuis *Collecte de données* → *Modèles* en cliquant sur *Web* pour le modèle correspondant.

Une liste des scénarios web existants s'affiche.



Données affichées :

Colonne	Description
Nom	Nom du scénario web. En cliquant sur le nom du scénario web, vous ouvrez le formulaire de configuration du scénario web. Si le scénario web est hérité d'un autre modèle, le nom du modèle s'affiche avant le nom du scénario web sous forme de lien gris. En cliquant sur le lien du modèle, vous ouvrez la liste des scénarios web à ce niveau de modèle.
Nombre d'étapes	Le nombre d'étapes contenues dans le scénario.
Intervalle de mise à jour	Fréquence d'exécution du scénario.
Tentatives	Le nombre de tentatives effectuées pour exécuter les étapes du scénario web.
Authentification	La méthode d'authentification affichée est l'une des suivantes : Basic, NTLM, Kerberos, Digest ou None.
Proxy HTTP	Affiche le proxy HTTP ou « No » s'il n'est pas utilisé.
Statut	Le statut du scénario web est affiché : *Activé* ou *Désactivé*. En cliquant sur le statut, vous pouvez 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 « ... » s'affiche et permet de voir tous les tags au survol de la souris.

Pour configurer un nouveau scénario web, cliquez sur le bouton *Créer un scénario web* dans le coin supérieur droit.

Options d'édition en masse

Les boutons situés sous la liste offrent des options d'édition en masse :

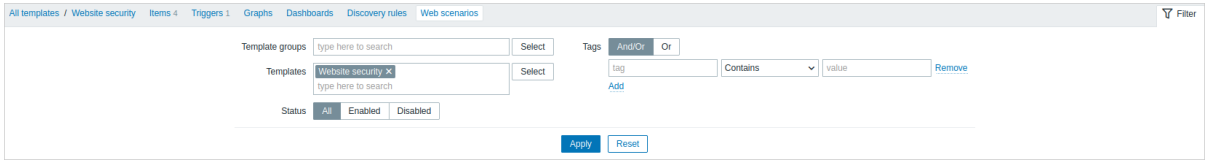
- *Activer* - changer le statut du scénario en *Activé*
- *Désactiver* - changer le statut du scénario en *Désactivé*
- *Supprimer* - supprimer les scénarios Web

Pour utiliser ces options, cochez les cases devant les scénarios Web respectifs, 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, 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, dans lequel vous pouvez filtrer les scénarios par groupe de modèles, modèle, statut et tags.

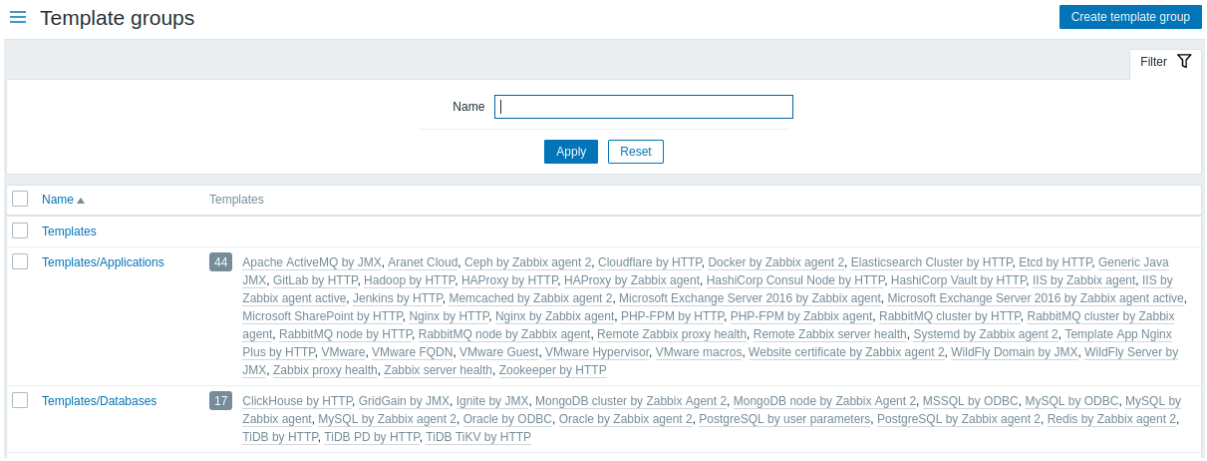


1 Groupes de modèles

Aperçu

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 :

Column	Description
Name	Nom du groupe de modèles. Un clic sur le nom du groupe ouvre le formulaire de configuration du groupe.
Templates	Nombre de modèles dans le groupe (affiché en gris), suivi de la liste des membres du groupe. Un clic sur le nom d'un modèle ouvre le formulaire de configuration du modèle. Un clic sur le nombre ouvre la liste des modèles de ce groupe.

Options de modification en masse

Pour supprimer plusieurs groupes de modèles à la fois, cochez les cases devant les groupes respectifs, puis cliquez sur le bouton **Supprimer** sous la liste.

Utilisation du filtre

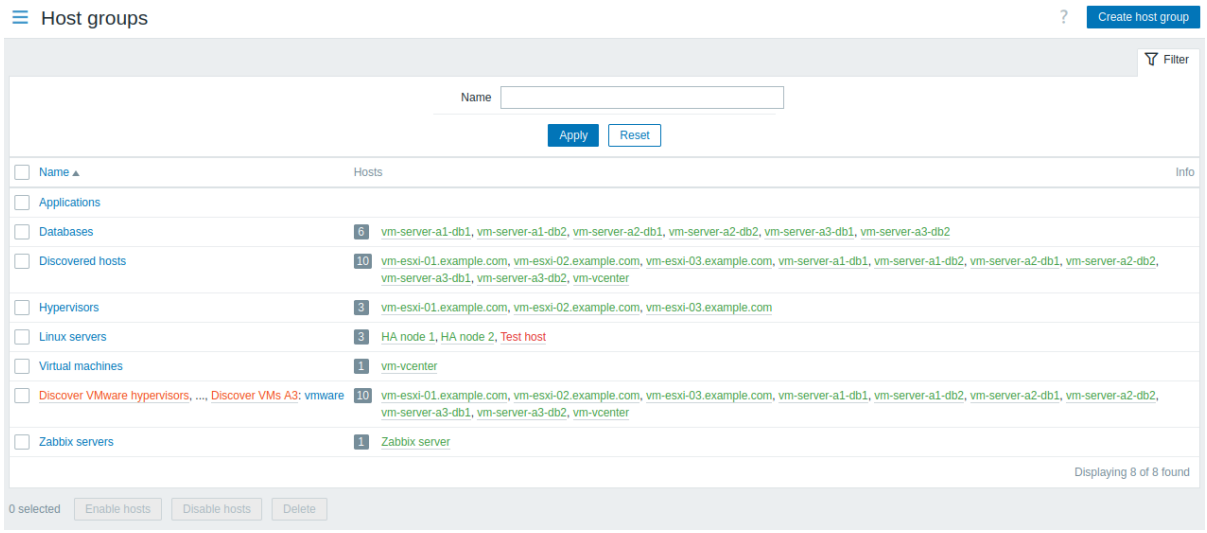
Vous pouvez utiliser le filtre pour afficher uniquement les groupes de modèles qui vous intéressent. Pour de meilleures performances de recherche, les données sont recherchées avec les macros non résolues.

2 Groupes d'hôtes

Aperçu

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 :

Colonne	Description
Nom	Nom du groupe d'hôtes. Un clic sur le nom du groupe ouvre le formulaire de configuration du groupe. Les groupes d'hôtes découverts sont affichés avec les noms des règles de découverte de bas niveau en préfixe. Un clic sur le nom de la règle LLD ouvre le formulaire de configuration du prototype d'hôte. Notez que les groupes d'hôtes découverts sont supprimés lorsque toutes les règles LLD qui les ont découverts sont supprimées.
Hôtes	Nombre d'hôtes dans le groupe (affiché en gris), suivi de la liste des membres du groupe. Un clic sur un nom d'hôte ouvre le formulaire de configuration de l'hôte. Un clic sur le nombre permet, dans la liste complète des hôtes, de filtrer ceux qui appartiennent au groupe.
Info	Les informations d'erreur (le cas échéant) concernant le groupe d'hôtes sont affichées.

Options d'édition de masse

Les boutons situés sous la liste offrent des options d'édition collective :

- *Activer les hôtes* - change le statut de tous les hôtes du groupe en "Supervisé"
- *Désactiver les hôtes* - modifie le statut de tous les hôtes du groupe en "Non supervisé".
- *Supprimer* - supprime les groupes d'hôtes

Pour utiliser ces options, cochez les cases devant les groupes d'hôtes respectifs, puis cliquez sur le bouton requis.

Utilisation de filtes

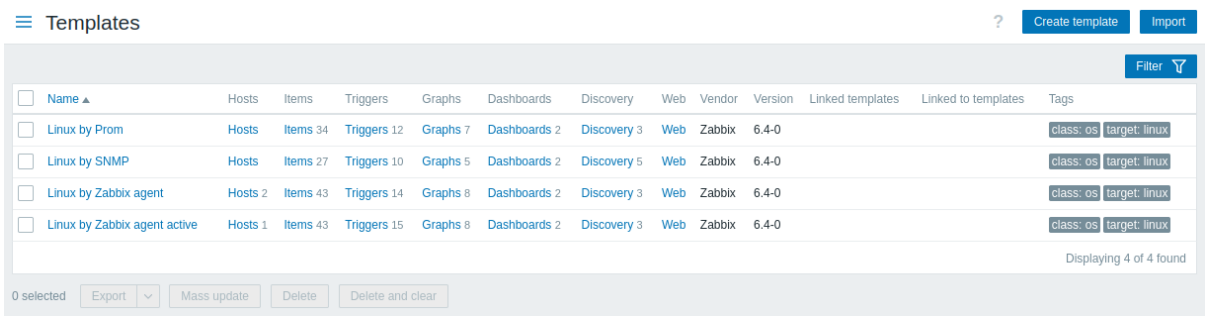
Vous pouvez utiliser le filtre pour afficher uniquement les groupes d'hôtes qui vous intéressent. Pour de meilleures performances de recherche, les données sont recherchées avec des macros non résolues.

3 Modèles

Aperçu

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.



Données affichées :

Column	Description
Name	Nom du modèle. Un clic sur le nom du modèle ouvre le formulaire de configuration du modèle.
Hosts	Nombre d'hôtes modifiables auxquels le modèle est lié ; les hôtes en lecture seule ne sont pas inclus. Un clic sur *Hosts* ouvre la liste des hôtes avec uniquement les hôtes liés au modèle filtrés.
Entities (Items, Triggers, Graphs, Dashboards, Discovery, Web)	Nombre des entités correspondantes dans le modèle (affiché en gris). Un clic sur le nom de l'entité permet, dans la liste complète de cette entité, de filtrer celles qui appartiennent au modèle.
Linked templates	Modèles liés au modèle.
Linked to templates	Modèles auxquels le modèle est lié.
Vendor, *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 la Template API. 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.
Tags	Tags du modèle, avec macros non résolues.

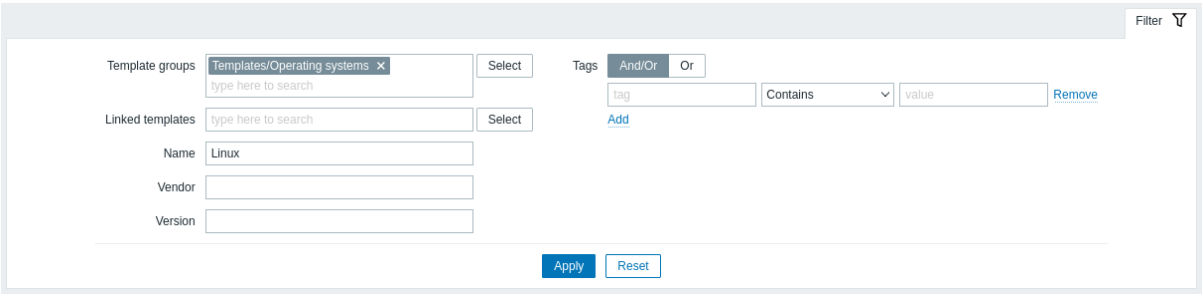
Pour configurer un nouveau modèle, cliquez sur le bouton *Create template* dans le coin supérieur droit.

Pour importer un modèle à partir d'un fichier YAML, XML ou JSON, cliquez sur le bouton *Import* dans le coin supérieur droit.

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, dans lequel vous pouvez filtrer les modèles par groupe de modèles, modèles liés, nom et tags.



Parameter	Description
Groupes de modèles	Filtrer par un ou plusieurs groupes de modèles. La spécification d'un groupe de modèles parent sélectionne implicitement tous les groupes imbriqués.
Modèles liés	Filtrer par modèles directement liés.
Nom	Filtrer par nom du modèle.
Éditeur	Filtrer par éditeur du modèle.
Version	Filtrer par version du modèle.

Parameter	Description
Tags	Filtrer par nom et valeur de tag du modèle. Le filtrage n'est possible que par les tags au niveau du modèle (pas ceux hérités). Il est possible d'inclure comme 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 : **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 dont 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 dont 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.

Options de modification en masse

Les boutons situés 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 effacer* - supprimer le modèle et ses entités liées des hôtes.

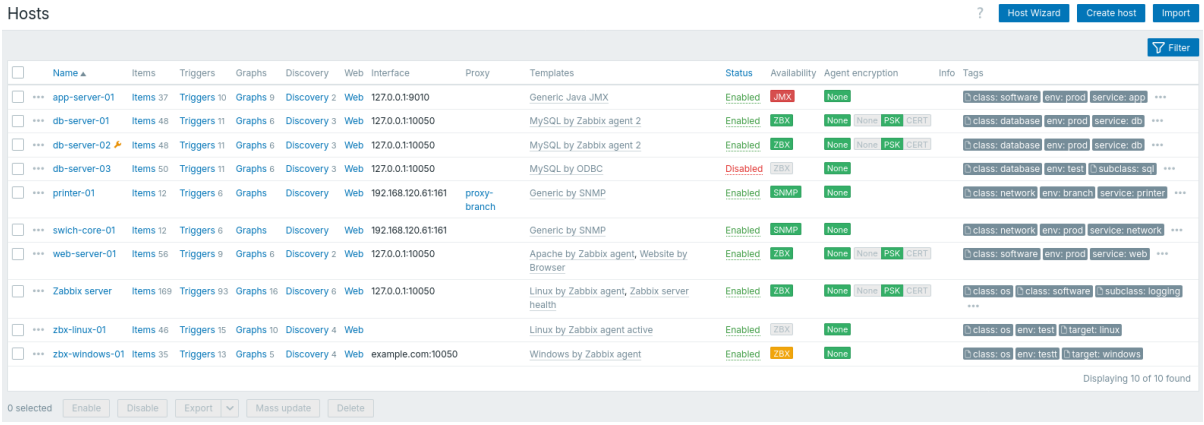
Pour utiliser ces options, cochez les cases devant les modèles respectifs, puis cliquez sur le bouton requis.

4 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.



Données affichées :

Column	Description
Menu de l'hôte	Cliquez sur l'icône à trois points pour ouvrir le menu de l'hôte.
Nom	Nom de l'hôte. Un clic sur le nom de l'hôte ouvre le formulaire de configuration de l'hôte. Une icône de clé orange après le nom de l'hôte indique que cet hôte est en maintenance. Les détails de maintenance sont affichés lorsque le pointeur de la souris est placé sur l'icône.

Column	Description
Entités (Éléments, Déclencheurs, Graphiques, Découverte, Web)	Un clic sur le nom de l'entité affichera 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 proxies attribués sont affichés dans cette colonne : <Proxy name> - l'hôte est surveillé par un proxy autonome (même si le proxy fait partie d'un groupe de proxies) ; <Proxy group name: proxy name> - l'hôte est surveillé par un groupe de proxies, et le serveur Zabbix a attribué un proxy pour surveiller l'hôte ; <Proxy group name> - l'hôte est surveillé par un groupe de proxies sans aucun proxy, ou si le serveur Zabbix n'a pas attribué de proxy pour surveiller l'hôte ; Nothing - l'hôte n'est surveillé ni par un proxy ni par un groupe de proxies. Cette colonne n'est affichée que si l'option de filtre *Surveillé par* est définie sur « Any », « Proxy » ou « Proxy group ».
Modèles	Les modèles liés à l'hôte sont affichés. Si d'autres modèles sont contenus dans le modèle lié, ils sont affichés entre parenthèses, séparés par une virgule. Un clic sur le nom d'un modèle ouvrira son formulaire de configuration.
Statut	Le statut de l'hôte est affiché - *Activé* ou *Désactivé*. En cliquant sur le statut, vous pouvez 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 statut.

Column	Description
Disponibilité	La disponibilité de l'hôte par interface configurée est affichée. Les icônes de disponibilité représentent l'état actuel des interfaces de l'hôte sur le serveur Zabbix. Par conséquent, si vous désactivez un hôte dans le frontend, 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 une liste de toutes les interfaces du même type avec les détails, le statut et les erreurs. Pour l'interface Agent, la fenêtre contextuelle affiche les interfaces (passives) et les contrôles actifs. Si un hôte n'a que des contrôles actifs, l'icône de l'interface Agent est affichée même si l'hôte n'a pas d'interface Agent configurée. La colonne est vide pour les hôtes sans interfaces. Le statut d'une interface d'hôte unique est déterminé par la connexion entre un élément activé utilisant l'interface et l'hôte. Le statut 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 façon dont le serveur Zabbix détermine le statut des interfaces, voir Statut d'interface inconnu et Paramètres d'interface d'hôte injoignable/non disponible. Le statut 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. Le statut 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 est disponible ou inconnue ; **Rouge** - toutes les interfaces ne sont pas disponibles ; **Gris** - au moins une interface est inconnue, mais aucune n'est non disponible. **Disponibilité des contrôles actifs.** Si au moins un contrôle actif est activé sur l'hôte, les contrôles actifs affectent également la disponibilité totale de l'interface Agent comme décrit ci-dessus. Pour déterminer la disponibilité des contrôles actifs, des messages heartbeat sont envoyés dans le thread de contrôle actif de l'agent. La fréquence des messages heartbeat est contrôlée par le paramètre `HeartbeatFrequency` dans la configuration de l'agent Zabbix ou de l'agent 2 (60 secondes par défaut, plage 0-3600). Les contrôles actifs sont considérés comme non disponibles lorsque le heartbeat du contrôle actif 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 contrôle actif, de sorte que la disponibilité de leurs hôtes reste inconnue.
Chiffrement de l'agent	Le statut de chiffrement des connexions vers l'hôte et depuis l'hôte est affiché : **Aucun** - aucun chiffrement ; **PSK** - utilisation d'une clé pré-partagée ; **Cert** - utilisation d'un certificat.
Info	Les informations d'erreur (le cas échéant) concernant l'hôte sont affichées.
Tags	Les tags de l'hôte avec macros non résolues.

Pour configurer un nouvel hôte, cliquez sur le bouton *Host Wizard* ou *Create host* dans le coin supérieur droit. Pour importer un hôte à partir d'un fichier YAML, XML ou JSON, cliquez sur le bouton *Import* dans le coin supérieur droit.

Options de modification en masse

Les boutons situés sous la liste offrent plusieurs options de modification en masse :

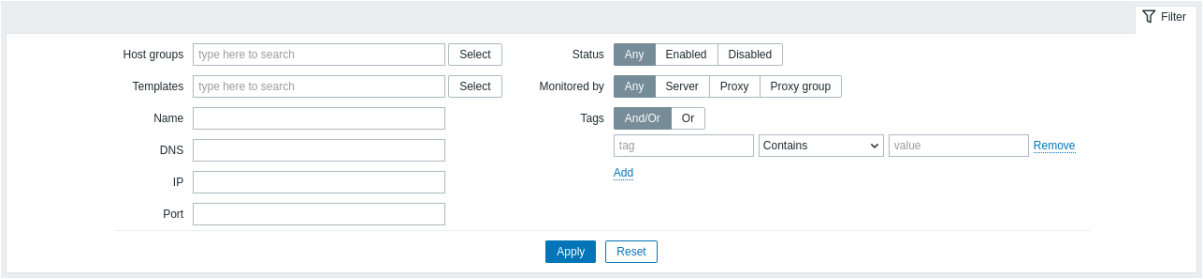
- *Activer* - modifie le statut de l'hôte en *Surveillé* ;
- *Désactiver* - modifie le statut de l'hôte en *Non surveillé* ;
- *Exporter* - exporte les hôtes vers un fichier YAML, XML ou JSON ;
- *Mise à jour en masse* - met à jour plusieurs propriétés pour plusieurs hôtes à la fois ;
- *Supprimer* - supprime les hôtes.

Pour utiliser ces options, cochez les cases devant les hôtes respectifs, puis cliquez sur le bouton requis.

Utilisation du filtre

Vous pouvez utiliser le filtre pour n'afficher que les hôtes 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 pourrez spécifier les critères de filtrage souhaités.



Paramètre	Description
Groupes d'hôtes	Filtrer par un ou plusieurs groupes d'hôtes. La spécification d'un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués.
Modèles	Filtrer par modèles liés.
Nom	Filtrer par nom visible de l'hôte.
DNS	Filtrer par nom DNS.
IP	Filtrer par adresse IP.
Port	Filtrer par numéro de port.
Statut	Filtrer par statut de l'hôte.
Supervisé par	Filtrer les hôtes supervisés par le serveur Zabbix, un proxy ou un groupe de proxys.
Proxys	Filtrer les hôtes supervisés par les proxys spécifiés ici. Ce champ n'est disponible que si « Proxy » est sélectionné dans le champ *Supervisé par*.
Groupes de proxys	Filtrer les hôtes supervisés par les groupes de proxys spécifiés ici. Ce champ n'est disponible que si « Groupe de proxys » est sélectionné dans le champ *Supervisé par*.
Tags	Filtrer par nom et valeur de tag d'hôte. Il est possible d'inclure comme 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 : **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). Il existe deux types de calcul 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.

Statut d'interface inconnu

Le serveur Zabbix affiche le statut « Inconnu » 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, par un autre proxy, ou par le serveur s'il était auparavant surveillé par un proxy.
- L'hôte est surveillé par un proxy qui semble être hors ligne (aucune mise à jour reçue du proxy pendant l'intervalle maximal de heartbeat, soit 1 heure).
- Tous les éléments de l'hôte utilisant 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 « Inconnu » après la synchronisation du cache de configuration du serveur Zabbix.

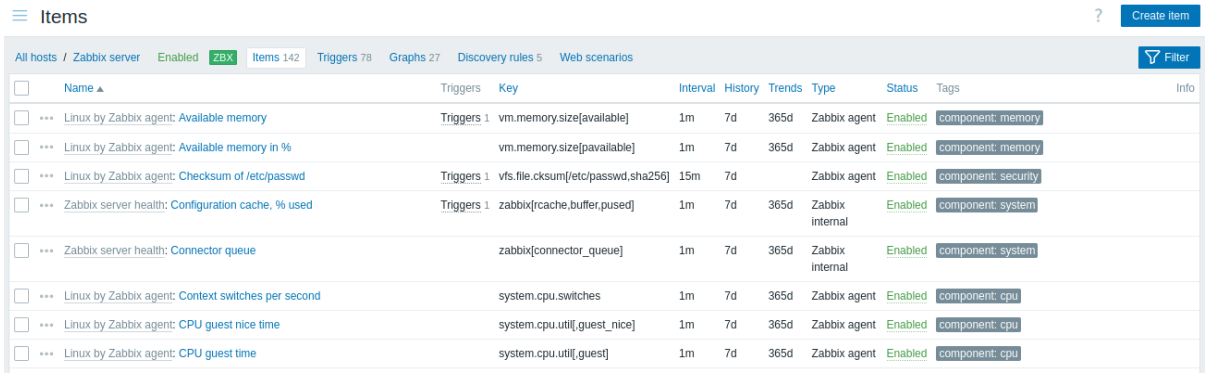
La disponibilité de l'interface (Disponible/Indisponible) sur les hôtes surveillés par des proxys est rétablie après la synchronisation du cache de configuration du proxy.

1 Éléments

Aperçu

La liste des éléments d'information d'un hôte est accessible depuis *Collecte de données* → *Hôtes* en cliquant sur *Éléments d'information* pour l'hôte correspondant.

Une liste des éléments d'information existants s'affiche.



Données affichées :

Colonne	Description
Menu de l'élément d'information	Cliquez sur l'icône à trois points pour ouvrir le menu de l'élément d'information.
Hôte	Hôte de l'élément d'information. Un clic 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 ne sont sélectionnés dans le filtre.
Nom	Nom de l'élément d'information affiché sous forme de lien bleu vers les détails de l'élément. Un clic sur le lien du nom de l'élément ouvre le formulaire de configuration de l'élément. Si l'élément d'information de l'hôte appartient à un modèle, le nom du modèle s'affiche avant le nom de l'élément sous forme de lien gris. Un clic 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. Un clic sur le nom de la règle de découverte ouvre la liste des prototypes d'élément.
Déclencheurs	En passant la souris sur Déclencheurs, une infobulle s'affiche avec les déclencheurs associés à l'élément. Le nombre de déclencheurs s'affiche en gris.
Clé	La clé de l'élément d'information est affichée.
Intervalle	La fréquence de la vérification est affichée. *Notez* que 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.).
Statut	Le statut de l'élément est affiché : *Activé*, *Désactivé* ou *Non pris en charge*. Vous pouvez modifier manuellement le statut en cliquant dessus : de Activé à Désactivé (et inversement) ; 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 statut.
Tags	Les tags de l'élément sont affichés. Jusqu'à trois tags (paires nom:valeur) peuvent être affichés. S'il y a davantage de tags, un lien "..." s'affiche et permet de voir tous les tags au survol de la souris.

Colonne	Description
Info	Si l'élément fonctionne correctement, aucune icône n'est affichée dans cette colonne. En cas d'erreurs, une icône carrée avec la lettre "i" est affichée. Survolez l'icône pour voir une infobulle contenant la description de l'erreur.

Pour configurer un nouvel élément d'information, cliquez sur le bouton *Créer un élément d'information* dans le coin supérieur droit.

Options de modification en masse

Les boutons situés sous la liste offrent plusieurs options de modification en masse :

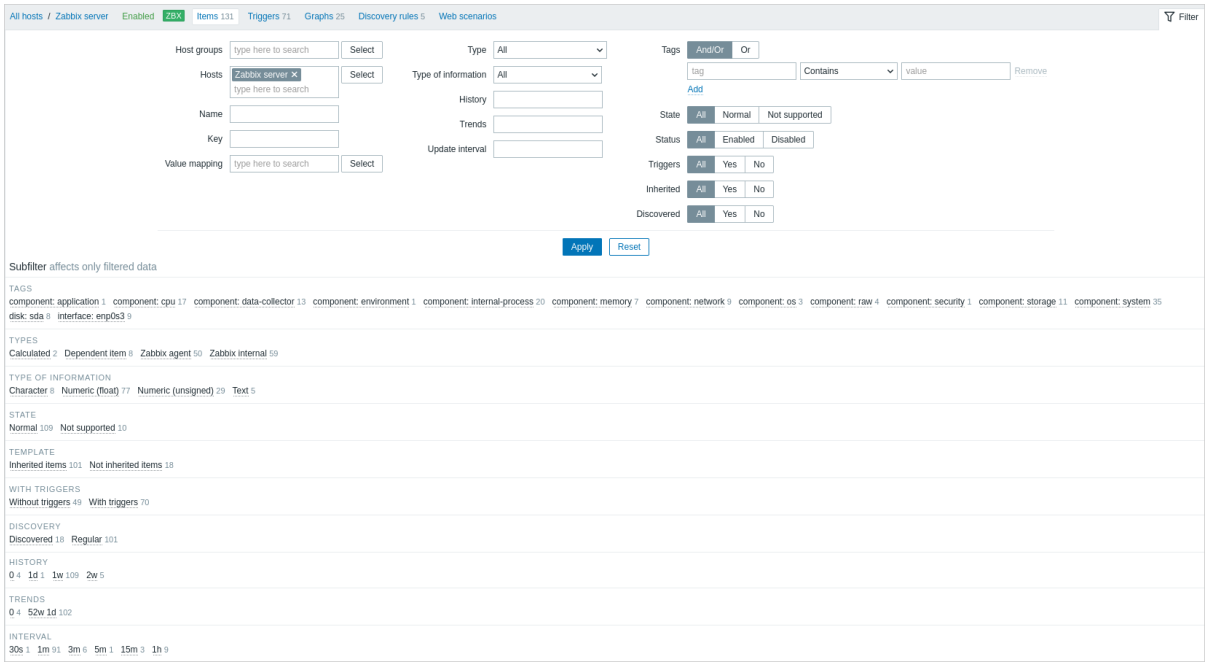
- *Activer* - modifie le statut de l'élément en *Activé*
- *Désactiver* - modifie le statut de l'élément en *Désactivé*
- *Exécuter maintenant* - exécute immédiatement une vérification pour obtenir de nouvelles valeurs d'élément. Pris en charge uniquement pour les vérifications **passives** (voir plus de détails). Notez que lors de la vérification immédiate des valeurs, le cache de configuration n'est pas mis à jour ; ainsi, les valeurs ne refléteront pas les modifications très récentes de la configuration de l'élément.
- *Effacer l'historique et les tendances* - supprime les données d'historique et de tendances des éléments.
- *Copier* - copie les éléments vers d'autres hôtes ou modèles.
- *Mise à jour en masse* - met à jour plusieurs propriétés pour plusieurs éléments à la fois.
- *Supprimer* - supprime 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 n'afficher que les éléments 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 pourrez spécifier les critères de filtrage souhaités.



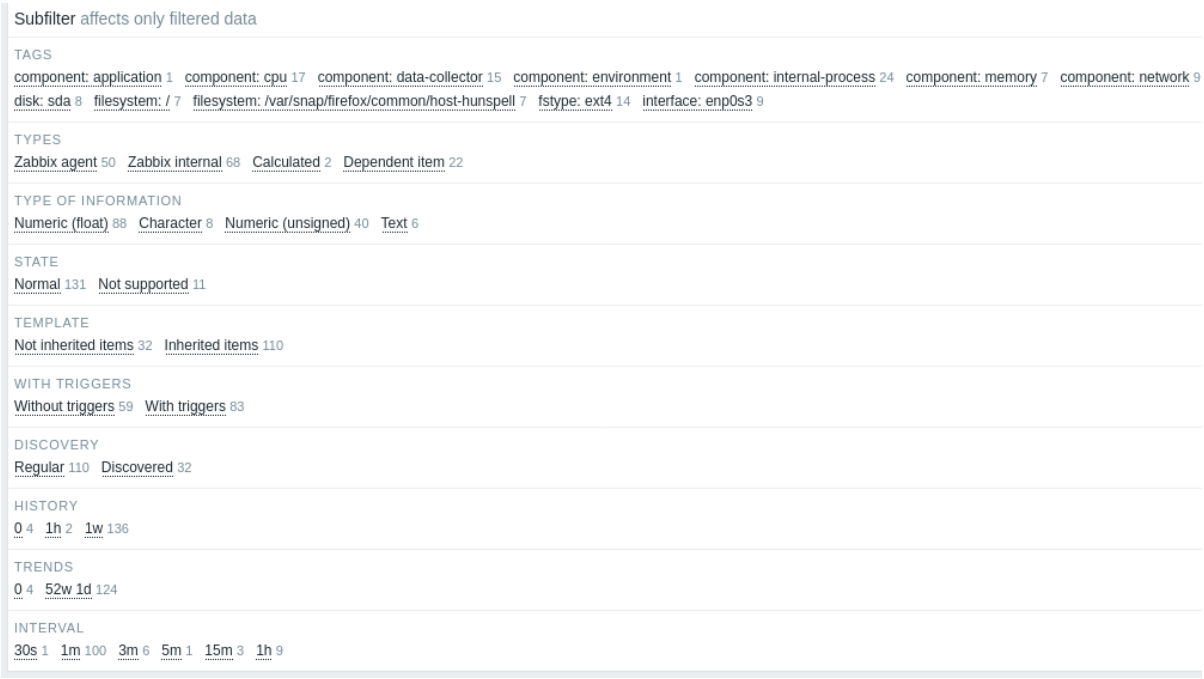
Paramètre	Description
Groupes d'hôtes	Filtrer par un ou plusieurs groupes d'hôtes. La spécification d'un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués. Les groupes d'hôtes contenant uniquement des modèles ne peuvent pas être sélectionnés.
Hôtes	Filtrer par un ou plusieurs hôtes.
Nom	Filtrer par nom d'élément.
Clé	Filtrer par clé d'élément.
Correspondance de valeurs	Filtrer selon la table de correspondance de valeurs utilisée. Ce paramètre n'est pas affiché si l'option *Hôtes* est vide.
Type	Filtrer par type d'élément (agent Zabbix, agent SNMP, etc.).

Paramètre	Description
Type d'information	Filtrer par type d'information (numérique non signé, flottant, etc.).
Historique	Filtrer selon la durée de conservation de l'historique des éléments.
Tendances	Filtrer selon la durée de conservation des tendances des éléments.
Intervalle de mise à jour	Filtrer par 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 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) Il existe deux types de calcul 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
État	Filtrer par état de l'élément - *Normal* ou *Non pris en charge*.
Statut	Filtrer par statut de l'élément - *Activé* ou *Désactivé*.
Déclencheurs	Filtrer les éléments avec (ou sans) déclencheurs.
Hérité	Filtrer les éléments hérités (ou non hérités) d'un modèle.
Découverte	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 du filtre principal.

Il contient des liens cliquables pour un accès rapide aux éléments associés. Les éléments sont associés par une entité commune : tag, type d'élément, état de l'élément, intervalle de mise à jour de l'élément, etc. Lorsqu'une entité est cliquée, elle est mise en évidence avec un arrière-plan gris, et les éléments sont immédiatement filtrés (il n'est pas nécessaire de cliquer sur *Appliquer* dans le filtre principal). Cliquer sur une autre entité l'ajoute aux résultats filtrés. Cliquer de nouveau sur l'entité supprime le filtrage.



Les sous-filtres sont générés à partir des données filtrées, qui sont limitées à 1000 enregistrements. Si vous souhaitez voir davantage d'enregistrements dans le sous-filtre, vous devez augmenter la valeur du paramètre *Limit for search and filter results* (dans *Administration -> General -> GUI*).

Contrairement au filtre principal, le sous-filtre est mis à jour à chaque requête de rafraîchissement du tableau afin de toujours disposer d'informations à jour sur les options de filtrage disponibles et leurs compteurs.

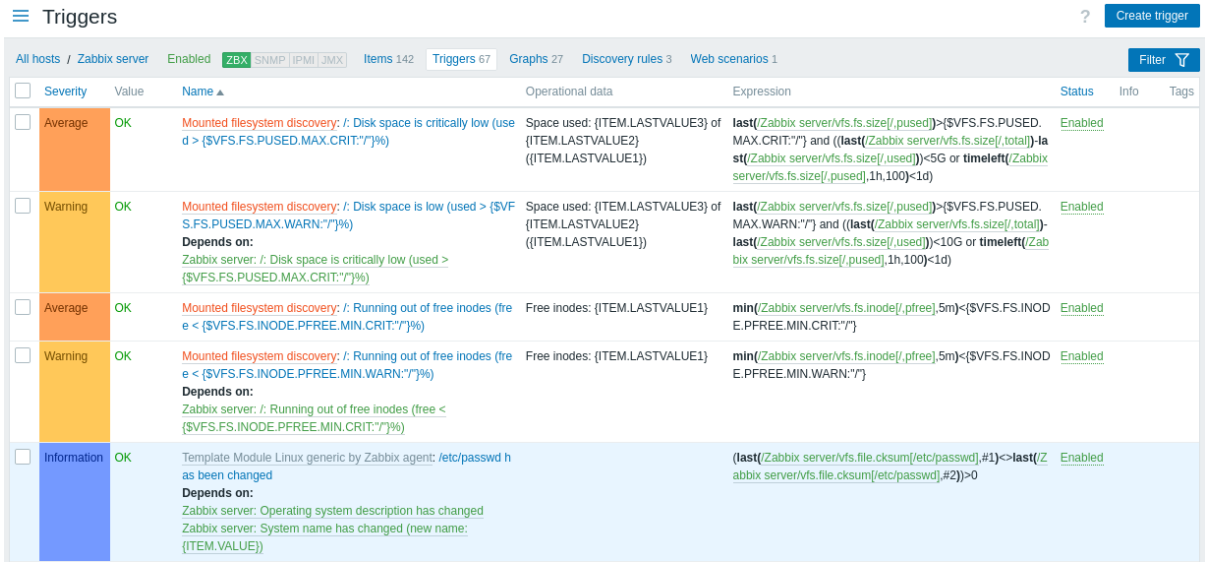
Le nombre d'entités affichées est limité à 100 horizontalement. S'il y en a davantage, une icône à trois points s'affiche à la fin ; elle n'est pas cliquable.

Un nombre à côté de chaque entité cliquable indique le nombre d'éléments regroupés dans celle-ci (sur la base des résultats du filtre principal). Lorsqu'une entité est cliquée, les nombres des autres entités disponibles s'affichent 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.

2 Déclencheurs

Aperçu

La liste des déclencheurs d'un hôte est accessible depuis *Collecte de données → Hôtes* en cliquant sur *Déclencheurs* pour l'hôte correspondant.



Données affichées :

Colonne	Description
Sévérité	La sévérité 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 de problème
Hôte	Hôte du déclencheur. Un clic sur le nom de l'hôte ouvre le formulaire de configuration de l'hôte. Cette colonne n'est affichée que si plusieurs hôtes ou aucun hôte ne sont sélectionnés dans le filtre.
Nom	Nom du déclencheur, affiché comme un lien bleu vers les détails du déclencheur. Un clic 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. Un clic 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. Un clic sur le nom de la règle de découverte ouvre la liste des prototypes de déclencheur.
Données opérationnelles	Définition des données opérationnelles du déclencheur, contenant des chaînes arbitraires et des macros qui seront résolues dynamiquement dans *Surveillance → Problèmes*.
Expression	L'expression du déclencheur est affichée. La partie hôte-élément de l'expression est affichée comme un lien menant au formulaire de configuration de l'élément.

Colonne	Description
Statut	Le statut du déclencheur est affiché - *Activé*, *Désactivé* ou *Inconnu*. En cliquant sur le statut, vous pouvez le modifier manuellement - de Activé à Désactivé (et inversement) ; de 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 statut.
Info	Si tout fonctionne correctement, aucune icône n'est affichée dans cette colonne. En cas d'erreurs, une icône carrée avec la lettre « i » est affichée. Survolez l'icône pour voir une infobulle contenant la description de l'erreur.
Tags	Si un déclencheur contient des tags, le nom et la valeur du tag sont affichés dans cette colonne.

Pour configurer un nouveau déclencheur, cliquez sur le bouton *Créer un déclencheur* dans le coin supérieur droit.

Options de modification en masse

Les boutons situés sous la liste offrent plusieurs options de modification en masse :

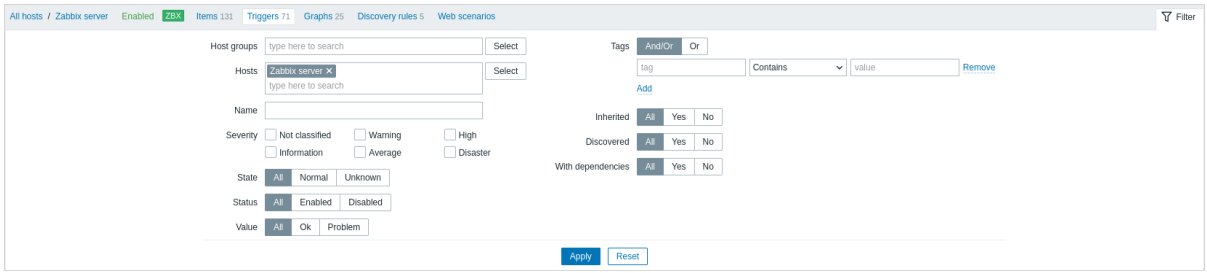
- *Activer* - modifie le statut du déclencheur en *Activé*.
- *Désactiver* - modifie le statut du déclencheur en *Désactivé*.
- *Copier* - copie les déclencheurs vers d'autres hôtes ou modèles.
- *Mise à jour en masse* - met à jour plusieurs propriétés pour plusieurs déclencheurs à la fois.
- *Supprimer* - supprime les déclencheurs.

Pour utiliser ces options, cochez les cases devant les déclencheurs correspondants, puis cliquez sur le bouton requis.

Utilisation du filtre

Vous pouvez utiliser le filtre pour n'afficher que 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 pourrez spécifier les critères de filtrage souhaités.



Paramètre	Description
Groupes d'hôtes	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.
Hôtes	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.
Nom	Filtrer par nom de déclencheur.
Sévérité	Sélectionnez cette option pour filtrer par une ou plusieurs sévérités de déclencheur.
État	Filtrer par état de l'expression du déclencheur (Normal/Inconnu).
Statut	Filtrer par statut d'activation du déclencheur (Activé/Désactivé).
Valeur	Filtrer par valeur du déclencheur.

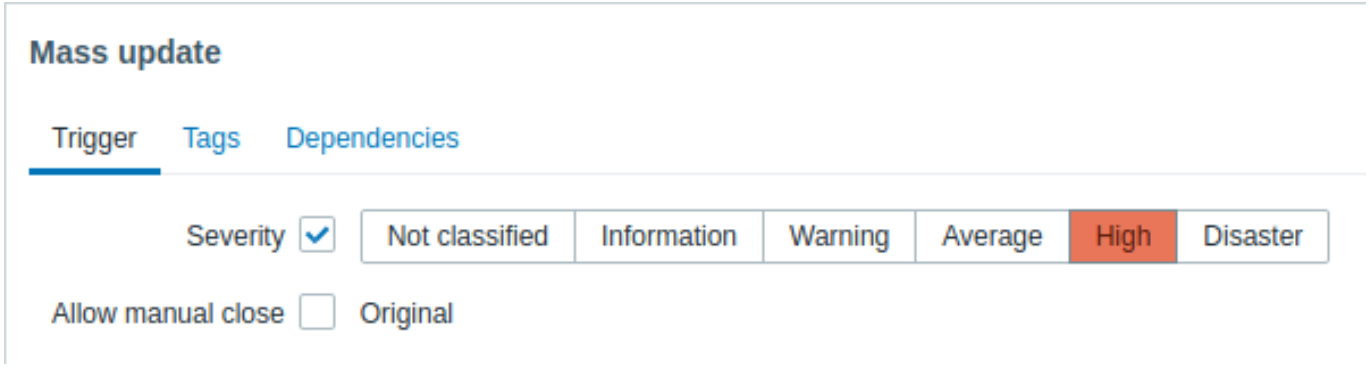
Paramètre	Description
Tags	Filtrer par nom et valeur de tag du déclencheur. Il est possible d'inclure ou d'exclure des tags et des valeurs de tag spécifiques. Plusieurs conditions peuvent être définies. La correspondance du nom de tag est toujours sensible à la casse. Plusieurs opérateurs sont disponibles 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 Les macros et les fonctions de macro sont prises en charge à la fois dans les champs de nom de tag et de valeur de tag.
Hérité	Filtrer les déclencheurs hérités (ou non hérités) d'un modèle.
Découvert	Filtrer les déclencheurs découverts (ou non découverts) par la découverte de bas niveau.
Avec dépendances	Filtrer les déclencheurs avec (ou sans) dépendances.

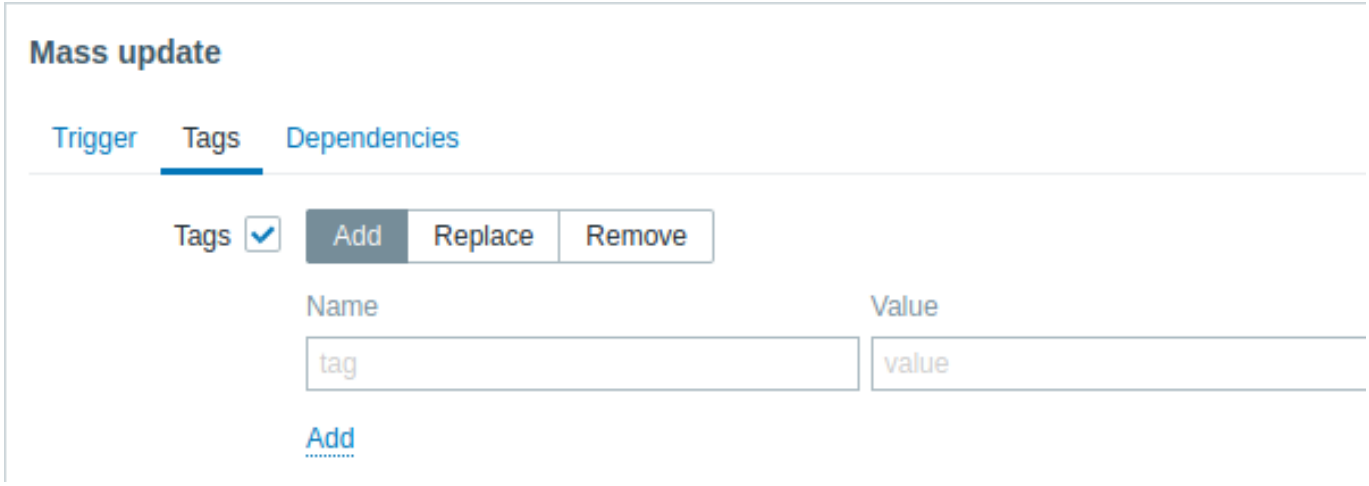
Utilisation de la mise à jour en masse

La mise à jour en masse vous permet de modifier certains attributs pour plusieurs déclencheurs à la fois, vous évitant ainsi d'ouvrir chaque déclencheur individuellement pour le modifier.

Pour mettre à jour certains 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*, *Tags* ou *Dépendances*)
- Cochez les cases des attributs à mettre à jour

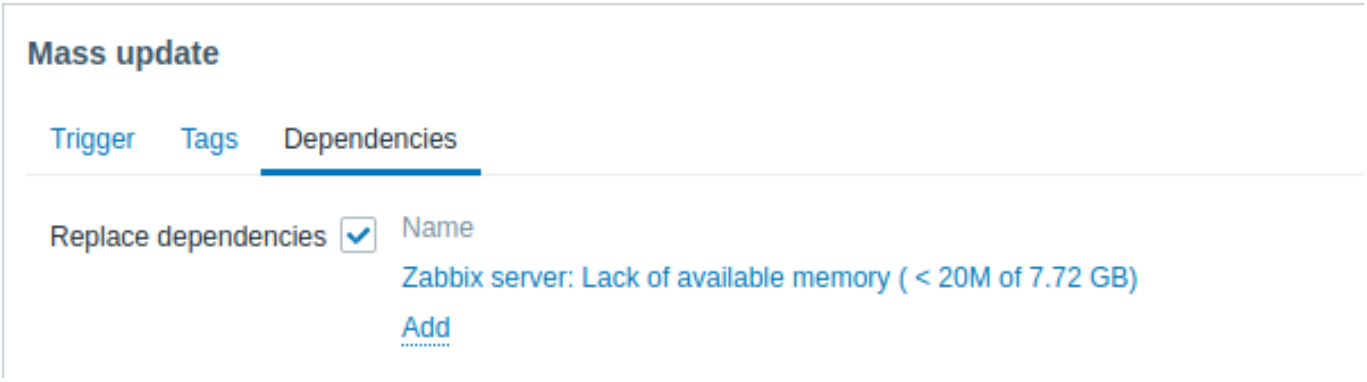




Les options suivantes sont disponibles lors de la sélection du bouton correspondant pour la mise à jour des tags :

- *Ajouter* - permet d'ajouter de nouveaux tags aux déclencheurs ;
- *Remplacer* - supprimera tous les tags existants du déclencheur et les remplacera par celui ou ceux spécifiés ci-dessous ;
- *Supprimer* - supprimera les tags spécifiés des déclencheurs.

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 déclencheur.



Remplacer les dépendances - supprimera toutes les dépendances existantes du déclencheur et les remplacera par celle(s) spécifiée(s).

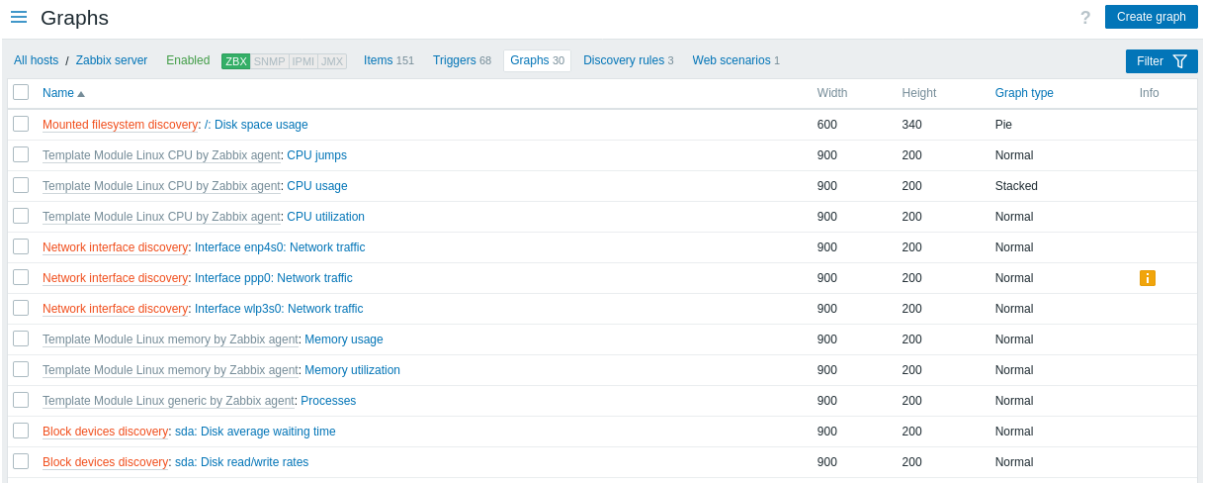
Cliquez sur *Mettre à jour* pour appliquer les modifications.

3 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 correspondant.

Une liste des graphiques existants s'affiche.



Données affichées :

Colonne	Description
Hôte	Hôte du graphique. Un clic 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 ne sont sélectionnés dans le filtre.
Nom	Nom du graphique personnalisé, affiché comme un lien bleu vers les détails du graphique. Un clic 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 s'affiche avant le nom du graphique sous forme de lien gris. Un clic 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. Un clic sur le nom de la règle de découverte ouvre la liste des prototypes de graphiques.

Colonne	Description
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 *Éclaté*.
Info	Si le graphique fonctionne correctement, aucune icône n'est affichée dans cette colonne. En cas d'erreur, une icône carrée avec la lettre « i » est affichée. Survolez l'icône pour voir une infobulle contenant la description de l'erreur.

Pour configurer un nouveau graphique, cliquez sur le bouton *Créer un graphique* dans le coin supérieur droit.

Options d'édition en masse

Les boutons situés sous la liste offrent des options d'édition 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 respectifs, puis cliquez sur le bouton requis.

Utilisation du filtre

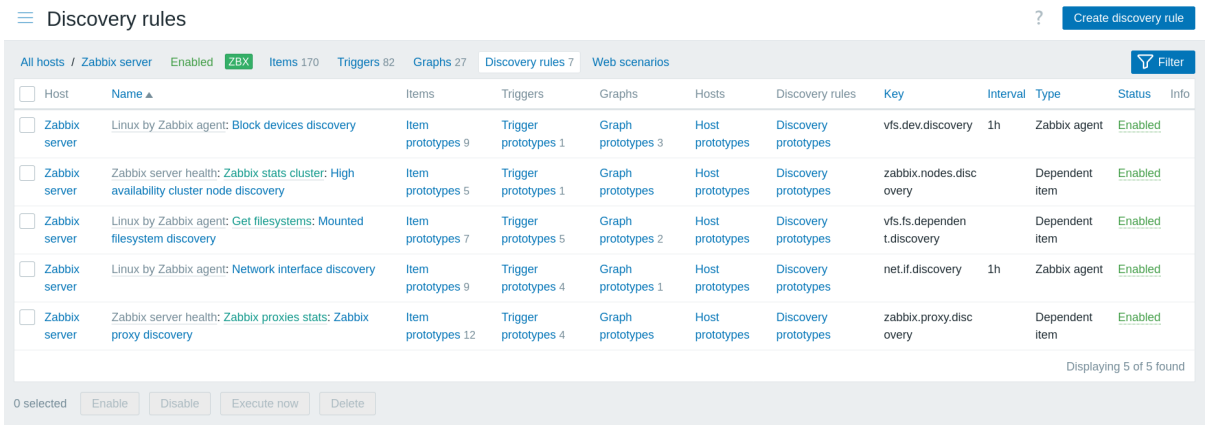
Vous pouvez filtrer les graphiques par groupe d'hôtes et par hôte. Pour de meilleures performances de recherche, les données sont recherchées avec les macros non résolues.

4 Règles de découverte

Aperçu

La liste des règles de découverte de bas niveau d'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 de voir 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. Un clic sur le nom de l'hôte ouvre le formulaire de configuration de l'hôte. En l'absence de nom visible de l'hôte, le nom technique de l'hôte est affiché.
Nom	Nom de la règle, affiché sous forme de lien bleu. Un clic 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. Un clic 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.

Colonne	Description
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 d'élément utilisée pour la découverte est affichée.
Intervalle	La fréquence d'exécution de la découverte est affichée. Veuillez noter que la découverte peut également être exécutée immédiatement en cliquant sur le bouton *Exécuter maintenant* sous la liste.
Type	Le type d'élément utilisé pour la découverte est affiché (agent Zabbix, agent SNMP, etc.).
Statut	Le statut de la règle de découverte est affiché : *Activé*, *Désactivé* ou *Non pris en charge*. En cliquant sur le statut, vous pouvez le modifier : de Activé à Désactivé (et inversement) ; de Non pris en charge à Désactivé (et inversement).
Info	Si tout est correct, aucune icône n'est affichée dans cette colonne. En cas d'erreurs, une icône carrée avec la lettre "i" est affichée. Survolez l'icône pour voir 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 le coin supérieur droit.

Options de modification en masse

Les boutons situés sous la liste offrent plusieurs options de modification en masse :

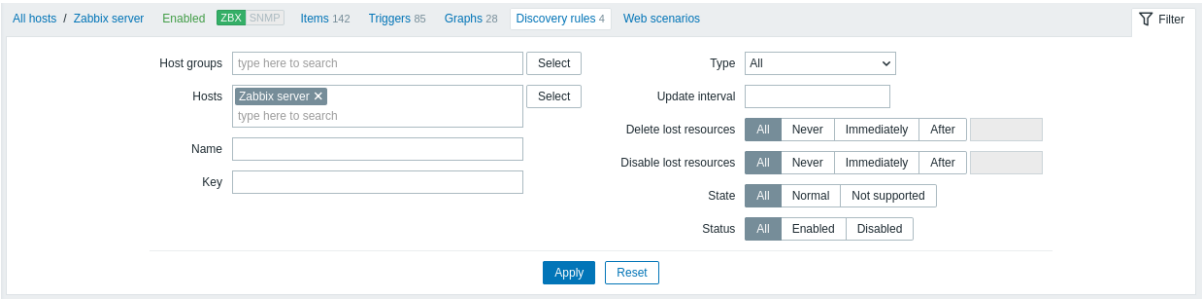
- *Activer* - modifie le statut de la règle de découverte de bas niveau en *Activé*.
- *Désactiver* - modifie le statut de la règle de découverte de bas niveau en *Désactivé*.
- *Exécuter maintenant* - exécute immédiatement la découverte selon les règles de découverte. Voir plus de détails. Notez que lors de l'exécution immédiate de la découverte, le cache de configuration n'est pas mis à jour ; le résultat ne reflétera donc pas les modifications très récentes apportées à la configuration des règles de découverte.
- *Supprimer* - supprime les règles de découverte de bas niveau.

Pour utiliser ces options, cochez les cases devant les règles de découverte respectives, 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, dans lequel vous pouvez filtrer les règles de découverte par groupe d'hôtes, hôte, nom, clé d'élément, type d'élément et autres paramètres.



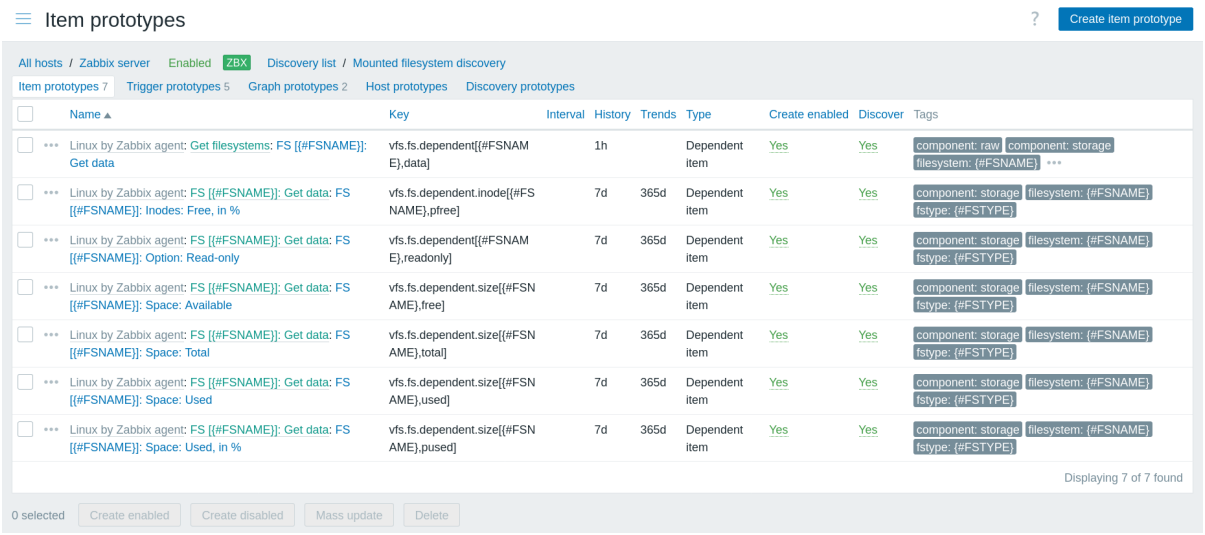
Paramètre	Description
Host groups	Filtrer par un ou plusieurs groupes d'hôtes. La spécification d'un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués.
Hosts	Filtrer par un ou plusieurs hôtes.
Name	Filtrer par nom de règle de découverte.
Key	Filtrer par clé d'élément de découverte.
Type	Filtrer par type d'élément de découverte.
Update interval	Filtrer par intervalle de mise à jour. Non disponible pour les éléments Zabbix trapper et les éléments dépendants.
Delete lost resources	Filtrer par période de *Delete lost resources*.

Paramètre	Description
Disable lost resources	Filtrer par période de *Disable lost resources*.
SNMP OID	Filtrer par OID SNMP. Disponible uniquement si *SNMP agent* est sélectionné comme type.
State	Filtrer par état de la règle de découverte (Tous/Normal/Non supporté).
Status	Filtrer par statut de la règle de découverte (Tous/Activé/Désactivé).

1 Prototypes d'éléments

Aperçu

Dans cette section, les prototypes d'élément d'une règle de découverte de bas niveau sur l'hôte sont affichés. Les prototypes d'élément constituent la base des éléments réels de l'hôte qui sont créés lors de la découverte de bas niveau.



Données affichées :

Column	Description
Name	Le nom du prototype d'élément est affiché sous forme de lien bleu. Un clic 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. Un clic sur le lien du modèle ouvre la liste des prototypes d'élément au niveau du modèle. Si le prototype d'élément est du type « Dependent item », le nom de l'élément maître est affiché en vert avant le nom du prototype d'élément.
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 balises du prototype d'élément sont affichées.

Pour configurer un nouveau prototype d'élément, cliquez sur le bouton *Create item prototype* dans le coin supérieur droit.

Options de modification en masse

Les boutons situés sous la liste offrent plusieurs options de modification en masse :

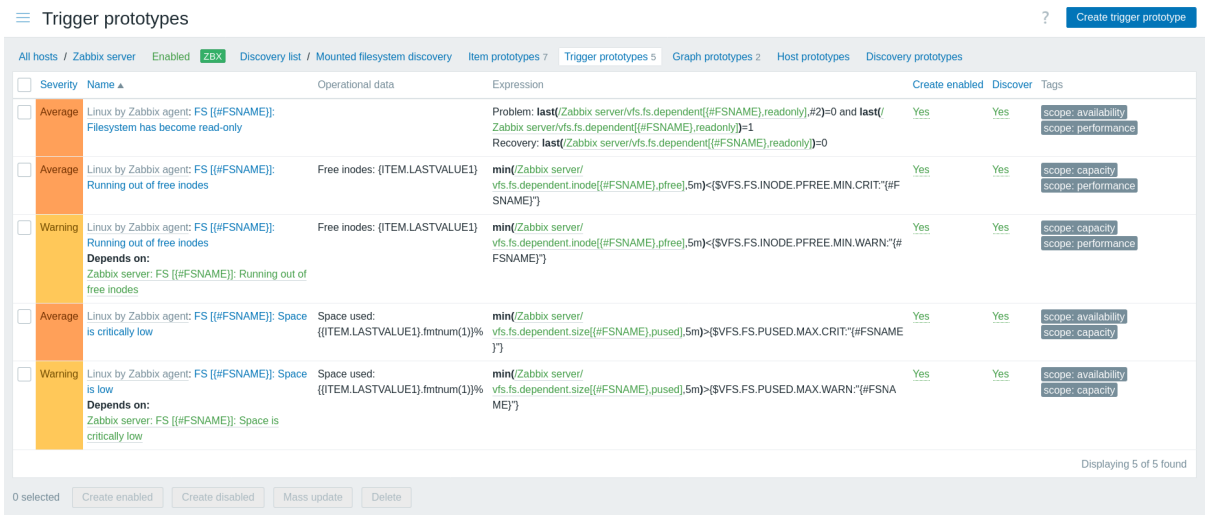
- *Create enabled* - créer ces prototypes d'éléments en tant qu'*Activé*
- *Create disabled* - créer ces prototypes d'éléments en tant que *Désactivé*
- *Mass update* - mettre à jour en masse ces prototypes d'éléments
- *Delete* - supprimer ces prototypes d'éléments

Pour utiliser ces options, cochez les cases devant les prototypes d'éléments correspondants, puis cliquez sur le bouton requis.

2 Prototypes de déclencheurs

Aperçu

Dans cette section, les prototypes de déclencheur d'une règle de découverte de bas niveau sur l'hôte sont affichés. Les prototypes de déclencheur constituent la base des déclencheurs réels de l'hôte, créés lors de la découverte de bas niveau.



Données affichées :

Column	Description
Name	Nom du prototype de déclencheur, affiché sous forme de lien bleu. Un clic 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. Un clic sur le lien du modèle ouvre la liste des prototypes de déclencheur au niveau du modèle lié.
Operational data	Le format des données opérationnelles du déclencheur est affiché, contenant des chaînes arbitraires et des macros qui seront résolues dynamiquement dans *Monitoring → Problems*.
Create enabled	Créer le déclencheur sur la base de ce prototype comme suit : **Yes** - activé **No** - désactivé. Vous pouvez basculer entre « Yes » et « No » en cliquant dessus.
Discover	Découvrir le déclencheur 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 de déclencheur sont affichés.

Pour configurer un nouveau prototype de déclencheur, cliquez sur le bouton *Create trigger prototype* dans le coin supérieur droit.

Options de modification en masse

Les boutons situés sous la liste offrent certaines options de modification en masse :

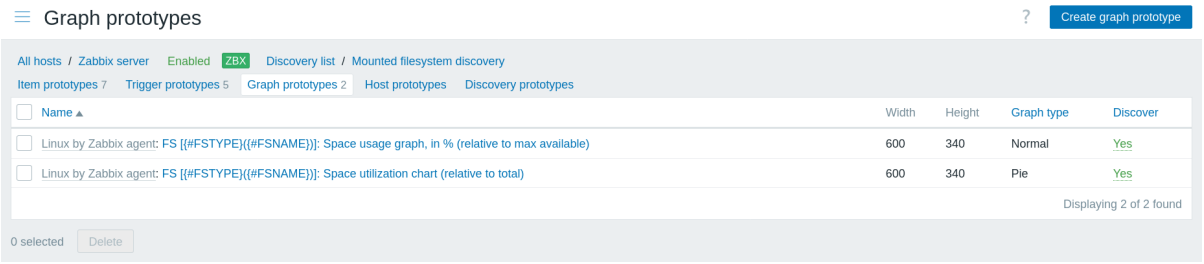
- *Créer activés* - créer ces déclencheurs comme *Activés*
- *Créer désactivés* - créer ces déclencheurs comme *Désactivés*
- *Mise à jour en masse* - mettre à jour en masse ces prototypes de déclencheur
- *Supprimer* - supprimer ces prototypes de déclencheur

Pour utiliser ces options, cochez les cases devant les prototypes de déclencheur correspondants, puis cliquez sur le bouton requis.

3 Prototypes de graphiques

Aperçu

Dans cette section, les prototypes de graphiques d'une règle de découverte de bas niveau sur l'hôte sont affichés. Les prototypes de graphiques constituent la base des graphiques réels de l'hôte, créés lors de la découverte de bas niveau.



Données affichées :

Column	Description
Name	Le nom du prototype de graphique est affiché sous forme de lien bleu. Un clic 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. Un clic 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*, *Empilé*, *Secteur* ou *Éclaté*.
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 le coin supérieur droit.

Options de modification en masse

Les boutons situés sous la liste offrent quelques options de modification en masse :

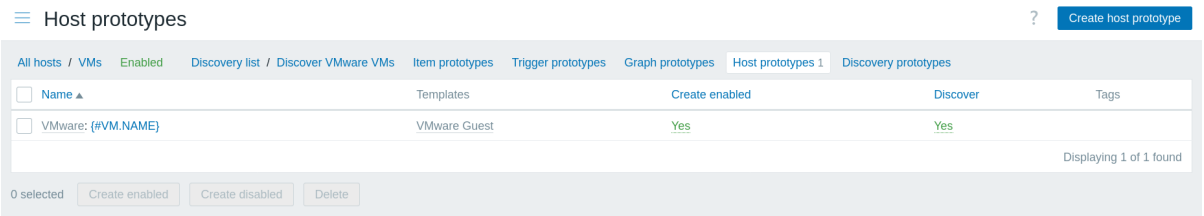
- *Supprimer* - supprimer ces prototypes de graphiques

Pour utiliser ces options, cochez les cases devant les prototypes de graphiques respectifs, puis cliquez sur le bouton requis.

4 Prototypes d'hôtes

Aperçu

Dans cette section, les prototypes d'hôte d'une règle de découverte de bas niveau sur l'hôte sont affichés. Les prototypes d'hôte constituent la base des hôtes réels 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'hôte 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 le coin supérieur droit.

Options de modification en masse

Les boutons situés sous la liste offrent quelques options de modification en masse :

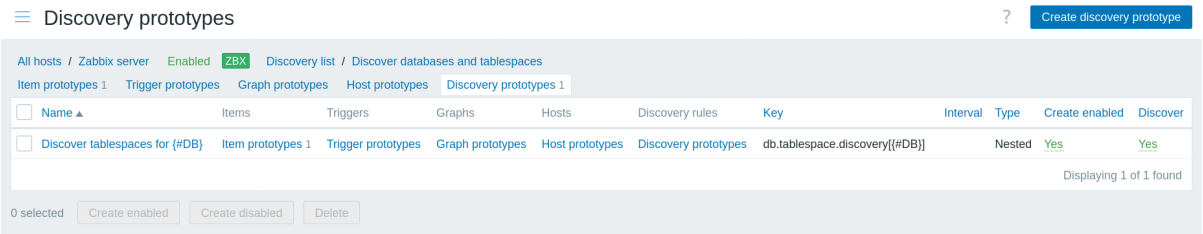
- *Créer activés* - créer ces hôtes comme *Activés*
- *Créer désactivés* - créer ces hôtes comme *Désactivés*
- *Supprimer* - supprimer ces prototypes d'hôte

Pour utiliser ces options, cochez les cases devant les prototypes d'hôte correspondants, puis cliquez sur le bouton requis.

5 Prototypes de découverte

Aperçu

Dans cette section, les prototypes de découverte d'une règle de découverte de bas niveau sur l'hôte sont affichés. Les prototypes de découverte sont des règles de découverte imbriquées de la règle de découverte parente.



Données affichées :

Column	Description
Name	Nom du prototype de découverte, affiché sous forme de lien bleu. Un clic 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 de l'hôte sous forme de lien gris. Un clic 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 d'élément utilisée pour la découverte est affichée.

Column	Description
Interval	La fréquence d'exécution de la découverte est affichée. Veuillez noter que la découverte peut également être exécutée immédiatement en cliquant sur le bouton *Execute now* sous la liste.
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*). En cliquant sur cet état, vous pouvez 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*). En cliquant sur cet état, vous pouvez le modifier, de Yes à No (et inversement).

Pour configurer un nouveau prototype de découverte, cliquez sur le bouton *Create discovery prototype* dans le coin supérieur droit.

Options de modification en masse

Les boutons situés sous la liste offrent quelques 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

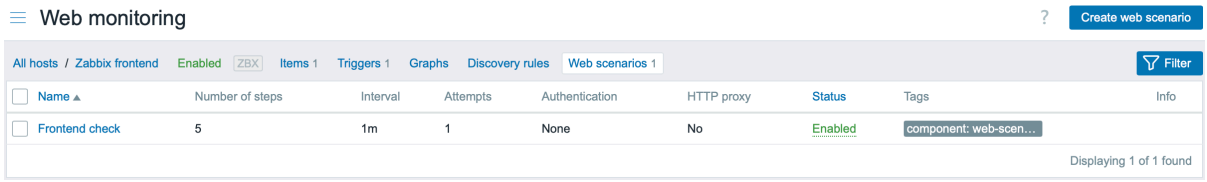
Pour utiliser ces options, cochez les cases devant les prototypes de découverte correspondants, puis cliquez sur le bouton requis.

5 Scénarios Web

Aperçu

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 :

Column	Description
Name	Nom du scénario web. En cliquant sur le nom du scénario web, vous ouvrez le formulaire de configuration du scénario web. Si le scénario web de l'hôte appartient à un modèle, le nom du modèle s'affiche avant le nom du scénario web sous forme de lien gris. En cliquant sur le lien du modèle, vous ouvrirez la liste des scénarios web au niveau du modèle.
Number of steps	Nombre d'étapes contenues dans le scénario.
Update interval	Fréquence d'exécution du scénario.
Attempts	Nombre de tentatives effectuées pour exécuter les étapes du scénario web.
Authentication	La méthode d'authentification affichée est Basic, NTLM, Kerberos, Digest ou None.
HTTP proxy	Affiche le proxy HTTP ou « No » s'il n'est pas utilisé.
Status	L'état du scénario web est affiché : *Enabled* ou *Disabled*. En cliquant sur l'état, vous pouvez 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 « ... » s'affiche et permet de voir tous les tags au survol de la souris.
Info	Si tout fonctionne correctement, aucune icône n'est affichée dans cette colonne. En cas d'erreurs, une icône carrée avec la lettre « i » est affichée. Survolez l'icône pour voir une infobulle contenant la description de l'erreur.

Pour configurer un nouveau scénario web, cliquez sur le bouton *Create web scenario* dans le coin supérieur droit.

Options de modification en masse

Les boutons situés sous la liste offrent plusieurs options de modification en masse :

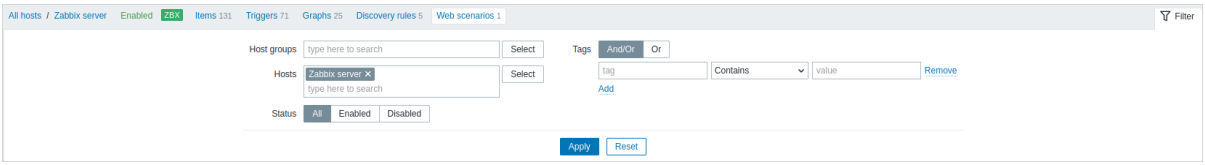
- *Activer* - modifie le statut du scénario en *Activé*
- *Désactiver* - modifie le statut du scénario en *Désactivé*
- *Effacer l'historique et les tendances* - efface les données d'historique et de tendances des scénarios
- *Supprimer* - supprime les scénarios web

Pour utiliser ces options, cochez les cases devant les scénarios web correspondants, 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, les données sont recherchées avec les macros non résolues.

Le lien *Filtre* 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.



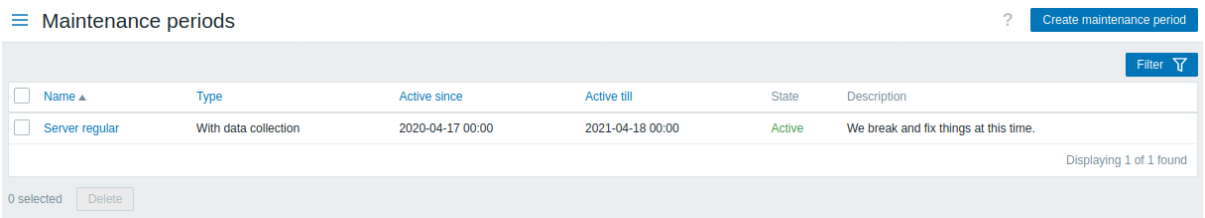
Paramètre	Description
Groupes d'hôtes	Filtrer par un ou plusieurs groupes d'hôtes. La spécification d'un groupe d'hôtes parent sélectionne implicitement tous les groupes d'hôtes imbriqués.
Hôtes	Filtrer par un ou plusieurs hôtes. Si des groupes d'hôtes sont déjà sélectionnés ci-dessus, la sélection des hôtes est limitée à ces groupes.
Statut	Filtrer par le statut du scénario web.
Tags	Filtrer par nom et valeur de tag du scénario web. Il est possible d'inclure comme 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 : **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). Il existe deux types de calcul 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.

5 Maintenance

Aperçu

Dans la section *Collecte de données* → *Maintenance*, les utilisateurs peuvent configurer et gérer des périodes de maintenance pour les hôtes.

Une liste des périodes de maintenance existantes avec leurs détails est affichée.



Données affichées :

Colonne	Description
Nom	Nom de la période de maintenance. Un clic 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*
Actif depuis	La date et l'heure à partir desquelles 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.
Actif jusqu'à	La date et l'heure auxquelles l'exécution des périodes de maintenance cesse d'être active.
État	L'état de la période de maintenance : **Imminente** - deviendra active prochainement **Active** - est active **Expirée** - n'est plus active
Description	La description de la période de maintenance est 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 :

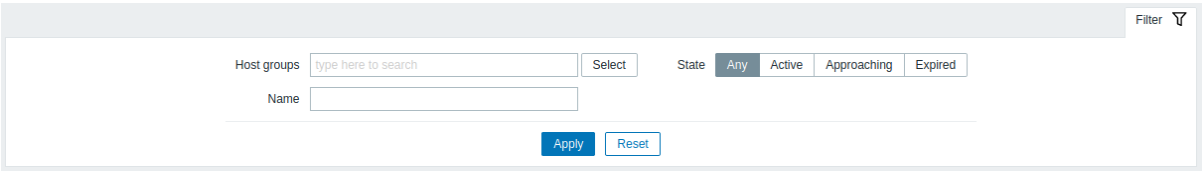
- *Supprimer* - supprimer les périodes de maintenance

Pour utiliser cette option, cochez les cases devant les périodes de maintenance respectives, 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 dans lequel vous pouvez filtrer les périodes de maintenance par groupe d'hôtes, nom et état.



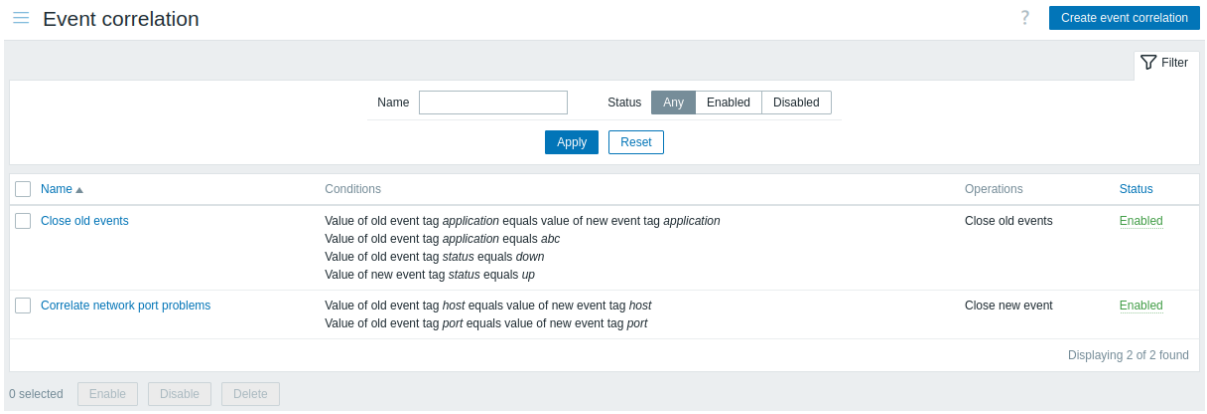
Calcul des files d'attente pendant la maintenance

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.

6 Corrélation d'événements

Aperçu

Dans la section *Collecte de données* → *Corrélation d'événements*, les utilisateurs peuvent configurer et gérer des règles de corrélation globale pour les événements Zabbix.



Données affichées :

Colonne	Description
Nom	Nom de la règle de corrélation. Un clic sur le nom de la règle de corrélation ouvre le formulaire de configuration de la règle.
Conditions	Les conditions de la règle de corrélation sont affichées.
Opérations	Les opérations de la règle de corrélation sont affichées.
Statut	Le statut de la règle de corrélation est affiché - *Activé* ou *Désactivé*. En cliquant sur le statut, vous pouvez le modifier.

Pour configurer une nouvelle règle de corrélation, cliquez sur le bouton *Créer une corrélation d'événements* dans le coin supérieur droit.

Options de modification en masse

Les boutons situés sous la liste offrent quelques options de modification en masse :

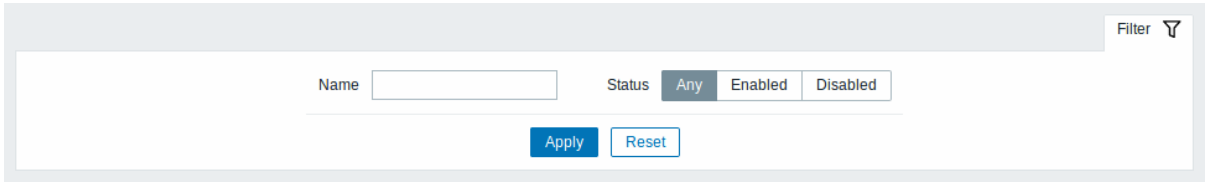
- *Activer* - changer le statut de la règle de corrélation en *Activé*
- *Désactiver* - changer le statut de la règle de corrélation en *Désactivé*
- *Supprimer* - supprimer les règles de corrélation

Pour utiliser ces options, cochez les cases devant les règles de corrélation respectives, puis cliquez sur le bouton requis.

Utilisation du filtre

Vous pouvez utiliser le filtre pour afficher uniquement les règles de corrélation 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 corrélation. Si vous cliquez dessus, un filtre devient disponible, dans lequel vous pouvez filtrer les règles de corrélation par nom et par statut.

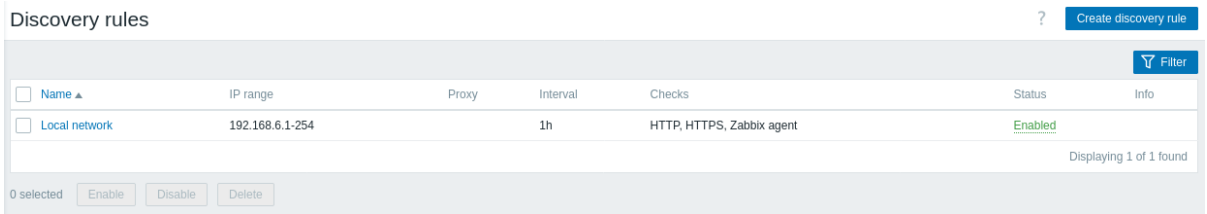


7 Découverte

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. En cliquant sur le nom de la règle de découverte, vous ouvrez 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.
Statut	Le statut de la règle de découverte est affiché - *Activé* ou *Désactivé*. En cliquant sur le statut, vous pouvez le modifier.
Info	Si tout fonctionne correctement, rien n'est affiché dans cette colonne. En cas d'erreurs, une icône d'information rouge avec la lettre « i » est affichée. Survolez l'icône pour voir 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 situés sous la liste offrent quelques options de modification en masse :

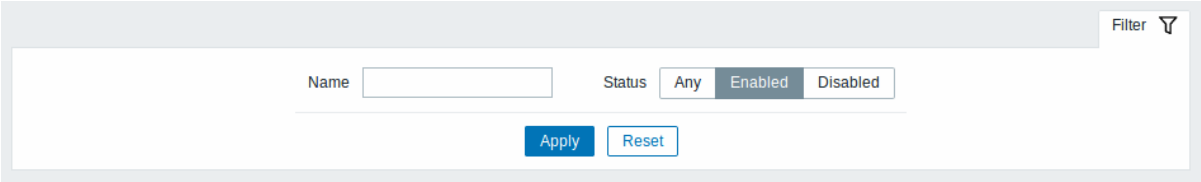
- *Activer* - modifie le statut de la règle de découverte en *Activé*
- *Désactiver* - modifie le statut de la règle de découverte en *Désactivé*
- *Supprimer* - supprime les règles de découverte

Pour utiliser ces options, cochez les cases devant les règles de découverte respectives, 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, dans lequel vous pouvez filtrer les règles de découverte par nom et par statut.



7 Alertes

Aperçu

Ce menu comprend des sections liées à la configuration des alertes dans Zabbix.

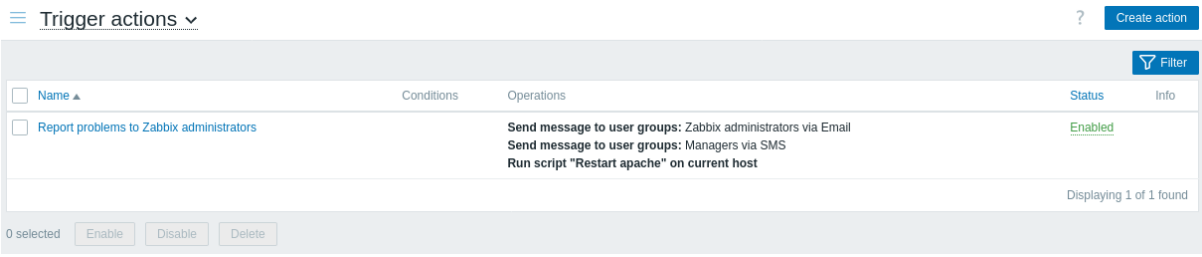
1 Actions

Aperçu

Dans la section *Alertes* → *Actions*, les utilisateurs peuvent configurer et gérer les actions.

Les actions affichées sont celles attribuées à la source d'événement sélectionnée (déclencheur, services, découverte, auto-enregistrement, actions internes).

Pour passer à une autre source d'événement, cliquez sur *Actions* dans la section du menu *Alertes*. Il est également possible de passer d'une source à l'autre en utilisant la liste déroulante du titre dans le coin supérieur gauche.



Données affichées :

Colonne	Description
Nom	Nom de l'action. Cliquer sur le nom de l'action ouvre 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 prénom et le nom (entre parenthèses après le nom d'utilisateur) du destinataire de la notification. Une opération d'action peut être soit une notification, soit une commande distante, selon le type d'opération sélectionné.
Statut	Le statut de l'action est affiché : *Activé* ou *Désactivé*. En cliquant sur le statut, vous pouvez le modifier. Consultez la section Escalades pour plus de détails sur ce qui se passe si une action est désactivée pendant une escalade en cours.
Infos	Si tout fonctionne correctement, aucune icône n'est affichée dans cette colonne. En cas d'erreurs, telles que des opérations d'action ou des conditions manquantes après une mise à niveau/découverte de bas niveau, une icône d'avertissement est affichée. Survolez l'icône pour voir 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 des droits Super admin, les actions sont affichées selon les paramètres d'autorisation. Cela signifie que, dans certains cas, un utilisateur sans droits Super admin ne peut pas voir la liste complète des actions en raison de certaines restrictions d'autorisation. Une action est affichée à l'utilisateur sans droits Super admin si les conditions suivantes sont remplies :

- L'utilisateur dispose d'un accès en lecture-écriture aux groupes d'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 d'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 d'édition en masse

Les boutons situés sous la liste offrent des options d'édition en masse :

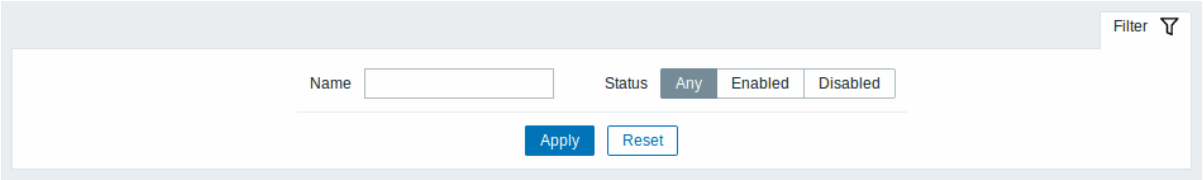
- *Activer* - changer le statut de l'action en *Activé*
- *Désactiver* - changer le statut de l'action en *Désactivé*
- *Supprimer* - supprimer les actions

Pour utiliser ces options, cochez les cases devant les actions respectives, 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.



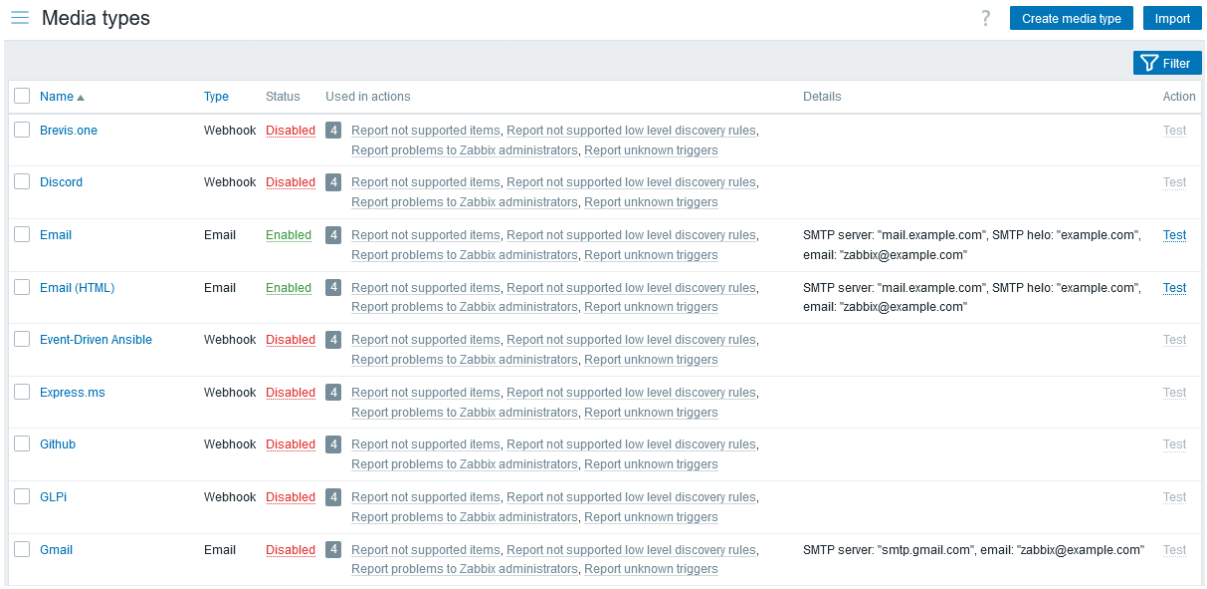
2 Types de média

Aperçu

Dans la section *Alertes* → *Types de média*, les utilisateurs peuvent configurer et gérer les informations des types de média.

Les informations d'un type de média contiennent des instructions générales sur l'utilisation d'un média comme canal de distribution des notifications. Les détails spécifiques, tels que les adresses e-mail individuelles auxquelles envoyer une notification, sont conservés au niveau des utilisateurs individuels.

Une liste des types de média existants avec leurs détails est affichée.



Données affichées :

Column	Description
Name	Nom du type de média. Un clic sur le nom ouvre le formulaire de configuration du type de média.
Type	Le type de média (e-mail, SMS, etc.) est affiché.
Status	L'état du type de média est affiché - *Activé* ou *Désactivé*. En cliquant sur l'état, vous pouvez le modifier.
Used in actions	Les actions dans lesquelles le type de média est utilisé sont affichées, précédées du nombre total de ces actions. Un clic sur le nom de l'action ouvre le formulaire de configuration de l'action. Si l'utilisateur n'a pas les permissions sur l'action, le nom n'est pas cliquable.
Details	Les informations détaillées du type de média sont affichées.
Actions	L'action suivante est disponible : **Test** - cliquez 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 sujet et un corps de message de test) et envoyer un message de test pour vérifier que le type de média configuré fonctionne. Voir aussi : Test de 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 d'édition en masse

Les boutons situés sous la liste offrent des options d'édition en masse :

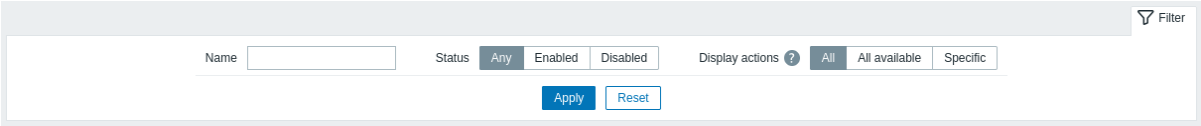
- *Activer* - changer l'état du type de média en *Activé*
- *Désactiver* - changer l'état du type de média en *Désactivé*
- *Exporter* - exporter les types de médias vers un fichier YAML, XML ou JSON
- *Supprimer* - supprimer les types de médias

Pour utiliser ces options, cochez les cases devant les types de média respectifs, 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 *Filtre* est disponible au-dessus de la liste des types de média. Si vous cliquez dessus, un filtre devient disponible, dans lequel 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 *Utilisé dans les actions* en fonction de la portée de l'utilisation de leur type de média (définie par le paramètre *Send to media type* dans les détails de l'opération d'une action).



3 Scripts

Aperçu

Dans la section *Alertes* → *Scripts*, les scripts globaux définis par l'utilisateur peuvent être configurés et gérés.

Les scripts globaux, selon la portée configurée ainsi que les autorisations de l'utilisateur, peuvent être exécutés :

- depuis le menu de l'hôte à divers emplacements de l'interface (*Tableau de bord*, *Problèmes*, *Dernières données*, *Cartes*, etc.)
- depuis le menu de l'événement
- comme opération d'action

Les scripts sont exécutés sur l'agent Zabbix, le serveur Zabbix (proxy) ou uniquement sur le serveur Zabbix. Voir aussi Exécution de commandes.

Sur l'agent Zabbix comme sur le proxy Zabbix, les scripts distants sont désactivés par défaut. Ils peuvent être activés comme suit :

- Pour les commandes distantes exécutées sur l'agent Zabbix :
  - ajout d'un paramètre AllowKey=system.run[<command>,*] pour chaque commande autorisée dans la configuration de l'agent, * correspond aux modes wait et nowait ;
- Pour les commandes distantes exécutées sur le proxy Zabbix :
  - **Avertissement : il n'est pas nécessaire d'activer les commandes distantes sur le proxy Zabbix si les commandes distantes sont exécutées sur l'agent Zabbix surveillé par le proxy Zabbix.** Toutefois, s'il est nécessaire d'exécuter des commandes distantes sur le proxy Zabbix, définissez le paramètre *EnableRemoteCommands* sur « 1 » dans la configuration du proxy.

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, depuis 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 :

Column	Description
Name	Nom du script. Un clic sur le nom du script ouvre le formulaire de configuration du script.
Scope	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.
Used in actions	Toutes les actions dans lesquelles le script est utilisé sont affichées, précédées du nombre total de ces actions. Un clic sur le nom de l'action ouvre le formulaire de configuration de l'action. Si l'utilisateur n'a pas les autorisations sur l'action, le nom n'est pas cliquable.
Type	Le type de script affiché est : *URL*, *Webhook*, *Script*, *SSH*, *Telnet* ou commande *IPMI*.
Execute on	Indique si le script sera exécuté sur l'agent Zabbix, le proxy Zabbix ou le serveur, ou uniquement sur le serveur Zabbix.

Column	Description
Commands	Toutes les commandes à exécuter dans le script sont affichées. Rien n'est affiché ici pour les webhooks.
User group	Le groupe d'utilisateurs auquel le script est disponible est affiché (ou *All* pour tous les groupes d'utilisateurs).
Host group	Le groupe d'hôtes pour lequel le script est disponible est affiché (ou *All* pour tous les groupes d'hôtes).
Host access	Le niveau d'autorisation pour le groupe d'hôtes est affiché : *Read* ou *Write*. Seuls les utilisateurs disposant du niveau d'autorisation requis auront accès à l'exécution du script.

Pour configurer un nouveau script, cliquez sur le bouton *Create script* dans le coin supérieur droit.

Options de modification en masse

Un bouton situé sous la liste propose une option de modification en masse :

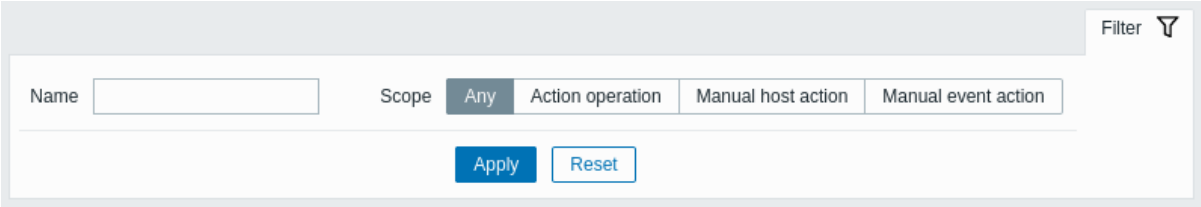
- *Supprimer* - supprimer les scripts

Pour utiliser cette option, cochez les cases devant les scripts respectifs et cliquez sur *Supprimer*.

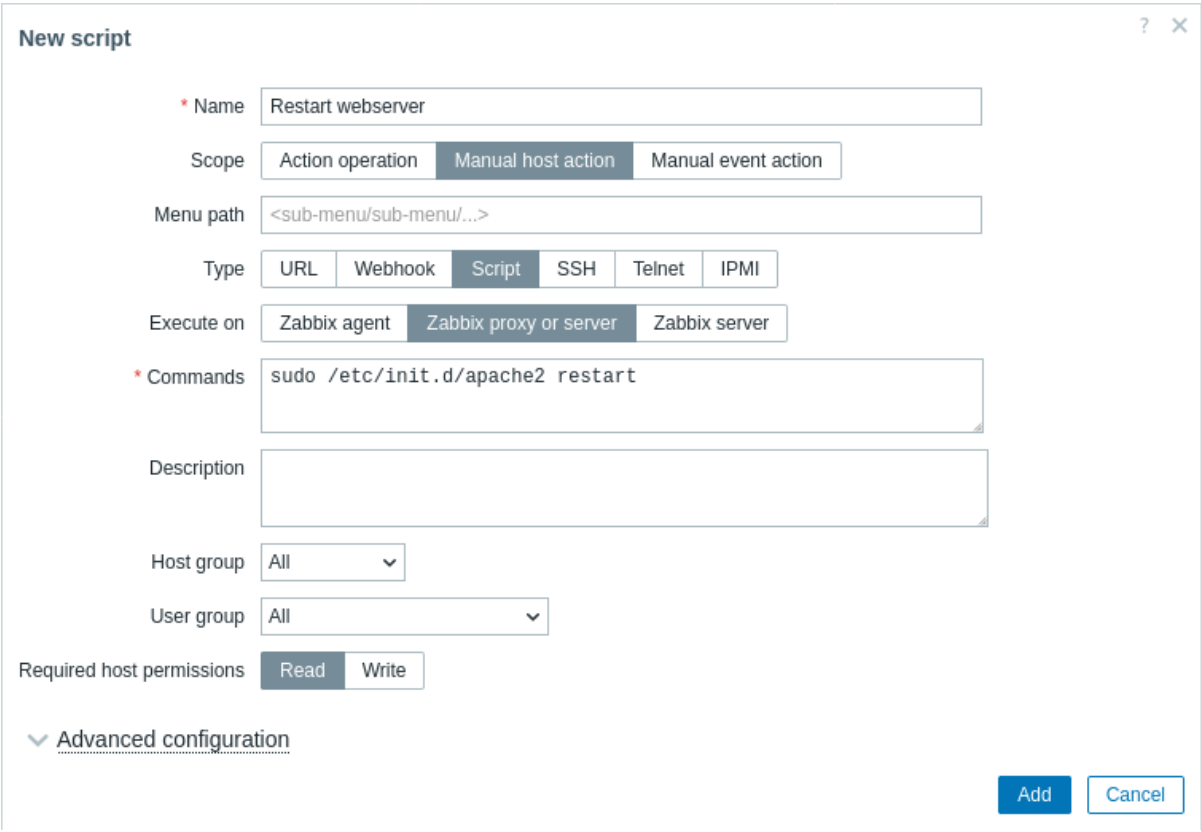
Utilisation du filtre

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 *Filtre* est disponible au-dessus de la liste des scripts. Si vous cliquez dessus, un filtre devient disponible, dans lequel vous pouvez filtrer les scripts par nom et portée.



Configuration d'un script global



Note:
Il est recommandé d'utiliser des fonctions de macro au lieu de macros simples dans les scripts globaux afin d'améliorer la sécurité, car les macros ne sont pas automatiquement échappées.

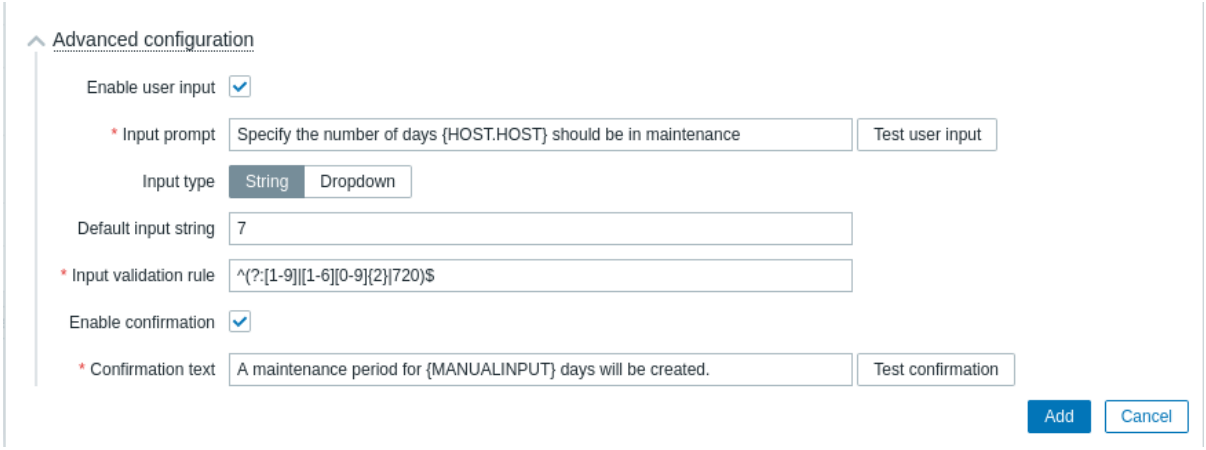
Attributs du script :

Paramètre	Description
Nom	Nom unique du script. Par ex. `Clear /tmp filesystem`
Portée	Portée du script : opération d'action, action manuelle sur l'hôte ou action manuelle sur l'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 « Action operation » rend le script disponible pour tous les utilisateurs ayant accès à *Alerts* → *Actions*. Si un script est effectivement utilisé dans une action, sa portée ne peut plus être modifiée pour autre chose que « action operation ». **Prise en charge des macros** La portée affecte l'étendue des macros disponibles. Par exemple, les macros liées à l'utilisateur ({USER.*}) sont prises en charge dans les scripts afin de permettre le passage d'informations sur l'utilisateur qui a lancé le script. Cependant, elles ne sont pas prises en charge si la portée du script est une 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 l'hôte et d'action manuelle sur l'événement. Pour savoir quelles autres macros sont prises en charge, recherchez « Trigger-based notifications and commands/Trigger-based commands », « Manual host action scripts » et « Manual event action scripts » dans le tableau des macros prises en charge. Notez que si une macro peut être résolue en une valeur contenant des espaces (par exemple, le nom de l'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 ex. `Main menu/Sub menu1/Sub menu2`. Lors de l'accès aux scripts via le menu hôte/événement dans les sections de supervision, ils seront organisés selon les répertoires indiqués. Ce champ s'affiche uniquement si « Manual host action » ou « Manual event action » est sélectionné comme *Portée*.
Type	Cliquez sur le bouton correspondant pour sélectionner le type de script : **URL**, **Webhook**, **Script**, **SSH**, **Telnet** ou commande **IPMI**. Le type **URL** est disponible uniquement lorsque « Manual host action » ou « Manual event action » est sélectionné comme *Portée*.
Type de script : **URL**	
URL	Spécifiez l'URL pour un accès rapide 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 en URL.
Ouvrir dans une nouvelle fenêtre	Détermine si l'URL doit être ouverte dans un nouvel onglet du navigateur ou dans le même.
Type de script : **Webhook**	
Paramètres	Spécifiez les variables du webhook sous forme de paires attribut-valeur. Voir aussi : 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 à côté. La prise en charge des macros dépend de la portée du script (voir *Portée* ci-dessus). Voir aussi : configuration du média Webhook, Objets JavaScript supplémentaires.

Paramètre	Description
Délai d'expiration	Délai d'expiration de l'exécution JavaScript (1-60s, 30s par défaut). Les suffixes de temps sont pris en charge, par ex. 30s, 1m.
Type de script : **Script**	
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 or server** - le script sera exécuté par Zabbix proxy ou le serveur, selon que l'hôte est supervisé par le proxy ou le serveur. Il sera exécuté sur le proxy si cela est activé par EnableRemoteCommands. Il sera exécuté sur le serveur si les scripts globaux sont activés par le paramètre serveur EnableGlobalScripts. **Zabbix server** - le script sera exécuté uniquement par Zabbix server. Cette option ne sera pas disponible si les scripts globaux sont désactivés par le paramètre 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 « Password » est sélectionné comme méthode d'authentification.
Fichier de clé publique	Saisissez le chemin vers le fichier de clé publique. Ce champ est disponible si « Public key » est sélectionné comme méthode d'authentification.
Fichier de clé privée	Saisissez le chemin vers le fichier de clé privée. Ce champ est disponible si « Public key » est sélectionné comme méthode d'authentification.
Phrase secrète	Saisissez la phrase secrète. Ce champ est disponible si « Public key » est sélectionné comme méthode d'authentification.
Port	Saisissez le port.
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.
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 *All* pour tous les groupes d'hôtes).
Groupe d'utilisateurs	Sélectionnez le groupe d'utilisateurs auquel le script sera disponible (ou *All* pour tous les groupes d'utilisateurs). Ce champ s'affiche uniquement si « Manual host action » ou « Manual event action » est sélectionné comme *Portée*.
Autorisations d'hôte requises	Sélectionnez le niveau d'autorisation pour le groupe d'hôtes : *Read* ou *Write*. Seuls les utilisateurs disposant du niveau d'autorisation requis auront accès à l'exécution du script. Ce champ s'affiche uniquement si « Manual host action » ou « Manual event action » est sélectionné comme *Portée*.
Configuration avancée	Cliquez sur le libellé *Configuration avancée* pour afficher les options de configuration avancée. Ce champ s'affiche uniquement si « Manual host action » ou « Manual event action » est sélectionné comme *Portée*.

Configuration avancée

Des options de configuration avancée sont disponibles dans une section repliable *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 aussi : Saisie manuelle de l'utilisateur.
Invite de saisie	Saisissez un texte personnalisé invitant à une saisie utilisateur personnalisée. Ce texte sera affiché au-dessus du champ de saisie dans la fenêtre contextuelle *Saisie manuelle*. Pour voir un aperçu de la fenêtre contextuelle *Saisie manuelle*, cliquez sur *Tester la saisie utilisateur*. L'aperçu permet également de vérifier si la chaîne saisie est conforme à la règle de validation de saisie (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 des valeurs uniques pour la liste déroulante de saisie utilisateur dans une liste délimitée 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 n'est affiché que 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 saisie*. La valeur saisie ici sera affichée par défaut dans la fenêtre contextuelle *Saisie manuelle*. Ce champ n'est affiché que si « Chaîne » est sélectionné comme *Type de saisie*.
Règle de validation de saisie	Saisissez une expression régulière pour valider la chaîne de saisie utilisateur. Les expressions régulières globales sont prises en charge. Ce champ n'est affiché que 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 des opérations potentiellement dangereuses (comme un script de redémarrage) ou susceptibles de prendre beaucoup de temps.
Texte de confirmation	Saisissez un texte de confirmation personnalisé pour la fenêtre contextuelle de confirmation activée avec la case à cocher 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 utilisateur

La saisie manuelle 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.

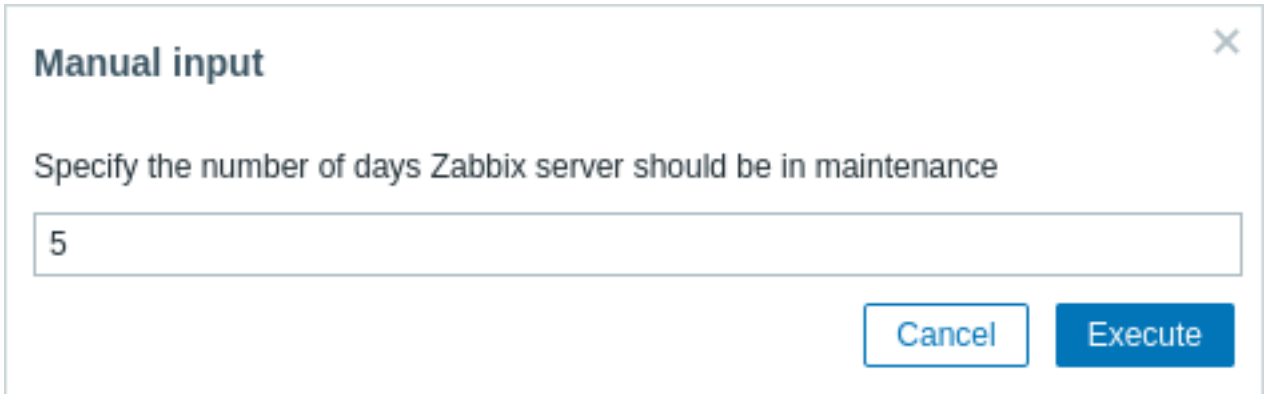
Par exemple, vous pouvez vouloir fournir un entier différent ou une adresse URL différente au script lors de son exécution.

Pour activer la saisie manuelle utilisateur :

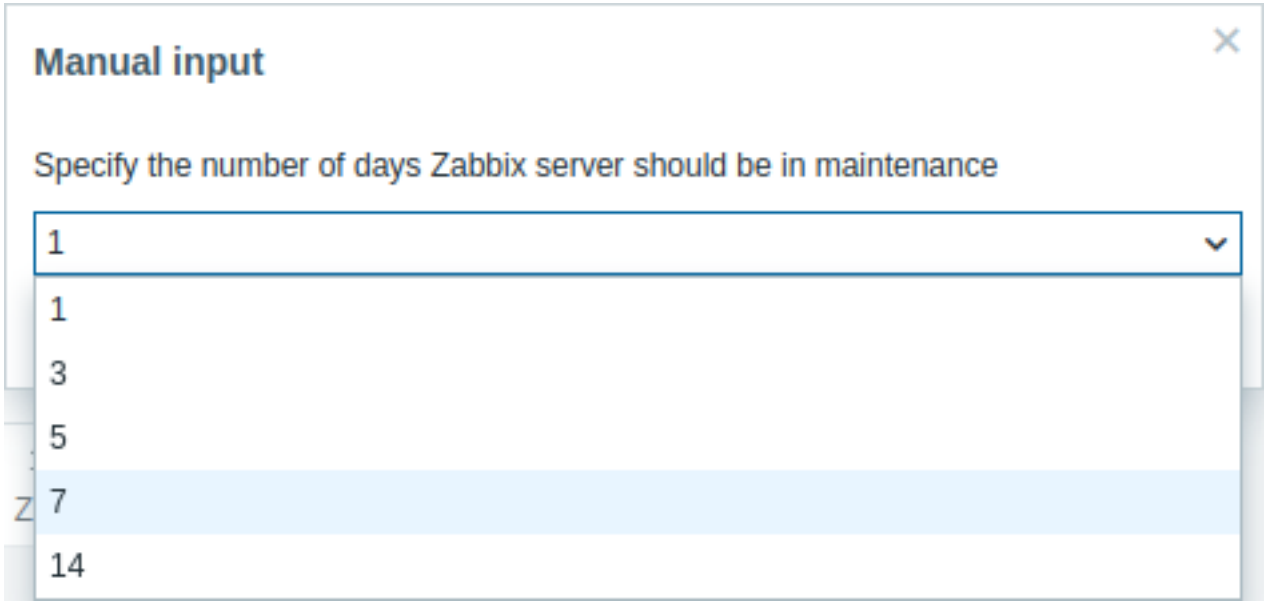
- utilisez la macro {MANUALINPUT} dans le script (commandes, script, paramètre de script) là où nécessaire ; ou dans le champ URL des scripts URL ;
- dans la configuration avancée du script, activez la saisie manuelle utilisateur et configurez les options de saisie.

Lorsque la saisie utilisateur est activée, avant l'exécution du script, une fenêtre contextuelle *Saisie manuelle* s'affichera pour demander à l'utilisateur de fournir une valeur personnalisée. La valeur fournie remplacera {MANUALINPUT} dans le script.

Selon la configuration, il sera demandé à l'utilisateur de saisir une valeur de type chaîne :



Ou de sélectionner une valeur dans une liste déroulante d'options prédéfinies :



La saisie manuelle utilisateur est disponible uniquement pour les scripts dont la portée est « Manual host action » ou « Manual event action ».

Exécution du script et résultat

Les scripts exécutés par le serveur Zabbix sont lancés 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 :

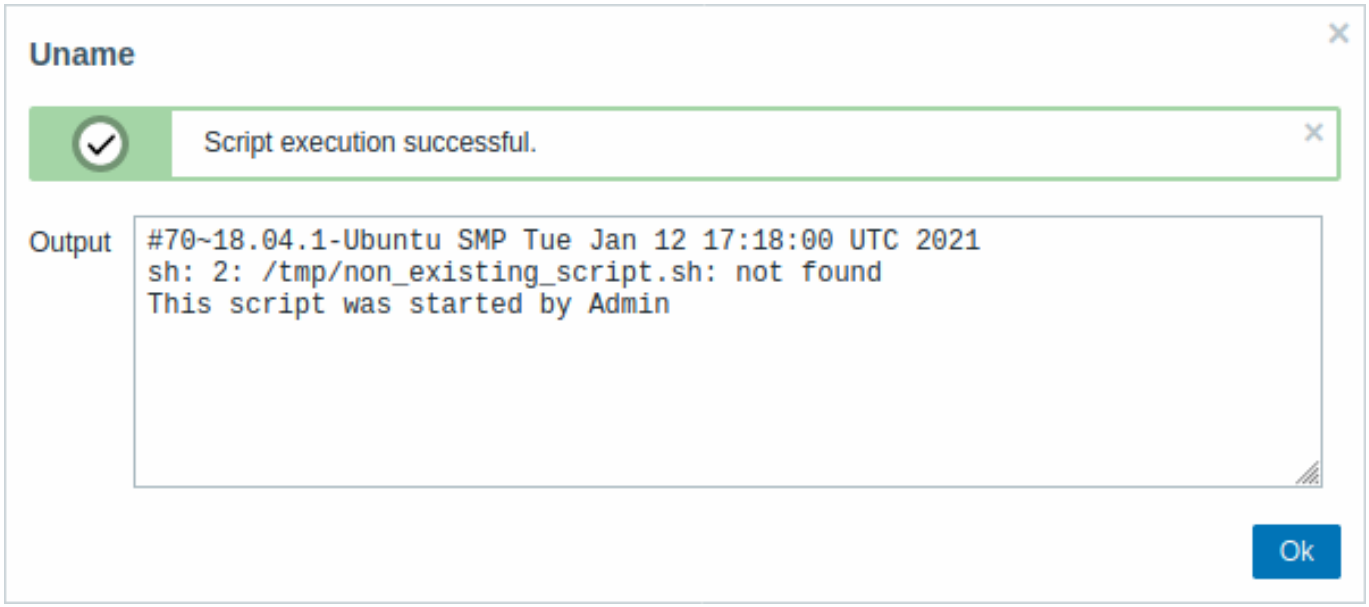
- Si le script se termine avec succès (code de sortie 0), la valeur de retour est limitée à 16 Mo (y compris les espaces de fin qui sont tronqués).
- Si le script se termine avec une erreur (code de sortie non nul), la valeur de retour correspond à la sortie d'erreur standard, limitée à 2 Ko.

Par défaut, Zabbix ne stocke pas la sortie étendue des scripts. Pour conserver tous les détails de la sortie, vous pouvez mettre en place 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.

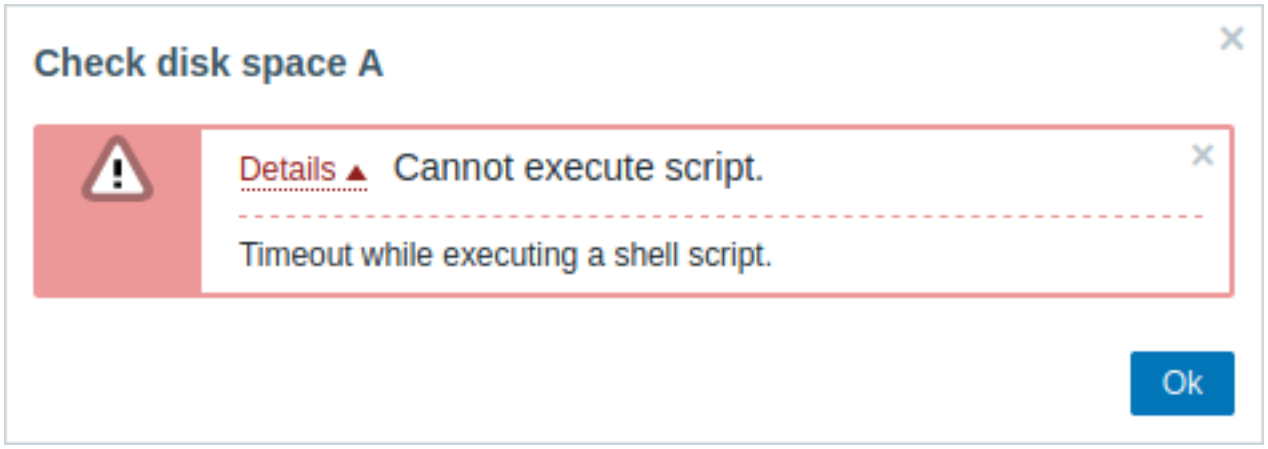
Délai d'expiration du script

Agent Zabbix

Il peut arriver qu'un délai d'expiration se produise lors de l'exécution d'un script.

Voir ci-dessous un exemple de script exécuté sur l'agent Zabbix ainsi que la fenêtre de résultat :

```
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 conseillé d'optimiser le script lui-même (dans l'exemple ci-dessus, « 5 ») au lieu d'ajuster le paramètre `Timeout` dans la configuration de l'agent Zabbix et la configuration du serveur Zabbix. Cependant, pour l'agent Zabbix en mode actif, le paramètre `Timeout` dans la configuration du serveur Zabbix doit être d'au moins plusieurs secondes supérieur au paramètre `RefreshActiveChecks` dans la configuration de l'agent Zabbix. Cela garantit que le serveur dispose de suffisamment de temps pour recevoir les résultats des contrôles actifs depuis l'agent. Notez que l'exécution de scripts sur un agent actif est prise en charge depuis l'agent Zabbix 7.0.

Si le paramètre `Timeout` a été modifié dans la configuration de l'agent Zabbix, le message d'erreur suivant s'affichera :

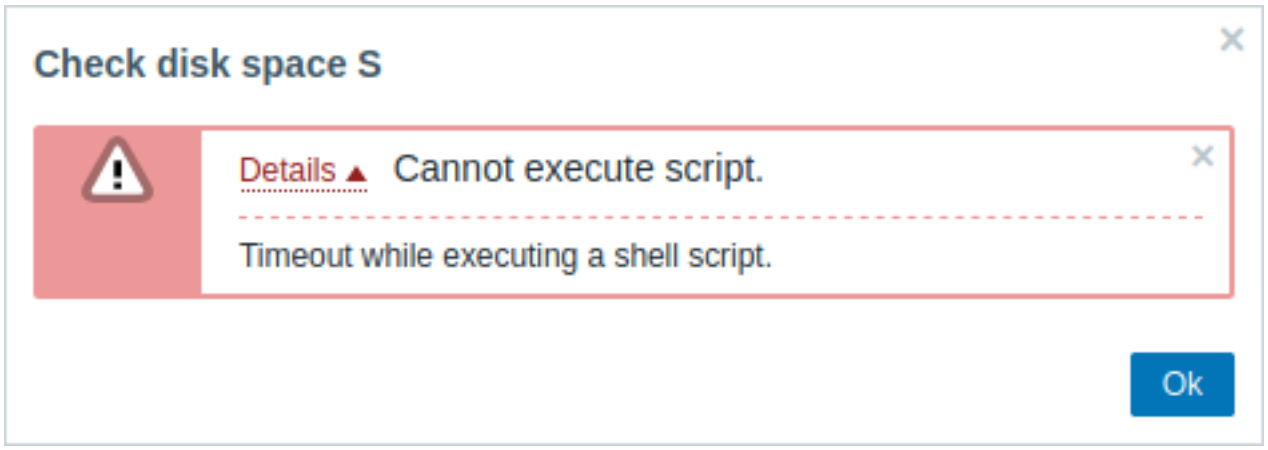
```
Get value from agent failed: ZBX_TCP_READ() timed out.
```

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.

Serveur/proxy Zabbix

Voir ci-dessous un exemple de script exécuté sur le serveur Zabbix et la fenêtre de résultat :

```
sleep 11
df -h
```



Il est également conseillé d'optimiser le script lui-même (au lieu d'ajuster le paramètre `TrapperTimeout` à une valeur correspondante (dans notre cas, > 11) en modifiant la configuration du serveur Zabbix).

8 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.

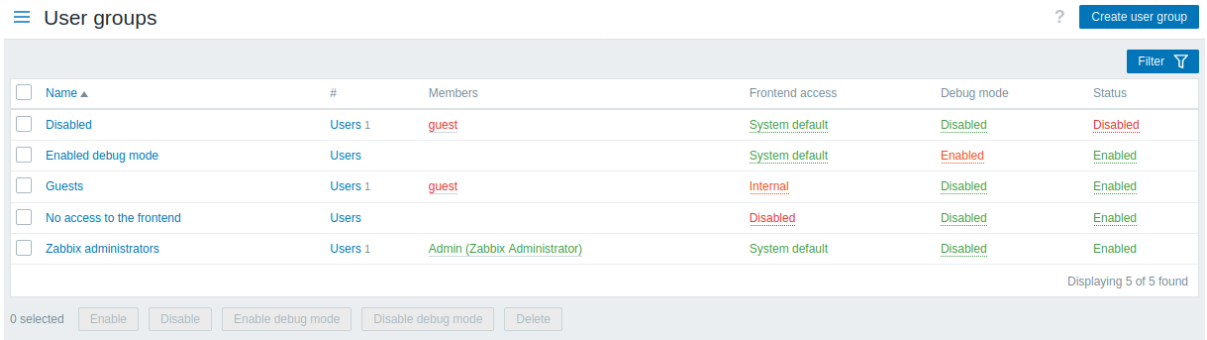
1 Groupes d'utilisateurs

Aperçu

Dans la section *Utilisateurs* → *Groupes d'utilisateurs*, les groupes d'utilisateurs du système sont gérés.

Groupes d'utilisateurs

Une liste des groupes d'utilisateurs existants avec leurs détails est affichée.



Données affichées :

Colonne	Description
Nom	Nom du groupe d'utilisateurs. En cliquant sur le nom du groupe d'utilisateurs, vous ouvrez le formulaire de configuration du groupe d'utilisateurs.
#	Le nombre d'utilisateurs dans le groupe. En cliquant sur *Utilisateurs*, les utilisateurs correspondants seront affichés, filtrés dans la liste des utilisateurs.
Membres	Noms d'utilisateur des utilisateurs individuels du groupe d'utilisateurs (avec le prénom et le nom entre parenthèses). En cliquant sur le nom d'utilisateur, vous ouvrez le formulaire de configuration de l'utilisateur. Les utilisateurs des groupes désactivés sont affichés en rouge.

Colonne	Description
Accès au frontend	Le niveau d'accès au frontend est affiché : **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 au frontend Zabbix est interdit pour ce groupe. En cliquant sur le niveau actuel, vous pouvez le modifier.
Mode de débogage	L'état du mode de débogage est affiché - *Activé* ou *Désactivé*. En cliquant sur l'état, vous pouvez le modifier.
Statut	Le statut du groupe d'utilisateurs est affiché - *Activé* ou *Désactivé*. En cliquant sur le statut, vous pouvez le modifier.

Pour configurer un nouveau groupe d'utilisateurs, cliquez sur le bouton *Créer un groupe d'utilisateurs* dans le coin supérieur droit.

Options d'édition en masse

Les boutons situés sous la liste offrent des options d'édition en masse :

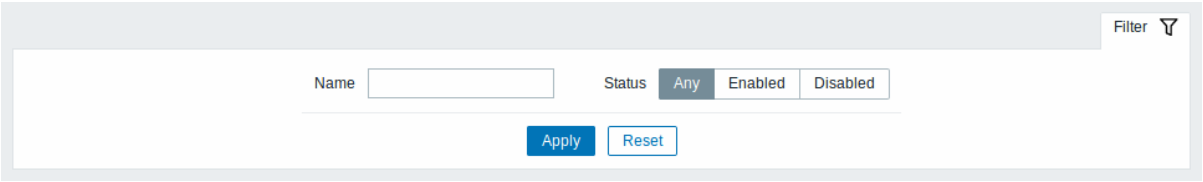
- *Activer* - changer le statut du groupe d'utilisateurs en *Activé*
- *Désactiver* - changer le statut du groupe d'utilisateurs en *Désactivé*
- *Activer le mode débogage* - activer le mode débogage pour les groupes d'utilisateurs
- *Désactiver le mode débogage* - désactiver le mode débogage pour les groupes d'utilisateurs
- *Supprimer* - supprimer les groupes d'utilisateurs

Pour utiliser ces options, cochez les cases devant les groupes utilisateur respectifs, puis cliquez sur le bouton requis.

Utilisation d'un 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 des macros non résolues.

Le lien *Filtre* 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 statut.



2 Rôles utilisateur

Aperçu

Dans la section *Utilisateurs* > *Rôles utilisateur*, vous pouvez créer des rôles utilisateur.

Les rôles utilisateur permettent de créer des autorisations granulaires en fonction du type d'utilisateur initialement sélectionné (*User*, *Admin*, *Super admin*).

Lors de la sélection d'un type d'utilisateur, toutes les autorisations disponibles pour ce type d'utilisateur sont accordées (cochées par défaut).

Les autorisations ne peuvent être révoquées qu'au sein du sous-ensemble disponible 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 non disponibles sont grisées ; les utilisateurs ne pourront pas accéder à l'élément, même en saisissant une URL directe vers cet élément dans le navigateur.

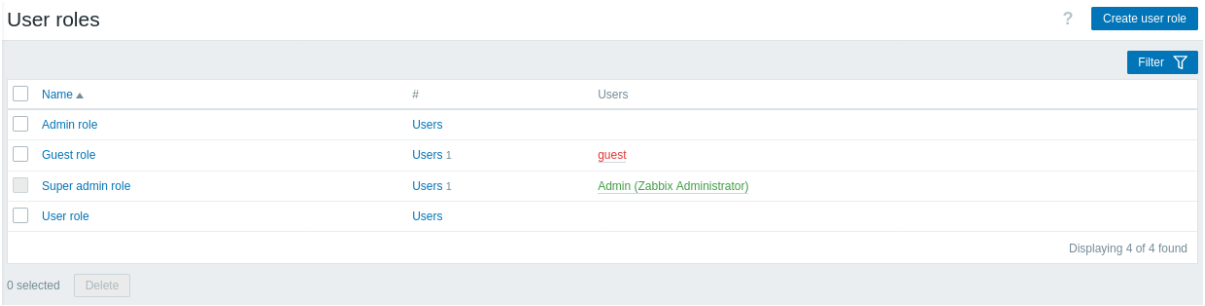
> **Note:**
> La restriction de l'accès à certains éléments de l'interface utilisateur empêche uniquement l'ouverture de cette page ; elle ne supprime pas la possibilité d'accéder aux données sous-jacentes dans d'autres parties de l'interface.

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

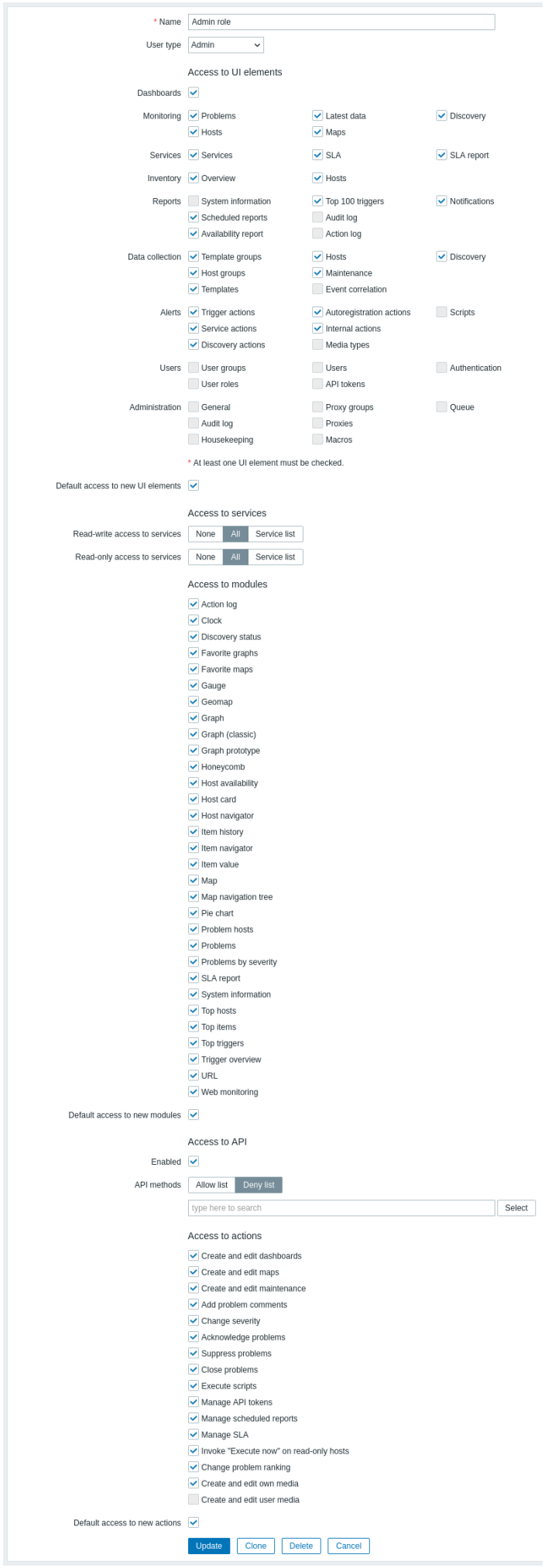


Ils sont basés sur les principaux types d'utilisateurs dans Zabbix. La liste de tous les utilisateurs auxquels le rôle correspondant est attribué est affichée. Les utilisateurs inclus dans des groupes désactivés sont indiqués en rouge. Le *rôle Invité* est un rôle de type utilisateur dont les seules autorisations permettent d'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 avec des 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 le nom du rôle pour ouvrir le formulaire de configuration.


* Name
Admin role
User type
Admin
Access to UI elements
Dashboards
Monitoring
Problems
Latest data
Discovery
Hosts
Maps
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 correlation
Alerts
Trigger actions
Autoregistration actions
Scripts
Service actions
Internal actions
Discovery actions
Media types
Users
User groups
Users
Authentication
User roles
API tokens
Administration
General
Proxy groups
Queue
Audit log
Proxies
Housekeeping
Macros
* At least one UI element must be checked.
Default access to new UI elements
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 history
Item navigator
Item value
Map
Map navigation tree
Pie chart
Problem hosts
Problems
Problems by severity
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
type here to search
Select
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
Default access to new actions
Update
Clone
Delete
Cancel

Les autorisations disponibles sont affichées. Pour révoquer une autorisation donnée, décochez sa case.

Les autorisations disponibles ainsi que les valeurs par défaut pour chaque rôle utilisateur préconfiguré dans Zabbix sont décrites ci-dessous.

Autorisations par défaut

Accès aux éléments de l'interface utilisateur

L'accès par défaut aux sections du menu dépend du type d'utilisateur. Voir la page Permissions pour plus de détails.

Accès aux autres options

Paramètre	**Description**	**Rôles utilisateur par défaut**			
		Rôle Super admin	*Rôle Admin*	*Rôle User*	*Rôle Guest*
Accès par défaut aux nouveaux éléments de l'interface utilisateur	Cette option spécifie comment les nouvelles sections du menu seront accessibles après une mise à niveau de Zabbix. Les sections de menu existantes des modules restent inchangées.	Oui	Oui	Oui	Oui
Accès aux services					
Accès en lecture-écriture aux services	Sélectionnez l'accès en lecture-écriture aux services : **Aucun** - aucun accès **Tous** - l'accès à tous les services est en lecture-écriture **Liste de services** - sélectionnez les services pour l'accès en lecture-écriture L'accès en lecture-écriture, s'il est accordé, prévaut 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 tag	Spécifiez le nom du tag et, éventuellement, sa valeur afin d'accorder en plus un accès en lecture-écriture aux services correspondant au tag. 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, prévaut 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** - l'accès à tous les services est en lecture seule **Liste de services** - sélectionnez les services pour l'accès en lecture seule L'accès en lecture seule ne prévaut pas sur l'accès en lecture-écriture et est hérité dynamiquement par les services enfants.			Tous	Tous
Accès en lecture seule aux services avec tag	Spécifiez le nom du tag et, éventuellement, sa valeur afin d'accorder en plus un accès en lecture seule aux services correspondant au tag. Cette option est disponible si « Liste de services » est sélectionné dans le paramètre *Accès en lecture seule aux services*. L'accès en lecture seule ne prévaut pas sur l'accès en lecture-écriture et est hérité dynamiquement par les services enfants.				
Accès aux modules					
<Nom du module>	Autoriser/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 spécifie 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 > General > Modules*.				
Accès à l'API					
Activé	Activer/désactiver l'accès à l'API.	Oui	Oui	Oui	Non
Méthodes API	Sélectionnez soit *Liste d'autorisation* pour autoriser, soit *Liste de refus* pour refuser les méthodes API spécifiées dans le champ de recherche. Notez qu'il n'est pas possible d'autoriser certaines méthodes API et d'en refuser d'autres. Dans le champ de recherche, commencez à saisir le nom de la méthode, puis sélectionnez la méthode dans la liste d'auto-complétion. Vous pouvez également appuyer sur le bouton Select et sélectionner des méthodes dans la liste complète disponible pour ce type d'utilisateur. Notez que si une certaine action du bloc Access to actions est décochée, les utilisateurs ne pourront pas utiliser les méthodes API liées à cette action. Les caractères génériques sont pris en charge. Exemples : `dashboard.*` (toutes les méthodes du service API « dashboard. ») `*` (n'importe quelle méthode), `*.export` (méthodes portant le nom « .export » de tous les services API). Si aucune méthode n'a été spécifiée, la règle *Allow/Deny list* sera ignorée.				

Accès aux actions					
Créer et modifier des tableaux de bord	Décocher cette case révoquera également les droits d'utiliser les méthodes API `.create`, `.update` et `.delete` pour les éléments correspondants.	Oui	Oui	Oui	Non
Créer et modifier des cartes					
Créer et modifier des maintenances				Non	
Ajouter des commentaires sur les problèmes	Décocher cette case révoquera également les droits d'effectuer l'action correspondante via la méthode API `event.acknowledge`.			Oui	
Modifier la sévérité					
Acquitter les problèmes					
Supprimer les problèmes					
Fermer les problèmes					
Exécuter des scripts	Décocher cette case révoquera également les droits d'utiliser la méthode API `script.execute`.				
Gérer les jetons API	Décocher cette case révoquera également les droits d'utiliser toutes les méthodes API `token.*`.				
Gérer les rapports planifiés	Décocher cette case révoquera également les droits d'utiliser toutes les méthodes API `report.*`.			Non	
Gérer les SLA	Activer/désactiver les droits de gestion des SLA.				
Invoquer « Execute now » sur des hôtes en lecture seule	Autoriser l'utilisation de l'option « Execute now » dans les dernières données pour les éléments des hôtes en lecture seule.			Oui	
Modifier le classement des problèmes	Autoriser la modification du classement des problèmes de cause à symptôme, et inversement.				
Créer et modifier ses propres médias	Autoriser la création/modification de ses propres médias.				
Créer et modifier les médias utilisateur	Autoriser la création/modification de médias pour les utilisateurs. Cette option n'est disponible que pour les utilisateurs Super admin.		Non	Non	
Accès par défaut aux nouvelles actions	Cette option spécifie comment les nouvelles actions seront accessibles après une mise à niveau de Zabbix.		Oui	Oui	

Voir aussi :

- Configuration d'un utilisateur

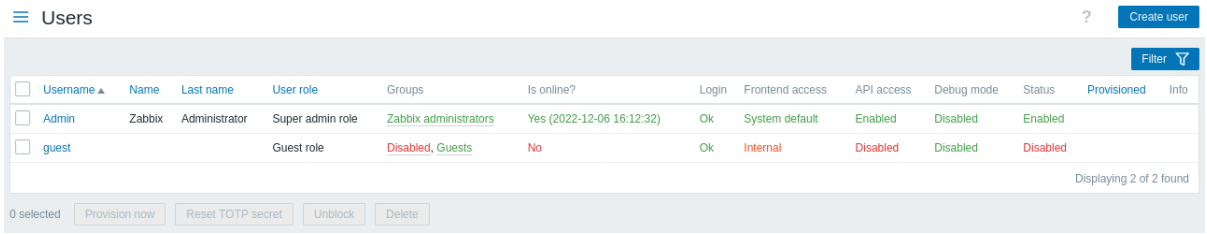
3 Utilisateurs

Aperçu

Dans la section *Utilisateurs* → *Utilisateurs*, les utilisateurs du système sont gérés.

Utilisateurs

Une liste des utilisateurs existants avec leurs détails est affichée.



Données affichées :

Column	Description
Username	Nom d'utilisateur pour se connecter à Zabbix. En cliquant sur le nom d'utilisateur, vous ouvrez le formulaire de configuration de l'utilisateur.
Name	Prénom de l'utilisateur.
Last name	Nom de famille de l'utilisateur.
User role	Le rôle utilisateur est affiché.
Groups	Les groupes dont l'utilisateur est membre sont listés. En cliquant sur le nom du groupe d'utilisateurs, vous ouvrez le formulaire de configuration du groupe d'utilisateurs. Les groupes désactivés sont affichés en rouge.
Is online?	Le statut en ligne de l'utilisateur est affiché - *Yes* ou *No*. L'heure de la dernière activité de l'utilisateur est affichée entre parenthèses.
Login	Le statut de connexion de l'utilisateur est affiché - *Ok* ou *Blocked*. Un utilisateur peut être temporairement bloqué après avoir dépassé le nombre de tentatives de connexion infructueuses défini dans la section *Administration → General → Other* (cinq par défaut). En cliquant sur *Blocked*, vous pouvez débloquer l'utilisateur.
Frontend access	Le niveau d'accès au frontend est affiché - *System default*, *Internal*, *LDAP* ou *Disabled*, selon celui défini pour l'ensemble du groupe d'utilisateurs.
API access	Le statut d'accès à l'API est affiché - *Enabled* ou *Disabled*, selon celui défini pour le rôle utilisateur.
Debug mode	Le statut du mode débogage est affiché - *Enabled* ou *Disabled*, selon celui défini pour l'ensemble du groupe d'utilisateurs.
Status	Le statut de l'utilisateur est affiché - *Enabled* ou *Disabled*, selon celui défini pour l'ensemble du groupe d'utilisateurs.
Provisioned	La date à laquelle l'utilisateur a été provisionné pour la dernière fois est affichée. Utilisé pour les utilisateurs créés par provisionnement JIT depuis LDAP/SAML.
Info	Les informations sur les erreurs sont affichées. Un avertissement jaune est affiché pour les utilisateurs sans groupes d'utilisateurs. Un avertissement rouge est affiché 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 *Create user* dans le coin supérieur droit.

Options de modification en masse

Les boutons situés sous la liste offrent plusieurs options de modification en masse :

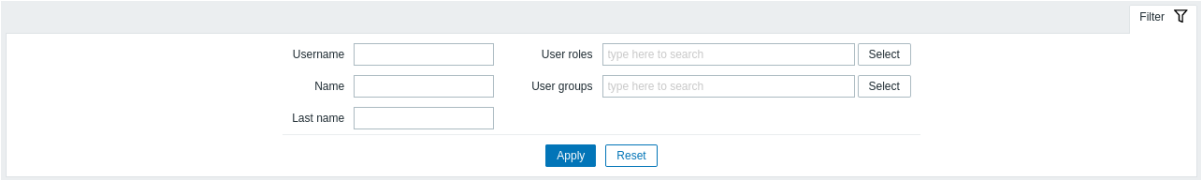
- *Provision now* - mettre à jour les informations utilisateur depuis LDAP (cette option n'est activée que si un utilisateur LDAP est sélectionné)
- *Reset TOTP secret* - réinitialiser les secrets TOTP de l'utilisateur pour toutes les méthodes TOTP et supprimer la session utilisateur (cette option n'est activée que si MFA est activé ; pour les utilisateurs sans secrets TOTP, leur session ne sera pas supprimée)
- *Unblock* - réactiver l'accès au système pour les utilisateurs bloqués
- *Delete* - 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.

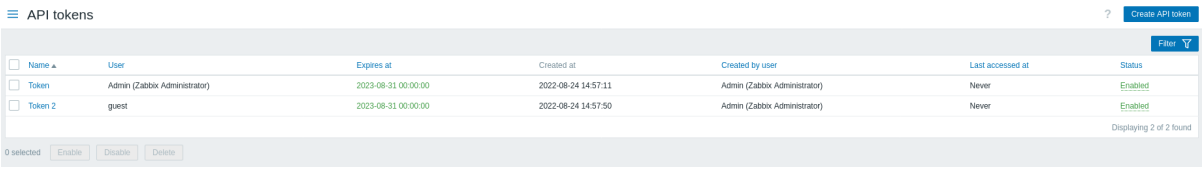
Le lien *Filtre* est disponible au-dessus de la liste des utilisateurs. Si vous cliquez dessus, un filtre s'affiche, dans lequel vous pouvez filtrer les utilisateurs par nom d'utilisateur, prénom, nom, rôle utilisateur et groupe d'utilisateurs.



4 Jetons d'API

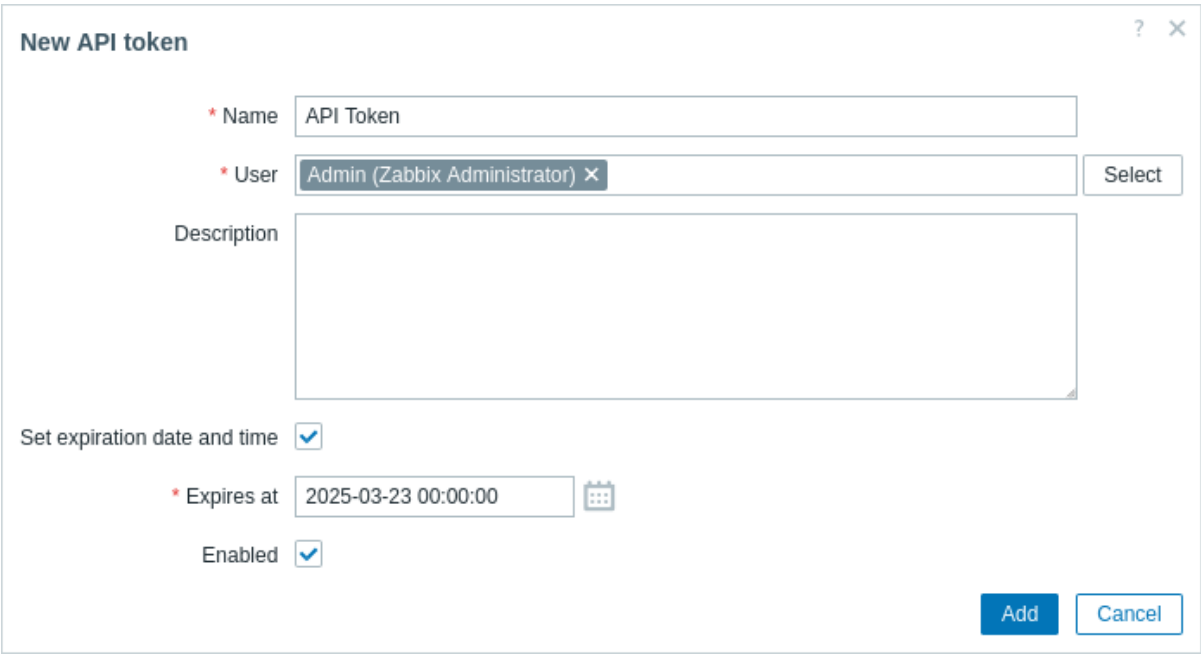
Aperçu

Cette section permet de créer et de gérer des jetons d'API.



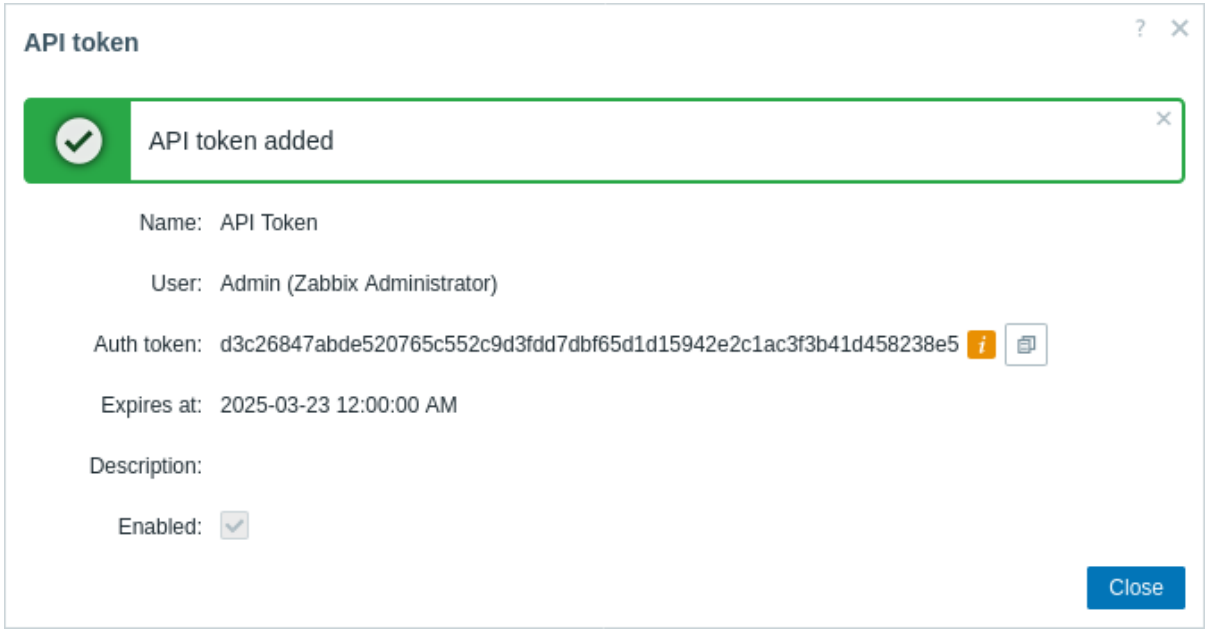
Vous pouvez filtrer les jetons d'API par nom, par utilisateurs auxquels les jetons sont attribués, par date d'expiration, par utilisateurs ayant créé les jetons, ou par statut (activé/désactivé). Cliquez sur le statut du jeton dans la liste pour l'activer/le désactiver rapidement. Vous pouvez également activer/désactiver plusieurs jetons à la fois en les sélectionnant dans la liste, puis en cliquant sur les boutons *Activer/Désactiver* sous la liste.

Pour créer un nouveau jeton, cliquez sur le bouton *Créer un jeton d'API* dans le coin supérieur droit, puis renseignez les champs requis dans l'écran de configuration du jeton :



Parameter	Description
Nom	Nom visible du jeton.
Utilisateur	Utilisateur auquel le jeton doit être attribué. Pour sélectionner rapidement un utilisateur, commencez à saisir le nom d'utilisateur, le prénom ou le nom, puis sélectionnez l'utilisateur requis dans la liste d'autocomplétion. Vous pouvez également cliquer sur le bouton *Sélectionner* et choisir 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.
Date d'expiration	Cliquez sur l'icône du calendrier pour sélectionner la date d'expiration du jeton ou saisissez la date manuellement au format YYYY-MM-DD hh:mm:ss
Activé	Décochez cette case si vous devez créer un jeton à l'état désactivé.

Cliquez sur *Ajouter* pour créer un jeton.



Copiez la valeur du *Jeton d'authentification* et enregistrez-la dans un endroit 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 affichée 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 son nom, sa description, les paramètres de date d'expiration ou son statut. 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 s'affichera et vous demandera de confirmer cette opération, car après validation, le jeton précédemment généré deviendra invalide.

Les utilisateurs n'ayant pas accès à la section de menu *Administration* peuvent voir et modifier les détails des jetons qui leur sont attribués dans la section *Profil utilisateur → Jetons d'API* uniquement si *Gérer les jetons d'API* est autorisé dans les permissions de leur rôle utilisateur.

5 Authentification

Aperçu

La section *Utilisateurs → Authentification* permet de spécifier la méthode d'authentification des utilisateurs pour Zabbix ainsi que les exigences internes relatives aux mots de passe.

Les méthodes d'authentification disponibles sont l'authentification interne, HTTP, LDAP, SAML et MFA.

Authentification par défaut

Par défaut, Zabbix utilise l'authentification Zabbix **interne** pour tous les utilisateurs.

Il est possible de modifier la méthode d'authentification par défaut à l'échelle du système vers LDAP. Pour ce faire, accédez à l'onglet *LDAP* et configurez les paramètres LDAP, puis revenez à l'onglet *Authentification* et basculez le sélecteur *Authentification par défaut* sur LDAP.

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. Ces groupes doivent avoir l'accès au frontend défini sur Interne.

Il est également possible d'activer l'authentification LDAP uniquement pour des groupes d'utilisateurs spécifiques, si l'authentification interne est utilisée globalement. Dans ce cas, les détails de l'authentification LDAP peuvent être spécifiés et utilisés pour des groupes d'utilisateurs spécifiques dont l'accès au frontend doit alors être défini sur LDAP. Si un utilisateur est inclus dans au moins un groupe d'utilisateurs avec authentification LDAP, cet utilisateur 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.

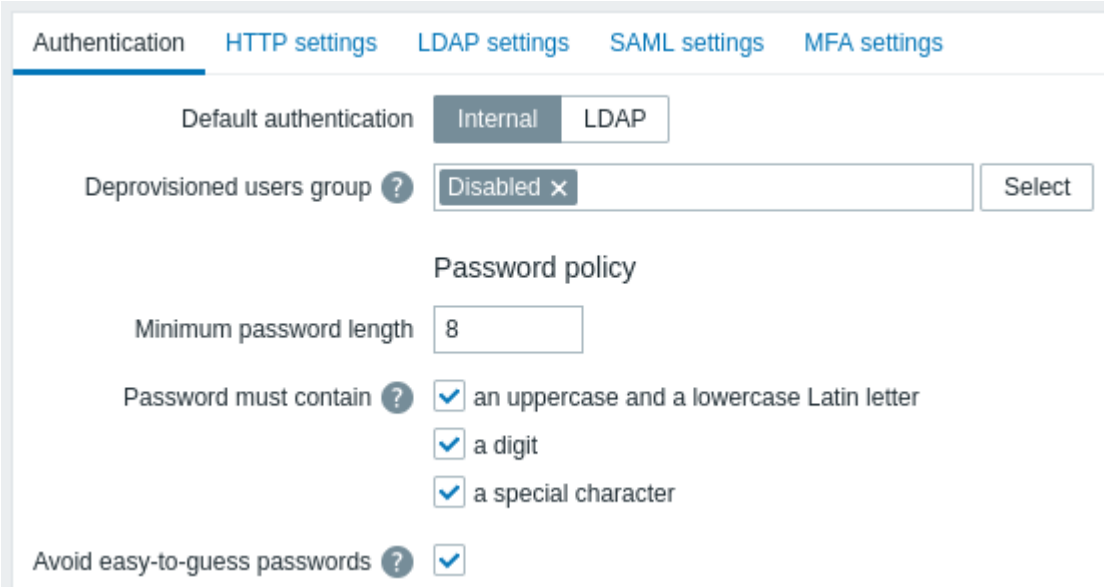
Zabbix prend en charge le provisionnement juste-à-temps (JIT), qui permet de créer des comptes utilisateurs dans Zabbix lors de la première authentification d'un utilisateur externe et de provisionner ces comptes utilisateurs. Le provisionnement JIT est pris en charge pour LDAP et SAML.

Voir aussi :

- 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 n'est requis que pour le provisionnement JIT, concernant les utilisateurs qui ont été créés dans Zabbix à partir de systèmes LDAP ou SAML, mais qui n'ont plus besoin d'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 info-bulle contenant la liste des caractères pour 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 NCSC des « 100k mots de passe les plus utilisés », de la liste SecLists des « 1M mots de passe les plus utilisés » et de la liste des mots de passe spécifiques au contexte de Zabbix. Les utilisateurs internes ne seront pas autorisés à définir des mots de passe inclus dans cette liste, car ces mots de passe sont considérés comme faibles en raison de leur usage courant.

Les modifications des exigences de complexité des mots de passe n'affecteront pas les mots de passe des utilisateurs existants, mais si un utilisateur existant choisit de modifier son mot de passe, le nouveau mot de passe devra répondre aux exigences actuelles. Une indication contenant la liste des exigences sera affichée à côté du champ *Password* dans le profil utilisateur et dans le formulaire de configuration utilisateur accessible depuis le menu *Utilisateurs → Utilisateurs*.

1 HTTP

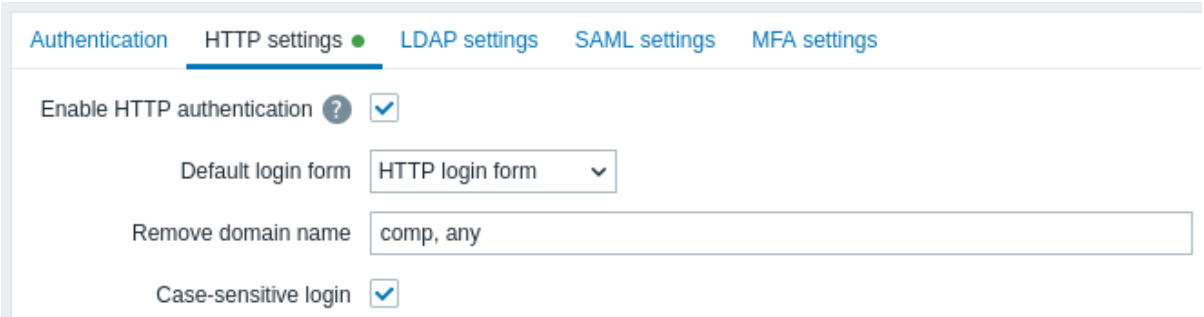
Aperçu

L'authentification HTTP ou basée sur le serveur web (par exemple : BasicAuthentication, NTLM/Kerberos) peut être utilisée pour vérifier les noms d'utilisateur et les mots de passe. Notez toutefois 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.

L'authentification HTTP peut être désactivée dans le frontend en configurant l'option correspondante dans l'onglet *Paramètres HTTP* de la section *Utilisateurs > Authentification*. Lorsque l'authentification HTTP est désactivée, l'onglet contenant les options d'authentification HTTP ne sera pas affiché dans le frontend. Notez que la réinstallation du frontend (exécution de setup.php) réinitialisera les paramètres d'authentification, y compris la configuration de l'authentification HTTP.

Configuration



Paramètres de configuration :

Paramètre	Description
Activer l'authentification HTTP	Cochez la case pour activer l'authentification HTTP. En survolant ?, une infobulle s'affichera pour avertir que, dans le cas d'une authentification par serveur web, tous les utilisateurs (même avec l'accès au frontend défini sur LDAP/Interne) seront authentifiés par le serveur web, et non par Zabbix.
Formulaire de connexion par défaut	Indiquez s'il faut rediriger les utilisateurs non authentifiés 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 identifiant utilisateur valide dans la variable `$_SERVER`. Les clés `$_SERVER` prises en charge sont `PHP_AUTH_USER`, `REMOTE_USER`, `AUTH_USER`.

Paramètre	Description
Supprimer le nom de domaine	Une liste de noms de domaine séparés par des virgules qui doivent être supprimés du nom d'utilisateur. Par ex. `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 sensibilité à la casse pour 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 qu'il existe plusieurs utilisateurs Zabbix avec des noms d'utilisateur similaires (par ex. Admin et admin), la connexion de ces utilisateurs sera toujours refusée avec le message d'erreur suivant : "Authentication failed: supplied credentials are not unique."

Note:
Pour les utilisateurs internes qui ne peuvent pas se connecter à l'aide des identifiants HTTP (avec le formulaire de connexion HTTP défini par défaut), ce qui entraîne l'erreur 401, vous pouvez ajouter une ligne `ErrorDocument 401 /index.php?form=default` aux directives d'authentification de base, ce qui redirigera vers le formulaire de connexion Zabbix habituel.

2 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 (just-in-time) pour les 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 LDAP et son mot de passe, 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 par défaut au moment de sa création. Modifier ultérieurement le serveur LDAP par défaut ne modifie ni ne met à jour le serveur LDAP lié 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 *Authentication*.

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 à un autre, l'utilisateur sera également déplacé d'un groupe à un autre dans Zabbix ; si un utilisateur est supprimé d'un groupe LDAP, il 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. 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 le frontend Zabbix lorsque l'utilisateur interagit avec celui-ci ou a au moins une page du frontend ouverte dans le navigateur. Il n'existe pas de processus d'arrière-plan dédiés au provisionnement des utilisateurs.

LDAP prend en charge trois façons de se lier 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.

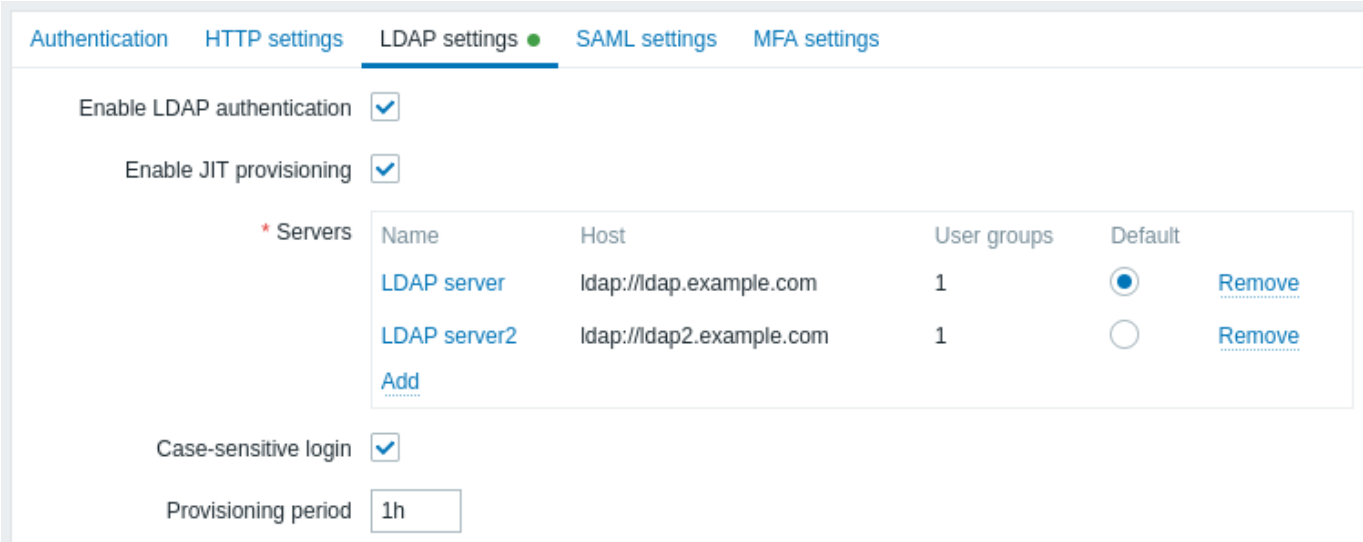
- Utilisateur de liaison dédié (compte de service) — un compte LDAP spécifique est défini dans *Bind DN* / *Bind password* et est utilisé par Zabbix pour la recherche et le provisionnement. Il s'agit de l'option recommandée et la plus flexible, car Zabbix peut effectuer des recherches et un provisionnement en arrière-plan sans les identifiants des utilisateurs finaux.
- Liaison directe de l'utilisateur — Zabbix effectue la liaison à l'aide des identifiants que l'utilisateur saisit lors de la connexion (aucun *Bind DN* / *Bind password* configuré) ; cela se configure en incluant un espace réservé tel que `uid=%{user}` dans le DN de base. 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 du provisionnement en arrière-plan lorsque l'utilisateur n'est pas activement connecté) ne peuvent pas s'authentifier et ne fonctionneront donc pas. Le provisionnement et les mises à jour pour la liaison directe de l'utilisateur n'ont lieu qu'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.

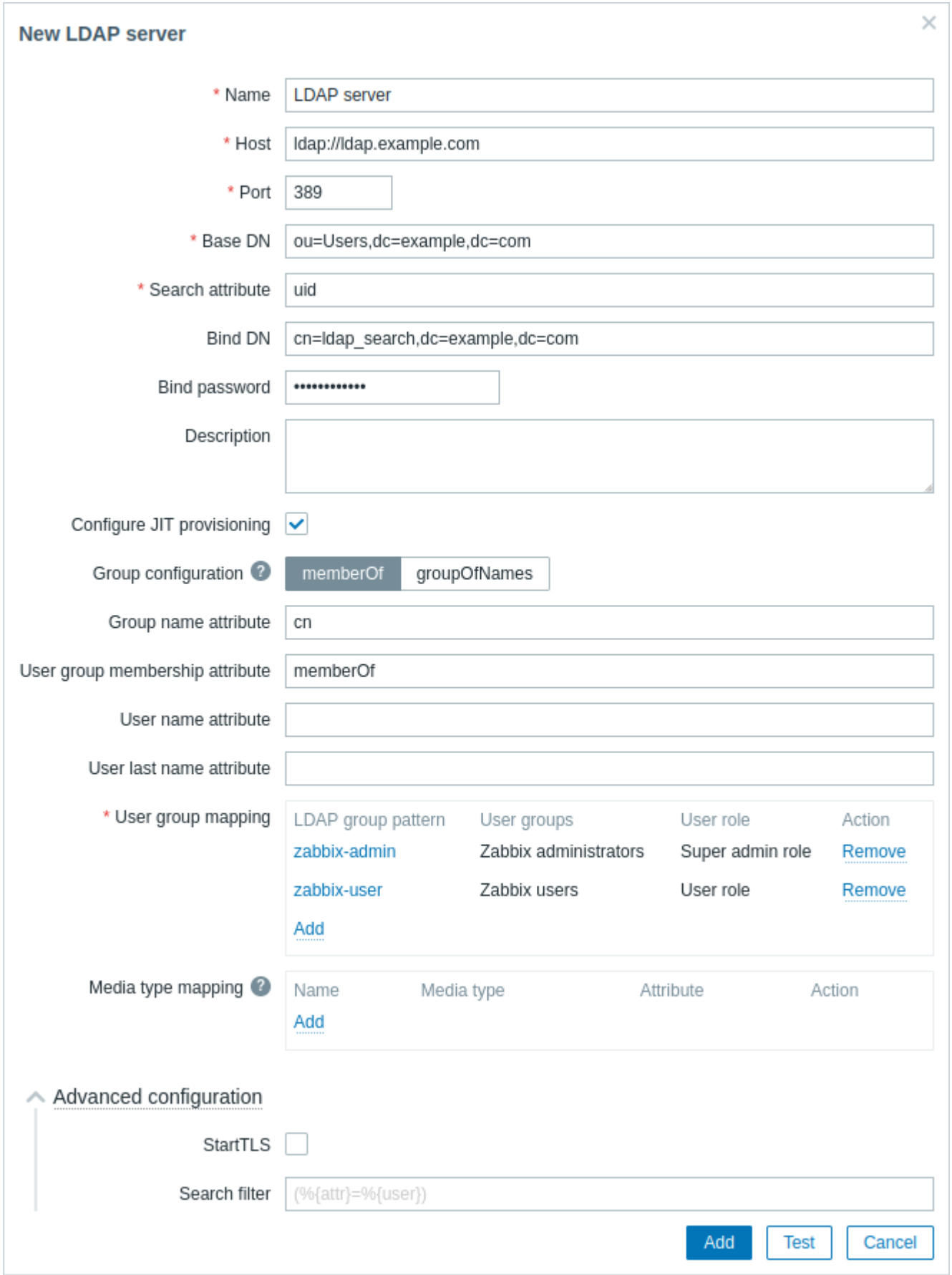
Configuration



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).
Connexion sensible à la casse	Décochez la case pour désactiver la prise en compte de la casse pour les noms d'utilisateur (activée par défaut). La désactivation de la prise en compte de 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 prise en compte de la casse est désactivée et qu'il existe plusieurs utilisateurs Zabbix avec 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 informations d'identification fournies ne sont pas uniques. »
Période de provisionnement	Définissez la période de provisionnement, c'est-à-dire la fréquence à laquelle l'utilisateur connecté sera provisionné pendant qu'il travaille avec l'interface web.

Configuration du serveur LDAP



Paramètres de configuration du serveur LDAP :

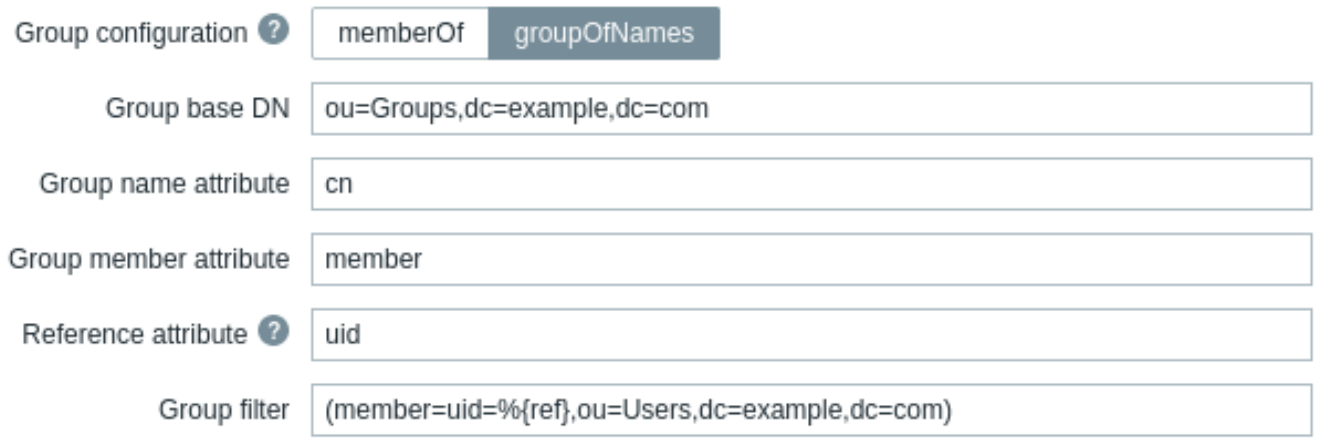
Parameter	Description
Name	Nom du serveur LDAP dans la configuration de Zabbix.

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 le 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 généralement 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 la liaison directe de l'utilisateur, voir la note ci-dessous)
Search attribute	Attribut de 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 peut potentiellement exposer 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 à la place 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 à un groupe **groupOfNames** - en recherchant les groupes via l'attribut member Notez que memberOf est préférable car 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 à partir duquel obtenir le nom du groupe pour 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 les informations sur les groupes auxquels l'utilisateur appartient (par ex. `memberOf`). Par exemple, l'attribut memberOf peut contenir des informations comme ceci : `memberOf=cn=zabbix-admin,ou=Groups,dc=example,dc=com` Ce champ n'est disponible que 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	Mappez un modèle de groupe d'utilisateurs LDAP à un groupe d'utilisateurs Zabbix et à un rôle utilisateur. Cela est nécessaire pour déterminer quel groupe/rôle l'utilisateur provisionné obtiendra 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, il devient membre de tous ces groupes. Si un utilisateur correspond à plusieurs rôles utilisateur Zabbix, il recevra celui ayant le niveau d'autorisation le plus élevé parmi eux.
Media type mapping	Mappez les attributs LDAP media de l'utilisateur (par ex. l'e-mail) aux médias utilisateur Zabbix pour l'envoi de notifications (la valeur de l'attribut est utilisée comme champ *Send to* du média).
Advanced configuration	Cliquez sur le libellé *Advanced configuration* pour afficher les options de configuration avancée (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 la **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 informations d'identification 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 à l'aide de 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 soit connecté.

Les champs suivants sont spécifiques à "groupOfNames" comme méthode *Group configuration* :



Parameter	Description
Group base DN	Chemin de base vers les groupes sur le serveur LDAP.
Group name attribute	Spécifiez l'attribut à partir duquel obtenir le nom du groupe dans le chemin de base spécifié vers les groupes. Le nom du groupe est nécessaire pour le mappage des groupes d'utilisateurs.
Group member attribute	Spécifiez l'attribut qui contient les informations sur les membres du groupe dans LDAP (par ex. `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 pour 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 se connecter 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 *Test* permet de tester l'accès de l'utilisateur :

Parameter	Description
Login	Nom d'utilisateur LDAP à tester (prérempli avec le nom d'utilisateur actuel du frontend Zabbix). Ce nom d'utilisateur doit exister sur le serveur LDAP. Zabbix n'activera pas l'authentification LDAP s'il n'est pas en mesure d'authentifier l'utilisateur de test.
User password	Mot de passe de l'utilisateur LDAP à tester.

3 SAML

Aperçu

L'authentification SAML 2.0 peut être utilisée pour se connecter à Zabbix.

Si seule la connexion SAML 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 SAML.

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 le frontend.

Pour utiliser l'authentification SAML, Zabbix doit être configuré de la manière suivante :

1. Fournissez la clé privée SP et les certificats. Un certificat IdP doit être fourni (coller ou choisir un fichier) pour activer SAML via le frontend. Un certificat SP et une clé privée SP doivent être fournis si les options *Sign* ou *Encrypt* sont sélectionnées.

Si `$SSO['CERT_STORAGE'] = 'database'` est défini dans zabbix.conf.php, vous pouvez coller le texte du certificat ou téléverser le fichier de certificat dans le frontend pendant la configuration SAML — aucun fichier n'est requis sur le système de fichiers.

Si `$SSO['CERT_STORAGE'] = 'file'` est défini dans zabbix.conf.php, le certificat doit être disponible sur le système de fichiers (par défaut dans ui/conf/certs ou dans le chemin configuré dans zabbix.conf.php) et le frontend ne stockera pas les certificats dans la base de données. Notez que si `$SSO['CERT_STORAGE']` n'est pas défini ou est commenté, le stockage dans des fichiers est supposé.

Par défaut, Zabbix recherchera 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:
Le matériel de certificat peut être importé via le frontend en collant le texte ou en téléversant des fichiers. Bien que les importations non chiffrées soient autorisées pour des raisons de commodité dans certains environnements, l'utilisation d'un transport chiffré est fortement recommandée (par exemple, activez HTTPS/TLS pour le frontend Zabbix). Lorsque les certificats/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 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 le matériel SAML. Limitez les privilèges de l'utilisateur de la base de données au strict minimum nécessaire.

2. Tous les paramètres les plus importants peuvent être configurés dans le frontend Zabbix. Cependant, il est possible de spécifier des paramètres supplémentaires dans le fichier de configuration.



Paramètres de configuration disponibles dans le frontend Zabbix :

Parameter	Description
Enable SAML authentication	Cochez la case pour activer l'authentification SAML.
Enable JIT provisioning	Cochez la case pour activer le provisionnement utilisateur JIT.
IDP entity ID	L'identifiant d'entité unique au sein du fournisseur d'identité SAML.
SSO service URL	L'URL vers laquelle les utilisateurs seront redirigés lors de la connexion.
SLO service URL	L'URL vers laquelle les utilisateurs seront redirigés lors de la déconnexion. Si elle est laissée vide, le service SLO ne sera pas utilisé.

Parameter	Description
Username attribute	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
SP entity ID	L'identifiant unique du fournisseur de services (s'il ne correspond pas, l'opération sera rejetée). Il est possible de spécifier une URL ou n'importe quelle chaîne de données.
SP name ID format	Demande un format name ID 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
IdP certificate	Contenu du certificat du fournisseur de services (SP) pour la configuration du service d'authentification unique SAML (SSO).
SP private key	Contenu de la clé privée du fournisseur de services (SP) pour la configuration du service d'authentification unique SAML (SSO). Fournit une authentification sécurisée et un échange de données avec le fournisseur d'identité (IdP).
SP certificate	Contenu du certificat du fournisseur de services (SP) pour la configuration du service d'authentification unique SAML (SSO).
Sign	Cochez les cases pour sélectionner les entités pour lesquelles la signature SAML doit être activée : *Messages* *Assertions* *AuthN requests* *Logout requests* *Logout responses*
Encrypt	Cochez les cases pour sélectionner les entités pour lesquelles le chiffrement SAML doit être activé : *Name ID* *Assertions*
Case-sensitive login	Décochez la case pour désactiver la sensibilité à la casse pour les noms d'utilisateur lors de la connexion (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 qu'il existe plusieurs utilisateurs Zabbix avec 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."
Configure JIT provisioning	Cochez cette case pour afficher les options liées au provisionnement utilisateur JIT.
Group name attribute	Spécifiez l'attribut de nom de groupe pour le provisionnement utilisateur JIT.
User name attribute	Spécifiez l'attribut de prénom de l'utilisateur pour le provisionnement utilisateur JIT.
User last name attribute	Spécifiez l'attribut de nom de famille de l'utilisateur pour le provisionnement utilisateur JIT.
User group mapping	Mappez un modèle de groupe d'utilisateurs SAML à un groupe d'utilisateurs Zabbix et à un rôle utilisateur. Cela est nécessaire pour déterminer quel groupe/rôle l'utilisateur provisionné obtiendra dans Zabbix. Cliquez sur *Add* pour ajouter un mappage. Le champ *SAML group pattern* 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, l'utilisateur deviendra membre de tous ces groupes. Si un utilisateur correspond à plusieurs rôles utilisateur Zabbix, il obtiendra le niveau d'autorisation le plus élevé parmi eux.

Parameter	Description
Media type mapping	Mappez les attributs média SAML de l'utilisateur (par ex. e-mail) aux médias utilisateur Zabbix pour l'envoi de notifications.
Enable SCIM provisioning	Cochez cette case pour activer le provisionnement SCIM 2.0.

Consultez des exemples de configuration de fournisseurs d'identité SAML pour la connexion et le provisionnement des utilisateurs dans Zabbix avec :

- Microsoft Azure AD
- Okta
- Onelogin

Remarques sur l'approvisionnement SCIM

Pour l'approvisionnement SCIM, indiquez le chemin vers le frontend Zabbix et ajoutez-y api_scim.php 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 à celui du 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 du frontend Zabbix (*zabbix.conf.php*) :

- $SSO['SP_KEY'] = '<chemin vers le fichier de clé privée du SP>';
- $SSO['SP_CERT'] = '<chemin vers le fichier de certificat du SP>';
- $SSO['IDP_CERT'] = '<chemin vers le fichier de certificat de l'IDP>';
- $SSO['SETTINGS']

Note:
Zabbix utilise la bibliothèque OneLogin's SAML PHP Toolkit (version 3.4.1). La structure de la section $SSO['SETTINGS'] doit être similaire à la structure utilisée par la bibliothèque. Pour la description des options de configuration, consultez la documentation officielle de la bibliothèque.

Seules les options suivantes peuvent être définies 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 seront prises depuis la base de données et ne peuvent pas être remplacées. L'option *debug* sera ignorée.

En outre, si l'interface utilisateur Zabbix se trouve derrière un proxy ou un équilibreur de charge, l'option personnalisée *use_proxy_headers* peut être utilisée :

- *false* (par défaut) - ignorer l'option ;
- *true* - utiliser les en-têtes HTTP X-Forwarded-* pour construire l'URL de base.

Si vous utilisez un équilibreur de charge pour vous connecter à une instance Zabbix, où l'équilibreur de charge utilise TLS/SSL et Zabbix non, vous devez indiquer 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 de configuration :

```
$SSO['SETTINGS'] = [
    'security' => [
        'signatureAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#rsa-sha384'
        'digestAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#sha384',
        // ...
    ],
    // ...
];
```

Configuration de l'interface avec Kerberos/ADFS

Le fichier de configuration de l'interface Zabbix (*zabbix.conf.php*) peut être utilisé pour configurer le SSO avec l'authentification Kerberos et ADFS :

```
$SSO['SETTINGS'] = [
    'security' => [
        'requestedAuthnContext' => [
            'urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos',
        ],
        'requestedAuthnContextComparison' => 'exact'
    ]
];
```

Dans ce cas, dans la configuration SAML, définissez le champ *SP name ID* sur :

```
urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified
```

4 MFA

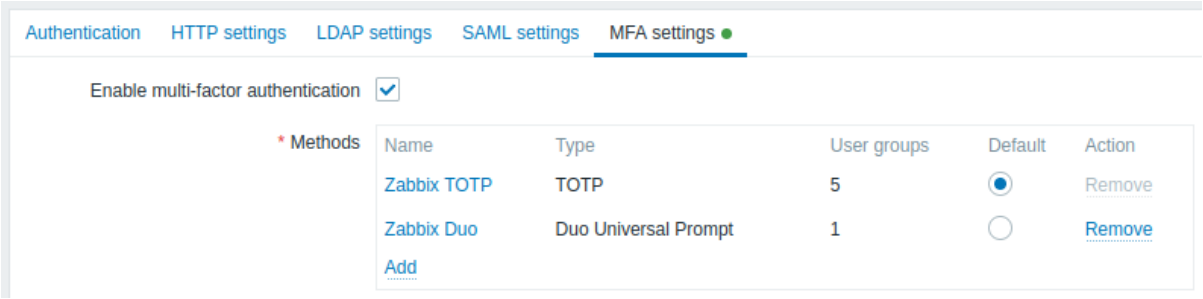
Aperçu

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.

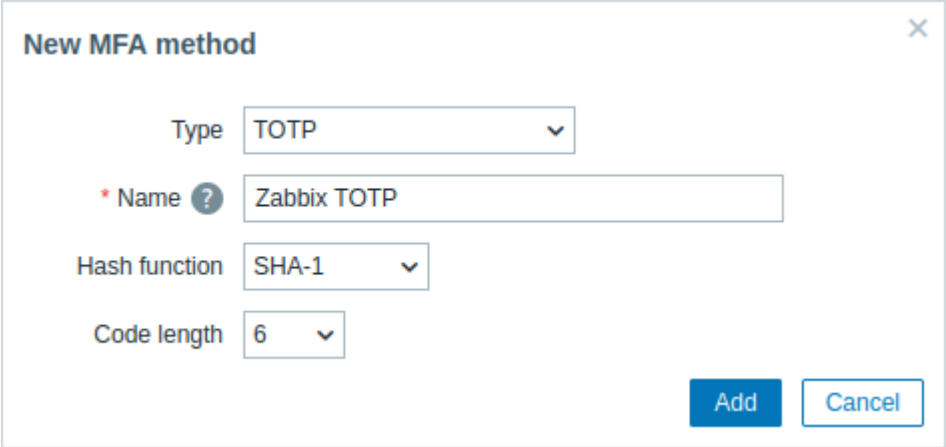
Configuration



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
   - Nom : Zabbix TOTP
   - Fonction de hachage : SHA-1
   - Longueur du code : 6

3. Cliquez sur le bouton *Add*, puis sur le bouton *Update*.

4. Accédez à *Users* → *User groups* et créez un nouveau groupe d'utilisateurs avec la configuration suivante :

   - Nom du groupe : Groupe TOTP
   - Utilisateurs : Admin
   - Authentification multifacteur : Default (ou "Zabbix TOTP" si elle n'est pas définie par défaut)

5. Déconnectez-vous de Zabbix, puis reconnectez-vous à l'aide de vos identifiants. Après une connexion réussie, il vous sera demandé de vous inscrire à la MFA ; un code QR et une clé secrète s'afficheront.



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.

Invite 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, un accès à Zabbix via HTTPS, ainsi que l'autorisation des connexions sortantes vers les serveurs Duo. De plus, si vous avez activé la 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. Créez un compte administrateur Duo gratuit sur Duo Signup.

2. Ouvrez le panneau d'administration Duo, accédez à *Applications* → *Protect an Application*, recherchez l'application *Web SDK*, puis cliquez sur *Protect*.

3. Notez les identifiants (Client ID, Client secret, API hostname) requis 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. Accédez à *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 ce n'est pas défini 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 et vous serez redirigé vers Duo. Terminez la configuration Duo et vérifiez votre identité avec l'application Duo sur votre téléphone pour vous connecter.

9. Lors des connexions suivantes, utilisez la méthode MFA appropriée fournie par l'application Duo (par exemple en récupérant un code de vérification, en répondant aux notifications push ou en utilisant des clés matérielles), puis saisissez les informations requises lors de la connexion.

9 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.

1 Général

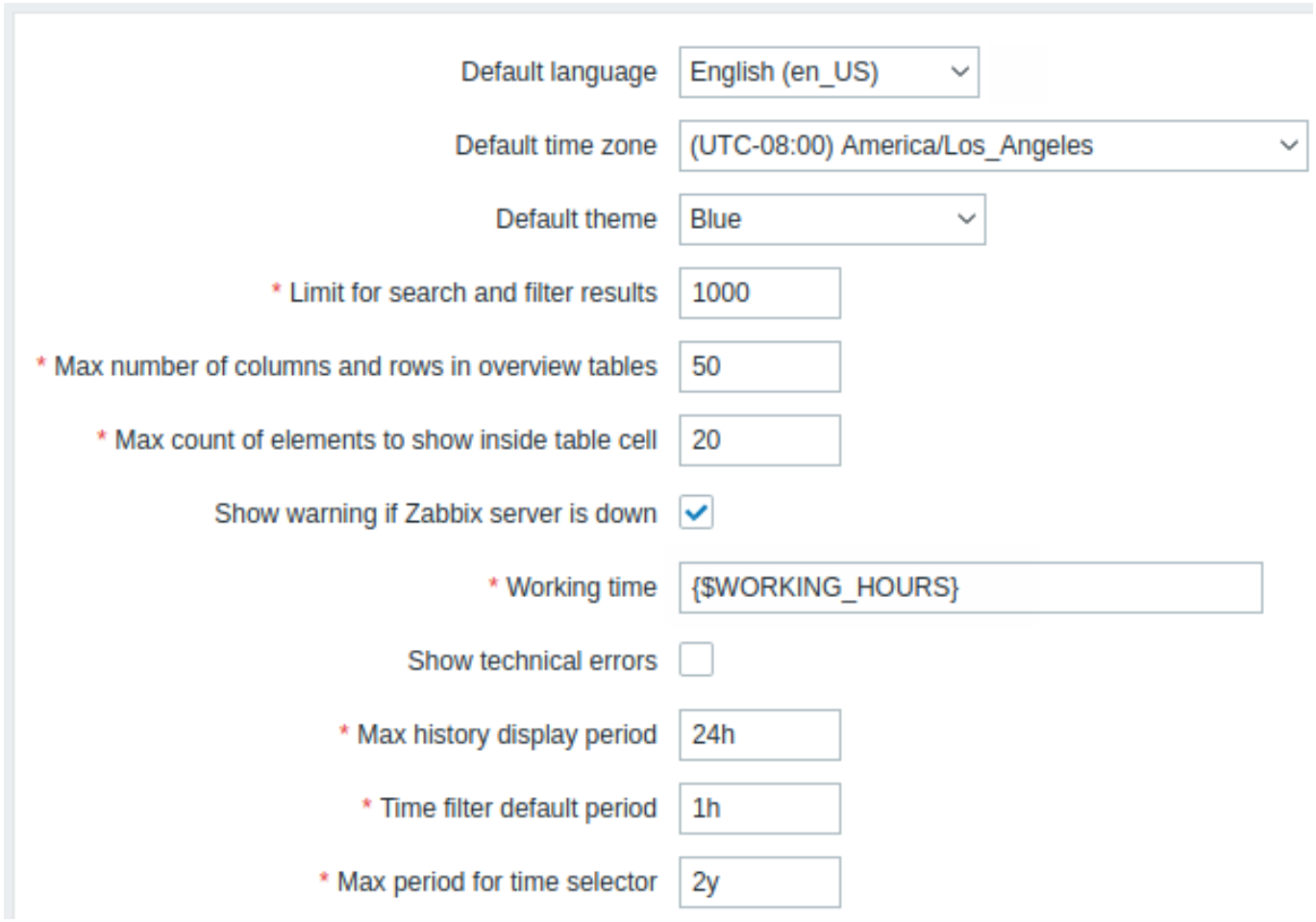
Vue d'ensemble

La section *Administration* → *General* contient un certain nombre de sous-sections permettant de définir les valeurs par défaut liées à l'interface web et de personnaliser Zabbix.

La liste des sous-sections disponibles s'affiche lorsque vous cliquez sur *General* dans la section *Administration* du menu. Il est également possible de passer d'une sous-section à l'autre à l'aide de la liste déroulante du titre dans le coin supérieur gauche.

Interface graphique

Cette section permet de personnaliser plusieurs valeurs par défaut liées au frontend.



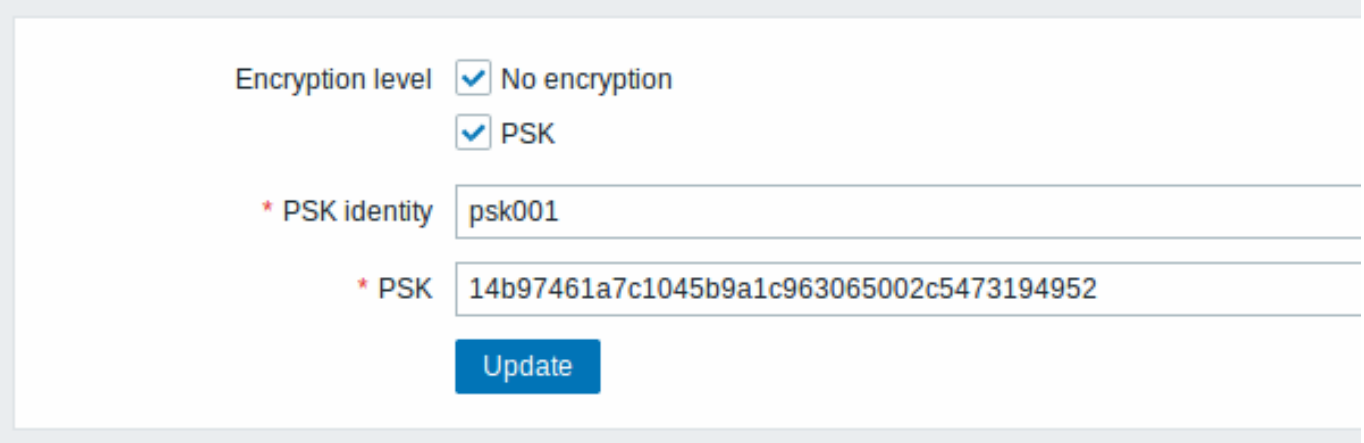
Paramètres de configuration :

Parameter	Description
Langue par défaut	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 du frontend.
Fuseau horaire par défaut	Fuseau horaire par défaut pour les utilisateurs qui n'ont pas spécifié de fuseau horaire dans leur profil ainsi que pour les utilisateurs invités.
Thème par défaut	Thème par défaut pour les utilisateurs qui n'ont pas spécifié de thème dans leur profil ainsi que pour les utilisateurs invités.
Limite pour les résultats de recherche et de filtre	Nombre maximal d'éléments (lignes) qui seront affichés dans une liste de l'interface web, par exemple dans *Collecte de données* > *Hôtes*. *Remarque* : si cette valeur est définie, par exemple, sur « 50 », seuls les 50 premiers éléments seront affichés dans toutes les listes du frontend concernées. Si une liste contient plus de cinquante éléments, cela sera indiqué par le signe « + » dans *"Displaying 1 to 50 of **50+** found"*. De plus, si un filtre est utilisé et qu'il y a toujours plus de 50 correspondances, seules les 50 premières seront affichées. Notez que l'augmentation de la valeur de ce paramètre peut entraîner une baisse des performances et une augmentation de la consommation mémoire côté serveur web.
Nombre maximal de colonnes et de lignes dans les tableaux d'aperçu	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. S'il existe plus de lignes et/ou de colonnes que celles affichées, le système affichera un avertissement au bas du tableau : « Not all results are displayed. Please provide more specific search criteria. »
Nombre maximal d'éléments à afficher dans une cellule de tableau	Pour les entrées affichées dans une seule cellule de tableau, le nombre affiché ne dépassera pas la valeur configurée ici.
Afficher un avertissement si le serveur Zabbix est indisponible	Ce paramètre permet d'afficher un message d'avertissement dans une fenêtre du navigateur si le serveur Zabbix ne peut pas être atteint (éventuellement hors service). Le message reste visible même si l'utilisateur fait défiler la page vers le bas. Au survol, le message est temporairement masqué afin de révéler le contenu situé en dessous.

Parameter	Description
Temps de travail	Ce paramètre global du système définit les heures de travail. Dans les graphiques, le temps de travail est affiché sur fond blanc et le temps hors travail sur fond gris. Voir la page Spécification de période pour une description du format de l'heure. Les macros utilisateur sont prises en charge.
Afficher les erreurs techniques	Si cette case est cochée, tous les utilisateurs enregistrés pourront voir les erreurs techniques (PHP/SQL). Si elle n'est pas cochée, l'information n'est disponible que pour les Super administrateurs Zabbix et les utilisateurs appartenant à des groupes d'utilisateurs pour lesquels le mode débogage est activé.
Période maximale d'affichage de l'historique	Période maximale pendant laquelle afficher les données historiques dans *Surveillance > Dernières données*, les informations de scénario web de l'hôte dans *Surveillance > Hôtes*, ainsi que dans le widget de tableau de bord *Top items*. Plage autorisée : 24 heures (par défaut) à 1 semaine. Les suffixes temporels, par exemple 1w (une semaine), 36h (36 heures), sont pris en charge.
Période par défaut du filtre temporel	Période à utiliser par défaut dans les graphiques et les tableaux de bord. Plage autorisée : 1 minute à 10 ans (par défaut : 1 heure). Les suffixes temporels, par exemple 10m (dix minutes), 5w (cinq semaines), sont pris en charge. Remarque : lorsqu'un utilisateur modifie la période en consultant un graphique, cette période est enregistrée comme préférence utilisateur, remplaçant la valeur par défaut globale ou une sélection utilisateur précédente.
Période maximale pour le sélecteur de temps	Période maximale disponible pour les graphiques et les tableaux de bord. Les utilisateurs ne pourront pas visualiser de données plus anciennes. Plage autorisée : 1 an à 10 ans (par défaut : 2 ans). Les suffixes temporels, par exemple 1y (un an), 365w (365 semaines), sont pris en charge.

Enregistrement automatique

Dans cette section, vous pouvez configurer le niveau de chiffrement pour l'enregistrement automatique de l'agent actif.



Les paramètres marqués d'un astérisque sont obligatoires.

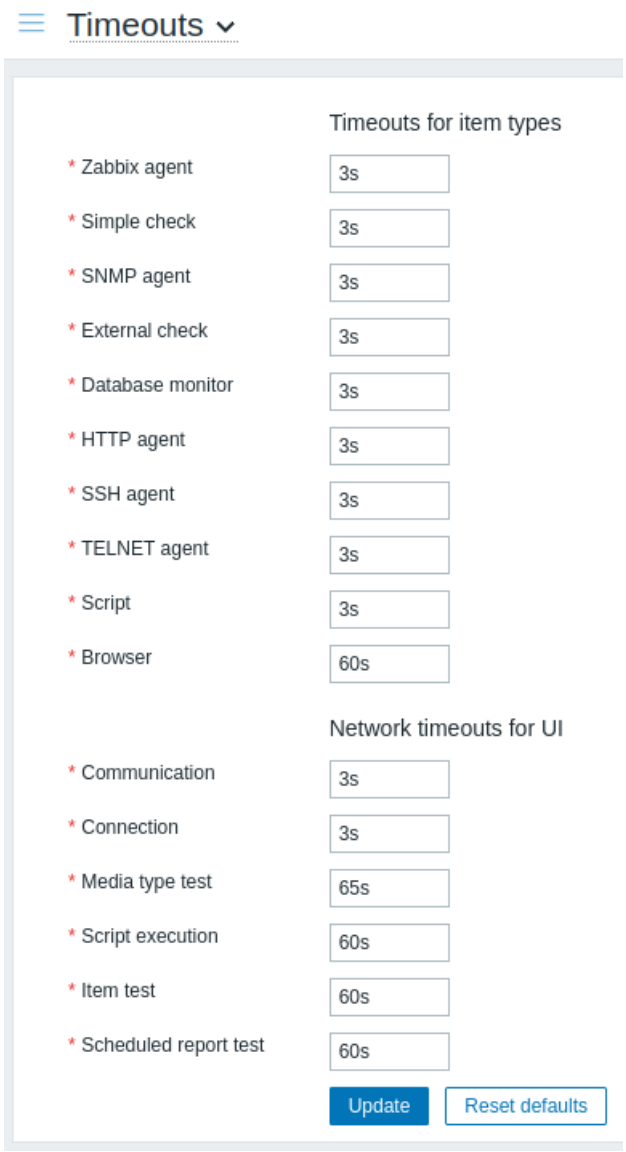
Paramètres de configuration :

Parameter	Description
Niveau de chiffrement	Sélectionnez une ou les deux options pour le niveau de chiffrement : **Aucun chiffrement** - les connexions non chiffrées sont autorisées **PSK** - les connexions chiffrées TLS avec une clé pré-partagée sont autorisées
Identité PSK	Saisissez la chaîne d'identité de la clé pré-partagée. Ce champ n'est disponible que si « PSK » est sélectionné comme *Niveau de chiffrement*. Ne mettez pas d'informations sensibles dans l'identité PSK, car elle est transmise sans chiffrement sur le réseau afin d'informer le récepteur de la PSK à utiliser.
PSK	Saisissez la clé pré-partagée (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 *Niveau de chiffrement*.

Voir aussi : Enregistrement automatique sécurisé

Délais d'attente

Dans cette section, il est possible de définir des délais d'attente globaux pour les types d'éléments ainsi que des délais d'attente réseau. Tous les champs de ce formulaire sont obligatoires.

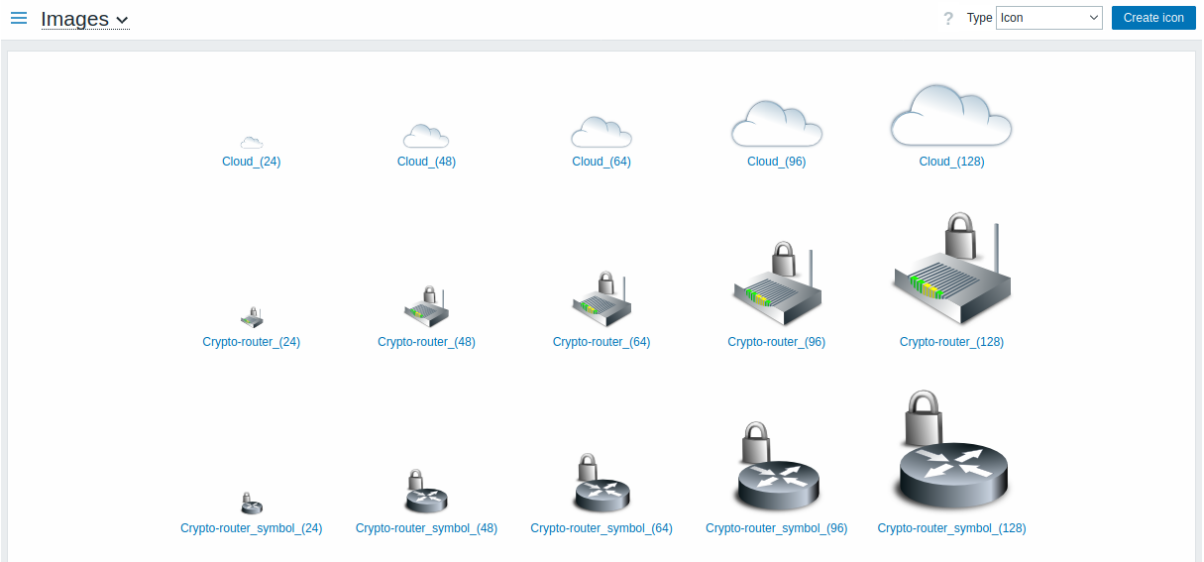


Paramètre		Description
Délais d'attente pour les types d'éléments		Nombre de secondes pendant lesquelles Zabbix doit attendre avant d'abandonner la vérification (selon son type). Pour plus de détails et les limitations, voir délai d'attente d'un élément. Plage autorisée : 1 - 600s (par défaut : 3s ; valeur par défaut pour les éléments browser : 60s). Les suffixes de temps, par ex. 30s, 1m, ainsi que les macros utilisateur sont pris en charge. Notez que les délais d'attente définis au niveau du proxy, s'ils sont configurés, remplacent ces paramètres globaux. En outre, 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 utilisateur		
	Communication	Nombre de secondes à attendre avant de fermer un socket inactif (si une connexion au serveur Zabbix a été établie auparavant, mais que l'interface web ne peut pas terminer l'opération de lecture/envoi des données pendant ce délai, la connexion sera interrompue). Plage autorisée : 1 - 300s (par défaut : 3s).

Paramètre		Description
	Connexion	Nombre de secondes à attendre avant d'arrêter une tentative de connexion au serveur Zabbix. Plage autorisée : 1 - 30s (par défaut : 3s).
	Test du type de média	Nombre de secondes à attendre une réponse lors du test d'un type de média. Plage autorisée : 1 - 300s (par défaut : 65s).
	Exécution du script	Nombre de secondes à attendre une réponse lors de l'exécution d'un script. Plage autorisée : 1 - 300s (par défaut : 60s). Ce délai d'attente s'applique à l'ensemble de la chaîne d'exécution du script, dont la longueur peut varier. Par exemple, si le script est exécuté sur l'agent, il effectue un aller-retour via le serveur (et éventuellement aussi le proxy) jusqu'à l'agent puis en 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 de rafraîchissement des vérifications actives, 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	Nombre de secondes à attendre les données renvoyées lors du test d'un élément. Plage autorisée : 1 - 600s (par défaut : 60s).
	Test de rapport planifié	Nombre de secondes à attendre les données renvoyées lors du test d'un rapport planifié. Plage autorisée : 1 - 300s (par défaut : 60s).

Images

La section Images affiche toutes les images disponibles dans Zabbix. Les images sont stockées dans la base de données.

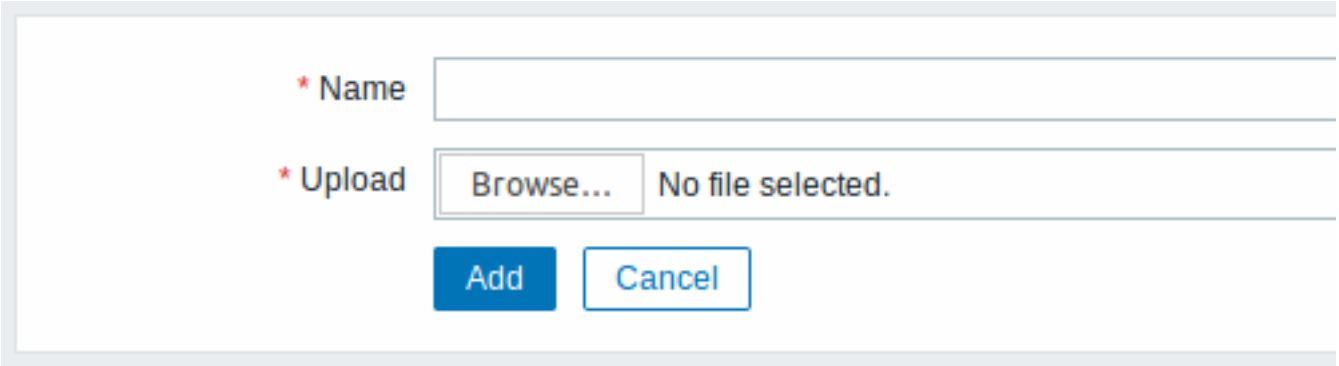


La liste déroulante *Type* vous permet de basculer entre les images d'icône et d'arrière-plan :

- Les icônes sont utilisées pour afficher les éléments de carte 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 *Create icon* ou *Create background* dans le coin supérieur droit.



Attributs de l'image :

Parameter	Description
Name	Nom unique d'une image.
Upload	Sélectionnez le fichier (PNG, JPEG, GIF ou WebP) depuis un système local pour le téléverser vers Zabbix. *Remarque* : il peut être possible de téléverser d'autres formats, qui seront convertis en PNG lors du téléversement. La bibliothèque GD est utilisée pour le traitement des images ; par conséquent, les formats pris en charge dépendent de la version de la bibliothèque utilisée (la version 2.0.28 ou supérieure est requise par Zabbix).

Note:
La taille maximale du fichier téléversé est limitée par la valeur de ZBX_MAX_IMAGE_SIZE, qui est de 1024x1024 octets ou 1 Mo.

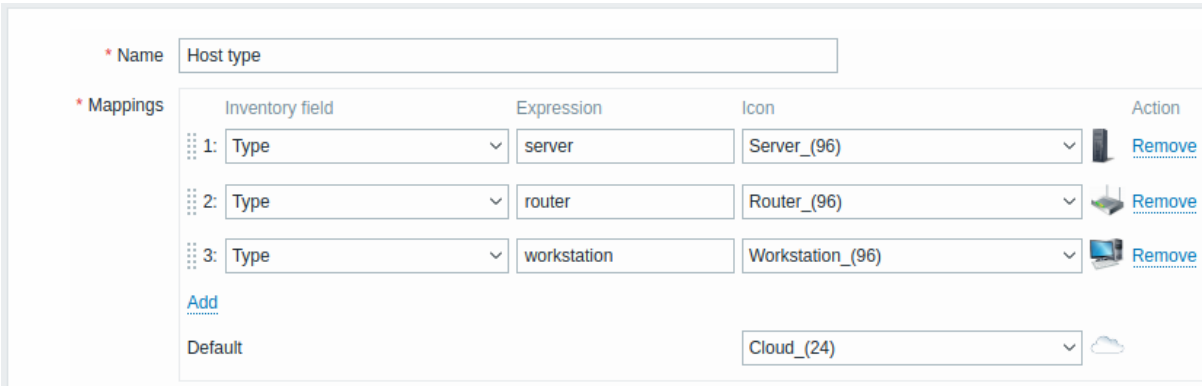
Le téléversement d'une image peut échouer si la taille de l'image est proche de 1 Mo et que le paramètre de configuration MySQL `max_allowed_packet` a sa valeur par défaut de 1 Mo. Dans ce cas, augmentez le paramètre max_allowed_packet.

Correspondance des icônes

Cette section permet de créer la correspondance de certains hôtes avec certaines icônes. Les informations des champs d'inventaire des hôtes sont utilisées pour créer la correspondance.

Les correspondances peuvent ensuite être utilisées dans la configuration des cartes réseau afin d'attribuer automatiquement les icônes appropriées aux hôtes correspondants.

Pour créer une nouvelle correspondance d'icônes, cliquez sur *Créer une correspondance d'icônes* dans le coin supérieur droit.



Paramètres de configuration :

Parameter	Description
Name	Nom unique de la correspondance d'icônes.
Mappings	Liste des correspondances. L'ordre des correspondances détermine laquelle aura la priorité. Vous pouvez déplacer les correspondances vers le haut ou vers le bas de la liste par glisser-déposer.
Inventory field	Champ d'inventaire de l'hôte dans lequel une recherche de correspondance sera effectuée.
Expression	Expression régulière décrivant la correspondance.

Parameter	Description
Icon	Icône à utiliser si une correspondance avec l'expression est trouvée.
Default	Icône par défaut à utiliser.

Expressions régulières

Cette section permet de créer des expressions régulières personnalisées pouvant être utilisées à plusieurs endroits dans l'interface web. Voir la section Expressions régulières pour plus de détails.

Options d'affichage des déclencheurs

Cette section permet de personnaliser la façon dont l'état des déclencheurs est affiché dans le frontend ainsi que les noms et les couleurs de la sévérité des déclencheurs.

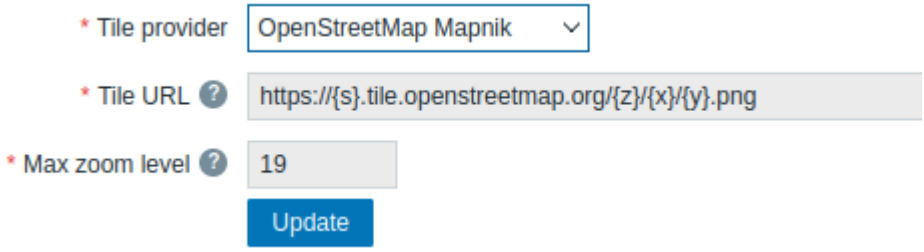


Paramètre	Description
Utiliser des couleurs personnalisées pour l'état des événements	Cocher ce paramètre active la personnalisation des couleurs pour les problèmes acquittés/non acquittés.
Événements PROBLEM non acquittés, *Événements PROBLEM acquittés*, *Événements RESOLVED non acquittés*, *Événements RESOLVED acquittés*	Saisissez un nouveau code couleur ou cliquez sur la couleur pour en sélectionner une nouvelle dans la palette proposée. Si la case *blinking* est cochée, les déclencheurs clignoteront pendant un certain temps lors du changement d'état afin d'être plus visibles.
Afficher les déclencheurs OK pendant	Période pendant laquelle les déclencheurs OK sont affichés. Plage autorisée : 0 - 24 heures. Les suffixes de temps, par ex. 5m, 2h, 1d, sont pris en charge.
Lors d'un changement d'état, les déclencheurs clignotent pendant	Durée du clignotement des déclencheurs. Plage autorisée : 0 - 24 heures. Les suffixes de temps, par ex. 5m, 2h, 1d, sont pris en charge.

Paramètre	Description
Non classé, *Information*, *Avertissement*, *Moyen*, *Élevé*, *Désastre*	Noms et/ou couleurs de sévérité personnalisés 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 proposée. Notez que les noms de sévérité 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 sévérités des déclencheurs.

Cartes géographiques

Cette section permet de sélectionner le fournisseur du service de tuiles cartographiques géographiques et de configurer les paramètres du fournisseur pour le widget Geomap du tableau de bord. Pour fournir une visualisation à l'aide de cartes géographiques, Zabbix utilise Leaflet, une bibliothèque open source de cartes interactives en JavaScript. 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.



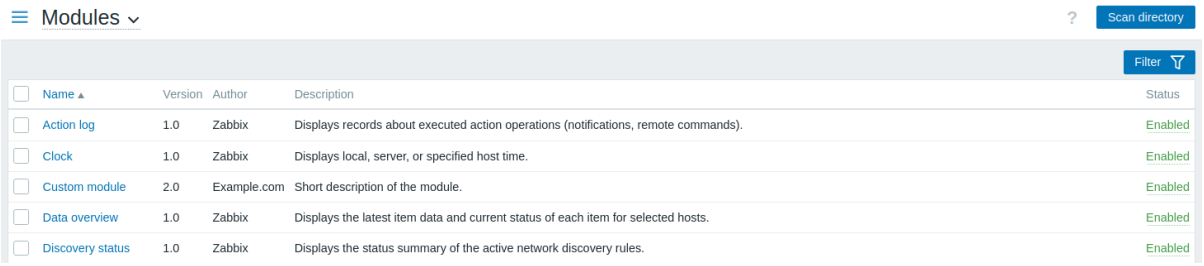
Paramètre	Description
Fournisseur de tuiles	Sélectionnez l'un des fournisseurs de services de tuiles disponibles ou sélectionnez *Autre* pour ajouter un autre fournisseur de tuiles ou des tuiles auto-hébergées (voir Utilisation d'un fournisseur de services de tuiles personnalisé).
URL des tuiles	Le modèle d'URL (jusqu'à 2048 caractères) pour charger et afficher la couche de tuiles sur les cartes géographiques. Ce champ est modifiable uniquement si *Fournisseur de tuiles* est défini sur *Autre*. Les espaces réservés suivants sont pris en charge : *{s}* représente l'un des sous-domaines disponibles ; *{z}* représente le paramètre du 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`
Texte d'attribution	Texte d'attribution du fournisseur de tuiles à afficher dans une petite zone de texte sur la carte. Ce champ est visible uniquement si *Fournisseur de tuiles* est défini sur *Autre*.
Niveau de zoom maximal	Niveau de zoom maximal de la carte. Ce champ est modifiable uniquement si *Fournisseur de tuiles* est défini sur *Autre*.

Utilisation d'un fournisseur de services de tuiles personnalisé

Le widget Geomap est capable de charger des images de tuiles à partir d'un service personnalisé auto-hébergé ou d'un fournisseur de tuiles tiers. Pour utiliser un service de fournisseur de tuiles tiers personnalisé ou un dossier ou un serveur de tuiles auto-hébergé, sélectionnez *Autre* dans le champ *Fournisseur de tuiles* et spécifiez l'URL personnalisée dans le champ *URL de tuile* en utilisant les espaces réservés appropriés.

Modules

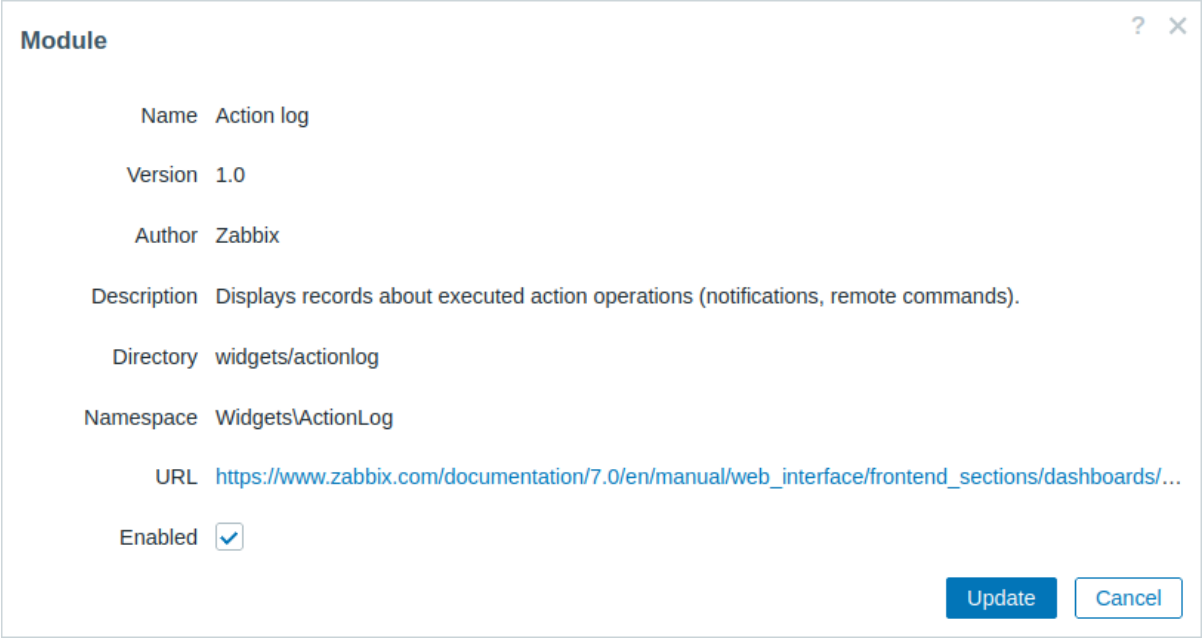
Cette section permet d'administrer les modules frontend personnalisés ainsi que ceux intégrés.



Cliquez sur *Scan directory* pour enregistrer/désenregistrer les modules personnalisés. Les modules enregistrés apparaîtront dans la liste ; les modules désenregistrés seront supprimés de la liste.

Cliquez sur le statut du module dans la liste pour activer/désactiver un module. Vous pouvez également activer/désactiver plusieurs modules à la fois en les sélectionnant dans la liste, puis en cliquant sur les boutons *Enable/Disable* sous la liste.

Cliquez sur le nom du module dans la liste pour afficher ses détails dans une fenêtre contextuelle.

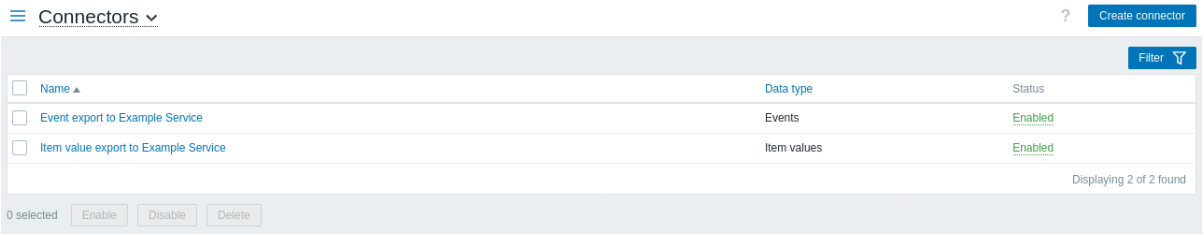


Le statut du module peut également être mis à jour dans la fenêtre contextuelle des détails du module ; pour ce faire, cochez/décochez la case *Enabled*, puis cliquez sur *Update*.

Vous pouvez filtrer les modules par nom ou par statut (activé/désactivé).

Connecteurs

Cette section permet de configurer des connecteurs pour le streaming des données Zabbix vers des systèmes externes via HTTP.



Cliquez sur *Create connector* pour configurer un nouveau connecteur.

Vous pouvez filtrer les connecteurs par nom ou par statut (activé/désactivé). Cliquez sur le statut 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 *Enable/Disable* sous la liste.

Autres

Cette section permet de configurer divers autres paramètres du frontend.



Paramètre	Description
URL du frontend	URL (jusqu'à 2048 caractères) de l'interface web Zabbix. Ce paramètre est utilisé par le service web Zabbix pour communiquer avec le frontend et doit être spécifié pour activer les rapports planifiés.
Groupe pour les hôtes découverts	Les hôtes découverts par la découverte réseau et l'auto-enregistrement des agents 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 le frontend, sauf s'il est remplacé lors de la découverte/auto-enregistrement 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 ou « Aucun ». 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

Parameter	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 : 0 - 3600 secondes. Les suffixes de temps, par ex. 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	Résolvez les macros du coffre de secrets par : **Zabbix server** - les secrets sont récupérés par Zabbix server et transmis aux proxies si nécessaire (par défaut) ; **Zabbix server and proxy** - les secrets sont récupérés à la fois par Zabbix server et les proxies, ce qui leur permet de résoudre les macros de manière indépendante.

Voir aussi : Stockage des secrets.

Sécurité

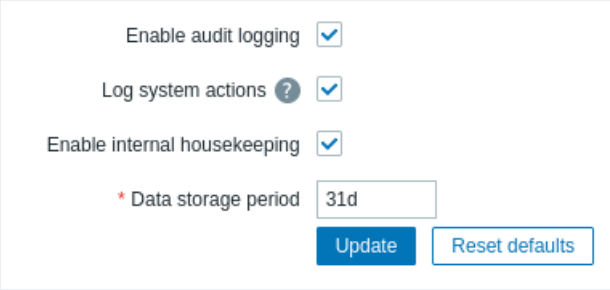
Paramètre	Description
Valider les schémas d'URI	Décochez cette case pour désactiver la validation des schémas d'URI (activée par défaut). Si elle est cochée, vous pouvez spécifier une liste de schémas d'URI autorisés, séparés par des virgules (par défaut : http,https,ftp,file,mailto,tel,ssh). Cela s'applique 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`.

2 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.

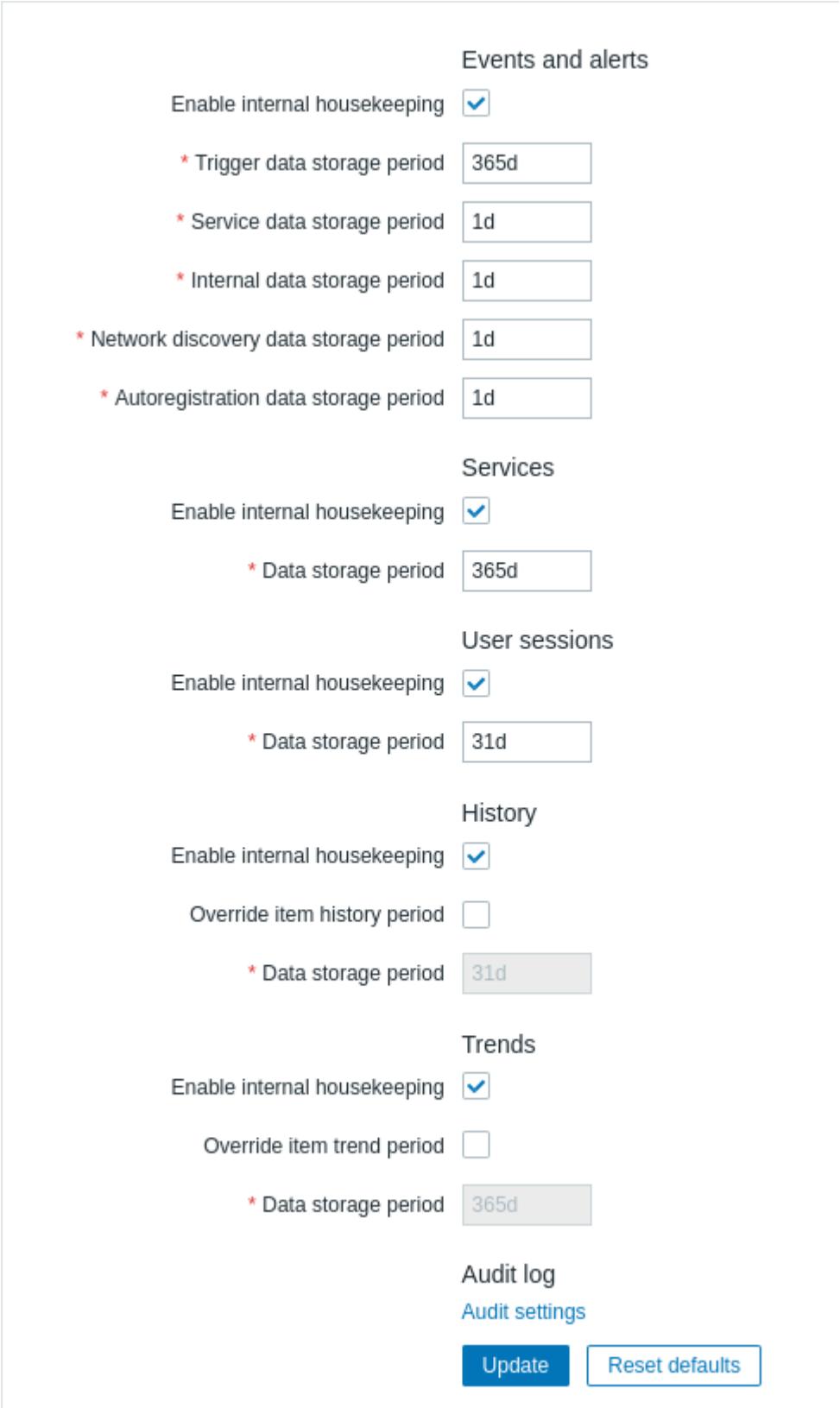
Les paramètres de compression du journal d'audit sont disponibles dans la section *Administration* > *Housekeeping*, dans le bloc *Compression de l'historique, des tendances et du journal d'audit*, qui devient visible lorsque TimescaleDB est utilisé.

3 Nettoyage

Aperçu

La section *Administration* > *Housekeeping* vous permet de définir à quel moment les données obsolètes ou supprimées par l'utilisateur sont retirées pour :

- Événements et alertes
- Services
- Sessions utilisateur
- Historique
- Tendances



Le nettoyage des audits est configuré séparément.

Configuration

Le tableau ci-dessous décrit les paramètres de configuration du nettoyage automatique.

Paramètre	Description
Activer le nettoyage automatique interne	Active (par défaut) ou désactive le nettoyage automatique interne effectué par le serveur Zabbix processus housekeeper.
Période de conservation des données	Spécifiez combien de temps les enregistrements doivent être conservés avant d'être supprimés par le housekeeper. Obligatoire si le nettoyage automatique interne est activé. Plage : 1 jour (1 heure pour l'historique) - 25 ans ; ou « 0 ». Les suffixes de temps (par ex., 1d, 1w) sont pris en charge. Pour *Événements et alertes*, la période de conservation des données est définie séparément pour les déclencheurs, les services, les données internes, la découverte réseau et l'enregistrement automatique. Le 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 le housekeeper supprime des entités obsolètes, il supprime d'abord les problèmes, puis les événements. Lorsqu'un déclencheur est supprimé, ses problèmes et les événements associés sont également supprimés. Pour *Historique* et *Tendances*, les périodes de conservation des données déterminent également combien de temps les données restent visibles dans *Surveillance > Dernières données*, même si le nettoyage automatique interne est désactivé.
Remplacer la période d'historique des éléments	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 *Période de conservation des données* (sauf pour les éléments pour lesquels l'option *Ne pas stocker* est activée). Cette option peut être utilisée même lorsque le nettoyage automatique interne est désactivé et qu'un housekeeper externe est utilisé.
Remplacer la période de tendances des éléments	

Si vous utilisez **TimescaleDB**, la section *Compression de l'historique, des tendances et du journal d'audit* devient disponible.

> **Note:**
> Pour TimescaleDB, activez *Remplacer la période d'historique des éléments*, *Remplacer la période de tendances des éléments* et *Activer le nettoyage automatique interne* pour l'historique et les tendances afin de tirer pleinement parti du partitionnement automatique. Si ces options sont désactivées, les données conservées dans les tables d'historique et de tendances seront toujours 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 web ne suivront plus les éléments supprimés, et l'historique de ces éléments sera effacé lorsqu'une partition obsolète sera supprimée.

Le bouton *Réinitialiser les valeurs par défaut* vous permet d'annuler toutes les modifications effectuées.

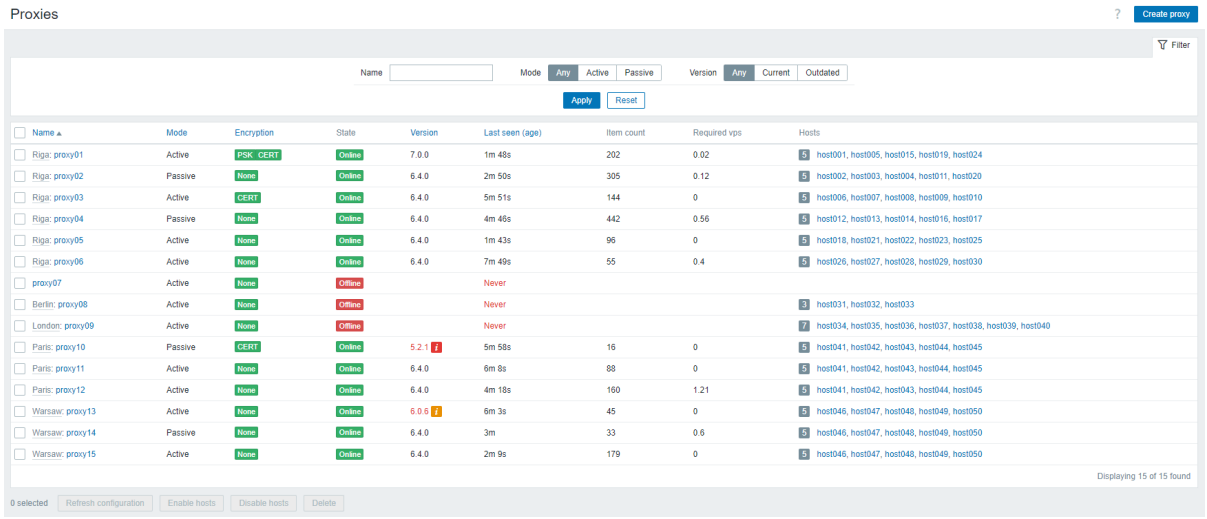
4 Proxys

Vue d'ensemble

Dans la section *Administration → Proxies*, les proxys pour la surveillance distribuée peuvent être configurés dans l'interface web de Zabbix.

Proxys

Une liste des proxys existants avec leurs détails est affichée.



Données affichées :

Column	Description
Name	Nom du proxy. Cliquer sur le nom du proxy ouvre le formulaire de configuration du proxy. Si le proxy appartient à un groupe de proxys, 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 proxys.
Mode	Mode du proxy - *Actif* ou *Passif*.
Encryption	État du chiffrement pour les connexions provenant du proxy : **Aucun** - pas de chiffrement ; **PSK** - utilisation d'une clé pré-partagée ; **Cert** - utilisation d'un certificat.
State	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 d'information (jaune ou rouge) est affichée. Survolez l'icône pour plus de détails.
Last seen (age)	Heure à laquelle le proxy a été vu pour la dernière fois par le serveur.
Item count	Nombre d'éléments activés sur les hôtes activés affectés au proxy.
Required vps	Performance requise du proxy (le nombre de valeurs devant être collectées par seconde).
Hosts	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 *Create proxy* dans le coin supérieur droit.

Options de modification en masse

Les boutons situés sous la liste offrent plusieurs options de modification en masse :

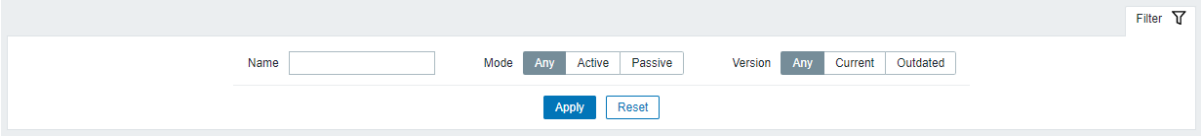
- *Actualiser la configuration* - actualiser la configuration des proxies ;
- *Activer les hôtes* - modifier le statut des hôtes surveillés par le proxy en *Surveillé* ;
- *Désactiver les hôtes* - modifier le statut des hôtes surveillés par le proxy en *Non surveillé* ;
- *Supprimer* - supprimer les proxies.

Pour utiliser ces options, cochez les cases devant 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, dans lequel vous pouvez 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 non pris en charge.



5 Groupes de proxys

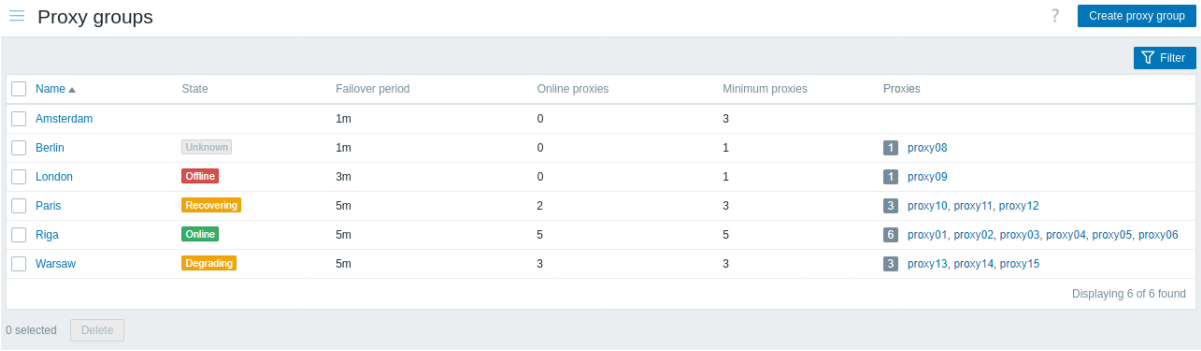
Aperçu

Dans *Administration* → *Groupes de proxys*, il est possible de configurer des groupes de proxys.

Les groupes de proxys sont utilisés dans l'équilibrage de charge des proxys avec une distribution automatisée des hôtes entre les proxys et une haute disponibilité entre les proxys.

Groupes de proxys

Une liste des groupes de proxys existants avec leurs détails est affichée.



Données affichées :

Colonne	Description
Nom	Nom du groupe de proxys. En cliquant sur le nom du groupe de proxys, vous ouvrez le formulaire de configuration du groupe de proxys.
État	État du groupe de proxys (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 proxys a communiqué avec le serveur Zabbix pendant la période de basculement ; **Dégradation** - le groupe va bientôt passer hors ligne car le nombre de proxys en ligne est tombé en dessous du seuil *Minimum proxies* ; **Hors ligne** - moins que le nombre minimal de proxys a communiqué avec le serveur Zabbix pendant la période de basculement ; **Récupération** - le groupe va bientôt revenir en ligne car le nombre de proxys en ligne a atteint le seuil *Minimum proxies*. L'état du groupe de proxys 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 proxys doit communiquer avec le serveur Zabbix pour être considéré comme en ligne.
Proxys en ligne	Nombre de proxys en ligne (affiché en rouge s'il est inférieur au minimum du groupe).
Minimum proxies	Nombre minimal de proxys en ligne requis pour maintenir le groupe de proxys en ligne.
Proxys	Nombre de proxys dans le groupe et liste des proxys du groupe, avec des liens vers le formulaire de configuration du proxy. Le nombre maximal de proxys listés est limité par la valeur *Max count of elements to show inside table cell*.

Pour configurer un nouveau groupe de proxys, cliquez sur le bouton *Create proxy groups* dans le coin supérieur droit.

Options de modification en masse

Les boutons situés sous la liste offrent certaines options de modification en masse :

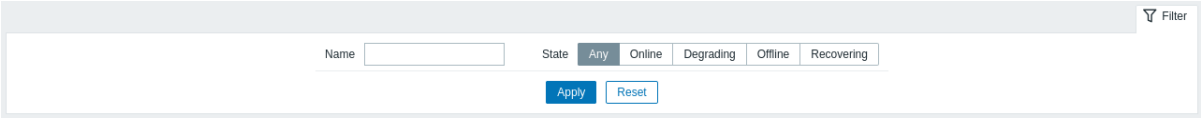
- *Supprimer* - supprimer les groupes de proxys.

Pour utiliser ces options, cochez les cases devant les groupes de proxys respectifs, puis cliquez sur le bouton requis.

Utilisation du filtre

Vous pouvez utiliser le filtre pour afficher uniquement les groupes de proxys qui vous intéressent. Pour de meilleures performances de recherche, les données sont recherchées avec les macros non résolues.

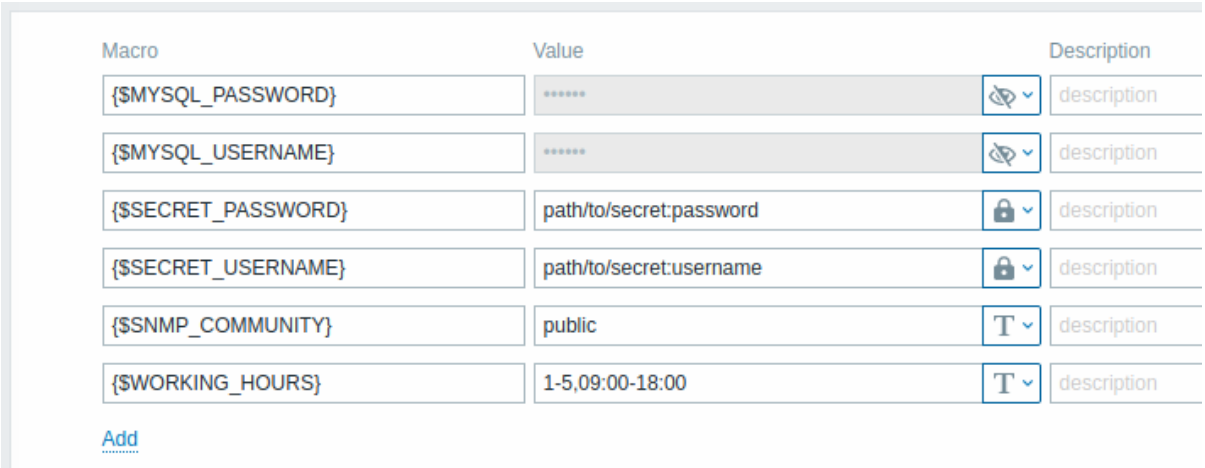
Le lien *Filtre* est disponible au-dessus de la liste des groupes de proxys. Si vous cliquez dessus, un filtre devient disponible, dans lequel vous pouvez filtrer les groupes de proxys par nom et par statut.



6 Macros

Aperçu

Cette section permet de définir des macros utilisateur à l'échelle du système sous forme de paires nom-valeur. Notez que les valeurs des macros peuvent être conservées en texte brut, texte secret ou secret Vault. L'ajout d'une description est également pris en charge.



7 File d'attente

Aperçu

Dans la section *Administration* → *Queue*, 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 n'est en attente. Cependant, en raison de performances insuffisantes du serveur, certains éléments peuvent être retardés et ces informations sont affichées dans cette section. Pour plus de détails, consultez la section Queue.

Note:
Le proxy Zabbix n'a pas connaissance des périodes de maintenance ; voir Calculation of queues during maintenance pour plus de détails.

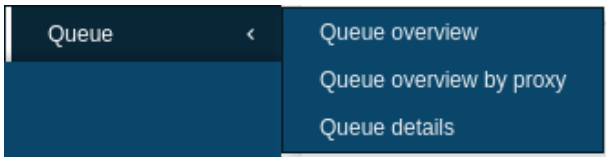
Note:
La file d'attente est disponible uniquement si le serveur Zabbix est en cours d'exécution. Les éléments ne sont pas comptabilisés dans la file d'attente si l'interface de l'élément devient indisponible en raison de problèmes de connexion ou si l'agent ne fonctionne pas correctement.

La section *Administration* → *Queue* contient les pages suivantes :

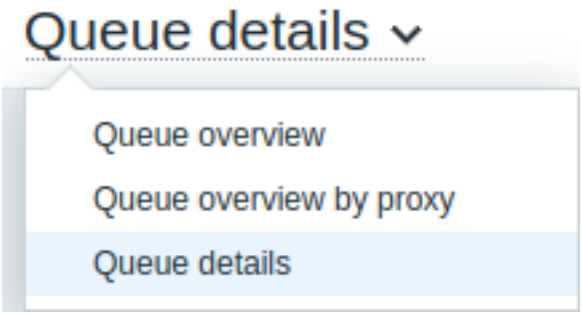
- Aperçu de la file d'attente — affiche la file d'attente par type d'élément ;
- Aperçu de la file d'attente par proxy — affiche la file d'attente par proxy ;

- Détails de la file d'attente — affiche une liste des éléments retardés.

La liste des pages disponibles apparaît lorsque vous cliquez sur *Queue* dans la section *Administration* du menu. Il est également possible de passer d'une page à l'autre à l'aide de la liste déroulante du titre dans le coin supérieur gauche.



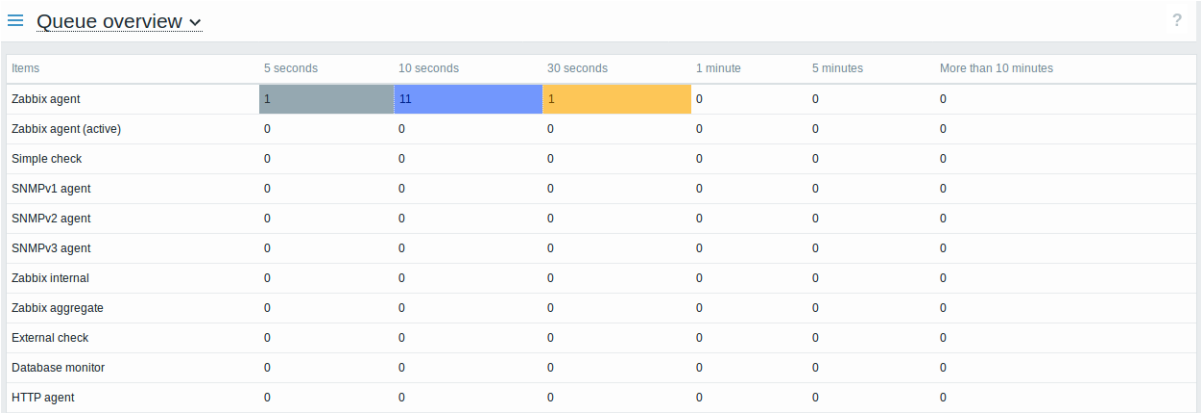
Menu de troisième niveau.



Liste déroulante du titre.

Aperçu par type d'élément

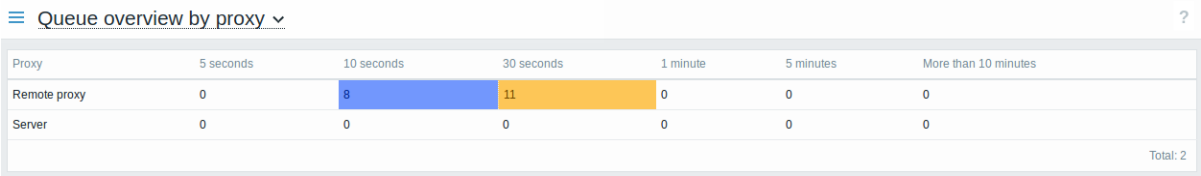
Dans cet écran, il est facile de localiser si le problème est lié à un ou plusieurs types d'éléments.



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

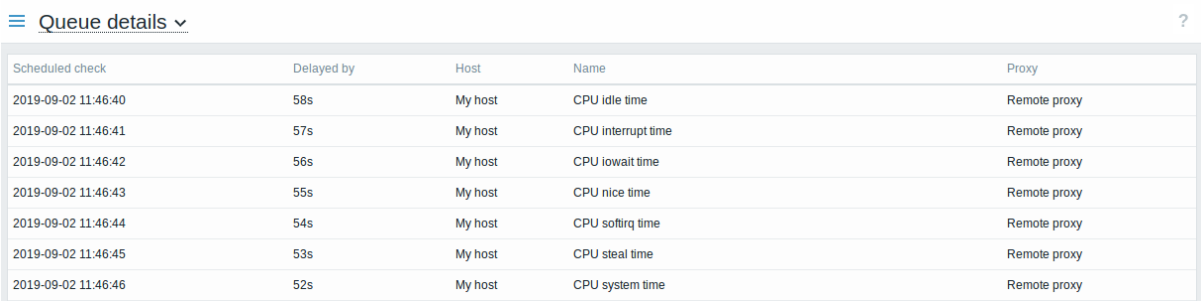
Dans cet écran, il est facile de localiser si le problème est lié à l'un des proxies ou au serveur.



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 soit affichée et que le message d'erreur suivant apparaisse :



Le message d'erreur dans ce cas est le suivant :

```
Cannot display item queue. Permission denied
```

Cela se produit lorsque les paramètres de configuration PHP dans le fichier *zabbix.conf.php* — `$ZBX_SERVER` ou à la fois `$ZBX_SERVER` et `$ZBX_SERVER_PORT` — pointent vers un serveur Zabbix existant qui utilise une base de données différente.

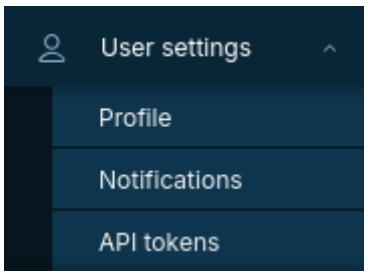
3 Paramètres utilisateur

Aperçu

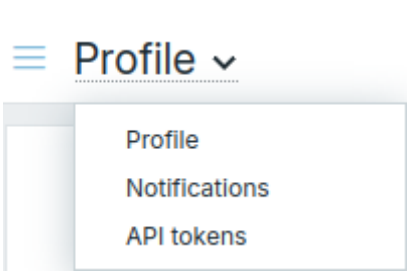
Selon les autorisations du rôle utilisateur, la section *Paramètres utilisateur* peut contenir les pages suivantes :

- *Profil* ou *Profil utilisateur* - pour personnaliser certaines fonctionnalités de l'interface web Zabbix ;
- *Notifications* - pour personnaliser les notifications de l'utilisateur actuel ;
- *Jetons d'API* - pour gérer les jetons d'API attribués à l'utilisateur actuel.

La liste des pages disponibles s'affiche après avoir cliqué sur l'icône utilisateur près du bas du menu Zabbix (non disponible pour l'utilisateur *guest*). Il est également possible de passer d'une page à l'autre en utilisant la liste déroulante du titre dans le coin supérieur gauche.



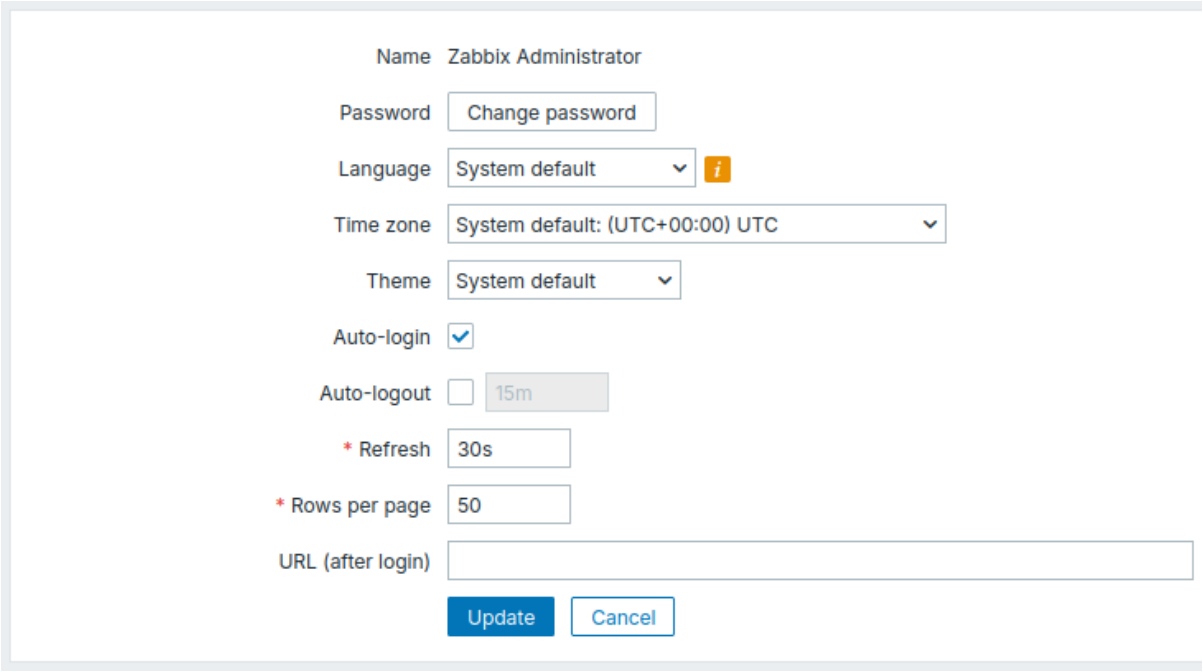
Menu de troisième niveau.



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.



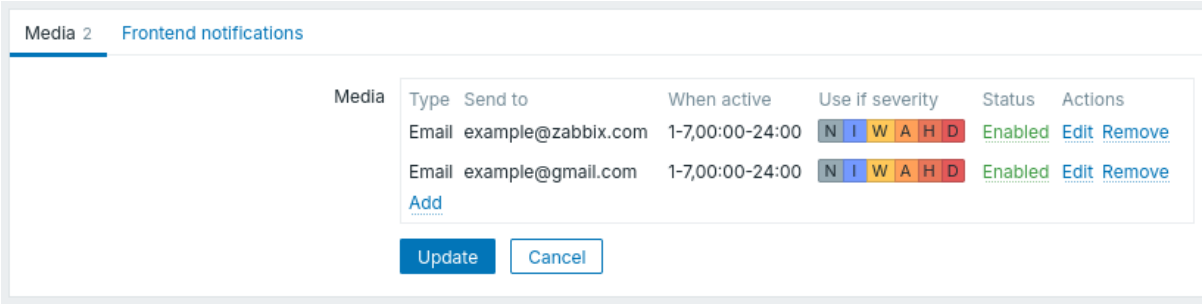
Parameter	Description
Name	Le nom et le prénom de l'utilisateur sont affichés. En l'absence de nom et de prénom, le nom d'utilisateur est affiché.
Password	Cliquez sur le bouton *Change password* pour ouvrir trois champs : *Old password*, *New password*, *New password (once again)*. Si le changement de mot de passe réussit, l'utilisateur sera déconnecté de toutes les sessions actives. Notez que le mot de passe ne peut être modifié que pour les utilisateurs utilisant l'authentification interne de Zabbix.
Language	Sélectionnez la langue d'interface de votre choix ou **System default** pour utiliser les paramètres système par défaut. La sélection de *English (en_US)* activera également le format de date/heure américain dans l'interface web. Pour plus d'informations, voir Installation de langues supplémentaires pour l'interface web.
Time zone	Sélectionnez le fuseau horaire afin de remplacer le fuseau horaire global au niveau de l'utilisateur, ou sélectionnez **System default** pour utiliser les paramètres globaux de fuseau horaire.
Theme	Sélectionnez un thème de couleurs spécifique pour votre profil : **System default** - utiliser les paramètres système par défaut ; **Blue** - thème bleu standard ; **Blue (classic)** - thème bleu standard utilisant les polices héritées ; **Dark** - thème sombre alternatif ; **Dark (classic)** - thème sombre alternatif utilisant les polices héritées ; **High-contrast light** - thème clair à contraste élevé ; **High-contrast dark** - thème sombre à contraste élevé.
Auto-login	Cochez cette case pour que Zabbix se souvienne de vous et vous connecte automatiquement pendant 30 jours. Lors d'une connexion avec l'option *Remember for 30 days* activée : - Le minuteur de déconnexion automatique est réinitialisé (la session reste active jusqu'à la fin de la période de 30 jours). - La connexion automatique est activée lors des visites suivantes dans les 30 jours. Lors d'une connexion sans *Remember for 30 days* : - Le paramètre de connexion automatique est effacé. - La déconnexion automatique suit le délai configuré comme d'habitude. Des cookies du navigateur sont utilisés à cette fin.

Parameter	Description
Auto-logout	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 *Monitoring* effectuent des actualisations d'informations en arrière-plan. Si des pages qui actualisent les données à un intervalle de temps spécifique (tableaux de bord, graphiques, dernières données, etc.) restent ouvertes, la durée de vie de la session est prolongée, ce qui désactive la fonction de déconnexion automatique. * En cas de connexion avec l'option *Remember me for 30 days* cochée ; dans ce cas, le minuteur de déconnexion automatique est ignoré, ce qui maintient votre session active pendant toute la période. * Lorsque l'authentification est effectuée via un fournisseur d'identité SSO (IdP), la déconnexion automatique met fin uniquement à la session Zabbix. La session SSO avec l'IdP reste active, de sorte que l'utilisateur peut être en mesure de se reconnecter sans saisir de mot de passe. *Auto-logout* peut également accepter la valeur "0", ce qui signifie que la fonction de déconnexion automatique sera désactivée après la mise à jour des paramètres du profil.
Refresh	Définissez la fréquence d'actualisation des informations sur les pages du menu *Monitoring* (minimum 0 seconde, maximum 1 heure). Les suffixes de temps sont pris en charge, par exemple : 30s, 90s, 1m, 1h.
Rows per page	Définissez le nombre de lignes affichées par page dans les listes. Un nombre plus faible de lignes (et donc moins d'enregistrements à afficher) permet des temps de chargement plus rapides.
URL (after login)	Définissez une URL spécifique à afficher après la connexion. Au lieu du *Dashboards* par défaut, il peut s'agir, par exemple, de l'URL de *Monitoring* > *Triggers*.

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*)

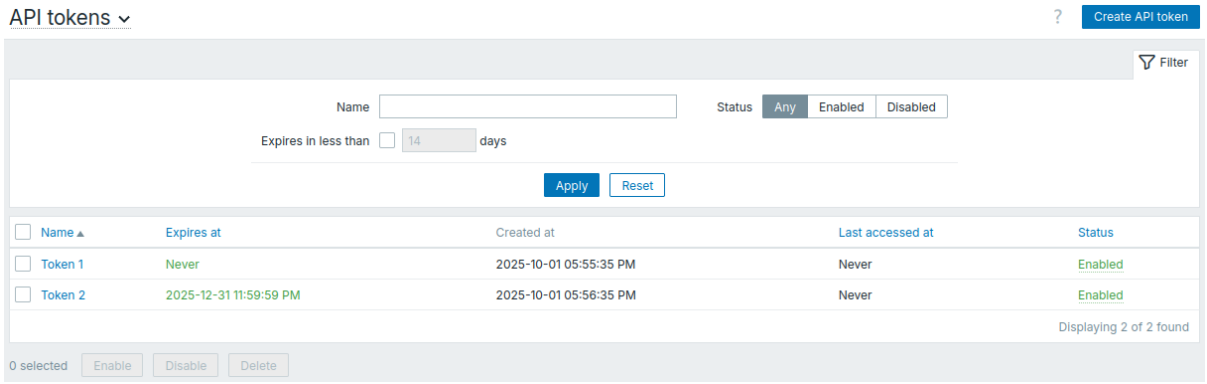
Notez que pour les utilisateurs provisionnés :

- 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 d'API

La section *Jetons d'API* vous permet d'afficher 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 d'API* est autorisée dans les paramètres du rôle utilisateur.



Vous pouvez filtrer les jetons d'API par nom, date d'expiration ou statut (*Activé/Désactivé*). Cliquez sur le statut du jeton dans la liste pour l'activer/le désactiver rapidement. 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 voir 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.

1 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.

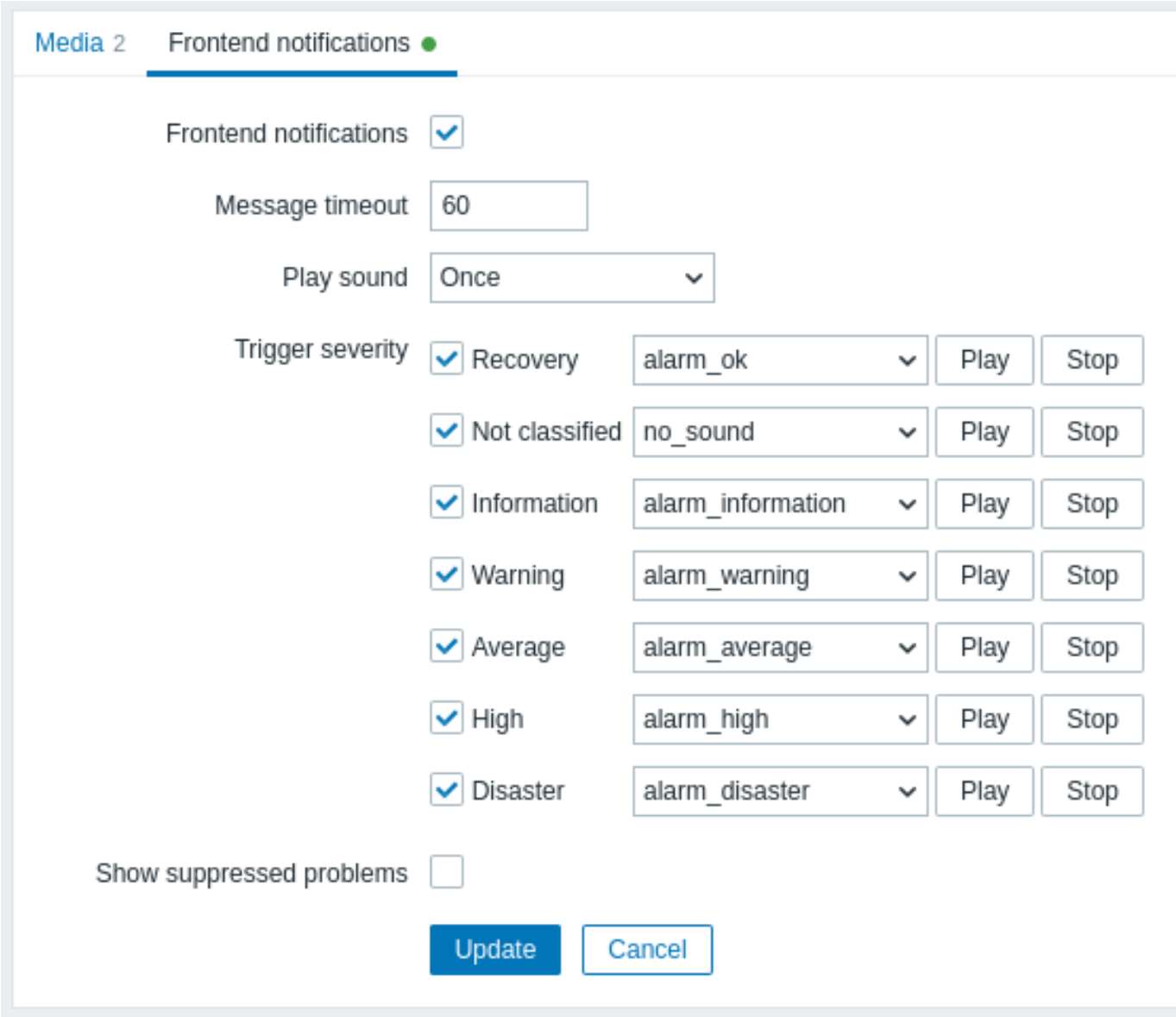
Sans notifications globales, lorsque vous travaillez en dehors des sections *Problèmes* ou *Tableau de bord*, vous ne recevriez aucune information sur les problèmes en cours. Les notifications globales garantissent que ces informations sont affichées, quel que soit l'endroit où vous vous trouvez dans 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

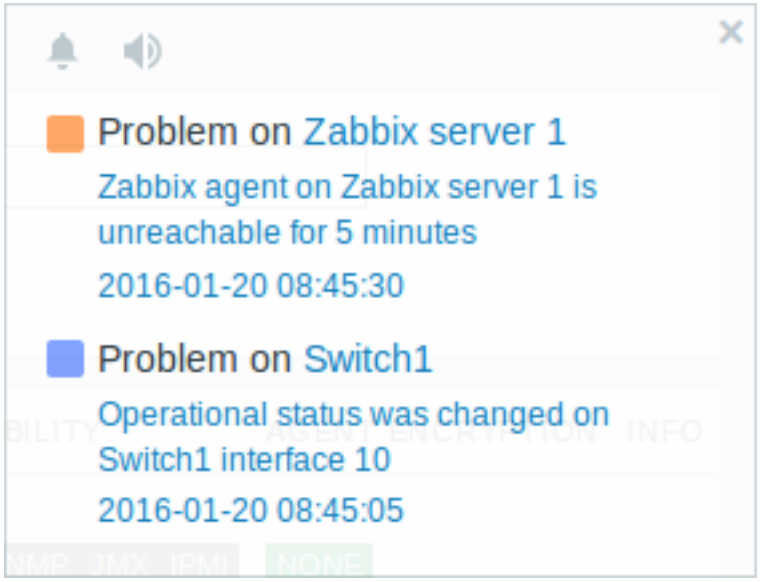
Les notifications globales peuvent être activées par utilisateur dans l'onglet **Frontend notifications** de la section Notifications.



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.

2 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.

4 Recherche globale

Il est possible d'effectuer une recherche dans l'interface web de Zabbix pour les hôtes, les groupes d'hôtes, les modèles et les groupes de modèles.

Le champ 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 à n'importe quel endroit, une liste déroulante apparaîtra, répertoriant tous ces hôtes (avec la partie correspondante mise en évidence en orange). La liste déroulante répertoriera également un hôte si son nom visible correspond au nom technique saisi comme chaîne de recherche ; l'hôte correspondant sera affiché, mais sans mise en évidence.

Attributs recherchables

Les hôtes peuvent être recherchés selon les propriétés suivantes :

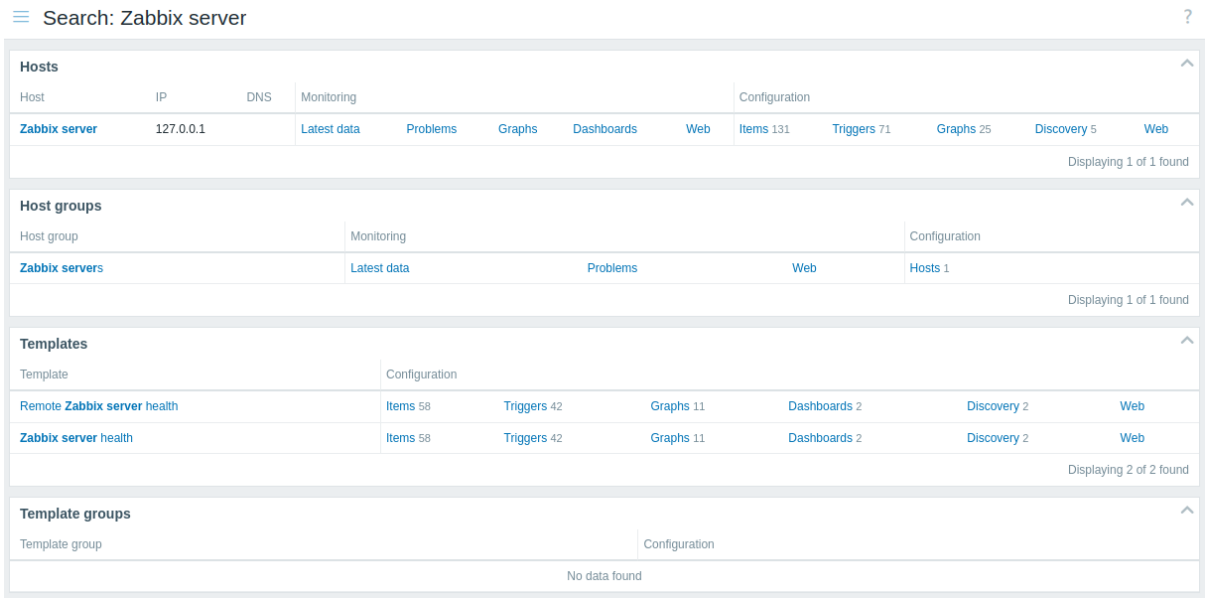
- Nom de l'hôte
- Nom visible
- Adresse IP
- Nom DNS

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 chiffre à côté du nom de l'entité, en gris. **Remarque** : si le nombre d'entités est nul, aucun chiffre 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

5 Mode maintenance de l'interface Web

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 des 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 au frontend (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 le répertoire `/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 :

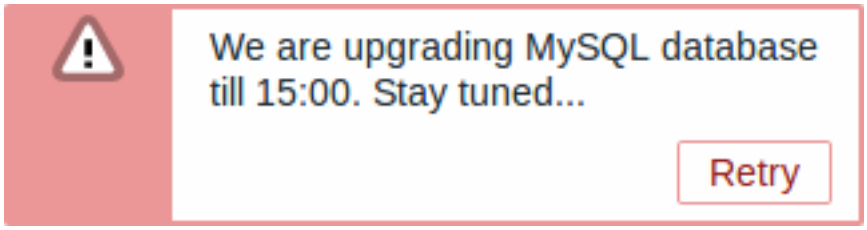
- SUSE et RedHat est `/etc/zabbix/web/maintenance.inc.php`.
- les systèmes basés sur Debian est `/usr/share/zabbix/conf/`.

Voir aussi Copie des fichiers PHP.

Paramètre	Détails
ZBX_DENY_GUI_ACCESS	Si défini avec n'importe quelle valeur, le mode maintenance sera activé. Pour désactiver le mode maintenance, commentez ou supprimez-le.
ZBX_GUI_ACCESS_IP_RANGE	Tableau d'adresses IP autorisées à se connecter à l'interface web (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 is under 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.

6 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 :

Paramètre	Description	Exemple
show	Option de filtre *Afficher*. Valeurs possibles : 1 - problèmes récents ; 2 - tous ; 3 - à l'état de problème.	show=1
name	Option de filtre *Problème* : chaîne libre.	name=Zabbix agent

Paramètre	Description	Exemple
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 - moyenne ; 4 - élevée ; 5 - désastre.	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. 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][ope
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 la priorité 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 *Statut d'acquittement*. Valeurs possibles : 0 - tous ; 1 - non acquittés ; 2 - acquittés.	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

Paramètre	Description	Exemple
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 : options de filtre de la page Problèmes.

Mode kiosque

Le mode kiosque dans les pages du frontend 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=1` - activer le diaporama

7 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 pouvant présenter un intérêt pour 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 web Zabbix.

Par défaut : zbx_sessionid

- ZBX_DATA_CACHE_TTL

Délai TTL en secondes utilisé pour invalider le cache de données de la réponse Vault. Définissez 0 pour désactiver la mise en cache de la réponse 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

8 Créer votre propre thème

Aperçu

Par défaut, Zabbix fournit un certain nombre de thèmes prédéfinis. Vous pouvez suivre la procédure étape par étape fournie ici afin de créer la vôtre. N'hésitez pas à partager le résultat de votre travail avec la communauté Zabbix si vous avez créé quelque chose de sympa.

Étape 1

Pour définir votre propre thème, vous devrez créer un fichier CSS et l'enregistrer dans le dossier `assets/styles/` (par exemple, *custom-theme.css*). Vous pouvez soit copier les fichiers d'un thème différent et créer votre thème en fonction de celui-ci, soit recommencer à zéro.

Étape 2

Ajoutez votre thème à la liste des thèmes renvoyés par la méthode APP::getThemes(). Vous pouvez le faire en remplaçant la méthode ZBase::getThemes() dans la classe APP. Cela peut être fait en ajoutant le code suivant avant l'accolade fermante dans *include/classes/core/APP.php* :

```
public static function getThemes() {
    return array_merge(parent::getThemes(), [
        'custom-theme' => _('Custom theme')
    ]);
}
```

Attention:
Notez que le nom que vous spécifiez dans la première paire de guillemets doit correspondre au nom du fichier de thème sans extension.

Pour ajouter plusieurs thèmes, listez-les simplement sous le premier thème, par exemple :

```
public static function getThemes() {
    return array_merge(parent::getThemes(), [
        'custom-theme' => _('Custom theme'),
        'anothertheme' => _('Another theme'),
        'onemoretheme' => _('One more theme')
    ]);
}
```

Notez que chaque thème, sauf le dernier, doit avoir une virgule en fin de ligne.

Note:
Pour changer les couleurs du graphique, l'entrée doit être ajoutée dans la table de la base de données *graph_theme*.

Étape 3

Activez le nouveau thème.

Dans l'interface Zabbix, vous pouvez soit définir ce thème comme étant celui par défaut, soit modifier votre thème dans le profil utilisateur.

Profitez du nouveau thème !

9 Mode Debug

Aperçu

Le mode Debug peut être utilisé pour diagnostiquer les problèmes de performance de l'interface web.

Configuration

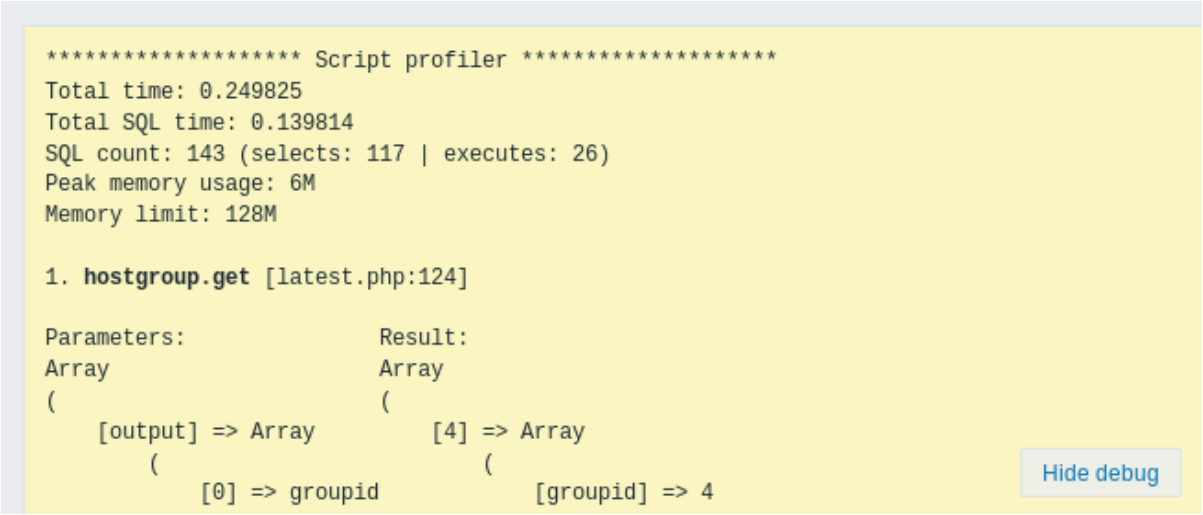
Le mode de débogage peut être activé pour des utilisateurs individuels appartenant à un groupe d'utilisateurs :

- 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 :



Un clic sur le bouton *Debug* ouvre une nouvelle fenêtre sous le contenu de la page, qui 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 première du problème.

Warning:
L'activation du *mode de débogage* affecte négativement les performances du frontend.

10 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

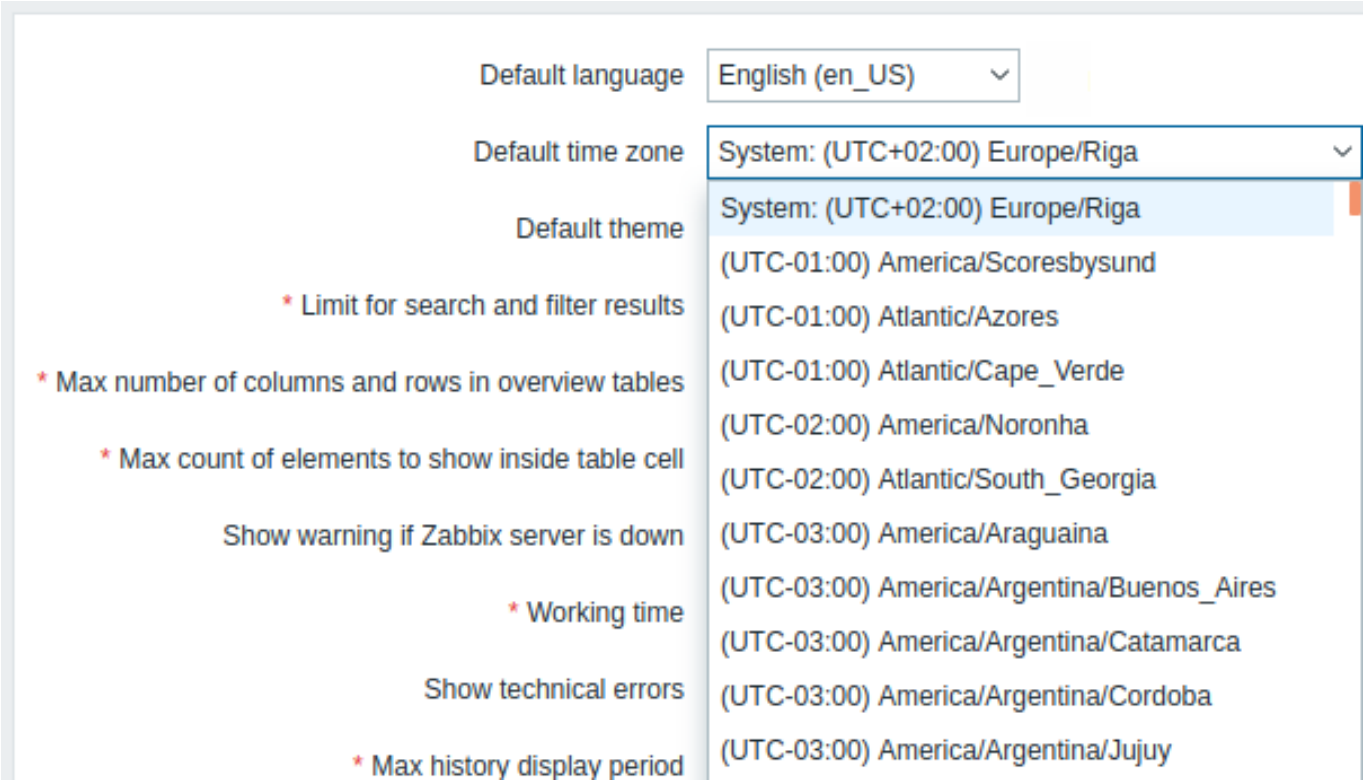
[1] Lorsque `HttpOnly` vaut 'true', le cookie sera accessible uniquement via le protocole HTTP. Cela signifie que le cookie ne sera pas accessible par des langages de script, tels que JavaScript. Ce paramètre peut contribuer efficacement à réduire le vol d'identité via des attaques XSS (bien qu'il ne 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.

11 Fuseaux horaires

Aperçu

Le fuseau horaire du frontend peut être défini globalement dans le frontend et ajusté pour chaque utilisateur.



Si *System* est sélectionné, le fuseau horaire du serveur web sera utilisé pour le frontend (y compris la valeur de « date.timezone » de 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 n'utilisera le fuseau horaire global/utilisateur spécifié que lors du développement des macros dans les notifications (par ex. {EVENT.TIME} peut être développé dans un fuseau horaire différent selon l'utilisateur) et pour la limite de temps lors de l'envoi des notifications (voir le paramètre « When active » dans la configuration des médias de l'utilisateur).

Le choix du fuseau horaire n'affecte pas le format de date/heure du frontend. À la place, vous pouvez ajuster la langue de l'interface (soit lors de l'installation, soit dans les paramètres utilisateur) — sélectionner *English (en_US)* activera également le format de date/heure américain dans le frontend.

Configuration

Le fuseau horaire global :

- peut être défini manuellement lors de l'installation du frontend
- peut être modifié dans *Administration* → *Général* → *GUI*

Fuseau horaire au niveau de l'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.

12 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 disposant du rôle 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.

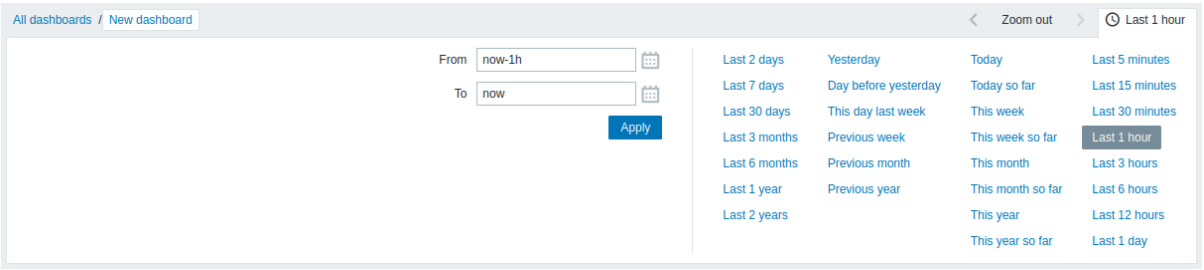
13 Sélecteurs de période et d'hôte

Vue d'ensemble

Cette page décrit les sélecteurs de période et d'hôte disponibles sur les tableaux de bord. Ces sélecteurs permettent d'ajuster dynamiquement les données affichées sur tous les widgets de tableau de bord compatibles, sans modifier les paramètres de chaque widget individuellement.

Sélecteur de période

Le sélecteur de période permet de sélectionner des périodes fréquemment requises en un seul clic de souris. Il peut être développé ou réduit en cliquant sur le bouton du sélecteur de période dans le coin supérieur droit du tableau de bord.



Des options telles que *Aujourd'hui*, *Cette semaine*, etc., affichent toute la période, y compris les heures/jours à venir. Des options telles que *Aujourd'hui jusqu'à présent*, *Cette semaine jusqu'à présent*, etc., affichent uniquement les heures déjà écoulées.

Une fois une période sélectionnée, elle peut être déplacée vers l'arrière ou vers l'avant dans le temps en cliquant sur les boutons fléchés . Le bouton *Zoom arrière* permet de réduire le zoom de la période de 50 % dans chaque direction.

> **Note:**
> Pour les graphiques (à l'exception de ceux avec la période *Personnalisée*), il est également possible de sélectionner la période affichée en mettant en surbrillance une zone du graphique avec le bouton gauche de la souris. Une fois le bouton gauche de la souris relâché, le graphique effectuera un zoom sur la zone mise en surbrillance. Il est également possible de faire un zoom arrière en double-cliquant dans le graphique.

Les champs *De/À* affichent la période sélectionnée soit en syntaxe de temps absolu (au format `Y-m-d H:i:s`), soit en syntaxe de temps relatif. Une période relative peut contenir une ou plusieurs opérations mathématiques (- ou +), par exemple `now-1d` ou `now-1d-2h+5m`.

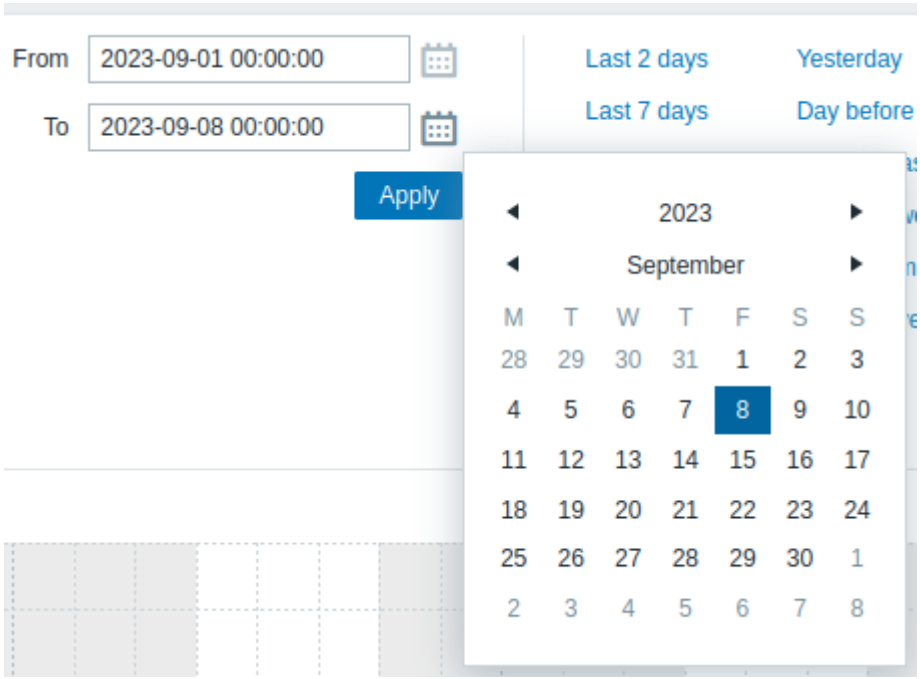
Les abréviations de temps relatif suivantes sont prises en charge :

- `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 :

Précision	De	À
`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
`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 à côté des champs *De/À*.



Note:
Dans le sélecteur de date, vous pouvez naviguer entre l'année/le mois/la date à l'aide de `Tab`, `Shift+Tab` et des touches fléchées du clavier. Appuyer sur `Enter` confirme 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 afin d'afficher ses données 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 *Override host* défini pour utiliser le tableau de bord comme source de données.



17 Bonnes pratiques

Vue d'ensemble Cette section présente les bonnes pratiques pour configurer Zabbix.

Bien que ces pratiques ne soient pas obligatoires pour le fonctionnement de Zabbix, leur mise en œuvre est fortement recommandée afin de garantir une utilisation optimale et sécurisée.

1 Bonnes pratiques de sécurité

Vue d'ensemble

Cette section contient les bonnes pratiques pour configurer Zabbix de manière sécurisée.

Les pratiques décrites dans cette section ne sont pas nécessaires au fonctionnement de Zabbix, mais sont recommandées pour une meilleure sécurité du système.

Encodage UTF-8

UTF-8 est le seul encodage pris en charge par Zabbix. Il est reconnu 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 au lieu de 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

Consultez *Avis de sécurité Zabbix et base de données CVE*.

HTML email templates

When creating or editing message templates used for HTML emails, always wrap every macro with the `htmlencode()` macro function. For example:

```
<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
```

Using `htmlencode()` ensures that any HTML characters in macro values are escaped and prevents injection of HTML into notifications (for example, an attacker inserting a malicious/phishing link into a notification).

Note: the default HTML email messages provided by Zabbix already apply `htmlencode()` to macros. This guidance applies when editing existing templates or creating new ones — verify that macros are encoded before using a template to send HTML email.

1 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, à tout moment, fonctionner avec le moins de privilèges possible. Cela signifie que les comptes utilisateur dans le frontend Zabbix, les utilisateurs de la base de données ou l'utilisateur des processus Zabbix server/proxy/agent ne doivent disposer que des privilèges strictement nécessaires 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'utilisateur du frontend de 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 toujours 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* soit disponible uniquement 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 des scripts les informations sensibles (par exemple, les identifiants d'accès) peut aider à éviter l'exposition d'informations sensibles présentes dans les scripts globaux.

Utilisateur sécurisé pour l'agent Zabbix

Par défaut, les processus du serveur Zabbix, du proxy et de l'agent (ou agent 2) partagent un même utilisateur `zabbix`. Pour empêcher l'agent Zabbix/agent 2 (s'exécutant 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 la base de données), l'agent doit être exécuté sous un utilisateur différent :

Pour l'agent Zabbix :

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 afin d'abandonner les privilèges au profit du nouvel utilisateur.

Pour l'agent Zabbix 2, la configuration doit être appliquée au niveau du service, car le fichier de configuration de l'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`), veillez à révoquer l'accès en écriture à ce répertoire pour les utilisateurs non administrateurs. Dans le cas contraire, l'agent charge les paramètres SSL depuis un chemin pouvant 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>,*] pour chaque commande autorisée dans la configuration de l'agent) ;
- l'authentification HTTP des utilisateurs peut être désactivée en définissant `$ALLOW_HTTP_AUTH=false` dans le fichier de configuration du frontend (zabbix.conf.php). Notez que la réinstallation du frontend (exécution de setup.php) supprimera ce paramètre.

Accès aux chemins UNC sous Windows par l'agent Zabbix

Les agents Zabbix sous Windows suivent les chemins UNC (partages SMB comme `\\server\share\file.txt`) dans des éléments tels que `vfs.file.*`, `vfs.dir.*`, `modbus.get` et `perf_counter*`. Cela peut représenter un risque de sécurité dans certains contextes.

Lorsque Windows est invité à accéder à un chemin UNC, il tente de s'authentifier sur ce serveur. Cela signifie qu'une requête malveillante adressée à l'agent Zabbix peut exposer le hachage NTLM au serveur du demandeur. Les utilisateurs peuvent atténuer ce risque à l'aide des paramètres de configuration AllowKey et DenyKey si nécessaire.

1 Sécurisation de MySQL/MariaDB

Vue d'ensemble

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 d'une 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 strictement nécessaires à l'exécution des fonctions prévues.

Création des rôles utilisateur

Créez les rôles suivants avec les privilèges correspondants :

- **zbx_srv** - rôle pour exécuter le serveur et le proxy Zabbix :

```
CREATE ROLE 'zbx_srv';
GRANT DELETE, INSERT, SELECT, UPDATE ON zabbix.* TO 'zbx_srv';
FLUSH PRIVILEGES;
```

- **zbx_web** - rôle pour exécuter le frontend et l'API Zabbix :

```
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 des tables et la mise à niveau 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';
-- Pour MariaDB : ignorez la ligne suivante (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 assignez-leur les rôles correspondants. Remplacez <user>, <host>, <role> et <password> selon vos besoins.

```
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 destiné à l'exécution du serveur et du proxy Zabbix :

```
CREATE USER 'usr_srv'@'localhost' IDENTIFIED BY 'password';
GRANT 'zbx_srv' TO 'usr_srv'@'localhost';
SET DEFAULT ROLE ALL TO 'usr_srv'@'localhost';
FLUSH PRIVILEGES;
```

2 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 modèle d'utilisation sécurisé des schémas, 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 d'en modifier la définition est un privilège inhérent au propriétaire de la base de données et ne peut être ni accordé ni révoqué.

> **Attention:**
> Les commandes suivantes de cette page doivent être exécutées lorsque la connexion à PostgreSQL est établie spécifiquement à 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 modèle d'utilisation sécurisé du schéma :

```
REVOKE CREATE ON SCHEMA public FROM PUBLIC;
REVOKE ALL ON DATABASE zabbix FROM PUBLIC;
-- Remarque : search_path doit pointer vers le schéma "zabbix" :
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 exécuter le serveur et le proxy Zabbix :

```
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 exécuter l'interface web et l'API Zabbix :

```
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 assignez-leur les rôles appropriés. Remplacez <user>, <role> et <password> selon les besoins.

```
CREATE USER <user> WITH ENCRYPTED password '<password>';
GRANT <role> TO <user>;
```

Par exemple, pour créer et attribuer le rôle destiné à l'exécution du serveur et du proxy Zabbix :

```
CREATE USER usr_srv WITH ENCRYPTED password '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 les privilèges du 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
```

2 Cryptographie

Vue d'ensemble

Cette section contient les meilleures pratiques pour configurer la cryptographie de manière sécurisée.

Configuration de SSL pour le frontend 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 -
```

Remplissez 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
```

3 Serveur web

Vue d'ensemble

Cette section contient les bonnes pratiques pour configurer le serveur web de manière sécurisée.

Activation de Zabbix à la racine de l'URL

Sur les systèmes basés sur RHEL, ajoutez un hôte virtuel à la configuration 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.

```
#### Ajoutez les lignes suivantes :

<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 le frontend 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 frontend Zabbix dans la configuration Apache, suivez ces étapes :

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 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'appliquer les attributs Secure et SameSite aux cookies de session afin de renforcer la sécurité et de prévenir les attaques de falsification de requête intersites (CSRF). Cependant, l'application de `SameSite=Strict` peut poser des problèmes dans certains scénarios, tels que :

- Les widgets d'URL du tableau de bord affichent « utilisateur non connecté » 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.
- L'impossibilité de partager des URL vers des sections spécifiques du menu Zabbix ou vers des hôtes.

Pour atténuer ces problèmes, les utilisateurs doivent disposer d'un moyen d'ajuster la politique SameSite.

1. Cookies Secure

Le paramètre `secure` garantit que les cookies ne sont transmis que via HTTPS, évitant ainsi leur exposition sur des connexions non chiffrées.

Pour activer les cookies Secure 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 l'interface web de Zabbix est accessible via HTTPS ; sinon, les cookies avec l'attribut `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; # Remplacez ~ par 'zbx_session' pour plus de précision
```

Activation de la 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 d'autres attaques similaires, nous recommandons d'activer la 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 remplacez *OpenStreetMap* par un autre moteur de carte, ou si vous ajoutez des widgets ou des feuilles de style CSS externes. Si vous utilisez la méthode d'authentification multifacteur Duo Universal Prompt, 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, suivez les étapes suivantes :

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 :

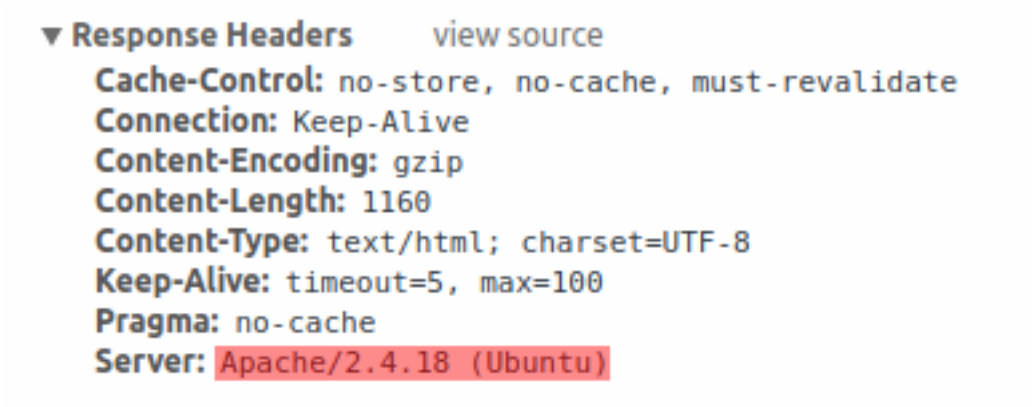
```
#### Sur les systèmes basés sur RHEL :
systemctl restart httpd.service

#### 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
```

Un redémarrage du serveur web est nécessaire pour que les modifications du fichier de configuration soient appliquées.

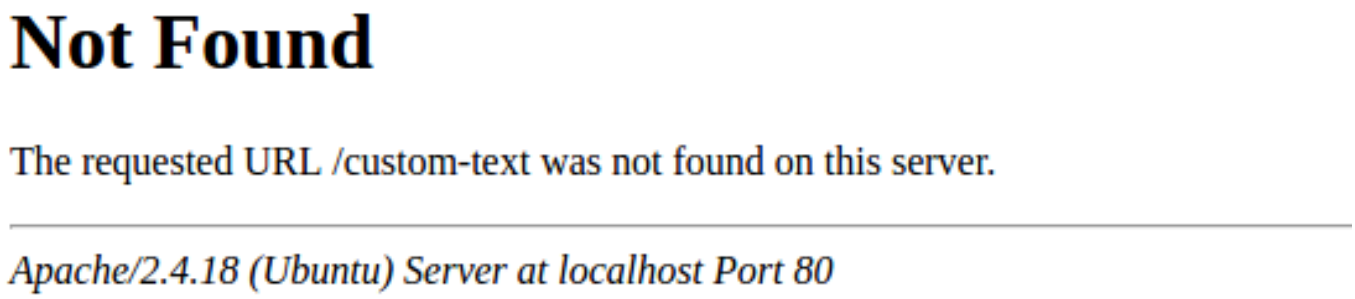
Pour une sécurité supplémentaire, vous pouvez utiliser l'outil *mod_security* avec Apache (paquet *libapache2-mod-security2*). Cet outil permet de supprimer la signature du serveur au lieu de supprimer uniquement la version de la signature du serveur. La signature du serveur peut être remplacée par n'importe 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/modifier les signatures logicielles.

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 :



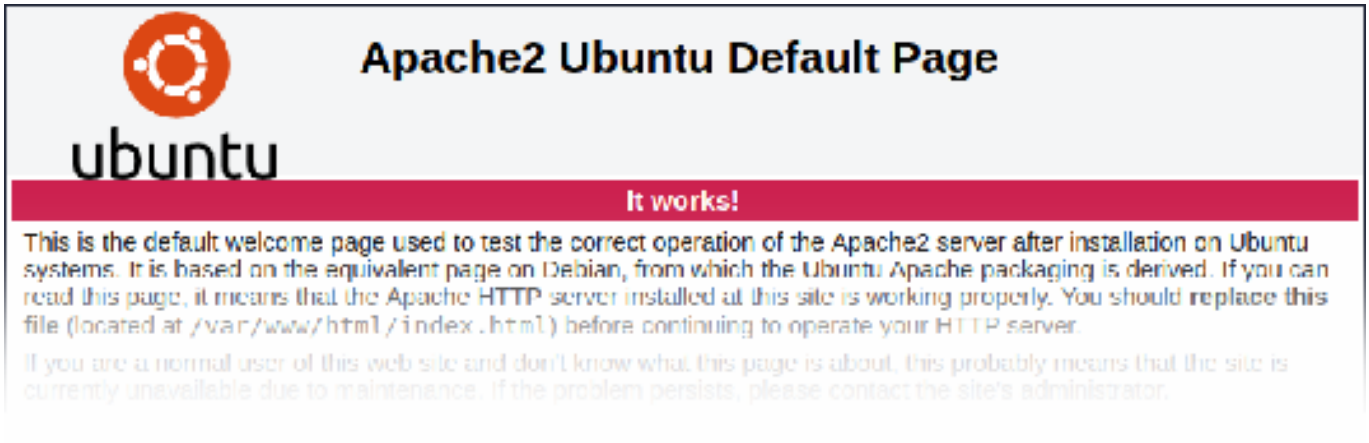
Ces pages d'erreur par défaut doivent être remplacées/supprimées. Par exemple, la directive « ErrorDocument » peut être utilisée pour définir une page/un texte d'erreur personnalisé pour le serveur web Apache.

Veuillez consulter la documentation de votre serveur web pour savoir comment remplacer/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 du frontend 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 bac à sable avec toutes les restrictions de sandboxing activées.

Ces paramètres renforcent la sécurité du frontend 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 attentivement 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.

Masquage du 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.

Sous 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>
```

Sous NGINX, l'accès au fichier peut être limité à l'aide de la directive `location` :

```
location = /data/top_passwords.txt {
    deny all;
    return 404;
}
```

2 Bonnes pratiques de configuration

Aperçu

Cette section présente les bonnes pratiques pour configurer Zabbix afin d'obtenir des performances optimales et une utilisation aisée.
Les recommandations sont fondées sur les conseils des développeurs de Zabbix ainsi que sur l'expérience pratique des formateurs et des ingénieurs du support Zabbix.

Chaque installation Zabbix est unique, et certaines de ces recommandations peuvent ne pas convenir à votre configuration spécifique.
Cependant, il est recommandé de suivre ces recommandations autant que possible afin d'éviter les problèmes potentiels courants.

Note:
Si vous pensez que cette page pourrait être améliorée, nous serions ravis de vous entendre ! Veuillez surligner le texte concerné et appuyer sur **ctrl+Enter** pour signaler une erreur ou partager vos commentaires.

Hôtes et éléments Définir un hôte

Dans Zabbix, un hôte n'est pas une machine ou un appareil physique, mais une entité logique. À des fins de supervision, vous pouvez créer des hôtes distincts pour une base de données ou, par exemple, une machine virtuelle. Vous pouvez également créer un hôte générique *ordinateur portable de John* et superviser toutes les métriques sous cet hôte.

La bonne pratique consiste à créer un hôte distinct pour chaque instance indépendante, telle qu'une machine virtuelle, une base de données, un conteneur ou un commutateur réseau. En adoptant cette approche, vous pourrez :

1. Éviter l'encombrement des données de supervision en disposant d'éléments, de déclencheurs et de notifications d'alerte distincts pour chaque hôte.
2. Affiner les niveaux d'accès des utilisateurs. Vous pouvez configurer les rôles utilisateur afin d'accorder l'accès à l'affichage et/ou à la configuration de certains hôtes uniquement. Voir aussi le principe du moindre privilège.

Hôtes avec des éléments dupliqués

Si vous avez plusieurs hôtes similaires, tels que *Commutateur réseau 1* et *Commutateur réseau 2*, Zabbix propose plusieurs moyens de recréer rapidement l'hôte. Vous pouvez simplement cloner un hôte avec toutes ses métriques en appuyant sur le bouton Clone, mais dans ce cas, pour mettre à jour un élément ultérieurement, vous devrez le faire manuellement sur chaque hôte.

La bonne pratique consiste à créer un modèle avec toutes les métriques requises, par exemple *Modèle de commutateur réseau*. Ensuite, regroupez les hôtes similaires dans un groupe d'hôtes ; pour l'exemple ci-dessus, il pourrait s'agir de *Commutateurs réseau*. À présent, dans la section *Data Collection -> Hosts*, vous pouvez filtrer tous les hôtes par groupe d'hôtes et utiliser le bouton *Mass update* pour lier le modèle à tous vos commutateurs réseau.

Éléments dépendants

Pour minimiser le nombre de requêtes vers l'entité cible, Zabbix permet de créer des éléments maîtres et des éléments dépendants. Dans ce cas, l'élément maître collecte un grand ensemble d'informations en une seule requête. Les éléments dépendants peuvent ensuite être configurés pour extraire des données spécifiques de cet ensemble via le prétraitement et les stocker en tant que métriques individuelles.

Par exemple, l'élément maître peut collecter une réponse JSON ou XML contenant plusieurs métriques, ou exécuter une requête de base de données qui renvoie plusieurs colonnes de données (par exemple, le nombre de connexions ouvertes, les connexions abandonnées, le nombre maximal de connexions simultanées autorisées et le nombre total cumulé de connexions depuis le démarrage), et les éléments dépendants analyseront et stockeront séparément chaque valeur requise.

La meilleure pratique pour cette configuration consiste à supprimer l'historique de l'élément maître immédiatement après la collecte et à ne conserver que les données des éléments dépendants.

Serveurs et proxies

Si tous les hôtes se trouvent sur le même réseau local que le serveur Zabbix et qu'il n'y a pas de problème de scalabilité ou de performance, vous n'avez peut-être pas besoin d'un proxy. Dans des environnements plus vastes ou plus complexes, la surveillance directe des hôtes par le serveur Zabbix peut ne pas être suffisante. L'ajout d'un proxy et l'affectation d'une partie des hôtes à ce proxy permettent une répartition plus équilibrée de la charge.

La bonne pratique consiste à ajouter un proxy Zabbix lorsque :

1. Vous surveillez plusieurs hôtes à l'aide de différentes méthodes de collecte de métriques derrière un pare-feu. Le proxy collectera les données des hôtes et les transmettra au serveur Zabbix, ce qui réduit la nécessité d'ouvrir plusieurs ports sur le pare-feu.
2. Vous surveillez des sites distants, des succursales et/ou des réseaux. En cas d'interruption réseau entre le serveur Zabbix et vos sites distants, les proxies Zabbix déployés sur ces sites continueront à collecter les données et renverront les données collectées au serveur Zabbix dès que la connexion réseau sera rétablie.
3. Vous disposez d'un déploiement à grande échelle et souhaitez réduire la charge sur le serveur Zabbix et améliorer les performances. La définition d'un déploiement à grande échelle est très large et dépend non seulement du nombre d'hôtes, mais aussi du nombre de valeurs collectées par seconde.

Macros secrètes

Vous pouvez utiliser des macros utilisateur secrètes soit comme texte secret, soit comme macros de coffre-fort secrètes.

Pour une sécurité renforcée lors de l'utilisation de macros de coffre-fort secrètes, il est recommandé de configurer les valeurs des macros afin qu'elles soient récupérées indépendamment par le serveur Zabbix et les proxys Zabbix. Par défaut, les valeurs des macros secrètes sont récupérées par le serveur Zabbix et propagées aux proxys Zabbix.

18 API

Aperçu L'API Zabbix vous permet de récupérer et de modifier par programmation la configuration de Zabbix, et fournit également un accès aux données historiques. Elle est largement utilisée pour :

- créer de nouvelles applications fonctionnant avec Zabbix ;
- intégrer Zabbix à un logiciel tiers ;
- automatiser les tâches de routine.

L'API Zabbix est une API basée sur HTTP et elle est fournie comme partie intégrante de l'interface web. 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 de Zabbix dans vos applications Python, consultez la bibliothèque Python pour Zabbix.

> **Note:**
> L'accès des utilisateurs dans Zabbix, y compris à la fois à la configuration et aux données historiques, dépend du type d'utilisateur, du rôle utilisateur attribué et des groupes d'utilisateurs.

Structure L'API possède un certain nombre de méthodes qui sont nominalement regroupées dans des 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".

> **Note:**
> La plupart des API contiennent au moins quatre méthodes: `get`, `create`, `update` et `delete` pour respectivement récupérer, créer, mettre à jour et supprimer des données, mais certaines API peuvent fournir un ensemble de méthodes totalement différentes.

Exécution des requêtes Une fois le frontend configuré, vous pouvez utiliser des requêtes HTTP distantes pour appeler l'API. Pour cela, vous devez envoyer des requêtes HTTP POST au fichier `api_jsonrpc.php` situé dans le répertoire du frontend. Par exemple, si votre frontend Zabbix est installé 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 des valeurs suivantes : `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 d'API appelée ;
- `params` - les paramètres qui seront transmis à la méthode d'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, à son tour, 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.

Exemple de flux de travail La section suivante vous présentera quelques exemples d'utilisation de manière plus détaillée.

Authentification Avant de pouvoir accéder aux données de Zabbix, vous devez vous connecter et obtenir un jeton d'authentification. Ceci peut être fait en utilisant la méthode user.login. Supposons que vous souhaitiez vous connecter en tant qu'administrateur Zabbix standard. Votre demande JSON ressemblera à ceci :

```
{
    "jsonrpc": "2.0",
    "method": "user.login",
    "params": {
        "user": "Admin",
        "password": "zabbix"
    },
    "id": 1,
    "auth": null
}
```

Examinons de plus près l'objet de la requête. Il a les propriétés suivantes :

- `jsonrpc` - version du protocole JSON-RPC utilisée par l'API ; l'API Zabbix implémente JSON-RPC version 2.0 ;
- `method` - la méthode API appelée ;
- `params` - paramètres qui seront transmis à la méthode API ;
- `id` - un identifiant arbitraire de la requête;
- `auth` - un jeton d'authentification utilisateur ; comme nous n'en avons pas encore, il est défini à `null`.

Si vous avez correctement fourni les informations d'identification, la réponse renvoyée par l'API contiendra le jeton d'authentification de l'utilisateur :

```
{
    "jsonrpc": "2.0",
    "result": "0424bd59b807674191e7d77572075f33",
    "id": 1
}
```

L'objet de réponse contient à son tour les propriétés suivantes :

- `jsonrpc` - encore une fois, la version du protocole JSON-RPC ;
- `result` - les données retournées par la méthode (ici le token dans cet exemple) ;
- `id` - identifiant de la requête correspondante.

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).

Récupération des hôtes Vous disposez maintenant d'un jeton d'authentification utilisateur valide (représenté par 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 transmettre 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 d'objet que vous souhaitez récupérer. Ainsi, vous éviterez de tout récupérer.

Création d'un nouvel élément Maintenant, créons un nouvel élément sur l'hôte "Zabbix server" en utilisant les données que nous avons obtenues de la précédente requête `host.get`. Cela peut être fait en utilisant la méthode item.create :

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "name": "Free disk space on /home/joe/",
        "key_": "vfs.fs.size[/home/joe/,free]",
        "hostid": "10084",
        "type": 0,
        "value_type": 3,
        "interfaceid": "1",
        "delay": 30
    },
    "auth": "0424bd59b807674191e7d77572075f33",
    "id": 3
}
```

Une réponse valide contiendra l'ID du nouvel élément créé, qui peut 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 d'autres méthodes de création peuvent également accepter des tableaux d'objets et créer plusieurs éléments avec un seul appel d'API.

Création de plusieurs déclencheurs Donc, comme les méthodes *create* acceptent les tableaux, nous pouvons ajouter plusieurs déclencheurs comme ceci :

```
{
    "jsonrpc": "2.0",
    "method": "trigger.create",
    "params": [
        {
            "description": "Processor load is too high on {HOST.NAME}",
            "expression": "last(/Linux server/system.cpu.load[percpu,avg1])>5",
        },
        {
            "description": "Too many processes on {HOST.NAME}",
            "expression": "avg(/Linux server/proc.num[],5m)>300",
        }
    ],
    "auth": "0424bd59b807674191e7d77572075f33",
    "id": 4
}
```

Une réponse valide contiendra les identifiants des déclencheurs nouvellement créés :

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17369",
            "17370"
        ]
    },
    "id": 4
```

```
}
```

Mise à jour d'un élément Activer un élément, autrement dit, définir son statut à "0" :

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "10092",
        "status": 0
    },
    "auth": "0424bd59b807674191e7d77572075f33",
    "id": 5
}
```

Une réponse valide contiendra l'ID de l'élément mis à jour :

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "10092"
        ]
    },
    "id": 5
}
```

Note:
La méthode `item.update` ainsi que d'autres méthodes de mise à jour peuvent également accepter des tableaux d'objets et mettre à jour plusieurs éléments avec un seul appel d'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 en cas de réussite contiendra les ID des déclencheurs mis à jour :

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938",
            "13939"
        ]
    },
    "id": 6
}
```

Gestion des erreurs Jusque-là, tout ce que nous avons essayé a bien fonctionné. Mais, que se passe-t-il si nous tentons de faire un appel incorrect à l'API ? Essayons de créer un autre hôte en appelant la méthode host.create mais en omettant le paramètre obligatoire `groups`.

```
{
    "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"
            }
        ]
    },
    "id": 7,
    "auth": "0424bd59b807674191e7d77572075f33"
}
```

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 survient, au lieu de la propriété `result`, l'objet de réponse contiendra une propriété `error` avec les données suivantes :

- `code` - un code d'erreur ;
- `message` - un résumé succinct de l'erreur ;
- `data` - un message d'erreur plus détaillé.

Des erreurs peuvent survenir dans différents cas, tels que l'utilisation de valeurs d'entrée incorrectes, l'expiration du délai d'une session ou la tentative d'accès à des objets non existants. Votre application doit pouvoir gérer ce type d'erreurs.

Versions de l'API Pour simplifier la gestion des versions 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 mettre à jour votre application afin qu'elle utilise les fonctionnalités spécifiques à la version.

Nous garantissons la compatibilité ascendante des fonctionnalités dans une version majeure. Lorsque nous effectuons des modifications incompatibles entre les versions majeures, nous conservons généralement les anciennes fonctionnalités comme obsolètes dans la version suivante et nous les supprimons ensuite de la version suivante. Parfois, nous pouvons supprimer des fonctionnalités entre versions majeures sans fournir de compatibilité ascendante. Il est important que vous ne vous reposiez jamais sur des fonctionnalités obsolètes et que vous migriez vers de nouvelles alternatives dès que possible.

> **Note:**
> Vous pouvez suivre toutes les modifications apportées à l'API dans le journal des modifications de l'API.

Pour aller plus loin Vous avez maintenant suffisamment de connaissances pour commencer à travailler avec l'API Zabbix, cependant, ne vous arrêtez pas ici. Pour approfondir le sujet, 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

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 au niveau des services et obtenez des informations détaillées sur les SLA de n'importe quel service.

API de service

Collecte de 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, y compris les interfaces d'hôte, les macros d'hôte et les périodes de maintenance.

API des hôtes | API des groupes d'hôtes | API des interfaces d'hôte | API des macros utilisateur | API des correspondances de valeurs | API de la maintenance

Éléments

Définissez les éléments à surveiller.

API des éléments

Déclencheurs

Configurez des déclencheurs pour être averti des problèmes dans votre système. Gérez les dépendances des 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

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

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 sur l'API Récupérez la version de l'API Zabbix afin que votre application puisse utiliser des fonctionnalités spécifiques à cette version.

API 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.

Propriété	Type	Description
actionid	ID	ID de l'action. Comportement de la propriété : - *lecture seule* - *obligatoire* pour les opérations de mise à jour
esc_period	string	Durée par défaut de l'étape d'opération. Doit être d'au moins 60 secondes. Accepte des secondes, une unité de temps avec suffixe ou une macro utilisateur. Comportement de la propriété : - *pris 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 du 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 : - *constant* - *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 mettre en pause l'escalade si l'événement est un événement symptôme. Valeurs possibles : 0 - Ne pas mettre en pause l'escalade pour les problèmes symptôme ; 1 - *(par défaut)* Mettre en pause l'escalade pour les problèmes symptôme. Comportement de la propriété : - *pris en charge* si `eventsource` est défini sur « événement créé par un déclencheur »
pause_suppressed	integer	Indique s'il faut mettre en pause l'escalade pendant les périodes de maintenance ou non. Valeurs possibles : 0 - Ne pas mettre en pause l'escalade ; 1 - *(par défaut)* Mettre en pause l'escalade. Comportement de la propriété : - *pris en charge* si `eventsource` est défini sur « événement créé par un déclencheur »

Propriété	Type	Description
notify_if_canceled	integer	Indique s'il faut envoyer une notification lorsque l'escalade est annulée. Valeurs possibles : 0 - Ne pas notifier lorsque l'escalade est annulée ; 1 - *(par défaut)* Notifier lorsque l'escalade est annulée. Comportement de la propriété : - *pris 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 exécutée lorsqu'une action est exécutée. Il possède les propriétés suivantes.

Propriété	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 tags d'hôte ; 14 - supprimer des tags d'hôte. 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 ». Valeurs possibles si `eventsource` de l'objet Action est défini sur « événement interne » : 0 - « envoyer un message ». Comportement de la propriété : - *obligatoire*
esc_period	string	Durée d'une étape d'escalade en secondes. Doit être supérieure à 60 secondes. Accepte des secondes, une unité de temps avec suffixe ou une macro utilisateur. Si la valeur est définie sur 0 ou 0s, la période d'escalade d'action par défaut sera utilisée. Par défaut : 0s. Comportement de la propriété : - *pris en charge* si `eventsource` de l'objet Action est défini sur « événement créé par un déclencheur », « événement interne » ou « événement créé lors d'une mise à jour de l'état du service »

Propriété	Type	Description
esc_step_from	integer	Étape à partir de laquelle démarrer l'escalade. Par défaut : 1. Comportement de la propriété : - *pris en charge* si `eventsource` de l'objet Action est défini sur « événement créé par un déclencheur », « événement interne » ou « événement créé lors d'une mise à jour de l'état du service »
esc_step_to	integer	Étape à laquelle terminer l'escalade. Par défaut : 1. Comportement de la propriété : - *pris en charge* si `eventsource` de l'objet Action est défini sur « événement créé par un déclencheur », « événement interne » ou « événement créé lors d'une mise à jour de l'état du service »
evaltype	integer	Méthode d'évaluation des conditions d'opération. Valeurs possibles : 0 - *(par défaut)* Et/Ou ; 1 - Et ; 2 - Ou.
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 `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 `operationtype` est défini sur « script global » et que `opcommand_grp` n'est pas défini
opconditions	array	Conditions d'opération utilisées pour les actions de déclencheur. L'objet de condition d'opération est décrit en détail ci-dessous.
opgroup	array	Groupes d'hôtes auxquels ajouter des hôtes. Les groupes d'hôtes doivent avoir la propriété `groupid` définie. Comportement de la propriété : - *obligatoire* si `operationtype` est défini sur « ajouter à un groupe d'hôtes » ou « supprimer d'un groupe d'hôtes »
opmessage	object	Objet contenant les données relatives au message envoyé par l'opération. L'objet de message d'opération est décrit en détail ci-dessous. Comportement de la propriété : - *obligatoire* si `operationtype` est défini sur « envoyer un message »

Propriété	Type	Description
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
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
optemplate	array	Modèles à lier aux hôtes. Les modèles doivent avoir la propriété `templateid` définie. Comportement de la propriété : - *obligatoire* 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 : - *obligatoire* si `operationtype` est défini sur « définir le mode d'inventaire de l'hôte »
optag	array	Tags d'hôte à ajouter ou à supprimer. Les tags doivent avoir la propriété `tag` définie. La propriété `value` est facultative. Comportement de la propriété : - *pris en charge* si `operationtype` est défini sur « ajouter des tags d'hôte » ou « supprimer des tags d'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 l'objet 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 « envoyer un message », ou si `operationtype` de l'objet d'opération de mise à jour d'action est défini sur « notifier toutes les parties impliquées »

Propriété	Type	Description
message	string	Texte du message d'opération. Comportement de la propriété : - *pris en charge* si `default_msg` est défini sur « utiliser les données de l'opération »
subject	string	Objet du message d'opération. Comportement de la propriété : - *pris en charge* si `default_msg` est défini sur « utiliser les données de l'opération »

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 de la 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.
opcommand	object	Comportement de la propriété : - *obligatoire* Script global à exécuter. Le script global doit avoir la propriété `scriptid` définie.
opcommand_grp	array	Comportement de la propriété : - *obligatoire* si `operationtype` est défini sur « script global » Groupes d'hôtes sur lesquels exécuter les scripts globaux. Les groupes d'hôtes doivent avoir la propriété `groupid` définie.
opcommand_hst	array	Comportement de la propriété : - *obligatoire* si `eventsource` de l'objet Action est défini sur « événement créé par un déclencheur », et `operationtype` est défini sur « script global », et `opcommand_hst` n'est pas défini Hôtes sur lesquels exécuter les scripts globaux. Les hôtes doivent avoir la propriété `hostid` définie.
opmessage	object	Comportement de la propriété : - *obligatoire* si `eventsource` de l'objet Action est défini sur « événement créé par un déclencheur », et `operationtype` est défini sur « script global », et `opcommand_grp` n'est pas défini Objet contenant les données du message envoyé par l'opération de récupération. L'objet de message d'opération est décrit en détail ci-dessus.
opmessage_grp	array	Comportement de la propriété : - *obligatoire* si `operationtype` est défini sur « envoyer un message » Groupes d'utilisateurs auxquels envoyer des messages. Les groupes d'utilisateurs doivent avoir la propriété `usrgrpid` définie.
opmessage_usr	array	Comportement de la propriété : - *obligatoire* si `operationtype` est défini sur « envoyer un message » et `opmessage_usr` n'est pas défini 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 `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 exécutée lorsqu'un problème est mis à jour (commenté, acquitté, avec changement de 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", et `operationtype` est défini sur "script global", et `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", et `operationtype` est défini sur "script global", et `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 `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 : - *obligatoire* si `operationtype` est défini sur "envoyer un message" et `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.

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. Property behavior: - *required*
evaltype	integer	Méthode d'évaluation des conditions du filtre. Valeurs possibles : 0 - Et/Ou ; 1 - Et ; 2 - Ou ; 3 - Expression personnalisée. Property behavior: - *required*
eval_formula	string	Expression générée qui sera utilisée pour évaluer les conditions du filtre. L'expression contient des identifiants 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. Property behavior: - *read-only*
formula	string	Expression définie par l'utilisateur à utiliser pour évaluer les conditions des filtres avec une expression personnalisée. L'expression doit contenir des identifiants qui font référence à des conditions de filtre spécifiques par leur `formulaid`. Les identifiants utilisés dans l'expression doivent correspondre exactement à ceux définis dans les conditions du filtre : aucune condition ne peut rester inutilisée ou omise. Property behavior: - *required* si `evaltype` est défini sur « custom expression »

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.

Property	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 d'hôtes ; 1 - hôte ; 2 - déclencheur ; 3 - nom de l'événement ; 4 - sévérité du déclencheur ; 6 - période ; 13 - modèle d'hôte ; 16 - le problème est supprimé ; 25 - tag d'événement ; 26 - valeur du tag d'événement. Valeurs possibles si `eventsource` de l'objet Action est défini sur « événement créé par une règle de découverte » : 7 - IP de l'hôte ; 8 - type de service découvert ; 9 - port du service découvert ; 10 - statut de découverte ; 11 - durée de disponibilité ou d'indisponibilité ; 12 - valeur reçue ; 18 - règle de découverte ; 19 - vérification de découverte ; 20 - proxy ; 21 - objet de découverte. Valeurs possibles si `eventsource` de l'objet Action est défini sur « événement créé par l'auto-enregistrement de l'agent actif » : 20 - proxy ; 22 - nom d'hôte ; 24 - métadonnées de l'hôte. 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 - tag d'événement ; 26 - valeur du tag d'événement. Valeurs possibles si `eventsource` de l'objet Action est défini sur « événement créé lors de la mise à jour du statut du service » : 25 - tag d'événement ; 26 - valeur du tag d'événement ; 27 - service ; 28 - nom du service. Comportement de la propriété : - *obligatoire*
value	string	Valeur à comparer. Comportement de la propriété : - *obligatoire*

Property	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 n'importe quelle 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 n'importe quelle 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 du statut du service », que `conditiontype` est défini sur n'importe quelle 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 il sera généré à nouveau lors de leur demande ultérieure.
operator	integer	Opérateur de condition. Valeurs possibles : 0 - *(par défaut)* égal à ; 1 - différent de ; 2 - contient ; 3 - ne contient pas ; 4 - dans ; 5 - supérieur ou égal à ; 6 - inférieur ou égal à ; 7 - pas dans ; 8 - correspond ; 9 - ne correspond pas ; 10 - Oui ; 11 - Non.

Note:
Pour mieux comprendre comment utiliser les filtres avec différents types d'expressions, consultez les exemples dans les pages des méthodes action.get et action.create.

Les opérateurs et valeurs suivants sont pris en charge pour chaque type de condition.

Condition	Condition name	Supported operators	Expected value
0	Groupe d'hôtes	égal à, différent de	ID du groupe d'hôtes.
1	Hôte	égal à, différent de	ID de l'hôte.
2	Déclencheur	égal à, différent de	ID du déclencheur.
3	Nom de l'événement	contient, ne contient pas	Nom de l'événement.
4	Sévérité du déclencheur	égal à, différent de, supérieur ou égal à, inférieur ou égal à	Sévérité du déclencheur. Consultez la propriété `severity` du déclencheur pour la liste des sévérités de déclencheur prises en charge.
5	Valeur du déclencheur	égal à	Valeur du déclencheur. Consultez la propriété `value` du déclencheur pour la liste des valeurs de déclencheur prises en charge.

Condition	Condition name	Supported operators	Expected value
6	Période	dans, pas dans	Heure à laquelle l'événement a été déclenché sous forme de période.
7	IP de l'hôte	égal à, différent de	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 pris en charge des plages d'adresses IP.
8	Type de service découvert	égal à, différent de	Type de 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 la liste des types pris en charge.
9	Port du service découvert	égal à, différent de	Une ou plusieurs plages de ports, séparées par des virgules.
10	Statut de découverte	égal à	Statut d'un objet découvert. Valeurs possibles : 0 - hôte ou service actif ; 1 - hôte ou service inactif ; 2 - hôte ou service découvert ; 3 - hôte ou service perdu.
11	Durée de disponibilité ou d'indisponibilité	supérieur ou égal à, inférieur ou égal à	Temps indiquant depuis combien de temps l'objet découvert se trouve dans le statut actuel, en secondes.
12	Valeurs reçues	égal à, différent de, supérieur ou égal à, 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	égal à, différent de	ID du modèle lié.
16	Le problème est supprimé	Oui, Non	Aucune valeur requise : l'utilisation de l'opérateur « Oui » signifie que le problème doit être supprimé, « Non » - non supprimé.
18	Règle de découverte	égal à, différent de	ID de la règle de découverte.
19	Vérification de découverte	égal à, différent de	ID de la vérification de découverte.
20	Proxy	égal à, différent de	ID du proxy.
21	Objet de découverte	égal à	Type d'objet ayant déclenché l'événement de découverte. Valeurs possibles : 1 - hôte découvert ; 2 - service découvert.
22	Nom d'hôte	contient, ne contient pas, correspond, ne correspond pas	Nom d'hôte. L'utilisation d'une expression régulière est prise en charge pour les opérateurs *correspond* et *ne correspond pas* dans les conditions d'auto-enregistrement.
23	Type d'événement	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, ne correspond pas	Métadonnées de l'hôte auto-enregistré. L'utilisation d'une expression régulière est prise en charge pour les opérateurs *correspond* et *ne correspond pas*.

Condition	Condition name	Supported operators	Expected value
25	Tag	égal à, différent de, contient, ne contient pas	Tag d'événement.
26	Valeur du tag	égal à, différent de, contient, ne contient pas	Valeur du tag d'événement.
27	Service	égal à, différent de	ID du service.
28	Nom du service	égal à, différent de	Nom du service.

action.create

Description

`object action.create(object/array actions)`

Cette méthode permet de créer de nouvelles actions.

Note:
Cette méthode est disponible uniquement pour les types d'utilisateurs *Admin* et *Super admin*. Les autorisations d'appeler la méthode peuvent être révoquées dans les paramètres de rôles utilisateur. Voir 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 également

- 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 permet de supprimer des actions.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateurs *Admin* et *Super admin*. Les autorisations d'appeler la méthode peuvent être révoquées dans les paramètres du le rôle d'utilisateur. Voir Rôles Utilisateur 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 en fonction des paramètres donnés.

Note:
Cette méthode est disponible à tous les types d'utilisateurs. Les autorisations d'appeler la méthode peuvent être révoquées dans les paramètres du le rôle d'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
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 également

- 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 permet de mettre à jour des actions existantes.

> **Note:**
> Cette méthode n'est disponible que pour les utilisateurs de type *Admin* et *Super admin*. Les autorisations d'appeler la méthode peuvent être révoquées dans les paramètres du rôle de l'utilisateur. Voir Rôles Utilisateur (/manual/web_interface/frontend_sections/users/user_roles) pour plus d'information.

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.

Paramètre	Type	Description
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 également

- 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 tentatives d'envoi du message par Zabbix.
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 l'agent Zabbix, mais celui-ci n'était pas disponible.
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 des alertes en fonction des paramètres donnés.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Autorisations pour appeler la méthode peut être révoqué dans les paramètres du rôle 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
alertids	ID/array	Retourner uniquement les alertes avec les ID donnés.
actionids	ID/array	Retourner uniquement les alertes générées par les actions données.
eventids	ID/array	Retourner uniquement les alertes générées par les événements donnés.
groupids	ID/array	Retourner uniquement les alertes générées par des objets provenant des groupes d'hôtes donnés.
hostids	ID/array	Retourner uniquement les alertes générées par des objets provenant des hôtes donnés.
mediatypeids	ID/array	Retourner uniquement les alertes de message ayant utilisé les types de média donnés.
objectids	ID/array	Retourner uniquement les alertes générées par les objets donnés
userids	ID/array	Retourner uniquement les alertes de message qui ont été envoyées aux utilisateurs donnés.
eventobject	integer	Retourner uniquement les alertes générées par des événements liés à des objets du type donné. Voir l'`object` d'événement pour une liste des types d'objets pris en charge. Par défaut : 0 - déclencheur.
eventsource	integer	Retourner uniquement les alertes générées par des événements du type donné. Voir la `source` d'événement pour une liste des types d'événements pris en charge. Par défaut : 0 - événements de déclencheur.
time_from	timestamp	Retourner uniquement les alertes qui ont été générées après l'heure donnée.
time_till	timestamp	Retourner uniquement les alertes qui ont été générées avant l'heure donnée.
selectHosts	query	Retourner une propriété `hosts` avec les données des hôtes qui ont déclenché l'opération d'action.
selectMediatypes	query	Retourner 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	Retourner 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	Trier le résultat selon les propriétés données. Valeurs possibles : `alertid`, `clock`, `eventid`, `mediatypeid`, `sendto`, `status`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	

Paramètre	Type	Description
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 également

- 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 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*.

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.

Propriété	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 `$ALLOW_HTTP_AUTH` est activé dans le fichier de configuration du frontend (*zabbix.conf.php*).
http_login_form	integer	Formulaire de connexion par défaut. Valeurs possibles : 0 - *(par défaut)* Formulaire de connexion Zabbix ; 1 - Formulaire de connexion HTTP. Comportement de la propriété : - *prise en charge* si `$ALLOW_HTTP_AUTH` est activé dans le fichier de configuration du frontend (*zabbix.conf.php*).
http_strip_domains	string	Nom de domaine à supprimer. Comportement de la propriété : - *prise en charge* si `$ALLOW_HTTP_AUTH` est activé dans le fichier de configuration du frontend (*zabbix.conf.php*).
http_case_sensitive	integer	Connexion HTTP sensible à la casse. Valeurs possibles : 0 - Désactivée ; 1 - *(par défaut)* Activée. Comportement de la propriété : - *prise en charge* si `$ALLOW_HTTP_AUTH` est activé dans le fichier de configuration du frontend (*zabbix.conf.php*).
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.

Propriété	Type	Description
ldap_userdirectoryid	ID	ID du répertoire d'utilisateurs par défaut pour l'authentification LDAP. Utilisé pour les groupes d'utilisateurs dont `gui_access` est défini sur LDAP ou Valeur par défaut du système. Comportement de la propriété : - *obligatoire* 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	Longueur minimale requise du mot de passe. Les valeurs possibles vont de 1 à 70. Par défaut : 8.
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 si le mot de passe ne figure pas dans la liste des mots de passe couramment utilisés et 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 binaire ; toute somme des valeurs bitmap possibles est acceptée (par exemple, 15 pour vérifier toutes les règles).
ldap_jit_status	integer	Statut du provisionnement LDAP. Valeurs possibles : 0 - Désactivé pour les fournisseurs d'identité LDAP configurés ; 1 - Activé pour les fournisseurs d'identité LDAP configurés.
saml_jit_status	integer	Statut du provisionnement SAML. Valeurs possibles : 0 - Désactivé pour les fournisseurs d'identité SAML configurés ; 1 - Activé pour les fournisseurs d'identité SAML configurés.
jit_provision_interval	string	Intervalle de temps entre les requêtes de provisionnement JIT pour l'utilisateur connecté. Accepte les secondes et une unité de temps avec suffixe, avec prise en charge des mois et des années (3600s,60m,1h,1d,1M,1y). Valeur minimale : 1h. Par défaut : 1h. Disponible uniquement pour le provisionnement LDAP.

Propriété	Type	Description
disabled_usrgrpid	ID	ID du groupe d'utilisateurs auquel affecter l'utilisateur déprovisionné. Le groupe d'utilisateurs doit être désactivé et ne peut pas être activé ni supprimé lorsqu'il est configuré. Comportement de la propriété : - *obligatoire* si `ldap_jit_status` est défini sur "Activé pour les fournisseurs d'identité LDAP configurés", ou si `saml_jit_status` est défini sur "Activé pour les fournisseurs d'identité 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 MFA activée. Comportement de la propriété : - *obligatoire* si `mfa_status` est défini sur "Activée"

authentication.get

Description

`object authentication.get(object parameters)`

La méthode permet de récupérer l'objet d'authentification en fonction des paramètres donnés.

> **Note:**
> Cette méthode n'est disponible que pour le type d'utilisateur *Super administrateur*. Les autorisations d'appeler la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir 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 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 permet de mettre à jour les paramètres d'authentification existants.

> **Note:**
> Cette méthode est disponible uniquement pour les utilisateurs de type *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir les rôles utilisateur 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
  - Hôte de l'élément de carte
  - Groupe d'hôtes de l'élément de carte
  - Carte de l'élément de carte
  - Déclencheur de l'élément de carte
  - Tag de l'élément de carte
  - URL de l'élément de carte
- 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

Propriété	Type	Description
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.
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 évidence des icônes est activée. Valeurs possibles : 0 - mise en évidence désactivée ; 1 - *(par défaut)* mise en évidence activée.
iconmapid	ID	ID de la correspondance 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)* bas ; 1 - gauche ; 2 - droite ; 3 - 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"

Propriété	Type	Description
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.
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.

Propriété	Type	Description
markelements	integer	Indique s'il faut mettre en évidence les éléments de la carte dont le statut a récemment changé. Valeurs possibles : 0 - *(par défaut)* ne pas mettre en évidence les éléments ; 1 - mettre en évidence les éléments.
severity_min	integer	Sévérité minimale des déclencheurs qui seront affichés sur la carte. Reportez-vous à la propriété `severity` du déclencheur pour obtenir la liste des sévérités 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.
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"

Propriété	Type	Description
elementtype	integer	Type de l'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 du groupe d'hôtes occupera toute la carte ; 1 - l'élément du 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 élément unique ; 1 - afficher séparément chaque hôte du groupe.
evaltype	integer	Méthode d'évaluation de la condition de filtrage par balises de l'élément de carte evaluation method. Valeurs possibles : 0 - *(par défaut)* Et/Ou ; 2 - Ou.
height	integer	Hauteur en pixels de l'élément de groupe d'hôtes à taille fixe. Par défaut : 200.
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.

Propriété	Type	Description
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 - masquage automatique.
sysmapid	ID	ID de la carte à laquelle appartient l'élément. 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.
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 d'élément de carte Groupe d'hôtes définit un élément de groupe d'hôtes.

Propriété	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 Trigger

L'objet d'élément de carte Trigger définit un ou plusieurs éléments de trigger.

Propriété	Type	Description
triggerid	ID	ID du trigger.

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 : - *obligatoire*
selementid2	ID	ID du premier élément de carte lié à l'autre extrémité. Comportement de la propriété : - *obligatoire*
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.
linktriggers	array	Déclencheurs de lien de carte à utiliser comme indicateurs d'état du lien. L'objet de déclencheur de lien de carte est décrit en détail ci-dessous. Comportement de la propriété : - *obligatoire* si `indicator_type` est défini sur "trigger" (1).
itemid	ID	ID de l'élément. Comportement de la propriété : - *obligatoire* 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 : - *obligatoire* 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 : - *obligatoire* si `indicator_type` est défini sur "item value" (2).

Propriété	Type	Description
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 en pointillés ; 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 surlignages.
sortorder	integer	Utilisé pour définir l'ordre de tri des surlignages. Disponible uniquement pour les surlignages.
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*
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 autorisations de carte basées sur les groupes d'utilisateurs. Elle possède les propriétés suivantes :

Propriété	Type	Description
sysmapusrgrpid	ID	ID du groupe d'utilisateurs de la carte. Comportement de la propriété : - *lecture seule*
usrgrpid	ID	ID du groupe d'utilisateurs. 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*

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. Cette 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.
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.

Propriété	Type	Description
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)`.
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.

Propriété	Type	Description
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 - – – –. Par défaut : 0.
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 permet de créer de nouvelles cartes.

> **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. Voir User roles 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 utilisateur de carte à créer sur la carte.
userGroups	array	Partages de groupes 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 devrez 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 également

- Elément 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 permet de supprimer des cartes.

Note:
Cette méthode est disponible pour tous les types d'utilisateur. Les autorisations pour appel la méthode peuvent être révoquées dans les réglages du rôle d'utilisateur. Voir les 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 donnés.

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. 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
sysmapids	ID/array	Renvoie uniquement les cartes avec les ID donnés.
userids	ID/array	Renvoie uniquement les cartes appartenant aux ID d'utilisateur donnés.

Paramètre	Type	Description
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 également

- Icones de la carte
- [Élément de la carte] (object#map_element)
- Lien vers la carte
- [URL de la carte] (object#map_url)
- [Utilisateur de la carte] (object#map_user)
- [Groupe d'utilisateurs de la carte] (object#map_user_group)
- Formes de carte (object#map_shapes)
- [Lignes de la carte] (object#map_lines)

Source

CMap::get() dans *ui/include/classes/api/services/CMap.php*.

map.update

Description

`object map.update(object/array maps)`

Cette méthode permet de mettre à jour des cartes existantes.

> **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. Voir User roles pour plus d'informations.

Paramètres

`(object/array)` Propriétés de la carte 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 remplaçant les liens existants.
selements	array	Éléments de carte remplaçant les éléments existants.
urls	array	URL de carte remplaçant les URL existantes.
users	array	Partages d'utilisateur de carte remplaçant les éléments existants.
userGroups	array	Partages de groupe d'utilisateurs de carte remplaçant les éléments existants.
shapes	array	Formes de carte remplaçant les formes existantes.
lines	array	Lignes de carte remplaçant 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 également

- Elément de carte
- Lien de carte
- URL de carte
- Utilisateur de carte
- Groupe utilisateur de carte
- Formes de cartes
- 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
- Mappage 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`.

Carte d'icônes

L'objet de carte d'icônes possède les propriétés suivantes.

Propriété	Type	Description
iconmapid	ID	ID de la carte 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 de la carte d'icônes. Comportement de la propriété : - *obligatoire* pour les opérations de création

Correspondance d'icônes

L'objet de correspondance d'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 d'icônes. Comportement de la propriété : - *obligatoire*
expression	string	Expression à faire correspondre avec 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*

Propriété	Type	Description
sortorder	integer	Position de la correspondance d'icônes dans la carte d'icônes. Comportement de la propriété : - *lecture seule*

iconmap.create

Description

`object iconmap.create(object/array iconMaps)`

Cette méthode permet de créer de nouvelles correspondances 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 du rôle utilisateur. Voir User roles pour plus d'informations.

Paramètres

`(object/array)` Cartes d'icônes à créer.

En plus des propriétés standard des cartes d'icônes, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
mappings	array	Correspondances d'icônes à créer pour la carte 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 également

- Correspondance d'icone

Source

CIconMap::create() dans *ui/include/classes/api/services/CIconMap.php*.

iconmap.delete

Description

`object iconmap.delete(array iconMapIds)`

Cette méthode permet de supprimer des plans 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 du rôle utilisateur. Voir 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)`

Cette méthode permet de récupérer des mappages d'icônes selon les paramètres donnés.

> **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.

Paramètre	Type	Description
iconmapids	ID/array	Retourner uniquement les plans d'icônes avec les ID donnés.
sysmapids	ID/array	Retourner uniquement les plans d'icônes utilisés dans les cartes données.
selectMappings	query	Retourner une propriété `mappings` avec les correspondances d'icônes utilisées.
sortfield	string/array	Trier le résultat selon les propriétés données. 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 également

- Correspondance d'icone

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 du rôle utilisateur. Voir User roles pour plus d'informations.

Paramètres

`(object/array)` Correspondance d'icônes à mettre à jour.

La propriété `iconmapid` doit être définie pour chaque correspondance 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 de correspondance d'icônes standard, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
mappings	array	Correspondances d'icônes en remplacement des correspondance 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 également

- Correspondance d'icone

Source

CIconMap::update() dans *ui/include/classes/api/services/CIconMap.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 du rôle utilisateur. Voir User roles pour plus d'informations.

Paramètres

`(object)` Paramètres définissant les objets à exporter et le format à utiliser.

Paramètre	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 : - *obligatoire*
prettyprint	boolean	Rend 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.
options	object	Objets à exporter. L'objet `options` comporte les paramètres suivants : `host_groups` - `(array)` ID des groupes d'hôtes à exporter ; `hosts` - `(array)` ID des hôtes à exporter ; `images` - `(array)` ID des images à exporter ; `maps` - `(array)` ID des cartes à exporter ; `mediaTypes` - `(array)` ID des types de média à exporter ; `template_groups` - `(array)` ID des groupes de modèles à exporter ; `templates` - `(array)` ID des modèles 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 : - *obligatoire*

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 les données de configuration sous forme de chaîne sérialisée.

Note:
Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appeler la méthode peuvent être révoquées dans les paramètres de rôle utilisateur. Voir Rôles utilisateur pour plus d'informations.

Paramètres

`(object)` Paramètres contenant les données à importer et les règles définissant la manière dont les données doivent être traitées.

Paramètre	Type	Description
format	string	Format de la chaîne sérialisée. Valeurs possibles : `yaml` - YAML ; `xml` - XML ; `json` - JSON. Comportement du paramètre : - *obligatoire*
source	string	Chaîne sérialisée contenant les données de configuration. Comportement du paramètre : - *obligatoire*
rules	object	Règles définissant la manière dont les objets nouveaux et existants doivent être importés. Les utilisateurs de type *Admin* peuvent importer uniquement les objets pour lesquels ils disposent d'une autorisation en *lecture-écriture*, 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 autorisation 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 : - *obligatoire*

Attention:
Si aucune règle n'est fournie, la configuration ne sera pas mise à jour.

L'objet `rules` prend en charge les paramètres suivants.

Paramètre	Type	Description
discoveryRules	object	Règles définissant la manière d'importer les règles 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`.
graphs	object	Règles définissant la manière d'importer les 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 définissant la manière d'importer les 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`.
template_groups	object	Règles définissant la manière d'importer les 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`.
hosts	object	Règles définissant la manière d'importer les 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 définissant la manière d'importer les 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 définissant la manière d'importer les 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`.

Paramètre	Type	Description
items	object	Règles définissant la manière d'importer les 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 définissant la manière d'importer les 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 définissant la manière d'importer les 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`.
templateLinkage	object	Règles définissant la manière d'importer les liaisons 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 définissant la manière d'importer les 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`.
templateDashboards	object	Règles définissant la manière d'importer les tableaux de bord de modèles. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux tableaux de bord de modèles seront créés ; valeur par défaut : `false` ; `updateExisting` - `(boolean)` si défini sur `true`, les tableaux de bord de modèles existants seront mis à jour ; valeur par défaut : `false` ; `deleteMissing` - `(boolean)` si défini sur `true`, les tableaux de bord de modèles absents des données importées seront supprimés de la base de données ; valeur par défaut : `false`.

Paramètre	Type	Description
triggers	object	Règles définissant la manière d'importer les 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 définissant la manière d'importer les correspondances de valeurs d'hôte ou de modèle. Paramètres pris en charge : `createMissing` - (`boolean`) si défini sur `true`, de nouvelles correspondances de valeurs seront créées ; valeur par défaut : `false` ; `updateExisting` - (`boolean`) si défini sur `true`, les correspondances de valeurs existantes seront mises à jour ; valeur par défaut : `false` ; `deleteMissing` - (`boolean`) si défini sur `true`, les correspondances de valeurs absentes des données importées seront supprimées 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 un fichier d'importation avec les éléments actuels du système et indique ce qui sera modifié si ce fichier d'importation 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. Voir User roles pour plus d'informations.

Paramètres

`(object)` Paramètres contenant les données possibles à importer et les règles définissant la manière dont les données doivent être traitées.

Paramètre	Type	Description
format	string	Format de la chaîne sérialisée. Valeurs possibles : `yaml` - YAML ; `xml` - XML ; `json` - JSON. Comportement du paramètre : - *obligatoire*
source	string	Chaîne sérialisée contenant les données de configuration. Comportement du paramètre : - *obligatoire*
rules	object	Règles définissant comment les objets nouveaux et existants doivent être comparés. Les utilisateurs de type *Admin* et *User* peuvent comparer les nouveaux objets uniquement avec les objets existants pour lesquels ils disposent d'une autorisation en *lecture seule* ou en *lecture-écriture*. Le paramètre `rules` est décrit en détail dans le tableau ci-dessous. Comportement du paramètre : - *obligatoire*

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 ; tous les autres seront considérés comme « nouveaux ».

L'objet `rules` prend en charge les paramètres suivants.

Paramètre	Type	Description
discoveryRules	object	Règles définissant comment importer les règles LLD. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouvelles règles LLD seront créées ; par défaut : `false` ; `updateExisting` - `(boolean)` si défini sur `true`, les règles LLD existantes seront mises à jour ; par défaut : `false` ; `deleteMissing` - `(boolean)` si défini sur `true`, les règles LLD absentes des données importées seront supprimées de la base de données ; par défaut : `false`.
graphs	object	Règles définissant comment importer les graphiques. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux graphiques seront créés ; par défaut : `false` ; `updateExisting` - `(boolean)` si défini sur `true`, les graphiques existants seront mis à jour ; par défaut : `false` ; `deleteMissing` - `(boolean)` si défini sur `true`, les graphiques absents des données importées seront supprimés de la base de données ; par défaut : `false`.
host_groups	object	Règles définissant comment importer les groupes d'hôtes. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux groupes d'hôtes seront créés ; par défaut : `false` ; `updateExisting` - `(boolean)` si défini sur `true`, les groupes d'hôtes existants seront mis à jour ; par défaut : `false`.
template_groups	object	Règles définissant comment importer les groupes de modèles. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux groupes de modèles seront créés ; par défaut : `false` ; `updateExisting` - `(boolean)` si défini sur `true`, les groupes de modèles existants seront mis à jour ; par défaut : `false`.
hosts	object	Règles définissant comment importer les hôtes. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux hôtes seront créés ; par défaut : `false` ; `updateExisting` - `(boolean)` si défini sur `true`, les hôtes existants seront mis à jour ; par défaut : `false`. Ce paramètre ne fera aucune différence dans la sortie. Il est autorisé uniquement pour assurer la cohérence avec `configuration.import`.
httptests	object	Règles définissant comment importer les scénarios web. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux scénarios web seront créés ; par défaut : `false` ; `updateExisting` - `(boolean)` si défini sur `true`, les scénarios web existants seront mis à jour ; par défaut : `false` ; `deleteMissing` - `(boolean)` si défini sur `true`, les scénarios web absents des données importées seront supprimés de la base de données ; par défaut : `false`.

Paramètre	Type	Description
images	object	Règles définissant comment importer les images. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouvelles images seront créées ; par défaut : `false` ; `updateExisting` - `(boolean)` si défini sur `true`, les images existantes seront mises à jour ; par défaut : `false`. Ce paramètre ne fera aucune différence dans la sortie. Il est autorisé uniquement pour assurer la cohérence avec `configuration.import`.
items	object	Règles définissant comment importer les éléments de données. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux éléments de données seront créés ; par défaut : `false` ; `updateExisting` - `(boolean)` si défini sur `true`, les éléments de données existants seront mis à jour ; par défaut : `false` ; `deleteMissing` - `(boolean)` si défini sur `true`, les éléments de données absents des données importées seront supprimés de la base de données ; par défaut : `false`.
maps	object	Règles définissant comment importer les cartes. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouvelles cartes seront créées ; par défaut : `false` ; `updateExisting` - `(boolean)` si défini sur `true`, les cartes existantes seront mises à jour ; par défaut : `false`. Ce paramètre ne fera aucune différence dans la sortie. Il est autorisé uniquement pour assurer la cohérence avec `configuration.import`.
mediaTypes	object	Règles définissant comment importer les types de média. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux types de média seront créés ; par défaut : `false` ; `updateExisting` - `(boolean)` si défini sur `true`, les types de média existants seront mis à jour ; par défaut : `false`. Ce paramètre ne fera aucune différence dans la sortie. Il est autorisé uniquement pour assurer la cohérence avec `configuration.import`.
templateLinkage	object	Règles définissant comment importer les 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 ; 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 de données, déclencheurs, etc.) héritées des modèles dissociés ; par défaut : `false`.
templates	object	Règles définissant comment importer les modèles. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux modèles seront créés ; par défaut : `false` ; `updateExisting` - `(boolean)` si défini sur `true`, les modèles existants seront mis à jour ; par défaut : `false`.

Paramètre	Type	Description
templateDashboards	object	Règles définissant comment importer les tableaux de bord de modèles. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux tableaux de bord de modèles seront créés ; par défaut : `false` ; `updateExisting` - `(boolean)` si défini sur `true`, les tableaux de bord de modèles existants seront mis à jour ; par défaut : `false` ; `deleteMissing` - `(boolean)` si défini sur `true`, les tableaux de bord de modèles absents des données importées seront supprimés de la base de données ; par défaut : `false`.
triggers	object	Règles définissant comment importer les déclencheurs. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouveaux déclencheurs seront créés ; par défaut : `false` ; `updateExisting` - `(boolean)` si défini sur `true`, les déclencheurs existants seront mis à jour ; par défaut : `false` ; `deleteMissing` - `(boolean)` si défini sur `true`, les déclencheurs absents des données importées seront supprimés de la base de données ; par défaut : `false`.
valueMaps	object	Règles définissant comment importer les correspondances de valeurs d'hôte ou de modèle. Paramètres pris en charge : `createMissing` - `(boolean)` si défini sur `true`, de nouvelles correspondances de valeurs seront créées ; par défaut : `false` ; `updateExisting` - `(boolean)` si défini sur `true`, les correspondances de valeurs existantes seront mises à jour ; par défaut : `false` ; `deleteMissing` - `(boolean)` si défini sur `true`, les correspondances de valeurs absentes des données importées seront supprimées de la base de données ; 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* - *obligatoire* pour les opérations de mise à jour
name	string	Nom du connecteur. Comportement de la propriété : - *obligatoire* pour les opérations de création
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 : - *obligatoire* pour les opérations de création
protocol	integer	Protocole de communication. Valeurs possibles : 0 - *(par défaut)* Zabbix Streaming Protocol v1.0.

Property	Type	Description
data_type	integer	Type de données. Valeurs possibles : 0 - *(par défaut)* Valeurs d'élément ; 1 - Événements.
item_value_type	integer	Somme des types de valeurs d'élément à envoyer. Valeurs possibles : 1 - Numérique (float) ; 2 - Caractère ; 4 - Journal ; 8 - Numérique (non signé) ; 16 - Texte ; 32 - Binaire ; 64 - JSON. Par défaut : 31 - Tous les types d'élément (sauf binaire et JSON). Comportement de la propriété : - *pris en charge* si `data_type` est défini sur "Valeurs d'élément".
max_records	integer	Nombre maximal d'événements ou d'éléments pouvant être envoyés dans un seul message. Valeurs possibles : 0-2147483647 (valeur maximale d'un entier signé sur 32 bits). Par défaut : 0 - Illimité.
max_senders	integer	Nombre de processus d'envoi à exécuter pour ce connecteur. Valeurs possibles : 1-100. Par défaut : 1.
max_attempts	integer	Nombre de tentatives. Valeurs possibles : 1-5. Par défaut : 1.
attempt_interval	string	Intervalle entre les tentatives de nouvelle tentative. Accepte les secondes. Valeurs possibles : 0s-10s. Par défaut : 5s. Comportement de la propriété : - *pris en charge* si `max_attempts` est supérieur à 1.
timeout	string	Délai d'expiration. Les suffixes de temps sont pris en charge (par exemple, 30s, 1m). Les macros utilisateur sont prises en charge. Valeurs possibles : 1s-60s. Par défaut : 5s.
http_proxy	string	Chaîne de connexion du proxy HTTP(S) fournie sous la forme *[protocol]://[username[:password]@]proxy.example.com[:port]*. Les macros utilisateur sont prises en charge.

Property	Type	Description
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 : - *obligatoire* si `authtype` est défini sur "Bearer"
verify_peer	integer	Indique s'il faut valider que le certificat de l'hôte est authentique. Valeurs possibles : 0 - Ne pas valider ; 1 - *(par défaut)* Valider.
verify_host	integer	Indique s'il faut valider que le nom d'hôte de la connexion correspond à celui figurant dans le certificat de l'hôte. Valeurs possibles : 0 - Ne pas valider ; 1 - *(par défaut)* Valider.
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.
tags_evaltype	integer	Méthode d'évaluation des tags. Valeurs possibles : 0 - *(par défaut)* Et/Ou ; 2 - Ou.

Filtre de tags

Le filtre de tags permet d'exporter uniquement les valeurs d'éléments ou les événements correspondants. S'il n'est pas défini, tout sera exporté. L'objet de filtre de tags possède les propriétés suivantes.

Propriété	Type	Description
tag	string	Nom du tag. Comportement de la propriété : - *obligatoire*
operator	integer	Opérateur de condition. Valeurs possibles : 0 - *(par défaut)* Égal à ; 1 - Différent de ; 2 - Contient ; 3 - Ne contient pas ; 12 - Existe ; 13 - N'existe pas.
value	string	Valeur du tag.

connector.create

Description

`object connector.create(object/array connectors)`

Cette méthode 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 du rôle utilisateur. Voir Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Objets de connecteur à créer.

En plus des propriétés de connecteur standard, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
tags	array	Filtre de balises 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 permet de supprimer des entrées de connecteur.

> **Note:**
> Cette méthode est uniquement disponible 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 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 connecteur selon les paramètres donnés.

> **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
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 permet de mettre à jour des connecteurs existants.

> **Note:**
> Cette méthode est uniquement disponible 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 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*.

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* - *requis* pour les opérations de mise à jour
name	string	Nom de la corrélation. Comportement de la propriété : - *requis* 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 de filtre de corrélation définit un ensemble de conditions qui doivent être remplies pour exécuter les opérations de corrélation 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 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 : - *obligatoire* 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.

Propriété	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 »

Propriété	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. 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 dans les pages des méthodes correlation.get et correlation.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
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 permet de créer de nouvelles corrélations.

Note:
Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les permissions d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir 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 également

- 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 permet de supprimer les corrélations.

> **Note:**
> Cette méthode est uniquement disponible pour les utilisateurs type *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir les rôles utilisateur 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 donnés.

> **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. 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
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 données. Valeurs possibles : `correlationid`, `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	

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 également

- 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 permet de mettre à jour des corrélations existantes.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations d'appeler la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir 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 remplaçant le filtre actuel.
operations	array	Opérations de corrélation remplaçant 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 également

- 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 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
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	Heure à laquelle 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 - moyen ; 4 - élevé ; 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 du 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 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	Fermetures d'événements OK. Valeurs possibles : 0 - *(par défaut)* Tous les problèmes ; 1 - Tous les problèmes si les valeurs des balises correspondent.
correlation_tag	string	Balise 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 du déclencheur. Comportement de la propriété: - *obligatoire*
value	string	Valeur de la balise du 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 de 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 disponible uniquement pour les types d'utilisateurs *Admin* et *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)` Déclencheurs à créer.

En plus des propriétés standard des déclencheurs, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
dependencies	array	Déclencheurs dont dépend le déclencheur. 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 permet de supprimer des déclencheurs.

Note:
Cette méthode est disponible uniquement pour les types d'utilisateurs *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

`(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)`

Cette méthode permet de récupérer des 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. 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
triggerids	ID/array	Retourner uniquement les déclencheurs avec les ID donnés.

Paramètre	Type	Description
groupids	ID/array	Retourner uniquement les déclencheurs qui appartiennent à des hôtes ou à des modèles issus des groupes d'hôtes ou groupes de modèles donnés.
templateids	ID/array	Retourner uniquement les déclencheurs qui appartiennent aux modèles donnés.
hostids	ID/array	Retourner uniquement les déclencheurs qui appartiennent aux hôtes donnés.
itemids	ID/array	Retourner uniquement les déclencheurs qui contiennent les éléments de données donnés.
functions	string/array	Retourner uniquement les déclencheurs qui utilisent les fonctions données. Consultez la page des fonctions prises en charge pour obtenir la liste des fonctions prises en charge.
group	string	Retourner uniquement les déclencheurs qui appartiennent à des hôtes ou à des modèles du groupe d'hôtes ou groupe de modèles portant le nom donné.
host	string	Retourner uniquement les déclencheurs qui appartiennent à l'hôte portant le nom technique donné.
inherited	boolean	Si défini sur `true`, retourner uniquement les déclencheurs hérités d'un modèle.
templated	boolean	Si défini sur `true`, retourner uniquement les déclencheurs qui appartiennent à des modèles.
dependent	boolean	Si défini sur `true`, retourner uniquement les déclencheurs qui ont des dépendances. Si défini sur `false`, retourner uniquement les déclencheurs qui n'ont pas de dépendances.
monitored	flag	Retourner uniquement les déclencheurs activés qui appartiennent à des hôtes surveillés et ne contiennent que des éléments de données activés.
active	flag	Retourner uniquement les déclencheurs activés qui appartiennent à des hôtes surveillés.
maintenance	boolean	Si défini sur `true`, retourner uniquement les déclencheurs activés qui appartiennent à des hôtes en maintenance.
withUnacknowledgedEvents	flag	Retourner uniquement les déclencheurs qui ont des événements non acquittés.
withAcknowledgedEvents	flag	Retourner uniquement les déclencheurs dont tous les événements sont acquittés.
withLastEventUnacknowledged	flag	Retourner uniquement les déclencheurs dont le dernier événement n'est pas acquitté.
skipDependent	flag	Ignorer les déclencheurs en état de problème qui dépendent d'autres déclencheurs. Notez que les autres déclencheurs sont ignorés s'ils sont désactivés, s'ils ont des éléments de données désactivés ou des hôtes d'éléments de données désactivés.
lastChangeSince	timestamp	Retourner uniquement les déclencheurs dont l'état a changé après l'heure donnée.
lastChangeTill	timestamp	Retourner uniquement les déclencheurs dont l'état a changé avant l'heure donnée.
only_true	flag	Retourner uniquement les déclencheurs qui ont récemment été dans un état de problème.
min_severity	integer	Retourner uniquement les déclencheurs dont la sévérité est supérieure ou égale à la sévérité donnée.
evaltype	integer	Méthode d'évaluation des balises. Valeurs possibles : 0 - *(par défaut)* Et/Ou ; 2 - Ou.

Paramètre	Type	Description
tags	array	Retourner uniquement les déclencheurs avec les balises données. Format : `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Un tableau vide retourne tous les déclencheurs. Valeurs possibles de operator : 0 - *(par défaut)* Contient ; 1 - Égal à ; 2 - Ne contient pas ; 3 - Différent de ; 4 - Existe ; 5 - N'existe pas.
expandComment	flag	Développer les macros dans la description du déclencheur.
expandDescription	flag	Développer les macros dans le nom du déclencheur.
expandExpression	flag	Développer les fonctions et les macros dans l'expression du déclencheur.
selectHostGroups	query	Retourner les groupes d'hôtes auxquels le déclencheur appartient dans la propriété `hostgroups`.
selectHosts	query	Retourner les hôtes auxquels le déclencheur appartient dans la propriété `hosts`.
selectItems	query	Retourner les éléments de données contenus dans le déclencheur dans la propriété `items`.
selectFunctions	query	Retourner les fonctions utilisées dans le déclencheur dans la propriété `functions`. Les objets de fonction représentent les fonctions utilisées dans l'expression du déclencheur et ont les propriétés suivantes : `functionid` - `(ID)` ID de la fonction ; `itemid` - `(ID)` ID de l'élément de données 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	Retourner les déclencheurs dont le déclencheur dépend dans la propriété `dependencies`.
selectDiscoveryData	query	Retourner une propriété `discoveryData` avec les données de l'objet de découverte du déclencheur. L'objet de découverte du déclencheur relie un 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 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 précisant 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	Retourner la règle de découverte de bas niveau qui a créé le déclencheur dans la propriété `discoveryRule`.
selectLastEvent	query	Retourner le dernier événement significatif du déclencheur dans la propriété `lastEvent`.
selectTags	query	Retourner les balises du déclencheur dans la propriété `tags`.

Paramètre	Type	Description
inheritedTags	boolean	Retourner les déclencheurs 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 balises données ; `false` - *(par défaut)* les balises du modèle/de l'hôte/des modèles liés sont ignorées.
selectInheritedTags	query	Retourner une propriété `inheritedTags` avec les balises héritées des modèles, des modèles liés et des hôtes, ainsi que des éléments de données référencés dans les expressions de déclencheur ou de récupération.
selectTemplateGroups	query	Retourner les groupes de modèles auxquels le déclencheur appartient 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 le déclencheur appartient ; `hostid` - ID de l'hôte auquel le déclencheur appartient.
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`, `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	Retourner l'objet de découverte du déclencheur dans la propriété `triggerDiscovery`. Les objets de découverte du déclencheur relient le déclencheur à un 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 permet de mettre à jour des déclencheurs existants.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateurs *Admin* et *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 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.

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 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 les connexions TLS avec PSK.

Propriété	Type	Description
tls_psk_identity	string	Identité PSK ; doit être associée à une seule PSK (parmi 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(paramètres d'objet)`

La méthode permet de récupérer l'objet d'enregistrement automatique en fonction des paramètres donnés.

> **Note:**
> Cette méthode n'est disponible que pour le type d'utilisateur *Super admin*. Les autorisations d'appeler la méthode peuvent être révoquées dans les paramètres de rôle utilisateur. Voir 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)` 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

`objet autoregistration.update (objet enregistrement automatique)`

Cette méthode permet de mettre à jour l'auto-enregistrement existant.

> **Note:**
> Cette méthode n'est disponible que pour le type d'utilisateur *Super admin*. Les autorisations d'appeler la méthode peuvent être révoquées dans les paramètres de rôle utilisateur. Voir Rôles Utilisateur 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.

Propriété	Type	Description
regexpid	ID	ID de l'expression régulière. Comportement de la propriété : - *lecture seule* - *obligatoire* pour les opérations de mise à jour
name	string	Nom de l'expression régulière. Comportement de la propriété : - *obligatoire* pour les opérations de création
test_string	string	Chaîne de test.

Expressions

L'objet expressions possède les propriétés suivantes.

Propriété	Type	Description
expression	string	Expression régulière. Comportement de la propriété: - *obligatoire*
expression_type	integer	Type d'expression régulière. Valeurs possibles : 0 - Chaîne de caractères incluse ; 1 - Toute chaîne de caractères incluse ; 2 - Chaîne de caractères non incluse ; 3 - Le résultat est TRUE ; 4 - Le résultat est FALSE. Comportement de la propriété: - *obligatoire*
exp_delimiter	string	Délimiteur d'expression. Valeur par défaut : `","`. Valeurs possibles : `","` ou `"."`, ou `"/"`. Comportement de la propriété: - *pris en charge* si `expression_type` est défini sur "Any character string included"
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 permet de créer de nouvelles expressions régulières globales.

> **Note:**
> Cette méthode est uniquement disponible pour les types d'utilisateurs *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)` 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.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "regexp.create",
    "params": {
      "name": "Storage devices for SNMP discovery",
      "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": [
            "16"
        ]
    },
    "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 du rôle 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)`

La méthode permet de récupérer les expressions régulières globales selon les paramètres donnés.

> **Note:**
> Cette méthode est disponible uniquement pour le *Super Admin*. Les permissions d'appeler 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
regexpids	ID/array	Retourner uniquement les expressions régulières avec les ID indiqués.

Paramètre	Type	Description
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"],
        "selectExpressions": ["expression", "expression_type"],
        "regexpids": [1, 2],
        "preservekeys": true
    },
    "id": 1
}
```

Réponse :

```
{
  "jsonrpc": "2.0",
  "result": {
    "1": {
      "regexpid": "1",
      "name": "File systems for discovery",
      "expressions": [
        {
          "expression": "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|
          "expression_type": "3"
        }
      ]
    },
    "2": {
      "regexpid": "2",
      "name": "Network interfaces for discovery",
      "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 permet de mettre à jour des expressions régulières globales existantes.

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 du rôle utilisateur. Voir 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.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.update",
    "params": {
      "regexpid": "1",
      "name": "File systems for discovery",
      "test_string": "",
      "expressions": [
        {
          "expression": "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|zfs)$",
          "expression_type": "3",
          "exp_delimiter": ",",
          "case_sensitive": "0"
        },
        {
          "expression": "^(ntfs|fat32|fat16)$",
          "expression_type": "3",
          "exp_delimiter": ",",
          "case_sensitive": "0"
        }
      ]
    },
    "id": 1
}
```

Réponse :

```
{
    "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)* afficher en 2D ; 1 - afficher 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 rendu 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 rendu 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 universel unique, 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 n'est disponible que pour les types d'utilisateurs *Admin* et *Super admin*. Les autorisations permettant d'appeler cette méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir User roles 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 également

- Élément de graphique

Source

CGraph::create() dans *ui/include/classes/api/services/CGraph.php*.

graph.delete

Description

`object graph.delete(array graphIds)`

Cette méthode permet de supprimer des graphiques.

> **Note:**
> Cette méthode est disponible uniquement pour les utilisateurs de type *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les réglages du rôle d'utilisateur. Voir les 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)`

La méthode permet de récupérer des graphiques selon les paramètres donnés.

> **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. 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
graphids	ID/array	Retourner uniquement les graphiques avec les ID donnés.
groupids	ID/array	Retourner uniquement les graphiques qui appartiennent à des hôtes ou à des modèles dans les 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)` statut 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 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 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'utilisateurs *Admin* et *Super admin*. Les autorisations permettant d'appeler cette méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Voir User roles 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'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.

Propriété	Type	Description
usrgrpid	ID	ID du groupe d'utilisateurs. Comportement de la propriété : - *lecture seule* - *obligatoire* pour les opérations de mise à jour
name	string	Nom du groupe d'utilisateurs. Comportement de la propriété : - *obligatoire* 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.
gui_access	integer	Méthode d'authentification du frontend 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 au frontend.
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).

Propriété	Type	Description
mfaid	ID	Méthode MFA utilisée pour les utilisateurs du groupe. Comportement de la propriété : - *pris en charge* pour les utilisateurs de type *Super admin* - *pris en charge* si `mfa_status` de l'objet d'authentification est défini sur "Enabled"
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 admin* - *pris en charge* si `gui_access` est défini sur "use the system default authentication method" ou "use LDAP authentication"

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.

usergroup.create

Description

`object usergroup.create(object/array userGroups)`

Cette méthode 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. Voir User roles 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.

Paramètre	Type	Description
hostgroup_rights	object/array	Autorisations de groupes d'hôtes à attribuer au groupe d'utilisateurs.
templategroup_rights	object/array	Autorisations de groupes de modèles à attribuer au groupe d'utilisateurs.
tag_filters	array	Autorisations basées sur les tags à attribuer au groupe d'utilisateurs.
users	object/array	Utilisateurs à ajouter au groupe d'utilisateurs. L'utilisateur doit avoir uniquement 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 des utilisateurs déprovisionnés (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 User roles 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 donnés.

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.

Paramètre	Type	Description
mfaids	ID/array	Retourner uniquement les groupes d'utilisateurs avec les méthodes MFA indiquées. Comportement du paramètre : - *pris en charge* pour les utilisateurs de type *Super admin*
mfa_status	integer	Retourner uniquement les groupes d'utilisateurs avec le statut MFA indiqué. Consultez la page des groupes d'utilisateurs pour obtenir la liste des statuts pris en charge.
status	integer	Retourner uniquement les groupes d'utilisateurs avec le statut indiqué. Consultez la page des groupes d'utilisateurs pour obtenir la liste des statuts pris en charge.
userids	ID/array	Retourner uniquement les groupes d'utilisateurs qui contiennent les utilisateurs indiqués.
usrgrpids	ID/array	Retourner uniquement les groupes d'utilisateurs avec les ID indiqués.
selectTagFilters	query	Retourner les autorisations des groupes d'utilisateurs basées sur les tags dans la propriété `tag_filters`.
selectUsers	query	Retourner les utilisateurs du groupe d'utilisateurs dans la propriété `users`. Voir `user.get` pour les restrictions selon le type d'utilisateur.
selectHostGroupRights	query	Retourner les autorisations des groupes 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.
selectTemplateGroupRights	query	Retourner les autorisations des groupes 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 renvoyés par les sous-sélections.
output	query	Propriétés de l'objet groupe d'utilisateurs à retourner. Les utilisateurs de type *Admin* et *User* ne peuvent récupérer que les propriétés suivantes : `usrgrpid`, `name`, `gui_access`, `users_status`, `debug_mode`, `mfa_status`. Par défaut : `extend`.
sortfield	string/array	Trier 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	

Paramètre	Type	Description
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 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 la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir User roles pour plus d'informations.

Paramètres

`(object/array)` Propriétés du groupe 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 du groupe d'utilisateurs, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
hostgroup_rights	object/array	Autorisations des groupes d'hôtes remplaçant les autorisations actuelles attribuées au groupe d'utilisateurs.
templategroup_rights	object/array	Autorisations des groupes de modèles remplaçant les autorisations actuelles attribuées au groupe d'utilisateurs.
tag_filters	array	Autorisations basées sur les tags remplaçant les autorisations actuelles attribuées au groupe d'utilisateurs.
users	object/array	Utilisateurs remplaçant les utilisateurs actuellement attribués 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 autorisations 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 universel unique, 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 n'est disponible que 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)` 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. Voir 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)`

Cette méthode permet de récupérer les groupes de modèles selon les paramètres donnés.

> **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.

Paramètre	Type	Description
graphids	ID/array	Retourner uniquement les groupes de modèles qui contiennent des modèles avec les graphiques donnés.
groupids	ID/array	Retourner uniquement les groupes de modèles avec les ID de groupes de modèles donnés.
templateids	ID/array	Retourner uniquement les groupes de modèles qui contiennent les modèles donnés.
triggerids	ID/array	Retourner uniquement les groupes de modèles qui contiennent des modèles avec les déclencheurs donnés.
with_graphs	boolean	Retourner uniquement les groupes de modèles qui contiennent des modèles avec des graphiques.
with_graph_prototypes	boolean	Retourner uniquement les groupes de modèles qui contiennent des modèles avec des prototypes de graphiques.
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 de données. 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 de données. 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 de données, 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 de données 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.

Paramètre	Type	Description
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 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	

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 liés à tous les groupes de modèles donnés.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateurs *Admin* et *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 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 permet de supprimer des objets liés de plusieurs groupes de modèles.

Note:
Cette méthode est disponible uniquement pour les types d'utilisateur *Admin* et *Super admin*. Les autorisations permettant d'appeler la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir User roles 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*

Paramètre	Type	Description
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 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 permettant d'appeler cette 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 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 autorisations. 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 permet de mettre à jour des groupes de modèles existants.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateurs *Admin* et *Super admin*. Les autorisations permettant d'appeler la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Voir 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 proxys

Cette classe est conçue pour fonctionner avec des groupes de proxys.

Références d'objet :

- Groupe de proxys

Méthodes disponibles :

- proxygroup.create - créer de nouveaux groupes de proxys
- proxygroup.delete - supprimer des groupes de proxys
- proxygroup.get - récupérer des groupes de proxys
- proxygroup.update - mettre à jour des groupes de proxys

Objet groupe de proxys

Les objets suivants sont directement liés à l'API `proxygroup`.

Groupe de proxys

L'objet groupe de proxys possède les propriétés suivantes.

Propriété	Type	Description
proxy_groupid	ID	ID du groupe de proxys. Comportement de la propriété : - *lecture seule* - *obligatoire* pour les opérations de mise à jour
name	string	Nom du groupe de proxys. Comportement de la propriété : - *obligatoire* pour les opérations de création
description	text	Description du groupe de proxys.
failover_delay	string	Période pendant laquelle un proxy du groupe de proxys doit communiquer avec le serveur Zabbix pour être considéré comme 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. Par défaut : 1m.
min_online	string	Nombre minimal de proxys en ligne requis pour maintenir le groupe de proxys en ligne. Les macros utilisateur sont prises en charge. Plage de valeurs possibles : 1-1000. Par défaut : 1.
state	integer	État du groupe de proxys. Valeurs possibles : 0 - Inconnu ; 1 - Hors ligne ; 2 - Récupération ; 3 - En ligne ; 4 - Dégradation. Comportement de la propriété : - *lecture seule*

proxygroup.create

Description

`object proxygroup.create(object/array proxyGroups)`

Cette méthode permet de créer de nouveaux groupes de proxys.

> **Note:**
> Cette méthode est uniquement disponible 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) Groupes de proxys à créer.

La méthode accepte des groupes de proxys avec les propriétés standard des groupes de proxys.

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 permet de supprimer des groupes de proxys.

> **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

(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)`

La méthode permet de récupérer des groupes de proxys 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 User roles pour plus d'informations.

Paramètres

`(object)` Paramètres définissant la sortie souhaitée.

La méthode prend en charge les paramètres suivants.

Paramètre	Type	Description
proxy_groupids	ID/array	Retourner uniquement les groupes de proxys avec les ID indiqués.
proxyids	ID/array	Retourner uniquement les groupes de proxys qui contiennent les proxys indiqués.
selectProxies	query	Retourner une propriété `proxies` avec les proxys appartenant au groupe de proxys. Prend en charge `count`.
sortfield	string/array	Trier le résultat selon les propriétés indiquées. 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	

Paramètre	Type	Description
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 permet de mettre à jour des groupes de proxys 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 du groupe de proxys à mettre à jour.

La propriété `proxy_groupid` doit être définie pour chaque groupe de proxys, 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 proxys avec les propriétés standard des groupes de proxys.

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*.

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 aux groupes d'hôtes
- hostgroup.massremove - supprimer des objets associés des 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 universel unique, 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 permettant d'appeler la méthode peuvent être révoquées dans les paramètres du rôle 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 permet de supprimer des groupes d'hôtes.

Un groupe d'hôtes ne peut pas être supprimé si :

- il contient des hôtes qui n'appartiennent qu'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 des rôles utilisateur. Voir Rôles 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 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
graphids	ID/array	Ne renvoyer que les groupes d'hôtes qui contiennent des hôtes avec les graphiques donnés.
groupids	ID/array	Ne renvoyer que les groupes d'hôtes avec les ID de groupes d'hôtes donnés.
hostids	ID/array	Ne renvoyer que les groupes d'hôtes qui contiennent les hôtes donnés.
maintenanceids	ID/array	Ne renvoyer que les groupes d'hôtes affectés par les maintenances données.

Paramètre	Type	Description
triggerids	ID/array	Ne renvoyer que les groupes d'hôtes qui contiennent des hôtes avec les déclencheurs donnés.
with_graphs	boolean	Ne renvoyer que les groupes d'hôtes qui contiennent des hôtes avec des graphiques.
with_graph_prototypes	boolean	Ne renvoyer que les groupes d'hôtes qui contiennent des hôtes avec des prototypes de graphiques.
with_hosts	boolean	Ne renvoyer que les groupes d'hôtes qui contiennent des hôtes.
with_httptests	boolean	Ne renvoyer que les groupes d'hôtes qui contiennent des hôtes avec des scénarios web. Remplace le paramètre `with_monitored_httptests`.
with_items	boolean	Ne renvoyer que les groupes d'hôtes qui contiennent des hôtes avec des éléments de données. Remplace les paramètres `with_monitored_items` et `with_simple_graph_items`.
with_item_prototypes	boolean	Ne renvoyer que les groupes d'hôtes qui contiennent des hôtes avec des prototypes d'éléments de données. Remplace le paramètre `with_simple_graph_item_prototypes`.
with_simple_graph_item_prototypes	boolean	Ne renvoyer que les groupes d'hôtes qui contiennent des hôtes avec des prototypes d'éléments de données, activés pour la création et ayant un type d'information numérique.
with_monitored_httptests	boolean	Ne renvoyer que les groupes d'hôtes qui contiennent des hôtes avec des scénarios web activés.
with_monitored_hosts	boolean	Ne renvoyer que les groupes d'hôtes qui contiennent des hôtes surveillés.
with_monitored_items	boolean	Ne renvoyer que les groupes d'hôtes qui contiennent des hôtes avec des éléments de données activés. Remplace le paramètre `with_simple_graph_items`.
with_monitored_triggers	boolean	Ne renvoyer que les groupes d'hôtes qui contiennent des hôtes avec des déclencheurs activés. Tous les éléments de données utilisés dans le déclencheur doivent également être activés.
with_simple_graph_items	boolean	Ne renvoyer que les groupes d'hôtes qui contiennent des hôtes avec des éléments de données numériques.
with_triggers	boolean	Ne renvoyer que les groupes d'hôtes qui contiennent des hôtes avec des déclencheurs. Remplace le paramètre `with_monitored_triggers`.
selectDiscoveryRules	query	Renvoyer une propriété `discoveryRules` avec les règles LLD qui ont découvert le groupe d'hôtes.
selectDiscoveryData	query	Renvoyer une propriété `discoveryData` avec les objets de découverte du groupe 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	Renvoyer une propriété `hostPrototypes` avec les prototypes d'hôtes qui ont découvert ce groupe d'hôtes.
selectHosts	query	Renvoyer une propriété `hosts` avec les hôtes qui appartiennent au groupe d'hôtes. Prend en charge `count`.

Paramètre	Type	Description
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	Renvoyer une propriété `groupDiscoveries` avec les objets de découverte du groupe 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 liés à tous les groupes d'hôtes 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 User roles 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 permet de supprimer des objets liés de plusieurs groupes d'hôtes.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateurs *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)` 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 des paramètres : - *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 permet d'appliquer les autorisations et les filtres de tags à tous les sous-groupes d'un groupe d'hôtes.

> **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 du rôle 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
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 balises. 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 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 permettant d'appeler la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Propriétés des groupes d'hôtes à mettre à jour.

La propriété `groupeid` doit être définie pour chaque groupe d'hôtes. Toutes les autres propriétés sont facultatives. Seules les propriétés transmises 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*.

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.

Propriété	Type	Description
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.
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.

Propriété	Type	Description
value	text	Valeur reçue.

history.clear

Description

`object history.clear(array itemids)`

Cette méthode permet d'effacer l'historique des éléments.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateurs *Admin* et *Super admin*. Les autorisations permettant d'appeler cette méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Consultez User roles 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)`

Cette 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'appeler la méthode peuvent être révoquées dans les paramètres des rôles 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
history	integer	Types d'objets d'historique à renvoyer. Valeurs possibles : 0 - numérique à virgule flottante ; 1 - caractère ; 2 - journal ; 3 - *(par défaut)* numérique non signé ; 4 - texte ; 5 - binaire ; 6 - JSON.
hostids	ID/array	Renvoyer uniquement l'historique des hôtes indiqués.
itemids	ID/array	Renvoyer uniquement l'historique des éléments indiqués.
maxValueSize	integer	Nombre maximal d'octets à renvoyer dans la propriété `value`. Plage de valeurs possibles : 1-134217728 (de 1 octet à 128 Mio). Si défini sur `null`, aucune limite n'est appliquée. Par défaut : 65536 (64 Kio). Comportement du paramètre : - *pris en charge* si `history` est défini sur "binary" ou "JSON"
time_from	timestamp	Renvoyer uniquement les valeurs reçues après ou à l'heure indiquée.
time_till	timestamp	Renvoyer uniquement les valeurs reçues avant ou à l'heure indiquée.
sortfield	string/array	Trier le résultat selon les propriétés indiquées. Valeurs possibles : `itemid`, `clock`, `ns`.
search	object	Renvoyer les résultats correspondant au motif indiqué (insensible à la casse). Accepte un objet dont les clés sont des noms de propriétés et les valeurs 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 d'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

Paramètre	Type	Description
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 liés aux hôtes
- host.massremove - supprimer des objets liés 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.

Propriété	Type	Description
hostid	ID	ID de l'hôte. Comportement de la propriété : - *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 - un hôte simple ; 4 - un hôte converti à partir d'un prototype. Comportement de la propriété : - *lecture seule*

Propriété	Type	Description
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 - callback ; 2 - *(par défaut)* utilisateur ; 3 - opérateur ; 4 - admin ; 5 - OEM.
ipmi_username	string	Nom d'utilisateur IPMI.
maintenance_from	timestamp	Heure de début de la maintenance effective. Comportement de la propriété : - *lecture seule*
maintenance_status	integer	Statut de la maintenance effective. Valeurs possibles : 0 - *(par défaut)* aucune maintenance ; 1 - maintenance en cours. Comportement de la propriété : - *lecture seule*
maintenance_type	integer	Type de maintenance effective. Valeurs possibles : 0 - *(par défaut)* maintenance avec collecte de données ; 1 - maintenance sans collecte de données. Comportement de la propriété : - *lecture seule*
maintenanceid	ID	ID de la maintenance actuellement en vigueur sur l'hôte. Comportement de la propriété : - *lecture seule*
name	string	Nom visible de l'hôte. Par défaut : valeur de la propriété `host`.
monitored_by	integer	Source utilisée pour surveiller l'hôte. Valeurs possibles : 0 - *(par défaut)* serveur Zabbix ; 1 - Proxy ; 2 - Groupe de proxys.

Propriété	Type	Description
proxyid	ID	ID du proxy utilisé pour surveiller l'hôte. Comportement de la propriété : - *obligatoire* si `monitored_by` est défini sur "Proxy"
proxy_groupid	ID	ID du groupe de proxys utilisé pour surveiller l'hôte. Comportement de la propriété : - *obligatoire* si `monitored_by` est défini sur "Proxy group"
status	integer	Statut 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)* Pas de chiffrement ; 2 - PSK ; 4 - certificat.
tls_accept	integer	Connexions depuis l'hôte. Valeurs bitmap possibles : 1 - *(par défaut)* Pas de chiffrement ; 2 - PSK ; 4 - certificat. Il s'agit d'un champ de masque de bits ; toute somme des 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 (dans autoregistration, hosts et proxies). N'incluez pas d'informations sensibles dans l'identité PSK, car elle est envoyée sans chiffrement sur le réseau pour indiquer au destinataire quelle PSK utiliser. Comportement de la propriété : - *écriture seule* - *obligatoire* si `tls_connect` est défini sur "PSK", ou si `tls_accept` contient le bit "PSK"
tls_psk	string	Clé pré-partagée (PSK) ; doit comporter au moins 32 chiffres hexadécimaux. Comportement de la propriété : - *écriture seule* - *obligatoire* si `tls_connect` est défini sur "PSK", ou si `tls_accept` contient le bit "PSK"
active_available	integer	Statut de disponibilité de l'interface active de l'hôte. Valeurs possibles : 0 - le statut de l'interface est inconnu ; 1 - l'interface est disponible ; 2 - l'interface n'est pas disponible. Comportement de la propriété : - *lecture seule*

Propriété	Type	Description
assigned_proxyid	ID	ID du proxy attribué par le serveur Zabbix, si l'hôte est surveillé par un groupe de proxys. Comportement de la propriété : *- 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, qui est 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
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	Adresse e-mail principale.	128 caractères
57	poc_1_name	string	Nom du contact principal.	128 caractères
63	poc_1_notes	string	Notes sur le 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

ID	Propriété	Type	Description	Longueur maximale
68	poc_2_cell	string	Numéro de mobile du contact secondaire.	64 caractères
65	poc_2_email	string	Adresse e-mail secondaire.	128 caractères
64	poc_2_name	string	Nom du contact secondaire.	128 caractères
70	poc_2_notes	string	Notes sur le 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 sur le 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
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

Tag d'hôte

L'objet de tag d'hôte possède les propriétés suivantes.

Propriété	Type	Description
tag	string	Nom du tag d'hôte. Comportement de la propriété : - *obligatoire*
value	string	Valeur du tag d'hôte. Comportement de la propriété : - *lecture seule*.
automatic	integer	Type de tag d'hôte. Valeurs possibles : 0 - *(par défaut)* manuel (tag créé par l'utilisateur) ; 1 - automatique (tag créé par la découverte de bas niveau) Comportement de la propriété : - *lecture seule*

Propriété	Type	Description
object	integer	Type d'objet dont le tag a été hérité. Valeurs possibles : 0 - modèle. Comportement de la propriété : - *lecture seule*.
objectid	ID	ID de l'objet dont le tag a été hérité. Comportement de la propriété : - *lecture seule*.

host.create

Description

`object host.create(object/array hosts)`

Cette méthode 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 du rôle utilisateur. Voir User roles pour plus d'informations.

Paramètres

`(object/array)` Hôtes à créer.

En plus des propriétés d'hôte standard, 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 nommé « 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 proxys

Source

CHost::create() dans *ui/include/classes/api/services/CHost.php*.

host.delete

Description

`object host.delete(array hosts)`

Cette méthode permet de supprimer des hôtes.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateurs *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

`(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)`

La méthode permet de récupérer des hôtes selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations permettant d'appeler la méthode peuvent être révoquées dans les paramètres des rôles 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
groupids	ID/array	Renvoie uniquement les hôtes appartenant aux groupes indiqués.
dserviceids	ID/array	Renvoie uniquement les hôtes liés aux services découverts indiqués.
graphids	ID/array	Renvoie uniquement les hôtes possédant les graphiques indiqués.
hostids	ID/array	Renvoie uniquement les hôtes ayant les ID d'hôte indiqués.
httptestids	ID/array	Renvoie uniquement les hôtes possédant les contrôles web indiqués.
interfaceids	ID/array	Renvoie uniquement les hôtes utilisant les interfaces indiquées.
itemids	ID/array	Renvoie uniquement les hôtes possédant les éléments de données indiqués.
maintenanceids	ID/array	Renvoie uniquement les hôtes affectés par les maintenances indiquées.
monitored_hosts	flag	Renvoie uniquement les hôtes supervisés.
proxyids	ID/array	Renvoie uniquement les hôtes supervisés par les proxies indiqués.
proxy_groupids	ID/array	Renvoie uniquement les hôtes supervisés 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 possédant les déclencheurs indiqués.

Parameter	Type	Description
with_items	flag	Renvoie uniquement les hôtes possédant des éléments de données. Remplace les paramètres `with_monitored_items` et `with_simple_graph_items`.
with_item_prototypes	flag	Renvoie uniquement les hôtes possédant des prototypes d'élément de données. Remplace le paramètre `with_simple_graph_item_prototypes`.
with_simple_graph_item_prototypes	flag	Renvoie uniquement les hôtes possédant des prototypes d'élément de données activés pour la création et ayant un type d'information numérique.
with_graphs	flag	Renvoie uniquement les hôtes possédant des graphiques.
with_graph_prototypes	flag	Renvoie uniquement les hôtes possédant des prototypes de graphique.
with_httptests	flag	Renvoie uniquement les hôtes possédant des contrôles web. Remplace le paramètre `with_monitored_httptests`.
with_monitored_httptests	flag	Renvoie uniquement les hôtes possédant des contrôles web activés.
with_monitored_items	flag	Renvoie uniquement les hôtes possédant des éléments de données activés. Remplace le paramètre `with_simple_graph_items`.
with_monitored_triggers	flag	Renvoie uniquement les hôtes possédant des déclencheurs activés. Tous les éléments de données utilisés dans le déclencheur doivent également être activés.
with_simple_graph_items	flag	Renvoie uniquement les hôtes possédant des éléments de données avec un type d'information numérique.
with_triggers	flag	Renvoie uniquement les hôtes possédant des déclencheurs. Remplace le paramètre `with_monitored_triggers`.
withProblemsSuppressed	boolean	Si défini sur `true`, renvoie uniquement les hôtes avec des problèmes supprimés.
evaltype	integer	Méthode d'évaluation des tags. Valeurs possibles : 0 - *(par défaut)* Et/Ou ; 2 - Ou.
severities	integer/array	Renvoie les hôtes ayant 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 l'opérateur : 0 - *(par défaut)* Contient ; 1 - Égal ; 2 - Ne contient pas ; 3 - Différent de ; 4 - Existe ; 5 - N'existe pas. Consultez le paramètre *Tags* pour des informations détaillées sur les opérateurs.
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.

Parameter	Type	Description
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)` statut 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`.
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 données 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.

Parameter	Type	Description
filter	object	Renvoie uniquement les résultats correspondant exactement au filtre indiqué. Accepte un objet dont 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 : propriétés de Host interface.
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`.
search	object	Renvoie les résultats correspondant au motif indiqué (insensible à la casse). Accepte un objet dont 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 Host interface.
searchInventory	object	Renvoie les hôtes dont les données d'inventaire correspondent au motif indiqué (insensible à la casse). Accepte un objet dont 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.

Parameter	Type	Description
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
}
```

Retrieving host items

Retrieve items for the "Zabbix server" host. The request is limited to the item properties `itemid`, `name`, and `status`. Due to the large response size, only a subset of items is shown in the example.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectItems": ["itemid","name","status"],
        "filter": {
            "host": [
                "Zabbix server"
            ]
        }
    },
    "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "hostid": "10084",
      "items": [
        {
          "itemid": "42227",
          "name": "Host name of Zabbix agent running",
          "status": "0"
        },
        {
          "itemid": "42237",
          "name": "Zabbix agent ping",
          "status": "0"
        },
        {
          "itemid": "42250",
          "name": "Version of Zabbix agent running",
          "status": "0"
        },
        {
          "itemid": "42251",
          "name": "Maximum number of open file descriptors",
          "status": "0"
        },
        {
          "itemid": "42252",
          "name": "Maximum number of processes",
          "status": "0"
```

```
        },
        {
          "itemid": "69869",
          "name": "Interface enp0s3: Inbound packets discarded",
          "status": "0"
        },
        {
          "itemid": "69870",
          "name": "Interface enp0s3: Inbound packets with errors",
          "status": "0"
        },
        {
          "itemid": "69871",
          "name": "Interface enp0s3: Bits received",
          "status": "0"
        },
        {
          "itemid": "69872",
          "name": "Interface enp0s3: Outbound packets discarded",
          "status": "0"
        },
        {
          "itemid": "69873",
          "name": "Interface enp0s3: Outbound packets with errors",
          "status": "0"
        },
        {
          "itemid": "69874",
          "name": "Interface enp0s3: Bits sent",
          "status": "0"
        },
        {
          "itemid": "42253",
          "name": "Number of processes",
          "status": "0"
        },
        {
          "itemid": "42254",
          "name": "Number of running processes",
          "status": "0"
        },
        {
          "itemid": "42255",
          "name": "System boot time",
          "status": "0"
        }
      ]
    }
  ],
  "id": 1
}
```


Voir aussi

- Groupe d’hôtes
- Modèle
- Macro utilisateur
- Interface d’hôte
- Proxy
- Groupe de proxys

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 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 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 également

- host.update
- Group d'hôte
- Modèle
- Macro utilisateur
- Interface de l'hôte

Source

CHost::massAdd() dans *ui/include/classes/api/services/CHost.php*.

host.massremove

Description

`object host.massremove(object parameters)`

Cette méthode 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. Voir User roles 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 du paramètre : - *obligatoire*
groupids	ID/array	ID des groupes d'hôtes desquels supprimer les hôtes indiqués.
interfaces	object/array	Interfaces d'hôte à supprimer des hôtes indiqués. 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 indiqués.
templateids	ID/array	ID des modèles à dissocier des hôtes indiqués.
templateids_clear	ID/array	ID des modèles à dissocier et à supprimer des hôtes indiqués.

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 également

- host.update
- Macro utilisateur
- Interface de l'hôte

Source

CHost::massRemove() dans *ui/include/classes/api/services/CHost.php*.

host.update

Description

`object host.update(object/array hosts)`

Cette méthode 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 permettant d'appeler cette méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir User roles 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 indiquées 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 indiqué séparément) avec la valeur du nom technique de l'hôte.

En plus des propriétés d'hôte standard, la méthode accepte les paramètres suivants.

Paramètre	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 d'hôte pour remplacer les interfaces d'hôte actuelles. Toutes les interfaces qui ne sont pas listées dans la requête seront supprimées.
tags	object/array	Tags d'hôte pour remplacer les tags d'hôte actuels. Tous les tags qui ne sont pas listés dans la requête seront supprimés.
inventory	object	Propriétés de l'inventaire de l'hôte.
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 à 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

Mettez à jour l'hôte « 10590 » afin d'utiliser uniquement le chiffrement PSK pour les connexions de l'hôte vers le serveur Zabbix, et modifiez 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 proxys

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 est redevenu disponible pour la dernière fois.
status	integer	Indique si l'hôte découvert est disponible ou indisponible. Un hôte est considéré comme 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)`

Cette méthode permet de récupérer les hôtes découverts selon les paramètres donnés.

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. 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
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	

Paramètre	Type	Description
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 également

- 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 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. Voir 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 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. Voir 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)`

La méthode permet de récupérer des images selon les paramètres fournis.

> **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. 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
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 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. Voir 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*.

Interface d'hôte

Cette classe est conçue pour fonctionner avec les interfaces d'hôte.

Références d'objet :

- Interface d'hôte
  - Détails

Méthodes disponibles :

- hostinterface.create - créer de nouvelles interfaces d'hôte
- hostinterface.delete - supprimer des interfaces d'hôte
- hostinterface.get - récupérer des interfaces d'hôte
- hostinterface.massadd - ajouter des interfaces d'hôte à des hôtes
- hostinterface.massremove - supprimer des interfaces d'hôte d'hôtes
- hostinterface.update - mettre à jour des interfaces d'hôte

Objet interface d'hôte

Les objets suivants sont directement liés à l'API `hostinterface`.

Interface d'hôte

L'objet d'interface d'hôte possède les propriétés suivantes.

> **Attention:**
> Notez que les propriétés `ip` et `dns` sont toutes deux *obligatoires* pour les opérations de création. Si vous ne souhaitez pas utiliser DNS, définissez-le comme une chaîne vide.

Propriété	Type	Description
interfaceid	ID	ID de l'interface. Comportement de la propriété : - *lecture seule* - *obligatoire* pour les opérations de mise à jour
available	integer	Disponibilité de l'interface d'hôte. Valeurs possibles : 0 - *(par défaut)* inconnue ; 1 - disponible ; 2 - indisponible. Comportement de la propriété : - *lecture seule*
hostid	ID	ID de l'hôte auquel l'interface appartient. Comportement de la propriété : - *constante* - *obligatoire* pour les opérations de création

Propriété	Type	Description
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
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* pour les opérations de création
main	integer	Indique si l'interface est utilisée comme interface par défaut sur l'hôte. Une seule interface d'un certain type peut être définie comme interface par défaut sur un hôte. Valeurs possibles : 0 - non 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	Heure du prochain sondage d'une interface d'hôte indisponible. Comportement de la propriété : - *lecture seule*
error	string	Texte d'erreur si l'interface d'hôte est indisponible. Comportement de la propriété : - *lecture seule*
errors_from	timestamp	Heure à laquelle l'interface d'hôte est devenue indisponible. Comportement de la propriété : - *lecture seule*

Détails

L'objet details possède les propriétés suivantes.

Property	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"
max_repetitions	integer	Valeur maximale de répétition 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.

Property	Type	Description
contextname	string	Nom de contexte SNMPv3. Utilisé uniquement par les interfaces SNMPv3.

hostinterface.create

Description

`object hostinterface.create(object/array hostInterfaces)`

Cette méthode permet de créer de nouvelles interfaces d'hôte.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateurs *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)` Interfaces d'hôte à créer.

La méthode accepte des interfaces d'hôte avec les propriétés standard des interfaces d'hôte.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des interfaces hôtes créées sous la propriété `interfaceids`. L'ordre des ID retournés correspond à l'ordre des interfaces hôtes transmises.

Exemples

Créer une nouvelle interface

Créez une interface d'agent IP secondaire sur l'hôte « 30052 ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.create",
    "params": {
        "hostid": "30052",
        "main": "0",
        "type": "1",
        "useip": "1",
        "ip": "127.0.0.1",
        "dns": "",
        "port": "10050"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30062"
        ]
    },
    "id": 1
}
```

Créer une interface avec les détails SNMP

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.create",
    "params": {
        "hostid": "10456",
        "main": "0",
        "type": "2",
        "useip": "1",
        "ip": "127.0.0.1",
        "dns": "",
        "port": "1601",
        "details": {
            "version": "2",
            "bulk": "1",
            "community": "{$SNMP_COMMUNITY}"
        }
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30063"
        ]
    },
    "id": 1
}
```

Voir également

- hostinterface.massadd
- host.massadd

Source

CHostInterface::create() dans *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.delete

Description

`object hostinterface.delete(array hostInterfaceIds)`

Cette méthode permet de supprimer des interfaces d'hôte.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateurs *Admin* et *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

`(array)` ID des interfaces d'hôte à supprimer.

Valeurs de retour

`(object)` Renvoie un objet contenant les ID des interfaces d'hôte supprimées dans la propriété `interfaceids`.

Exemples

Supprimer une interface d'hôte

Supprimez l'interface d'hôte avec l'ID 30062.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.delete",
    "params": [
        "30062"
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30062"
        ]
    },
    "id": 1
}
```

Voir aussi

- hostinterface.massremove
- host.massremove

Source

CHostInterface::delete() dans *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.get

Description

`integer/array hostinterface.get(object parameters)`

Cette méthode permet de récupérer les interfaces d'hôte selon les paramètres donnés.

> **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. 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
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 de données donnés.
triggerids	ID/array	Retourner uniquement les interfaces d'hôte utilisées par les éléments de données dans les déclencheurs donnés.
selectItems	query	Retourner une propriété `items` avec les éléments de données 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`.

Paramètre	Type	Description
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*.

hostinterface.massadd

Description

`object hostinterface.massadd(object parameters)`

Cette méthode permet d'ajouter simultanément des interfaces d'hôte à 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 des rôles utilisateur. Consultez User roles pour plus d'informations.

Paramètres

`(object)` Paramètres contenant les interfaces d'hôte à créer sur les hôtes donnés.

La méthode accepte les paramètres suivants.

Paramètre	Type	Description
interfaces	object/array	Interfaces d'hôte à créer sur les hôtes donnés. Comportement du paramètre: - *obligatoire*
hosts	object/array	Hôtes à mettre à jour. Les hôtes doivent avoir uniquement la propriété `hostid` définie. Comportement du paramètre: - *obligatoire*

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des interfaces hôtes créées sous la propriété `interfaceids`.

Exemples

Création d'interfaces

Créez une interface sur deux hôtes.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.massadd",
    "params": {
        "hosts": [
            {
                "hostid": "30050"
            },
            {
                "hostid": "30052"
            }
        ],
        "interfaces": {
            "dns": "",
            "ip": "127.0.0.1",
            "main": 0,
            "port": "10050",
            "type": 1,
            "useip": 1
        }
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30069",
            "30070"
        ]
    },
    "id": 1
}
```

Voir également

- hostinterface.create
- host.massadd
- Hôte

Source

CHostInterface::massAdd() dans *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.massremove

Description

`object hostinterface.massremove(object parameters)`

Cette méthode permet de supprimer des interfaces d'hôte des hôtes donnés.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateurs *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 hôtes à mettre à jour et les interfaces à supprimer.

Paramètre	Type	Description
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. Comportement du paramètre : - *obligatoire*
hostids	ID/array	ID des hôtes à mettre à jour. Comportement du paramètre : - *obligatoire*

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des interfaces hôtes supprimées sous la propriété `interfaceids`.

Exemples

Suppression d'interfaces

Supprimez l'interface SNMP "127.0.0.1" de deux hôtes.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.massremove",
    "params": {
        "hostids": [
            "30050",
            "30052"
        ],
        "interfaces": {
            "dns": "",
            "ip": "127.0.0.1",
            "port": "161"
        }
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30069",
            "30070"
        ]
    },
    "id": 1
}
```

Voir également

- hostinterface.delete
- host.massremove

Source

CHostInterface::massRemove() dans *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.update

Description

`object hostinterface.update(object/array hostInterfaces)`

Cette méthode permet de mettre à jour des interfaces d'hôte existantes.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateurs *Admin* et *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 de l'interface hôte à mettre à jour.

La propriété `interfaceid` doit être définie pour chaque interface hôte. Toutes les autres propriétés sont facultatives. Seules les propriétés données seront mises à jour, toutes les autres resteront inchangées.

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des interfaces hôtes mises à jour sous la propriété `interfaceids`.

Exemples

Modification du port d'une interface d'hôte

Modifiez le port d'une interface d'hôte.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.update",
    "params": {
        "interfaceid": "30048",
        "port": "10055"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30048"
        ]
    },
    "id": 1
}
```

Modification des ports de plusieurs interfaces d'hôte

Modifiez le port de plusieurs interfaces d'hôte.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.update",
    "params": [
        {
            "interfaceid": "30049",
            "port": "10055"
        },
        {
            "interfaceid": "30050",
            "port": "10055"
        },
        {
            "interfaceid": "30051",
            "port": "10055"
```

```
        }
    ],
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30049",
            "30050",
            "30051"
        ]
    },
    "id": 1
}
```

Source

CHostInterface::update() dans *ui/include/classes/api/services/CHostInterface.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 - récupère 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	ID utilisateur 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 - Graphe ; 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 graphe ; 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 proxys ; 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. 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
auditids	ID/array	Retourne uniquement le journal d'audit avec les ID donnés.
userids	ID/array	Retourne uniquement le journal d'audit créé par les utilisateurs donnés.
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 les 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 de 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 la 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 "Secret macro" - *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 une 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 manière 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 appartient la macro. 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 "Secret macro" - *obligatoire* pour les opérations de création

Configuration des macros

L'objet de configuration des macros définit la manière dont une macro est affichée dans le Host Wizard.

Propriété	Type	Description
type	integer	Type du champ de saisie de la macro. Valeurs possibles : 0 - La macro n'est pas utilisée dans le Host Wizard ; 1 - Zone de texte ; 2 - Liste ; 3 - Case à cocher. Comportement de la propriété : - *obligatoire*
label	string	Libellé du champ de saisie de la 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 la macro. Prend en charge le formatage Markdown.
priority	integer	Position du champ de saisie de la 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 du rôle utilisateur. Voir User roles pour plus d’informations.

Paramètres

`(object/array)` Macros hôtes à créer.

La méthode accepte les macros hôtes avec des propriétés standard de macro 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”.

Requête:

```
{
    "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 du rôle utilisateur. Voir User roles pour plus d'informations.

Paramètres

`(object/array)` Macros globales à créer

La méthode accepte les macros globales avec les propriétés de macros globales standard.

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".

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.createglobal",
    "params":  {
        "macro": "{$SNMP_COMMUNITY}",
        "value": "public"
    },
    "id": 1
}
```

Réponse :

```
{
    "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 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 du rôle utilisateur. Voir User roles 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 permet de supprimer des macros globales.

> **Note:**
> Cette méthode est uniquement disponible 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

`(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)`

Cette méthode permet de récupérer les macros d'hôte et globales selon les paramètres donnés.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appeler la méthode peuvent être révoquées dans les paramètres des rôles 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
globalmacro	flag	Renvoyer les macros globales au lieu des macros d'hôte.
globalmacroids	ID/array	Renvoyer uniquement les macros globales avec les ID indiqués.
groupids	ID/array	Renvoyer uniquement les macros d'hôte appartenant à des hôtes ou modèles issus des groupes d'hôtes ou groupes de modèles indiqués.
hostids	ID/array	Renvoyer uniquement les macros appartenant aux hôtes ou modèles indiqués.
hostmacroids	ID/array	Renvoyer uniquement les macros d'hôte avec les ID indiqués.
inherited	boolean	Si défini sur `true`, renvoyer uniquement les macros utilisateur de prototype 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 indiquées. Valeurs possibles : `macro`.
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)` 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 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 la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir User roles pour plus d'informations.

Paramètres

(object/array) Propriétés des macros 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 permettant d'appeler la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir User roles pour plus d'informations.

Paramètres

`(object/array)` Propriétés des macros globales) 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 de maintenance possède les propriétés suivantes.

Propriété	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 balises 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. 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 : - *obligatoire* 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 : - *obligatoire* 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 : - *obligatoire* si `timeperiod_type` est défini sur "monthly"

Tag de problème

L'objet de tag de problème est utilisé pour définir quels problèmes doivent être supprimés lorsque la maintenance entre en vigueur. Les tags ne peuvent être spécifiés que si `maintenance_type` de l'objet de maintenance est défini sur « with data collection ». Il possède les propriétés suivantes.

Propriété	Type	Description
tag	string	Nom du tag de problème. Comportement de la propriété : - *obligatoire*

Propriété	Type	Description
operator	integer	Opérateur de condition. Valeurs possibles : 0 - Égal à ; 2 - *(par défaut)* Contient.
value	string	Valeur du tag de problème.

maintenance.create

Description

`object maintenance.create(object/array maintenances)`

Cette méthode permet de créer de nouvelles maintenances.

Note:
Cette méthode est disponible uniquement pour les types d'utilisateurs *Admin* et *Super admin*. Les autorisations pour appeler la méthode peuvent être révoquées dans les paramètres de rôles utilisateur. Voir Rôles Utilisateur pour plus d'informations.

Paramètres

`(object/array)` Maintenances à créer.

En plus des propriétés standard de maintenance, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
groups	object/array	Groupes d'hôtes qui seront mis 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 qui seront mis en maintenance. Les hôtes doivent avoir uniquement la propriété `hostid` définie. Comportement du paramètre : - *obligatoire* si `groups` n'est pas défini
timeperiods	object/array	Périodes de temps de maintenance. Comportement du paramètre : - *obligatoire*
tags	object/array	Tags de problème. Définissez quels problèmes doivent être supprimés. Si aucun tag n'est fourni, tous les problèmes des hôtes en maintenance active seront supprimés. Comportement du paramètre : - *pris en charge* si `maintenance_type` de l'objet Maintenance est défini sur "with data collection"

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 période de maintenance

Créez une période de maintenance avec collecte de données pour le groupe d'hôtes ayant l'ID « 2 » et avec les tags de problème **service:mysqld** et **error**. Elle doit être active du 22.01.2013 au 22.01.2014, entrer en vigueur chaque dimanche à 18:00 et durer une heure.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.create",
    "params": {
        "name": "Sunday maintenance",
        "active_since": 1358844540,
        "active_till": 1390466940,
        "tags_evaltype": 0,
        "groups": [
            {"groupid": "2"}
        ],
        "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
}
```

Voir également

- Période

Source

CMaintenance::create() dans *ui/include/classes/api/services/CMaintenance.php*.

maintenance.delete

Description

`object maintenance.delete(array maintenanceIds)`

Cette méthode permet de supprimer des périodes de maintenance.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateurs *Admin* et *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 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 donnés.

> **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.

Paramètre	Type	Description
groupids	ID/array	Retourner uniquement les maintenances affectées aux groupes d'hôtes donnés.
hostids	ID/array	Retourner uniquement les maintenances affectées aux hôtes donnés.
maintenanceids	ID/array	Retourner uniquement les maintenances ayant les ID donnés.
selectHostGroups	query	Retourner une propriété `hostgroups` avec les groupes d'hôtes affectés à la maintenance.
selectHosts	query	Retourner une propriété `hosts` avec les hôtes affectés à la maintenance.
selectTags	query	Retourner une propriété `tags` avec les tags de problème de la maintenance.
selectTimeperiods	query	Retourner une propriété `timeperiods` avec les périodes de temps de la maintenance.
sortfield	string/array	Trier le résultat selon les propriétés données. 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, ainsi que les données sur les groupes d'hôtes assignés, les périodes de temps définies et les tags de problème.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.get",
    "params": {
        "output": "extend",
        "selectHostGroups": "extend",
        "selectTimeperiods": "extend",
        "selectTags": "extend"
    },
    "id": 1
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "maintenanceid": "3",
            "name": "Sunday maintenance",
```

```
            "maintenance_type": "0",
            "description": "",
            "active_since": "1358844540",
            "active_till": "1390466940",
            "tags_evaltype": "0",
            "hostgroups": [
                {
                    "groupid": "4",
                    "name": "Zabbix servers",
                    "flags": "0",
                    "uuid": "6f6799aa69e844b4b3918f779f2abf08"
                }
            ],
            "timeperiods": [
                {
                    "timeperiod_type": "3",
                    "every": "1",
                    "month": "0",
                    "dayofweek": "1",
                    "day": "0",
                    "start_time": "64800",
                    "period": "3600",
                    "start_date": "2147483647"
                }
            ],
            "tags": [
                {
                    "tag": "service",
                    "operator": "0",
                    "value": "mysqld",
                },
                {
                    "tag": "error",
                    "operator": "2",
                    "value": ""
                }
            ]
        }
    ],
    "id": 1
}
```


Voir également

- Hôte
- Groupe d'hôtes
- Période

Source

CMaintenance::get() dans *ui/include/classes/api/services/CMaintenance.php*.

maintenance.update

Description

`object maintenance.update(object/array maintenances)`

Cette méthode permet de mettre à jour des maintenances existantes.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateurs *Admin* et *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 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.

Paramètre	Type	Description
groups	object/array	Groupes d'hôtes pour remplacer les groupes actuels. 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 pour remplacer les hôtes actuels. Les hôtes doivent avoir uniquement la propriété hostid définie. Comportement du paramètre : - *obligatoire* si groups n'est pas défini
timeperiods	object/array	Périodes de temps de maintenance pour remplacer les périodes actuelles.
tags	object/array	Balises de problème pour remplacer les balises actuelles. Comportement du paramètre : - *pris en charge* si maintenance_type de l'objet Maintenance est défini sur "with data collection"

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 également

- Période

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
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 - 6 chiffres ; 8 - 8 chiffres. Comportement de la propriété : - *obligatoire* si `type` est défini sur "TOTP"

Propriété	Type	Description
api_hostname	string	Nom d'hôte de l'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 uniquement disponible 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)` 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 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 du rôle utilisateur. Voir 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 uniquement disponible 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
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 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

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 permet de mettre à jour des méthodes MFA existantes.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations permettant d'appeler la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir User roles 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.

Propriété	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

Propriété	Type	Description
id	string	ID unique du module tel que 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. Comportement de la propriété : - *obligatoire* pour les opérations de création
relative_path	string	Chemin vers le répertoire du module, relatif au répertoire du frontend Zabbix. Valeurs possibles : `widgets/*` - pour les modules de widgets intégrés ; `modules/*` - pour les modules tiers. Comportement de la propriété : - *obligatoire* pour les opérations de création
status	integer	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 permet d'installer de nouveaux modules de l'interface web.

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.

Attention:
Les fichiers du module doivent être décompressés manuellement dans les sous-répertoires corrects, 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 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.

Attention:
Les fichiers du module 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)`

La méthode permet de récupérer des modules selon les paramètres donnés.

> **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
moduleids	ID/array	Retourner uniquement les modules avec les ID indiqués.
sortfield	string/array	Trier le résultat selon les propriétés indiquées. 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	
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 permet de mettre à jour des modules existants.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations pour appeler la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir User roles pour plus d'informations.

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 d'hôte. Prend en charge le formatage Markdown.
uuid	string	Identifiant unique universel, 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 être soit définis tous les deux, soit 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 être soit définis tous les deux, soit 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 d'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 de modèle. Comportement de la propriété : - *obligatoire*
value	string	Valeur de la balise de modèle. Comportement de la propriété : - *lecture seule*.

Propriété	Type	Description
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'utilisateurs *Admin* et *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)` 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	Tags 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 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 des rôles 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)`

La 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. 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
templateids	ID/array	Retourner uniquement les modèles avec les ID de modèle indiqués.
groupids	ID/array	Retourner uniquement les modèles qui appartiennent aux groupes de modèles indiqués.
parentTemplateids	ID/array	Retourner uniquement les modèles auxquels le modèle indiqué est lié.
hostids	ID/array	Retourner uniquement les modèles qui sont liés aux hôtes/modèles indiqués.
graphids	ID/array	Retourner uniquement les modèles qui contiennent les graphiques indiqués.
itemids	ID/array	Retourner uniquement les modèles qui contiennent les éléments de données indiqués.
triggerids	ID/array	Retourner uniquement les modèles qui contiennent les déclencheurs indiqués.
with_items	flag	Retourner uniquement les modèles qui ont des éléments de données.
with_triggers	flag	Retourner uniquement les modèles qui ont des déclencheurs.
with_graphs	flag	Retourner uniquement les modèles qui ont des graphiques.
with_httptests	flag	Retourner uniquement les modèles qui ont des scénarios web.

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 modèles avec les tags indiqués. Format : `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Un tableau vide retourne 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	Retourner les modèles qui ont les `tags` indiqués également dans les modèles liés. Valeurs possibles : `true` - les modèles liés doivent également avoir les tags indiqués ; `false` - *(par défaut)* les tags des modèles liés sont ignorés.
selectInheritedTags	query	Retourner une propriété `inheritedTags` avec les tags présents sur les modèles liés.
selectTags	query	Retourner les tags du modèle dans la propriété `tags`.
selectDiscoveryRules	query	Retourner une propriété `discoveryRules` avec les règles LLD du modèle. Prend en charge `count`.
selectHosts	query	Retourner les hôtes liés au modèle dans la propriété `hosts`. Prend en charge `count`.
selectTemplateGroups	query	Retourner les groupes de modèles auxquels le modèle appartient dans la propriété `templategroups`.
selectTemplates	query	Retourner les modèles auxquels le modèle indiqué est lié dans la propriété `templates`. Prend en charge `count`.
selectParentTemplates	query	Retourner les modèles qui sont liés au modèle indiqué dans la propriété `parentTemplates`. Prend en charge `count`.
selectHttpTests	query	Retourner les scénarios web du modèle dans la propriété `httpTests`. Prend en charge `count`.
selectItems	query	Retourner les éléments de données du modèle dans la propriété `items`. Prend en charge `count`.
selectTriggers	query	Retourner les déclencheurs du modèle dans la propriété `triggers`. Prend en charge `count`.
selectGraphs	query	Retourner les graphiques du modèle dans la propriété `graphs`. Prend en charge `count`.
selectMacros	query	Retourner les macros du modèle dans la propriété `macros`.
selectDashboards	query	Retourner les tableaux de bord du modèle dans la propriété `dashboards`. Prend en charge `count`.
selectValueMaps	query	Retourner une propriété `valuemaps` avec les correspondances de valeurs du modèle.

Paramètre	Type	Description
limitSelects	integer	Limite le nombre d'enregistrements retournés 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	Trier 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	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectDiscoveries	query	Retourner 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 liés aux modèles donnés.

> **Note:**
> Cette méthode n'est disponible que 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 User roles pour plus d'informations.

Paramètres

`(object)` Paramètres contenant les ID des modèles à mettre à jour et les objets à ajouter aux modèles.

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 permet de supprimer des objets liés de plusieurs modèles.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateurs *Admin* et *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)` 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 du paramètre : - *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 à 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 également

- 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 permet de mettre à jour des modèles existants.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateurs *Admin* et *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 des modèles 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 à 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*.

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 automatique

L'objet des paramètres possède les propriétés suivantes.

Propriété	Type	Description
hk_events_mode	integer	Active le nettoyage automatique interne pour les événements et les alertes. Valeurs possibles : 0 - Désactiver ; 1 - *(par défaut)* Activer.
hk_events_trigger	string	Période de conservation des données des déclencheurs. Accepte les secondes et une unité de temps avec suffixe. Par défaut : 365d.
hk_events_service	string	Période de conservation des données des services. Accepte les secondes et une unité de temps avec suffixe. Par défaut : 1d.
hk_events_internal	string	Période de conservation des données internes. Accepte les secondes et une unité de temps avec suffixe. Par défaut : 1d.
hk_events_discovery	string	Période de conservation des données de découverte réseau. Accepte les secondes et une unité de temps avec suffixe. Par défaut : 1d.
hk_events_autoreg	string	Période de conservation des données d'auto-enregistrement. Accepte les secondes et une unité de temps avec suffixe. Par défaut : 1d.
hk_services_mode	integer	Active le nettoyage automatique interne pour les services. Valeurs possibles : 0 - Désactiver ; 1 - *(par défaut)* Activer.
hk_services	string	Période de conservation des données des services. Accepte les secondes et une unité de temps avec suffixe. Par défaut : 365d.
hk_audit_mode	integer	Active le nettoyage automatique interne pour l'audit. Valeurs possibles : 0 - Désactiver ; 1 - *(par défaut)* Activer.

Propriété	Type	Description
hk_audit	string	Période de conservation des données d'audit. Accepte les secondes et une unité de temps avec suffixe. Par défaut : 31d.
hk_sessions_mode	integer	Active le nettoyage automatique interne pour les sessions. Valeurs possibles : 0 - Désactiver ; 1 - *(par défaut)* Activer.
hk_sessions	string	Période de conservation des données des sessions. Accepte les secondes et une unité de temps avec suffixe. Par défaut : 31d.
hk_history_mode	integer	Active le nettoyage automatique interne pour l'historique. Valeurs possibles : 0 - Désactiver ; 1 - *(par défaut)* Activer.
hk_history_global	integer	Remplace la période d'historique de l'élément. Valeurs possibles : 0 - Ne pas remplacer ; 1 - *(par défaut)* Remplacer.
hk_history	string	Période de conservation des données d'historique. Accepte les secondes et une unité de temps avec suffixe. Par défaut : 31d.
hk_trends_mode	integer	Active le nettoyage automatique interne pour les tendances. Valeurs possibles : 0 - Désactiver ; 1 - *(par défaut)* Activer.
hk_trends_global	integer	Remplace la période de tendances de l'élément. Valeurs possibles : 0 - Ne pas remplacer ; 1 - *(par défaut)* Remplacer.
hk_trends	string	Période de conservation des données de tendances. Accepte les secondes et une unité de temps avec suffixe. Par défaut : 365d.
db_extension	string	Indicateur de configuration de l'extension de base de données. Si cet indicateur est défini sur "timescaledb", le serveur modifie son comportement pour le nettoyage automatique 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	Active la compression TimescaleDB pour l'historique et les tendances. Valeurs possibles : 0 - *(par défaut)* Désactivée ; 1 - Activée.

Propriété	Type	Description
compress_older	string	Compresse les enregistrements d'historique et de tendances plus anciens que la période spécifiée. Accepte les secondes et une unité de temps avec suffixe. Par défaut : 7d.

housekeeping.get

Description

`object housekeeping.get(object parameters)`

La méthode permet de récupérer l'objet de nettoyage interne selon les paramètres fournis.

> **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. Voir 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 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 permet de mettre à jour les paramètres de nettoyage existants.

> **Note:**
> Cette méthode est uniquement disponible 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)` 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 faisant 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.

Nœud de haute disponibilité

> **Note:**
> Les nœuds sont créés par le serveur Zabbix et ne peuvent pas être modifiés via l'API.

L'objet Nœud de haute disponibilité possède les propriétés suivantes.

Propriété	Type	Description
ha_nodeid	ID	ID du nœud.
name	string	Nom attribué au nœud, en utilisant l'entrée de configuration HANodeName de zabbix_server.conf. Vide pour un serveur fonctionnant en mode autonome.
address	string	Nom IP ou DNS à partir duquel le nœud se connecte.
port	integer	Port sur lequel le nœud s'exécute.
lastaccess	integer	Heure du heartbeat, c'est-à-dire l'heure de la dernière mise à jour du nœud. Horodatage UTC.
status	integer	État du nœud. Valeurs possibles : 0 - en veille ; 1 - arrêté manuellement ; 2 - indisponible ; 3 - actif.

hanode.get

Description

`integer/array hanode.get(object parameters)`

La méthode permet de récupérer une liste des nœuds du cluster de haute disponibilité selon les paramètres fournis.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateurs *Super admin*. 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
ha_nodeids	ID/array	Retourner uniquement les nœuds ayant les ID de nœud indiqués.

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 à comparer.
sortfield	string/array	Prend en charge les propriétés suivantes : `name`, `address`, `status`. Trier le résultat selon les propriétés indiquées. Valeurs possibles : `name`, `lastaccess`, `status`.
countOutput	flag	Ces paramètres sont décrits dans les commentaires de référence.
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

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 settings 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 la liste complète des fuseaux horaires pris en charge, veuillez consulter la documentation PHP.
default_theme	string	Thème par défaut. Valeurs possibles : `blue-theme` - *(par défaut)* Bleu ; `dark-theme` - Sombre ; `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.

Propriété	Type	Description
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 indisponible. Valeurs possibles : 0 - Ne pas afficher d'avertissement ; 10 - *(par défaut)* Afficher un avertissement.
work_period	string	Temps de travail. Par défaut : 1-5,09:00-18:00.
show_technical_errors	integer	Afficher les erreurs techniques (PHP/SQL) aux utilisateurs non Super admin et aux utilisateurs qui ne font pas partie de groupes d'utilisateurs avec le mode débogage 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 les secondes et une unité de temps avec suffixe. Par défaut : 24h.
period_default	string	Période par défaut du filtre temporel. Accepte les secondes et une unité de temps avec suffixe, avec prise en charge des mois et des années (30s, 1m, 2h, 1d, 1M, 1y). Par défaut : 1h.
max_period	string	Période maximale pour le filtre temporel. Accepte les secondes et une unité de temps avec suffixe, avec prise en charge des mois et des années (30s, 1m, 2h, 1d, 1M, 1y). Par défaut : 2y.
severity_color_0	string	Couleur pour la sévérité « Non classé » sous forme de code couleur hexadécimal. Par défaut : 97AAB3.
severity_color_1	string	Couleur pour la sévérité « Information » sous forme de code couleur hexadécimal. Par défaut : 7499FF.
severity_color_2	string	Couleur pour la sévérité « Avertissement » sous forme de code couleur hexadécimal. Par défaut : FFC859.
severity_color_3	string	Couleur pour la sévérité « Moyenne » sous forme de code couleur hexadécimal. Par défaut : FFA059.
severity_color_4	string	Couleur pour la sévérité « Haute » sous forme de code couleur hexadécimal. Par défaut : E97659.
severity_color_5	string	Couleur pour la sévérité « Désastre » sous forme de code couleur hexadécimal. Par défaut : E45959.

Propriété	Type	Description
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.
severity_name_3	string	Nom de la sévérité « Moyenne ». Par défaut : Moyenne.
severity_name_4	string	Nom de la sévérité « Haute ». Par défaut : Haute.
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	Période d'affichage des déclencheurs OK. Accepte les secondes et une unité de temps avec suffixe. Par défaut : 5m.
blink_period	string	Période de clignotement lors d'un changement d'état du déclencheur. Accepte les secondes et une unité de temps avec suffixe. Par défaut : 2m.
problem_unack_color	string	Couleur des événements PROBLEM non acquittés sous forme de code couleur hexadécimal. Par défaut : CC0000.
problem_ack_color	string	Couleur des événements PROBLEM acquittés sous forme de code couleur hexadécimal. Par défaut : CC0000.
ok_unack_color	string	Couleur des événements RESOLVED non acquittés sous forme de code couleur hexadécimal. Par défaut : 009900.
ok_ack_color	string	Couleur des événements RESOLVED acquittés sous forme de code couleur hexadécimal. Par défaut : 009900.
problem_unack_style	integer	Clignotement pour les événements PROBLEM non acquittés. Valeurs possibles : 0 - Ne pas afficher de clignotement ; 1 - *(par défaut)* Afficher le clignotement.
problem_ack_style	integer	Clignotement pour les événements PROBLEM acquittés. Valeurs possibles : 0 - Ne pas afficher de clignotement ; 1 - *(par défaut)* Afficher le clignotement.

Propriété	Type	Description
ok_unack_style	integer	Clignotement pour les événements RESOLVED non acquittés. Valeurs possibles : 0 - Ne pas afficher de clignotement ; 1 - *(par défaut)* Afficher le clignotement.
ok_ack_style	integer	Clignotement pour les événements RESOLVED acquittés. Valeurs possibles : 0 - Ne pas afficher de clignotement ; 1 - *(par défaut)* Afficher le clignotement.
url	string	URL du frontend.
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 sera envoyé le message d'alarme d'indisponibilité de la base de données. Si défini sur « 0 », le message d'alarme ne sera pas envoyé.
snmptrap_logging	integer	Journaliser les traps SNMP non correspondants. Valeurs possibles : 0 - Ne pas journaliser les traps SNMP non correspondants ; 1 - *(par défaut)* Journaliser 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 de temps 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 les secondes et une unité de temps avec suffixe. Par défaut : 30s.
validate_uri_schemes	integer	Valider les schémas d'URI. Valeurs possibles : 0 - Ne pas valider ; 1 - *(par défaut)* Valider.
uri_valid_schemes	string	Schémas d'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 sandboxing iframe. Valeurs possibles : 0 - Ne pas utiliser ; 1 - *(par défaut)* Utiliser.
iframe_sandboxing_exceptions	string	Exceptions de sandboxing iframe.

Propriété	Type	Description
connect_timeout	string	Délai d'expiration de 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*
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 d'élément. Plage de valeurs possibles : 1-600s. Par défaut : 60s. Comportement de la propriété : - *required*
script_timeout	string	Délai d'expiration réseau pour l'exécution de 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 des rapports planifiés. 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'auto-enregistrement effectuées par le serveur (utilisateur système). Valeurs possibles : 0 - Désactiver ; 1 - *(par défaut)* Activer.

Propriété	Type	Description
ha_failover_delay	string	Délai de basculement en secondes. Par défaut : 1m. Comportement de la propriété : - *read-only*
geomaps_tile_provider	string	Fournisseur de tuiles Geomap. 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 Geomap. Comportement de la propriété : - *supported* si `geomaps_tile_provider` est défini sur une chaîne vide
geomaps_max_zoom	integer	Niveau de zoom maximal Geomap. 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 Geomap. 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)* serveur uniquement ; 1 - serveur et proxies indépendamment.
timeout_zabbix_agent timeout_simple_check timeout_snmp_agent timeout_external_check timeout_db_monitor	string	Ne pas consacrer plus de `timeout_*` secondes au traitement. Accepte les secondes ou une unité de temps avec suffixe (par ex., 30s, 1m). Accepte également les macros utilisateur. Plage de valeurs possibles : 1-600s. Par défaut : 3s. Par défaut pour `timeout_browser` : 60s. Comportement de la propriété : - *required*

Propriété	Type	Description
timeout_http_agent		
timeout_ssh_agent		
timeout_telnet_agent		
timeout_script		
timeout_browser		

settings.get

Description

`object settings.get(object parameters)`

Cette méthode permet de récupérer l'objet des paramètres selon les paramètres donnés.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appeler cette 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 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": {
        "default_theme": "blue-theme",
        "search_limit": "1000",
        "max_in_table": "50",
        "server_check_interval": "10",
        "work_period": "1-5,09:00-18:00",
        "show_technical_errors": "0",
        "history_period": "24h",
        "period_default": "1h",
        "max_period": "2y",
        "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": "Not classified",
        "severity_name_1": "Information",
        "severity_name_2": "Warning",
        "severity_name_3": "Average",
        "severity_name_4": "High",
        "severity_name_5": "Disaster",
        "custom_color": "0",
        "ok_period": "5m",
        "blink_period": "2m",
        "problem_unack_color": "CC0000",
        "problem_ack_color": "CC0000",
        "ok_unack_color": "009900",
        "ok_ack_color": "009900",
        "problem_unack_style": "1",
        "problem_ack_style": "1",
        "ok_unack_style": "1",
        "ok_ack_style": "1",
        "discovery_groupid": "5",
        "default_inventory_mode": "-1",
        "alert_usrgrpid": "7",
        "snmptrap_logging": "1",
        "default_lang": "en_US",
        "default_timezone": "system",
        "login_attempts": "5",
        "login_block": "30s",
        "validate_uri_schemes": "1",
        "uri_valid_schemes": "http,https,ftp,file,mailto,tel,ssh",
        "x_frame_options": "SAMEORIGIN",
        "iframe_sandboxing_enabled": "1",
        "iframe_sandboxing_exceptions": "",
        "max_overview_table_size": "50",
        "connect_timeout": "3s",
        "socket_timeout": "3s",
        "media_type_test_timeout": "65s",
        "script_timeout": "60s",
        "item_test_timeout": "60s",
        "url": "",
        "report_test_timeout": "60s",
        "auditlog_enabled": "1",
        "auditlog_mode": "1",
        "ha_failover_delay": "1m",
        "geomaps_tile_provider": "OpenStreetMap.Mapnik",
        "geomaps_tile_url": "",
        "geomaps_max_zoom": "0",
        "geomaps_attribution": "",
        "vault_provider": "0",
        "timeout_zabbix_agent": "3s",
        "timeout_simple_check": "3s",
        "timeout_snmp_agent": "3s",
        "timeout_external_check": "3s",
        "timeout_db_monitor": "3s",
        "timeout_http_agent": "3s",
        "timeout_ssh_agent": "3s",
        "timeout_telnet_agent": "3s",
        "timeout_script": "3s"
    },
    "id": 1
}
```


Source

CSettings::get() dans *ui/include/classes/api/services/CSettings.php*.

settings.update

Description

`object settings.update(object settings)`

Cette méthode permet de mettre à jour les paramètres communs existants.

> **Note:**
> Cette méthode est uniquement disponible 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)` 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
- Tag 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.

Propriété	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 du statut 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 du statut 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 résolution.
r_clock	timestamp	Heure de création de l'événement de résolution.
r_ns	integer	Nanosecondes au moment de la création de l'événement de résolution.
cause_eventid	ID	ID de l'événement cause.
correlationid	ID	ID de la règle de corrélation si cet événement a été résolu par une règle de corrélation globale.
userid	ID	ID de l'utilisateur ayant fermé le problème (si le problème a été fermé manuellement).
name	string	Nom du problème résolu.
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 - moyenne ; 4 - élevée ; 5 - désastre.
suppressed	integer	Indique si le problème est supprimé. Valeurs possibles : 0 - le problème est dans un 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 a 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 permet de récupérer les problèmes selon les paramètres donnés.

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 à partir de quand un problème est considéré comme « récemment » résolu est définie dans *Administration* → *General*. Les problèmes qui ont été résolus avant cette période ne sont pas conservés dans la table des problèmes. Pour récupérer les problèmes qui ont été 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 permettant d'appeler la méthode peuvent être révoquées dans les paramètres des rôles 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
eventids	ID/array	Retourner uniquement les problèmes avec les ID donnés.
groupids	ID/array	Retourner uniquement les problèmes créés par des objets appartenant aux groupes d'hôtes donnés.
hostids	ID/array	Retourner uniquement les problèmes créés par des objets appartenant aux hôtes donnés.
objectids	ID/array	Retourner uniquement les problèmes créés par les objets donnés.
source	integer	Retourner uniquement 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 trigger.

Paramètre	Type	Description
object	integer	Retourner uniquement 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 - trigger.
acknowledged	boolean	Si défini sur `true`, retourner uniquement les problèmes acquittés.
action	integer	Retourner uniquement les problèmes pour lesquels les actions de mise à jour d'événement données ont été effectuées. Pour plusieurs actions, utilisez la somme de toutes les valeurs bitmap acceptables comme masque de bits (par exemple, 34 pour acquitter et supprimer un événement).
action_userids	ID/array	Retourner uniquement les problèmes avec les ID donnés des utilisateurs ayant effectué les actions de mise à jour de l'événement de problème.
suppressed	boolean	Si défini sur `true`, retourner uniquement les problèmes supprimés.
symptom	boolean	Si défini sur `true`, retourner uniquement les événements de problème symptomatiques.
severities	integer/array	Retourner uniquement les problèmes avec les sévérités d'événement données. S'applique uniquement si l'objet est un trigger.
evaltype	integer	Méthode d'évaluation des tags. Valeurs possibles : 0 - *(par défaut)* Et/Ou ; 2 - Ou.
tags	array	Retourner uniquement les problèmes avec les tags donnés. 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 uniquement les problèmes récemment résolus (dépend de `ok_period`).
eventid_from	string	Retourner uniquement les problèmes avec des ID supérieurs ou égaux à l'ID donné.
eventid_till	string	Retourner uniquement les problèmes avec des ID inférieurs ou égaux à l'ID donné.
time_from	timestamp	Retourner uniquement les problèmes créés après ou au moment indiqué.
time_till	timestamp	Retourner uniquement les problèmes créés avant ou au moment indiqué.

Paramètre	Type	Description
selectAcknowledges	query	Retourner une propriété `acknowledges` avec les mises à jour du problème. Les mises à jour du problème sont triées dans l'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 ayant 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. Prend en charge `count`.
selectTags	query	Retourner une propriété `tags` avec les tags 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 ayant supprimé le problème ; `suppress_until` - `(integer)` heure jusqu'à laquelle le problème est supprimé.
filter	object	Retourner uniquement les résultats correspondant 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` data type.
sortfield	string/array	Trier le résultat selon les propriétés données. Valeurs possibles : `eventid`.
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 événements de problème du déclencheur

Récupérez 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": "L'agent Zabbix sur localhost est injoignable depuis 5 minutes",
            "acknowledged": "1",
            "severity": "3",
            "cause_eventid": "0",
            "opdata": "",
            "acknowledges": [
                {
                    "acknowledgeid": "14443",
                    "userid": "1",
                    "eventid": "1245463",
                    "clock": "1472457281",
                    "message": "problème résolu",
                    "action": "6",
                    "old_severity": "0",
                    "new_severity": "0",
                    "suppress_until": "1472511600",
                    "taskid": "0"
                }
            ],
            "suppression_data": [
                {
                    "maintenanceid": "15",
                    "suppress_until": "1472511600",
                    "userid": "0"
                }
            ],
            "suppressed": "1",
            "tags": [
```

```
                {
                    "tag": "test-tag",
                    "value": "test-value"
                }
            ]
        }
    ],
    "id": 1
}
```

Récupération des problèmes acquittés par l'utilisateur spécifié

Récupération des problèmes acquittés par l'utilisateur avec 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": "L'agent Zabbix sur localhost est injoignable depuis 5 minutes",
            "acknowledged": "1",
            "severity": "3",
            "cause_eventid": "0",
            "opdata": "",
            "acknowledges": [
                {
                    "userid": "10",
                    "action": "2"
                }
            ],
            "suppressed": "0"
        }
    ],
    "id": 1
}
```

See also

- Alerte
- Elément
- Hôte
- Règle LLD
- Déclencheur

Source

CEvent::get() dans *ui/include/classes/api/services/CProblem.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 du 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.

Propriété	Type	Description
itemid	ID	ID du prototype d'élément. Comportement de la propriété : - *lecture seule* - *obligatoire* pour les opérations de mise à jour

Propriété	Type	Description
delay	string	Intervalle de mise à jour du prototype d'élément. Accepte des secondes ou une unité de temps avec suffixe (par exemple, 30s, 1m, 2h, 1d) et, éventuellement, un ou plusieurs intervalles personnalisés, tous séparés par des points-virgules. Les intervalles personnalisés peuvent être un mélange d'intervalles flexibles et d'intervalles de planification. Accepte les macros utilisateur et les macros LLD. Si elles sont utilisées, la valeur doit être une macro unique. Plusieurs macros ou des macros mélangées avec 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), 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 d'élément. Comportement de la propriété : - *constant* - *obligatoire* pour les opérations de création
interfaceid	ID	ID de l'interface 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 - *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 - *lecture seule* pour les objets hérités

Propriété	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. Comportement de la propriété : - *obligatoire* pour les opérations de création - *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" - *lecture seule* pour les objets hérités
value_type	integer	Type d'information du prototype d'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 - *lecture seule* pour les objets hérités
allow_traps	integer	Autoriser le remplissage de la valeur de manière similaire à l'élément trapper. 0 - *(par défaut)* Ne pas autoriser l'acceptation de données entrantes ; 1 - Autoriser l'acceptation de 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 - basique ; 2 - NTLM ; 3 - Kerberos ; 4 - Digest. Comportement de la propriété : - *pris en charge* si `type` est défini sur "SSH agent" ou "HTTP agent" - *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 : - *lecture seule*
follow_redirects	integer	Suivre les redirections de 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" - *lecture seule* pour les objets hérités
headers	array	Tableau des 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" - *lecture seule* pour les objets hérités
history	string	Unité de temps indiquant 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 du proxy HTTP(S). Comportement de la propriété : - *pris en charge* si `type` est défini sur "HTTP agent" - *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" - *lecture seule* pour les objets hérités

Propriété	Type	Description
jmx_endpoint	string	Chaîne de connexion personnalisée de l'agent JMX. 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" - *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" - *lecture seule* pour les objets hérités
output_format	integer	Indique si la réponse doit être convertie en JSON. Valeurs possibles : 0 - *(par défaut)* Stocker brut ; 1 - Convertir en JSON. Comportement de la propriété : - *pris en charge* si `type` est défini sur "HTTP agent" - *lecture seule* pour les objets hérités
params	string	Paramètres supplémentaires selon le type du prototype d'élément : - script exécuté pour les prototypes d'élément SSH agent et TELNET agent ; - requête SQL pour les prototypes d'élément Database monitor ; - formule pour les prototypes d'élément Calculated ; - le script pour les prototypes d'élément Script et Browser. Comportement de la propriété : - *obligatoire* si `type` est défini sur "Database monitor", "SSH agent", "TELNET agent", "Calculated", "Script" ou "Browser" - *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 avec 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" - *lecture seule* pour les objets hérités
password	string	Mot de passe pour l'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" - *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” - *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 ”JSON data” ou ”XML data” - *pris en charge* si `type` est défini sur ”HTTP agent” et que `post_type` est défini sur ”Raw data” - *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 ”public key”
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”
query_fields	array	Tableau des 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” - *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” - *lecture seule* pour les objets hérités
retrieve_mode	integer	Quelle partie de la réponse doit être stockée. 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 tous deux 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” - *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” - *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” - *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” - *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” - *lecture seule* pour les objets hérités
status	integer	Statut 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 des codes d’état HTTP requis, séparées par des virgules. Prend également en charge les macros utilisateur ou les macros LLD comme partie d’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” - *lecture seule* pour les objets hérités
templateid	ID	ID du prototype d’élément du modèle parent. Comportement de la propriété : - *lecture seule*
timeout	string	Délai d’expiration de la requête d’interrogation des données de l’élément. Accepte des secondes ou une unité de temps avec suffixe (par exemple, 30s, 1m). Accepte également les macros utilisateur et les macros LLD. Plage de valeurs possible : 1-600s. Par défaut : ”” - utiliser les paramètres du proxy/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) - *lecture seule* pour les objets hérités

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 to accept incoming data”
trends	string	Unité de temps indiquant combien de temps les données de tendances 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 valeur. Comportement de la propriété : - *pris en charge* si `value_type` est défini sur ”numeric float” ou ”numeric unsigned” - *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” - *lecture seule* pour les objets hérités (si `type` est défini sur ”HTTP agent”)
uuid	string	Identifiant universel unique, utilisé pour lier les prototypes d’élément 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 du mappage de valeurs associé. Comportement de la propriété : - *pris en charge* si `value_type` est défini sur ”numeric float”, ”character” ou ”numeric unsigned” - *lecture seule* pour les objets hérités
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” - *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” - *lecture seule* pour les objets hérités

Propriété	Type	Description
discover	integer	Statut 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.

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*

Tag de prototype d'élément

L'objet tag de prototype d'élément possède les propriétés suivantes.

Propriété	Type	Description
tag	string	Nom du tag de prototype d'élément. Comportement de la propriété : - *obligatoire*
value	string	Valeur du tag de prototype d'élément. Comportement de la propriété : - *lecture seule*.

Propriété	Type	Description
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.

Property	Type	Description
type	integer	Le type d'option de prétraitement. Valeurs possibles : 1 - Multiplicateur personnalisé ; 2 - Supprimer à droite ; 3 - Supprimer à gauche ; 4 - Supprimer ; 5 - Expression régulière ; 6 - Booléen vers décimal ; 7 - Octal vers décimal ; 8 - Hexadécimal vers décimal ; 9 - Changement simple ; 10 - Changement par seconde ; 11 - XML XPath ; 12 - JSONPath ; 13 - Dans l'intervalle ; 14 - Correspond à l'expression régulière ; 15 - Ne correspond pas à l'expression régulière ; 16 - Vérifier une erreur dans JSON ; 17 - Vérifier une erreur dans XML ; 18 - Vérifier une erreur à l'aide d'une expression régulière ; 19 - Ignorer si inchangé ; 20 - Ignorer si inchangé avec heartbeat ; 21 - JavaScript ; 22 - Motif Prometheus ; 23 - Prometheus vers JSON ; 24 - CSV vers JSON ; 25 - Remplacer ; 26 - Vérifier non pris en charge ; 27 - XML vers JSON ; 28 - Valeur SNMP walk ; 29 - SNMP walk vers JSON ; 30 - Valeur SNMP get. Comportement de la propriété : - *obligatoire*

Property	Type	Description
params	string	Paramètres supplémentaires utilisés par l'option de prétraitement. Les paramètres multiples sont séparés par le caractère de nouvelle ligne (\n). Si `type` est défini sur "Check unsupported", les paramètres suivent la syntaxe `<scope>[\n<pattern>]`, où *pattern* est une expression régulière et *scope* est l'un des suivants : -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 : - *obligatoire* si `type` est défini sur "Custom multiplier" (1), "Right trim" (2), "Left trim" (3), "Trim" (4), "Regular expression" (5), "XML XPath" (11), "JSONPath" (12), "In range" (13), "Matches regular expression" (14), "Does not match regular expression" (15), "Check for error in JSON" (16), "Check for error in XML" (17), "Check for error using regular expression" (18), "Discard unchanged with heartbeat" (20), "JavaScript" (21), "Prometheus pattern" (22), "Prometheus to JSON" (23), "CSV to JSON" (24), "Replace" (25), "Check unsupported" (26), "SNMP walk value" (28), "SNMP walk to JSON" (29) ou "SNMP get value" (30)
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. Valeurs possibles si `type` est défini sur "Check unsupported" : 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 "Custom multiplier" (1), "Regular expression" (5), "Boolean to decimal" (6), "Octal to decimal" (7), "Hexadecimal to decimal" (8), "Simple change" (9), "Change per second" (10), "XML XPath" (11), "JSONPath" (12), "In range" (13), "Matches regular expression" (14), "Does not match regular expression" (15), "Check for error in JSON" (16), "Check for error in XML" (17), "Check for error using regular expression" (18), "Prometheus pattern" (22), "Prometheus to JSON" (23), "CSV to JSON" (24), "Check unsupported" (26), "XML to JSON" (27), "SNMP walk value" (28), "SNMP walk to JSON" (29) ou "SNMP get value" (30)
error_handler_params	string	Paramètres du gestionnaire d'erreurs. Comportement de la propriété : - *obligatoire* si `error_handler` est défini sur "Set custom value" ou "Set custom error message"

Les paramètres suivants et les gestionnaires d'erreurs suivants sont pris en charge pour chaque type de prétraitement.

Preprocessing type	Name	Parameter 1	Parameter 2	Parameter 3	Supported error handlers
1	Multiplication personnalisé	nombre[1, 6]			0, 1, 2, 3

Preprocessing type	Name	Parameter 1	Parameter 2	Parameter 3	Supported error handlers
2	Supprimer à droite	liste de caractères[2]			
3	Supprimer à gauche	liste de caractères[2]			
4	Supprimer	liste de caractères[2]			
5	Expression régulière	motif[3]	sortie[2]		0, 1, 2, 3
6	Booléen vers décimal				0, 1, 2, 3
7	Octal vers décimal				0, 1, 2, 3
8	Hexadécimal vers décimal				0, 1, 2, 3
9	Changement simple				0, 1, 2, 3
10	Changement par seconde				0, 1, 2, 3
11	XML XPath	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	motif[3]			0, 1, 2, 3
15	Ne correspond pas à l’expression régulière	motif[3]			0, 1, 2, 3
16	Vérifier une erreur dans JSON	chemin[4]			0, 1, 2, 3
17	Vérifier une erreur dans XML	chemin[4]			0, 1, 2, 3

Preprocessing type	Name	Parameter 1	Parameter 2	Parameter 3	Supported error handlers
18	Vérifier une erreur à l'aide d'une expression régulière	motif[3]	sortie[2]		0, 1, 2, 3
19	Ignorer si inchangé				
20	Ignorer si inchangé avec heartbeat	secondes[5, 6]			
21	JavaScript	script[2]			
22	Motif Prometheus	motif[6, 7]	`value`, `label`, `function`	sortie[8, 9]	0, 1, 2, 3
23	Prometheus vers JSON	motif[6, 7]			0, 1, 2, 3
24	CSV vers JSON	caractère[2]	caractère[2]	0,1	0, 1, 2, 3
25	Remplacer	chaîne recherchée[2]	remplacement[2]		
26	Vérifier non pris en charge	scope[1]	motif[3, 6]		1, 2, 3
27	XML vers JSON				0, 1, 2, 3
28	Valeur SNMP walk	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	SNMP walk vers 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

Preprocessing type	Name	Parameter 1	Parameter 2	Parameter 3	Supported error handlers
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] entier ou nombre à virgule flottante
[2] chaîne
[3] expression régulière
[4] JSONPath ou XML XPath
[5] entier positif (avec prise en charge des suffixes de temps, par ex. 30s, 1m, 2h, 1d)
[6] macro utilisateur, macro LLD
[7] Motif Prometheus suivant la syntaxe : `<metric name>{<label name>="<label value>", ...} == <value>`. Chaque composant du motif Prometheus (métrique, nom d'étiquette, valeur d'étiquette et valeur de 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] Une des fonctions d'agrégation : `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 permet de créer de nouveaux prototypes d'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 des rôles utilisateur. Voir User roles 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	Tags 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 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 du rôle utilisateur. Voir 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)`

Cette méthode permet de récupérer les prototypes d'élément selon les paramètres donnés.

> **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.

Paramètre	Type	Description
discoveryids	ID/array	Retourner uniquement les prototypes d'éléments appartenant aux règles LLD données.
graphids	ID/array	Retourner uniquement les prototypes d'éléments utilisés dans les prototypes de graphiques donnés.
hostids	ID/array	Retourner uniquement les prototypes d'éléments appartenant aux hôtes donnés.
inherited	boolean	Si défini sur `true`, retourner uniquement les prototypes d'éléments hérités d'un modèle.
itemids	ID/array	Retourner uniquement les prototypes d'éléments avec les ID donnés.
monitored	boolean	Si défini sur `true`, retourner uniquement les prototypes d'éléments activés appartenant à des hôtes surveillés.
templated	boolean	Si défini sur `true`, retourner uniquement les prototypes d'éléments appartenant à des modèles.
templateids	ID/array	Retourner uniquement les prototypes d'éléments appartenant aux modèles donnés.
triggerids	ID/array	Retourner uniquement les prototypes d'éléments utilisés dans les prototypes de déclencheurs donnés.
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 graphiques 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.

Paramètre	Type	Description
selectTags	query	Retourner les tags du prototype d'élément dans la propriété `tags`.
selectTriggers	query	Retourner une propriété `triggers` avec les prototypes de déclencheurs 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`.
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 également

- 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 permet de mettre à jour des prototypes d'éléments 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 User roles pour plus d'informations.

Paramètres

`(object/array)` Propriétés des prototypes 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 des prototypes d'élément, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
preprocessing	array	Options de prétraitement des prototypes d'élément pour remplacer les options de prétraitement actuelles. Comportement des paramètres : - *lecture seule* pour les objets hérités
tags	array	Balises des prototypes 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 les prototypes de déclencheur.

Références d'objet :

- Prototype de déclencheur
- Balise 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
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 - moyenne ; 4 - élevée ; 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.

Propriété	Type	Description
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 balises correspondent.
correlation_tag	string	Balise pour la correspondance.
manual_close	integer	Autoriser la fermeture manuelle. Valeurs possibles : 0 - *(par défaut)* Non ; 1 - Oui.
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 du prototype de déclencheur. Comportement de la propriété : - *obligatoire*
value	string	Valeur de la balise du 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*.

Propriété	Type	Description
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 permet de créer de nouveaux prototypes de déclencheur.

> **Note:**
> Cette méthode n'est disponible que 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)` 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. Seule la propriété `triggerid` doit être définie pour les déclencheurs.
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 permet de supprimer des prototypes de déclencheur.

> **Note:**
> Cette méthode est disponible uniquement pour les utilisateurs de type *Admin* et *Super admin*. Les autorisations d'appel de la méthode peuvent être révoquées dans les réglages du rôle d'utilisateur. Voir les 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 des prototypes de déclencheur selon les paramètres donnés.

> **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.

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 modèles issus 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 modèles issus des groupes d'hôtes ou groupes de modèles donnés.
host	string	Retourner uniquement les prototypes de déclencheur qui appartiennent aux 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 dont la sévérité est 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 de données 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 ayant 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`.

Paramètre	Type	Description
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 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 prototype de déclencheur a été créé ; `status` - `(int)` état 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.
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 de données 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 renvoyée.
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 de données référencés dans les expressions de déclenchement ou de récupération.
selectItems	query	Retourner les éléments de données et prototypes d'élément de données 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`.

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 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 déclencheur ; `hostid` - ID de l'hôte auquel appartient le prototype de déclencheur.
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 : `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 permet de mettre à jour des prototypes de déclencheur existants.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateurs *Admin* et *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 des prototypes 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 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	Balises 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

Propriété	Type	Description
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.
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 rendu comme si `ymax_type` était défini sur « calculated ».

Propriété	Type	Description
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 rendu 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	Statut 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 permet de créer de nouveaux prototypes de graphiques.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateurs *Admin* et *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)` 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 du paramètre : - *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 également

- Elément graphique

Source

CGraphPrototype::create() dans *ui/include/classes/api/services/CGraphPrototype.php*.

graphprototype.delete

Description

`object graphprototype.delete(array graphPrototypeIds)`

Cette méthode permet de supprimer des prototypes de graphiques.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateurs *Admin* et *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 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)`

Cette méthode permet de récupérer des prototypes de graphiques selon les paramètres donnés.

> **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.

Paramètre	Type	Description
discoveryids	ID/array	Retourner uniquement les prototypes de graphique appartenant aux règles de découverte indiquées.
graphids	ID/array	Retourner uniquement les prototypes de graphique ayant les ID indiqués.
groupids	ID/array	Retourner uniquement les prototypes de graphique appartenant aux hôtes ou aux modèles des groupes d'hôtes ou groupes de modèles indiqués.
hostids	ID/array	Retourner uniquement les prototypes de graphique appartenant aux hôtes indiqués.
inherited	boolean	Si défini sur `true`, retourner uniquement les prototypes de graphique hérités d'un modèle.
itemids	ID/array	Retourner uniquement les prototypes de graphique contenant les prototypes d'élément indiqués.
templated	boolean	Si défini sur `true`, retourner uniquement les prototypes de graphique appartenant à des modèles.

Paramètre	Type	Description
templateids	ID/array	Retourner uniquement les prototypes de graphique 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 au 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 la règle prototype LLD parente à laquelle 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'élément 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.
filter	object	Retourner uniquement les résultats correspondant 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` 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 permet de mettre à jour des prototypes de graphiques existants.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateurs *Admin* et *Super admin*. Les autorisations permettant d'appeler la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir User roles pour plus d'informations.

Paramètres

`(object/array)` Propriétés des prototypes de graphique 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 graphique, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
gitems	array	Éléments de graphique remplaçant 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
    - Balise 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.

Property	Type	Description
itemid	ID	ID du prototype de règle LLD. Comportement de la propriété : - *lecture seule* - *obligatoire* pour les opérations de mise à jour
ruleid	ID	ID de la règle LLD/prototype de règle LLD parent. Comportement de la propriété : - *lecture seule* - *obligatoire* pour les opérations de création

Property	Type	Description
delay	string	Intervalle de mise à jour du prototype de règle LLD. Accepte des secondes ou une unité de temps avec suffixe (par exemple, 30s, 1m, 2h, 1d) et, éventuellement, un ou plusieurs intervalles personnalisés, tous séparés par des points-virgules. Les intervalles personnalisés peuvent être un mélange d'intervalles flexibles et d'intervalles de planification. Accepte les macros utilisateur. Si elles sont utilisées, la valeur doit être une macro unique. Plusieurs macros ou des macros mélangées avec du texte ne sont pas prises en charge. Les intervalles flexibles peuvent être écrits sous 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 - un prototype de règle de découverte de bas niveau ; 7 - un prototype de règle de découverte de bas niveau découvert. Comportement de la propriété : - *lecture seule*
interfaceid	ID	ID de l'interface 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 - *lecture seule* pour les objets hérités
name	string	Nom du prototype de règle LLD. Comportement de la propriété : - *obligatoire* pour les opérations de création - *lecture seule* pour les objets hérités

Property	Type	Description
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 - *lecture seule* pour les objets hérités
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” - *lecture seule* pour les objets hérités
allow_traps	integer	Permet de renseigner la valeur de manière similaire à un élément trapper. Valeurs possibles : 0 - *(par défaut)* Ne pas autoriser l’acceptation de données entrantes ; 1 - Autoriser l’acceptation de 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. Comportement de la propriété : - *pris en charge* si `type` est défini sur ”SSH agent” ou ”HTTP agent” - *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 s’il y a des problèmes lors de la mise à jour de la valeur du prototype de règle LLD. Comportement de la propriété : - *lecture seule*

Property	Type	Description
follow_redirects	integer	Suivre les redirections de 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" - *lecture seule* pour les objets hérités
headers	array	Tableau des 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" - *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" - *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" - *lecture seule* pour les objets hérités
jmx_endpoint	string	Chaîne de connexion personnalisée de l'agent JMX. 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 des secondes, une unité de temps avec suffixe ou une macro utilisateur. Par défaut : 7d.
lifetime_type	integer	Scénario de suppression des ressources LLD perdues. Valeurs possibles : 0 - *(par défaut)* Supprimer après l'atteinte du seuil de durée de vie ; 1 - Ne pas supprimer ; 2 - Supprimer immédiatement.
enabled_lifetime	string	Période après laquelle les éléments qui ne sont plus découverts seront désactivés. Accepte des secondes, une unité de temps avec suffixe ou une macro utilisateur. Par défaut : 0.
enabled_lifetime_type	integer	Scénario de désactivation des ressources LLD perdues. Valeurs possibles : 0 - Désactiver après l'atteinte du seuil de durée de vie ; 1 - Ne pas désactiver ; 2 - *(par défaut)* Désactiver immédiatement.

Property	Type	Description
master_itemid	ID	ID de l'élément maître. Une règle de découverte ne peut pas être un élément maître pour une autre règle de découverte. Comportement de la propriété : - *obligatoire* si `type` est défini sur "Dependent item" - *lecture seule* pour les objets hérités
output_format	integer	Indique si la réponse doit être convertie en JSON. Valeurs possibles : 0 - *(par défaut)* Stocker brut ; 1 - Convertir en JSON. Comportement de la propriété : - *pris en charge* si `type` est défini sur "HTTP agent" - *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 surveillance de base de données ; - formule pour les règles LLD calculées ; - script pour les règles LLD de type script et browser. Comportement de la propriété : - *obligatoire* si `type` est défini sur "Database monitor", "SSH agent", "TELNET agent", "Script" ou "Browser" - *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 avec 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" - *lecture seule* pour les objets hérités
password	string	Mot de passe pour l'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" - *lecture seule* pour les objets hérités (si `type` est défini sur "HTTP agent")
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" - *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 "JSON data" ou "XML data" - *pris en charge* si `type` est défini sur "HTTP agent" et que `post_type` est défini sur "Raw data" - *lecture seule* pour les objets hérités

Property	Type	Description
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”
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”
query_fields	array	Tableau des 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” - *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” - *lecture seule* pour les objets hérités
retrieve_mode	integer	Quelle partie de la réponse doit être stockée. 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” - *lecture seule* pour les objets hérités
snmp_oid	string	OID SNMP. Comportement de la propriété : - *obligatoire* si `type` est défini sur ”SNMP agent” - *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” - *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” - *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” - *lecture seule* pour les objets hérités

Property	Type	Description
state	integer	État du prototype de règle LLD. Valeurs possibles : 0 - *(par défaut)* normal ; 1 - non pris en charge. Comportement de la propriété : - *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 comme partie d'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" - *lecture seule* pour les objets hérités
templateid	ID	ID du prototype de règle LLD du modèle parent. Comportement de la propriété : - *lecture seule*
timeout	string	Délai d'expiration de la requête d'interrogation des données de l'élément. Accepte des secondes ou une unité de temps avec suffixe (par exemple, 30s, 1m). Accepte également les macros utilisateur. Plage de valeurs possibles : 1-600s. Par défaut : "" - utiliser les paramètres proxy/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) - *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 "Allow to accept incoming data"
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" - *lecture seule* pour les objets hérités (si `type` est défini sur "HTTP agent")

Property	Type	Description
uuid	string	Identifiant universel unique, utilisé pour lier les prototypes de règle 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 valider que le nom d'hôte de la connexion correspond à celui figurant dans le certificat de l'hôte. Valeurs possibles : 0 - *(par défaut)* Ne pas valider ; 1 - Valider. Comportement de la propriété : - *pris en charge* si `type` est défini sur "HTTP agent" - *lecture seule* pour les objets hérités
verify_peer	integer	Indique s'il faut valider que le certificat de l'hôte est authentique. Valeurs possibles : 0 - *(par défaut)* Ne pas valider ; 1 - Valider. Comportement de la propriété : - *pris en charge* si `type` est défini sur "HTTP agent" - *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*

Propriété	Type	Description
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*
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 identifiants 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 identifiants qui référencent des conditions de filtre spécifiques par leur `formulaid`. Les identifiants 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 « custom expression »

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*

Propriété	Type	Description
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	Le type d'option de prétraitement. Valeurs possibles : 5 - Expression régulière ; 11 - XML XPath ; 12 - JSONPath ; 14 - Correspond à l'expression régulière ; 15 - Ne correspond pas à l'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 SNMP walk ; 29 - SNMP walk 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), "XML XPath" (11), "JSONPath" (12), "Correspond à l'expression régulière" (14), "Ne correspond pas à l'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 SNMP walk" (28), "SNMP walk vers JSON" (29) ou "Valeur SNMP get" (30)
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), "XML XPath" (11), "JSONPath" (12), "Correspond à l'expression régulière" (14), "Ne correspond pas à l'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 SNMP walk" (28), "SNMP walk 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 suivants et les 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	motif[1]	sortie[2]		0, 1, 2, 3
11	XML XPath	chemin[3]			0, 1, 2, 3
12	JSONPath	chemin[3]			0, 1, 2, 3
14	Correspond à l’expression régulière	motif[1]			0, 1, 2, 3
15	Ne correspond pas à l’expression régulière	motif[1]			0, 1, 2, 3
16	Vérifier la présence d’une erreur dans JSON	chemin[3]			0, 1, 2, 3
17	Vérifier la présence d’une erreur dans XML	chemin[3]			0, 1, 2, 3
20	Ignorer les valeurs inchangées avec heartbeat	secondes[4, 5]			
21	JavaScript	script[2]			
23	Prometheus vers JSON	motif[5, 6]			0, 1, 2, 3
24	CSV vers JSON	caractère[2]	caractère[2]	0,1	0, 1, 2, 3
25	Remplacer	chaîne de recherche[2]	remplacement[2]		
27	XML vers JSON				0, 1, 2, 3
28	Valeur SNMP walk	OID[2]	Format : 0 - Inchangé 1 - UTF-8 à partir de Hex-STRING 2 - MAC à partir de Hex-STRING 3 - Entier à partir de BITS		0, 1, 2, 3

Type de prétraitement	Nom	Paramètre 1	Paramètre 2	Paramètre 3	Gestionnaires d'erreurs pris en charge
29	SNMP walk vers JSON[7]	Nom du champ[2]	Préfixe OID[2]	Format : 0 - Inchangé 1 - UTF-8 à partir de Hex-STRING 2 - MAC à partir de Hex-STRING 3 - Entier à partir de BITS	0, 1, 2, 3
30	Valeur SNMP get	Format : 1 - UTF-8 à partir de Hex-STRING 2 - MAC à partir de Hex-STRING 3 - Entier à partir de BITS			0, 1, 2, 3

[1] expression régulière
[2] chaîne
[3] JSONPath ou XML XPath
[4] entier positif (avec prise en charge des suffixes de temps, par ex. 30s, 1m, 2h, 1d)
[5] macro utilisateur
[6] motif Prometheus suivant la syntaxe : `<metric name>{<label name>="<label value>", ...} == <value>`. Chaque composant du motif Prometheus (métrique, nom d'étiquette, valeur d'étiquette et valeur de métrique) peut être une macro utilisateur.
[7] Prend en charge plusieurs enregistrements "Nom du champ,Préfixe OID,Format", délimités par un caractère de nouvelle ligne.

Remplacements de prototypes de règle LLD

L'objet de remplacements de prototypes 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 surcharge du prototype de règle LLD

L'objet de filtre de surcharge du prototype de règle LLD définit un ensemble de conditions qui, si elles correspondent à l'objet découvert, appliquent la surcharge. Il possède les propriétés suivantes :

Property	Type	Description
conditions	object/array	Ensemble de conditions de filtre de surcharge 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 surcharge. 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 surcharge. L'expression contient des identifiants qui référencent des conditions spécifiques du filtre de surcharge 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 de surcharge avec une expression personnalisée. L'expression doit contenir des identifiants qui référencent des conditions spécifiques du filtre de surcharge par leur `formulaid`. Les identifiants utilisés dans l'expression doivent correspondre exactement à ceux définis dans les conditions du filtre de surcharge : aucune condition ne peut rester inutilisée ou omise. Comportement de la propriété : - *obligatoire* si `evaltype` est défini sur « custom expression »

Condition de filtre de remplacement de prototype de règle LLD

L'objet de condition de filtre de remplacement 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 remplacement de prototype de règle LLD est défini sur « expression personnalisée »

Propriété	Type	Description
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 du prototype de règle LLD

L'opération de remplacement du 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*
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 de l'opération de remplacement pour les objets prototype d'élément, de déclencheur et d'hôte.
opdiscover	object	Objet d'état de découverte de l'opération de remplacement (tous les types d'objet).
opperiod	object	Objet de période (intervalle de mise à jour) de l'opération de remplacement pour l'objet prototype d'élément.
ophistory	object	Objet d'historique de l'opération de remplacement pour l'objet prototype d'élément.
optrends	object	Objet de tendances de l'opération de remplacement pour l'objet prototype d'élément.
opseverity	object	Objet de sévérité de l'opération de remplacement pour l'objet prototype de déclencheur.
optag	object/array	Objet de balise de l'opération de remplacement pour les objets prototype de déclencheur et d'hôte.
optemplate	object/array	Objet de modèle de l'opération de remplacement pour l'objet prototype d'hôte.
opinventory	object	Objet d'inventaire de l'opération de remplacement 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 remplacement du prototype de règle LLD

Statut de découverte de l'opération de remplacement du prototype de règle LLD défini pour 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 d'opération de surcharge du prototype de règle LLD

La période d'opération de surcharge du prototype de règle LLD est une valeur d'intervalle de mise à jour définie pour l'élément découvert. Elle possède les propriétés suivantes :

Propriété	Type	Description
delay	string	Remplace l'intervalle de mise à jour du prototype d'élément. Accepte des secondes ou une unité de temps avec suffixe (par exemple, 30s, 1m, 2h, 1d) et, éventuellement, un ou plusieurs intervalles personnalisés, tous séparés par des points-virgules. Les intervalles personnalisés peuvent être un mélange d'intervalles flexibles et d'intervalles planifiés. Accepte les macros utilisateur ou les macros LLD. Si elles sont utilisées, la valeur doit être une macro unique. Plusieurs macros ou des macros mélangées avec 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*

Historique de l'opération de surcharge du prototype de règle LLD

Valeur de l'historique de l'opération de surcharge du prototype de règle LLD définie pour l'élément découvert. Elle possède les propriétés suivantes :

Propriété	Type	Description
history	string	Surcharge 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 la macro utilisateur et la 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 - désastre. 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 :

Propriété	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 :

Property	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 substitution du prototype de règle LLD

Valeur du mode d'inventaire de l'opération de substitution du prototype de règle LLD définie pour 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 permet de créer de nouveaux prototypes de règles LLD.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateurs *Admin* et *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)` Prototypes de règle LLD à créer.

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 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 substitutions 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 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. Voir User roles 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 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 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
discoveryids	ID/array	Retourner uniquement les prototypes de règle LLD qui appartiennent aux règles LLD ou aux prototypes de règle LLD donnés.
itemids	ID/array	Retourner uniquement les prototypes de règle LLD avec les ID donnés.
groupids	ID/array	Retourner uniquement les prototypes de règle LLD qui appartiennent aux hôtes des groupes donnés.
hostids	ID/array	Retourner uniquement les prototypes de règle LLD qui appartiennent aux hôtes donnés.
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 données.
monitored	boolean	Si défini sur `true`, retourner uniquement les prototypes de règle LLD activés qui appartiennent à des hôtes surveillés.
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 donnés.
selectDiscoveryRule	query	Retourner une propriété `discoveryRule` avec la règle LLD parente du prototype de règle LLD. Retournera une réponse vide 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. Retournera une réponse vide 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`.

Paramètre	Type	Description
selectFilter	query	Retourner une propriété `filter` avec les données du filtre utilisé par le prototype de règle LLD.
selectGraphs	query	Retourner une propriété `graphs` avec les prototypes de graphique qui appartiennent au prototype de règle LLD. Prend en charge `count`.
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 des 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 effectuées 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 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 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 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 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 des rôles utilisateur. Voir User roles 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 du prototype de règle LLD pour remplacer les options de prétraitement existantes. Comportement du paramètre: - *lecture seule* pour les objets hérités

Paramètre	Type	Description
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
overrides	object/array	Options de surcharges du 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*.

Prototype d'hôte

Cette classe est conçue pour fonctionner avec les 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
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 prototype d'hôte du modèle parent. 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.

Propriété	Type	Description
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 associe 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*.

Propriété	Type	Description
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 « utiliser les interfaces personnalisées des prototypes d'hôte ». 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*
useip	integer	Indique si la connexion doit être établie via l'adresse 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 « se connecter en utilisant l'adresse IP de l'hôte »
dns	string	Nom DNS utilisé par l'interface. Peut contenir des macros. Comportement de la propriété : - *obligatoire* si `useip` est défini sur « se connecter en utilisant le nom DNS de l'hôte »
port	string	Numéro de port utilisé par l'interface. Peut contenir des macros utilisateur et LLD. Comportement de la propriété : - *obligatoire*

Propriété	Type	Description
main	integer	Indique si l'interface est utilisée comme interface par défaut sur l'hôte. Une seule interface d'un certain type peut être définie comme interface 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.
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"

Propriété	Type	Description
authpassphrase	string	Phrase secrète d'authentification SNMPv3. Comportement de la propriété : - *pris en charge* si `version` est définie sur "SNMPv3" et `securitylevel` est définie 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 définie 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 définie 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 définie sur "authPriv"
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 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 du rôle utilisateur. Voir User roles 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 du paramètre: - *obligatoire*
ruleid	ID	ID de la règle LLD à laquelle appartient le prototype d'hôte. Comportement du paramètre: - *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 du prototype d'hôte.
interfaces	object/array	Interfaces personnalisées du prototype d'hôte.
templates	object/array	Modèles à lier au prototype d'hôte. Les modèles doivent avoir uniquement la propriété `templateid` définie.

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 également

- Lien du groupe
- Prototype du groupe
- Tag du 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 permet de supprimer des prototypes d'hôte.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateurs *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

`(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)`

Cette 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. 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
hostids	ID/array	Retourner uniquement les prototypes d'hôte avec les ID donnés.
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.

Paramètre	Type	Description
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 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 du rôle utilisateur. Voir User roles pour plus d'informations.

Paramètres

`(object/array)` Propriétés des prototypes 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 des prototypes 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 du paramètre : - *lecture seule* pour les objets hérités
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 du paramètre : - *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 de prototype d'hôte pour remplacer les tags actuels. Tous les tags qui ne sont pas listés dans la requête seront supprimés. Comportement du paramètre : - *lecture seule* pour les objets hérités

Paramètre	Type	Description
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 du paramètre : - *pris en charge* si `custom_interfaces` de Host prototype object 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 également

- Lien du groupe
- Prototype du groupe
- Tag du prototype d'hôte

- Interface personnalisée
- Macro utilisateur

Source

CHostPrototype::update() dans *ui/include/classes/api/services/CHostPrototype.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.

Propriété	Type	Description
proxyid	ID	ID du proxy. Comportement de la propriété : - *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 0
local_port	string	Numéro de port du proxy local auquel se connecter. Les macros utilisateur sont prises en charge. Par défaut : 10051. Comportement de la propriété : - *pris en charge* si `proxy_groupid` n'est pas 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.

Propriété	Type	Description
lastaccess	timestamp	Heure à laquelle le proxy s'est connecté au serveur pour la dernière fois. Comportement de la propriété : - *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. 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.
tls_connect	integer	Connexions à l'hôte. Valeurs possibles : 1 - *(par défaut)* Sans chiffrement ; 2 - PSK ; 4 - certificat.
tls_accept	integer	Connexions depuis l'hôte. Valeurs bitmap possibles : 1 - *(par défaut)* Sans chiffrement ; 2 - PSK ; 4 - certificat. Il s'agit d'un champ de masque de bits ; toute somme des valeurs bitmap possibles est acceptable (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 (dans autoregistration, hosts et proxies). N'incluez pas d'informations sensibles dans l'identité PSK, car elle est envoyée sans chiffrement sur le réseau pour informer le destinataire de la PSK à utiliser. Comportement de la propriété : - *écriture seule* - *obligatoire* si `tls_connect` est défini sur "PSK", ou si `tls_accept` contient le bit "PSK"
tls_psk	string	Clé pré-partagée (PSK) ; doit comporter au moins 32 chiffres hexadécimaux. Comportement de la propriété : - *écriture seule* - *obligatoire* si `tls_connect` est défini sur "PSK", ou si `tls_accept` contient le bit "PSK"

Propriété	Type	Description
custom_timeouts	integer	Indique s'il faut remplacer les délais d'attente globaux des éléments au niveau du proxy. Valeurs possibles : 0 - *(par défaut)* utiliser les paramètres globaux ; 1 - remplacer les délais d'attente.
timeout_zabbix_agent	string	Ne pas consacrer plus de `timeout_*` secondes au traitement. Accepte des secondes ou une unité de temps avec suffixe (par exemple, 30s, 1m). Accepte également les macros utilisateur. Plage de valeurs possible : 1-600s. Par défaut : "". Comportement de la propriété : - *obligatoire* si `custom_timeouts` est défini sur 1.
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		
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 : - *lecture seule*
compatibility	integer	Version du proxy comparée à la version 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 plus ancienne que celle du serveur, mais est partiellement prise en charge) ; 3 - Version non prise en charge (la version du proxy est plus ancienne que la version de la précédente publication LTS du serveur ou la version majeure du serveur est plus ancienne que la version majeure du proxy). Comportement de la propriété : - *lecture seule*
state	integer	État du proxy. Valeurs possibles : 0 - Inconnu ; 1 - Hors ligne ; 2 - En ligne. Comportement de la propriété : - *lecture seule*

proxy.create

Description

`object proxy.create(object/array proxies)`

Cette méthode permet de créer de nouveaux proxies.

> **Note:**
> Cette méthode est uniquement disponible 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)` Proxys à créer.

En plus des propriétés standard du proxy, 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 avoir uniquement la propriété `hostid` définie.

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
}
```

Voir aussi

- Hôte
- Groupe de proxys

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 proxys.

> **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

`(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 donnés.

> **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.

Paramètre	Type	Description
proxyids	ID/array	Retourner uniquement les proxys avec les ID donnés.
proxy_groupids	ID/array	Retourner uniquement les proxys qui appartiennent aux groupes de proxys donnés.
selectAssignedHosts	query	Retourner une propriété `assignedHosts` avec les hôtes assignés au proxy. Prend en charge `count`.
selectHosts	query	Retourner une propriété `hosts` avec les hôtes surveillés par le proxy. Prend en charge `count`.
selectProxyGroup	query	Retourner une propriété `proxyGroup` avec l'objet groupe de proxys.
sortfield	string/array	Trier le résultat selon les propriétés données. 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 proxys

Source

CProxy::get() dans *ui/include/classes/api/services/CProxy.php*.

proxy.update

Description

`object proxy.update(object/array proxies)`

Cette méthode permet de mettre à jour des proxies 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 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 de proxy standard, 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 avoir uniquement la propriété `hostid` définie.

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 proxys

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 :

Propriété	Type	Description
reportid	ID	ID du rapport. Comportement de la propriété : - *lecture seule* - *obligatoire* pour les opérations de mise à jour
userid	ID	ID de l'utilisateur qui a 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	Planification de répétition de la période. Valeurs possibles : 0 - *(par défaut)* quotidien ; 1 - hebdomadaire ; 2 - mensuel ; 3 - annuel.
start_time	integer	Heure de la journée, en secondes, à laquelle le rapport sera préparé pour l'envoi. Par défaut : 0.

Propriété	Type	Description
weekdays	integer	Jours de la semaine pour l'envoi du 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 des valeurs bitmap possibles est acceptée (par exemple, 21 pour lundi, mercredi et vendredi). Par défaut : 0. Comportement de la propriété : - *obligatoire* si `cycle` est défini sur "weekly".
active_since	string	Date de début. Valeurs possibles : chaîne vide - *(par défaut)* non spécifiée (stockée comme 0) ; date spécifique au format YYYY-MM-DD (stockée comme horodatage du début de la journée (00:00:00)).
active_till	string	Date de fin. Valeurs possibles : chaîne vide - *(par défaut)* non spécifiée (stockée comme 0) ; date spécifique au format YYYY-MM-DD (stockée comme horodatage de la fin de la journée (23:59:59)).
subject	string	Sujet du message du rapport.
message	string	Texte du message du rapport.
status	integer	Indique si le rapport est activé ou désactivé. Valeurs possibles : 0 - Désactivé ; 1 - *(par défaut)* Activé.
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 des informations sur l'erreur ; 3 - le rapport a été généré, mais l'envoi à certains (ou à tous les) destinataires a échoué ; "info" contient des informations sur l'erreur. Comportement de la propriété : - *lecture seule*
lastsent	timestamp	Horodatage Unix du dernier rapport envoyé avec succès. Comportement de la propriété : - *lecture seule*
info	string	Description de l'erreur ou informations supplémentaires. Comportement de la propriété : - *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 :

Propriété	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'utilisateurs *Admin* et *Super admin*. Les autorisations permettant d'appeler la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Consultez User roles 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 également

- Utilisateurs
- Groupes d'utilisateurs (object#user_groups)

Source

CReport::create() dans *ui/include/classes/api/services/CReport.php*.

report.delete

Description

`object report.delete(array reportids)`

Cette méthode permet de supprimer des rapports planifiés.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Admin* et *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

`(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 donnés.

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 User roles 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 de rapport

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 d'accompagnement du 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 d'accompagnement du rapport",
            "status": "1",
            "description": "",
            "state": "0",
            "lastsent": "0",
            "info": "",
            "users": [
                {
                    "userid": "1",
                    "access_userid": "1",
                    "exclude": "0"
                }
            ],
            "user_groups": []
        }
    ],
```

```
    "id": 1
}
```

Voir également

- 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 le type 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 des rapports planifiés 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 des rapports planifiés, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
users	object/array	Utilisateurs remplaçant les utilisateurs actuellement affectés au rapport planifié. Comportement du paramètre : - *obligatoire* si `user_groups` n'est pas défini
user_groups	object/array	Groupes d'utilisateurs remplaçant les groupes d'utilisateurs actuellement affectés au rapport planifié. Comportement du paramètre : - *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 également

- Utilisateurs
- Groupes d'utilisateurs (object#user_groups)

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 pris en charge des plages d'adresses IP. 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 des secondes ou une unité de temps avec suffixe (par exemple, 30s, 1m, 2h, 1d), ou une macro utilisateur. 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 d'erreur s'il y a eu des problèmes 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'utilisateurs *Admin* et *Super admin*. Les autorisations permettant d'appeler la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir User roles 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 du paramètre : - *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 également

- Test de découverte

Source

CDRule::create() dans *ui/include/classes/api/services/CDRule.php*.

drule.delete

Description

`object drule.delete(array discoveryRuleIds)`

Cette méthode permet de supprimer des règles de découverte.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateurs *Admin* et *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

`(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)`

Cette méthode permet de récupérer les règles de découverte selon les paramètres donnés.

> **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.

Paramètre	Type	Description
dhostids	ID/array	Retourner uniquement les règles de découverte qui ont créé les hôtes découverts donnés.
druleids	ID/array	Retourner uniquement les règles de découverte ayant les ID donnés.
dserviceids	ID/array	Retourner uniquement les règles de découverte qui ont créé les services découverts donnés.
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`.
sortfield	string/array	Trier le résultat selon les propriétés données. Valeurs possibles : `druleid`, `name`.

Paramètre	Type	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 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 également

- 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 la méthode peuvent être révoquées dans les paramètres du rôle 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	Contrôles de découverte pour remplacer les contrôles existants.

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 également

- Test 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
- Surcharges de règle LLD
  - Filtre de surcharge de règle LLD
    - Condition de filtre de surcharge de règle LLD
  - Opération de surcharge de règle LLD
    - Statut de l'opération de surcharge de règle LLD
    - Découverte de l'opération de surcharge de règle LLD
    - Période de l'opération de surcharge de règle LLD
    - Historique de l'opération de surcharge de règle LLD
    - Tendances de l'opération de surcharge de règle LLD
    - Sévérité de l'opération de surcharge de règle LLD
    - Tag de l'opération de surcharge de règle LLD
    - Modèle de l'opération de surcharge de règle LLD
    - Inventaire de l'opération de surcharge 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 : - *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 des secondes ou une unité de temps avec suffixe (par ex. 30s, 1m, 2h, 1d) et, éventuellement, un ou plusieurs intervalles personnalisés, tous séparés par des points-virgules. Les intervalles personnalisés peuvent être un mélange d'intervalles flexibles et d'intervalles de planification. Accepte les macros utilisateur. Si elles sont utilisées, la valeur doit être une macro unique. Plusieurs macros ou des macros mélangées avec 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 ex. `{$FLEX_INTERVAL}/{$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 : - *constant* - *obligatoire* pour les opérations de création
flags	integer	Origine de la règle de découverte. Valeurs possibles : 1 - une règle de découverte de bas niveau ; 5 - une règle de découverte de bas niveau convertie à partir d'un prototype. Comportement de la propriété : - *lecture seule*
interfaceid	ID	ID de l'interface 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 - *lecture seule* pour les objets hérités
name	string	Nom de la règle LLD. Comportement de la propriété : - *obligatoire* pour les opérations de création - *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 - Élément dépendant ; 19 - HTTP agent ; 20 - SNMP agent ; 21 - Script ; 22 - Browser ; 23 - Imbriqué. Ce type n'est autorisé que 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 a une règle LLD à sa racine). Comportement de la propriété : - *obligatoire* pour les opérations de création - *lecture seule* pour les objets hérités
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" - *lecture seule* pour les objets hérités
allow_traps	integer	Autorise le remplissage de la valeur de manière similaire à l'élément trapper. Valeurs possibles : 0 - *(par défaut)* Ne pas autoriser l'acceptation des données entrantes ; 1 - Autoriser l'acceptation des 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" - *lecture seule* pour les objets hérités (si `type` est défini sur "HTTP agent")
description	string	Description de la règle LLD.

Propriété	Type	Description
error	string	Texte d'erreur en cas de problèmes lors de la mise à jour de la valeur de la règle LLD. Comportement de la propriété : - *lecture seule*
follow_redirects	integer	Suivre les redirections de 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" - *lecture seule* pour les objets hérités
headers	array	Tableau des 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" - *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" - *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" - *lecture seule* pour les objets hérités
jmx_endpoint	string	Chaîne de connexion personnalisée de l'agent JMX. 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 des secondes, une unité de temps avec suffixe ou une macro utilisateur. Par défaut : 7d.
lifetime_type	integer	Scénario de suppression des ressources LLD perdues. Valeurs possibles : 0 - *(par défaut)* Supprimer après que le seuil de durée de vie est atteint ; 1 - Ne pas supprimer ; 2 - Supprimer immédiatement.
enabled_lifetime	string	Période après laquelle les éléments qui ne sont plus découverts seront désactivés. Accepte des secondes, une unité de temps avec suffixe ou une macro utilisateur. Par défaut : 0.
enabled_lifetime_type	integer	Scénario de désactivation des ressources LLD perdues. Valeurs possibles : 0 - Désactiver après que le seuil de durée de vie est atteint ; 1 - Ne pas désactiver ; 2 - *(par défaut)* Désactiver immédiatement.

Propriété	Type	Description
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" - *lecture seule* pour les objets hérités
output_format	integer	Indique si la réponse doit être convertie en JSON. Valeurs possibles : 0 - *(par défaut)* Stocker brut ; 1 - Convertir en JSON. Comportement de la propriété : - *pris en charge* si `type` est défini sur "HTTP agent" - *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 base de données ; - formule pour les règles LLD calculées ; - le script pour les règles LLD de type script et browser. Comportement de la propriété : - *obligatoire* si `type` est défini sur "Database monitor", "SSH agent", "TELNET agent", "Script" ou "Browser" - *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 avec 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" - *lecture seule* pour les objets hérités
password	string	Mot de passe pour l'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" - *lecture seule* pour les objets hérités (si `type` est défini sur "HTTP agent")
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" - *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 "JSON data" ou "XML data" - *pris en charge* si `type` est défini sur "HTTP agent" et que `post_type` est défini sur "Raw data" - *lecture seule* pour les objets hérités

Propriété	Type	Description
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”
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”
query_fields	array	Tableau des 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” - *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” - *lecture seule* pour les objets hérités
retrieve_mode	integer	Quelle partie de la réponse doit être stockée. 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 tous deux 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” - *lecture seule* pour les objets hérités
snmp_oid	string	OID SNMP. Comportement de la propriété : - *obligatoire* si `type` est défini sur ”SNMP agent” - *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” - *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” - *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” - *lecture seule* pour les objets hérités

Propriété	Type	Description
state	integer	État de la règle LLD. Valeurs possibles : 0 - *(par défaut)* normal ; 1 - non pris en charge. Comportement de la propriété : - *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 des codes d'état HTTP requis, séparées par des virgules. Prend également en charge les macros utilisateur comme partie d'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" - *lecture seule* pour les objets hérités
templateid	ID	ID de la règle LLD du modèle parent. Comportement de la propriété : - *lecture seule*
timeout	string	Délai d'expiration de la requête d'interrogation des données de l'élément. Accepte des secondes ou une unité de temps avec suffixe (par ex. 30s, 1m). Accepte également les macros utilisateur. Plage de valeurs possible : 1-600s. Par défaut : "" - utiliser les paramètres proxy/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) - *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 "Allow to accept incoming data"
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" - *lecture seule* pour les objets hérités (si `type` est défini sur "HTTP agent")

Propriété	Type	Description
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 valider que le nom d'hôte de la connexion correspond à celui figurant dans le certificat de l'hôte. Valeurs possibles : 0 - *(par défaut)* Ne pas valider ; 1 - Valider. Comportement de la propriété : - *pris en charge* si `type` est défini sur "HTTP agent" - *lecture seule* pour les objets hérités
verify_peer	integer	Indique s'il faut valider que le certificat de l'hôte est authentique. Valeurs possibles : 0 - *(par défaut)* Ne pas valider ; 1 - Valider. Comportement de la propriété : - *pris en charge* si `type` est défini sur "HTTP agent" - *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*

Propriété	Type	Description
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 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 de filtre : aucune condition ne peut rester inutilisée ou omise. Comportement de la propriété : - *obligatoire* si `evaltype` est défini sur « custom expression »

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*

Propriété	Type	Description
value	string	Valeur à comparer. Comportement de la propriété : - *obligatoire* si `operator` est défini sur « matches regular expression » ou « does not match regular expression »
formulaid	string	ID unique arbitraire utilisé pour référencer la condition à partir d'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 « custom expression »
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*
path	string	Sélecteur de la valeur qui sera attribuée à la macro correspondante. Comportement de la propriété : - *obligatoire*

Prétraitement des règles LLD

L'objet de prétraitement des règles LLD possède les propriétés suivantes.

Propriété	Type	Description
type	integer	Le type d'option de prétraitement. Valeurs possibles : 5 - Expression régulière ; 11 - XML XPath ; 12 - JSONPath ; 14 - Correspond à l'expression régulière ; 15 - Ne correspond pas à l'expression régulière ; 16 - Vérifier une erreur dans JSON ; 17 - Vérifier 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 SNMP walk ; 29 - SNMP walk 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), "XML XPath" (11), "JSONPath" (12), "Correspond à l'expression régulière" (14), "Ne correspond pas à l'expression régulière" (15), "Vérifier une erreur dans JSON" (16), "Vérifier 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 SNMP walk" (28), "SNMP walk vers JSON" (29) ou "Valeur SNMP get" (30)
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), "XML XPath" (11), "JSONPath" (12), "Correspond à l'expression régulière" (14), "Ne correspond pas à l'expression régulière" (15), "Vérifier une erreur dans JSON" (16), "Vérifier une erreur dans XML" (17), "Prometheus vers JSON" (23), "CSV vers JSON" (24), "XML vers JSON" (27), "Valeur SNMP walk" (28), "SNMP walk 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	motif[1]	sortie[2]		0, 1, 2, 3
11	XML XPath	chemin[3]			0, 1, 2, 3
12	JSONPath	chemin[3]			0, 1, 2, 3
14	Correspond à l’expression régulière	motif[1]			0, 1, 2, 3
15	Ne correspond pas à l’expression régulière	motif[1]			0, 1, 2, 3
16	Vérifier une erreur dans JSON	chemin[3]			0, 1, 2, 3
17	Vérifier une erreur dans XML	chemin[3]			0, 1, 2, 3
20	Ignorer les valeurs inchangées avec heartbeat	secondes[4, 5]			
21	JavaScript	script[2]			
23	Prometheus vers JSON	motif[5, 6]			0, 1, 2, 3
24	CSV vers JSON	caractère[2]	caractère[2]	0,1	0, 1, 2, 3
25	Remplacer	chaîne recherchée[2]	remplacement[2]		
27	XML vers JSON				0, 1, 2, 3
28	Valeur SNMP walk	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	SNMP walk vers JSON[7]	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] expression régulière
[2] chaîne
[3] JSONPath ou XML XPath
[4] entier positif (avec prise en charge des suffixes de temps, par ex. 30s, 1m, 2h, 1d)
[5] macro utilisateur
[6] motif Prometheus suivant la syntaxe : `<metric name>{<label name>="<label value>", ...} == <value>`. Chaque composant du motif Prometheus (métrique, nom du label, valeur du label et valeur de la métrique) peut être une macro utilisateur.
[7] Prend en charge plusieurs enregistrements "Nom du champ,Préfixe OID,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 substitution de règle LLD

L'objet de filtre de substitution 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 identifiants 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*
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 identifiants qui référencent des conditions spécifiques du filtre de substitution par leur `formulaid`. Les identifiants 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 « custom expression »

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 »

Propriété	Type	Description
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 substitution de règle LLD

L'opération de substitution de règle LLD est une combinaison de conditions et d'actions à exécuter sur l'objet prototype. Elle possède les propriétés suivantes :

Propriété	Type	Description
operationobject	integer	Type d'objet découvert sur lequel exécuter 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 substitution. 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 de l'opération de substitution pour les objets prototype d'élément, de déclencheur et d'hôte.
opdiscover	object	Objet d'état de découverte de l'opération de substitution (tous les types d'objet).
opperiod	object	Objet de période (intervalle de mise à jour) de l'opération de substitution pour l'objet prototype d'élément.
ophistory	object	Objet d'historique de l'opération de substitution pour l'objet prototype d'élément.
optrends	object	Objet de tendances de l'opération de substitution pour l'objet prototype d'élément.
opseverity	object	Objet de gravité de l'opération de substitution pour l'objet prototype de déclencheur.
optag	object/array	Objet d'étiquette de l'opération de substitution pour les objets prototype de déclencheur et d'hôte.
optemplate	object/array	Objet de modèle de l'opération de substitution pour l'objet prototype d'hôte.
opinventory	object	Objet d'inventaire de l'opération de substitution pour l'objet prototype d'hôte.

Statut de l'opération de substitution de règle LLD

Statut de l'opération de substitution 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*

Opération de découverte de substitution de règle LLD

Statut de découverte de l'opération de substitution 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 d'opération de surcharge de règle LLD

La période d'opération de surcharge de règle LLD est une valeur d'intervalle de mise à jour définie pour l'élément découvert. Elle possède les propriétés suivantes :

Propriété	Type	Description
delay	string	Remplace l'intervalle de mise à jour du prototype d'élément. Accepte des secondes ou une unité de temps avec suffixe (par exemple, 30s, 1m, 2h, 1d) et, éventuellement, un ou plusieurs intervalles personnalisés, tous séparés par des points-virgules. Les intervalles personnalisés peuvent être un mélange d'intervalles flexibles et d'intervalles planifiés. Accepte des macros utilisateur ou des macros LLD. Si elles sont utilisées, la valeur doit être une seule macro. Plusieurs macros ou des macros mélangées avec du texte ne sont pas prises en charge. Les intervalles flexibles peuvent être écrits sous 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*

Historique de l'opération de surcharge de règle LLD

Valeur de l'historique de l'opération de surcharge de règle LLD définie pour l'élément découvert. Elle possède les propriétés suivantes :

Propriété	Type	Description
history	string	Surcharge 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 les macros utilisateur et les macros LLD. Comportement de la propriété : - *obligatoire*

Tendances des opérations de substitution de règle LLD

Valeur des tendances des opérations 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 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 remplacement de règle LLD

Valeur de gravité de l'opération de remplacement 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 - moyen ; 4 - élevé ; 5 - désastre. Comportement de la propriété : - *obligatoire*

Balise d'opération de substitution de règle LLD

L'objet de balise d'opération de substitution de règle LLD contient le nom et la valeur de la balise qui sont définis sur l'objet découvert. Il possède les propriétés suivantes :

Propriété	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 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	Remplace 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 du rôle utilisateur. Voir User roles 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 substitutions 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 également

- 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 LLD.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateurs *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

`(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 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
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	Retourner 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 des hôtes auxquels la règle LLD appartient.
selectItems	query	Retourner une propriété `items` avec les prototypes d'élément de données 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 des macros LLD et des 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.

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 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 la règle LLD appartient.
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 réguliè
            "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 également

- Prototype de graphique
- Hôte
- Prototype d'élément
- Filtre de règle de découverte bas niveau

- 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 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 la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir User roles pour plus d'informations.

Paramètres

`(object/array)` Propriétés de la 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 standard de la règle LLD, 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 du paramètre : - *lecture seule* pour les objets hérités
lld_macro_paths	object/array	Options lld_macro_path de la règle LLD pour remplacer les options lld_macro_path existantes. Comportement du paramètre : - *lecture seule* pour les objets hérités
overrides	object/array	Options de surcharges de règle LLD pour remplacer les options de surcharge existantes. Comportement du paramètre : - *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*.

Requête:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "22450",
        "filter": {
            "evaltype": 1,
            "conditions": [
                {
                    "macro": "{#FSTYPE}",
```

```
                    "value": "@File systems for discovery"
                }
            ]
        }
    },
    "id": 1
}
```

Réponse :

```
{
    "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 répertoire d'utilisateurs possède les propriétés suivantes.

Property	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 utilisateur est définie sur "0" pour tous les utilisateurs liés au répertoire d'utilisateurs supprimé. Comportement de la propriété : - *lecture seule* - *obligatoire* pour les opérations de mise à jour
idp_type	integer	Type du 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 qui contient 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 d'authentification 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") qui contient le prénom de l'utilisateur, utilisé comme valeur de la propriété `name` de l'objet utilisateur 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") qui contient le nom de famille de l'utilisateur, utilisé comme valeur de la propriété `surname` de l'objet utilisateur 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 d'authentification) doit être activé).
provision_groups	array	Tableau d'objets de mappages de groupes de provisionnement pour mapper le modèle de groupe d'utilisateurs LDAP/SAML à un groupe d'utilisateurs Zabbix et à un rôle utilisateur. Comportement de la propriété : - *obligatoire* si `provision_status` est défini sur "Enabled"
provision_media	array	Tableau d'objets de mappages de types de média pour mapper les attributs de média LDAP/SAML de l'utilisateur (par exemple, l'e-mail) aux médias utilisateur Zabbix pour l'envoi de notifications.
Propriétés spécifiques à LDAP :		

Property	Type	Description
name	string	Nom unique du répertoire d'utilisateurs. Comportement de la propriété : - *obligatoire* si `idp_type` est défini sur "User directory of type LDAP"
host	string	Nom d'hôte, IP ou URI du serveur LDAP. L'URI doit contenir le schéma (`ldap://` ou `ldaps://`), l'hôte et le port (facultatif). 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 requête 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 la liaison et la recherche 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"

Property	Type	Description
bind_password	string	Mot de passe LDAP du compte utilisé pour la liaison et la recherche 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"
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 pour 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. Ignoré 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 de groupe (remplacé par la valeur de la propriété `group_member`) ; *%{host}* - nom d'hôte, IP ou URI du serveur LDAP (remplacé par la valeur de la propriété `host`) ; *%{user}* - nom d'utilisateur Zabbix. 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"

Property	Type	Description
group_member	string	Attribut du répertoire d'utilisateurs LDAP qui contient des 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"
group_membership	string	Attribut du répertoire d'utilisateurs LDAP qui contient des 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 requête 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. 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 qui permet 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 des valeurs à partir 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 :		

Property	Type	Description
idp_entityid	string	URI qui identifie le fournisseur d'identité et est utilisée pour communiquer avec le fournisseur d'identité 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"
sp_entityid	string	URL ou toute chaîne qui identifie 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") qui contient le nom d'utilisateur de l'utilisateur, comparé à la valeur de la propriété `username` de l'objet utilisateur 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* Comportement de la propriété : - *obligatoire* si `idp_type` est défini sur "User directory of type SAML"
sso_url	string	URL du service SAML d'authentification unique du fournisseur d'identité, vers laquelle Zabbix enverra les requêtes d'authentification SAML. Exemple : *http://idp.example.com/idp/sso/saml* Comportement de la propriété : - *obligatoire* si `idp_type` est défini sur "User directory of type SAML"
slo_url	string	URL du service SAML de déconnexion unique du fournisseur d'identité, vers laquelle Zabbix enverra les requêtes de déconnexion SAML. Exemple : *https://idp.example.com/idp/slo/saml* Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur "User directory of type SAML"
encrypt_nameid	integer	Indique si l'identifiant de nom SAML doit être chiffré. Valeurs possibles : 0 - *(par défaut)* Ne pas chiffrer l'identifiant de nom ; 1 - Chiffrer l'identifiant de nom. Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur "User directory of type SAML"
encrypt_assertions	integer	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"

Property	Type	Description
nameid_format	string	Format de l'identifiant 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 - *(par défaut)* Ne pas signer les assertions ; 1 - 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 requêtes AuthN SAML doivent être signées avec une signature SAML. Valeurs possibles : 0 - *(par défaut)* Ne pas signer les requêtes AuthN ; 1 - Signer les requêtes 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 - *(par défaut)* Ne pas signer les messages ; 1 - 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 requêtes de déconnexion SAML doivent être signées avec une signature SAML. Valeurs possibles : 0 - *(par défaut)* Ne pas signer les requêtes de déconnexion ; 1 - Signer les requêtes de déconnexion. Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur "User directory of type SAML"

Property	Type	Description
sign_logout_responses	integer	Indique si les réponses de déconnexion SAML doivent être signées avec une signature SAML. Valeurs possibles : 0 - *(par défaut)* Ne pas signer les réponses de déconnexion ; 1 - 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 `$SSO['CERT_STORAGE']` est défini sur `database` dans zabbix.conf.php - *écriture seule*
sp_private_key	string	Contenu de la clé privée du fournisseur de services (SP) pour la configuration du service SAML d'authentification unique (SSO). Fournit une authentification sécurisée et un échange de données avec le fournisseur d'identité (IdP). Exemples : *-----BEGIN CERTIFICATE----- ...more encoded data... -----END CERTIFICATE-----* Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur "User directory of type SAML" et si `$SSO['CERT_STORAGE']` est défini sur `database` dans zabbix.conf.php - *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 `$SSO['CERT_STORAGE']` est défini sur `database` dans zabbix.conf.php - *écriture seule*
idp_certificate_hash	string	Le hachage md5 de la valeur de `idp_certificate`. Renvoie une chaîne vide si `idp_certificate` est vide. Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur "User directory of type SAML" et si `$SSO['CERT_STORAGE']` est défini sur `database` dans zabbix.conf.php - *lecture seule*

Property	Type	Description
sp_private_key_hash	string	Le 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 `$SSO['CERT_STORAGE']` est défini sur `database` dans zabbix.conf.php - *lecture seule*
sp_certificate_hash	string	Le hachage md5 de la valeur de `sp_certificate`. Renvoie une chaîne vide si `sp_certificate` est vide. Comportement de la propriété : - *pris en charge* si `idp_type` est défini sur "User directory of type SAML" et si `$SSO['CERT_STORAGE']` est défini sur `database` dans zabbix.conf.php - *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 ; utilisé comme valeur de la propriété `mediatypeid` de l'objet Media. Comportement de la propriété : - *obligatoire*
attribute	string	Attribut de l'annuaire 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 la 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 lors de la création du média pour l'utilisateur provisionné. Valeurs possibles : 0 - *(par défaut)* activé ; 1 - désactivé.
severity	integer	Valeur de la propriété `severity` du média utilisateur lors de la création du média pour l'utilisateur provisionné. Par défaut : 63.
period	string	Valeur de la propriété `period` du média utilisateur lors de la création du média 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 ex., *Administrateurs Zabbix*) dans l'annuaire d'utilisateurs LDAP/SAML (également SCIM si `scim_status` est défini sur « Le provisionnement SCIM est activé »). 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 (*Utilisateur*, *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 permet de supprimer des annuaires d'utilisateurs. Un annuaire d'utilisateurs ne peut pas être supprimé lorsqu'il est utilisé directement par au moins un groupe d'utilisateurs.
L'annuaire d'utilisateurs LDAP par défaut ne peut pas être supprimé lorsque `authentication.ldap_configured` est défini sur 1 ou lorsqu'il reste d'autres annuaires 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)`

La méthode permet de récupérer les répertoires d'utilisateurs selon les paramètres fournis.

Note:
Cette méthode est disponible uniquement pour les types d'utilisateurs *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 mappages de types de média associés à un répertoire d'utilisateurs.
selectProvisionGroups	query	Retourner une propriété `provision_groups` avec les mappages de groupes de provisionnement associés à un répertoire d'utilisateurs.
sortfield	string/array	Trier le résultat selon les propriétés données. Valeurs possibles : `name`.
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 : `userdirectoryid`, `idp_type`, `provision_status`.

Paramètre	Type	Description
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 permet de tester les paramètres de connexion de l'annuaire d'utilisateurs.

> **Note:**
> Cette méthode permet également de tester quelles données configurées correspondent aux paramètres de l'annuaire 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 annuaire d'utilisateurs dont `provision_status` est défini sur activé.

> **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 permet de mettre à jour des annuaires 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 utilisateur
  - Service
  - Balise 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 utilisateur.
ui.default_access	integer	Indique si l'accès aux nouveaux éléments d'interface utilisateur est activé. Valeurs possibles : 0 - Désactivé ; 1 - *(par défaut)* Activé.
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 `services.read.tag` ; 1 - *(par défaut)* Accès en lecture seule à tous les services.

Propriété	Type	Description
services.read.list	array	Tableau d'objets Service. Les services spécifiés, y compris les services enfants, accorderont un accès en lecture seule au rôle utilisateur. L'accès en lecture seule ne remplacera pas l'accès en lecture-écriture aux services. Comportement de la propriété : - *pris en charge* si `services.read.mode` est défini sur "0"
services.read.tag	object	Tableau d'objets balise de service. Les services correspondant à la balise, y compris les services enfants, accorderont un accès en lecture seule au rôle utilisateur. L'accès en lecture seule ne remplacera pas l'accès en lecture-écriture aux services. Comportement de la propriété : - *pris 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 `services.write.tag` ; 1 - Accès en lecture-écriture à tous les services.
services.write.list	array	Tableau d'objets Service. Les services spécifiés, y compris les services enfants, accorderont un accès en lecture-écriture au rôle utilisateur. L'accès en lecture-écriture remplacera l'accès en lecture seule aux services. Comportement de la propriété : - *pris en charge* si `services.write.mode` est défini sur "0"
services.write.tag	object	Tableau d'objets balise de service. Les services correspondant à la balise, y compris les services enfants, accorderont un accès en lecture-écriture au rôle utilisateur. L'accès en lecture-écriture remplacera l'accès en lecture seule aux services. Comportement de la propriété : - *pris en charge* si `services.write.mode` est défini sur "0"
modules	array	Tableau d'objets module.
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 - Désactivé ; 1 - *(par défaut)* Activé.
api.mode	integer	Mode de traitement des méthodes d'API répertoriées dans la propriété `api`. Valeurs possibles : 0 - *(par défaut)* Liste de refus ; 1 - Liste d'autorisation.
api	array	Tableau de méthodes d'API.
actions	array	Tableau d'objets action.

Propriété	Type	Description
actions.default_access	integer	Indique si l'accès aux nouvelles actions est activé. Valeurs possibles : 0 - Désactivé ; 1 - *(par défaut)* Activé.

Élément d'interface utilisateur

L'objet d'élément d'interface utilisateur possède les propriétés suivantes :

Property	Type	Description
name	string	Nom de l'élément d'interface utilisateur. Valeurs possibles si `type` de l'objet Role est défini sur "User", "Admin" ou "Super admin" : `monitoring.dashboard` - *Tableaux de bord*; `monitoring.problems` - *Supervision → Problèmes*; `monitoring.hosts` - *Supervision → Hôtes*; `monitoring.latest_data` - *Supervision → Dernières données*; `monitoring.maps` - *Supervision → 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` - *Supervision → 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 sur les déclencheurs*; `configuration.service_actions` - *Alertes → Actions → Actions sur les services*; `configuration.discovery_actions` - *Alertes → Actions → Actions de découverte*; `configuration.autoregistration_actions` - *Alertes → Actions → Actions d'auto-enregistrement*; `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 d'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 d'API*; `administration.authentication` - *Utilisateurs → Authentification*; `administration.general` - *Administration → Général*; `administration.audit_log` - *Administration → Journal d'audit*; `administration.housekeeping` - *Administration → Housekeeping*; `administration.proxy_groups` - *Administration → Groupes de proxys*; `administration.proxies` - *Administration → Proxys*; `administration.macros` - *Administration → Macros*; `administration.queue` - *Administration → File d'attente*.

Property	Type	Description
status	integer	Indique si l'accès à l'élément d'interface utilisateur est activé. Valeurs possibles : 0 - Désactivé ; 1 - *(par défaut)* Activé.

Service

Propriété	Type	Description
serviceid	ID	ID du service. Comportement de la propriété : - *obligatoire*

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 :

Propriété	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 sur les problèmes ; `change_severity` - Modifier la sévérité du problème ; `acknowledge_problems` - Acquitter les 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 la création/modification de ses propres médias. Valeurs possibles si `type` de l'objet Role est défini sur "Admin" ou "Super admin" : `edit_maintenance` - Créer et modifier des périodes de maintenance ; `manage_scheduled_reports` - Gérer les rapports planifiés, `manage_sla` - Gérer les SLA. Valeurs possibles si `type` de l'objet Role est défini sur "User" ou "Admin" : `invoke_execute_now` - permet d'exécuter des vérifications d'éléments pour les utilisateurs qui n'ont que des droits de lecture sur l'hôte. Valeurs possibles si `type` de l'objet Role est défini sur "Super admin" : `edit_user_media` - Autoriser la création/modification de médias pour les utilisateurs. Comportement de la propriété : - *obligatoire*
status	integer	Indique si l'accès pour effectuer l'action est activé. Valeurs possibles : 0 - Désactivé ; 1 - *(par défaut)* Activé.

role.create

Description

`object role.create(object/array roles)`

Cette méthode permet de créer de nouveaux rôles.

> **Note:**
> Cette méthode est uniquement disponible 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)` 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 également

- Règles de rôle (object#role_rules)
- [Elément de l'interface utilisateur] (objet#ui_element)
- Module
- Action

Source

CRole::create() dans *ui/include/classes/api/services/CRole.php*.

role.delete

Description

`object role.delete(array roleids)`

Cette méthode permet de supprimer des rôles.

Note:
Cette méthode est uniquement disponible 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

`(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 donnés.

Note:
Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appeler 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
roleids	ID/array	Retourner uniquement les rôles avec les ID indiqués.
selectRules	query	Retourner une propriété `rules` avec les règles du rôle.

Paramètre	Type	Description
selectUsers	query	Retourner une propriété `users` avec les utilisateurs auxquels le rôle est attribué. Voir `user.get` pour les restrictions basées sur le type d'utilisateur.
sortfield	string/array	Trier le résultat selon les propriétés indiquées. Valeurs possibles : `roleid`, `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 données du rôle

Récupérez les données du rôle « Super admin role » et 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",
        "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 également

- 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 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. Voir 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 les 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 : - *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"

Propriété	Type	Description
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	Où exécuter le script. Valeurs possibles : 0 - exécuter sur l'agent Zabbix ; 1 - exécuter sur le serveur Zabbix. Ceci est *pris en charge* uniquement si l'exécution des scripts globaux est activée sur le serveur Zabbix ; 2 - *(par défaut)* exécuter sur le serveur Zabbix ou le proxy. Comportement de la propriété : - *pris en charge* si `type` est défini sur "Script"
menu_path	string	Dossiers séparés par des barres obliques formant une navigation de type menu dans le frontend lors d'un clic sur un hôte ou un événement. Comportement de la propriété : - *pris en charge* si `scope` est défini sur "manual host action" ou "manual event action"
authtype	integer	Méthode d'authentification utilisée pour le type de script SSH. Valeurs possibles : 0 - mot de passe ; 1 - clé publique. Comportement de la propriété : - *pris 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 : - *pris en charge* si `type` est défini sur "SSH" et `authtype` est défini 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` est défini 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` est défini sur "public key"
port	string	Numéro de port utilisé pour les scripts SSH et TELNET. Comportement de la propriété : - *pris en charge* si `type` est défini sur "SSH" ou "TELNET"

Propriété	Type	Description
groupid	ID	ID du groupe d'hôtes sur lequel le script peut être exécuté. S'il est défini sur "0", le script sera disponible pour tous les groupes d'hôtes. Par défaut : 0.
usrgrpid	ID	ID du groupe d'utilisateurs autorisé à exécuter le script. S'il est défini sur "0", le script sera disponible pour tous les groupes d'utilisateurs. Par défaut : 0.
host_access	integer	Comportement de la propriété : - *pris en charge* si `scope` est défini sur "manual host action" ou "manual event action" Autorisations sur l'hôte nécessaires pour exécuter le script. Valeurs possibles : 2 - *(par défaut)* lecture ; 3 - écriture.
confirmation	string	Comportement de la propriété : - *pris en charge* si `scope` est défini sur "manual host action" ou "manual event action" Texte de la fenêtre contextuelle de confirmation. La fenêtre contextuelle apparaîtra lors de la tentative d'exécution du script depuis le frontend Zabbix.
timeout	string	Comportement de la propriété : - *pris en charge* si `scope` est défini sur "manual host action" ou "manual event action" 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. Par défaut : 30s.
parameters	array	Comportement de la propriété : - *obligatoire* si `type` est défini sur "Webhook" Tableau des paramètres d'entrée du webhook.
description	string	Comportement de la propriété : - *pris en charge* si `type` est défini sur "Webhook" 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 : - *pris en charge* si `type` est défini sur "URL"

Propriété	Type	Description
manualinput	integer	Indique si le script accepte une entrée fournie par l'utilisateur. Valeurs possibles : 0 - *(par défaut)* Désactivé ; 1 - Activé ; Comportement de la propriété : - *pris 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 chaîne de caractères 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"
manualinput_validator_type	integer	Détermine le type de saisie utilisateur attendu. Valeurs possibles : 0 - *(par défaut)* Chaîne. Indique que manualinput_validator doit être traité comme une expression régulière ; 1 - Liste. Indique que manualinput_validator doit être traité comme une liste de valeurs d'entrée possibles séparées par des virgules. Comportement de la propriété : - *pris en charge* si `manualinput` est défini sur "Enabled"
manualinput_default_value	string	Valeur par défaut pour le préremplissage de la saisie utilisateur. Comportement de la propriété : - *pris en charge* si `manualinput_validator_type` est défini sur "String"

Paramètres du webhook

Les paramètres transmis au script 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 uniquement disponible 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)` 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 webhook

Créez un script webhook qui envoie une requête HTTP à un service externe.

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "Webhook script",
        "command": "try {\n var request = new HttpRequest(),\n response,\n data;\n\n request.addHeader('Co
        "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 permet de supprimer des scripts.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations permettant d'appeler la méthode peuvent être révoquées dans les paramètres du rôle 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 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 du rôle utilisateur. Voir User roles pour plus d'informations.

Paramètres

`(object)` Paramètres contenant l'ID du script à exécuter, ainsi que 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, 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, 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 une entrée 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'appeler 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
groupids	ID/array	Retourne uniquement les scripts pouvant être exécutés sur les groupes d'hôtes indiqués.
hostids	ID/array	Retourne uniquement les scripts pouvant être exécutés sur les hôtes indiqués.
scriptids	ID/array	Retourne uniquement les scripts ayant les ID indiqués.
usrgrpids	ID/array	Retourne uniquement les scripts pouvant être exécutés par les utilisateurs des groupes d'utilisateurs indiqués.
selectHostGroups	query	Retourne une propriété `hostgroups` contenant les groupes d'hôtes sur lesquels le script peut être exécuté.
selectHosts	query	Retourne une propriété `hosts` contenant les hôtes sur lesquels le script peut être exécuté.
selectActions	query	Retourne une propriété `actions` contenant les actions auxquelles le script est associé.
sortfield	string/array	Trie le résultat selon les propriétés indiquées. Valeurs possibles : `scriptid`, `name`.
countOutput	boolean	Ces paramètres sont décrits dans le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	

Paramètre	Type	Description
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 également

- 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 un ID de script est fourni. Lorsque manualinput est fourni, elle remplace la macro {MANUALINPUT} par la valeur spécifiée.

> **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. Voir Rôles 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'appeler la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir Rôles 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 n'est disponible que 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 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 de scénario web possède les propriétés suivantes.

Propriété	Type	Description
httptestid	ID	ID du scénario web. Comportement de la propriété : - *lecture seule* - *obligatoire* pour les opérations de mise à jour
hostid	ID	ID de l'hôte auquel appartient le scénario web. Comportement de la propriété : - *constant* - *obligatoire* pour les opérations de création
name	string	Nom du scénario web. Comportement de la propriété : - *obligatoire* pour les opérations de création
agent	string	Chaîne d'agent utilisateur qui sera utilisée par le scénario web. Par défaut : Zabbix
authentication	integer	Méthode d'authentification qui sera utilisée par le scénario web. Valeurs possibles : 0 - *(par défaut)* aucune ; 1 - authentification HTTP basique ; 2 - authentification NTLM ; 3 - authentification Kerberos ; 4 - authentification Digest.
delay	string	Intervalle d'exécution du scénario web. Accepte des secondes ou une unité de temps avec suffixe (par exemple, 30s, 1m, 2h, 1d), ou une macro utilisateur. 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 basique, NTLM, Kerberos ou Digest.
http_proxy	string	Proxy qui sera utilisé par le scénario web, indiqué sous la forme *http://[username[:password]@]proxy.example.com[:port]*.
http_user	string	Nom d'utilisateur utilisé pour l'authentification HTTP basique, NTLM, Kerberos ou Digest.
retries	integer	Nombre de fois qu'un scénario web tentera d'exécuter chaque étape avant d'échouer. 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 : - *lecture seule*
variables	array	Variables du scénario web.

Propriété	Type	Description
verify_host	integer	Indique s'il faut valider que le nom d'hôte de la connexion correspond à celui figurant dans le certificat de l'hôte. Valeurs possibles : 0 - *(par défaut)* ignorer la vérification de l'hôte ; 1 - vérifier l'hôte.
verify_peer	integer	Indique s'il faut valider que le certificat de l'hôte est authentique. Valeurs possibles : 0 - *(par défaut)* ignorer la vérification du 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 de scénario

L'objet d'étape de scénario définit une vérification spécifique d'un scénario web. Il possède les propriétés suivantes.

Propriété	Type	Description
name	string	Nom de l'étape de scénario. Comportement de la propriété : - *obligatoire*
no	integer	Numéro de séquence de l'étape dans un scénario web. Comportement de la propriété : - *obligatoire*
url	string	URL à vérifier. Comportement de la propriété : - *obligatoire*
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 de scénario remplaceront les en-têtes spécifiés pour le scénario web.
posts	string/array	Variables HTTP POST sous forme de chaîne (données POST brutes) ou sous forme 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 de 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 des secondes, une unité de temps avec suffixe ou une macro utilisateur. Par défaut : 15s. Maximum : 1h. Minimum : 1s.
variables	array	Variables de l'étape de 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 paramètres de requête. Il possède les propriétés suivantes.

Propriété	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 permet de créer de nouveaux scénarios web.

Note:
La création d'un scénario web créera automatiquement un ensemble d' éléments de supervision 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. Voir 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	Balises 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 également

- Etape de scenario

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 permettant d'appeler cette méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir User roles 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)`

Cette méthode permet de récupérer des scénarios web selon les paramètres donnés.

> **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.

Paramètre	Type	Description
groupids	ID/array	Retourner uniquement les scénarios web appartenant aux groupes d'hôtes indiqués.
hostids	ID/array	Retourner uniquement les scénarios web appartenant aux hôtes indiqués.
httptestids	ID/array	Retourner uniquement les scénarios web ayant les ID indiqués.
inherited	boolean	Si défini sur `true`, retourner uniquement les scénarios web hérités d'un modèle.
inheritedTags	boolean	Retourner les scénarios web qui possèdent les `tags` indiqués également dans le modèle/l'hôte/les modèles liés. Valeurs possibles : `true` - le modèle/l'hôte/les modèles liés doivent également posséder les tags indiqués ; `false` - *(par défaut)* les tags du modèle/de l'hôte/des modèles liés sont ignorés.
monitored	boolean	Si défini sur `true`, retourner uniquement les scénarios web activés appartenant 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 appartenant à des modèles.
templateids	ID/array	Retourner uniquement les scénarios web appartenant aux modèles indiqués.
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 indiqués. Format : `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Un tableau vide retourne tous les scénarios web. Valeurs possibles de l'operator : 0 - *(par défaut)* Contient ; 1 - Égal à ; 2 - Ne contient pas ; 3 - Différent de ; 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 indiquées. 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	

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 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 également

- Hôte
- Etape de scenario

Source

CHttpTest::get() dans *ui/include/classes/api/services/CHttpTest.php*.

httptest.update

Description

`object httptest.update(object/array webScenarios)`

Cette méthode 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 permettant d'appeler la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Voir User roles 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.

Paramètre	Type	Description
steps	array	Étapes du scénario pour remplacer les étapes existantes.
tags	array	Balises 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 également

- Etape de scenario

Source

CHttpTest::update() dans *ui/include/classes/api/services/CHttpTest.php*.

Service

Cette classe est conçue pour fonctionner avec les services métier/l'infrastructure informatique.

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.

Propriété	Type	Description
serviceid	ID	ID du service. Comportement de la propriété : - *lecture seule* - *obligatoire* pour les opérations de mise à jour
algorithm	integer	Règle de calcul du statut. Applicable uniquement si des services enfants existent. Valeurs possibles : 0 - définir le statut sur OK ; 1 - le plus critique si tous les enfants ont des problèmes ; 2 - le plus critique des services enfants. Comportement de la propriété : - *obligatoire* pour les opérations de création
name	string	Nom du service. Comportement de la propriété : - *obligatoire* pour les opérations de création
sortorder	integer	Position du service utilisée pour le tri. Valeurs possibles : 0-999. Comportement de la propriété : - *obligatoire* pour les opérations de création
weight	integer	Poids du service. Valeurs possibles : 0-1000000. Par défaut : 0.
propagation_rule	integer	Règle de propagation du statut. Valeurs possibles : 0 - *(par défaut)* propager le statut du service tel quel, sans aucune modification ; 1 - augmenter le statut propagé d'une valeur `propagation_value` donnée (de 1 à 5 niveaux de sévérité) ; 2 - diminuer le statut propagé d'une valeur `propagation_value` donnée (de 1 à 5 niveaux de sévérité) ; 3 - ignorer ce service - le statut n'est pas propagé au service parent ; 4 - définir un statut de service fixe à l'aide d'une valeur `propagation_value` donnée. Comportement de la propriété : - *obligatoire* si `propagation_value` est défini

Propriété	Type	Description
propagation_value	integer	Valeur de propagation du statut. 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. Comportement de la propriété : - *obligatoire* si `propagation_rule` est défini
status	integer	Indique si le service est dans l'état OK ou dans un état de problème. Si le service est dans un état de problème, `status` est égal soit à : - la sévérité du problème le plus critique ; - le statut le plus élevé d'un service enfant en état de problème. Si le service est dans l'état OK, `status` est égal à : -1. Comportement de la propriété : - *lecture seule*
description	string	Description du service.
uuid	string	Identifiant unique universel, utilisé pour lier les services importés à ceux qui existent déjà. 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. Comportement de la propriété : - *lecture seule*

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 d'au moins (W) ; 5 - si le poids des services enfants ayant le statut (Statut) ou supérieur est d'au moins (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 de la propriété : - *obligatoire*
limit_value	integer	Valeur limite. Valeurs possibles : - pour N et W : 1-100000 ; - pour N% : 1-100. Comportement de la propriété : - *obligatoire*
limit_status	integer	Statut limite. Valeurs possibles : -1 - OK ; 0 - Non classé ; 1 - Information ; 2 - Avertissement ; 3 - Moyen ; 4 - Élevé ; 5 - Désastre. Comportement de la propriété : - *obligatoire*
new_status	integer	Valeur du nouveau statut. Valeurs possibles : 0 - Non classé ; 1 - Information ; 2 - Avertissement ; 3 - Moyen ; 4 - Élevé ; 5 - Désastre. Comportement de la propriété : - *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. Ils possèdent les propriétés suivantes.

Propriété	Type	Description
clock	timestamp	Heure à laquelle le changement d'état du service s'est produit.
value	integer	Statut du service. Consultez la propriété `status` du service pour obtenir la liste des valeurs possibles.

Tag de problème

Les tags de problème permettent de lier des services à des événements de problème. L'objet de tag de problème possède les propriétés suivantes.

Propriété	Type	Description
tag	string	Nom du tag de problème. Comportement de la propriété : - *obligatoire*
operator	integer	Opérateur de condition de correspondance. Valeurs possibles : 0 - *(par défaut)* Égal à ; 2 - Contient.
value	string	Valeur du tag de problème.

service.create

Description

`object service.create(object/array services)`

Cette méthode permet de créer de nouveaux services.

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. Voir User roles 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 permet de supprimer des services.

> **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. Voir User roles 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 donnés.

> **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. 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
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)* And/Or ; 2 - Or.
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 à ; 2 - Ne contient pas ; 3 - Différent de ; 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 à ; 2 - Ne contient pas ; 3 - Différent de ; 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 d'état. 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 cours 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 permet de mettre à jour des services existants.

> **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. Voir User roles 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. Les services enfants doivent avoir uniquement la propriété `serviceid` définie.
parents	array	Services parents remplaçant les services parents actuels. Les services parents doivent avoir uniquement la propriété `serviceid` définie.
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 d'agent Zabbix, SNMPv1, SNMPv2 ou SNMPv3.

dservice.get

Description

`integer/array dservice.get(object parameters)`

Cette méthode permet de récupérer les services découverts selon les paramètres donnés.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les permissions pour appeler la méthode peuvent être révoquées dans les paramètres des rôles 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
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 également

- Hôte découvert
- Test 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 estimer 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 minimal acceptable, exprimé en pourcentage. Si l'indicateur de niveau de service (SLI) descend en dessous de cette valeur, 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 : timestamp. Comportement de la propriété: - *obligatoire*
period_to	integer	Heure de fin de la période d'indisponibilité exclue (exclusive). Valeurs possibles : timestamp. 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 permet de créer de nouveaux objets SLA.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateurs *Admin* et *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`) Objets SLA à créer.

En plus des propriétés SLA standard, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
service_tags	array	Balises 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'indisponibilité 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 permet de supprimer des entrées de SLA.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateurs *Admin* et *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 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. 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
slaids	ID/array	Retourner uniquement les SLA avec les ID donnés.
serviceids	ID/array	Retourner uniquement les SLA correspondant aux services spécifiés.
selectSchedule	query	Retourner une propriété `schedule` avec les planifications des SLA. Prend en charge `count`.
selectExcludedDowntimes	query	Retourner une propriété `excluded_downtimes` avec les temps d'arrêt exclus des SLA. Prend en charge `count`.
selectServiceTags	query	Retourner une propriété `service_tags` avec les tags de service des SLA. Prend en charge `count`.
sortfield	string/array	Trier 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 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 permettant d'appeler la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir 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 des paramètres : - *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 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 de rapport et service comprennent :

Propriété	Type	Description
uptime	integer	Durée pendant laquelle le service est resté dans un état *OK* pendant le temps de fonctionnement 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 fonctionnement planifié, moins les temps d'arrêt exclus.
sli	float	SLI (pourcentage du temps de fonctionnement total), basé sur uptime et downtime.
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 permet de mettre à jour des entrées de SLA 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. Voir User roles 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. La spécification du paramètre comme vide sera interprétée comme une planification 24x7.
excluded_downtimes	array	Périodes d'indisponibilité exclues du 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*.

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.

Propriété	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 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	Période d'affichage par défaut de la page (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 de tableau de bord

L'objet de page de tableau de bord possède les propriétés suivantes.

Propriété	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	Période d'affichage de la page de tableau de bord (en secondes). Valeurs possibles : 0, 10, 30, 60, 120, 600, 1800, 3600. Par défaut : 0 (utilisera la période d'affichage de page par défaut).
widgets	array	Tableau des objets widget de 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 - Géocarte ; 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 ; itemhistory - Historique d'élément ; itemnavigator - Navigateur d'éléments ; item - Valeur d'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 - Vue d'ensemble des déclencheurs ; url - URL ; web - Supervision web. Comportement de la propriété: - *requis*
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 ;

Propriété	Type	Description
fields	array	Tableau d'objets champ de widget de tableau de bord. Comportement de la propriété: - voir les widgets individuels dans Champs de widget 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. Elle possède les propriétés suivantes.

Propriété	Type	Description
usrgrpid	ID	ID du groupe d'utilisateurs. 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*

Utilisateur du tableau de bord

Liste des autorisations du tableau de bord basées sur les 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 permet de créer de nouveaux tableaux de bord.

> **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. Voir User roles pour plus d'informations.

Paramètres

`(object/array)` Tableaux de bord à créer.

En plus des propriétés standard des tableaux 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 du paramètre : - *obligatoire*
users	array	Partages d'utilisateur de tableau de bord à créer sur le tableau de bord.
userGroups	array	Partages de groupe d'utilisateurs de 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 également

- Page de tableau de bord
- Widget de tableau de bord
- Champ de widget de tableau de bord
- Utilisateur de tableau de bord
- Groupe d'utilisateurs de tableau de bord

Source

CDashboard::create() dans *ui/include/classes/api/services/CDashboard.php*.

dashboard.delete

Description

`object dashboard.delete(array dashboardids)`

Cette méthode permet de supprimer des tableaux de bord.

> **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. Voir User roles 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)`

La méthode permet de récupérer des tableaux de bord en fonction des paramètres donnés.

Note:
Cette méthode est disponible pour tout type d'utilisateurs. Les autorisations d'appel de la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir les 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 également

- Page de tableau de bord
- Widget de tableau de bord
- Champ de widget de tableau de bord
- Utilisateur de tableau de bord
- Groupe d'utilisateurs de 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 permet de mettre à jour des tableaux de bord existants.

> **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. Voir User roles 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 ne possédant pas la 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 des utilisateurs du tableau de bord pour remplacer les éléments existants.
userGroups	array	Partages des groupes 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 également

- Page de tableau de bord
- Widget de tableau de bord
- Champ de widget de tableau de bord
- Utilisateur de tableau de bord
- Groupe d'utilisateurs de 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 du widget de tableau de bord respectifs 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 réussite de la création ou de la mise à jour du widget *Journal des actions*, veuillez vous référer 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 - Pas d'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.
Statut	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	N'importe quelle valeur de chaîne.
Période	1	time_period._referenc	`DASHBOARD._timeperiod` - définit le sélecteur de période 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 uniquement dans les paramètres *From* et *To*.
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 *Time period* n'est pas défini - *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 *Time period* n'est pas défini - *obligatoire* si `time_period.from` est défini

Paramètre	type	name	value
Trier les entrées par	0	sort_triggers	3 - Heure (croissant) ; 4 - *(par défaut)* Heure (décroissant) ; 5 - Type (croissant) ; 6 - Type (décroissant) ; 7 - Statut (croissant) ; 8 - Statut (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 du widget de tableau de bord correspondants permettent de configurer le widget *Clock* 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 *Clock*, veuillez vous référer 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 du widget de tableau de bord respectifs permettent de configurer le widget *Statu de 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 respectifs 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 du widget de tableau de bord respectifs permettent de configurer le widget *Cartes préférées* 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 du widget de tableau de bord correspondants permettent de configurer le widget *Gauge* 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 *Gauge*, veuillez vous référer 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 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 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 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 - Haut ; 1 - *(par défaut)* 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 - Désactivé ; 1 - *(par défaut)* Activé.
Unités (valeur)	1	units	Toute valeur de chaîne. Comportement du paramètre : - *pris en charge* si *Unités* (case à cocher) est défini sur ”Activé”
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 ”Activé”
Gras	0	units_bold	0 - *(par défaut)* Désactivé ; 1 - Activé. Comportement du paramètre : - *pris en charge* si *Unités* (case à cocher) est défini sur ”Activé”
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 ”Activé” 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 pour *Show* avec la valeur ”Value arc” est défini, ou si *Show arc* est défini sur ”Activé”
Échelle			

Paramètre	type	name	value
Afficher les unités	0	scale_show_units	0 - Désactivé ; 1 - *(par défaut)* Activé. Comportement du paramètre : - *pris en charge* si *Unités* (case à cocher) est défini sur ”Activé” et si un objet de champ de widget de tableau de bord pour *Show* avec la valeur ”Value arc” est défini, ou si *Show 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 pour *Show* avec la valeur ”Value arc” est défini, ou si *Show 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 pour *Show* avec la valeur ”Value arc” est défini, ou si *Show 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 étiquettes	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 pour *Show* avec la valeur ”Value arc” est défini ou si *Show 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 du widget de tableau de bord correspondants 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 réussite de la création ou de la mise à jour du widget *Geomap*, veuillez vous référer au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Geomap*.

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			
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.

Paramètre	type	name	value
Nom du tag	1	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* 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 - 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 du tag dans la liste d'évaluation des tags. Comportement du paramètre : - *obligatoire* lors de la configuration de *Tags* 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	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* Ce paramètre n'est pas pris en charge lors de la configuration du widget sur un tableau de bord de modèle.
Vue initiale	1	default_view	*Latitude*, *longitude*, *niveau de zoom* (*facultatif*, valeurs possibles de 0 à 30) séparés par des virgules. Exemple : `40.6892494,-74.0466891,10`.
Regroupement			
Mode de regroupement	0	clustering_mode	0 - *(par défaut)* Auto ; 1 - Niveau de zoom.
Niveau de zoom du regroupement	0	clustering_zoom_level	Les valeurs possibles vont de 0 à 30. Par défaut : 0. Comportement du paramètre : - *pris en charge* si *Mode de regroupement* est défini sur "Niveau de zoom"
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 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 du widget de tableau de bord respectifs permettent de configurer le widget *Graph* 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 *Graph*, veuillez vous référer 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*

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 ex. ds.0.hosts.0, ds.0.items.0) représente le jeu de données concerné, tandis que le second nombre, s'il est présent, représente l'hôte ou l'élément configuré.

Paramètre	type	name	value
Type de jeu de don-nées	0	ds.0.dataset_type	0 - Liste d'éléments ; 1 - *(par défaut)* Modèles d'éléments.
Éléments	4	ds.0.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 *Éléments (widget)* n'est pas défini

Paramètre	type	name	value
Éléments (widget)	1	ds.0.itemids.0._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. 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 *Éléments* n'est pas défini
Couleur	1	ds.0.color.0	Code couleur hexadécimal (par ex. `FF0000`). Comportement du paramètre : - *obligatoire* si *Type de jeu de données* est défini sur "Liste d'éléments"
Modèles d'hôte	1	ds.0.hosts.0	Nom ou modèle d'hôte (par ex. "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	1	ds.0.items.0	Nom ou modèle d'élément (par ex. "*: 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"
Couleur	1	ds.0.color	Code couleur hexadécimal (par ex. `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. Les valeurs possibles vont de 0 à 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
Tracer	0	ds.0.type	0 - *(par défaut)* Ligne ; 1 - Points ; 2 - Escalier ; 3 - Barre.
Empilé	0	ds.0.stacked	0 - *(par défaut)* Désactivé ; 1 - Activé. Comportement du paramètre : - *pris en charge* si *Tracer* est défini sur "Ligne", "Escalier" ou "Barre"

Paramètre	type	name	value
Largeur	0	ds.0.width	Les valeurs possibles vont de 1 à 10. Par défaut : 1. Comportement du paramètre : - *pris en charge* si *Tracer* est défini sur ”Ligne” ou ”Escalier”
Taille des points	0	ds.0.pointsize	Les valeurs possibles vont de 1 à 10. Par défaut : 3. Comportement du paramètre : - *pris en charge* si *Tracer* est défini sur ”Points”
Transparence	0	ds.0.transparency	Les valeurs possibles vont de 1 à 10. Par défaut : 5.
Remplissage	0	ds.0.fill	Les valeurs possibles vont de 1 à 10. Par défaut : 3. Comportement du paramètre : - *pris en charge* si *Tracer* est défini sur ”Ligne” ou ”Escalier”
Données manquantes	0	ds.0.missingdatafunc	0 - *(par défaut)* Aucune ; 1 - Connectées ; 2 - Traiter comme 0 ; 3 - Dernière valeur connue. Comportement du paramètre : - *pris en charge* si *Tracer* est défini sur ”Ligne” ou ”Escalier”
Remplacer l’hôte	1	ds.0.override_hostid.ABCDE	`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.
Axe Y	0	ds.0.axisy	0 - *(par défaut)* Gauche ; 1 - Droite.
Décalage temporel	1	ds.0.timeshift	Chaîne temporelle valide (par ex. 3600, 1h, etc.). Vous pouvez utiliser des suffixes temporels. Les valeurs négatives sont également autorisées. 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 ex. 3600, 1h, etc.). Vous pouvez utiliser des suffixes temporels. Par défaut : 1h.
Agréger	0	ds.0.aggregate_grouping	0 - *(par défaut)* Chaque élément ; 1 - Jeu de données. 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
Approximation	0	ds.0.approximation	1 - min ; 2 - *(par défaut)* avg ; 4 - max ; 7 - all.
Libellé du jeu de données Tags d'élément	1	ds.0.data_set_label	N'importe quelle valeur de chaîne. Par défaut : "" (vide).
Type d'évaluation	0	ds.0.item_tags_evaltype	0 - *(par défaut)* Et/Ou ; 2 - Ou.
Nom du tag	1	ds.0.item_tags.0.tag	N'importe quelle valeur de chaîne. Remarque : le nombre 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'élément*
Opérateur	0	ds.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 nombre 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'élément*
Valeur du tag	1	ds.0.item_tags.0.value	N'importe quelle valeur de chaîne. Remarque : le nombre 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'élément*

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.
Déclencheurs simples	0	simple_triggers	0 - *(par défaut)* Désactivé ; 1 - Activé.
Temps de travail	0	working_time	0 - *(par défaut)* Désactivé ; 1 - Activé.

Paramètre	type	name	value
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.
Ligne de percentile (gauche)			
Statut	0	percentile_left	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 "Gauche"
Valeur	0	percentile_left_value	Les valeurs possibles vont de 1 à 100. Comportement du paramètre : - *pris en charge* si *Axe Y* (dans la configuration de *Jeu de données*) est défini sur "Gauche"
Ligne de percentile (droite)			
Statut	0	percentile_right	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"
Valeur	0	percentile_right_value	Les valeurs possibles vont de 1 à 100. Comportement du paramètre : - *pris en charge* si *Axe Y* (dans la configuration de *Jeu 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*.

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
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	0 - *(par défaut)* Linéaire; 1 - Logarithmique.
		righty_scale	
Min	1	lefty_min	Toute valeur numérique. Par défaut : "" (vide).
		righty_min	
Max	1	lefty_max	Toute valeur numérique. Par défaut : "" (vide).
		righty_max	
Unités (type)	0	lefty_units	0 - *(par défaut)* Auto; 1 - Statique.
		righty_units	
Unités (valeur)	1	lefty_static_units	Toute valeur de chaîne. Par défaut : "" (vide).
		righty_static_units	
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 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. 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. Par défaut : 4. Comportement du paramètre: - *pris en charge* si *Afficher la légende* est défini sur ”Activé”, et *Afficher 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.
Hôtes des problèmes	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.

Paramètre	type	name	value
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 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 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 propriété (par ex. or.0.hosts.0, or.0.items.0) représente l'ensemble de données particulier, tandis que le second nombre, s'il est présent, représente l'hôte ou l'élément configuré.

Paramètre	type	name	value
Modèles d'hôte	1	or.0.hosts.0	Nom ou modèle d'hôte (par ex. `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*
Modèles d'élément	1	or.0.items.0	Nom ou modèle d'élément (par ex. `*: 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* lors de la configuration des *Remplacements*
Couleur de base	1	or.0.color	Code couleur hexadécimal (par ex. `FF0000`).
Largeur	0	or.0.width	Les valeurs possibles vont de 1 à 10.
Tracer	0	or.0.type	0 - Ligne ; 1 - Points ; 2 - Escalier ; 3 - Barre.
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 - Traiter comme 0 ; 3 - Dernière valeur connue.
Axe Y	0	or.0.axisy	0 - Gauche ; 1 - Droite.
Décalage temporel	1	or.0.timeshift	Chaîne de temps valide (par ex. `3600`, `1h`, etc.). Vous pouvez utiliser les suffixes de temps. 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 du widget de tableau de bord correspondants permettent de configurer le widget *Graph (classic)* 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 cela 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 *Graph (classic)*, veuillez vous référer 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
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 *De* et *À*.

Paramètre	type	name	value
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._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.
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`

10 Prototype de graphique

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ du widget de tableau de bord respectifs 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 réussite de la création ou de la mise à jour du widget *Graph prototype*, veuillez vous référer 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 définie sur « Prototype de graphique »
Prototype d'élément	5	itemid.0	ID du prototype d'élément. Comportement du paramètre : - *obligatoire* si *Source* est définie 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. 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 définie - *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 définie - *obligatoire* si `time_period.from` est défini
Afficher la lé-gende	0	show_legend	0 - Désactivé ; 1 - *(par défaut)* Activé.

Paramètre	type	name	value
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.
Colonnes	0	columns	Les valeurs possibles vont de 1 à 24. Par défaut : 2.
Lignes	0	rows	Les valeurs possibles vont de 1 à 16. Par défaut : 1.
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 *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 du widget de tableau de bord correspondants 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 référer 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.
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.

Paramètre	type	name	value
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*
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*

Paramètre	type	name	value
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 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 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.
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 "Text"
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 "Value"
Taille (type)	0	primary_label_size_type	0 - *(par défaut)* Auto; 1 - Personnalisée.

Paramètre	type	name	value
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 ”Custom”
Gras	0	primary_label_bold	0 - *(par défaut)* Désactivé; 1 - Activé.
Couleur	1	primary_label_color	Code couleur hexadécimal (par exemple, FF0000). Par défaut : basé sur `theme` de l’objet Settings et l’objet User : 1F2C33 pour ”blue-theme” ou ”hc-light” ; EEEEEE pour ”dark-theme” ou ”hc-dark”.
Unités (case à cocher)	0	primary_label_units_show	0 - Désactivé; 1 - *(par défaut)* Activé. Comportement du paramètre : - *pris en charge* si *Type* est défini sur ”Value”
Unités (valeur)	1	primary_label_units	Toute valeur de chaîne. "" (vide) Comportement du paramètre : - *pris en charge* si *Type* est défini sur ”Value” et *Unités* (case à cocher) est défini sur ”Activé”
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 ”Value” et *Unités* (case à cocher) est défini sur ”Activé” 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.
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 ”Text”
Décimales	0	secondary_label_decimals	Les valeurs possibles vont de 0 à 6. Par défaut : 2. Comportement du paramètre : - *pris en charge* si *Type* est défini sur ”Value”
Taille (type)	0	secondary_label_size_type	0 - *(par défaut)* Auto; 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 ”Custom”
Gras	0	secondary_label_bold	0 - Désactivé; 1 - *(par défaut)* Activé.

Paramètre	type	name	value
Couleur	1	secondary_label_color	Code couleur hexadécimal (par exemple, FF0000). Par défaut : basé sur theme de l'objet Settings et l'objet User : 1F2C33 pour "blue-theme" ou "hc-light" ; EEEEEE pour "dark-theme" ou "hc-dark".
Unités (case à cocher)	0	secondary_label_units_show	0 - Désactivé; 1 - *(par défaut)* Activé. Comportement du paramètre : - *pris en charge* si *Type* est défini sur "Value"
Unités (valeur)	1	secondary_label_units	Toute valeur de chaîne. "" (vide) Comportement du paramètre : - *pris en charge* si *Type* est défini sur "Value" et *Unités* (case à cocher) est défini sur "Activé"
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 "Value" et *Unités* (case à cocher) est défini sur "Activé" 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 l'objet User : D9E7ED pour "blue-theme" ; 3D5059 pour "dark-theme" ; AAD7E9 pour "hc-light" ; 335463 pour "hc-dark".
Seuils			
Interpolation des couleurs	0	interpolation	0 - Désactivé; 1 - *(par défaut)* Activé.
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 du widget de tableau de bord respectifs permettent de configurer le widget *Disponibilité de l'hôte* 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.

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 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.
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 du widget de tableau de bord correspondants permettent de configurer le widget *Host card* 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 *Host card*, veuillez vous référer 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.
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.

Paramètre	type	name	value
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 du widget de tableau de bord respectifs permettent de configurer le widget *Navigateur d'hôte* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés des « champs » 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 création ou de mise à jour de widgets. incorrectement. Pour garantir la création ou la mise à jour réussie du widget *Navigateur d'hôte*, veuillez vous référer 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 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.

Paramètre	type	name	value
Modèles d'hôtes	1	hosts.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 propriété. Pour plusieurs modèles d'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 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.
Statut 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.
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 des paramètres : - *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 propriété fait référence à l'ordre du tag dans la liste d'évaluation des tags. Comportement des paramètres : - *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 des paramètres : - *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
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.
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 du tag; 2 - Gravité. Remarque : le numéro dans le nom de propriété fait référence à l'ordre des attributs dans la liste des attributs de regroupement. Comportement des paramètres : - *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 propriété fait référence à l'attribut de regroupement défini dans le paramètre *Attribut*. Comportement des paramètres : - *obligatoire* lors de la configuration de *Regrouper par* et si *Attribut* est défini sur "Valeur du tag"
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 dans le tableau de bord auquel appartient le widget. Comportement des paramètres : - *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 du widget de tableau de bord respectifs 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 réussite de la création ou de la mise à jour du widget *Carte d'élément*, veuillez vous référer 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

Paramètre	type	name	value
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.

Mini-graphique

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. Par défaut : 1.
Remplissage	0	sparkline.fill	Les valeurs possibles vont de 0 à 10. Par défaut : 3.
Couleur	1	sparkline.color	Code couleur hexadécimal (par exemple `FF0000`). 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` égal à `ABCDE`) comme source de données. Par défaut : `""` (vide) Vous pouvez également définir la période uniquement dans les paramètres *De* et *À*.

Paramètre	type	name	value
De	1	sparkline.time_period.from	Chaîne temporelle valide en syntaxe absolue (`YYYY-MM-DD hh:mm:ss`) ou relative (`now`, `now/d`, `now/w-1w`, etc.). Par défaut : now-1h. Comportement du paramètre : - *pris en charge* si *Période* n'est pas défini - *obligatoire* si `sparkline.time_period.to` est défini
À	1	sparkline.time_period.to	Chaîne temporelle valide en syntaxe absolue (`YYYY-MM-DD hh:mm:ss`) ou relative (`now`, `now/d`, `now/w-1w`, etc.). 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)* Auto ; 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 du widget de tableau de bord respectifs 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 réussite de la création ou de la mise à jour du widget *Historique des éléments*, veuillez vous référer 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 - Pas d'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. Par défaut : 25.
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 (voir ci-dessous)			
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*

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*

Parameter	type	name	value
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 de type caractère, texte ou journal.

Paramètre	type	name	value
Mises en évidence			
Mise en évidence	1	columns.0.highlights.color	Code couleur hexadécimal (par ex. FF0000).
Seuil	1	columns.0.highlights.pattern	Toute expression régulière.
Affichage	0	columns.0.display	1 - *(par défaut)* Tel quel ; 4 - HTML ; 5 - Ligne unique.
Ligne unique	0	columns.0.max_length	Les valeurs possibles vont de 1 à 500. Par défaut : 100. Comportement du paramètre : - *pris en charge* si *Affichage* est défini sur « Ligne unique »

Paramètre	type	name	value
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 l'*élément* est défini sur un élément de type journal et que *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
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 du widget de tableau de bord respectifs permettent de configurer le widget *Navigateur d'hôte* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés des « champs » 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 création ou une mise à jour widgets erronée. Pour garantir la création ou la mise à jour valide du widget *Navigateur d'hôte*, veuillez vous référer 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 - 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 *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.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.
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 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 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.
Modèles d'élément	1	items.0	Nom ou modèle d'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 propriété.
Tags d'élément			
Type d'évaluation	0	item_tags_evaltype	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 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*
Opérateur	0	item_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 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 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
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.
Grouper par			
Attribut	0	group_by.0.attribute	0 - Groupe d'hôtes ; 1 - Nom de l'hôte ; 2 - Valeur du tag d'hôte ; 3 - Valeur du tag d'élément. Remarque : Le numéro dans le nom de 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 *Grouper par*
Valeur	1	group_by.0.tag_name	Toute valeur de chaîne. Remarque : Le numéro dans le nom de propriété fait référence à l'attribut de regroupement défini dans le paramètre *Attribut*. Comportement du paramètre : - *obligatoire* lors de la configuration de *Grouper par* et si *Attribut* est défini sur "Valeur du tag d'hôte" ou "Valeur du tag d'élément"
Limite d'éléments	0	show_lines	Les valeurs possibles vont de 1 à 9999. Par défaut : 100.
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 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 du widget de tableau de bord respectifs permettent de configurer le widget *Valeur 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 réussite de la création ou de la mise à jour du widget *Valeur d'élément*, veuillez vous référer 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.
Élément (widget)	1	itemid._reference	Comportement du paramètre : - *obligatoire* si *Élément (widget)* n'est pas défini 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.
Afficher	0	show.0	Comportement du paramètre : - *obligatoire* si *Élément* n'est pas défini 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"
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	nom	valeur
Description	1	description	Toute valeur de chaîne, y compris les macros. Macros prises en charge : {HOST.*}, {ITEM.*}, {INVENTORY.*}, macros utilisateur. 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. Par défaut : 15.
Gras	0	desc_bold	0 - *(par défaut)* Désactivé ; 1 - Activé.
Couleur	1	desc_color	Code couleur hexadécimal (par ex. FF0000). Par défaut : "" (vide).

Valeur

Les paramètres de configuration avancés suivants sont pris en charge si *Afficher* est défini sur « Valeur ».

Paramètre	type	name	value
Décimales			
Décimales	0	decimal_places	Les valeurs possibles vont de 1 à 10. Par défaut : 2.
Taille	0	decimal_size	Les valeurs possibles vont de 1 à 100. Par défaut : 35.
Position			
Position horizontale	0	value_h_pos	0 - Gauche ; 1 - *(par défaut)* Centre ; 2 - Droite.
Position verticale	0	value_v_pos	Deux éléments ou plus (Description, Valeur, Heure) ne peuvent pas partager la même *Position horizontale* et *Position verticale*. 0 - Haut ; 1 - *(par défaut)* Milieu ; 2 - Bas.
Taille	0	value_size	Deux éléments ou plus (Description, Valeur, Heure) ne peuvent pas partager la même *Position horizontale* et *Position verticale*. Les valeurs possibles vont de 1 à 100. Par défaut : 45.
Gras	0	value_bold	0 - Désactivé ; 1 - *(par défaut)* Activé.
Couleur	1	value_color	Code couleur hexadécimal (par ex. FF0000). Par défaut : "" (vide).
Unités			

Paramètre	type	name	value
Unités (case à cocher)	0	units_show	0 - Désactivé ; 1 - *(par défaut)* Activé.
Unités (valeur)	1	units	Toute valeur de 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. Par défaut : 35.
Gras	0	units_bold	0 - Désactivé ; 1 - *(par défaut)* Activé.
Couleur	1	units_color	Code couleur hexadécimal (par ex. FF0000). Par défaut : "" (vide).

Heure

Les paramètres de configuration avancée suivants sont pris en charge si *Afficher* est défini sur « Heure ».

Paramètre	type	nom	valeur
Position horizontale	0	time_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	time_v_pos	0 - *(par défaut)* Haut; 1 - 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	time_size	Les valeurs possibles vont de 1 à 100. Par défaut : 15.
Gras	0	time_bold	0 - *(par défaut)* Désactivé; 1 - Activé.
Couleur	1	time_color	Code couleur hexadécimal (par ex. FF0000). 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. Par défaut : 1.
Remplissage	0	sparkline.fill	Les valeurs possibles vont de 0 à 10. Par défaut : 3.
Couleur	1	sparkline.color	Code couleur hexadécimal (par exemple FF0000). Par défaut : 42A5F5.
Période	1	sparkline.time_period.from	DASHBOARD._timeperiod - définit le sélecteur de période du tableau de bord comme source de données ; ABCDE._timeperiod - définit 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 *De* et *À*.
De	1	sparkline.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 : - *obligatoire* si sparkline.time_period.to est défini
À	1	sparkline.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 : - *obligatoire* si sparkline.time_period.from est défini
Données d'historique	0	sparkline.history	0 - *(par défaut)* Auto ; 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 du widget de tableau de bord correspondants 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 réussite de la création ou de la mise à jour du widget *Map*, veuillez vous référer 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

Paramètre	type	name	value
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

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ du widget de tableau de bord correspondants 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 réussite de la création ou de la mise à jour 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.

Paramètre	type	name	value
	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 du widget de tableau de bord correspondants permettent de configurer le widget *Pie chart* 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 *Pie chart*, veuillez vous référer 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.

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 concerné, tandis que le second nombre, s'il est présent, représente l'hôte ou l'élément configuré.

Parameter	type	name	value
Type de jeu 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 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 *Éléments (Widget)* n'est pas défini

Parameter	type	name	value
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 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 *Éléments* n'est pas défini
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 »
Type d'élément	0	ds.0.type.0	0 - *(par défaut)* Normal ; 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 *Type de jeu de données* est défini sur « Liste d'éléments »
Modèles d'hôte	1	ds.0.hosts.0	Nom ou modèle de l'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	1	ds.0.items.0	Nom ou modèle 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 modèles des éléments configurés sur le modèle doivent être définis. Comportement du paramètre : - *obligatoire* si *Type de jeu 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 *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. Les valeurs possibles vont de 0 à 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

Parameter	type	name	value
Fonction d'agrégation	0	ds.0.aggregate_function	1 - min ; 2 - max ; 3 - avg ; 4 - count ; 5 - sum ; 6 - first ; 7 - *(par défaut)* last.
Agrégation du jeu de données	0	ds.0.dataset_aggregation	0 - *(par défaut)* aucune ; 1 - min ; 2 - max ; 3 - avg ; 4 - count ; 5 - sum. Comportement du paramètre : - *pris en charge* si *Type d'élément* est défini sur « Total »
Libellé du jeu de données	1	ds.0.data_set_label	N'importe quelle valeur de chaîne. Par défaut : "" (vide).
Tags d'élément			
Type d'évaluation	0	ds.0.item_tags_evaltype	0 - *(par défaut)* Et/Ou ; 2 - Ou.
Nom du tag	1	ds.0.item_tags.0.tag	N'importe quelle valeur de chaîne. Remarque : Le nombre dans le nom de propriété fait référence à l'ordre du tag dans la liste d'évaluation des tags. Comportement du paramètre : - *obligatoire* si *Tags d'élément* est configuré
Opérateur	0	ds.0.item_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 du tag dans la liste d'évaluation des tags. Comportement du paramètre : - *obligatoire* si *Tags d'élément* est configuré
Valeur du tag	1	ds.0.item_tags.0.value	N'importe quelle valeur de chaîne. Remarque : Le nombre dans le nom de propriété fait référence à l'ordre du tag dans la liste d'évaluation des tags. Comportement du paramètre : - *obligatoire* si *Tags d'élément* est configuré

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.
Dessiner	0	draw_type	0 - *(par défaut)* Camembert; 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. 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"
Taille	0	value_size_type	0 - *(par défaut)* Auto; 1 - Personnalisée. Comportement du paramètre : - *pris en charge* si *Afficher la valeur totale* est défini sur "Activé"
Taille (valeur pour la taille personnalisée)	0	value_size	Les valeurs possibles vont de 1 à 100. Par défaut : 20. Comportement du paramètre : - *pris en charge* si *Afficher la valeur totale* est défini sur "Activé"
Décimales	0	decimal_places	Les valeurs possibles vont de 0 à 6. Par défaut : 2. Comportement du paramètre : - *pris en charge* si *Afficher la valeur totale* est défini sur "Activé"
Unités (case à cocher)	0	units_show	0 - *(par défaut)* Désactivé; 1 - Activé. Comportement du paramètre : - *pris en charge* si *Afficher la valeur totale* est défini sur "Activé"
Unités (valeur)	1	units	Toute valeur de chaîne. Comportement du paramètre : - *pris en charge* si *Unités* (case à cocher) est défini sur "Activé"
Gras	0	value_bold	0 - *(par défaut)* Désactivé; 1 - Activé. Comportement du paramètre : - *pris en charge* si *Afficher la valeur totale* est défini sur "Activé"
Couleur	1	value_color	Code couleur hexadécimal (par exemple FF0000). Comportement du paramètre : - *pris en charge* si *Afficher la valeur totale* est défini sur "Activé"

Paramètre	type	name	value
Espace entre les secteurs	0	space	Les valeurs possibles vont de 0 à 10. 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. Par défaut : 1. Comportement du paramètre : - *pris en charge* si *Fusionner les secteurs inférieurs à N %* (case à cocher) est défini sur "Activé"
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

Paramètre	type	name	value
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"
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. 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. Par défaut : 4. Comportement du paramètre: - *pris en charge* si *Afficher la légende* est défini sur "Activé", et *Afficher la valeur* est défini 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 du widget de tableau de bord correspondants 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 réussite de la création ou de la mise à jour du widget *Hôtes avec problèmes*, veuillez vous référer 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.
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.

Paramètre	type	name	value
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 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 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.

Paramètre	type	name	value
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 du widget de tableau de bord respectifs permettent de configurer le widget *Problems* 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 *Problems*, veuillez vous référer au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Problems*.

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 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.
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).

Paramètre	type	name	value
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.
Tags de problème			
Type d'évaluation	0	evaltype	0 - *(par défaut)* Et/Ou ; 2 - Ou.
Nom du tag	1	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 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 du tag dans la liste d'évaluation des tags. Comportement du paramètre : - *obligatoire* lors de la configuration de *Tags de problème*
Valeur du tag	1	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 de problème*
Afficher les tags	0	show_tags	0 - *(par défaut)* Aucun ; 1 - 1 ; 2 - 2 ; 3 - 3.
Nom du tag (format)	0	tag_name_format	0 - *(par défaut)* Complet ; 1 - Abrégé ; 2 - Aucun. Comportement du paramètre : - *pris en charge* si *Afficher les tags* est défini sur "1", "2" ou "3"
Priorité d'affichage des tags	1	tag_priority	Liste de tags séparés par des virgules. Comportement du paramètre : - *pris en charge* si *Afficher les tags* est défini sur "1", "2" ou "3"

Paramètre	type	name	value
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.
Statut d'accusé de réception	0	acknowledgement_status	0 - *(par défaut)* tous ; 1 - Non acquitté ; 2 - Acquitté.
Par moi	0	acknowledged_by_me	0 - *(par défaut)* Désactivé ; 1 - Activé.
Trier les entrées par	0	sort_triggers	1 - Sévérité (décroissant) ; 2 - Hôte (croissant) ; 3 - Heure (croissant) ; 4 - *(par défaut)* Heure (décroissant) ; 13 - Sévérité (croissant) ; 14 - Hôte (décroissant) ; 15 - Problème (croissant) ; 16 - Problème (décroissant). Pour toutes les valeurs, sauf "Heure (décroissant)" et "Heure (croissant)", 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 (décroissant)" ou "Heure (croissant)"
Mettre en surbrillance 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.
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*. 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`

24 Problèmes par gravité

Description

Ces paramètres et les valeurs de propriété possibles pour les objets de champ du widget de tableau de bord respectifs permettent de configurer le widget *Problems by severity* 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 *Problems by severity*, veuillez vous référer 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.

Paramètre	type	name	value
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.
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*

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	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".
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.

Paramètre	type	name	value
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 du widget de tableau de bord correspondants permettent de configurer le widget *Scatter plot* 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 *Scatter plot*, veuillez vous référer 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"
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	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

Paramètre	type	name	value
Axe Y (Widget)	1	ds.0.y_axis_itemids.0	Une 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.1	`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.
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"

Paramètre	type	name	value
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.
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	`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.

Paramètre	type	name	value
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.
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*.
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.

Paramètre	type	name	value
Unités (valeur)	1	y_axis_units x_axis_static_units y_axis_static_units	Toute valeur de chaîne. Par défaut : "" (vide).

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 »
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	Texte ou valeur numérique. Les suffixes (par exemple, "1d", "2w", "4K", "8G") sont pris en charge.
Axe Y	1	thresholds.0.y_axis_threshold	Texte ou 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 du widget de tableau de bord correspondants permettent de configurer le widget *Rapport de SLA* 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 *Rapport de SLA*, veuillez vous référer 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	nom	valeur
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 de 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. Par défaut : 20.
De	1	date_from	Chaîne de date valide au format `YYYY-MM-DD`. Les dates relatives avec les modificateurs `d`, `w`, `M`, `y` (par ex. `now`, `now/d`, `now/w-1w`, etc.) sont prises en charge.
À	1	date_to	Chaîne de date valide au format `YYYY-MM-DD`. Les dates relatives avec les modificateurs `d`, `w`, `M`, `y` (par ex. `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 du widget de tableau de bord correspondants 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 réussite de la création ou de la mise à jour du widget *Informations système*, veuillez vous référer au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Informations système*.

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.
Afficher	0	info_type	0 - *(par défaut)* Statistiques 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 que *Afficher* est défini sur "Statistiques système"

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 du widget de tableau de bord correspondants 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 réussite de la création ou de la mise à jour du widget *Top hosts*, veuillez vous référer au comportement des paramètres présenté dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Top hosts*.

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 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.

Paramètre	type	name	value
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	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	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.
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 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	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.
Afficher les hôtes en maintenance	0	maintenance	0 - *(par défaut)* Désactivé ; 1 - Activé.
Colonnes (voir ci-dessous)			

Paramètre	type	name	value
Trier par	0	column	Valeur numérique de colonne parmi les colonnes configurées.
Ordre	0	order	2 - *(par défaut)* Top N ; 3 - Bottom N.
Limite d'hôtes	0	show_lines	Les valeurs possibles vont de 1 à 1000. 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 d'élément

Les paramètres suivants sont pris en charge si *Données* est défini sur « Valeur d'élément ».

Note:
Le premier nombre dans le nom de propriété *Thresholds* (par ex. 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 ex. `"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 comme	0	columns.0.display_value	0 - *(par défaut)* Numérique; 1 - Texte; 2 - Binaire.

Paramètre	type	name	value
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 comme* 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 comme* est défini sur « Numérique » et *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 comme* est défini sur « Numérique » et *Affichage* est défini sur « Barre » ou « Indicateurs »
Nombre de déci-males	0	columns.0.decimal_places	Les valeurs possibles vont de 0 à 10. Par défaut : 2. Comportement du paramètre: - *pris en charge* si *Afficher la valeur de l'élément comme* est défini sur « Numérique »
Sparkline			
Largeur	0	columns.0.sparkline.width	Les valeurs possibles vont de 0 à 10. Par défaut : 1. Comportement du paramètre: - *pris en charge* si *Afficher la valeur de l'élément comme* est défini sur « Numérique » et *Affichage* est défini sur « Sparkline »
Remplissage	0	columns.0.sparkline.fill	Les valeurs possibles vont de 0 à 10. Par défaut : 3. Comportement du paramètre: - *pris en charge* si *Afficher la valeur de l'élément comme* est défini sur « Numérique » et *Affichage* est défini sur « Sparkline »
Couleur	1	columns.0.sparkline.color	Code couleur hexadécimal (par ex. `FF0000`). Par défaut : `42A5F5`. Comportement du paramètre: - *pris en charge* si *Afficher la valeur de l'élément comme* est défini sur « Numérique » et *Affichage* est défini sur « Sparkline »
Période	1	columns.0.sparkline.time_period	`DASHBOARD._timeperiod` - définit le sélecteur de période du tableau de bord comme source de données ; `ABCDE._timeperiod` - définit 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 *De* et *À*. Comportement du paramètre: - *pris en charge* si *Afficher la valeur de l'élément comme* est défini sur « Numérique » et *Affichage* est défini sur « Sparkline »

Paramètre	type	name	value
De	1	columns.0.sparkline.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 *Afficher la valeur de l'élément comme* est défini sur « Numérique » et *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 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 *Afficher la valeur de l'élément comme* est défini sur « Numérique » et *Affichage* est défini sur « Sparkline » - *requis* si `columns.0.sparkline.time_period.from` est défini
Données d'historique	0	columns.0.sparkline.history	0 - *(par défaut)* Auto; 1 - Historique; 2 - Tendances. Comportement du paramètre: - *pris en charge* si *Afficher la valeur de l'élément comme* est défini sur « Numérique » et *Affichage* est défini sur « Sparkline »
Seuils			
Couleur	1	columnsthresholds.0.color	Code couleur hexadécimal (par ex. `FF0000`). Comportement du paramètre: - *pris en charge* si *Afficher la valeur de l'élément comme* est défini sur « Numérique »
Seuil	1	columnsthresholds.0.threshold	Toute valeur de chaîne. Comportement du paramètre: - *pris en charge* si *Afficher la valeur de l'élément comme* est défini sur « Numérique »
Mises en évidence			
Couleur	1	columns.0.highlights.0.color	Code couleur hexadécimal (par ex. `FF0000`). Comportement du paramètre: - *pris en charge* si *Afficher la valeur de l'élément comme* 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 comme* 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 comme* est défini sur « Binaire »
Fonction d'agrégation	0	columns.0.aggregate_function	0 - *(par défaut)* non utilisé ; 1 - min ; 2 - max ; 3 - avg ; 4 - count ; 5 - sum ; 6 - first ; 7 - last.

Paramètre	type	name	value
Période	1	columns.0.time_period.from	DASHBOARD._timeperiod - définit le sélecteur de période du tableau de bord comme source de données ; ABCDE._timeperiod - définit 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 *De* et *À*. Comportement du paramètre: - *pris en charge* si *Fonction d'agrégation* est définie sur « min », « max », « avg », « count », « sum », « first », « last »
De	1	columns.0.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 *Fonction d'agrégation* est définie sur « min », « max », « avg », « count », « sum », « first », « last » - *requis* si columns.0.time_period.to est défini
À	1	columns.0.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 *Fonction d'agrégation* est définie sur « min », « max », « avg », « count », « sum », « first », « last » - *requis* si columns.0.time_period.from est défini
Données d'historique	0	columns.0.history	0 - *(par défaut)* Auto; 1 - Historique; 2 - Tendances. Comportement du paramètre: - *pris en charge* si *Afficher la valeur de l'élément comme* est défini sur « Numérique »
Référence	1	reference	Toute valeur de chaîne composée de 5 caractères (par ex. ABCDE ou JBPNL). Cette valeur doit être unique dans le 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 ».

Paramètre	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 : - *obligatoire* 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 du widget de tableau de bord correspondants permettent de configurer le widget *Top items* 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 *Top items*, veuillez vous référer 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.

Paramètre	type	name	value
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.
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 *Top items*.

Paramètre	type	name	value
Ordre des hôtes			
Trier par	0	host_ordering_order_by	0 - *(par défaut)* Nom de l'hôte ; 3 - Valeur de l'élément.

Paramètre	type	name	value
Modèles d'élément	1	host_ordering_item.0	Nom ou modèle de l'élément (par ex. "*: Number of processed *values per second"). 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. 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* si *Trier par* est défini sur « Valeur de l'élément »
Ordre	0	host_ordering_order	2 - *(par défaut)* Top N ; 3 - Bottom N.
Limite	0	host_ordering_limit	Les valeurs possibles vont de 1 à 100. Par défaut : 10.
Ordre des éléments			
Trier par	0	item_ordering_order_by	1 - Hôte ; 2 - Nom de l'élément ; 3 - *(par défaut)* Valeur de l'élément.
Modèles d'hôte	1	item_ordering_host.0	Nom ou modèle de l'hôte. Remarque : pour configurer plusieurs modèles d'hôte, 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 *Trier par* est défini sur « Hôte »
Ordre	0	item_ordering_order	2 - *(par défaut)* Top N ; 3 - Bottom N.
Limite	0	item_ordering_limit	Les valeurs possibles vont de 1 à 100. Par défaut : 10.

Colonnes

Note:
Le premier nombre dans le nom de propriété (par ex. columns.0.items.0, columns.0.item_tags_evaltype) représente la colonne particulière, tandis que le second nombre, s'il est présent, représente l'entité configurée (par ex. modèle d'élément, tag).

Paramètre	type	name	value
Modèles d'élément	1	columns.0.items.0	Nom ou modèle de Item (par ex. "*: Number of processed *values per second"). 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 second nombre incrémenté dans le nom de propriété. 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*
Tags d'élément			
Type d'évaluation	0	columns.0.item_tags_evaltype	0 - *(par défaut)* And/Or; 2 - Or.
Nom du tag	1	columns.0.item_tags.0.tag	Toute valeur de chaîne. Remarque : le second nombre 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'élément*
Opérateur	0	columns.0.item_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 second nombre 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'élément*
Valeur du tag	1	columns.0.item_tags.0.value	Toute valeur de chaîne. Remarque : le second nombre 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'élément*
Couleur de base	1	columns.0.base_color	Code couleur hexadécimal (par ex. 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 ex. "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* est défini sur "Barre" ou "Indicateurs"

Paramètre	type	name	value
Max	1	columns.0.max	Toute valeur numérique. Les suffixes (par ex. ”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* est défini sur ”Barre” ou ”Indicateurs”
Sparkline			
Largeur	0	columns.0.sparkline.width	Les valeurs possibles vont de 0 à 10. Par défaut : 1. Comportement du paramètre : - *pris en charge* si *Affichage* est défini sur ”Sparkline”
Remplissage	0	columns.0.sparkline.fill	Les valeurs possibles vont de 0 à 10. 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 ex. `FF0000`). 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. 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 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 *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 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 *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)* Auto ; 1 - Historique ; 2 - Tendances. Comportement du paramètre : - *pris en charge* si *Affichage* est défini sur ”Sparkline”
Seuils			

Paramètre	type	name	value
Couleur	1	columns.0.thresholds.color	Code couleur hexadécimal (par ex. FF0000). Comportement du paramètre : - *pris en charge* si *Afficher la valeur comme* est défini sur ”Numérique”
Seuil	1	columns.0.thresholds.threshold	Threshold numérique. Les suffixes (par ex. ”1d”, ”2w”, ”4K”, ”8G”) sont pris en charge. Comportement du paramètre : - *pris en charge* si *Afficher la valeur comme* est défini sur ”Numérique”
Mises en évidence			
Mise en évidence	1	columns.0.highlights.color	Code couleur hexadécimal (par ex. FF0000). Comportement du paramètre : - *pris en charge* si *Afficher la valeur comme* est défini sur ”Texte”
Seuil	1	columns.0.highlights.pattern	Une 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. 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 utilisé ; 1 - min ; 2 - max ; 3 - avg ; 4 - count ; 5 - sum ; 6 - first ; 7 - last. Si *Afficher la valeur comme* est défini sur ”Texte” : 0 - *(par défaut)* non utilisé ; 4 - count ; 6 - first ; 7 - last.
Période	1	columns.0.time_period.reference	DASHBOARD - 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 *De* et *À*. Comportement du paramètre : - *pris en charge* si *Fonction d’agrégation* est définie sur ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first”, ”last”

Paramètre	type	name	value
De	1	columns.0.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 *Fonction d'agrégation* est définie sur "min", "max", "avg", "count", "sum", "first", "last" - *obligatoire* si `columns.0.time_period.to` est défini
À	1	columns.0.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 *Fonction d'agrégation* est définie sur "min", "max", "avg", "count", "sum", "first", "last" - *obligatoire* si `columns.0.time_period.from` est défini
Données d'historique	0	columns.0.history	0 - *(par défaut)* Auto ; 1 - Historique ; 2 - Tendances. Comportement du paramètre : - *pris en charge* si *Afficher la valeur comme* est défini sur "Numérique"
Agréger	0	columns.0.aggregate_column	0 - *(par défaut)* Chaque élément 1 - Combiné. Comportement du paramètre : - *pris en charge* si *Afficher la valeur comme* est défini sur "Numérique" et *Affichage* est défini sur "Tel quel", "Barre" ou "Indicateurs".
Fonction d'agrégation combinée	0	columns.0.column_aggregate_function	1 - min ; 2 - max ; 3 - avg ; 4 - count ; 5 - sum *(par défaut)*. Comportement du paramètre : - *obligatoire* lors de la configuration de *Agréger*.
Nom de colonne combinée	1	columns.0.combined_column_name	Toute valeur de chaîne. Comportement du paramètre : - *obligatoire* lors de la configuration de *Agréger*.

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 du widget de tableau de bord correspondants permettent de configurer le widget *Top triggers* 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 *Top triggers*, veuillez vous référer au comportement des paramètres décrit dans les tableaux ci-dessous.

Paramètres

Les paramètres suivants sont pris en charge pour le widget *Top triggers*.

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.

Paramètre	type	name	value
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.
Problème	1	problem	Nom d'événement du problème (insensible à la casse, nom complet ou partie de celui-ci).
Sévérité	0	severities.0	Sévérités du déclencheur. 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.
Tags de problème			
Type d'évaluation	0	evaltype	0 - *(par défaut)* Et/Ou ; 2 - Ou.
Nom du tag	1	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 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 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 de problème*
Valeur du tag	1	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 de problème*

Paramètre	type	name	value
Période	1	time_period._referenceDASHBOARD._timeperiod - définit le sélecteur de période 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 uniquement dans les paramètres *From* et *To*.
From	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
To	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
Limite de déclencheurs	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 du widget de tableau de bord respectifs permettent de configurer le widget *Aperçu des déclencheurs* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés des « champs » 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 inclus et de créer des widgets customisés, mais introduit également le risque de création ou de mise à jour de widgets. incorrectement. Pour garantir la création ou la mise à jour réussie du widget *Aperçu du déclencheur*, veuillez vous référer 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.

Paramètre	type	name	value
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.
Opérateur	0	tags.0.operator	Comportement du paramètre : - *obligatoire* lors de la configuration de *Balises de problème* 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 du widget de tableau de bord respectifs permettent de configurer le widget *URL* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés des 'champs' 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 built-in widgets et de créer des custom widgets, mais introduit également le risque de création ou de mise à jour de widgets. incorrectement. Pour garantir la création ou la mise à jour réussie du widget *URL*, veuillez vous référer 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 du widget de tableau de bord respectifs permettent de configurer le widget *Surveillance Web* dans les méthodes `dashboard.create` et `dashboard.update`.

Attention:
Les propriétés des « champs » 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 création ou de mise à jour de widgets. incorrectement. Pour garantir la création ou la mise à jour réussie du widget *Surveillance Web*, veuillez vous référer 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.
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 scé-nario			
Type d'évaluation	0	evaltype	0 - *(par défaut)* Et/Ou; 2 - Ou.

Paramètre	type	name	value
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*
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.

Propriété	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 : - *constant* - *obligatoire* pour les opérations de création
display_period	integer	Période d'affichage par défaut de la page (en secondes). Valeurs possibles : 10, 30, 60, 120, 600, 1800, 3600. Par défaut : 30.
auto_start	integer	Démarrer automatiquement le diaporama. Valeurs possibles : 0 - ne pas démarrer automatiquement le diaporama ; 1 - *(par défaut)* démarrer automatiquement le diaporama.
uuid	string	Identifiant universel unique, utilisé pour lier les tableaux de bord de modèle importés à ceux qui existent déjà. Généré automatiquement s'il n'est pas fourni.

Page de tableau de bord de modèle

L'objet page de tableau de bord de modèle possède les propriétés suivantes.

Propriété	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	Période d'affichage de la page de tableau de bord (en secondes). Valeurs possibles : 0, 10, 30, 60, 120, 600, 1800, 3600. Par défaut : 0 (utilisera la période d'affichage par défaut de la page).
widgets	array	Tableau des objets widget de tableau de bord de 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 d’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 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. 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 ;

Propriété	Type	Description
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'utilisateurs *Admin* et *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)` 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 également

- Modèle de page de tableau de bord
- Modèle de widget de tableau de bord
- Champs de modèle de widget de tableau de bord

Source

CTemplateDashboard::create() dans *ui/include/classes/api/services/CTemplateDashboard.php*.

templatedashboard.delete

Description

`object templatedashboard.delete(array templateDashboardIds)`

Cette méthode permet de supprimer des tableaux de bord de modèle.

> **Note:**
> Cette méthode est disponible uniquement pour les types d'utilisateurs *Admin* et *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

`(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)`

Cette méthode permet de récupérer les tableaux de bord de modèles selon les paramètres donnés.

> **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 User roles 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 de modèle avec les ID indiqués.
templateids	ID/array	Retourner uniquement les tableaux de bord de modèle appartenant aux modèles indiqués.
selectPages	query	Retourner une propriété `pages` avec les pages du tableau de bord de modèle, correctement ordonnées.
sortfield	string/array	Trier le résultat selon les propriétés indiquées. Valeurs possibles : `dashboardid`, `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 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 également

- Modèle de page de tableau de bord
- Modèle de widget de tableau de bord
- Champs de modèle de widget de tableau de bord

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 la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir Rôles utilisateur pour plus d'informations.

Paramètres

`(object/array)` Propriétés des tableaux de bord de 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 des tableaux de bord de modèle, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
pages	array	Pages de tableau de bord de modèle pour remplacer les pages de tableau de bord existantes. Les pages du tableau de bord sont mises à jour à l'aide de la propriété `dashboard_pageid`. De nouvelles pages de tableau de bord seront créées pour les objets sans propriété `dashboard_pageid` et les pages de tableau de bord existantes 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 également

- Modèle de widget de tableau de bord
- Champs de modèle de widget de tableau de bord

Source

CTemplateDashboard::update() dans *ui/include/classes/api/services/CTemplateDashboard.php*.

Table de correspondance des valeurs

Cette classe est conçue pour fonctionner avec les tables de correspondance des valeurs.

Références d'objet :

- Table de correspondance des valeurs
  - Correspondances de valeurs

Méthodes disponibles :

- valuemap.create - créer de nouvelles tables de correspondance des valeurs
- valuemap.delete - supprimer des tables de correspondance des valeurs
- valuemap.get - récupérer des tables de correspondance des valeurs
- valuemap.update - mettre à jour des tables de correspondance des 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* - *obligatoire* 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* - *obligatoire* pour les opérations de création
name	string	Nom de la correspondance de valeurs. Comportement de la propriété : - *obligatoire* 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 : - *obligatoire* pour les opérations de création
uuid	string	Identifiant universel unique, 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 - 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 l'intervalle (les intervalles sont inclusifs ; plusieurs intervalles, séparés par une virgule, peuvent être définis)[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 permet de créer de nouvelles correspondances de valeurs.

> **Note:**
> Cette méthode est disponible uniquement pour le type d'utilisateur *Super admin*. Les autorisations pour appeler la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir User roles pour plus d'informations.

Paramètres

`(object/array)` Tables de correspondance à créer.

La méthode accepte les tables de correspondance avec les propriétés de table de correspondance standard.

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 permet de supprimer des 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 du rôle utilisateur. Voir 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 les correspondances de valeurs selon les paramètres donnés.

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 User roles 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 permet de mettre à jour des correspondances de valeurs 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 la table de correspondance à mettre à jour.

La propriété `valuemapid` doit être définie pour chaque table de correspondance, 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*.

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 de l'heure pour laquelle la valeur a été calculée. Par exemple, l'horodatage de « 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	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)`

Cette méthode permet de récupérer les données de tendances 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. 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
itemids	ID/array	Retourner uniquement les tendances avec les ID d'élément donnés.
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.

Propriété	Type	Description
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 permet de créer de nouveaux jetons.

Note:
La permission *Gérer les jetons d'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 *Manage API tokens* 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 permet de générer des jetons.

Note:
L'autorisation *Manage API tokens* permission 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 en fonction des paramètres donnés.

> **Note:**
> Seul le type d'utilisateur *Super admin* est autorisé à afficher les jetons des 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 jetons avec les ID donnés.
userids	ID/array	Retourner uniquement les jetons créés pour les utilisateurs donnés.
token	string	Retourner uniquement les jetons créés pour le *jeton d'authentification* donné.
valid_at	timestamp	Retourner uniquement les jetons qui sont valides (non expirés) à la date et à l'heure données.
expired_at	timestamp	Retourner uniquement les jetons 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 permet de mettre à jour des jetons existants.

> **Note:**
> L'autorisation *Manage API tokens* 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*.

Type de média

Cette classe est conçue pour fonctionner avec les types de média.

Références d'objet :

- Type de média
  - Paramètres du webhook
  - Paramètres du 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 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

Propriété	Type	Description
type	integer	Transport utilisé par le type de média. Valeurs possibles : 0 - Email ; 1 - Script ; 2 - SMS ; 4 - Webhook. 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 email à partir de 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”
smtp_port	integer	Port du serveur SMTP auquel se connecter. Par défaut : 25. Comportement de la propriété : - *pris en charge* si `type` est défini sur ”Email”
smtp_security	integer	Niveau de sécurité de 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”

Propriété	Type	Description
smtp_verify_host	integer	Vérification de l'hôte SSL 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 du pair SSL 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 *Office365 relay*. Comportement de la propriété : - *pris en charge* si `type` est défini sur "Email"
redirection_url	string	URL du frontend Zabbix vers laquelle rediriger après l'autorisation OAuth. Par défaut : Valeur de la propriété des paramètres de l'API `url` 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 dans le serveur d'autorisation OAuth. Comportement de la propriété : - *obligatoire* si `smtp_authentication` est défini sur "OAuth token"
client_secret	string	Secret client enregistré dans le serveur d'autorisation OAuth. Accessible uniquement pour un utilisateur de type Super Admin. Comportement de la propriété : - *obligatoire* si `smtp_authentication` est défini sur "OAuth token"
authorization_url	string	URL OAuth, avec paramètres, pour 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 pour 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"

Propriété	Type	Description
tokens_status	integer	Masque de bits de l'état des jetons. Valeurs possibles : 0 - (par défaut) Les deux jetons contiennent une valeur invalide 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 des modifications de 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'il soit nécessaire d'effectuer une requête vers `refresh_url`. Elle est définie par le serveur Zabbix lors de l'actualisation de `access_token` ou par l'API lors des modifications de 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"
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. Par défaut : 1.
maxattempts	integer	Nombre maximal de tentatives d'envoi d'une alerte. Valeurs possibles : 1-100. Par défaut : 3.

Propriété	Type	Description
attempt_interval	string	Intervalle entre les tentatives de nouvelle tentative. Accepte les secondes et une unité de temps avec suffixe. Valeurs possibles : 0-1h. Par défaut : 10s.
message_format	integer	Format du message. Valeurs possibles : 0 - Texte brut ; 1 - *(par défaut)* HTML.
script	text	Comportement de la propriété : - *pris en charge* si `type` est défini sur ”Email” Corps du script Webhook (JavaScript).
timeout	string	Comportement de la propriété : - *obligatoire* si `type` est défini sur ”Webhook” Délai d’expiration du script Webhook. Accepte les secondes et une unité de temps avec suffixe. Valeurs possibles : 1-60s. Par défaut : 30s.
process_tags	integer	Comportement de la propriété : - *pris en charge* si `type` est défini sur ”Webhook” Traiter les valeurs des propriétés JSON dans la réponse du script Webhook comme des tags. Ces tags sont ajoutés à tous les 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.
show_event_menu	integer	Comportement de la propriété : - *pris en charge* si `type` est défini sur ”Webhook” Inclure une entrée dans le menu d’événement qui pointe 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 d’événement ni la propriété `urls` ; 1 - Inclure l’entrée du menu d’événement 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 d’événement 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 d’événement et dans la propriété `urls` renvoyée par problem.get et event.get. Comportement de la propriété : - *pris en charge* si `show_event_menu` est défini sur ”Include event menu entry and `urls` property”

Propriété	Type	Description
parameters	array	Paramètres du Webhook ou du script. Comportement de la propriété : - *pris en charge* si `type` est défini sur ”Webhook” ou ”Script”
description	text	Description du type de média.

Paramètres du webhook

Les paramètres du webhook 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. 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*

Propriété	Type	Description
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 permet de créer de nouveaux types de média.

> **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)` 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.TI
```

```
            },
            {
                "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.
            },
            {
                "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 permet de supprimer des types de média.

> **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 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)`

La méthode permet de récupérer les types de média selon les paramètres donnés.

> **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 User roles 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 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 médias 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 du paramètre : - *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`.
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 du type 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 le commentaire de référence.
editable	boolean	
excludeSearch	boolean	
limit	integer	
preservekeys	boolean	

Paramètre	Type	Description
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 permet de mettre à jour des types de média 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 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 les 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 « Exécuter maintenant »
  - Objet de requête « Actualiser la configuration du proxy »
  - Objet de requête « Informations de diagnostic »
    - Objet de requête de statistiques
    - Objet de résultat de statistiques

Méthodes disponibles :

- task.create - créer de nouvelles tâches
- task.get - récupérer des tâches

Objet task

Les objets suivants sont directement liés à l'API `task`.

Task

L'objet task 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* ne peuvent créer que des tâches « Exécuter maintenant ». Notez que `task.get` renvoie toujours « 7 » (Résumé de l'exécution de la 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*

Propriété	Type	Description
clock	timestamp	Heure de création de la tâche. Comportement de la propriété : - *lecture seule*
ttl	integer	Délai 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 la 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 de 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 « Execute now » des éléments sur les hôtes pour lesquels ils disposent d'une permission en *lecture-écriture*, ou d'une permission en *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 des proxys.

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 concernant 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 requête 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

Type de diagnostic	Champs disponibles	Type
	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 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 du rôle utilisateur. Voir User roles 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 : - *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 - *lecture seule* pour les utilisateurs provisionnés si l'utilisateur est lié à un répertoire d'utilisateurs (`userdirectoryid` est défini sur une valeur valide différente de "0", que le statut de provisionnement du répertoire d'utilisateurs est activé (`provision_status` de l'objet Répertoire d'utilisateurs est défini sur "1"), et que le statut d'authentification de tous les provisionnements LDAP ou SAML est activé (`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 répertoire d'utilisateurs. Comportement de la propriété : - *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.
attempt_clock	timestamp	Heure de la dernière tentative de connexion infructueuse. Comportement de la propriété : - *lecture seule*
attempt_failed	integer	Nombre récent de tentatives de connexion échouées. Comportement de la propriété : - *lecture seule*
attempt_ip	string	Adresse IP d'où provenait la dernière tentative de connexion infructueuse. Comportement de la propriété : - *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 des secondes et une unité de temps avec suffixe. Si défini sur 0s, la session n'expirera jamais. Par défaut : 15m.
lang	string	Code de langue de l'utilisateur, par exemple `en_US`. Par défaut : `default` - valeur par défaut du système.

Propriété	Type	Description
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 : - *lecture seule*
refresh	string	Période de rafraîchissement automatique. Accepte des secondes ou une unité de temps avec suffixe (par ex., 30s, 90s, 1m, 1h). Par défaut : 30s.
rows_per_page	integer	Nombre de lignes d'objet à afficher par page. 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 ; `dark-theme` - Sombre.
ts_provisioned	timestamp	Heure à laquelle la dernière opération de provisionnement a été effectuée. Comportement de la propriété : - *lecture seule* - *pris en charge* pour les utilisateurs de type *Super admin*
url	string	URL de la page vers laquelle rediriger l'utilisateur après la connexion.
userdirectoryid	ID	ID du répertoire 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 répertoire d'utilisateurs. Pour les opérations de connexion, la valeur de cette propriété aura priorité sur la propriété `userdirectoryid` des groupes d'utilisateurs auxquels l'utilisateur appartient. Par défaut : 0. Comportement de la propriété : - *lecture seule* - *pris en charge* pour les utilisateurs de type *Super admin*
timezone	string	Fuseau horaire de l'utilisateur, par exemple `Europe/London`, `UTC`. Par défaut : `default` - valeur par défaut du système. Pour la liste complète des fuseaux horaires pris en charge, veuillez consulter la documentation PHP.

Média

L'objet média possède les propriétés suivantes.

Propriété	Type	Description
mediaid	ID	ID du média de l'utilisateur. Comportement de la propriété : - *lecture seule*

Propriété	Type	Description
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 le `type` du type de média est défini sur « Email », les valeurs sont représentées sous forme de tableau. Pour les autres types de types de média, 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.
severity	integer	Sévérités de déclencheur pour lesquelles envoyer des notifications. Valeurs bitmap possibles : 1 - Non classé ; 2 - Information ; 4 - Avertissement ; 8 - Moyenne ; 16 - Haute ; 32 - Désastre. Il s'agit d'un champ de masque de bits ; toute somme des valeurs bitmap possibles est acceptée (par exemple, 48 pour Moyenne, Haute et Désastre). Par défaut : 63.
period	string	Période pendant laquelle les notifications peuvent être envoyées, sous la forme d'une période de temps ou de macros utilisateur séparées par un point-virgule. 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 du média de l'annuaire utilisateur pour les médias provisionnés. Comportement de la propriété : - *lecture seule* - *pris en charge* pour les utilisateurs de type *Super admin*

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.

Paramètre	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 : - *pris en charge* si `sessionid` est défini
sessionid	string	Jeton d'authentification de l'utilisateur. Comportement du paramètre : - *obligatoire* si `token` n'est pas défini
token	string	Jeton d'API de l'utilisateur. Comportement du paramètre : - *obligatoire* 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 web. 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 Rôle 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 permet de créer de nouveaux 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. Voir User roles pour plus d'informations.

Note:
La robustesse du mot de passe utilisateur est validée conformément aux règles de politique de mot de passe définies par l'Authentication API. Voir Authentication API 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.

Paramètre	Type	Description
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": "Doe123",
        "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 également

- Authentification
- Media
- 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 permet de supprimer des utilisateurs.

> **Note:**
> Cette méthode est uniquement disponible 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

`(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'appeler 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.

> **Note:**
> Lors de la demande d'informations sur les médias utilisateur, les autorisations 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	Retourner uniquement les utilisateurs qui utilisent le média donné.
mediatypeids	ID/array	Retourner uniquement les utilisateurs qui utilisent les types de média donnés.
userids	ID/array	Retourner uniquement les utilisateurs ayant les ID donnés.
usrgrpids	ID/array	Retourner uniquement les utilisateurs appartenant aux groupes d'utilisateurs donnés.
getAccess	flag	Ajoute des informations supplémentaires sur les autorisations de l'utilisateur. Ajoute les propriétés suivantes pour chaque utilisateur : `gui_access` - `(integer)` méthode d'authentification frontend de l'utilisateur. Reportez-vous à la propriété `gui_access` de l'objet groupe d'utilisateurs pour 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	Retourner les médias utilisés par l'utilisateur dans la propriété `medias`.
selectMediatypes	query	Retourner 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	Retourner 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	Retourner le rôle utilisateur dans la propriété `role`.
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`.
output	query	Propriétés possibles de l'objet User 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 User à retourner. 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`.

Paramètre	Type	Description
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 User 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`.
sortfield	string/array	Trier le résultat selon les propriétés données. 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

Récupérez 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",
            "refresh": "0s",
```

```
            "theme": "default",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "50",
            "timezone": "default",
            "roleid": "3",
            "userdirectoryid": "0",
            "ts_provisioned": "0"
        },
        {
            "userid": "2",
            "username": "guest",
            "name": "",
            "surname": "",
            "url": "",
            "autologin": "0",
            "autologout": "15m",
            "lang": "default",
            "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"
        },
        {
            "userid": "3",
            "username": "user",
            "name": "Zabbix",
            "surname": "User",
            "url": "",
            "autologin": "0",
            "autologout": "0",
            "lang": "ru_RU",
            "refresh": "15s",
            "theme": "dark-theme",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "100",
            "timezone": "default",
            "roleid": "1",
            "userdirectoryid": "0",
            "ts_provisioned": "0"
        }
    ],
    "id": 1
}
```


Récupération des utilisateurs en tant qu'*Admin*

En tant qu'utilisateur de type *Admin*, récupérez des données détaillées sur votre propre utilisateur ainsi que des données limitées pour les 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",
            "usrgrps": [
                {
                    "usrgrpid": "7",
                    "name": "Zabbix administrators",
                    "gui_access": "0",
                    "users_status": "0",
                    "debug_mode": "0",
                    "mfa_status": "0"
                }
            ]
        },
        {
            "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",
            "gui_access": "0",
            "debug_mode": "0",
            "users_status": "0",
            "usrgrps": [
                {
                    "usrgrpid": "7",
                    "name": "Zabbix administrators",
                    "gui_access": "0",
                    "users_status": "0",
                    "debug_mode": "0",
                    "mfa_status": "0"
                }
            ],
```

```
            "medias": [
                {
                    "mediaid": "2",
                    "mediatypeid": "1",
                    "sendto": [
                        "john.doe@example.com"
                    ],
                    "active": "0",
                    "severity": "63",
                    "period": "1-7,00:00-24:00",
                    "provisioned": 0
                }
            ],
            "mediatypes": [
                {
                    "mediatypeid": "1",
                    "type": "0",
                    "name": "Email",
                    "status": "0",
                    "description": "",
                    "maxattempts": "3"
                }
            ],
            "role": {
                "roleid": "2",
                "name": "Admin role",
                "type": "2",
                "readonly": "0"
            }
        }
    ],
    "id": 1
}
```

Récupération des données utilisateur

Récupérez les données d'un utilisateur avec l'ID « 12 ».

Requête :

```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": ["userid", "username"],
        "selectRole": "extend",
        "userids": "12"
    },
    "id": 1
}
```

Réponse :

```
{
  "jsonrpc": "2.0",
  "result": [
      {
          "userid": "12",
          "username": "John",
          "role": {
              "roleid": "5",
              "name": "Operator",
              "type": "1",
              "readonly": "0"
          }
```

```
        }
    ],
    "id": 1
}
```

Voir également

- Media
- Type de média
- Groupe utilisateur
- Rôle

Source

CUser::get() dans *ui/include/classes/api/services/CUser.php*.

user.login

Description

`string/object user.login(object parameters)`

Cette méthode permet de se connecter à l'API et de générer un jeton d'authentification.

Warning:
Lorsque vous utilisez cette méthode, vous devez également effectuer user.logout afin d'éviter la génération d'un grand nombre d'enregistrements de session ouverts.

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 les 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. Consultez la propriété `authentication_type` de l'objet Authentication pour obtenir la liste des valeurs possibles.

Propriété	Type	Description
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 web. Consultez la propriété `gui_access` de l'objet Groupe d'utilisateurs pour obtenir 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. Consultez la propriété `type` de l'objet Rôle pour obtenir 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 permet de se déconnecter de l'API et invalide le jeton d'authentification actuel.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations permettant d'appeler la méthode peuvent être révoquées dans les paramètres du rôle utilisateur. Voir User roles 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 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 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 du rôle utilisateur. Voir User roles 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 la 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 permet de mettre à jour des utilisateurs existants.

> **Note:**
> Cette méthode est disponible pour les utilisateurs de tout type. Les autorisations d'appeler la méthode peuvent être révoquées dans les paramètres des rôles utilisateur. Voir User roles pour plus d'informations.

> **Note:**
> La robustesse du mot de passe utilisateur est validée conformément aux règles de politique de mot de passe définies par l'Authentication API. Voir Authentication API 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. Seule la propriété `usrgrpid` doit être définie pour les groupes d'utilisateurs.
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 appartient la vérification.
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"
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"

Propriété	Type	Description
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*
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.

Propriété	Type	Description
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 donnés.

> **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 User roles 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 vérifications de découverte avec les ID donnés.
druleids	ID/array	Retourner uniquement les vérifications de découverte qui appartiennent aux règles de découverte données.
dserviceids	ID/array	Retourner uniquement les vérifications de découverte qui ont détecté les services découverts donnés.
selectDRules	query	Retourner les règles de découverte liées aux vérifications 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.

Propriété	Type	Description
itemid	ID	ID de l'élément. Comportement de la propriété : - *lecture seule* - *obligatoire* pour les opérations de mise à jour
delay	string	Intervalle de mise à jour de l'élément. Accepte des secondes ou une unité de temps avec suffixe (par exemple, 30s, 1m, 2h, 1d) et, éventuellement, un ou plusieurs intervalles personnalisés, tous séparés par des points-virgules. Les intervalles personnalisés peuvent être un mélange d'intervalles flexibles et d'intervalles de planification. Accepte les macros utilisateur. Si elles sont utilisées, la valeur doit être une seule macro. Plusieurs macros ou des macros mélangées avec du texte ne sont pas prises en charge. Les intervalles flexibles peuvent être écrits sous 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), 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 l'élément appartient. Comportement de la propriété : - *constant* - *obligatoire* pour les opérations de création
interfaceid	ID	ID de l'interface 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" - *lecture seule* pour les objets découverts
key_	string	Clé de l'élément. Comportement de la propriété : - *obligatoire* pour les opérations de création - *lecture seule* pour les objets hérités ou les objets découverts

Propriété	Type	Description
name	string	Nom de l'élément. Prend en charge les macros utilisateur. Comportement de la propriété : - *obligatoire* pour les opérations de création - *lecture seule* pour les objets hérités ou les objets découverts
name_resolved	string	Nom de l'élément avec les macros utilisateur résolues. Comportement de la propriété : - *lecture seule*
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 - Élément Web ; 10 - External check ; 11 - Database monitor ; 12 - IPMI agent ; 13 - SSH agent ; 14 - TELNET agent ; 15 - Calculated ; 16 - JMX agent ; 17 - SNMP trap ; 18 - Élément dépendant ; 19 - HTTP agent ; 20 - SNMP agent ; 21 - Script ; 22 - Browser. Comportement de la propriété : - *obligatoire* pour les opérations de création - *lecture seule* pour les objets hérités ou les objets 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" - *lecture seule* pour les objets hérités ou les objets 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 - *lecture seule* pour les objets hérités ou les objets découverts

Propriété	Type	Description
allow_traps	integer	Permet de renseigner la valeur de manière similaire à un élément trapper. 0 - *(par défaut)* Ne pas autoriser l'acceptation de données entrantes ; 1 - Autoriser l'acceptation de données entrantes. Comportement de la propriété : - *pris en charge* si `type` est défini sur "HTTP agent" - *lecture seule* pour les objets découverts
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" - *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 : - *lecture seule* pour les objets découverts
error	string	Texte d'erreur s'il y a des problèmes lors de la mise à jour de la valeur de l'élément. Comportement de la propriété : - *lecture seule*
flags	integer	Origine de l'élément. Valeurs possibles : 0 - un élément simple ; 4 - un élément converti à partir d'un prototype. Comportement de la propriété : - *lecture seule*
follow_redirects	integer	Suivre les redirections de 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" - *lecture seule* pour les objets hérités ou les objets découverts
headers	array	Tableau des 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" - *lecture seule* pour les objets hérités ou les objets découverts

Propriété	Type	Description
history	string	Unité de temps indiquant combien de temps les données d'historique doivent être conservées. Accepte également une macro utilisateur. Par défaut : 31d. Comportement de la propriété : - *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" - *lecture seule* pour les objets hérités ou les objets découverts
inventory_link	integer	ID du champ d'inventaire de l'hôte qui est renseigné par l'élément. Consultez la page d'inventaire de l'hôte pour obtenir la liste des champs d'inventaire d'hôte pris en charge et leurs ID. Par défaut : 0. Comportement de la propriété : - *pris en charge* si `value_type` est défini sur "numeric float", "character", "numeric unsigned" ou "text" - *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" - *lecture seule* pour les objets hérités ou les objets 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" - *lecture seule* pour les objets découverts
lastclock	timestamp	Heure à laquelle la valeur de l'élément a été mise à jour pour la dernière fois. Par défaut, seules les valeurs comprises dans les dernières 24 heures sont affichées. Vous pouvez étendre cette période en modifiant la valeur du paramètre *Max history display period* dans la section de menu *Administration → General*. Comportement de la propriété : - *lecture seule*
lastns	integer	Nanosecondes au moment où la valeur de l'élément a été mise à jour pour la dernière fois. Par défaut, seules les valeurs comprises dans les dernières 24 heures sont affichées. Vous pouvez étendre cette période en modifiant la valeur du paramètre *Max history display period* dans la section de menu *Administration → General*. Comportement de la propriété : - *lecture seule*

Propriété	Type	Description
lastvalue	string	Dernière valeur de l'élément. Par défaut, seules les valeurs comprises dans les dernières 24 heures sont affichées. Vous pouvez étendre cette période en modifiant la valeur du paramètre *Max history display period* dans la section de menu *Administration → General*. Comportement de la propriété : - *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" - *lecture seule* pour les objets hérités ou les objets découverts
master_itemid	ID	ID de l'élément maître. Comportement de la propriété : - *obligatoire* si `type` est défini sur "Dependent item" - *lecture seule* pour les objets hérités ou les objets découverts
output_format	integer	Indique si la réponse doit être convertie en JSON. 0 - *(par défaut)* Stocker brut ; 1 - Convertir en JSON. Comportement de la propriété : - *pris en charge* si `type` est défini sur "HTTP agent" - *lecture seule* pour les objets hérités ou les objets découverts
params	string	Paramètres supplémentaires selon le type de l'élément : - script exécuté pour les éléments SSH agent et TELNET agent ; - requête SQL pour les éléments Database monitor ; - formule pour les éléments Calculated ; - le 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" - *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 avec 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" - *lecture seule* pour les objets hérités ou les objets découverts
password	string	Mot de passe pour l'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" - *lecture seule* pour les objets hérités (si `type` est défini sur "HTTP agent") ou les objets découverts

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” - *lecture seule* pour les objets hérités ou les objets 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” - *lecture seule* pour les objets hérités ou les objets découverts
prevvalue	string	Valeur précédente de l’élément. Par défaut, seules les valeurs comprises dans les dernières 24 heures sont affichées. Vous pouvez étendre cette période en modifiant la valeur du paramètre *Max history display period* dans la section de menu *Administration → General*. Comportement de la propriété : - *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” - *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” - *lecture seule* pour les objets découverts
query_fields	array	Tableau des 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” - *lecture seule* pour les objets hérités ou les objets 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” - *lecture seule* pour les objets hérités ou les objets découverts

Propriété	Type	Description
retrieve_mode	integer	Quelle partie de la réponse doit être stockée. 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” - *lecture seule* pour les objets hérités ou les objets découverts
snmp_oid	string	OID SNMP. Comportement de la propriété : - *obligatoire* si `type` est défini sur ”SNMP agent” - *lecture seule* pour les objets hérités ou les objets 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” - *lecture seule* pour les objets hérités ou les objets découverts
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” - *lecture seule* pour les objets hérités ou les objets 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” - *lecture seule* pour les objets hérités ou les objets découverts
state	integer	État de l’élément. Valeurs possibles : 0 - *(par défaut)* normal ; 1 - non pris en charge. Comportement de la propriété : - *lecture seule*
status	integer	Statut de l’élément. Valeurs possibles : 0 - *(par défaut)* élément activé ; 1 - élément désactivé.
status_codes	string	Plages des codes d’état HTTP requis, séparées par des virgules. Prend également en charge les macros utilisateur comme partie d’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” - *lecture seule* pour les objets hérités ou les objets découverts

Propriété	Type	Description
templateid	ID	ID de l'élément du modèle parent. *Astuce* : utilisez la propriété `hostid` pour spécifier le modèle auquel l'élément appartient. Comportement de la propriété : - *lecture seule*
timeout	string	Délai d'expiration de la requête d'interrogation des données de l'élément. Accepte des secondes ou une unité de temps avec suffixe (par exemple, 30s, 1m). Accepte également les macros utilisateur. Plage de valeurs possible : 1-600s. Par défaut : "" - utiliser les paramètres proxy/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) - *lecture seule* pour les objets hérités et découverts
trapper_hosts	string	Hôtes autorisés. Comportement de la propriété : - *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 to accept incoming data"
trends	string	Unité de temps indiquant combien de temps les données de tendances doivent être conservées. Accepte également une macro utilisateur. Par défaut : 365d. Comportement de la propriété : - *pris en charge* si `value_type` est défini sur "numeric float" ou "numeric unsigned" - *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" - *lecture seule* pour les objets hérités ou les objets 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" - *lecture seule* pour les objets hérités (si `type` est défini sur "HTTP agent") ou les objets découverts
uuid	string	Identifiant universel unique, utilisé pour lier 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

Propriété	Type	Description
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” - *lecture seule* pour les objets hérités ou les objets découverts
verify_host	integer	Indique s’il faut vérifier que le nom d’hôte de la connexion correspond à celui figurant 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” - *lecture seule* pour les objets hérités ou les objets découverts
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” - *lecture seule* pour les objets hérités ou les objets découverts

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*

Propriété	Type	Description
value	string	Valeur du paramètre. Comportement de la propriété : - *obligatoire*

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 des éléments

L'objet de prétraitement des éléments possède les propriétés suivantes.

Propriété	Type	Description
type	integer	Le type d'option de prétraitement. Valeurs possibles : 1 - Multiplicateur personnalisé ; 2 - Supprimer à droite ; 3 - Supprimer à gauche ; 4 - Supprimer ; 5 - Expression régulière ; 6 - Booléen vers décimal ; 7 - Octal vers décimal ; 8 - Hexadécimal vers décimal ; 9 - Changement simple ; 10 - Changement par seconde ; 11 - XML XPath ; 12 - JSONPath ; 13 - Dans l'intervalle ; 14 - Correspond à l'expression régulière ; 15 - Ne correspond pas à l'expression régulière ; 16 - Vérifier une erreur dans JSON ; 17 - Vérifier une erreur dans XML ; 18 - Vérifier une erreur à l'aide d'une expression régulière ; 19 - Ignorer si inchangé ; 20 - Ignorer si inchangé avec pulsation ; 21 - JavaScript ; 22 - Modèle Prometheus ; 23 - Prometheus vers JSON ; 24 - CSV vers JSON ; 25 - Remplacer ; 26 - Vérifier non pris en charge ; 27 - XML vers JSON ; 28 - Valeur SNMP walk ; 29 - SNMP walk vers JSON ; 30 - Valeur SNMP get. Comportement de la propriété : - *obligatoire*
params	string	Paramètres supplémentaires utilisés par l'option de prétraitement. Les paramètres multiples sont séparés par le caractère de nouvelle ligne (\n). Si `type` est défini sur "Check unsupported", les paramètres suivent la syntaxe `<scope>[\n<pattern>]`, où *pattern* est une expression régulière et *scope* est l'un des suivants : -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 : - *obligatoire* si `type` est défini sur "Custom multiplier" (1), "Right trim" (2), "Left trim" (3), "Trim" (4), "Regular expression" (5), "XML XPath" (11), "JSONPath" (12), "In range" (13), "Matches regular expression" (14), "Does not match regular expression" (15), "Check for error in JSON" (16), "Check for error in XML" (17), "Check for error using regular expression" (18), "Discard unchanged with heartbeat" (20), "JavaScript" (21), "Prometheus pattern" (22), "Prometheus to JSON" (23), "CSV to JSON" (24), "Replace" (25), Check unsupported (26), "SNMP walk value" (28), "SNMP walk to JSON" (29) ou "SNMP get value" (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. Valeurs possibles si `type` est défini sur "Check unsupported" : 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 "Custom multiplier" (1), "Regular expression" (5), "Boolean to decimal" (6), "Octal to decimal" (7), "Hexadecimal to decimal" (8), "Simple change" (9), "Change per second" (10), "XML XPath" (11), "JSONPath" (12), "In range" (13), "Matches regular expression" (14), "Does not match regular expression" (15), "Check for error in JSON" (16), "Check for error in XML" (17), "Check for error using regular expression" (18), "Prometheus pattern" (22), "Prometheus to JSON" (23), "CSV to JSON" (24), "Check unsupported" (26), "XML to JSON" (27), "SNMP walk value" (28), "SNMP walk to JSON" (29) ou "SNMP get value" (30)
error_handler_params	string	Paramètres du gestionnaire d'erreurs. Comportement de la propriété : - *obligatoire* si `error_handler` est défini sur "Set custom value" ou "Set custom error message"

Les paramètres suivants et les 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	number[1, 6]			0, 1, 2, 3
2	Supprimer à droite	liste de caractères[2]			
3	Supprimer à gauche	liste de caractères[2]			
4	Supprimer	liste de caractères[2]			
5	Expression régulière	pattern[3]	output[2]		0, 1, 2, 3
6	Booléen vers décimal				0, 1, 2, 3
7	Octal vers décimal				0, 1, 2, 3

Type de prétraitement	Nom	Paramètre 1	Paramètre 2	Paramètre 3	Gestionnaires d’erreurs pris en charge
8	Hexadécimal vers décimal				0, 1, 2, 3
9	Changement simple				0, 1, 2, 3
10	Changement par seconde				0, 1, 2, 3
11	XML XPath	path[4]			0, 1, 2, 3
12	JSONPath	path[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	pattern[3]			0, 1, 2, 3
15	Ne correspond pas à l’expression régulière	pattern[3]			0, 1, 2, 3
16	Vérifier une erreur dans JSON	path[4]			0, 1, 2, 3
17	Vérifier une erreur dans XML	path[4]			0, 1, 2, 3
18	Vérifier une erreur à l’aide d’une expression régulière	pattern[3]	output[2]		0, 1, 2, 3
19	Ignorer si inchangé				
20	Ignorer si inchangé avec pulsation	seconds[5, 6]			

Type de prétraitement	Nom	Paramètre 1	Paramètre 2	Paramètre 3	Gestionnaires d'erreurs pris en charge
21	JavaScript	script[2]			
22	Modèle Prometheus	pattern[6, 7]	`value`, `label`, `function`	output[8, 9]	0, 1, 2, 3
23	Prometheus vers JSON	pattern[6, 7]			0, 1, 2, 3
24	CSV vers JSON	character[2]	character[2]	0,1	0, 1, 2, 3
25	Remplacer	chaîne de recherche[2]	remplacement[2]		
26	Check un-sup-ported	scope[1]	pattern[3, 6]		1, 2, 3
27	XML vers JSON				0, 1, 2, 3
28	Valeur SNMP walk	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	SNMP walk vers 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 à virgule flottante (les entiers sont convertis implicitement en valeurs flottantes)
[2] chaîne
[3] expression régulière
[4] JSONPath ou XML XPath
[5] entier positif (avec prise en charge des suffixes temporels, par ex. 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] Une des fonctions d'agrégation : `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 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'utilisateurs *Admin* et *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)` 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	Balises 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 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. Voir User roles 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 donnés.

> **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.

Paramètre	Type	Description
itemids	ID/array	Retourner uniquement les éléments avec les ID donnés.
groupids	ID/array	Retourner uniquement les éléments qui appartiennent aux hôtes des groupes donnés.
templateids	ID/array	Retourner uniquement les éléments qui appartiennent aux modèles donnés.
hostids	ID/array	Retourner uniquement les éléments qui appartiennent aux hôtes donnés.
proxyids	ID/array	Retourner uniquement les éléments qui sont surveillés par les proxies donnés.
interfaceids	ID/array	Retourner uniquement les éléments qui utilisent les interfaces d'hôte données.
graphids	ID/array	Retourner uniquement les éléments qui sont utilisés dans les graphiques donnés.
triggerids	ID/array	Retourner 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`, retourner uniquement les éléments hérités d'un modèle.
inheritedTags	boolean	Retourner les éléments qui ont les `tags` donnés également dans l'hôte/le modèle/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.
templated	boolean	Si défini sur `true`, retourner uniquement les éléments qui appartiennent à des modèles.
monitored	boolean	Si défini sur `true`, retourner uniquement les éléments activés qui appartiennent à des hôtes surveillés.
group	string	Retourner uniquement les éléments qui appartiennent à un groupe portant le nom donné.
host	string	Retourner uniquement les éléments qui appartiennent à un hôte portant le nom donné.

Paramètre	Type	Description
evaltype	integer	Méthode d'évaluation des tags. Valeurs possibles : 0 - *(par défaut)* And/Or ; 2 - Or.
tags	array	Retourner uniquement les éléments avec les tags donnés. Format : `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Un tableau vide retourne 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`, retourner uniquement les éléments qui sont utilisés dans des déclencheurs.
selectHosts	query	Retourner une propriété `hosts` avec un tableau des hôtes auxquels l'élément appartient.
selectInterfaces	query	Retourner une propriété `interfaces` avec un tableau des interfaces d'hôte utilisées par l'élément.
selectTriggers	query	Retourner une propriété `triggers` avec les déclencheurs dans lesquels l'élément est utilisé. Prend en charge `count`.
selectGraphs	query	Retourner une propriété `graphs` avec les graphiques qui contiennent l'élément. Prend en charge `count`.
selectDiscoveryData	query	Retourner 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 indiquant 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	Retourner une propriété `discoveryRule` avec la règle LLD qui a créé l'élément.
selectInheritedTags	query	Retourner une propriété `inheritedTags` avec les tags présents sur le modèle/l'hôte/les modèles liés.
selectPreprocessing	query	Retourner une propriété `preprocessing` avec les options de prétraitement de l'élément.
selectTags	query	Retourner les tags de l'élément dans la propriété `tags`.
selectValueMap	query	Retourner une propriété `valuemap` avec la table de correspondance des valeurs de l'élément.

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 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 retournés 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	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectItemDiscovery	query	Retourner 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 également

- Règle de découverte
- Graphique

- Hôte
- Interface de l'hôte
- Déclencheur

Source

CItem::get() dans *ui/include/classes/api/services/CItem.php*.

item.update

Description

`object item.update(object/array items)`

Cette méthode 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'utilisateurs *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 de l'élément 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 de l'élément, la méthode accepte les paramètres suivants.

Paramètre	Type	Description
preprocessing	array	Options de prétraitement de l'élément pour remplacer les options de prétraitement actuelles. Comportement du paramètre: - *lecture seule* pour les objets hérités ou les objets découverts
tags	array	Tags de l'élément. Comportement du paramètre: - *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 les 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 d'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 tracé 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 de données. Comportement de la propriété : - *obligatoire* pour les opérations de création
calc_fnc	integer	Valeur de l'élément de données 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.

Propriété	Type	Description
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 des éléments de graphique selon les paramètres donnés.

> **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.

Paramètre	Type	Description
graphids	ID/array	Retourner uniquement les éléments de graphique appartenant aux graphiques spécifiés.
itemids	ID/array	Retourner uniquement les éléments de graphique avec les ID d'élément spécifiés.
type	integer	Retourner uniquement les éléments de graphique du type spécifié. 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` contenant un tableau des graphiques auxquels l'élément appartient.
sortfield	string/array	Trier le résultat selon les propriétés spécifiées. Valeurs possibles : `gitemid`.
countOutput	boolean	Ces paramètres sont décrits dans les commentaires 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.

Propriété	Type	Description
eventid	ID	ID de l'événement.
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'auto-enregistrement de l'agent actif ; 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'auto-enregistrement de l'agent actif » : 3 - hôte auto-enregistré. Valeurs possibles si `source` est défini sur « événement interne » : 0 - déclencheur ; 4 - élément de données ; 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.

Propriété	Type	Description
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 ». Comportement de la propriété : - *pris en charge* 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 »
severity	integer	Gravité actuelle de l'événement. Valeurs possibles : 0 - non classé ; 1 - information ; 2 - avertissement ; 3 - moyenne ; 4 - élevée ; 5 - désastre.
r_eventid	ID	ID de l'événement de récupération.
c_eventid	ID	ID de l'événement qui a été 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 tag d'événement a les propriétés suivantes.

Propriété	Type	Description
tag	string	Nom du tag d'événement.
valeur	chaîne	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 ne contiendront des entrées que 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*.

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.
- Acquitter l'événement. Si l'événement est déjà acquitté, cette action sera ignorée.
- Annuler l'acquittement de l'événement. Si l'événement n'est pas acquitté, cette action sera ignorée.
- Ajouter un message.
- Modifier la sévérité de l'événement. Si l'événement a déjà la même sévérité, cette action sera ignorée.
- Supprimer 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.
Les droits de lecture/écriture sur le déclencheur sont requis pour fermer l'événement ou pour modifier la sévérité de l'événement.
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 User roles pour plus d'informations.

Paramètres

`(object/array)` Paramètres contenant les ID des événements et les opérations de mise à jour à effectuer.

Paramètre	Type	Description
eventids	ID/array	ID des événements à acquitter. Comportement du paramètre: - *obligatoire*

Paramètre	Type	Description
action	integer	Action(s) de mise à jour d'événement. Valeurs bitmap possibles : 1 - fermer le problème ; 2 - acquitter l'événement ; 4 - ajouter un message ; 8 - changer la sévérité ; 16 - désacquitter l'événement ; 32 - supprimer l'événement ; 64 - annuler la suppression de l'événement ; 128 - changer le rang de l'événement en cause ; 256 - changer le rang de l'événement en symptôme. Il s'agit d'un champ de masque de bits ; toute somme des valeurs bitmap possibles est acceptée (par exemple, 34 pour acquitter et supprimer l'événement). Comportement du paramètre: - *obligatoire*
cause_eventid	ID	ID de l'événement cause. Comportement du paramètre: - *obligatoire* si `action` contient le bit « changer le rang de l'événement en symptôme »
message	string	Texte du message. Comportement du paramètre: - *obligatoire* si `action` contient le bit « ajouter un message »
severity	integer	Nouvelle sévérité des événements. Valeurs possibles : 0 - non classé ; 1 - information ; 2 - avertissement ; 3 - moyen ; 4 - élevé ; 5 - désastre. Comportement du paramètre: - *obligatoire* si `action` contient le bit « changer 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: - *obligatoire* si `action` contient le bit « supprimer l'événement »

Valeurs retournées

`(object)` Retourne un objet contenant les identifiants des événements mis à jour sous 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
}
```

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 donnés.

> **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.

Paramètre	Type	Description
eventids	ID/array	Retourner uniquement les événements avec les ID donnés.
groupids	ID/array	Retourner uniquement les événements créés par des objets appartenant aux groupes d'hôtes donnés.
hostids	ID/array	Retourner uniquement les événements créés par des objets appartenant aux hôtes donnés.
objectids	ID/array	Retourner uniquement les événements créés par les objets donnés.
source	integer	Retourner uniquement les événements du type donné. Consultez la page de l'objet event pour obtenir la liste des types d'événements pris en charge. Par défaut : 0 - événements de déclencheur.
object	integer	Retourner uniquement les événements créés par des objets du type donné. Consultez la page de l'objet event pour obtenir la liste des types d'objets pris en charge. Par défaut : 0 - déclencheur.
acknowledged	boolean	Si défini sur `true`, retourner uniquement les événements acquittés.
action	integer	Retourner uniquement les événements pour lesquels les actions de mise à jour d'événement données ont été effectuées. Pour plusieurs actions, utilisez la somme de toutes les valeurs bitmap acceptables comme masque de bits (par exemple, 34 pour acquitter et supprimer un événement).
action_userids	ID/array	Retourner uniquement les événements avec les ID donnés des utilisateurs ayant effectué les actions de mise à jour d'événement.
suppressed	boolean	Si défini sur `true`, retourner uniquement les événements supprimés.
symptom	boolean	Si défini sur `true`, retourner uniquement les événements symptomatiques.
severities	integer/array	Retourner uniquement les événements avec les gravités d'événement données. S'applique uniquement si l'objet est un déclencheur.
trigger_severities	integer/array	Retourner uniquement les événements avec les gravités de déclencheur données. S'applique uniquement si l'objet est un déclencheur.
evaltype	integer	Méthode d'évaluation des tags. Valeurs possibles : 0 - *(par défaut)* Et/Ou ; 2 - Ou.
tags	array	Retourner uniquement les événements avec les tags donnés. Format : `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Un tableau vide retourne tous les événements. Valeurs possibles de l'opérateur : 0 - *(par défaut)* Contient ; 1 - Égal ; 2 - Ne contient pas ; 3 - N'est pas égal ; 4 - Existe ; 5 - N'existe pas.

Paramètre	Type	Description
eventid_from	string	Retourner uniquement les événements avec des ID supérieurs ou égaux à l'ID donné.
eventid_till	string	Retourner uniquement les événements avec des ID inférieurs ou égaux à l'ID donné.
time_from	timestamp	Retourner uniquement les événements créés après ou à l'heure donnée.
time_till	timestamp	Retourner uniquement les événements créés avant ou à l'heure donnée.
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.
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	Retourner uniquement les événements avec les valeurs données.
selectAcknowledges	query	Retourner une propriété `acknowledges` avec les mises à jour d'événement. Les mises à jour d'événement sont triées dans l'ordre chronologique inverse. L'objet de mise à jour d'événement possède les propriétés suivantes : `acknowledgeid` - `(ID)` ID de l'acquittement ; `userid` - `(ID)` ID de l'utilisateur ayant 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 qui a été 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 ; `username` - `(string)` nom d'utilisateur de l'utilisateur ayant mis à jour l'événement ; `name` - `(string)` prénom de l'utilisateur ayant mis à jour l'événement ; `surname` - `(string)` nom de famille de l'utilisateur ayant mis à jour l'événement. Prend en charge `count`.
selectAlerts	query	Retourner une propriété `alerts` avec les alertes générées par l'événement. Les alertes sont triées dans l'ordre chronologique inverse.
selectHosts	query	Retourner 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	Retourner une propriété `relatedObject` avec l'objet qui a créé l'événement. Le type d'objet retourné dépend du type d'événement.
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 l'événement ; `suppress_until` - `(integer)` heure jusqu'à laquelle l'événement est supprimé.
selectTags	query	Retourner une propriété `tags` avec les tags d'événement.

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 à faire correspondre. Ne prend pas en charge les propriétés du type de données `text`.
sortfield	string/array	Trier le résultat selon les propriétés données. Valeurs possibles : `eventid`, `objectid`, `clock`. Valeurs possibles lorsqu'utilisé avec `groupBy` : `objectid`.
groupBy	string/array	Valeurs possibles lorsqu'utilisé avec `countOutput` et `groupBy` : `objectid`, `rowscount`. 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": "Load average is too high (per CPU load over 1.5 for 5m)",
            "severity": "0",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "acknowledges": [],
            "opdata": "Load averages(1m 5m 15m): (1.6328 3.0522 2.2515), # of CPUs: 2",
            "suppression_data": [],
            "suppressed": "0",
            "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": "Load average is too high (per CPU load over 1.5 for 5m)",
            "severity": "3",
            "r_eventid": "20",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
```

```
            "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",
                    "username": "Admin",
                    "name": "Zabbix",
                    "surname": "Administrator"
                }
            ],
            "opdata": "Load averages(1m 5m 15m): (1.6328 3.0522 2.2515), # of CPUs: 2",
            "suppression_data": [
                {
                    "maintenanceid": "0",
                    "suppress_until": "1728744338",
                    "userid": "1"
                }
            ],
            "suppressed": "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 acquittés par l'utilisateur spécifié

Récupération des événements acquittés par l'utilisateur avec 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",
            "cause_eventid": "0",
            "acknowledges": [
                {
                    "userid": "10",
                    "action": "2"
                }
            ],
            "opdata": "",
            "suppressed": "0",
            "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*.

Annexe 1. Commentaire de référence

Notation 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	Retourner le nombre d'enregistrements dans le résultat au lieu des données réelles.
editable	boolean	Si défini sur `true`, retourner uniquement les objets pour lesquels l'utilisateur dispose des droits d'écriture. Par défaut : `false`.
excludeSearch	boolean	Retourner les résultats qui ne correspondent pas aux critères indiqués dans le paramètre `search`.
filter	object	Retourner 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 de type de données `text` data type. 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 Host).
limit	integer	Limiter le nombre d'enregistrements retournés.
output	query	Propriétés de l'objet à retourner. Par défaut : `extend`.
preservekeys	boolean	Utiliser les ID comme clés dans le tableau résultant.
search	object	Retourner les résultats qui correspondent au motif donné (insensible à la 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 uniquement en charge les propriétés de type de données `string` et `text` data type. 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 Host).
searchByAny	boolean	Si défini sur `true`, retourner 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`.

Paramètre	Type	Description
sortfield	string/array	Trier le résultat selon les propriétés indiquées. Reportez-vous à la description d'une méthode get spécifique de l'API 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 retournées 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 qu'il 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.
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 historique

ZBXNEXT-10069 Ajout du nouveau paramètre `maxValueSize` à la méthode history.get. Si vous demandez des données pour des éléments avec une valeur binaire (`history=5`), définissez `maxValueSize` sur `null` ; sinon, `history.get` renverra la valeur de l'élément tronquée à 64 Kio.

hôte

ZBXNEXT-10387 `host.massupdate` : Méthode supprimée. Veuillez utiliser `host.update` à la place.

hostgroup

ZBXNEXT-10387 `hostgroup.massupdate` : méthode supprimée. Veuillez utiliser `hostgroup.update`, `hostgroup.massadd` ou `hostgroup.massremove` à la place.

hostinterface

ZBXNEXT-10387 `hostinterface.replacehostinterfaces` : méthode supprimée. Veuillez utiliser `hostinterface.update`, `hostinterface.massadd` ou `hostinterface.massremove` à la place.

modèle

ZBXNEXT-10387 `template.massupdate` : Méthode supprimée. Veuillez utiliser `template.update`, `template.massadd` ou `template_massremove` à la place.

templategroup

ZBXNEXT-10387 `templategroup.massupdate` : Méthode supprimée. Veuillez utiliser `templategroup.update`, `templategroup.massac`
ou `templategroup.massremove` à la place.

Autres modifications et corrections de bugs connecteur

ZBXNEXT-10069 Ajout d'une nouvelle valeur `item_value_type` (64 - JSON) à l'objet connector.

tableau de bord

ZBXNEXT-4769 Ajout du nouveau champ de widget de tableau de bord `show_hostnames` au widget graph.
ZBXNEXT-9546 Ajout des nouveaux champs de widget de tableau de bord `ds.0.item_tags_evaltype`, `ds.0.item_tags.0.tag`, `ds.0.item_tags.0.operator` et `ds.0.item_tags.0.value` aux widgets svggraph et piechart.
ZBXNEXT-10158 Ajout des nouveaux champs de widget de tableau de bord `columns.0.aggregate_columns`, `columns.0.combined_colu`
et `columns.0.column_aggregate_function` au widget topitems.
ZBXNEXT-9424, ZBXNEXT-4769 Ajout du nouveau type de widget de tableau de bord scatterplot.
ZBXNEXT-8160 Ajout des nouveaux champs de widget de tableau de bord `clustering_mode` et `clustering_zoom_level` au widget geomap.

templatedashboard

ZBXNEXT-9424 Ajout d'un nouveau type de widget de tableau de bord de modèle scatterplot.

historique

ZBXNEXT-10069 Ajout de la nouvelle valeur `history` (6 - JSON) à la méthode history.get.
ZBXNEXT-10069 Ajout du nouvel objet `JSON history`.

hostprototype

ZBXNEXT-9546 Ajout du nouveau paramètre `selectInheritedTags` à la méthode hostprototype.get.

httptest

ZBXNEXT-9546 Ajout des nouveaux paramètres `inheritedTags` et `selectInheritedTags` à la méthode httptest.get.

élément

ZBXNEXT-10069 Ajout d'une nouvelle valeur `value_type` (6 - JSON) à l'objet item.
ZBXNEXT-9546 Ajout des nouveaux paramètres `inheritedTags` et `selectInheritedTags` à la méthode item.get.

itemprototype

ZBXNEXT-10069 Ajout d'une nouvelle valeur `value_type` (6 - JSON) à l'objet itemprototype.
ZBXNEXT-9546 Ajout du nouveau paramètre `selectInheritedTags` à la méthode itemprototype.get.

modèle

ZBXNEXT-9546 Ajout des nouveaux paramètres `inheritedTags` et `selectInheritedTags` à la méthode template.get.

déclencheur

ZBXNEXT-9546 Ajout des nouveaux paramètres `inheritedTags` et `selectInheritedTags` à la méthode trigger.get.

triggerprototype

ZBXNEXT-9546 Ajout du nouveau paramètre `selectInheritedTags` à la méthode triggerprototype.get.

triggertag

ZBXNEXT-5467 Ajout de la propriété `automatic` à l'objet trigger tag.

userdirectory

ZBXNEXT-9834 Propriétés `idp_certificate`, `sp_private_key`, `sp_certificate`, `idp_certificate_hash`, `sp_private_key_hash` et `sp_certificate_hash` ajoutées à l'objet user directory.

Annexe 3. Modifications dans la version 8.0

8.0.1 Cette version n'est pas encore publiée.

19 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 ou au proxy Zabbix, par exemple des données périodiques de disponibilité et de performance dans le cas de scripts utilisateur de longue durée d'exécution. L'envoi de données au serveur ou au proxy Zabbix 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 d'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 Zabbix agent `system.run[]`

L'élément agent Zabbix `system.run[]` est un élément pour 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 en cours d'exécution.

Remarque : 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 l'agent 2 (paramètre de configuration `AllowKey`).

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 d'agent HTTP

L'élément HTTP agent est un élément permettant d'exécuter des requêtes de données via HTTP/HTTPS. Les éléments d'agent HTTP sont utiles pour envoyer des requêtes à des points de terminaison HTTP afin de récupérer des données depuis des services tels que *Elasticsearch* et *OpenWeatherMap*, pour vérifier l'état de l'API Zabbix ou l'état du serveur web Apache ou Nginx, etc. Les éléments d'agent HTTP (avec le trapping activé) 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 constituent une alternative aux modules chargeables (écrits en C).

Un plugin est un package Go qui définit la structure et implémente une ou plusieurs interfaces de plugin (*Exporter*, *Collector*, *Configurator*, *Runner*, *Watcher*). Deux types de plugins Zabbix agent 2 sont pris en charge :

- Plugins intégrés (pris en charge depuis Zabbix 4.4.0)
- Plugins chargeables (pris en charge depuis Zabbix 6.0.0)

Consultez la liste des plugins fournis par Zabbix.

Pour des instructions et des tutoriels sur la création de vos propres plugins, consultez 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.

1 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 du code source, 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_history_write_c** 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" est défini sur NULL.

```
void    zbx_module_item_timeout(int timeout);
```

Si le module exporte **zbx_module_item_list()**, cette fonction est alors utilisée par Zabbix pour spécifier les paramètres de délai d'expiration définis dans le fichier de configuration de 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 de 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), par exemple la libération des ressources allouées, la fermeture des descripteurs de fichiers, etc.

Toutes les fonctions sont appelées une fois au démarrage de Zabbix lorsque le module est chargé, à l'exception de zbx_module_uninit(), qui est appelée une fois à l'arrêt de Zabbix lorsque le module est déchargé.

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 ex. « dummy.random »), **flags** vaut soit CF_HAVEPARAMS, soit 0 (selon que l'élément accepte des paramètres ou non), **function** est une fonction C qui implémente l'élément (par ex. « 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 ex. « 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 pointeurs, 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 depuis AGENT_REQUEST et de définir des informations dans AGENT_RESULT.

Fourniture de callbacks d'exportation de l'historique

> **Attention:**
> L'exportation de l'historique via un module n'est plus prise en charge par Zabbix proxy.

Le module peut spécifier des fonctions pour exporter les données d'historique par type : numérique (float), numérique (unsigned), caractère, texte et journal :

```
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;
```

Chacune d'elles doit prendre comme arguments le tableau "history" de "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 l'écriture des données dans la base de données Zabbix et leur enregistrement dans le cache de valeurs.

Attention:
En cas d'erreur interne dans le module d'exportation de l'historique, il est recommandé d'écrire le module de manière à ce qu'il ne bloque pas l'ensemble de la supervision jusqu'à son rétablissement, mais qu'il ignore plutôt les données et permette 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**.

En gardant ces informations à l'esprit, 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 de 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

Zabbix agent, 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 être situés dans un 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. Il est permis d'inclure plusieurs paramètres LoadModule.

Par exemple, pour étendre Zabbix agent, nous pourrions 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/agent, tandis que dummy.so sera chargé depuis /usr/local/lib/zabbix. L'agent ne pourra pas démarrer si un module est manquant, en cas de permissions incorrectes ou si une bibliothèque partagée n'est pas un module Zabbix.

Configuration du frontend

Les modules chargeables sont pris en charge par l'agent Zabbix, le serveur et le proxy. Par conséquent, le type d'élément dans le frontend Zabbix dépend de l'endroit où le module est chargé. Si le module est chargé dans l'agent, le type d'élément doit alors être « Agent Zabbix » ou « Agent Zabbix (actif) ». Si le module est chargé dans le serveur ou le proxy, le type d'élément doit alors être « Vérification simple ».

L'export de l'historique via les modules Zabbix ne nécessite aucune configuration du frontend. 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 factice

Zabbix inclut un exemple de module 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 et 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**.

```
/*
** 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"

/* the variable keeps timeout setting for item processing */
static int  item_timeout = 0;

/* module SHOULD define internal functions as static and use a naming pattern different from Zabbix intern
/* symbols (zbx_*) and loadable module API functions (zbx_module_*) to avoid conflicts
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;
}
```

```c
/******************************************************************************
 *                                                                            *
 * 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 la plage 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*. Cela n'est actuellement pas pris en charge. 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.

3 Modules du frontend

Vue d'ensemble

Il est possible d'améliorer les fonctionnalités du frontend 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 que :

- 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 le frontend. Si cela se produit, supprimez simplement le code du module du frontend. Dès que vous rechargez le frontend Zabbix, vous verrez une note indiquant que certains modules sont absents. Accédez à Administration des modules (dans *Administration* → *Général* → *Modules*) et cliquez de 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 de 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 dossier `modules` du frontend Zabbix
- Assurez-vous que le dossier de votre module contient au minimum le fichier manifest.json
- Accédez à l'administration des modules et cliquez sur le bouton *Scan directory*
- 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
Le module n'est pas apparu dans la liste	Si votre module n'est pas apparu 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 le fichier existe, le module peut ne pas être compatible avec votre version de Zabbix. Vérifiez également que l'utilisateur du serveur web dispose au minimum des droits de lecture et de parcours (`r-x`) sur le dossier du module et tous ses sous-répertoires, ainsi que des droits de lecture (`r--`) sur tous les fichiers qu'il contient.
Le frontend a cessé de fonctionner	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 le frontend. Vous verrez un avis indiquant que certains modules sont absents. Accédez à l'administration des modules et cliquez de nouveau sur *Scan directory* pour supprimer de la base de données les modules inexistants.
Un message d'erreur concernant un espace de noms, un ID ou des actions identiques apparaît	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.
Des messages d'erreur techniques apparaissent	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.

20 Annexes

Veuillez utiliser la barre latérale pour accéder au contenu de la section Annexes.

1 Installation et configuration

Veuillez utiliser la barre latérale pour accéder au contenu de cette section.

1 Création de la base de données

Aperçu

Une base de données Zabbix doit être créée lors de l'installation du serveur ou du proxy Zabbix.

Cette section fournit des instructions pour créer une base de données Zabbix. Un ensemble d'instructions distinct 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 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

Les jeux de caractères utf8 (également appelé utf8mb3) et utf8mb4 sont pris en charge (avec les interclassements utf8_bin et utf8mb4_bin respectivement) afin que le serveur/proxy Zabbix fonctionne correctement avec une base de données MySQL. Il est recommandé d'utiliser utf8mb4 pour les nouvelles installations.

Les déclencheurs déterministes doivent être créés lors de l'importation du schéma. Sur MySQL et MariaDB, cela nécessite que `GLOBAL log_bin_trust_function_creators = 1` soit défini si la journalisation binaire est activée, qu'il n'y a pas de privilèges superutilisateur et que `log_bin_trust_function_creators = 1` n'est pas défini dans le fichier de configuration MySQL.

Si vous installez à partir des **paquets** Zabbix, suivez les instructions pour votre plateforme.

Si vous installez Zabbix à partir des sources :

- Créez et configurez une base de données et un utilisateur.

```
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 utf8mb4 comme jeu de caractères 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
#### arrêtez-vous ici si vous créez une base de données pour le proxy Zabbix
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 après l'importation réussie du schéma :

```
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 permissions nécessaires pour créer des objets de base de données.

Si vous installez à partir des **paquets** Zabbix, consultez les instructions correspondant à votre plateforme.

Si vous installez Zabbix à partir des sources :

- Créez un utilisateur de base de données.

La commande shell suivante créera l'utilisateur `zabbix`. Indiquez un mot de passe lorsque vous y êtes invité, puis saisissez-le à nouveau (notez qu'il se peut que le mot de passe `sudo` vous soit d'abord demandé) :

```
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 précédemment créé 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 Zabbix). Pour une base de données de proxy Zabbix, seul `schema.sql` doit être importé (`images.sql` et `data.sql` ne doivent pas l'être).

```
cd database/postgresql
cat schema.sql | sudo -u zabbix psql zabbix
#### arrêtez-vous ici si vous créez une base de données pour le proxy Zabbix
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 des commandes différentes 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.

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.

2 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 par MySQL du standard UTF8, qui est en réalité sur 4 octets. À partir de MySQL 8.0.28 et MariaDB 10.6.1, le jeu de caractères « utf8mb3 » est déprécié et, à terme, sa prise en charge sera abandonnée, tandis que « utf8 » deviendra une référence à « utf8mb4 ». Depuis Zabbix 6.0, « utf8mb4 » est pris en charge. Afin d'éviter des problèmes à l'avenir, 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 prennent pas en charge utf8mb4, assurez-vous d'abord de 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 vers utf8mb4.

1. Vérifiez le jeu de caractères et le classement 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 nous pouvons le voir, le jeu de caractères ici n'est pas « utf8mb4 » et le classement n'est pas « utf8mb4_bin », nous devons donc les corriger.

2. Arrêtez Zabbix.

3. Créez une copie de sauvegarde de la base de données !

4. Corrigez le jeu de caractères et le classement 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 le classement au niveau des tables et des colonnes :

```
mysql <your DB name> < utf8mb4_convert.sql
```

6. Exécutez le script :

```
                SET @ZABBIX_DATABASE = '<your DB name>';
If MariaDB →    set innodb_strict_mode = OFF;
                CALL zbx_convert_utf8();
If MariaDB →    set innodb_strict_mode = ON;
                drop procedure zbx_convert_utf8;
```

Veuillez noter que « utf8mb4 » devrait consommer légèrement plus d'espace disque.

7. S'il n'y a pas d'erreurs, vous pouvez créer une copie de sauvegarde de la base de données corrigée.

8. Démarrez Zabbix.

3 Mise à niveau de la base de données vers des clés primaires

Aperçu

Cette section fournit des instructions pour mettre à niveau manuellement les tables des installations existantes vers des clés primaires.

La mise à niveau vers des clés primaires optimise la façon dont les données sont indexées et consultées, ce qui peut accélérer les requêtes et économiser de l'espace. Elle améliore également la gestion et la synchronisation des données dans les configurations

en cluster, ce qui facilite la montée en charge et garantit que le système reste fiable même en cas de défaillance de certains serveurs.

Attention:
Les instructions fournies sur cette page sont destinées aux utilisateurs avancés et peuvent nécessiter des ajustements en fonction de 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 afin qu'elles utilisent 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 d'abord de mettre à niveau les tables vers la double précision. Pour plus d'informations, consultez Mise à niveau vers des valeurs numériques à plage étendue dans la documentation de Zabbix 7.0.

Des instructions sont disponibles pour :

- 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.

MySQL

L'exportation et l'importation doivent être effectuées dans tmux/screen pour s'assurer que la session n'est pas abandonnée.

Voir également : 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 durée de 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 qu'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 MySQL Shell officiel 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 avec tout 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 tables 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 délai d'expiration 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 les chemins des fichiers dans les requêtes en conséquence ou définissez la valeur de *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 contenant les données des tables exportées ('/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'exportation des données afin d'éviter un délai d'expiration pendant l'exportation.

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 effectués 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 le serveur arrêté

- Exportez l'historique actuel, importez-le dans la table temporaire, puis insérez les données dans les nouvelles tables 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 des conseils pour améliorer les performances des 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 un 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 des 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

4 Préparation de la table d'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

Remarques importantes sur la reconstruction des index

MySQL reconstruit automatiquement les index de la clé primaire pendant l'opération `ALTER TABLE`. Cependant, il est fortement recommandé de reconstruire également manuellement les index à l'aide de l'instruction `OPTIMIZE TABLE` afin de garantir des performances optimales de la base de données.

La reconstruction des index peut temporairement nécessiter autant d'espace disque supplémentaire que la table elle-même en utilise. 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 diminuer le paramètre *Période de conservation des données* de housekeeper avant d'exécuter le housekeeper.

Modification de la clé primaire avec interruption de service

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 Remarques importantes sur la reconstruction des index).

```
OPTIMIZE TABLE auditlog;
```

Modification de la clé primaire sans interruption de service

La méthode manuelle de modification de la clé primaire est décrite ici. Vous pouvez également utiliser l'outil pt-online-schema-change de Percona. Cet outil effectue automatiquement les actions suivantes, tout en minimisant l'espace utilisé pour modifier la table `auditlog`.

1. Créer une nouvelle table avec la nouvelle clé primaire et créer 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. Permuter les tables.

```
RENAME TABLE auditlog TO auditlog_old, auditlog_new TO auditlog;
```

3. Copier les données de l'ancienne table vers la nouvelle table.

```
INSERT INTO auditlog SELECT * FROM auditlog_old;
```

Cela peut être fait par blocs (plusieurs instructions `INSERT INTO` avec des clauses `WHERE clock` selon les besoins) afin d'éviter une utilisation excessive des ressources.

4. Supprimer l'ancienne table.

```
DROP TABLE auditlog_old;
```

PostgreSQL

Remarques 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 temporairement nécessiter 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 un sujet de préoccupation, suivez les instructions de Modification de la clé primaire sans interruption de service. 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 tablespace avec davantage d'espace disponible, vous pouvez envisager de modifier l'emplacement de stockage temporaire pour la reconstruction des index. Vous pouvez définir le paramètre PostgreSQL `temp_tablespaces` afin de 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 conservation des données* de housekeeper avant d'exécuter le housekeeper.

Modification de la clé primaire avec interruption de service

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 Remarques importantes sur la reconstruction des index).

```
REINDEX TABLE CONCURRENTLY auditlog;
```

Modification de la clé primaire sans interruption de service

La méthode manuelle de modification de la clé primaire est décrite ici. Comme alternative, 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 effectué par lots (plusieurs instructions `INSERT INTO` avec des clauses `WHERE clock` selon les besoins) afin d'éviter une utilisation excessive des ressources.

4. Supprimez l'ancienne table.

```
DROP TABLE auditlog_old;
```

Voir aussi

- Mise à niveau de la base de données vers des clés primaires

5 Connexion sécurisée à la base de données

Aperçu

Cette section fournit les étapes de configuration de Zabbix et des exemples de configuration pour des connexions TLS sécurisées entre :

Base de données	Composants Zabbix
MySQL	Zabbix frontend, Zabbix server, Zabbix proxy
PostgreSQL	Zabbix frontend, Zabbix server, Zabbix proxy

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 : serveurs de base de données de réplication de groupe, etc.
- PostgreSQL options de chiffrement.

Tous les exemples sont basés sur les versions GA de MySQL CE (8.0) et de PgSQL (13) disponibles dans les dépôts officiels utilisant CentOS 8.

Spécifications

Les éléments suivants sont requis pour configurer le chiffrement :

- Le système d'exploitation pris en charge par les développeurs avec OpenSSL >=1.1.X ou alternative.

Note:
Il est recommandé d'éviter les systèmes d'exploitation en fin de vie, en particulier dans le cas de nouvelles installations

- Le moteur de base de données (RDBMS) installé et maintenu à partir du référentiel officiel fourni par le développeur. Les systèmes d'exploitation sont souvent livrés avec des versions de logiciels de base de données obsolètes pour lesquelles la prise en charge du chiffrement n'est pas mise en œuvre, 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 le frontend 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 de Zabbix

Interface vers la base de données

Une connexion sécurisée à la base de données peut être configurée pendant 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 cryptage avec certificats.

Note:
Pour MySQL, la case à cocher *Chiffrement TLS de la base de données* est désactivée, si *Hôte de la base de données* est défini sur localhost, car la connexion qui utilise un fichier socket (sous Unix) ou de la mémoire partagée (sous Windows) ne peut pas être cryptée.
Pour PostgreSQL, la case à cocher *Chiffrement TLS* est désactivée, si la valeur du champ *Hôte de la base de données* commence par une barre oblique ou le champ est vide.

Les paramètres suivants deviennent disponibles dans le chiffrement TLS en mode certificats (si les deux cases sont cochées) :

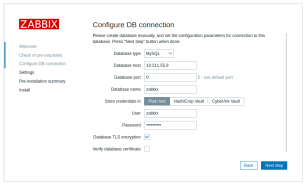
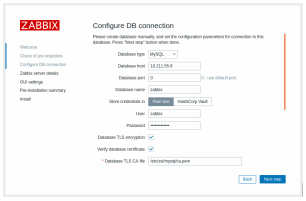
Paramètre	Description
Fichier CA TLS de la base de données	Spécifiez le chemin d'accès complet à un fichier d'autorité de certification TLS valide.
Fichier de clé TLS de la base de données	Spécifiez le chemin d'accès complet à un fichier de clé TLS valide.
Fichier de certificat TLS de base de données	Spécifiez le chemin d'accès complet à 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 pair.
Liste de chiffrement TLS de la base de données	Spécifiez une liste personnalisée de chiffrements valides. Le format de la liste de chiffrement doit être conforme à la norme OpenSSL. Disponible pour MySQL uniquement.

Attention:
Les paramètres TLS doivent pointer vers des fichiers valides. Si ils pointent vers des fichiers inexistants ou invalides, cela conduira à une erreur d'autorisation.
Si les fichiers de certificat sont accessibles en écriture, l'interface génère un avertissement dans les informattions système signalant que "les fichiers de certificat TLS doivent être en lecture seule". (affiché uniquement si l'utilisateur PHP est le propriétaire du certificat).

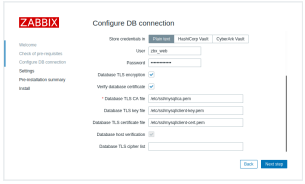
Les certificats protégés par des mots de passe ne sont pas pris en charge.

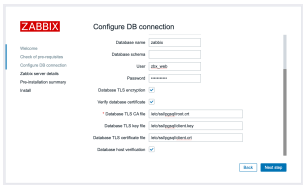
Cas d'utilisation

Le frontend Zabbix utilise l'interface graphique (GUI) pour définir les options possibles : `required`, `verify_ca`, `verify_full`. Spécifiez les options requises à l'étape *Configure DB connections* de l'assistant d'installation. Ces options sont mappées vers le fichier de configuration (zabbix.conf.php) de la manière suivante :

Paramètres GUI	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'] = ''; ...	Cochez *Database TLS encryption* Laissez *Verify database certificate* 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 *Database TLS encryption* et *Verify database certificate* 2. Indiquez le chemin vers *Database TLS CA file*	Active le mode `verify_ca`.

Paramètres GUI	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>';
	...
	
	Or:
	
	...
	// 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 *Database TLS encryption* et *Verify database certificate*
		2. Indiquez le chemin vers *Database TLS key file*
		3. Indiquez le chemin vers *Database TLS CA file*
		4. Indiquez le chemin vers *Database TLS certificate file*
		5. Indiquez *Database TLS cipher list* (facultatif)
			Active le mode `verify_full` pour MySQL.

	... $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 *Database TLS encryption* et *Verify database certificate* 2. Indiquez le chemin vers *Database TLS key file* 3. Indiquez le chemin vers *Database TLS CA file* 4. Indiquez le chemin vers *Database TLS certificate file* 5. Cochez *Database host verification*	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 avec les paramètres correspondants dans le fichier de configuration du server Zabbix et/ou du proxy.

Configuration	Résultat
Aucune	Connexion à la base de données sans chiffrement.
1. 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 du 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.
1. Définir `DBTLSCipher` - la liste des suites de chiffrement que le client autorise pour les connexions utilisant des protocoles TLS jusqu'à TLS 1.2 ou `DBTLSCipher13` - la liste des suites 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'une suite de chiffrement de la liste fournie. (PostgreSQL) La définition de cette option sera considérée comme une erreur.

1 Configuration du chiffrement MySQL

Aperçu

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 le frontend Zabbix et la base de données utilise un fichier socket (sous Unix) ou la mémoire partagée (sous Windows) et ne peut pas être chiffrée.

Note:
La liste des combinaisons de chiffrement n'est pas limitée à celles indiquées sur cette page. Il existe beaucoup d'autres combinaisons disponibles.

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 composants principaux :

```
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 de détails sur la configuration des utilisateurs.

Exécutez la commande suivante pour vérifier la connexion (une connexion par socket ne peut pas être utilisée pour tester des 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

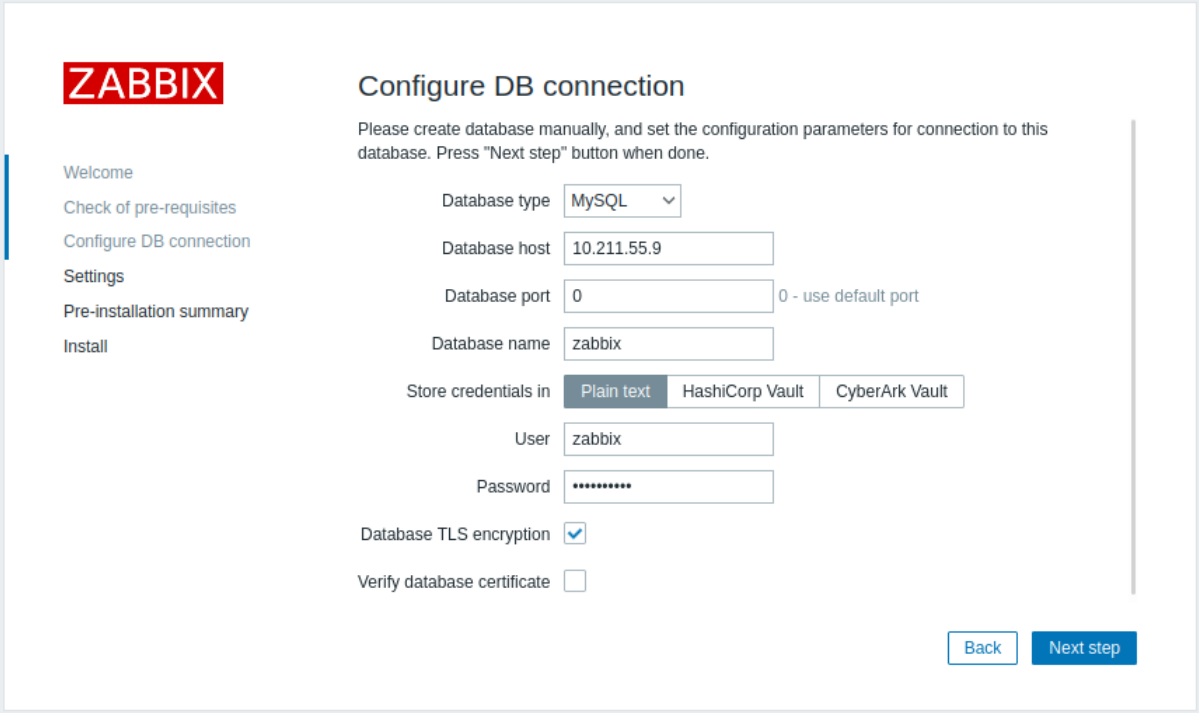

mysql> SHOW SESSION STATUS LIKE 'Ssl_cipher_list'\G;
*************************** 1. row ***************************
Variable_name: Ssl_cipher_list
Value: TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:TLS_AES_128_CCM_SHA256:E
1 row in set (0.00 sec)

ERROR:
No query specified
```

Interface web

Pour activer le chiffrement du transport uniquement pour les connexions entre l'interface web Zabbix et la base de données :

- Cochez *Database TLS encryption*
- Laissez *Verify database certificate* décoché



Serveur

Pour activer le chiffrement du transport uniquement pour les connexions entre le serveur et la base de données, configurez */etc/zabbix/zabbix_server.conf* :

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

Chiffrement avec vérification de l'autorité de certification

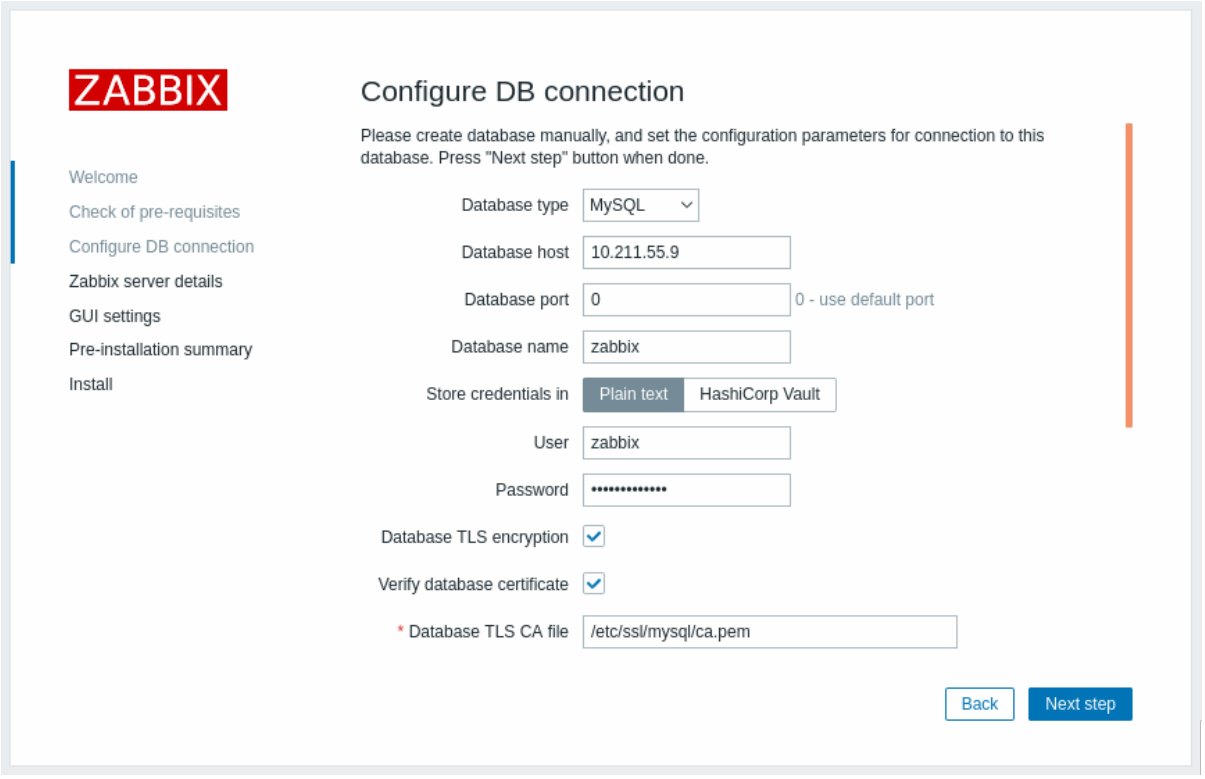
Copiez l'autorité de certification MySQL requise sur le serveur frontend Zabbix, puis attribuez les permissions appropriées pour permettre au serveur web de lire ce fichier.

> **Note:**
> Ce mode ne fonctionne pas sur RHEL 7 en raison de bibliothèques MySQL plus anciennes.

Frontend

Pour activer le chiffrement avec vérification du certificat pour les connexions entre le frontend Zabbix et la base de données :

- Cochez *Database TLS encryption* et *Verify database certificate*
- Indiquez le chemin vers le fichier CA TLS de la base de données



Vous pouvez également définir cela dans */etc/zabbix/web/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 le dépannage, utilisez l'outil en 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, configurez */etc/zabbix/zabbix_server.conf* :

```
...
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 de MySQL

Définissez l'option de configuration du serveur MySQL CE (*/etc/my.cnf.d/server-tls.cnf*) comme suit :

```
[mysqld]
...
##### dans cet exemple, les clés sont situées 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 (frontend 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 le frontend 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 s'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
```

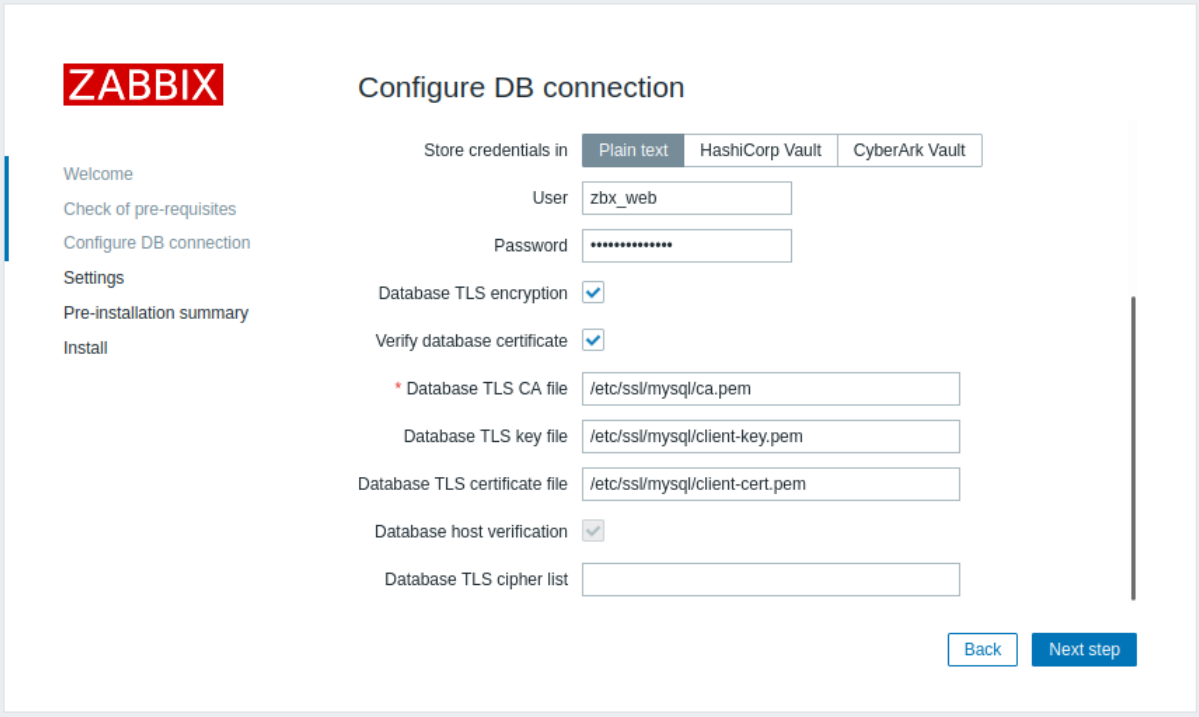
Interface web

Pour activer le chiffrement avec vérification complète pour les connexions entre l'interface web Zabbix et la base de données :

- Cochez *Database TLS encryption* et *Verify database certificate*
- Indiquez le chemin vers *Database TLS key file*
- Indiquez le chemin vers *Database TLS CA file*
- Indiquez le chemin vers *Database TLS certificate file*

Notez que *Database host verification* est coché et grisé : cette étape ne peut pas être ignorée pour MySQL.

Warning:
Si le champ *Database TLS cipher list* est laissé vide, les suites de chiffrement communes autorisées à la fois par l'interface web (client) et le serveur seront activées. Sinon, les suites de chiffrement peuvent être définies explicitement, conformément aux exigences de configuration des suites de chiffrement.



Sinon, cela peut être défini dans */etc/zabbix/web/zabbix.conf.php* :

```
...
// Used for TLS connection with strictly defined Cipher list.
$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
...
// or
...
// Used for TLS connection without Cipher list defined - selected by MySQL server
$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 pour les connexions entre le serveur Zabbix et la base de données, configurez */etc/zabbix/zabbix_server.conf* :

```
...
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
...
```

2 Configuration du chiffrement PostgreSQL

Aperçu

Cette section fournit plusieurs exemples de configuration de 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 *Hôte de la base de données* commence par une barre oblique ou si le champ est vide.

Prérequis

Installez la base de données PostgreSQL en utilisant le dépôt officiel.

Par défaut, PostgreSQL n'est pas configuré 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 ; pour autoriser les connexions réseau distantes, activez l'écoute sur l'interface réseau réelle.

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

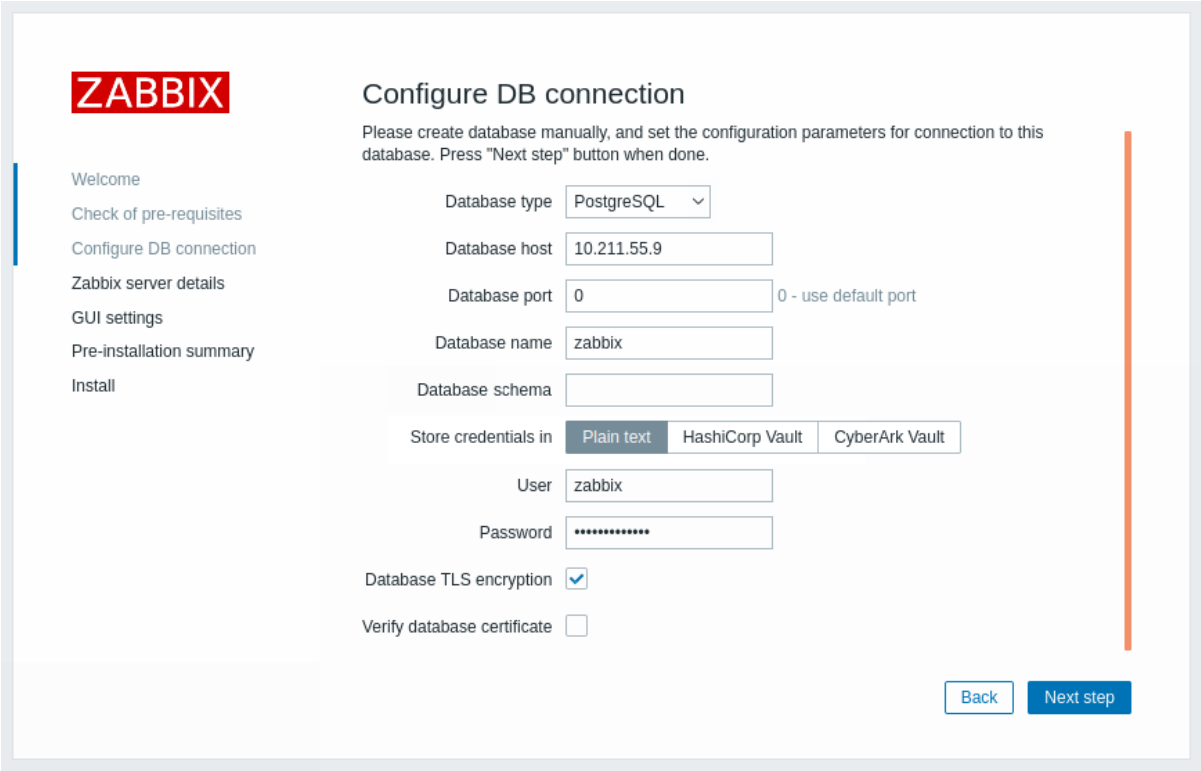
### verify full
hostssl all all 0.0.0.0/0 md5 clientcert=verify-full
...
```

Chiffrement du transport uniquement

Frontend

Pour activer le chiffrement de transport uniquement pour les connexions entre l'interface Zabbix et la base de données :

- Cochez *Cryptage TLS de la base de données*
- Laissez *Vérifier le certificat de la base de données* décoché



Serveur

Pour activer le chiffrement de transport uniquement pour les connexions entre le serveur et la base de données, configurez */etc/zabbix/zabbix_server.conf* :

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

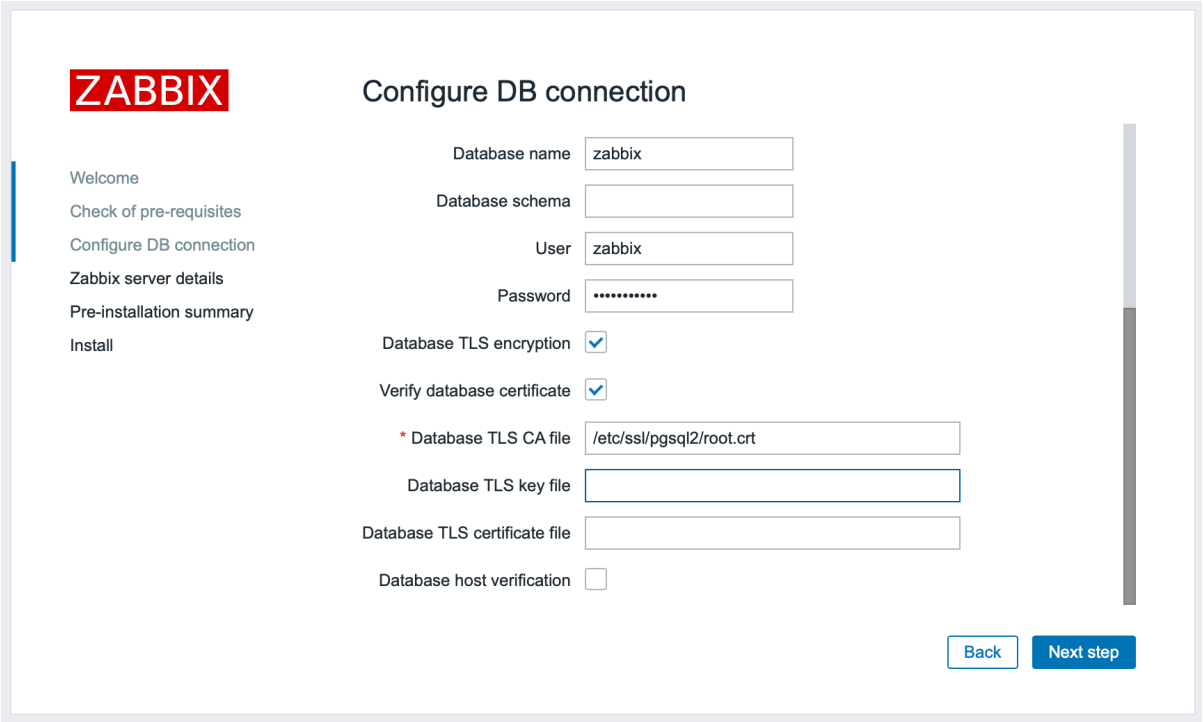
Chiffrement avec vérification par autorité de certification

Interface web

Pour activer le chiffrement avec vérification de l'autorité de certification pour les connexions entre l'interface web Zabbix et la base de données :

- Cochez *Database TLS encryption* et *Verify database certificate*

- Indiquez le chemin vers *Database TLS CA file*



Il est également possible de le définir dans */etc/zabbix/web/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 de certificat pour les connexions entre le serveur Zabbix et la base de données, configurez */etc/zabbix/zabbix_server.conf :*

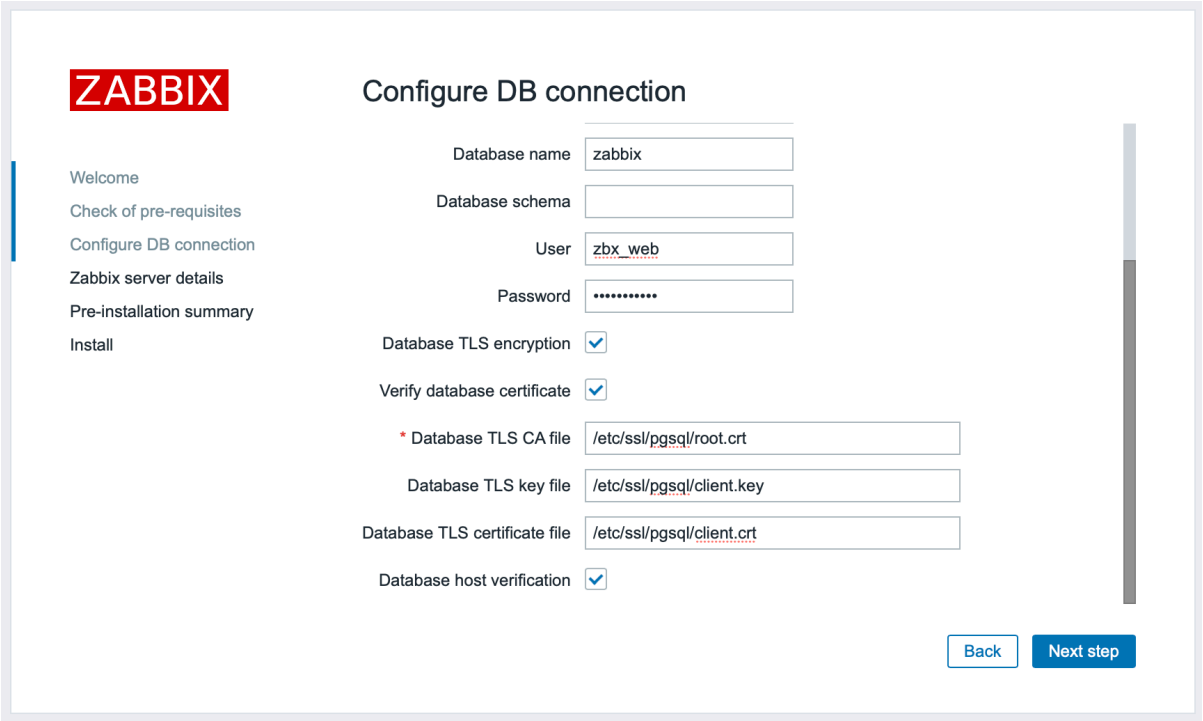
```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_ca
DBTLSCAFile=/etc/ssl/pgsql/root.crt
...
```

Chiffrement avec vérification complète

Frontend

Pour activer le chiffrement avec certificat et 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 *Cryptage TLS de la base de données* et *Vérifiez le certificat de la base de données*
- Spécifiez le chemin d'accès au *fichier de clé TLS de la base de données*
- Spécifiez le chemin d'accès au * fichier CA TLS de la base de données *
- Spécifiez le chemin d'accès au *fichier de certificat TLS de la base de données*
- Cochez * Vérification de l'hôte de la base de données *



Alternativement, cela peut être défini dans */etc/zabbix/web/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 certificat et vérification de l'identité de l'hôte de la base de données pour les connexions entre le serveur Zabbix et la base de données, configurez */etc/zabbix/zabbix_server.conf* :

```
...
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
...
```

6 Connexion sécurisée au frontend

Aperçu

Cette section fournit les étapes de configuration de Zabbix ainsi que des exemples de configuration pour des connexions TLS sécurisées entre le frontend Zabbix et le serveur Zabbix.

Configuration

Par défaut, la communication entre le frontend Zabbix et le serveur Zabbix n'est pas chiffrée. Pour une meilleure sécurité, activez TLS des deux côtés. Vous trouverez ci-dessous 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) (ajustez 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 (ajustez 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 le frontend Zabbix (ajustez 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 permissions appropriées.

Pour le serveur Zabbix (ajustez le propriétaire/groupe selon 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 le frontend (ajustez le propriétaire/groupe selon 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 `zabbix_server.conf`, ajoutez :

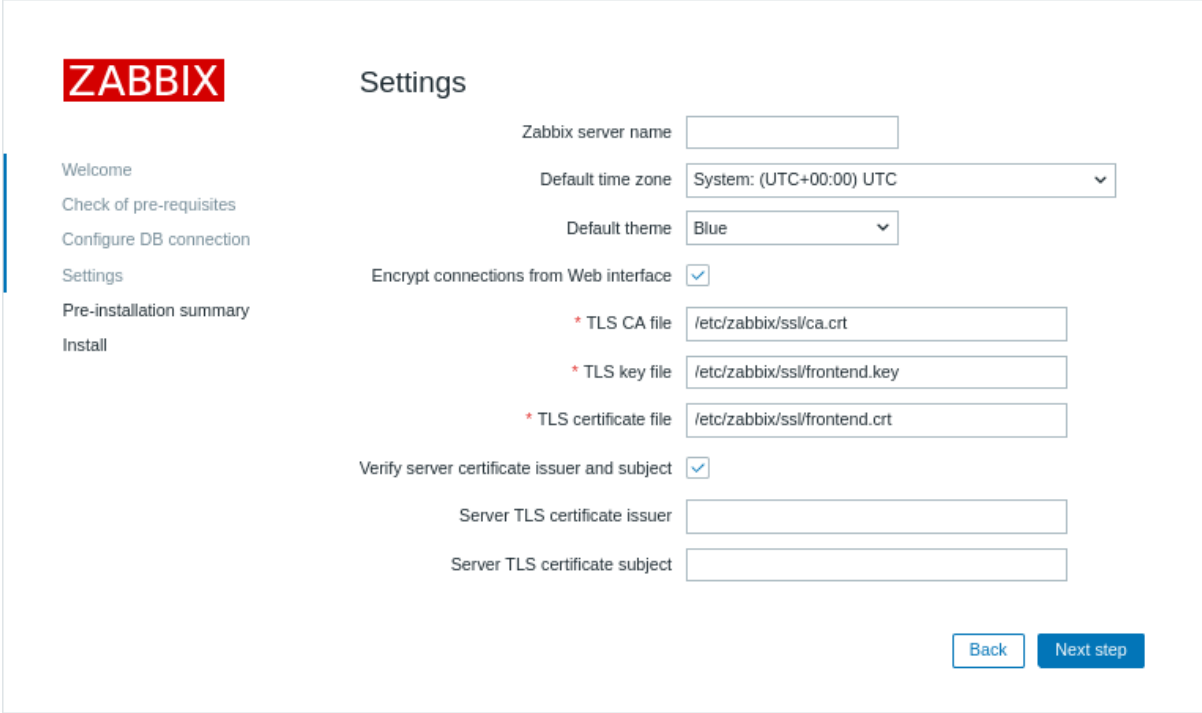
```
TLSFrontendAccept=cert
TLSCertFile=/etc/zabbix/ssl/server.crt
TLSKeyFile=/etc/zabbix/ssl/server.key
TLSCAFile=/etc/zabbix/ssl/ca.crt
#### Facultatif :
#### TLSFrontendCertIssuer=/CN=MyZabbixCA/
#### TLSFrontendCertSubject=/CN=zabbix-frontend-node/
```

Redémarrez ensuite le serveur :

```
sudo systemctl restart zabbix-server
```

4. Configurez le frontend Zabbix.

Lors de l'installation de l'interface web, activez l'option *Encrypt connections from Web interface* (ainsi que l'option *Verify server certificate issuer and subject*, si nécessaire) et renseignez les champs *TLS CA file*, *TLS key file*, *TLS certificate file* (ainsi que *Server TLS certificate issuer* et *Server TLS certificate subject*, si nécessaire) :



Paramètre	Description
TLS CA file	Indiquez le chemin complet vers le fichier de certificat de l'autorité de certification (CA) utilisé pour vérifier le certificat du serveur.
TLS key file	Indiquez le chemin complet vers le fichier de clé privée du client correspondant au certificat client.
TLS certificate file	Indiquez le chemin complet vers le fichier de certificat client si une authentification TLS mutuelle est requise.
Server TLS certificate issuer	Indiquez un nom distinctif (DN) de l'émetteur à faire correspondre avec le certificat du serveur.
Server TLS certificate subject	Indiquez un nom distinctif (DN) du sujet à faire correspondre avec le certificat du serveur.

Sur les installations existantes, modifiez les champs suivants dans `zabbix.conf.php` :

```
$ZBX_SERVER_TLS['ACTIVE'] = '1';
$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 confirmant qu'il n'y a aucun message d'erreur dans le frontend Zabbix ou dans le fichier journal du serveur Zabbix :

```
tail -f /var/log/zabbix/zabbix_server.log
```

7 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, consultez 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 concernée 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 terminée avec succès.

La migration des données existantes d'historique, de tendances et du journal d'audit peut prendre beaucoup de temps. Le serveur Zabbix et le frontend doivent être arrêtés pendant toute la durée de la migration.

Le script `schema.sql` définit les paramètres de nettoyage automatique suivants :

- Remplacer la période d'historique des éléments
- Remplacer la période de tendances des éléments

Pour utiliser le nettoyage automatique 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 datant 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 avec la licence Timescale Community "timescale" et qu'elle n'est pas prise en charge avec 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 web.

Note:
Il est recommandé aux utilisateurs de se familiariser avec la compression dans la documentation TimescaleDB avant d'utiliser la compression.

Notez qu'il existe certaines limitations imposées par la compression, notamment :

- Les modifications des chunks compressés (insertions, suppressions, mises à jour) ne sont pas autorisées
- Les modifications de schéma pour les tables compressées ne sont pas autorisées.

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 web Zabbix.

Paramètre	Par défaut	Commentaires
Activer la compression	Activé	Cocher ou décocher la case n'active/désactive pas immédiatement la compression. 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 précédemment 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.
Compresser les enregistrements antérieurs à	7d	Ce paramètre ne peut pas être inférieur à 7 jours. En raison de l'immuabilité des chunks compressés, toutes les données tardives (par exemple, des données retardées par un proxy) plus anciennes que cette valeur seront ignorées.

Pour de meilleures performances de mise à jour des tendances, vous pouvez réduire le "chunk_time_interval" des tables `trends` et `trends_uint` de 30 jours à 7 jours ou moins, selon le nombre d'éléments utilisant 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.

8 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 d'Elasticsearch 7.X. Si vous utilisez une autre version, certaines fonctionnalités peuvent ne pas fonctionner comme prévu.

La configuration implique la création d'un emplacement de stockage Elasticsearch pour chaque type de valeur, la mise en place du prétraitement (si nécessaire) et la connexion de Zabbix à votre instance Elasticsearch.

Elasticsearch peut stocker les types de valeurs suivants :

Type de valeur de l'élément	Table de base de données	Type Elasticsearch
Numérique (non signé)	history_uint	uint
Numérique (virgule flottante)	history	dbl
Caractère	history_str	str
Journal	history_log	log
Texte	history_text	text
Binaire	history_bin	non pris en charge par Zabbix
JSON	history_json	json

Remarques importantes

- Elasticsearch nécessite libcurl. Voir les prérequis pour plus de détails.
- Le housekeeper ne supprime pas les données d'Elasticsearch.
- Si toutes les données d'historique sont stockées dans Elasticsearch, les tendances ne sont **pas** calculées ni stockées dans la base de données. Envisagez d'étendre la période de conservation de l'historique.
- 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 Zabbix proxy ; veuillez utiliser SQLite à la place.

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 entre 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 leur rétention au fil du temps.

Pour activer cela, 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 l'index basé sur la date approprié.
- Configurer le paramètre `HistoryStorageDateIndex` dans le fichier de configuration du serveur Zabbix — cela active le stockage des valeurs dans plusieurs index basés 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 votre fichier de configuration du serveur Zabbix (`zabbix_server.conf`), définissez les paramètres suivants :

- `HistoryStorageURL` - l'URL de votre instance Elasticsearch.
- `HistoryStorageTypes` - liste des types de valeurs à stocker dans Elasticsearch, séparés par des virgules.

Par exemple, pour stocker dans Elasticsearch les valeurs de type *Character*, *Log*, *Text* et *JSON* (tout en conservant les valeurs *Numeric* dans une base de données) :

```
HistoryStorageURL=http://localhost:9200
HistoryStorageTypes=str,log,text,json
```

Si vous utilisez des index basés sur la date pour **toutes** les valeurs stockées dans Elasticsearch, définissez également le paramètre `HistoryStorageDateIndex` :

```
HistoryStorageDateIndex=1
```

Après avoir effectué les modifications, redémarrez le serveur Zabbix :

```
systemctl restart zabbix-server
```

Configuration de l'interface web Zabbix

Dans votre fichier de configuration de l'interface web Zabbix (`zabbix.conf.php`), déclarez `$HISTORY` comme variable globale

et définissez ses valeurs `url` et `types` afin qu'elles correspondent à la configuration du serveur :

```
// Zabbix GUI configuration file.
global $DB, $HISTORY;

$HISTORY['url']   = 'http://localhost:9200';
$HISTORY['types'] = ['str', 'log', 'text', 'json'];
```

Dépannage

Les étapes suivantes peuvent vous aider à résoudre les problèmes liés à votre configuration Elasticsearch :

1. 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": {} }
```

2. 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.

3. Vérifiez si des shards sont dans un état d'échec ; le redémarrage d'Elasticsearch peut résoudre ce problème.
4. Vérifiez que votre configuration Elasticsearch autorise l'accès depuis Zabbix server et Zabbix frontend.
5. Utilisez le paramètre de configuration du serveur Zabbix `LogSlowQueries` pour identifier les requêtes lentes.
6. Vérifiez les journaux Elasticsearch pour détecter d'éventuelles erreurs.
7. Si vous devez réinitialiser votre configuration Elasticsearch et repartir de zéro, 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/*"
```

9 Notes spécifiques à la distribution sur la configuration de Nginx pour Zabbix

RHEL

Sur RHEL 8, 9 et 10, Nginx est inclus dans AppStream — aucun dépôt supplémentaire n'est nécessaire. Il suffit de l'activer et de l'installer 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) via :

```
yum -y install nginx
```

Si vous préférez les toutes dernières versions upstream, 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
```

10 Exécution de l'agent en tant que root

Depuis Zabbix **5.0.0**, le fichier de service systemd de l'agent Zabbix dans les paquets officiels inclut explicitement les directives `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 effectuer 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 démons 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 `zabbix_agentd.conf`. Vous devez maintenant définir les paramètres de configuration `User=root` et `AllowRoot=1` dans le fichier de configuration de l'agent.

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.

11 Agent Zabbix 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 soit :

- le fichier de configuration par défaut, situé dans le même répertoire que le binaire de l'agent ;
- un fichier de configuration spécifié dans la ligne de commande.

Dans le cas de plusieurs instances, chaque instance d'agent doit avoir 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 :

- `conf/zabbix_agentd.conf` pour l'agent Zabbix ;
- `conf/zabbix_agent2.conf` pour Zabbix agent2.

Si vous souhaitez installer l'agent Zabbix/l'agent 2 pour Windows en tant que service à partir d'une archive sans spécifier explicitement le fichier de configuration, alors, 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 pour plus de détails sur la configuration de l'agent Zabbix Windows.

Paramètre Hostname

Pour effectuer des contrôles actifs sur un hôte, Zabbix agent 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 de l'hôte" configuré pour l'hôte dans le frontend.

La valeur du nom d'hôte du côté de l'agent peut être définie soit par le paramètre **Hostname**, soit par le paramètre **HostnameItem** dans le fichier de configuration de l'agent — ou bien 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, elle renvoie le résultat de la fonction gethostname(), qui interroge les fournisseurs d'espace de noms afin de déterminer le nom de l'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, en pratique, si ces deux paramètres ne sont pas spécifiés, le nom d'hôte réel sera le nom NetBIOS de l'hôte ; Zabbix agent utilisera le nom d'hôte NetBIOS pour récupérer la liste des contrôles actifs depuis Zabbix server et lui envoyer les résultats.

La clé "system.hostname" prend en charge deux paramètres optionnels : *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 uniquement en 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. Il renverra la chaîne complète si le nom ne contient pas de point.
- *fqdn* - renvoie le nom de domaine pleinement qualifié (sans le point final).

Transform permet de spécifier une règle de transformation supplémentaire pour le nom d'hôte :

- *none* (par défaut) - utilise la casse d'origine ;
- *lower* - convertit le texte en minuscules.

Ainsi, pour simplifier la configuration du fichier zabbix_agentd.conf et la rendre uniforme, trois approches différentes peuvent être utilisées :

1. Laisser les paramètres **Hostname** ou **HostnameItem** non définis, et Zabbix agent utilisera 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 Zabbix agent utilise le nom d'hôte Windows complet et réel (sensible à la casse) comme nom d'hôte
   HostnameItem=system.hostname[shorthost,lower] - pour que Zabbix agent utilise uniquement la partie du nom d'hôte avant le premier point, convertie en minuscules.
   HostnameItem=system.hostname[fqdn] - pour que Zabbix agent utilise le nom de domaine pleinement qualifié 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 Zabbix agent spécifie `Hostname=Windows_db_server`, alors 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 Zabbix agent, 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 seule instance 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
```

Le 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 à partir de la ligne de commande.

Pour démarrer une seule instance de l'agent Zabbix avec le fichier de configuration par défaut :

```
zabbix_agentd.exe --start
```

Pour démarrer une seule instance de l'agent Zabbix avec un autre fichier de configuration :

```
zabbix_agentd.exe --config <your_configuration_file> --start
```

Pour démarrer l'une des multiples 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 à partir de ligne de commande.

Pour arrêter une seule instance de l'agent Zabbix démarrée avec la valeur par défaut du fichier de configuration:

```
zabbix_agentd.exe --stop
```

Pour arrêter une seule instance de l'agent Zabbix démarrée avec un autre fichier de configuration:

```
zabbix_agentd.exe --config <votre_fichier_de_configuration> --stop
```

Pour arrêter l'une des multiples instances de l'agent Zabbix :

```
zabbix_agentd.exe --config <fichier_de_configuration_pour_cette_instance> --stop --multiple-agents
```

Désinstallation du service Windows de l'agent

Pour désinstaller une seule instance de l'agent Zabbix en utilisant la valeur par défaut du fichier de configuration:

```
zabbix_agentd.exe --uninstall
```

Pour désinstaller une seule instance de l'agent Zabbix à l'aide d'un autre fichier de configuration que celui par défaut :

```
zabbix_agentd.exe --config <votre_fichier_de_configuration> --uninstall
```

Pour désinstaller plusieurs instances de l'agent Zabbix des services Windows :

```
zabbix_agentd.exe --config <configuration_file_for_instance_1> --uninstall --multiple-agents
zabbix_agentd.exe --config <fichier_de_configuration_pour_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 processeurs sont répartis de manière non uniforme sur les nœuds NUMA. Si les processeurs logiques ne sont pas distribués de manière uniforme, les métriques de performances du processeur peuvent ne pas être disponibles pour certains processeurs. Par exemple, s'il y a 72 CPU logiques avec 2 nœuds NUMA, les deux nœuds doivent avoir 36 CPU chacun.

12 Configuration de 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.

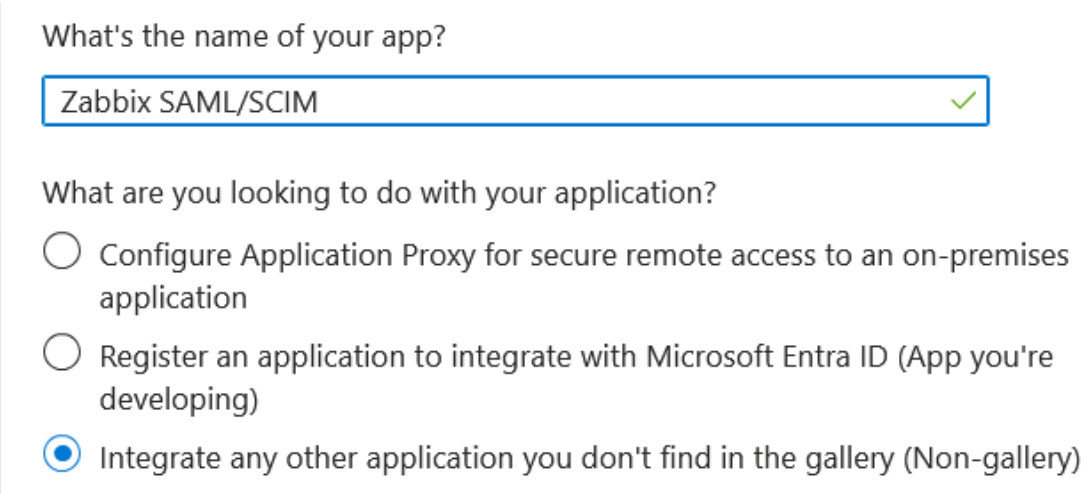
Configuration de Microsoft Entra ID

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 -> Créez votre propre application*.

3. Ajoutez le nom de votre application et sélectionnez l'option *Intégrer toute autre application...*. Ensuite, cliquez sur *Créer*.

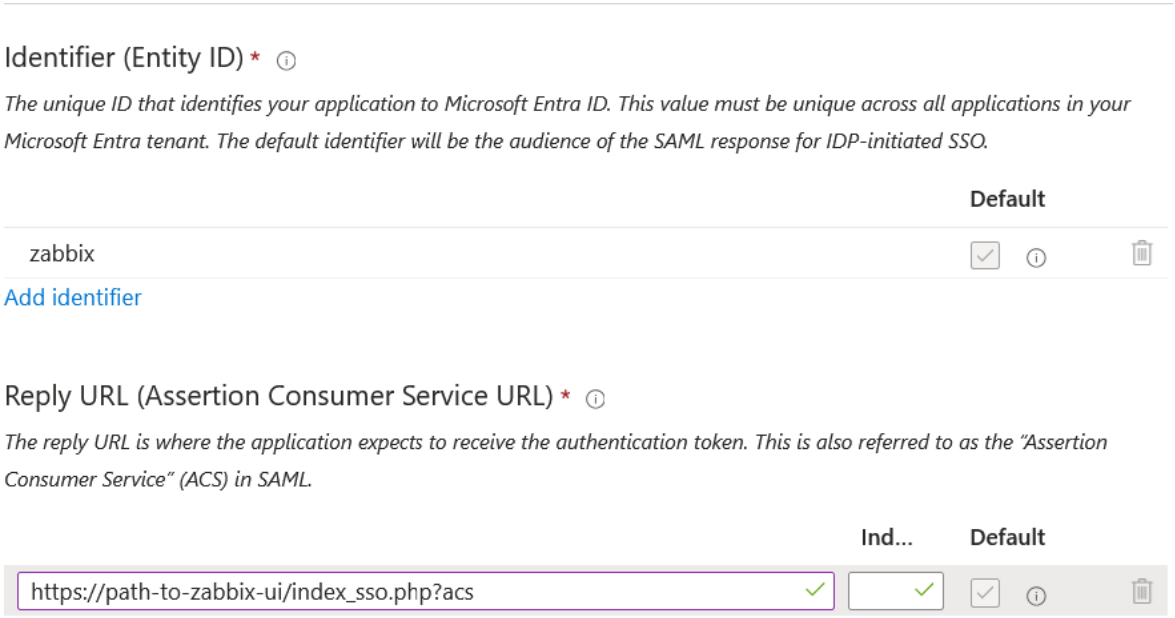


Configuration de l'authentification unique

1. Sur 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 pour 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 Zabbix : `https://<path-to-zabbix-ui>/index_sso.php?acs` :



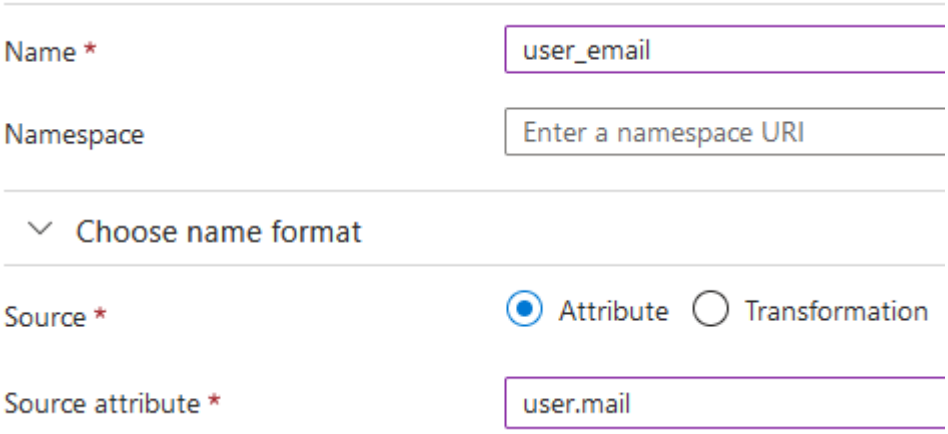
Notez que "https" est requis. Pour que cela fonctionne avec Zabbix, il est nécessaire d'ajouter la ligne suivante à `conf/zabbix.conf.php` :

```
$SSO['SETTINGS'] = ['use_proxy_headers' => true];
```

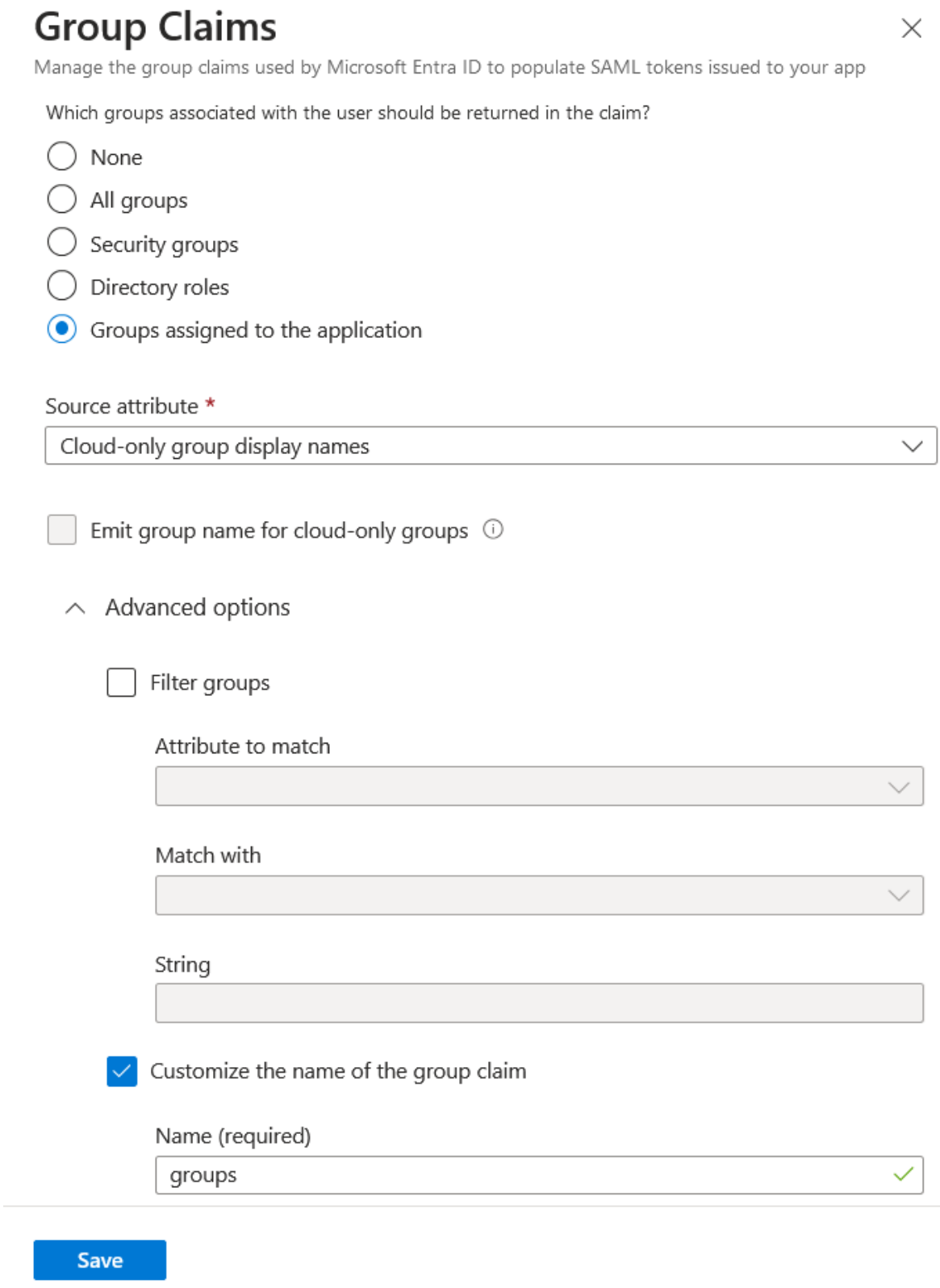
3. Modifiez *Attributes & Claims*. Vous devez ajouter tous les attributs que vous souhaitez transmettre à Zabbix (user_name, user_lastname, user_email, user_mobile, groups).

Les noms des attributs sont arbitraires. Des noms d'attributs différents peuvent être utilisés ; toutefois, il est nécessaire qu'ils correspondent 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 des groupes à Zabbix :



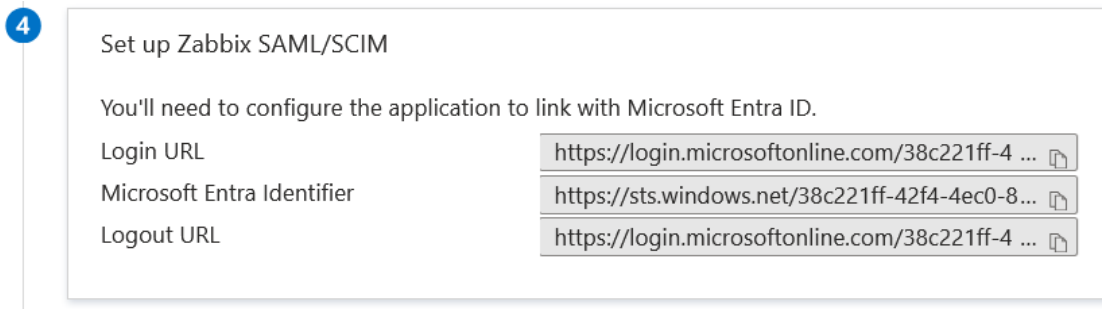
Dans cette revendication, il est important que les noms des groupes (et non les ID des groupes) soient transmis à Zabbix par l'attribut *Source attribute* sélectionné. Sinon, l'approvisionnement JIT des utilisateurs ne fonctionnera pas correctement.

4. Dans *SAML Certificates*, téléchargez le certificat Base64 fourni par Entra ID et placez-le dans `conf/certs` de l'installation du frontend Zabbix.

Définissez les permissions 644 sur celui-ci en exécutant :

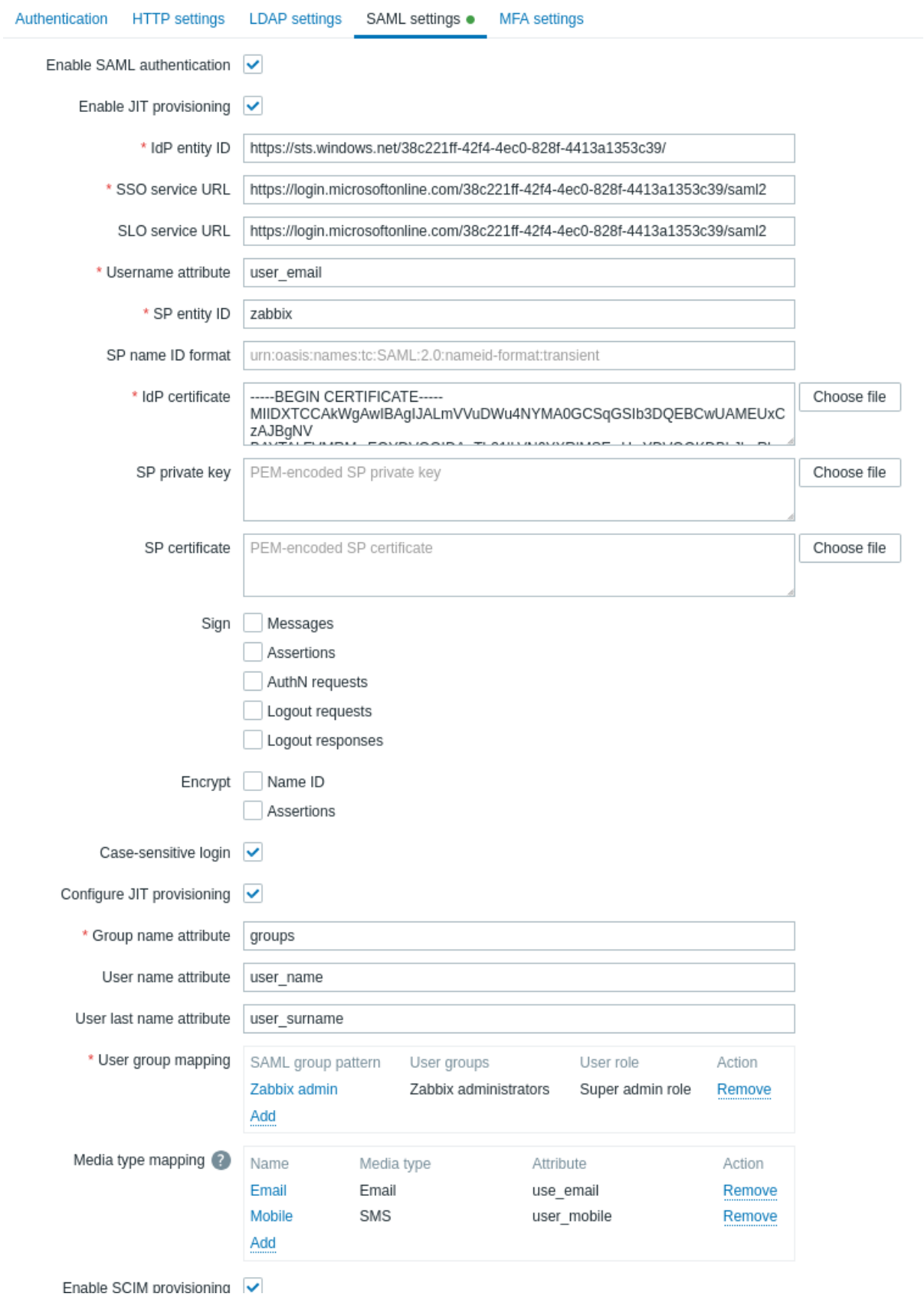
```
chmod 644 entra.cer
```

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 Entra ID :



Champ Zabbix	Champ de configuration dans Entra ID	Valeur d'exemple
ID d'entité IdP	Identifiant Microsoft Entra	
URL du service SSO	URL de connexion	
URL du service SLO	URL de déconnexion	
ID d'entité SP	Identifiant (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	Valeur d'exemple
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 zabbix.conf.php, vous pouvez coller le texte du certificat ou téléverser le fichier du certificat dans l'interface web lors de la configuration SAML — aucun fichier n'est requis sur le système de fichiers.

Si `$SSO['CERT_STORAGE'] = 'file'` est défini dans zabbix.conf.php, le certificat doit être disponible sur le système de fichiers (par défaut dans ui/conf/certs ou dans le chemin configuré dans zabbix.conf.php) et l'interface web ne stockera pas les certificats dans la base de données. Notez que si `$SSO['CERT_STORAGE']` n'est pas défini ou est commenté, le stockage dans des fichiers est supposé et les certificats sont lus depuis ui/conf/certs.

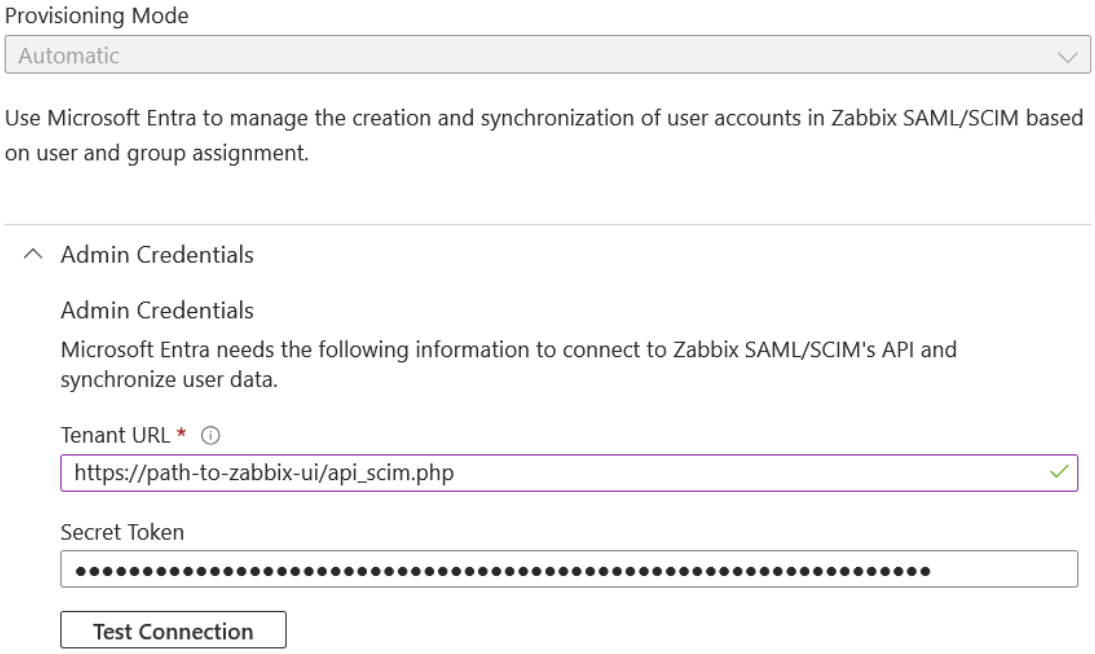
Il est également nécessaire de configurer le mappage des groupes d'utilisateurs. Le mappage des médias est facultatif.

3. Cliquez sur le bouton *Update* pour enregistrer ces paramètres.

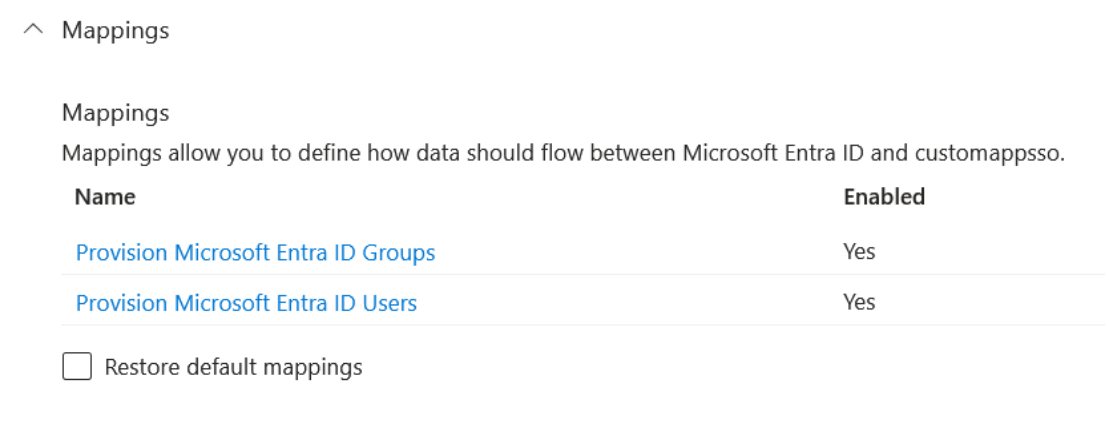
Provisionnement des utilisateurs SCIM

1. Sur 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 d'API Zabbix avec des permissions de Super admin.
- Cliquez sur *Test connection* pour vérifier si la connexion est établie.

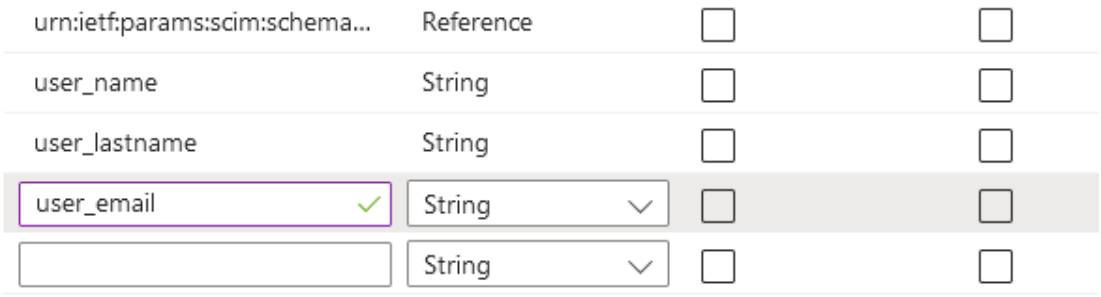


2. Vous pouvez maintenant ajouter tous les attributs qui seront transmis avec SCIM à Zabbix. Pour cela, cliquez sur *Mappings*, puis sur *Provision Microsoft Entra ID Users*.



En bas de la liste *Attribute Mapping*, activez *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 :



Lorsque tous les mappages ont été ajoutés, enregistrez la liste des mappages.



4. Comme prérequis au provisionnement des utilisateurs dans Zabbix, vous devez avoir des utilisateurs et des groupes configurés dans Entra ID.

Pour cela, accédez à *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 accéder au menu *Users and groups* de votre application et les ajouter à l'application.

6. Accédez au menu *Provisioning* de votre application, puis cliquez sur *Start provisioning* pour provisionner les utilisateurs dans Zabbix.

Notez que la requête PATCH des utilisateurs dans Entra ID ne prend pas en charge les modifications des médias.

Signature des requêtes d'authentification

Il est possible de configurer Entra ID pour valider la signature des requêtes d'authentification signées.

Pour que cela fonctionne, créez des clés publique/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 du frontend 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, essayant de se connecter à Zabbix via SAML, est déjà connecté avec le profil MicroSoft Edge. Un signe de ce problème est que l'utilisateur peut être en mesure de se connecter à Zabbix en utilisant 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 du frontend Zabbix (*zabbix.conf.php*).

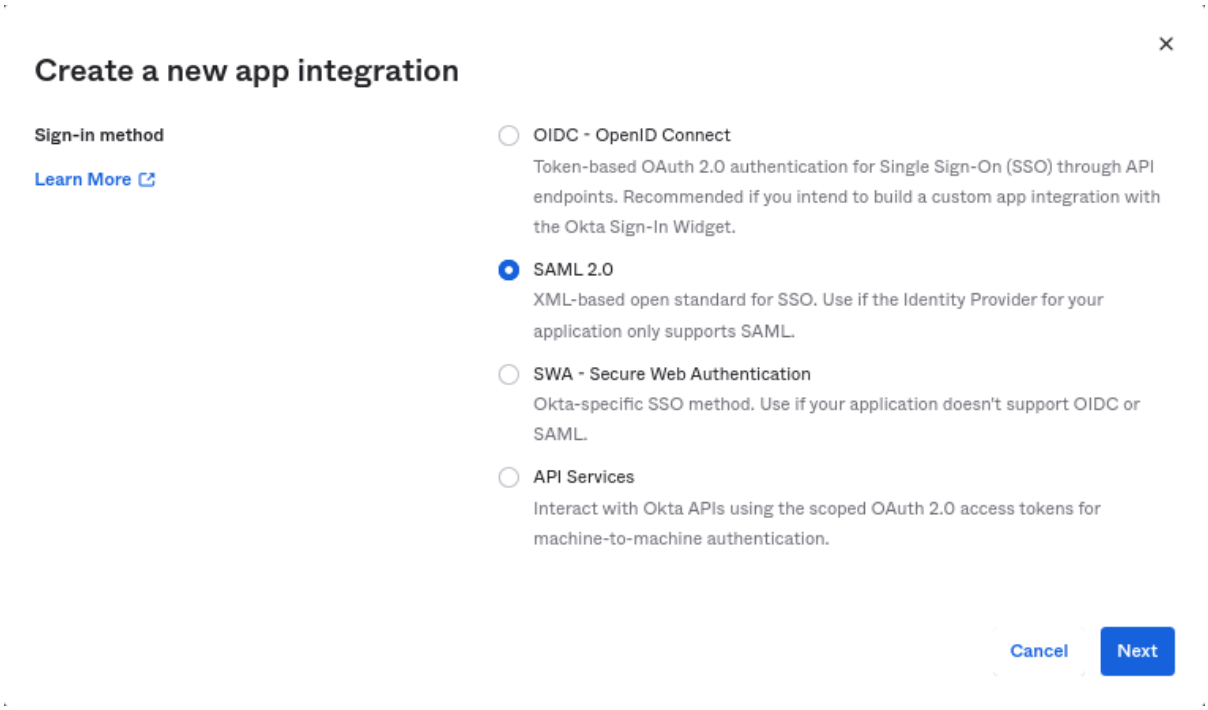
```
$SSO['SETTINGS'] = [
    'security' => [
        'requestedAuthnContext' => false
    ]
];
```

13 Configuration de 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.

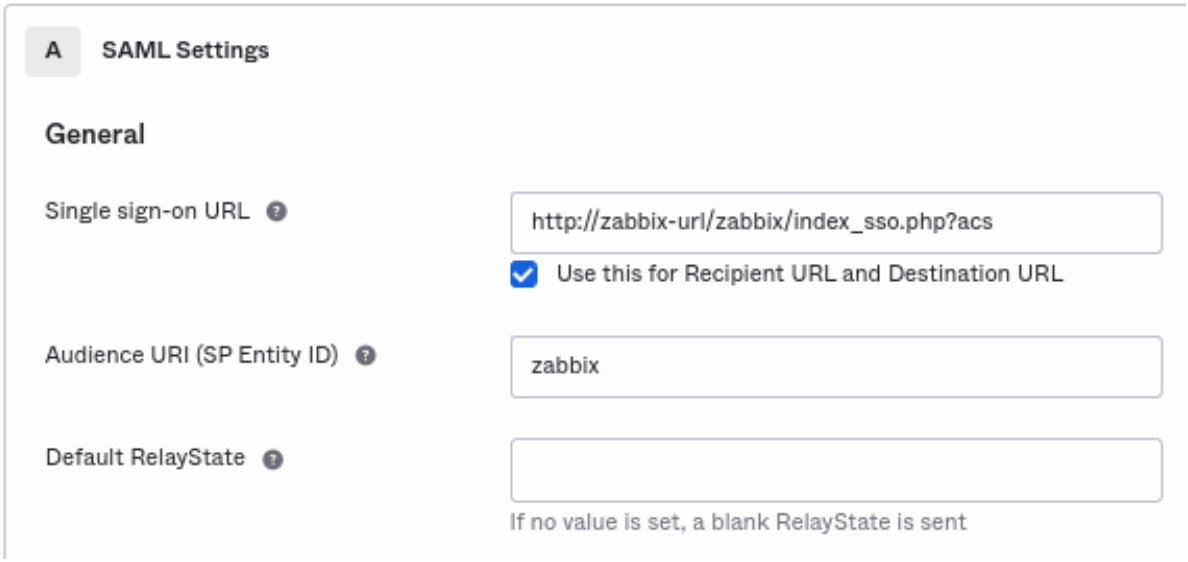
Configuration d'Okta

1. Accédez à https://developer.okta.com/signup/ et inscrivez-vous/connectez-vous à votre compte.
2. Dans l'interface web d'Okta, accédez à *Applications* → *Applications*.
3. Cliquez sur *Create App Integration*.

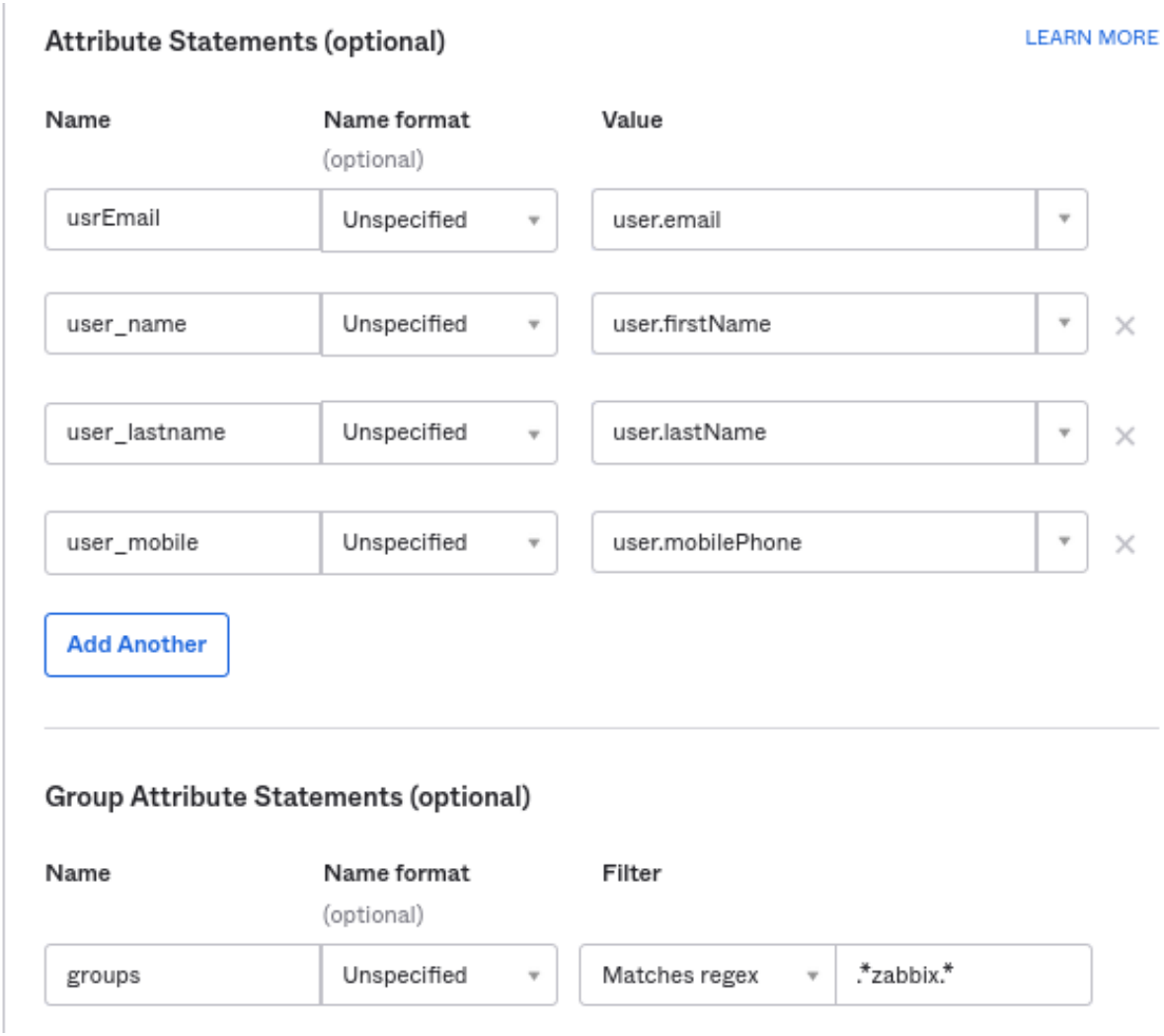


Sélectionnez "SAML 2.0" comme méthode de connexion, puis cliquez sur *Next*.

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`
    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`
    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 chaîne de données dans ce champ.
  - *Default RelayState* :
    Laissez ce champ vide ; si une redirection personnalisée est requise, 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.

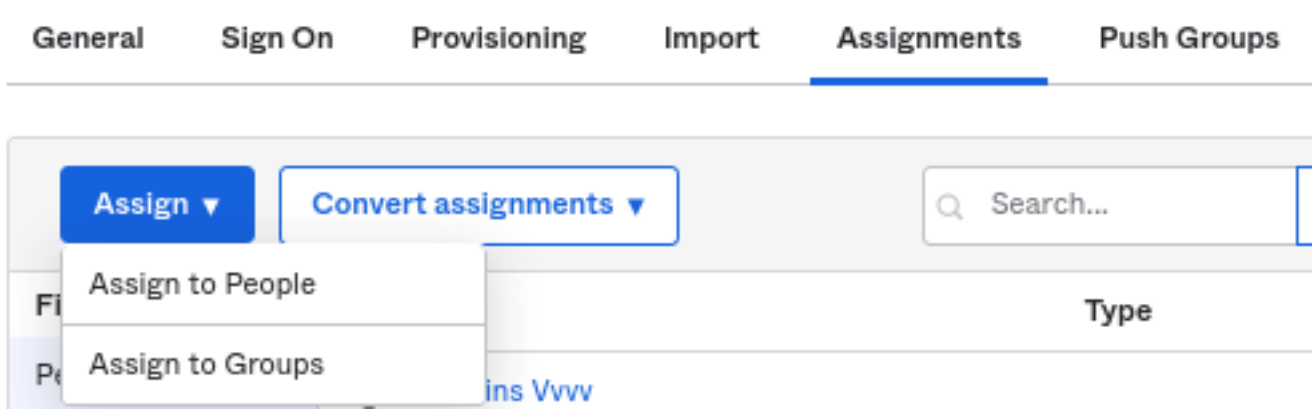
Les noms d'attribut utilisés ici sont des exemples arbitraires. Vous pouvez utiliser des noms d'attribut différents ; toutefois, il est nécessaire qu'ils correspondent à la valeur du champ respectif dans les paramètres SAML de Zabbix.

Si vous souhaitez configurer la connexion SAML à Zabbix *sans* provisionnement d'utilisateurs JIT, seul l'attribut d'adresse 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 dans 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 s'affiche, assignez l'application aux personnes qui utiliseront SAML 2.0 pour s'authentifier dans Zabbix, puis cliquez sur *Save and go back*.

9. Téléchargez le certificat IdP. Définissez les permissions 644 en exécutant :

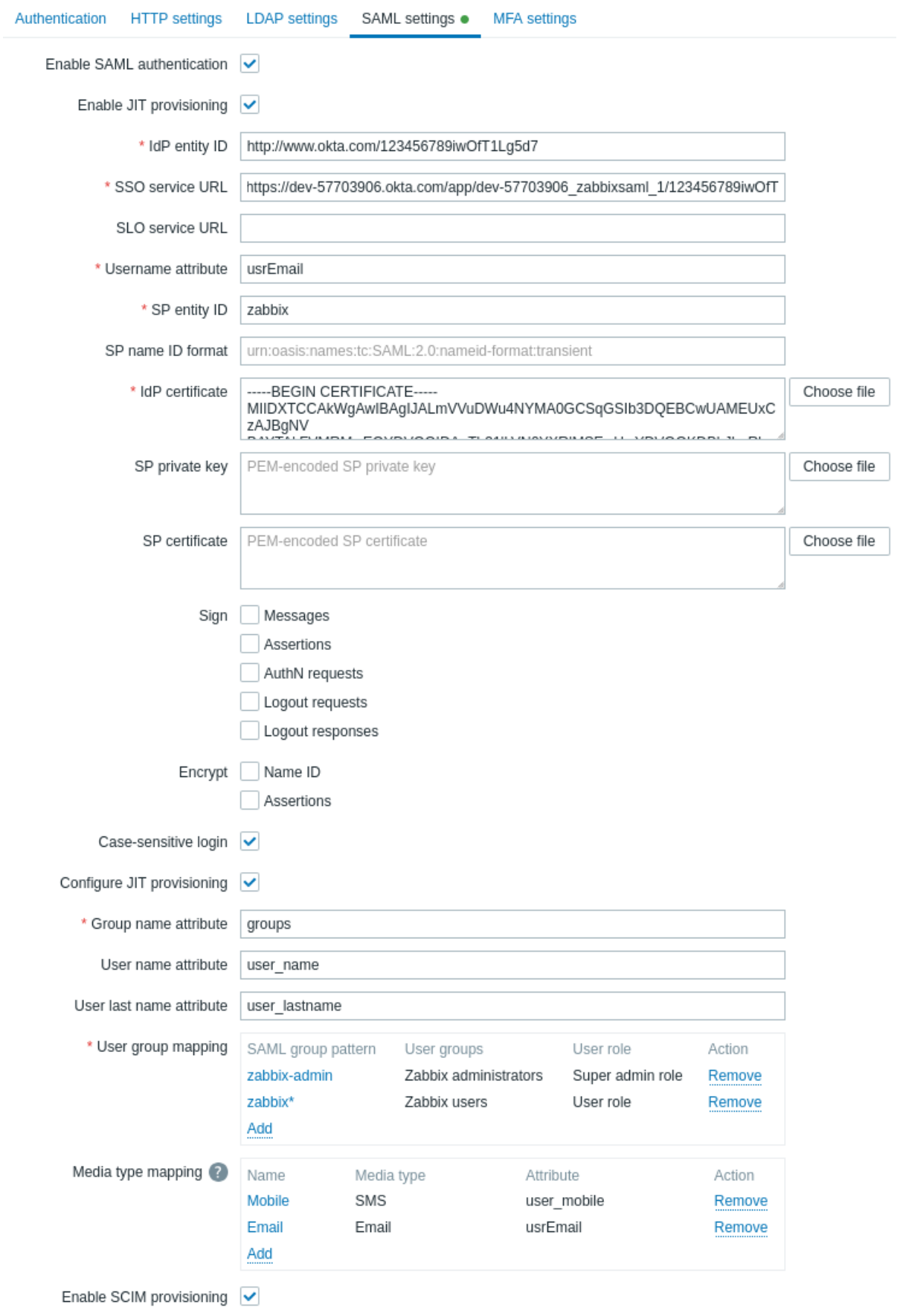
```
chmod 644 idp.crt
```

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 vous basant sur les instructions de configuration d'Okta :



Champ Zabbix	Champ de configuration dans Okta	Valeur d'exemple
ID d'entité IdP	Émetteur du fournisseur d'identité	
URL du service SSO	URL de connexion unique du fournisseur d'identité	
Attribut du nom d'utilisateur	Nom de l'attribut	`usrEmail`
ID d'entité SP	URI d'audience	`zabbix`

Champ Zabbix	Champ de configuration dans Okta	Valeur d'exemple
Attribut du nom du groupe	Nom de l'attribut	`groups`
Attribut du prénom de l'utilisateur	Nom de l'attribut	`user_name`
Attribut du nom de famille de l'utilisateur	Nom de l'attribut	`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 zabbix.conf.php, vous pouvez coller le texte du certificat ou téléverser le fichier du certificat dans l'interface web pendant la configuration SAML — aucun fichier n'est requis sur le système de fichiers.

Si `$SSO['CERT_STORAGE'] = 'file'` est défini dans zabbix.conf.php, le certificat doit être disponible sur le système de fichiers (par défaut dans ui/conf/certs ou dans le chemin configuré dans zabbix.conf.php) et l'interface web ne stockera pas les certificats dans la base de données.
Notez que si `$SSO['CERT_STORAGE']` n'est pas défini ou est commenté, le stockage dans des fichiers est supposé et les certificats sont lus depuis ui/conf/certs.

3. Si *Assertion Encryption* a été 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.

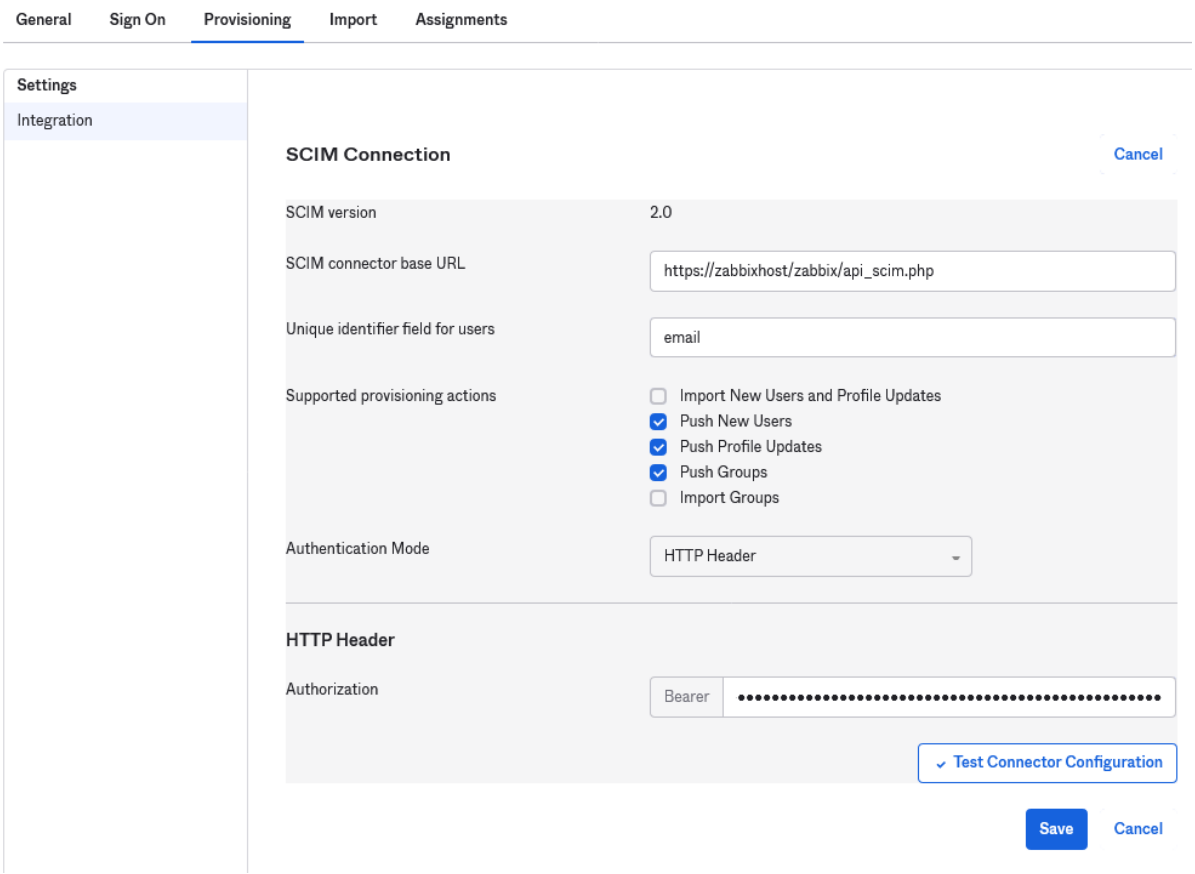
Provisionnement SCIM

1. Pour activer le provisionnement SCIM, accédez à « General » -> « App Settings » de l'application dans Okta.

Cochez la case *Enable SCIM provisioning*. Par conséquent, un nouvel onglet *Provisioning* apparaît.

2. Accédez à l'onglet « Provisioning » pour configurer une connexion SCIM :

- Dans *SCIM connector base URL*, indiquez le chemin vers le frontend Zabbix et ajoutez-y `api_scim.php`, par exemple : `https://<your-zabbix-url>/zabbix/api_scim.php`
- *Unique identifier field for users* : `email`
- *Authentication mode* : `HTTP header`
- Dans *Authorization*, saisissez un jeton d'API valide avec les droits Super admin



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'affichera.

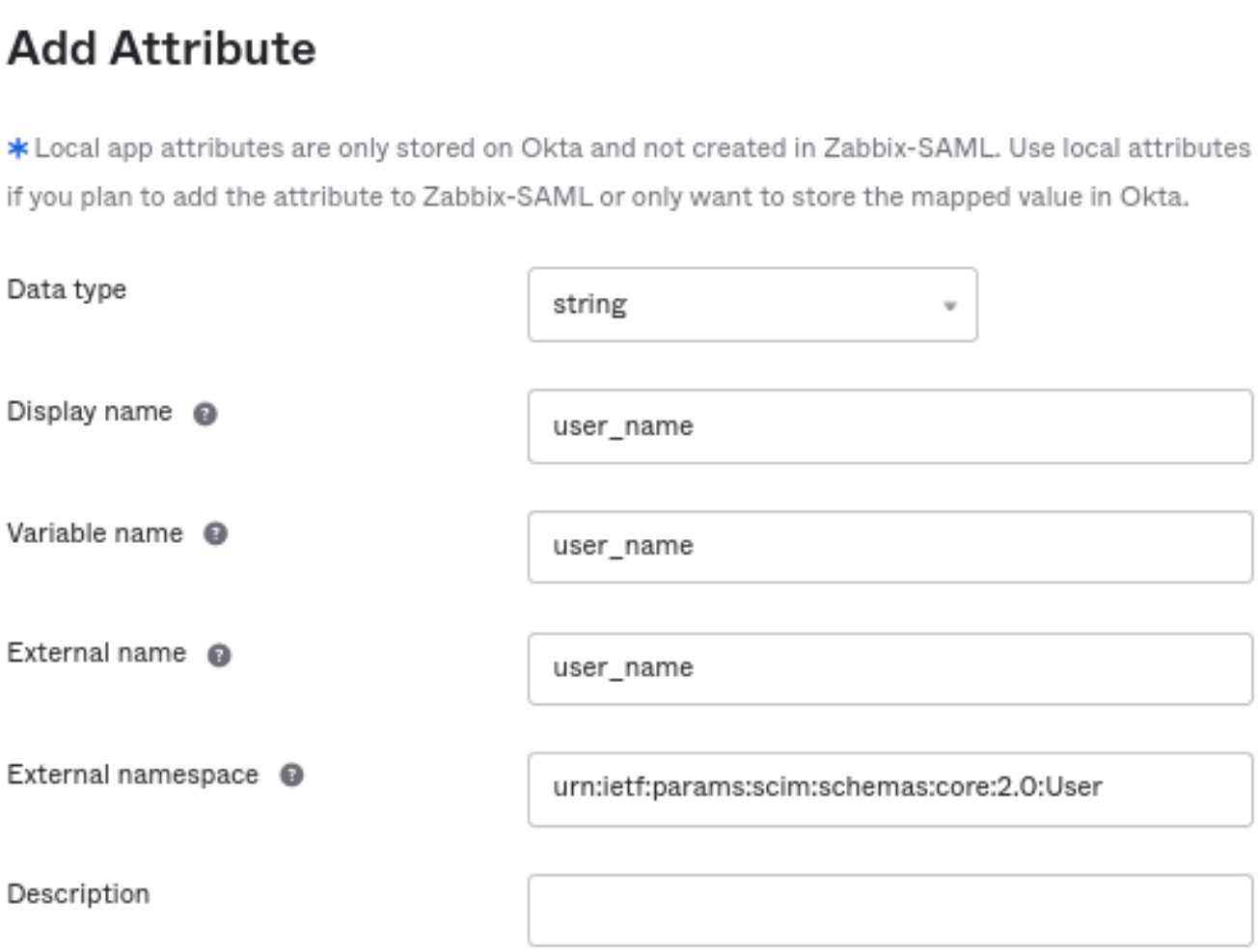
4. Dans « Provisioning » -> « To App », assurez-vous de cocher les cases suivantes :

- Create Users
- Update User Attributes
- Deactivate Users

Cela garantit que ces types de requêtes seront envoyés à Zabbix.

5. Assurez-vous 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*.

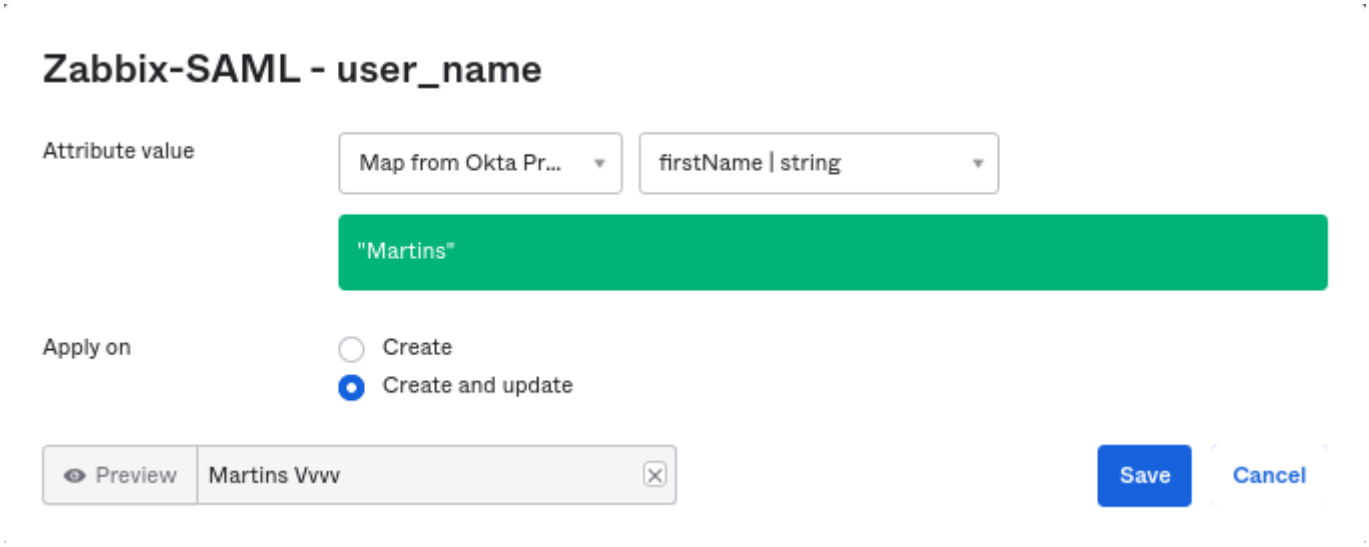
Cliquez sur *Add Attribute*. Renseignez les valeurs de *Display name*, *Variable name* et *External name* avec le nom de l'attribut SAML, par exemple `user_name`.



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 d'abord être 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. S'il n'y a aucune correspondance, l'utilisateur ne peut pas être créé dans Zabbix.

Les informations sur les membres des groupes sont envoyées à chaque modification.

14 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 OneLogin.
À des fins de test, vous pouvez créer un compte développeur gratuit dans OneLogin.

2. Dans l'interface web de 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 de l'application *SCIM Provisioner with SAML (SCIM v2 Enterprise, full SAML)*.

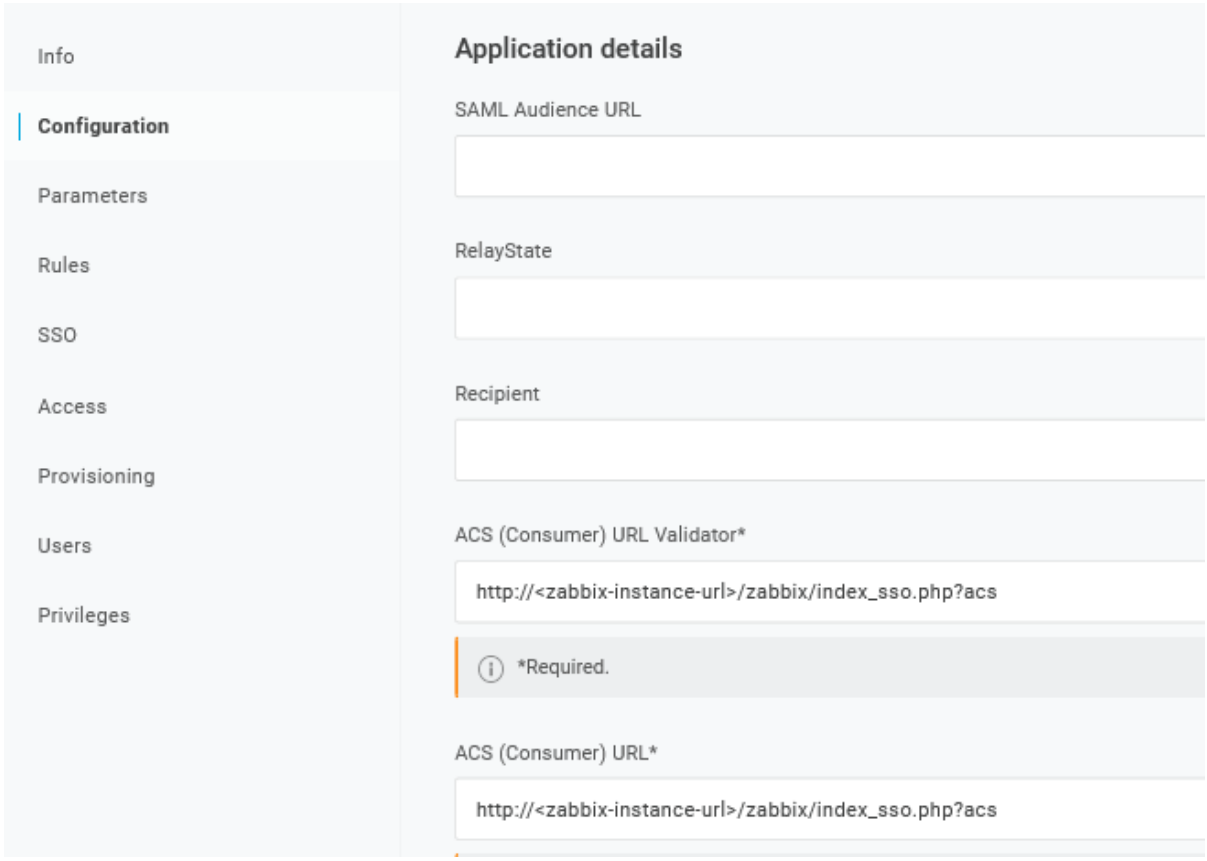
4. Pour commencer, vous pouvez personnaliser le nom d'affichage de votre application.
Vous pouvez également ajouter l'icône et les détails de l'application.
Ensuite, cliquez sur *Save*.

Configuration de l'approvisionnement 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 que cela fonctionne avec Zabbix, il est nécessaire d'ajouter la ligne suivante à `conf/zabbix.conf.php` :

```
$SSO['SETTINGS'] = ['use_proxy_headers' => true];
```

Laissez les autres options à 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` :

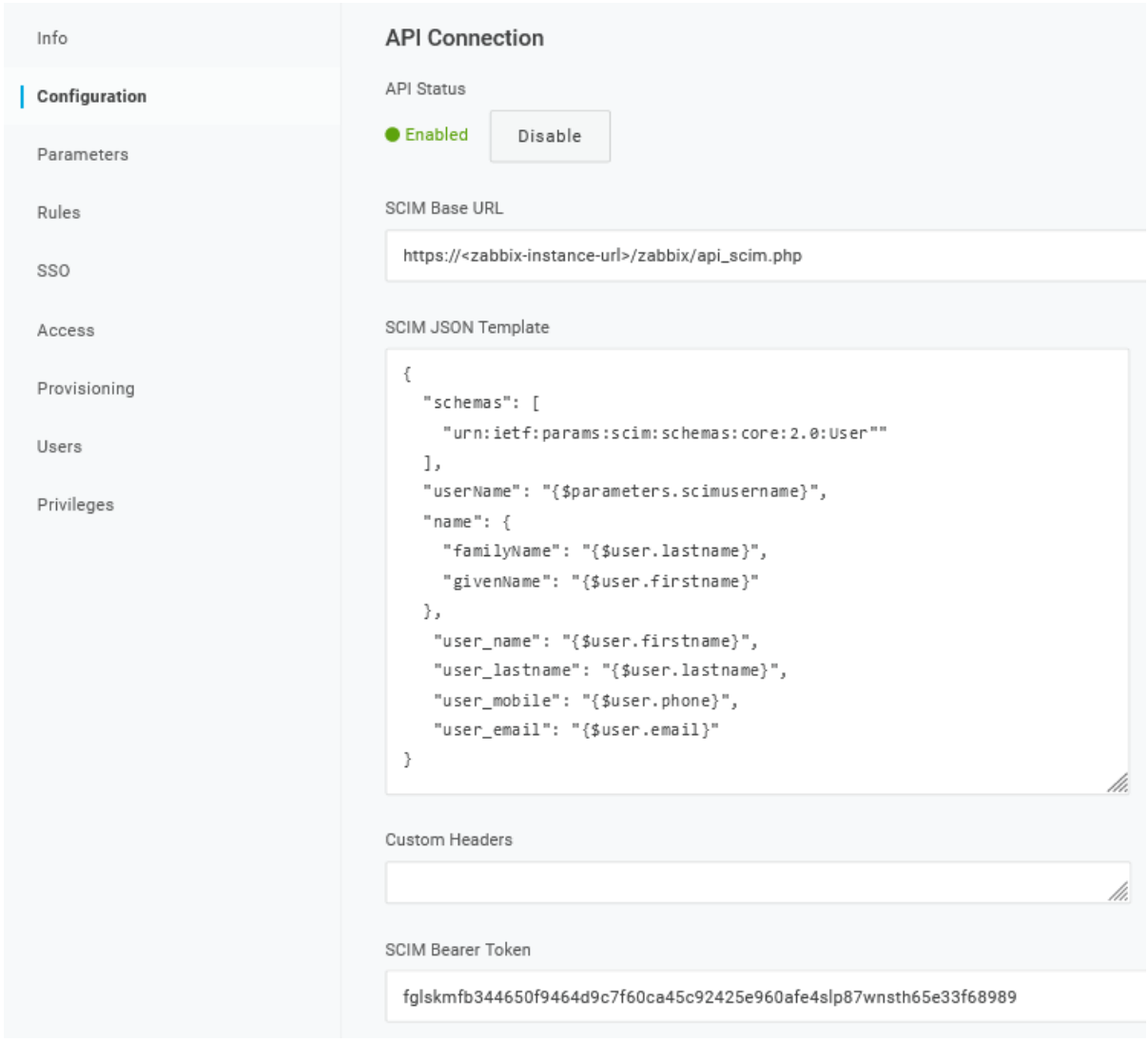
```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User"
  ],
  "userName": "{$parameters.scimusername}",
  "name": {
    "familyName": "{$user.lastname}",
    "givenName": "{$user.firstname}"
  },
   "user_name": "{$user.firstname}",
   "user_lastname": "{$user.lastname}",
   "user_mobile": "{$user.phone}",
   "user_email": "{$user.email}"
}
```

Les noms des attributs sont arbitraires. Des noms d'attributs différents peuvent être utilisés ; toutefois, il est nécessaire qu'ils correspondent à la valeur du champ respectif dans les paramètres SAML de Zabbix.

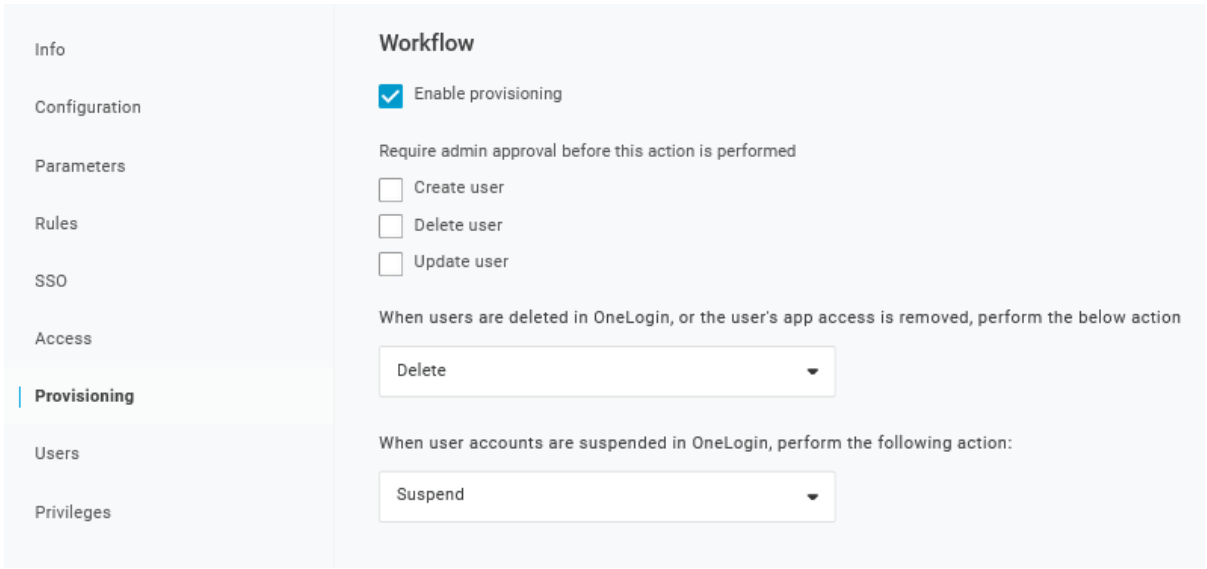
Notez que, pour que l'approvisionnement des utilisateurs fonctionne, OneLogin doit recevoir en réponse un attribut « name » avec « givenName » et « familyName », même si cela n'était pas requis par le fournisseur de services. Il est donc nécessaire de le spécifier dans le schéma, dans la partie configuration de l'application.

- *SCIM Bearer Token* : saisissez un jeton d'API Zabbix avec des autorisations de Super admin.

Cliquez sur *Enable* pour activer la connexion.

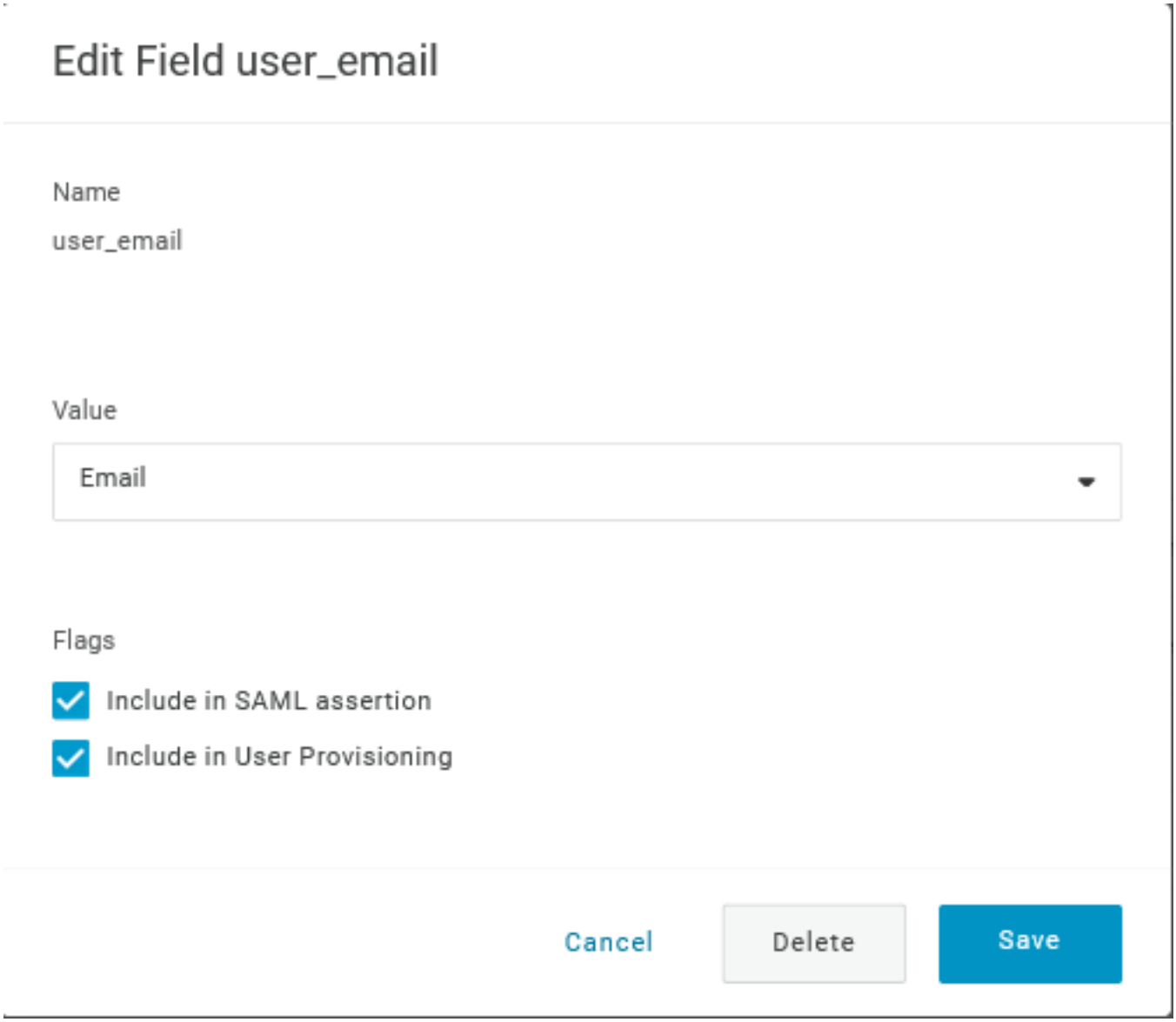


3. Dans la page *Provisioning*, activez l'option Provisioning :



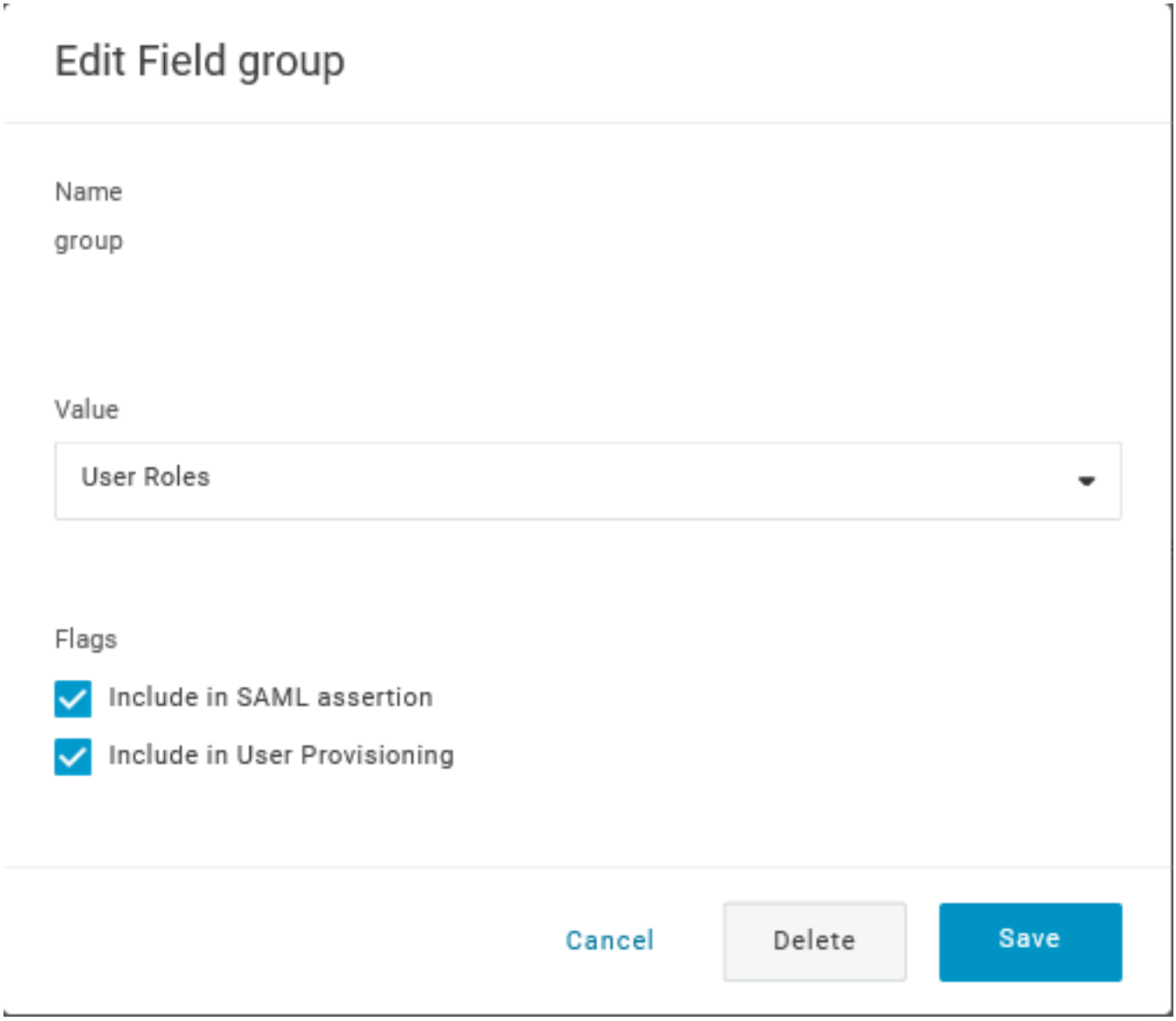
4. La page *Parameters* contient une liste de paramètres par défaut :

- Assurez-vous que « scimusername » correspond à la valeur de connexion de l'utilisateur dans OneLogin (par exemple, l'e-mail) ;
- Cochez l'option *Include in User Provisioning* pour le paramètre « Groups » ;
- Cliquez sur « + » pour créer les paramètres personnalisés requis pour les assertions SAML et l'approvisionnement des utilisateurs, tels que `user_name`, `user_lastname`, `user_email` et `user_mobile` :

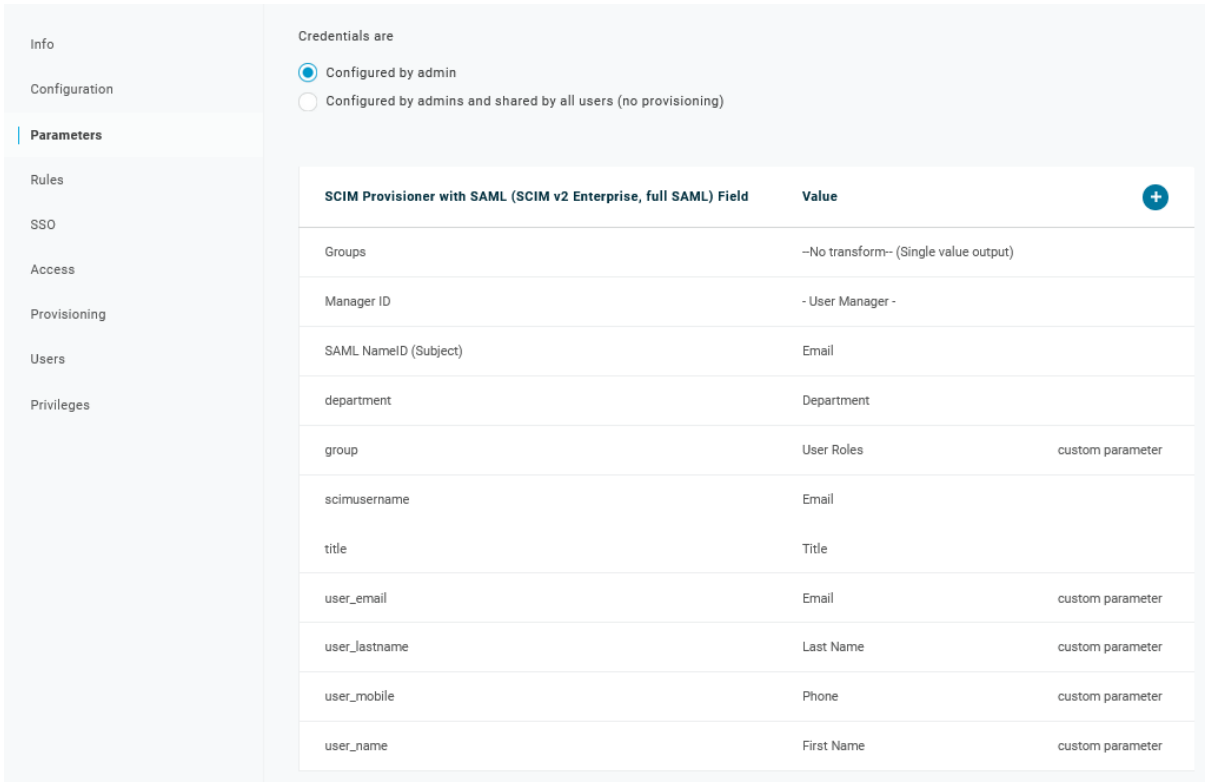


Lors de l'ajout d'un paramètre, veillez à cocher à la fois les options *Include in SAML assertion* et *Include in User Provisioning*.

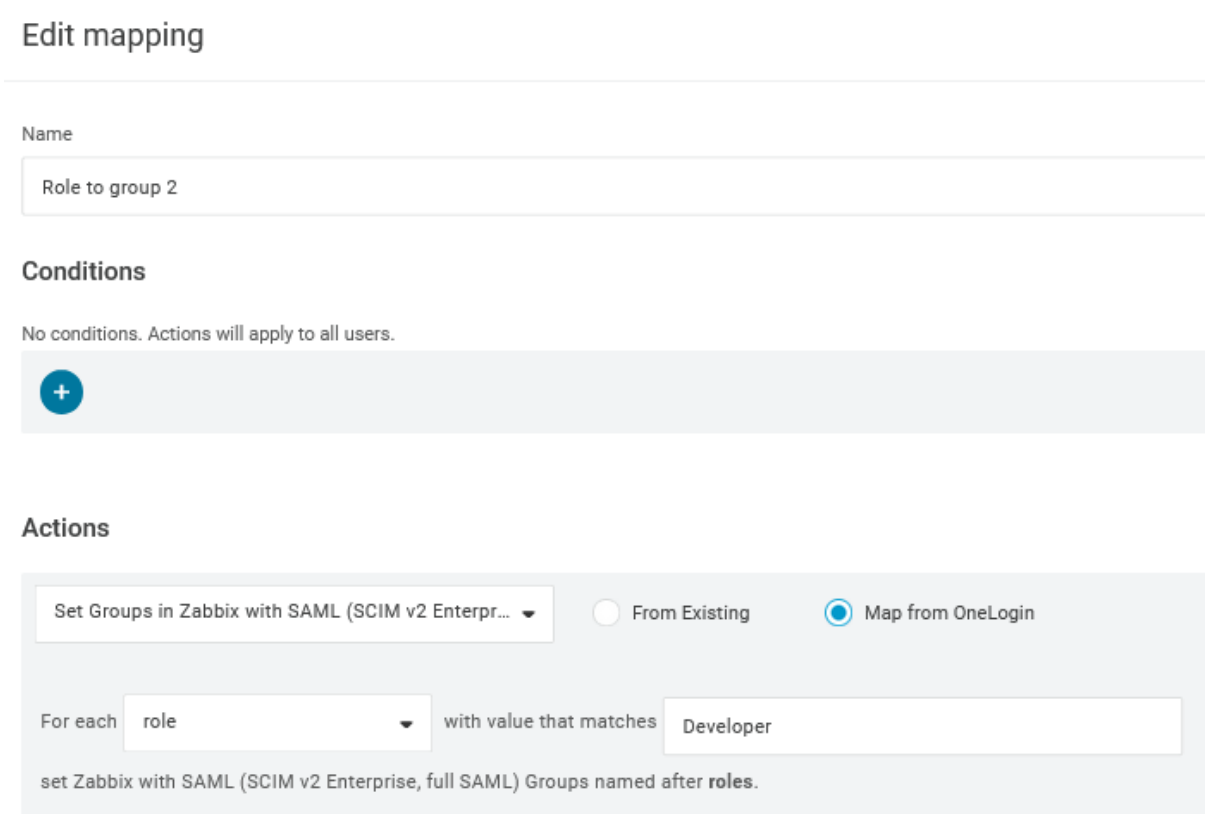
- Ajoutez un paramètre « group » correspondant aux rôles des utilisateurs dans OneLogin. Les rôles des utilisateurs seront transmis sous forme de chaîne, séparés par un point-virgule `;`. Les rôles utilisateur OneLogin seront utilisés pour créer des groupes d'utilisateurs dans Zabbix :



Vérifiez la liste des paramètres :



5. Dans la page *Rules*, créez des correspondances de rôles utilisateur avec 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 contenant plusieurs rôles.

6. Téléchargez le certificat IdP. Définissez-lui les permissions 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	Valeur d'exemple
ID d'entité IdP	URL de l'émetteur (voir l'onglet SSO de votre application dans OneLogin)	

Champ Zabbix	Champ de configuration dans OneLogin	Valeur d'exemple
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 de 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 zabbix.conf.php, vous pouvez coller le texte du certificat ou téléverser le fichier du certificat dans l'interface web lors de la configuration SAML — aucun fichier n'est requis sur le système de fichiers.

Si `$SSO['CERT_STORAGE'] = 'file'` est défini dans zabbix.conf.php, le certificat doit être disponible sur le système de fichiers (par défaut dans ui/conf/certs ou dans le chemin configuré dans zabbix.conf.php) et l'interface web ne stockera pas les certificats dans la base de données. Notez que si `$SSO['CERT_STORAGE']` n'est pas défini ou est commenté, le stockage dans des fichiers est supposé et les certificats sont lus depuis ui/conf/certs.

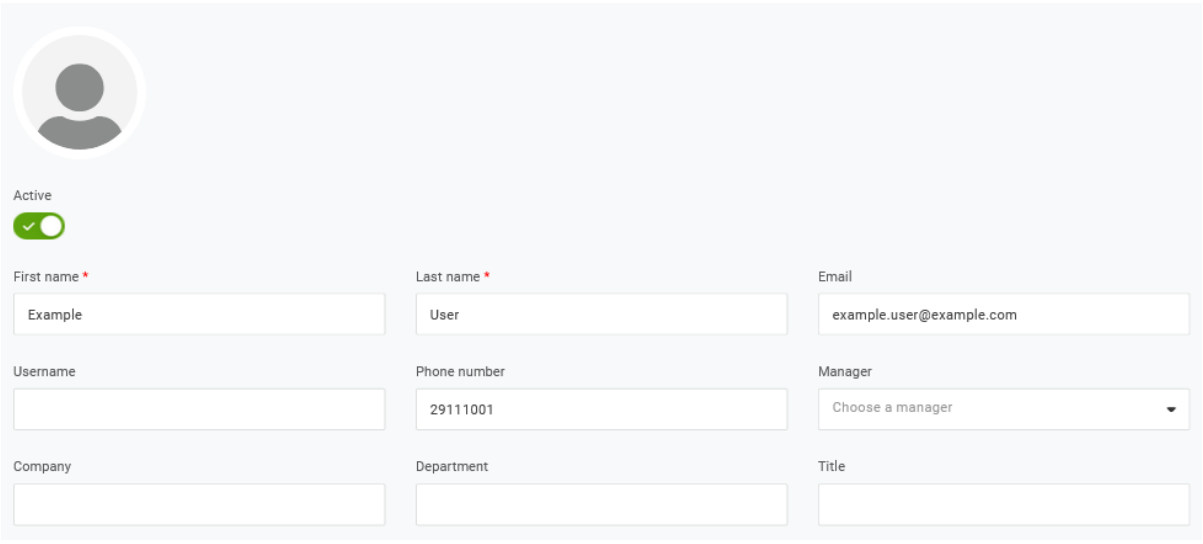
Vous pouvez accéder au téléchargement du certificat dans OneLogin via *Applications* -> *SSO* -> cliquez sur *View details* sous le certificat actuel.

3. Appuyez 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/de mettre à jour des utilisateurs et leurs rôles dans OneLogin et de les provisionner immédiatement dans Zabbix.

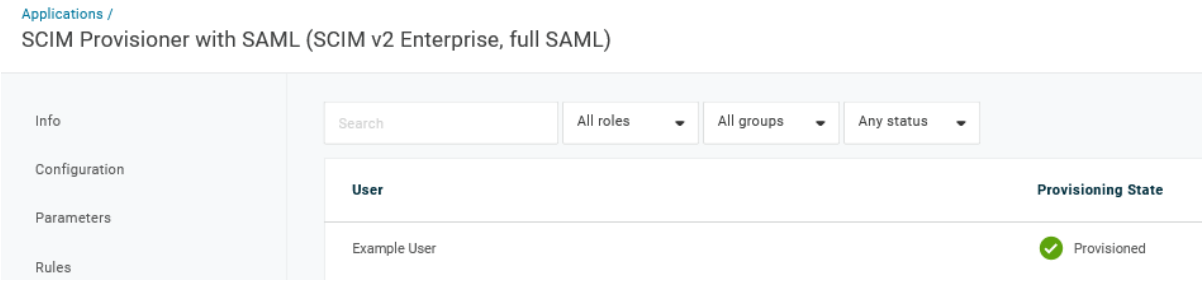
Par exemple, vous pouvez créer un nouvel utilisateur :



L'ajouter 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 :



Si le provisionnement a réussi, l'utilisateur peut être vu dans la liste des utilisateurs de 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

15 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, voir 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 assurer une communication correcte entre tous les éléments impliqués, assurez-vous que le fichier de configuration du serveur et les paramètres de configuration du frontend sont correctement configurés.

Serveur Zabbix

Les paramètres suivants dans le 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 être au format `http[s]://host:port/report`.

- Par défaut, le service web écoute sur le port 10053. Un port différent peut être spécifié dans le fichier de configuration du service web.
- La spécification du chemin `/report` est obligatoire (le 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 combien de processus d'écriture de rapports doivent être démarrés. S'il n'est pas défini ou s'il est égal à 0, la génération de rapports est désactivée. En fonction du nombre et de la fréquence des rapports requis, il est possible d'activer de 1 à 100 processus d'écriture de rapports.

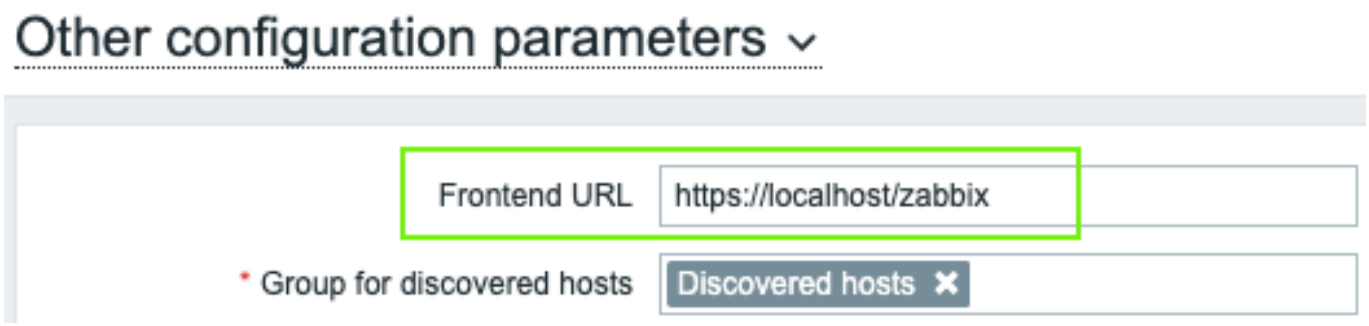
Exemple :

```
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.

16 Langues supplémentaires de l'interface web

Aperçu

Pour utiliser une autre langue que l'anglais dans l'interface Web de Zabbix, ses paramètres régionaux doivent être installés sur le serveur Web. De plus, l'extension PHP gettext est requise pour que les traductions fonctionnent.

Installation des paramètres régionaux

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 paramètres régionaux requis. Étant donné que Zabbix utilise l'encodage UTF-8, vous devez sélectionner des paramètres régionaux avec le jeu de caractères UTF-8.

Maintenant, exécutez :

```
 locale-gen
```

Redémarrez le serveur Web.

Les paramètres régionaux devraient maintenant être installées. Il peut être nécessaire de recharger la page du frontal Zabbix dans le navigateur en utilisant Ctrl + F5 pour que de nouvelles langues apparaissent.

Installer Zabbix

Si vous installez Zabbix directement depuis le repository git Zabbix, les fichiers de traduction doivent être générés manuellement. Pour générer des fichiers de traduction, exécutez :

```
make gettext
locale/make_mo.sh
```

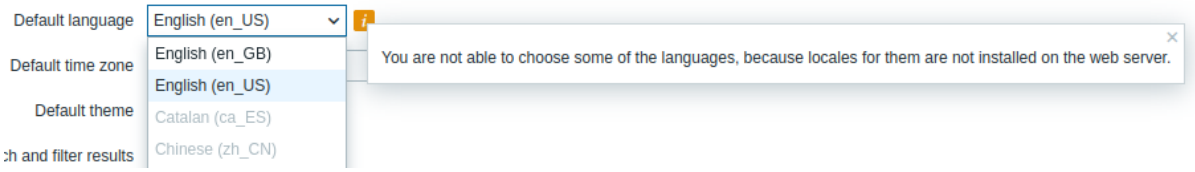
Cette étape n'est pas nécessaire lors de l'installation de Zabbix à partir de packages ou des fichiers source tar.gz.

Sélection d'une langue

Il existe plusieurs façons de sélectionner une langue dans l'interface web de Zabbix :

- Lors de l'installation de l'interface web - dans l'assistant d'installation du frontend. La langue sélectionnée sera définie comme langue par défaut du système.
- Après l'installation, la langue par défaut du système peut être modifiée dans la section de menu *Administration→General→GUI*.
- La langue d'un utilisateur particulier peut être modifiée dans le profil utilisateur.

Si une locale pour une langue n'est pas installée 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 une locale est manquante. En cliquant sur cette icône, le message suivant s'affichera : « Vous ne pouvez pas choisir certaines langues, car les locales correspondantes ne sont pas installées sur le serveur web. »



2 Configuration des processus

Veuillez utiliser la barre latérale pour accéder au contenu de cette section.

1 Serveur Zabbix

Aperçu

Les paramètres pris en charge par le fichier de configuration du serveur Zabbix (zabbix_server.conf) sont répertoriés dans cette section.

Les paramètres sont listés sans informations supplémentaires. Cliquez sur un paramètre pour voir tous les détails.

Parameter	Description
AlertScriptsPath	L'emplacement des scripts d'alerte personnalisés.
AllowRoot	Autoriser le serveur à s'exécuter en tant que « root ».
AllowSoftwareUpdateCheck	Autoriser l'interface Zabbix à recevoir des 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.
CacheSize	La taille du cache de configuration.
CacheUpdateFrequency	Ce paramètre détermine à quelle fréquence Zabbix mettra à jour le cache de configuration, en secondes.
DBHost	Le nom d'hôte de la base de données.
DBName	Le nom de la base de données.
DBPassword	Le mot de passe de la base de données.
DBPort	Le port de la base de données lorsqu'un socket local n'est pas utilisé.
DBSchema	Le nom du schéma de la base de données. Utilisé pour PostgreSQL.
DBSocket	Le chemin vers le fichier socket MySQL.
DBUser	L'utilisateur de la base de données.
DBTLSConnect	Définir cette option sur la valeur spécifiée impose l'utilisation d'une connexion TLS à la base de données.
DBTLSCAFile	Le chemin complet d'un fichier contenant les certificats de la ou des autorités de certification racines pour la vérification du certificat de la base de données.
DBTLSCertFile	Le chemin complet d'un fichier contenant le certificat du serveur Zabbix pour l'authentification auprès de la base de données.
DBTLSKeyFile	Le chemin complet d'un fichier contenant la clé privée pour l'authentification auprès de la base de données.
DBTLSCipher	La liste des suites de chiffrement que le serveur Zabbix autorise pour les protocoles TLS jusqu'à TLS v1.2 inclus. Pris en charge uniquement pour MySQL.
DBTLSCipher13	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	Spécifie le niveau de débogage.
EnableGlobalScripts	Activer les scripts globaux sur le serveur Zabbix.
ExportDir	Le répertoire pour l'export en temps réel des événements, de l'historique et des tendances au format JSON délimité par des retours à la ligne. Si défini, active l'export en temps réel.

Parameter	Description
ExportFileSize	La taille maximale par fichier d'export, en octets.
ExportType	La liste des types d'entités séparés par des virgules (events, history, trends) pour l'export en temps réel (tous les types par défaut).
ExternalScripts	L'emplacement des scripts externes.
Fping6Location	L'emplacement de fping6.
FpingLocation	L'emplacement de fping.
FrontendAllowedIP	Une liste d'adresses IP ou de plages CIDR séparées par des virgules qui sont autorisées à se connecter depuis le frontend.
HANodeName	Le nom du nœud du cluster haute disponibilité.
HistoryCacheSize	La taille du cache d'historique.
HistoryIndexCacheSize	La taille du cache d'index d'historique.
HistoryStorageDateIndex	Activer le prétraitement des valeurs d'historique dans le stockage de l'historique afin de stocker les valeurs dans différents index selon la date.
HistoryStorageURL	L'URL HTTP[S] du stockage de l'historique.
HistoryStorageTypes	Une liste de types de valeurs séparés par des virgules à envoyer au stockage de l'historique.
HousekeepingFrequency	Ce paramètre détermine à quelle fréquence Zabbix exécutera la procédure de nettoyage, en heures.
Include	Vous pouvez inclure des fichiers individuels ou tous les fichiers d'un répertoire dans le fichier de configuration.
JavaGateway	L'adresse IP (ou le nom d'hôte) du Java gateway Zabbix.
JavaGatewayPort	Le port sur lequel le Java gateway Zabbix écoute.
ListenBacklog	Le nombre maximal de connexions en attente dans la file TCP.
ListenIP	Une liste d'adresses IP séparées par des virgules sur lesquelles le trapper doit écouter.
ListenPort	Le port d'écoute du trapper.
LoadModule	Le module à charger au démarrage du serveur.
LoadModulePath	Le chemin complet vers l'emplacement des modules du serveur.
LogFile	Le nom du fichier journal.
LogFileSize	La taille maximale du fichier journal.
LogSlowQueries	Détermine combien de temps une requête de base de données peut prendre avant d'être journalisée, en millisecondes.
LogType	Le type de sortie du journal.
MaxConcurrentChecksPerPoller	Le nombre maximal de contrôles asynchrones pouvant être exécutés simultanément par chaque poller d'agent HTTP, poller d'agent ou poller SNMP.
MaxHousekeeperDelete	Pas plus de « MaxHousekeeperDelete » lignes (correspondant à [tablename], [field], [value]) ne seront supprimées par tâche lors d'un cycle de nettoyage.
NodeAddress	L'IP ou le nom d'hôte avec port facultatif pour remplacer la façon dont le frontend doit se connecter au serveur.
PidFile	Le nom du fichier PID.
ProblemHousekeepingFrequency	Détermine à quelle fréquence Zabbix supprimera les problèmes des déclencheurs supprimés.
ProxyConfigFrequency	Détermine à quelle fréquence le serveur Zabbix envoie les données de configuration à un proxy Zabbix.
ProxyDataFrequency	Détermine à quelle fréquence le serveur Zabbix demande les données d'historique à un proxy Zabbix.
ServiceManagerSyncFrequency	Détermine à quelle fréquence Zabbix synchronisera la configuration d'un gestionnaire de services.
SMSDevices	Une liste de fichiers modem séparés par des virgules que le serveur Zabbix est autorisé à utiliser.
SNMPTrapperFile	Le fichier temporaire utilisé pour transmettre les données du démon de trap SNMP au serveur.
SocketDir	Le répertoire où stocker les sockets IPC utilisés par les services internes de Zabbix.
SourceIP	L'adresse IP source.
SSHKeyLocation	L'emplacement des clés publiques et privées pour les contrôles SSH et les actions.
SSLCertLocation	L'emplacement des fichiers de certificat client SSL pour l'authentification du client.
SSLKeyLocation	L'emplacement des fichiers de clé privée SSL pour l'authentification du client.
SSLCALocation	Remplacer l'emplacement des fichiers d'autorité de certification (CA) pour la vérification du certificat SSL du serveur.
StartAgentPollers	Le nombre d'instances pré-forkées des pollers asynchrones de l'agent Zabbix.
StartAlerters	Le nombre d'instances pré-forkées des alerters.
StartBrowserPollers	Le nombre d'instances pré-forkées des pollers d'éléments de navigateur.
StartConnectors	Le nombre d'instances pré-forkées des workers de connecteur.
StartDBSyncers	Le nombre d'instances pré-forkées des synchroniseurs d'historique.
StartDiscoverers	Le nombre d'instances pré-forkées des workers de découverte.
StartEscalators	Le nombre d'instances pré-forkées des escalators.

Parameter	Description
StartHistoryPollers	Le nombre d'instances pré-forkées des pollers d'historique.
StartHTTPAgentPollers	Le nombre d'instances pré-forkées des pollers asynchrones d'agent HTTP.
StartHTTPPollers	Le nombre d'instances pré-forkées des pollers HTTP.
StartIPMIPollers	Le nombre d'instances pré-forkées des pollers IPMI.
StartJavaPollers	Le nombre d'instances pré-forkées des pollers Java.
StartLLDProcessors	Le nombre d'instances pré-forkées des workers de découverte de bas niveau (LLD).
StartODBCPollers	Le nombre d'instances pré-forkées des pollers ODBC.
StartPingers	Le nombre d'instances pré-forkées des pingers ICMP.
StartPollersUnreachable	Le nombre d'instances pré-forkées des pollers pour les hôtes injoignables (y compris IPMI et Java).
StartPollers	Le nombre d'instances pré-forkées des pollers.
StartPreprocessors	Le nombre d'instances pré-démarrées des workers de prétraitement.
StartProxyPollers	Le nombre d'instances pré-forkées des pollers pour les proxys passifs.
StartReportWriters	Le nombre d'instances pré-forkées des rédacteurs de rapports.
StartSNMPPollers	Le nombre d'instances pré-forkées des pollers SNMP asynchrones.
StartSNMPTrapper	Si défini sur 1, un processus trapper SNMP sera démarré.
StartTimers	Le nombre d'instances pré-forkées des timers.
StartTrappers	Le nombre d'instances pré-forkées des trappers.
StartVMwareCollectors	Le nombre d'instances pré-forkées des collecteurs VMware.
StatsAllowedIP	Une liste d'adresses IP séparées par des virgules, éventuellement en notation CIDR, ou de noms DNS d'instances Zabbix externes. La requête de statistiques ne sera acceptée que depuis les adresses listées ici.
Timeout	Spécifie combien de temps attendre (en secondes) pour établir une connexion et échanger des données avec un proxy Zabbix, un agent, le service web, ainsi que pour les contrôles SNMP (à l'exception des éléments SNMP `walk[OID]` et `get[OID]`).
TLSCAFile	Le chemin complet d'un fichier contenant les certificats de la ou des autorités de certification 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 serveur ou la chaîne de certificats, utilisé pour les communications chiffrées entre les composants Zabbix.
TLSCipherAll	La chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplace les critères de sélection par défaut des suites de chiffrement pour le chiffrement basé sur certificat et PSK.
TLSCipherAll13	La chaîne de chiffrement pour OpenSSL 1.1.1 ou version ultérieure en TLS 1.3. Remplace les critères de sélection par défaut des suites de chiffrement pour le chiffrement basé sur certificat et PSK.
TLSCipherCert	La chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplace les critères de sélection par défaut des suites de chiffrement pour le chiffrement basé sur certificat.
TLSCipherCert13	La chaîne de chiffrement pour OpenSSL 1.1.1 ou version ultérieure en TLS 1.3. Remplace les critères de sélection par défaut des suites de chiffrement pour le chiffrement basé sur certificat.
TLSCipherPSK	La chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplace les critères de sélection par défaut des suites de chiffrement pour le chiffrement basé sur PSK.
TLSCipherPSK13	La chaîne de chiffrement pour OpenSSL 1.1.1 ou version ultérieure en TLS 1.3. Remplace les critères de sélection par défaut des suites de chiffrement pour le chiffrement basé sur PSK.
TLSCRLFile	Le chemin complet d'un fichier contenant les certificats révoqués. Ce paramètre est utilisé pour les communications chiffrées entre les composants Zabbix.
TLSFrontendAccept	Quels types de connexions entrantes accepter depuis le frontend.
TLSFrontendCertIssuer	Émetteur de certificat frontend autorisé.
TLSFrontendCertSubject	Sujet de certificat frontend autorisé.
TLSKeyFile	Le chemin complet d'un fichier contenant la clé privée du serveur, utilisé pour les communications chiffrées entre les composants Zabbix.
TLSListen	Contrôle TLS sur le socket du trapper.
TmpDir	Le répertoire temporaire.
TrapperTimeout	Spécifie le délai d'expiration en secondes pour : - la récupération des données historiques depuis un proxy Zabbix ; - l'envoi des données de configuration à un proxy Zabbix ; - l'exécution d'un script global ou d'une commande distante sur le serveur Zabbix.
TrendCacheSize	La taille du cache de tendances.
TrendFunctionCacheSize	La taille du cache des fonctions de tendance.
UnavailableDelay	Détermine à quelle fréquence l'hôte est vérifié pour sa disponibilité pendant la période d'indisponibilité.

Parameter	Description
UnreachableDelay	Détermine à quelle fréquence l'hôte est vérifié pour sa disponibilité pendant la période d'injoignabilité.
UnreachablePeriod	Détermine après combien de secondes d'injoignabilité un hôte est considéré comme indisponible.
User	Abandonner les privilèges au profit d'un utilisateur spécifique existant sur le système.
ValueCacheSize	La taille du cache des valeurs d'historique.
Vault	Spécifie le fournisseur de coffre-fort.
VaultDBPath	Spécifie un emplacement à partir duquel les identifiants de base de données doivent être récupérés par clés.
VaultPrefix	Préfixe personnalisé pour le chemin ou la requête du coffre-fort.
VaultTLSCertFile	Le nom du fichier de certificat SSL utilisé pour l'authentification du client.
VaultTLSKeyFile	Le nom du fichier de clé privée SSL utilisé pour l'authentification du client.
VaultToken	Le jeton d'authentification du coffre-fort HashiCorp.
VaultURL	L'URL HTTP[S] du serveur de coffre-fort.
VMwareCacheSize	La taille de la mémoire partagée pour stocker les données VMware.
VMwareFrequency	Le délai en secondes entre les collectes de données depuis un même service VMware.
VMwarePerfFrequency	Le délai en secondes entre les récupérations des statistiques des compteurs de performance depuis un même service VMware.
VMwareTimeout	Le nombre maximal de secondes pendant lesquelles un collecteur vmware attendra une réponse du service VMware.
WebDriverURL	L'URL HTTP[S] de l'interface WebDriver.
WebServiceURL	L'URL HTTP[S] du service web Zabbix au format <host:port>/report.

Tous les paramètres sont facultatifs, sauf indication explicite qu'un paramètre est obligatoire.

Notez que :

- Les valeurs par défaut reflètent celles du démon, et non les valeurs des fichiers de configuration fournis.
- Les valeurs prennent en charge les variables d'environnement.
- 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 des paramètres

AlertScriptsPath

L'emplacement des scripts d'alerte personnalisés (dépend de la variable d'installation à la compilation *datadir*).

Par défaut : `/usr/local/share/zabbix/alertscripts`

AllowRoot

Autorise 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 de basculer vers l'utilisateur « zabbix ». N'a aucun effet si le serveur est démarré sous un utilisateur standard.

Par défaut : 0
 Valeurs : 0 - ne pas autoriser ; 1 - autoriser

AllowSoftwareUpdateCheck

Autoriser l'interface utilisateur 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

Permet au serveur de fonctionner avec des versions de base de données non prises en charge.

Par défaut : 0
 Valeurs : 0 - ne pas autoriser ; 1 - autoriser

CacheSize

La taille du cache de configuration, en octets. La taille de la mémoire partagée pour le stockage des données des hôtes, des éléments et des déclencheurs.

Par défaut : 32M
 Plage : 128K-64G

CacheUpdateFrequency

Ce paramètre détermine à quelle fréquence Zabbix effectuera la mise à jour du cache de configuration, en secondes. Voir aussi les options de contrôle d'exécution.

Par défaut : 10
 Plage : 1-3600

DBHost

Le nom d'hôte de la base de données.
Avec MySQL, `localhost` ou une chaîne vide entraîne l'utilisation d'un socket. Avec PostgreSQL, si la valeur est une chaîne vide, le socket Unix-domain par défaut est utilisé ; si elle est définie sur un chemin (par exemple, /var/run/pgbouncer), le socket Unix-domain spécifié est utilisé.

Par défaut : `localhost`

DBName

Le nom de la base de données.

Obligatoire : Oui

DBPassword

Le mot de passe de la base de données. Commentez cette ligne si aucun mot de passe n'est utilisé.

DBPort

Le port de la base de données lorsqu'un socket de domaine Unix par défaut n'est pas utilisé.[3]

Par défaut pour MySQL : 3306

Par défaut pour PostgreSQL : 5432

Plage : 1024-65535

DBSchema

Le nom du schéma de base de données. Utilisé pour PostgreSQL.

DBSocket

Le chemin vers le fichier de socket MySQL.[3]

DBUser

L'utilisateur de la base de données.

DBTLSConnect

Définir cette option sur l'une des valeurs suivantes impose l'utilisation d'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

Le chemin d'accès complet d'un fichier contenant les certificats de la ou des autorités de certification (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

Le chemin complet d'un fichier contenant le certificat du serveur Zabbix pour l'authentification auprès de la base de données.

DBTLSKeyFile

Le chemin complet d'un fichier contenant la clé privée pour l'authentification à la base de données.

DBTLSCipher

La liste des suites de chiffrement que le serveur Zabbix autorise pour les protocoles TLS jusqu'à TLS v1.2 inclus. Pris en charge uniquement pour MySQL.

DBTLSCipher13

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

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 (produit beaucoup d'informations) ;
5 - débogage étendu (produit encore plus d'informations).
Voir aussi les options de contrôle d'exécution.

Par défaut : 3
 Plage : 0-5

EnableGlobalScripts

Activer 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

ExportDir

Le répertoire pour l'export en temps réel des événements, de l'historique et des tendances au format JSON délimité par des sauts de ligne. S'il est défini, active l'export en temps réel.

ExportFileSize

La taille maximale par fichier d'exportation en octets. Utilisé pour la rotation si `ExportDir` est défini.

Par défaut : 1G
 Plage : 1M-1G

ExportType

La liste des types d'entités délimités par des virgules (events, history, trends) pour l'exportation en temps réel (tous les types par défaut). Valide uniquement si ExportDir est défini.
Notez que si ExportType est spécifié, mais que ExportDir ne l'est pas, il s'agit d'une erreur de configuration et le serveur ne démarrera pas.

Exemple pour l'exportation de l'historique et des tendances :

```
ExportType=history,trends
```

Exemple pour l'exportation des événements uniquement :

```
ExportType=events
```

ExternalScripts

L'emplacement des scripts externes (dépend de la variable d'installation de compilation `datadir`).

Par défaut : `/usr/local/share/zabbix/externalscripts`

Fping6Location

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

L'emplacement de fping. Assurez-vous que le binaire fping appartient à root et que l'indicateur SUID est défini.

Par défaut : `/usr/sbin/fping`

FrontendAllowedIP

Une liste d'adresses IP ou de plages CIDR, séparées par des virgules, autorisées à se connecter depuis le frontend. La connexion au frontend ne sera acceptée que depuis les adresses indiquées ici si ce paramètre est défini. Par défaut, toutes les connexions sont acceptées pour les requêtes du frontend. 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 :

```
FrontendAllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

HANodeName

Le nom du nœud du cluster de haute disponibilité. Lorsqu'il est vide, le serveur fonctionne en mode autonome et un nœud avec un nom vide est créé.

HistoryCacheSize

La taille du cache d'historique, en octets. Taille de la mémoire partagée pour le stockage des données d'historique.

Par défaut : 16M
 Plage : 128K-16G

HistoryIndexCacheSize

La taille du cache d'index d'historique, en octets. Taille de la mémoire partagée 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

HistoryStorageDateIndex

Active le prétraitement des valeurs d'historique dans le stockage de l'historique afin de stocker les valeurs dans différents index en fonction de la date.

Par défaut : 0
 Valeurs : 0 - désactiver ; 1 - activer

HistoryStorageURL

L'URL HTTP[S] du stockage de l'historique. Ce paramètre est utilisé pour la configuration d'Elasticsearch.

HistoryStorageTypes

Une liste de types de valeurs séparés par des virgules à envoyer au stockage de l'historique. Ce paramètre est utilisé pour la configuration de Elasticsearch.

Par défaut : `uint,dbl,str,log,text,json`

HousekeepingFrequency

Ce paramètre détermine à quelle fréquence Zabbix exécutera la procédure de nettoyage, en heures. Le nettoyage consiste à supprimer les informations obsolètes de la base de données.
Remarque : Pour éviter de surcharger le processus de nettoyage (par exemple, lorsque les périodes d'historique et de tendances sont fortement réduites), pas plus de 4 fois HousekeepingFrequency heures d'informations obsolètes ne sont supprimées en un seul cycle de nettoyage, pour chaque élément. Ainsi, si HousekeepingFrequency vaut 1, pas plus de 4 heures d'informations obsolètes (à partir de l'entrée la plus ancienne) ne seront supprimées par cycle.
Remarque : Afin de réduire la charge au démarrage du serveur, le nettoyage est reporté de 30 minutes après le démarrage du serveur. Ainsi, si HousekeepingFrequency vaut 1, la toute première procédure de nettoyage 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 nettoyage automatique en définissant HousekeepingFrequency sur 0. Dans ce cas, la procédure de nettoyage ne peut être lancée que par l'option de contrôle d'exécution *housekeeper_execute* et la période d'informations obsolètes supprimée en un seul cycle de nettoyage est égale à 4 fois la période écoulée depuis le dernier cycle de nettoyage, mais pas inférieure à 4 heures ni supérieure à 4 jours.
Voir aussi les options de contrôle d'exécution.

Par défaut : 1
 Plage : 0-24

Include

Vous pouvez inclure des fichiers individuels ou tous les fichiers d'un répertoire dans le fichier de configuration. Pour n'inclure que les fichiers pertinents dans le répertoire spécifié, le caractère générique astérisque est pris en charge pour la correspondance de motifs. Voir les notes spéciales concernant les limitations.

Exemple :

```
Include=/absolute/path/to/config/files/*.conf
```

JavaGateway

L'adresse IP (ou le nom d'hôte) du Zabbix Java gateway. Requis uniquement si des collecteurs Java sont démarrés.

JavaGatewayPort

Le port sur lequel le Java gateway Zabbix écoute.

Par défaut : 10052
 Plage : 1024-32767

ListenBacklog

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 jusqu'au « maximum spécifié par l'implémentation ».

Par défaut : SOMAXCONN
 Plage : 0 - INT_MAX

ListenIP

Une liste d'adresses IP séparées par des virgules sur lesquelles le trapper doit écouter.
Le trapper écoutera sur toutes les interfaces réseau si ce paramètre est absent.

Par défaut : 0.0.0.0

ListenPort

Le port d'écoute pour le trapper.

Par défaut : 10051
 Plage : 1024-32767

LoadModule

Le module à charger au démarrage du serveur. Les modules sont utilisés pour étendre les fonctionnalités du serveur. Le module doit être situé 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 est permis d'inclure plusieurs paramètres LoadModule.

LoadModulePath

Le chemin complet vers l'emplacement des modules du serveur. La valeur par défaut dépend des options de compilation.

LogFile

Le nom du fichier journal.

Obligatoire : Oui, si LogType est défini sur *file* ; sinon non

LogFileSize

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, pour quelque raison que ce soit, le fichier journal existant est tronqué et redémarré à zéro.

Par défaut : 1
 Plage : 0-1024
 Obligatoire : Oui, si LogType est défini sur *file* ; sinon non

LogSlowQueries

Détermine combien de temps une requête de base de données peut prendre avant d'être consignée dans le journal, en millisecondes.
0 - ne pas consigner les requêtes lentes.
Cette option est activée à partir de DebugLevel=3.

Par défaut : 0
 Plage : 0-3600000

LogType

Le type de la 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`

MaxConcurrentChecksPerPoller

Le nombre maximal de contrôles asynchrones pouvant être exécutés simultanément par chaque poller d'agent HTTP, poller d'agent ou poller SNMP. Voir StartHTTPAgentPollers, StartAgentPollers et StartSNMPPollers.

Par défaut : 1000
 Plage : 1-1000

MaxHousekeeperDelete

Pas plus de `MaxHousekeeperDelete` lignes (correspondant à [tablename], [field], [value]) ne seront supprimées par tâche lors d'un cycle de nettoyage.
Si cette valeur est définie sur 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 ! [2]
Ce paramètre s'applique uniquement à la suppression de l'historique et des tendances des éléments déjà supprimés.

Par défaut : 5000
 Plage : 0-1000000

NodeAddress

Adresse IP ou nom d'hôte avec port facultatif pour remplacer la façon dont l'interface web doit se connecter au serveur.
Format : <adresse>[:<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 web.

Voir aussi : paramètre HANodeName ; Activation de la haute disponibilité.

Par défaut : `localhost:10051`

PidFile

Nom du fichier PID.

Par défaut : `/tmp/zabbix_server.pid`

ProblemHousekeepingFrequency

Détermine à quelle fréquence Zabbix supprimera les problèmes des déclencheurs supprimés, en secondes.

Par défaut : 60
 Plage : 1-3600

ProxyConfigFrequency

Détermine à quelle fréquence le serveur Zabbix envoie les données de configuration à un proxy Zabbix, en secondes. Utilisé uniquement pour les proxys en mode passif.

Par défaut : 10
 Plage : 1-604800

ProxyDataFrequency

Détermine à quelle fréquence le serveur Zabbix demande les données d'historique à un proxy Zabbix, en secondes. Utilisé uniquement pour les proxys en mode passif.

Par défaut : 1
 Plage : 1-3600

ServiceManagerSyncFrequency

Détermine à quelle fréquence Zabbix synchronisera la configuration d'un gestionnaire de services, en secondes.

Par défaut : 60
 Plage : 1-3600

SMSDevices

Une liste de fichiers de modem séparés par des virgules autorisés à être utilisés par le serveur Zabbix.
L'envoi de SMS n'est pas possible si ce paramètre n'est pas défini.

Exemple :

```
SMSDevices=/dev/ttyUSB0,/dev/ttyUSB1
```

SNMPTrapperFile

Fichier temporaire utilisé pour transmettre les données du démon de traps SNMP au serveur.
Doit être identique à celui défini dans le fichier de configuration de zabbix_trap_receiver.pl ou de SNMPTT.

Par défaut : `/tmp/zabbix_traps.tmp`

SocketDir

Répertoire où stocker les sockets IPC utilisés par les services internes de Zabbix.

Par défaut : `/tmp`

SourceIP

Adresse IP source pour :

- les connexions sortantes vers Zabbix proxy 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 des éléments de script
- les requêtes HTTP JavaScript de prétraitement
- l'envoi d'e-mails de notification (connexions au serveur SMTP)
- les notifications webhook (connexions HTTP JavaScript)
- les connexions au Vault

SSHKeyLocation

Emplacement des clés publiques et privées pour les vérifications SSH et les actions.

SSLCertLocation

Emplacement des fichiers de certificat client SSL pour l'authentification du client.
Ce paramètre est utilisé uniquement dans la surveillance web.

SSLKeyLocation

Emplacement des fichiers de clé privée SSL pour l'authentification du client.
Ce paramètre est utilisé uniquement dans la supervision web.

SSLCALocation

Remplace l'emplacement des fichiers d'autorité de certification (CA) pour la vérification du certificat SSL du serveur. S'il n'est pas défini, le répertoire système global sera utilisé.
Notez que la valeur de ce paramètre sera définie comme l'option libcurl CURLOPT_CAPATH. Pour les versions de libcurl antérieures à 7.42.0, cela 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 dans la supervision web et dans l'authentification SMTP.

StartAgentPollers

Le nombre d'instances pré-forkées des pollers de l'agent Zabbix. Voir MaxConcurrentChecksPerPoller.

Par défaut : 1
 Plage : 0-1000

StartAlerters

Le nombre d'instances pré-forkées des alerters.

Par défaut : 3
 Plage : 1-100

StartBrowserPollers

Le nombre d'instances pré-forkées des pollers d'éléments de navigateur.

Par défaut : 1
 Plage : 0-1000

StartConnectors

Le nombre d'instances pré-forkées des workers de connecteur. Le processus de gestionnaire de connecteurs est démarré automatiquement lorsqu'un worker de connecteur est démarré.

Par défaut : 0
 Plage : 0-1000

StartDBSyncers

Le nombre d'instances pré-forkées des history syncers.
Remarque : Soyez prudent lors de la modification de cette valeur ; 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

Le nombre d'instances pré-forkées des workers de découverte[1].

Par défaut : 5
 Plage : 0-1000

StartEscalators

Le nombre d'instances pré-forkées des escalators.

Par défaut : 1
 Plage : 1-100

StartHistoryPollers

Le nombre d'instances pré-forkées des history pollers.
Requis uniquement pour les contrôles calculés.

Par défaut : 5
 Plage : 0-1000

StartHTTPAgentPollers

Le nombre d'instances pré-forkées des pollers d'agent HTTP. Voir MaxConcurrentChecksPerPoller.

Par défaut : 1
 Plage : 0-1000

StartHTTPPollers

Le nombre d'instances pré-forkées des collecteurs HTTP[1].

Par défaut : 1
 Plage : 0-1000

StartIPMIPollers

Le nombre d'instances pré-forkées des collecteurs IPMI.

Par défaut : 0
 Plage : 0-1000

StartJavaPollers

Le nombre d'instances pré-forkées de pollers Java[1].

Par défaut : 0
 Plage : 0-1000

StartLLDProcessors

Le nombre d'instances pré-forkées des workers de découverte de bas niveau (LLD)[1].
Le processus gestionnaire LLD est démarré automatiquement lorsqu'un worker LLD est démarré.

Par défaut : 2
 Plage : 1-100

StartODBCPollers

Le nombre d'instances pré-forkées des collecteurs ODBC[1].

Par défaut : 1
 Plage : 0-1000

StartPingers

Le nombre d'instances pré-forkées des ICMP pingers[1].

Par défaut : 1
 Plage : 0-1000

StartPollersUnreachable

Le nombre d'instances pré-forkées des 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

Le nombre d'instances pré-forkées de pollers[1].

Par défaut : 5
 Plage : 0-1000

StartPreprocessors

Le nombre d'instances pré-démarrées des threads de worker[1] de prétraitement doit être défini à une valeur au moins égale au nombre de cœurs CPU disponibles. Un plus grand nombre 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

StartProxyPollers

Le nombre d'instances pré-forkées des pollers pour les proxys passifs[1].

Par défaut : 1
 Plage : 0-250

StartReportWriters

Le nombre d'instances pré-forkées de report writers.
Si défini à 0, la génération planifiée de rapports est désactivée.
Le processus de gestion des rapports est démarré automatiquement lorsqu'un report writer est démarré.

Par défaut : 0
 Plage : 0-100

StartSNMPPollers

Le nombre d'instances pré-forkées des pollers SNMP. Voir MaxConcurrentChecksPerPoller.

Par défaut : 1
 Plage : 0-1000

StartSNMPTrapper

Si cette option est définie sur 1, un processus SNMP trapper sera démarré.

Par défaut : 0
 Plage : 0-1

StartTimers

Le nombre d'instances pré-forkées des timers.
Les timers traitent les périodes de maintenance.

Par défaut : 1
 Plage : 1-1000

StartTrappers

Le nombre d'instances pré-forkées de trappers[1].
Les trappers acceptent les connexions entrantes de Zabbix sender, des agents actifs et des proxys actifs.

Par défaut : 5
 Plage : 0-1000

StartVMwareCollectors

Le nombre d'instances VMware collector pré-forkées.

Par défaut : 0
 Plage : 0-250

StatsAllowedIP

Une liste d'adresses IP séparées par des virgules, éventuellement en notation CIDR, ou de noms DNS d'instances Zabbix externes. Les requêtes de statistiques seront acceptées uniquement depuis les adresses indiquées ici. 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 manière équivalente 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
```

Délai d'expiration

Indique combien de temps attendre (en secondes) pour établir une connexion et échanger des données avec le proxy Zabbix, l'agent, le service web et les vérifications SNMP héritées (numéro 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
- requêtes vers le service web Zabbix
- délai de communication pour une 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 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 à l'exécution
- connexions JMX
- récupération des statistiques depuis un proxy ou un serveur Zabbix distant
- envoi des réponses à l'interface web Zabbix
- requêtes DNS du poller asynchrone
- réponse pour le heartbeat des vérifications actives
- récupération des données (valeurs) de l'agent Zabbix 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'expiration **ne** sera **pas** utilisé pour les vérifications pour lesquelles des paramètres de délai d'expiration flexible sont configurés dans l'interface web (au niveau global, du proxy ou de l'élément).
Par exemple, les éléments SNMP `walk[OID]` et `get[OID]` utilisent le délai d'expiration configuré dans l'interface web ; les vérifications SNMP héritées utilisent toujours la valeur de délai d'expiration du serveur.

Par défaut : 3
 Plage : 1-30

TLSCAFile

Le chemin complet d'un fichier contenant les certificats de la ou des autorités de certification (CA) de niveau supérieur 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 serveur ou la chaîne de certificats, utilisé pour les communications chiffrées entre les composants Zabbix.

TLSCipherAll

La chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplace les critères de sélection par défaut de la suite 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

La chaîne de chiffrement pour OpenSSL 1.1.1 ou version plus récente en TLS 1.3. Remplace les critères de sélection par défaut des suites cryptographiques pour le chiffrement basé sur des 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

La chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplace les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur des 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

La chaîne de chiffrement pour OpenSSL 1.1.1 ou version plus récente en TLS 1.3. Remplace les critères de sélection par défaut de la suite cryptographique pour le chiffrement basé sur des certificats.

TLSCipherPSK

La chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplace 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

La chaîne de chiffrement pour OpenSSL 1.1.1 ou version plus récente en TLS 1.3. Remplace 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

Le chemin complet d'un fichier contenant les certificats révoqués. Ce paramètre est utilisé pour les communications chiffrées entre les composants Zabbix.

TLSFrontendAccept

Quelles connexions entrantes accepter depuis le frontend.

Plusieurs valeurs peuvent être spécifiées, séparées par des virgules :

- unencrypted - accepter les connexions sans chiffrement.
- cert - accepter les connexions sécurisées avec TLS et un certificat.

Par défaut : `unencrypted`

TLSFrontendCertIssuer

Émetteur de certificat frontend autorisé.

TLSFrontendCertSubject

Sujet de certificat frontend autorisé.

TLSKeyFile

Le chemin complet d'un fichier contenant la clé privée du serveur, utilisée pour les communications chiffrées entre les composants Zabbix.

TLSListen

Contrôle TLS sur le socket trapper.

Valeurs prises en charge :

- required - accepter uniquement les connexions TLS

TmpDir

Le répertoire temporaire.

Par défaut : `/tmp`

TrapperTimeout

Spécifie le délai d'expiration 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 distantes sur le serveur Zabbix

Par défaut : 300
 Plage : 1-300

TrendCacheSize

La taille du cache des tendances, en octets.
La taille de la mémoire partagée pour le stockage des données de tendances.

Par défaut : 4M
 Plage : 128K-16G

TrendFunctionCacheSize

La taille du cache des fonctions de tendance, en octets.
La taille de la mémoire partagée pour la mise en cache des données calculées des fonctions de tendance.

Par défaut : 4M
 Plage : 128K-2G

UnavailableDelay

Détermine à quelle fréquence l'hôte est vérifié pour sa disponibilité pendant la période d'indisponibilité, en secondes.

Par défaut : 60
 Plage : 1-3600

UnreachableDelay

Détermine à quelle fréquence l'hôte est vérifié pour sa disponibilité pendant la période d'inaccessibilité, en secondes.

Par défaut : 15
 Plage : 1-3600

UnreachablePeriod

Détermine après combien de secondes d'injoignabilité un hôte est considéré comme indisponible.

Par défaut : 45
 Plage : 1-3600

Utilisateur

Abandonne les privilèges au profit d'un utilisateur spécifique existant sur le système.
N'a d'effet que si le processus est exécuté en tant que « root » et que AllowRoot est désactivé.

Par défaut : `zabbix`

ValueCacheSize

La taille du cache des valeurs d'historique, en octets.
La taille de la mémoire partagée pour la mise en cache des requêtes de données d'historique des éléments.
La définition à 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.

Par défaut : 8M
 Plage : 0,128K-64G

Coffre-fort

Spécifie le fournisseur de coffre-fort :
HashiCorp - Moteur de secrets HashiCorp KV version 2
CyberArk - Fournisseur central d'identifiants CyberArk
Doit correspondre au fournisseur de coffre-fort défini dans l'interface web.

Par défaut : `HashiCorp`

VaultDBPath

Chemin ou requête du coffre-fort, selon le Vault, à partir duquel les identifiants de la base de données seront récupérés à l'aide de clés.

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

Un préfixe personnalisé pour le chemin ou la requête Vault, selon le Vault. Les valeurs par défaut les plus appropriées seront utilisées s'il 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

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-fort CyberArkCCP.

VaultTLSKeyFile

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-fort CyberArkCCP.

VaultToken

Le jeton d'authentification HashiCorp Vault qui doit avoir été généré exclusivement pour le serveur Zabbix, avec une autorisation en lecture seule sur les chemins spécifiés dans les macros Vault et une autorisation en lecture seule sur le chemin spécifié dans le paramètre de configuration optionnel VaultDBPath.
C'est une erreur si VaultToken et la variable d'environnement VAULT_TOKEN sont définis en même temps.

Obligatoire : Oui, si Vault est défini sur *HashiCorp* ; sinon non

VaultURL

L'URL HTTP[S] du serveur vault. Le répertoire système des certificats CA sera utilisé si SSLCALocation n'est pas spécifié.

Par défaut : `https://127.0.0.1:8200`

VMwareCacheSize

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'y a aucune instance de collecteur VMware configurée pour démarrer.

Par défaut : 8M
 Plage : 256K-2G

VMwareFrequency

Le délai, en secondes, entre les collectes de données à partir d'un seul service VMware.
Ce délai doit être défini sur le plus petit intervalle de mise à jour de tout élément de supervision VMware.

Par défaut : 60
 Plage : 10-86400

VMwarePerfFrequency

Le délai, en secondes, entre les récupérations des statistiques des compteurs de performance à partir d'un seul service VMware. Ce délai doit être défini sur le plus petit intervalle de mise à jour de tout élément de surveillance VMware qui utilise des compteurs de performance VMware.

Par défaut : 60
 Plage : 10-86400

VMwareTimeout

Le nombre maximal de secondes pendant lesquelles un collecteur vmware attendra une réponse du service VMware (vCenter ou hyperviseur ESX).

Par défaut : 10
 Plage : 1-300

WebServiceURL

L'URL HTTP[S] du service web Zabbix au format `http[s]://host:port/report`.

Exemple :

```
WebServiceURL=http://localhost:10053/report
```

Remarque : le schéma (`http://`) peut être omis uniquement pour les connexions non TLS (HTTP) ; si TLS est configuré, `https://` doit être utilisé.

WebDriverURL

URL HTTP[S] de l'interface WebDriver.

Exemple (utilisé avec le serveur autonome Selenium WebDriver) :

```
WebDriverURL=http://localhost:4444
```

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, peuvent 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 plus tard. Pour éviter ce type de problème, consultez le fichier de configuration du serveur Zabbix afin d'optimiser le nombre de vérifications et de processus simultanés. En outre, si nécessaire, assurez-vous que la limite de descripteurs de fichiers est définie à une valeur suffisamment élevée en vérifiant et en ajustant les limites système.

[2] Lorsqu'un grand nombre d'éléments est supprimé, cela augmente la charge sur la base de données, car le housekeeper devra supprimer toutes les données d'historique associées à ces éléments. Par exemple, si nous devons supprimer seulement 1 prototype d'élément du modèle, mais que ce modèle est lié à 50 hôtes et que, pour chaque hôte, le prototype est étendu à 100 éléments réels, 5000 éléments au total devront être supprimés (1*50*100). Si 500 est défini pour MaxHousekeeperDelete (MaxHousekeeperDelete=500), le processus housekeeper devra supprimer jusqu'à 2500000 valeurs (5000*500) des tables history et trends en un cycle pour les éléments supprimés.

[3] DBSocket et DBPort s'excluent mutuellement dans la configuration du serveur. Spécifiez-en un seul, ou laissez les deux non définis.

2 Proxy Zabbix

Aperçu

Les paramètres pris en charge par le fichier de configuration du proxy Zabbix (zabbix_proxy.conf) sont répertoriés dans cette section.

Les paramètres sont listés sans informations supplémentaires. Cliquez sur un paramètre pour voir tous les détails.

Parameter	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.
CacheSize	La taille du cache de configuration.
ConfigFrequency	Ce paramètre est **obsolète** (utilisez plutôt ProxyConfigFrequency). Fréquence à laquelle le proxy récupère les données de configuration depuis le serveur Zabbix, en secondes.
DataSenderFrequency	Le proxy enverra les données collectées au serveur toutes les N secondes.
DBHost	Le nom d'hôte de la base de données.

Parameter	Description
DBName	Le nom de la base de données ou le chemin vers le fichier de base de données pour SQLite3.
DBPassword	Le mot de passe de la base de données.
DBPort	Le port de la base de données lorsqu'un socket local n'est pas utilisé.
DBSchema	Le nom du schéma de la base de données. Utilisé pour PostgreSQL.
DBSocket	Le chemin vers le fichier socket MySQL.
DBUser	L'utilisateur de la base de données.
DBTLSConnect	Définir cette option sur la valeur spécifiée impose l'utilisation d'une connexion TLS à la base de données.
DBTLSCAFile	Le chemin complet d'un fichier contenant les certificats de la ou des autorités de certification racines pour la vérification du certificat de la base de données.
DBTLSCertFile	Le chemin complet d'un fichier contenant le certificat du proxy Zabbix pour l'authentification auprès de la base de données.
DBTLSKeyFile	Le chemin complet d'un fichier contenant la clé privée pour l'authentification auprès de la base de données.
DBTLSCipher	La liste des suites de chiffrement que le proxy Zabbix autorise pour les protocoles TLS jusqu'à TLS v1.2 inclus. Pris en charge uniquement pour MySQL.
DBTLSCipher13	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	Le niveau de débogage.
EnableRemoteCommands	Indique si les commandes distantes provenant du serveur Zabbix sont autorisées.
ExternalScripts	L'emplacement des scripts externes.
Fping6Location	L'emplacement de fping6.
FpingLocation	L'emplacement de fping.
HistoryCacheSize	La taille du cache d'historique.
HistoryIndexCacheSize	La taille du cache d'index d'historique.
Hostname	Un nom de proxy unique, sensible à la casse.
HostnameItem	L'élément utilisé pour définir Hostname s'il n'est pas défini.
HousekeepingFrequency	Fréquence à laquelle Zabbix exécutera la procédure de nettoyage, en heures.
Include	Vous pouvez inclure des fichiers individuels ou tous les fichiers d'un répertoire dans le fichier de configuration.
JavaGateway	L'adresse IP (ou le nom d'hôte) du Zabbix Java gateway.
JavaGatewayPort	Le port sur lequel le Zabbix Java gateway écoute.
ListenBacklog	Le nombre maximal de connexions en attente dans la file TCP.
ListenIP	Une liste d'adresses IP séparées par des virgules sur lesquelles le trapper doit écouter.
ListenPort	Le port d'écoute du trapper.
LoadModule	Le module à charger au démarrage du proxy.
LoadModulePath	Le chemin complet vers l'emplacement des modules du proxy.
LogFile	Le nom du fichier journal.
LogFileSize	La taille maximale du fichier journal.
LogRemoteCommands	Activer la journalisation des commandes shell exécutées en tant qu'avertissements.
LogSlowQueries	Durée maximale qu'une requête de base de données peut prendre avant d'être journalisée.
LogType	Le type de sortie du journal.
MaxConcurrentChecksPerPoller	Le nombre maximal de vérifications asynchrones pouvant être exécutées simultanément par chaque poller HTTP agent, poller agent ou poller SNMP.
PidFile	Le nom du fichier PID.
ProxyBufferMode	Spécifie le mécanisme de stockage des données d'historique, de découverte et d'auto-enregistrement (disque/mémoire/hybride).
ProxyConfigFrequency	Fréquence à laquelle le proxy récupère les données de configuration depuis le serveur Zabbix, en secondes.
ProxyLocalBuffer	Le proxy conservera les données localement pendant N heures, même si elles ont déjà été synchronisées avec le serveur.
ProxyMemoryBufferAge	L'âge maximal des données dans le tampon mémoire du proxy, en secondes.
ProxyMemoryBufferSize	La taille du cache en mémoire partagée pour les données collectées d'historique, de découverte et d'auto-enregistrement.
ProxyMode	Le mode de fonctionnement du proxy (actif/passif).
ProxyOfflineBuffer	Le proxy conservera les données pendant N heures en cas d'absence de connectivité avec le serveur Zabbix.
Server	Si ProxyMode est défini sur le mode actif : adresse IP ou nom DNS du serveur Zabbix (adresse:port) ou cluster (adresse:port;adresse2:port) à partir duquel récupérer les données de configuration et auquel envoyer les données. Si ProxyMode est défini sur le mode passif : liste d'adresses IP séparées par des virgules, éventuellement en notation CIDR, ou noms DNS du serveur Zabbix.

Parameter	Description
SNMPTrapperFile	Le fichier temporaire utilisé pour transmettre les données du démon de trap SNMP au proxy.
SocketDir	Le répertoire où stocker les sockets IPC utilisés par les services internes de Zabbix.
SourceIP	L'adresse IP source.
SSHKeyLocation	L'emplacement des clés publiques et privées pour les vérifications SSH et les actions.
SSLCertLocation	L'emplacement des fichiers de certificat client SSL pour l'authentification du client.
SSLKeyLocation	L'emplacement des fichiers de clé privée SSL pour l'authentification du client.
SSLCALocation	Remplace l'emplacement des fichiers d'autorité de certification (CA) pour la vérification du certificat SSL du serveur.
StartAgentPollers	Le nombre d'instances pré-forkées de pollers asynchrones Zabbix agent.
StartBrowserPollers	Le nombre d'instances pré-forkées de pollers d'éléments navigateur.
StartDBSyncers	Le nombre d'instances pré-forkées de synchroniseurs d'historique.
StartDiscoverers	Le nombre d'instances pré-forkées de workers de découverte.
StartHTTPAgentPollers	Le nombre d'instances pré-forkées de pollers asynchrones HTTP agent.
StartHTTPPollers	Le nombre d'instances pré-forkées de pollers HTTP.
StartIPMIPollers	Le nombre d'instances pré-forkées de pollers IPMI.
StartJavaPollers	Le nombre d'instances pré-forkées de pollers Java.
StartODBCPollers	Le nombre d'instances pré-forkées de pollers ODBC.
StartPingers	Le nombre d'instances pré-forkées de pingers ICMP.
StartPollersUnreachable	Le nombre d'instances pré-forkées de pollers pour les hôtes injoignables (y compris IPMI et Java).
StartPollers	Le nombre d'instances pré-forkées de pollers.
StartPreprocessors	Le nombre d'instances pré-démarrées de workers de prétraitement.
StartSNMPPollers	Le nombre d'instances pré-forkées de pollers SNMP asynchrones.
StartSNMPTrapper	Si défini sur 1, un processus de trapper SNMP sera démarré.
StartTrappers	Le nombre d'instances pré-forkées de trappers.
StartVMwareCollectors	Le nombre d'instances pré-forkées de collecteurs VMware.
StatsAllowedIP	Une liste d'adresses IP séparées par des virgules, éventuellement en notation CIDR, ou de noms DNS d'instances Zabbix externes. La requête de statistiques ne sera acceptée que depuis les adresses listées ici.
Timeout	Spécifie combien de temps attendre (en secondes) pour établir une connexion et échanger des données avec le proxy Zabbix, l'agent, le service web, ainsi que pour les vérifications SNMP (à l'exception des éléments SNMP `walk[OID]` et `get[OID]`).
TLSAccept	Quelles connexions entrantes accepter depuis le serveur Zabbix.
TLSCAFile	Le chemin complet d'un fichier contenant les certificats de la ou des autorités de certification 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 serveur ou la chaîne de certificats, utilisé pour les communications chiffrées entre les composants Zabbix.
TLSCipherAll	La chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplace les critères par défaut de sélection des suites de chiffrement pour le chiffrement basé sur certificat et PSK.
TLSCipherAll13	La chaîne de chiffrement pour OpenSSL 1.1.1 ou version ultérieure en TLS 1.3. Remplace les critères par défaut de sélection des suites de chiffrement pour le chiffrement basé sur certificat et PSK.
TLSCipherCert	La chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplace les critères par défaut de sélection des suites de chiffrement pour le chiffrement basé sur certificat.
TLSCipherCert13	La chaîne de chiffrement pour OpenSSL 1.1.1 ou version ultérieure en TLS 1.3. Remplace les critères par défaut de sélection des suites de chiffrement pour le chiffrement basé sur certificat.
TLSCipherPSK	La chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplace les critères par défaut de sélection des suites de chiffrement pour le chiffrement basé sur PSK.
TLSCipherPSK13	La chaîne de chiffrement pour OpenSSL 1.1.1 ou version ultérieure en TLS 1.3. Remplace les critères par défaut de sélection des suites de chiffrement pour le chiffrement basé sur PSK.
TLSConnect	Comment le proxy doit se connecter au serveur Zabbix.
TLSCRLFile	Le chemin complet d'un fichier contenant les certificats révoqués. Ce paramètre est utilisé pour les communications chiffrées entre les composants Zabbix.
TLSKeyFile	Le chemin complet d'un fichier contenant la clé privée du proxy, utilisée pour les communications chiffrées entre les composants Zabbix.
TLSListen	Contrôle TLS sur le socket du trapper.
TLSPSKFile	Le chemin complet d'un fichier contenant la clé pré-partagée du proxy, utilisée pour les communications chiffrées avec le serveur Zabbix.
TLSPSKIdentity	La chaîne d'identité de la clé pré-partagée, utilisée pour les communications chiffrées avec le serveur Zabbix.

Parameter	Description
TLSServerCertIssuer	L'émetteur autorisé du certificat du serveur.
TLSServerCertSubject	Le sujet autorisé du certificat du serveur.
TmpDir	Le répertoire temporaire.
TrapperTimeout	Spécifie le délai d'expiration 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	Fréquence à laquelle la disponibilité d'un hôte est vérifiée pendant la période d'indisponibilité.
UnreachableDelay	Fréquence à laquelle la disponibilité d'un hôte est vérifiée pendant la période d'injoignabilité.
UnreachablePeriod	Après combien de secondes d'injoignabilité l'hôte doit être considéré comme indisponible.
User	Abandonner les privilèges au profit d'un utilisateur spécifique existant sur le système.
Vault	Le fournisseur de coffre-fort.
VaultDBPath	L'emplacement à partir duquel les identifiants de base de données doivent être récupérés par clés.
VaultPrefix	Préfixe personnalisé pour le chemin ou la requête du coffre-fort.
VaultTLSCertFile	Le nom du fichier de certificat SSL utilisé pour l'authentification du client.
VaultTLSKeyFile	Le nom du fichier de clé privée SSL utilisé pour l'authentification du client.
VaultToken	Le jeton d'authentification HashiCorp vault.
VaultURL	L'URL HTTP[S] du serveur de coffre-fort.
VMwareCacheSize	La taille de la mémoire partagée pour le stockage des données VMware.
VMwareFrequency	Le délai en secondes entre les collectes de données depuis un même service VMware.
VMwarePerfFrequency	Le délai en secondes entre les récupérations des statistiques des compteurs de performance depuis un même service VMware.
VMwareTimeout	Le nombre maximal de secondes pendant lesquelles un collecteur vmware attendra une réponse du service VMware.
WebDriverURL	L'URL HTTP[S] de l'interface WebDriver.

Tous les paramètres sont facultatifs, sauf indication explicite qu'un paramètre est obligatoire.

Notez que :

- Les valeurs par défaut reflètent celles du démon, et non celles des fichiers de configuration fournis.
- Les valeurs prennent en charge les variables d'environnement.
- Zabbix ne prend en charge que 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 des paramètres

AllowRoot

Autorise 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 plutôt de basculer vers l'utilisateur « zabbix ». N'a aucun effet s'il est démarré sous un utilisateur ordinaire.

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

CacheSize

La taille du cache de configuration, en octets. La taille de la mémoire partagée pour le stockage des données des hôtes et des éléments.

Par défaut : 32M
 Plage : 128K-64G

ConfigFrequency

Ce paramètre est **obsolète** (utilisez ProxyConfigFrequency à la place).
Fréquence à laquelle le proxy récupère les données de configuration du serveur Zabbix, en secondes.
Paramètre de proxy actif. Ignoré pour les proxys passifs (voir le paramètre ProxyMode).

Par défaut : 3600
 Plage : 1-604800

DataSenderFrequency

Le proxy enverra les données collectées au serveur toutes les N secondes. Notez qu'un proxy actif interrogera toujours le serveur Zabbix chaque seconde pour les tâches de commandes distantes.
Paramètre de proxy actif. Ignoré pour les proxys passifs (voir le paramètre ProxyMode).

Par défaut : 1
 Plage : 1-3600

DBHost

Le nom d'hôte de la base de données.
Avec MySQL, `localhost` ou une chaîne vide entraînent l'utilisation d'un socket. Avec PostgreSQL, seule une chaîne vide entraîne une tentative d'utilisation d'un socket.

Par défaut : `localhost`

DBName

Le nom de la base de données ou le chemin vers le 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

Le mot de passe de la base de données. Commentez cette ligne si aucun mot de passe n'est utilisé. Ignoré pour SQLite.

DBPort

Le port de la base de données lorsqu'un socket local n'est pas utilisé.[1] Ignoré pour SQLite.

Valeur par défaut pour MySQL : 3306

Valeur par défaut pour PostgreSQL : 5432

Plage : 1024-65535

DBSchema

Le nom du schéma de base de données. Utilisé pour PostgreSQL.

DBSocket

Le chemin vers le fichier de socket MySQL.[1]
Le port de la base de données lorsqu'un socket local n'est pas utilisé. Ignoré pour SQLite.

Par défaut : 3306

DBUser

L'utilisateur de la base de données. Ignoré pour SQLite.

DBTLSConnect

La définition de cette option impose l'utilisation d'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

Le chemin d'accès complet d'un fichier contenant les certificats de la ou des autorités de certification (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

Le chemin complet d'un fichier contenant le certificat du proxy Zabbix pour l'authentification auprès de la base de données.

DBTLSKeyFile

Le chemin complet d'un fichier contenant la clé privée utilisée pour l'authentification auprès de la base de données.

DBTLSCipher

La liste des suites de chiffrement que le proxy Zabbix autorise pour les protocoles TLS jusqu'à TLS v1.2 inclus. Pris en charge uniquement pour MySQL.

DBTLSCipher13

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

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 (produit beaucoup d'informations) ;
5 - débogage étendu (produit encore plus d'informations).

Par défaut : 3
 Plage : 0-5

EnableRemoteCommands

Indique si les commandes distantes provenant du serveur Zabbix sont autorisées.

Par défaut : 0
 Valeurs : 0 - non autorisé ; 1 - autorisé

ExternalScripts

L'emplacement des scripts externes (dépend de la variable d'installation à la compilation `datadir`).

Par défaut : `/usr/local/share/zabbix/externalscripts`

Fping6Location

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

L'emplacement de fping. Assurez-vous que le binaire fping appartient à root et que l'indicateur SUID est défini.

Par défaut : `/usr/sbin/fping`

HistoryCacheSize

La taille du cache d'historique, en octets. La taille de la mémoire partagée pour le stockage des données d'historique.

Par défaut : `16M`
 Plage : 128K-16G

HistoryIndexCacheSize

La taille du cache d'index d'historique, en octets. La taille de la mémoire partagée 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

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

L'élément utilisé pour définir Hostname s'il n'est pas défini (il sera exécuté sur le proxy de la même manière que sur un agent). Ignoré si Hostname est défini.
Ne prend pas en charge les UserParameters, les compteurs de performance ni les alias, mais prend en charge system.run[].

Par défaut : system.hostname

HousekeepingFrequency

Fréquence à laquelle Zabbix exécutera la procédure de nettoyage (en heures). Le nettoyage consiste à supprimer les informations obsolètes de la base de données.
Remarque : afin de réduire la charge au démarrage du proxy, le nettoyage est reporté de 30 minutes après le démarrage du proxy. Ainsi, si HousekeepingFrequency vaut 1, la toute première procédure de nettoyage après le démarrage du proxy sera exécutée après 30 minutes, puis se répétera toutes les heures par la suite.
Il est possible de désactiver le nettoyage automatique en définissant HousekeepingFrequency à 0. Dans ce cas, la procédure de nettoyage ne peut être lancée que par l'option de contrôle d'exécution *housekeeper_execute*.

Par défaut : 1
 Plage : 0-24

Inclure

Vous pouvez inclure des fichiers individuels ou tous les fichiers d'un répertoire dans le fichier de configuration.
Pour n'inclure que les fichiers pertinents dans le répertoire spécifié, le caractère générique astérisque est pris en charge pour la correspondance de motifs.
Consultez les notes spéciales concernant les limitations.

Exemple :

```
Include=/absolute/path/to/config/files/*.conf
```

JavaGateway

L'adresse IP (ou le nom d'hôte) du Java gateway Zabbix. Requis uniquement si des collecteurs Java sont démarrés.

JavaGatewayPort

Le port sur lequel le Java gateway Zabbix écoute.

Par défaut : 10052
 Plage : 1024-32767

ListenBacklog

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 jusqu'au « maximum spécifié par l'implémentation ».

Par défaut : SOMAXCONN
 Plage : 0 - INT_MAX

ListenIP

Une liste d'adresses IP séparées par des virgules sur lesquelles le trapper doit écouter.
Le trapper écoutera sur toutes les interfaces réseau si ce paramètre est absent.

Par défaut : 0.0.0.0

ListenPort

Le port d'écoute pour le trapper.

Par défaut : 10051
 Plage : 1024-32767

LoadModule

Le module à charger au démarrage du proxy. Les modules sont utilisés pour étendre les fonctionnalités du proxy. Le module doit être situé 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 est permis d'inclure plusieurs paramètres LoadModule.

LoadModulePath

Le chemin complet vers l'emplacement des modules du proxy. La valeur par défaut dépend des options de compilation.

LogFile

Le nom du fichier journal.

Obligatoire : Oui, si LogType est défini sur *file* ; sinon non

LogFileSize

La taille maximale d'un fichier journal en Mo.
0 - désactiver la rotation automatique des journaux.
Remarque : si la limite de taille du fichier journal est atteinte et que la rotation du fichier échoue, pour quelque raison que ce soit, le fichier journal existant est tronqué et redémarré à zéro.

Par défaut : 1
 Plage : 0-1024

LogRemoteCommands

Active la journalisation des commandes shell exécutées en tant qu'avertissements.

Par défaut : 0
 Valeurs : 0 - désactivé, 1 - activé

LogType

Le type de la 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

LogSlowQueries

Durée maximale qu'une requête de base de données peut prendre avant d'être consignée dans le journal (en millisecondes).
0 - ne pas consigner les requêtes lentes.
Cette option est activée à partir de DebugLevel=3.

Par défaut : 0
 Plage : 0-3600000

MaxConcurrentChecksPerPoller

Le nombre maximal de contrôles asynchrones pouvant être exécutés 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

Le nom du fichier PID.

Par défaut : `/tmp/zabbix_proxy.pid`

ProxyBufferMode

Spécifie le mécanisme de stockage des données d'historique, de découverte réseau et d'auto-enregistrement : *disk* - les données sont stockées dans la base de données et envoyées depuis la base de données ; *memory* - les données sont stockées en mémoire et envoyées depuis la mémoire. Si le tampon n'a plus assez de mémoire, les anciennes données seront supprimées. Lors de l'arrêt, le tampon est supprimé. *hybrid* - le tampon du proxy fonctionne normalement comme en mode mémoire jusqu'à ce qu'il n'ait plus assez de mémoire ou que l'enregistrement le plus ancien dépasse l'ancienneté configurée. Dans ce cas, le tampon est vidé dans la base de données et fonctionne comme en mode disque jusqu'à ce que toutes les données aient été envoyées, puis il recommence à fonctionner en mémoire. Lors de l'arrêt, le tampon mémoire est vidé dans la base de données.

Voir aussi : Tampon mémoire du proxy.

Par défaut : `disk`
 Valeurs : disk ; memory ; hybrid

ProxyConfigFrequency

Fréquence à laquelle le proxy récupère les données de configuration depuis le serveur Zabbix, en secondes.
Paramètre de proxy actif. Ignoré pour les proxys passifs (voir le paramètre ProxyMode).

Par défaut : 10
 Plage : 1-604800

ProxyLocalBuffer

Le proxy conservera les données localement pendant N heures, même si les données ont déjà été synchronisées avec le serveur.
Ce paramètre peut être utilisé si les données locales doivent être utilisées par des applications tierces.

Par défaut : 0
 Plage : 0-720

ProxyMemoryBufferAge

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 basculer en mode base de données jusqu'à ce que tous les enregistrements soient téléversés vers le serveur. Ce paramètre doit être inférieur ou égal au paramètre ProxyOfflineBuffer.

Par défaut : 0
 Plage : 0;600-864000

ProxyMemoryBufferSize

Taille du cache de mémoire partagée pour les données d'historique, de découverte et d'auto-enregistrement collectées, en octets. S'il est activé (non nul), le proxy conservera en mémoire les données d'historique, de découverte et d'auto-enregistrement, sauf si le cache est plein ou si les enregistrements stockés sont plus anciens que la valeur définie par ProxyMemoryBufferAge. Ce paramètre ne peut pas être utilisé avec le paramètre ProxyLocalBuffer.

Par défaut : 0
 Plage : 0;128K-2G

ProxyMode

Le mode de fonctionnement du proxy.
0 - proxy en mode actif
1 - proxy en mode passif
Notez que les données de configuration (sensibles) du proxy peuvent devenir accessibles aux parties 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

Le proxy conservera les données pendant N 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 le *mode actif* :
Adresse IP ou nom DNS (adresse:port) du serveur Zabbix, ou cluster (adresse:port;adresse2:port), à partir duquel obtenir les données de configuration et auquel 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 le *mode passif* :
Liste d'adresses IP du serveur Zabbix séparées par des virgules, éventuellement en notation CIDR, ou de noms DNS. Les connexions entrantes seront acceptées uniquement depuis les adresses indiquées 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 traitées de manière identique.
'::/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 :

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Obligatoire : oui

SNMPTrapperFile

Un fichier temporaire utilisé pour transmettre les données du démon de traps SNMP au proxy.
Doit être identique à celui défini dans le fichier de configuration de zabbix_trap_receiver.pl ou de SNMPTT.

Par défaut : `/tmp/zabbix_traps.tmp`

SocketDir

Le répertoire dans lequel stocker les sockets IPC utilisés par les services internes de Zabbix.

Par défaut : `/tmp`

SourceIP

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 des éléments de script
- les requêtes HTTP JavaScript de prétraitement
- les connexions au coffre-fort

SSHKeyLocation

L'emplacement des clés publiques et privées pour les contrôles SSH et les actions.

SSLCertLocation

L'emplacement des fichiers de certificat client SSL pour l'authentification du client.
Ce paramètre est utilisé uniquement dans la supervision web.

SSLKeyLocation

L'emplacement des fichiers de clé privée SSL pour l'authentification du client.
Ce paramètre est utilisé uniquement dans la supervision web.

SSLCALocation

L'emplacement des fichiers d'autorité de certification (CA) pour la vérification du certificat SSL du serveur.
Notez que la valeur de ce paramètre sera définie comme l'option libcurl CURLOPT_CAPATH. Pour les versions de libcurl antérieures à 7.42.0, cela 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 dans la supervision web et dans l'authentification SMTP.

StartAgentPollers

Le nombre d'instances pré-forkées des pollers de l'agent Zabbix. Voir MaxConcurrentChecksPerPoller.

Par défaut : 1
 Plage : 0-1000

StartBrowserPollers

Le nombre d'instances pré-forkées des pollers d'éléments de navigateur.

Par défaut : 1
 Plage : 0-1000

StartDBSyncers

Le nombre d'instances pré-forkées des history syncers.
Remarque : Soyez prudent lors de la modification de cette valeur ; l'augmenter peut faire plus de mal que de bien.

Par défaut : 4
 Plage : 1-100

StartDiscoverers

Le nombre d'instances pré-forkées des workers de découverte.

Par défaut : 5
 Plage : 0-1000

StartHTTPAgentPollers

Le nombre d'instances pré-forkées des pollers d'agent HTTP. Voir MaxConcurrentChecksPerPoller.

Par défaut : 1
 Plage : 0-1000

StartHTTPPollers

Le nombre d'instances pré-forkées des pollers HTTP.

Par défaut : 1
 Plage : 0-1000

StartIPMIPollers

Le nombre d'instances pré-forkées des pollers IPMI.

Par défaut : 0
 Plage : 0-1000

StartJavaPollers

Le nombre d'instances pré-forkées de Java pollers.

Par défaut : 0
 Plage : 0-1000

StartODBCPollers

Le nombre d'instances pré-forkées des pollers ODBC.

Par défaut : 1
 Plage : 0-1000

StartPingers

Le nombre d'instances pré-forkées des ICMP pingers.

Par défaut : 1
 Plage : 0-1000

StartPollersUnreachable

Le nombre d'instances pré-forkées des pollers pour les hôtes injoignables (y compris IPMI et Java). 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

Le nombre d'instances pré-forkées de pollers.

Par défaut : 5
 Plage : 0-1000

StartPreprocessors

Le nombre d'instances pré-démarrées des threads worker de prétraitement doit être défini à une valeur au moins égale au nombre de cœurs CPU disponibles. Un plus grand nombre 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

Le nombre d'instances pré-forkées des pollers SNMP. Voir MaxConcurrentChecksPerPoller.

Par défaut : 1
 Plage : 0-1000

StartSNMPTrapper

Si cette option est définie sur 1, un processus de SNMP trapper sera démarré.

Par défaut : 0
 Plage : 0-1

StartTrappers

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

Le nombre d'instances VMware collector pré-forkées.

Par défaut : 0
 Plage : 0-250

StatsAllowedIP

Une liste d'adresses IP séparées par des virgules, éventuellement en notation CIDR, ou de noms DNS d'instances Zabbix externes. La requête de statistiques sera acceptée uniquement depuis les adresses indiquées ici. 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 considérées comme équivalentes 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
```

Délai d'attente

Indique combien de temps attendre (en secondes) pour établir une connexion et échanger des données avec le serveur Zabbix, l'agent, le service web et les vérifications SNMP héritées (numéro 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 à l'exécution
- Requêtes DNS du poller asynchrone
- Réponse pour le heartbeat des vérifications actives
- Récupération des données (valeurs) de l'agent Zabbix à partir 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 pour lesquelles des paramètres de délai d'attente flexible sont configurés dans le frontend (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 le frontend ; les vérifications SNMP héritées utilisent toujours la valeur de délai d'attente du serveur.

Par défaut : 3
 Plage : 1-30

TLSAccept

Quelles connexions entrantes accepter depuis le 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 pré-partagée (PSK)
cert - accepter les connexions avec TLS et un certificat

Obligatoire : oui pour un proxy passif, si les paramètres de certificat TLS ou de PSK sont définis (même pour une connexion *unencrypted*) ; sinon non

TLSCAFile

Le chemin complet d'un fichier contenant les certificats de la ou des autorités de certification (CA) de niveau supérieur 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 proxy ou la chaîne de certificats, utilisé pour les communications chiffrées entre les composants Zabbix.

TLSCipherAll

La chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplace les critères de sélection par défaut de la suite 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

La chaîne de chiffrement pour OpenSSL 1.1.1 ou version plus récente en TLS 1.3. Remplace les critères de sélection par défaut des suites cryptographiques pour le chiffrement basé sur des 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

La chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplace les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur des 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

La chaîne de chiffrement pour OpenSSL 1.1.1 ou version ultérieure en TLS 1.3. Remplace les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur des certificats.

TLSCipherPSK

La chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplace 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

La chaîne de chiffrement pour OpenSSL 1.1.1 ou version plus récente en TLS 1.3. Remplace les critères de sélection par défaut de la suite cryptographique pour le chiffrement basé sur PSK.

Exemple :

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Comment le proxy doit se connecter au serveur Zabbix. Utilisé pour un proxy actif, ignoré pour un proxy passif. Une seule valeur peut être spécifiée :
unencrypted - se connecter sans chiffrement (par défaut)
psk - se connecter en utilisant TLS et une clé pré-partagée (PSK)
cert - se connecter en utilisant TLS et un certificat

Obligatoire : oui pour un proxy actif, si les paramètres de certificat TLS ou de PSK sont définis (même pour une connexion *unencrypted*) ; sinon non

TLSCRLFile

Le chemin complet d'un fichier contenant les certificats révoqués. Ce paramètre est utilisé pour les communications chiffrées entre les composants Zabbix.

TLSKeyFile

Le chemin complet d'un fichier contenant la clé privée du proxy, utilisée pour les communications chiffrées entre les composants Zabbix.

TLSListen

Contrôle TLS sur le socket trapper.

Valeurs prises en charge :

- required - accepter uniquement les connexions TLS

TLSPSKFile

Le chemin complet d'un fichier contenant la clé pré-partagée du proxy, utilisée pour les communications chiffrées avec le serveur Zabbix.

TLSPSKIdentity

La chaîne d'identité de clé pré-partagée, utilisée pour les communications chiffrées avec le serveur Zabbix.

TLSServerCertIssuer

L'émetteur autorisé du certificat du serveur.

TLSServerCertSubject

Le sujet autorisé du certificat du serveur.

TmpDir

Le répertoire temporaire.

Par défaut : `/tmp`

TrapperTimeout

Spécifie le délai d'expiration en secondes pour :
 - la récupération des données de configuration depuis le serveur Zabbix ;
 - l'exécution de scripts globaux ou de commandes distantes.

Par défaut : 300
 Plage : 1-300

UnavailableDelay

Fréquence à laquelle un hôte est vérifié pour sa disponibilité pendant la période d'indisponibilité, en secondes.

Par défaut : 60
 Plage : 1-3600

UnreachableDelay

Fréquence à laquelle un hôte est vérifié pour sa disponibilité pendant la période d'injoignabilité, en secondes.

Par défaut : 15
 Plage : 1-3600

UnreachablePeriod

Après combien de secondes d'injoignabilité un hôte doit être considéré comme indisponible.

Par défaut : 45
 Plage : 1-3600

Utilisateur

Abandonne les privilèges au profit d'un utilisateur spécifique existant sur le système.
N'a d'effet que si l'exécution se fait en tant que « root » et que AllowRoot est désactivé.

Par défaut : `zabbix`

Coffre-fort

Le fournisseur de coffre-fort :
HashiCorp - Moteur de secrets KV HashiCorp version 2
CyberArk - Fournisseur central d'identifiants CyberArk
Doit correspondre au fournisseur de coffre-fort défini dans l'interface web.

Par défaut : `HashiCorp`

VaultDBPath

Chemin ou requête Vault, selon le Vault, à partir duquel les identifiants de la base de données seront récupérés par clés.

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

Un préfixe personnalisé pour le chemin ou la requête Vault, selon le Vault. Les valeurs par défaut les plus appropriées seront utilisées s'il 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

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-fort CyberArkCCP.

VaultTLSKeyFile

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-fort CyberArkCCP.

VaultToken

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.
C'est une erreur si VaultToken et la variable d'environnement VAULT_TOKEN sont définis en même temps.

Obligatoire : Oui, si Vault est défini sur *HashiCorp* ; sinon non

VaultURL

L'URL HTTP[S] du serveur vault. Le répertoire système des certificats CA sera utilisé si SSLCALocation n'est pas spécifié.

Par défaut : `https://127.0.0.1:8200`

VMwareCacheSize

Taille de la mémoire partagée pour le stockage des 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'y a aucune instance de collecteur VMware configurée pour démarrer.

Par défaut : 8M
 Plage : 256K-2G

VMwareFrequency

Le délai, en secondes, entre les collectes de données à partir d'un seul service VMware.
Ce délai doit être défini sur le plus petit intervalle de mise à jour de tout élément de supervision VMware.

Par défaut : 60
 Plage : 10-86400

VMwarePerfFrequency

Le délai en secondes entre les récupérations des statistiques des compteurs de performance à partir d'un seul service VMware.
Ce délai doit être défini sur le plus petit intervalle de mise à jour de tout élément de supervision VMware qui utilise des compteurs de performance VMware.

Par défaut : 60
 Plage : 10-86400

VMwareTimeout

Le nombre maximal de secondes pendant lesquelles un collecteur VMware attendra une réponse du service VMware (vCenter ou hyperviseur ESX).

Par défaut : 10
 Plage : 1-300

WebDriverURL

URL HTTP[S] de l'interface WebDriver.

Exemple (utilisé avec le serveur autonome Selenium WebDriver) :

```
WebDriverURL=http://localhost:4444
```

Notes de bas de page

[1] DBSocket et DBPort s'excluent mutuellement dans la configuration du proxy. Spécifiez-en un seul, ou laissez les deux non définis.

3 Zabbix agent (UNIX)

Aperçu

Les paramètres pris en charge par le fichier de configuration de l'agent Zabbix (zabbix_agentd.conf) sont répertoriés dans cette section.

Les paramètres sont listés sans informations supplémentaires. Cliquez sur un paramètre pour voir tous les détails.

Paramètre	Description
Alias	Définit un alias pour une clé d'élément.
AllowKey	Autorise l'exécution des clés d'élément correspondant à un modèle.
AllowRoot	Autorise l'agent à s'exécuter en tant que 'root'.
BufferSend	Ne pas conserver les données plus de N secondes dans le tampon.
BufferSize	Le nombre maximal de valeurs dans le tampon mémoire.
DebugLevel	Le niveau de débogage.
DenyKey	Refuse l'exécution des clés d'élément correspondant à un modèle.
EnableRemoteCommands	Indique si les commandes distantes du serveur Zabbix sont autorisées.
HeartbeatFrequency	La fréquence des messages de pulsation en secondes.
HostInterface	Paramètre facultatif qui définit l'interface de l'hôte.
HostInterfaceItem	Paramètre facultatif qui définit un élément utilisé pour obtenir l'interface de l'hôte.
HostMetadata	Paramètre facultatif qui définit les métadonnées de l'hôte.
HostMetadataItem	Paramètre facultatif qui définit un élément de l'agent Zabbix utilisé pour obtenir les métadonnées de l'hôte.
Hostname	Paramètre facultatif qui définit le nom d'hôte.
HostnameItem	Paramètre facultatif qui définit un élément de l'agent Zabbix utilisé pour obtenir le nom d'hôte.
Include	Vous pouvez inclure des fichiers individuels ou tous les fichiers d'un répertoire dans le fichier de configuration.
ListenBacklog	Le nombre maximal de connexions en attente dans la file TCP.
ListenIP	Une liste d'adresses IP séparées par des virgules sur lesquelles l'agent doit écouter.
ListenPort	L'agent écoutera sur ce port les connexions provenant du serveur.
LoadModule	Le module à charger au démarrage de l'agent.
LoadModulePath	Le chemin complet vers l'emplacement des modules de l'agent.
LogFile	Le nom du fichier journal.
LogFileSize	La taille maximale du fichier journal.
LogRemoteCommands	Active la journalisation des commandes shell exécutées en tant qu'avertissements.
LogType	Le type de sortie du journal.
MaxLinesPerSecond	Le nombre maximal de nouvelles lignes que l'agent enverra par seconde au serveur ou proxy Zabbix lors du traitement des contrôles actifs 'log' et 'logrt'.
PidFile	Le nom du fichier PID.
RefreshActiveChecks	À quelle fréquence la liste des contrôles actifs est actualisée.
Server	Une liste d'adresses IP séparées par des virgules, éventuellement en notation CIDR, ou de noms DNS des serveurs Zabbix et des proxies Zabbix.
ServerActive	L'adresse du serveur/proxy Zabbix ou la configuration du cluster à partir de laquelle obtenir les contrôles actifs.
SourceIP	L'adresse IP source.
StartAgents	Le nombre d'instances pré-forkées de zabbix_agentd qui traitent les contrôles passifs.
Timeout	Spécifie combien de temps attendre (en secondes) pour établir une connexion et échanger des données avec le proxy ou le serveur Zabbix.
TLSAccept	Quels types de connexions entrantes accepter.
TLSCAFile	Le chemin complet d'un fichier contenant les certificats de la ou des autorités de certification 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 de l'agent ou la chaîne de certificats, utilisé pour les communications chiffrées entre les composants Zabbix.

Paramètre	Description
TLSCipherAll	La chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplace les critères de sélection par défaut de la suite cryptographique pour le chiffrement basé sur certificat et PSK.
TLSCipherAll13	La chaîne de chiffrement pour OpenSSL 1.1.1 ou version ultérieure en TLS 1.3. Remplace les critères de sélection par défaut de la suite cryptographique pour le chiffrement basé sur certificat et PSK.
TLSCipherCert	La chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplace les critères de sélection par défaut de la suite cryptographique pour le chiffrement basé sur certificat.
TLSCipherCert13	La chaîne de chiffrement pour OpenSSL 1.1.1 ou version ultérieure en TLS 1.3. Remplace les critères de sélection par défaut de la suite cryptographique pour le chiffrement basé sur certificat.
TLSCipherPSK	La chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplace les critères de sélection par défaut de la suite cryptographique pour le chiffrement basé sur PSK.
TLSCipherPSK13	La chaîne de chiffrement pour OpenSSL 1.1.1 ou version ultérieure en TLS 1.3. Remplace les critères de sélection par défaut de la suite cryptographique pour le chiffrement basé sur PSK.
TLSConnect	Comment l'agent doit se connecter au serveur ou proxy Zabbix.
TLSCRLFile	Le chemin complet d'un fichier contenant les certificats révoqués. Ce paramètre est utilisé pour les communications chiffrées entre les composants Zabbix.
TLSKeyFile	Le chemin complet d'un fichier contenant la clé privée de l'agent, utilisé pour les communications chiffrées entre les composants Zabbix.
TLSPSKFile	Le chemin complet d'un fichier contenant la clé pré-partagée de l'agent, utilisé pour les communications chiffrées avec le serveur Zabbix.
TLSPSKIdentity	La chaîne d'identité de la clé pré-partagée, utilisée pour les communications chiffrées avec le serveur Zabbix.
TLSServerCertIssuer	L'émetteur autorisé du certificat du serveur (proxy).
TLSServerCertSubject	Le sujet autorisé du certificat du serveur (proxy).
UnsafeUserParameters	Autorise le passage de tous les caractères dans les arguments des paramètres définis par l'utilisateur.
User	Abandonne les privilèges au profit d'un utilisateur spécifique existant sur le système.
UserParameter	Paramètre défini par l'utilisateur à surveiller.
UserParameterDir	Le chemin de recherche par défaut pour les commandes UserParameter.

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 démon, et non les valeurs des fichiers de configuration fournis.
- Les valeurs prennent en charge les variables d'environnement.
- Zabbix ne prend en charge que 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 des paramètres

Alias

Définit 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 référencer la même clé d'élément.
 Les alias peuvent être utilisés dans *HostMetadataItem* mais pas dans le paramètre *HostnameItem*.

Exemple 1 : Récupération de l'identifiant 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 des paramètres personnalisés.

```
Alias=cpu.util:system.cpu.util
Alias=cpu.util[*]:system.cpu.util[*]
```

Cela 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 désormais 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 motif. Le motif de clé est une expression générique qui prend en charge le caractère « * » pour correspondre à n'importe quel nombre de caractères quelconques.
Plusieurs règles de correspondance de clés peuvent être définies en combinaison avec DenyKey. Les paramètres sont traités un par un selon leur ordre d'apparition. Voir aussi : Restriction des vérifications de l'agent.

AllowRoot

Autorise 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 de basculer vers l'utilisateur « zabbix » à la place. N'a aucun effet s'il est démarré sous un utilisateur ordinaire.

Par défaut : 0
 Valeurs : 0 - ne pas autoriser ; 1 - autoriser

BufferSend

Ne pas conserver les données plus de N secondes dans le tampon.

Par défaut : 5
 Plage : 1-3600

BufferSize

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

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 (produit beaucoup d'informations) ;
5 - débogage étendu (produit encore plus d'informations).

Par défaut : 3
 Plage : 0-5

DenyKey

Refuse l'exécution des clés d'élément qui correspondent à un motif. Le motif de clé est une expression générique qui prend en charge le caractère « * » pour correspondre à n'importe quel nombre de caractères quelconques.
Plusieurs règles de correspondance de clés peuvent être définies en combinaison avec AllowKey. Les paramètres sont traités un par un selon leur ordre d'apparition. Voir aussi : Restriction des vérifications de l'agent.

EnableRemoteCommands

Indique si les commandes distantes provenant du serveur Zabbix sont autorisées. Ce paramètre est **obsolète** ; utilisez plutôt AllowKey=system.run[*] ou DenyKey=system.run[*].
Il s'agit d'un alias interne des paramètres AllowKey/DenyKey selon la valeur :
0 - DenyKey=system.run[*]
1 - AllowKey=system.run[*]

Par défaut : 0
 Valeurs : 0 - ne pas autoriser, 1 - autoriser

HeartbeatFrequency

La fréquence des messages de pulsation en secondes. Utilisée pour surveiller la disponibilité des vérifications actives.
0 - messages de pulsation désactivés.

Par défaut : 60
 Plage : 0-3600

HostInterface

Un paramètre facultatif qui définit l'interface de l'hôte (adresse IP ou nom DNS) utilisée pendant le processus d'autoregistration. Cette valeur sera utilisée pour renseigner l'interface sur l'hôte nouvellement créé et permet de configurer explicitement soit une adresse IP, soit une adresse DNS. Pour plus de détails, voir Using DNS as default interface.

S'il n'est pas défini, 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

Un paramètre facultatif qui définit un élément utilisé pour déterminer l'interface de l'hôte (adresse IP ou nom DNS) pendant le processus d'autoregistration de l'hôte. Cette valeur est utilisée uniquement si HostInterface n'est pas défini. Pour plus de détails, voir Utilisation de DNS comme interface par défaut.

Lors d'une requête d'autoregistration, 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 независимо des paramètres AllowKey/DenyKey.

HostMetadata

Un paramètre facultatif qui définit les métadonnées utilisées pour identifier ou distinguer l'hôte pendant le processus d'enregistrement automatique de l'hôte (agent actif). HostMetadata permet de distinguer les hôtes au-delà du nom d'hôte.

S'il n'est pas défini, la valeur sera obtenue à partir de HostMetadataItem.

L'agent signalera une erreur et ne démarrera pas si la valeur spécifiée dépasse la limite de 2034 octets ou s'il s'agit d'une chaîne non UTF-8. Lorsqu'un paramètre attend une adresse IP ou un nom DNS, les valeurs qui sont en UTF-8 valide mais qui ne sont pas des adresses IP ou des noms DNS valides seront également rejetées et signalées comme invalides.

Plage : 0-2034 octets

HostMetadataItem

Un paramètre facultatif qui définit un élément d'agent Zabbix utilisé pour obtenir les métadonnées de l'hôte. Cette option n'est utilisée que lorsque HostMetadata n'est pas défini.

La valeur de HostMetadataItem est récupérée à chaque tentative d'auto-enregistrement et n'est utilisée que lors du processus d'auto-enregistrement 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 indépendamment des paramètres AllowKey/DenyKey.

Lors d'une requête d'auto-enregistrement, l'agent consignera un message d'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 ne sont pas des adresses IP ou des noms DNS valides seront également rejetées et signalées comme invalides.

Hostname

Une liste de noms d'hôte uniques, sensibles à la casse, séparés par des virgules. Obligatoire pour les contrôles actifs 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 : caractères alphanumériques, '.', ' ', '_' et '-'. Longueur maximale : 128 caractères par nom d'hôte, 2048 caractères pour l'ensemble de la ligne.

Par défaut : défini par HostnameItem

HostnameItem

Un paramètre facultatif qui définit un élément d'agent Zabbix utilisé pour obtenir le nom d'hôte. Cette option n'est utilisée que 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/DenyKey.

Par défaut : `system.hostname`

Include

Vous pouvez inclure des fichiers individuels ou tous les fichiers d'un répertoire dans le fichier de configuration. Pour n'inclure que les fichiers pertinents dans le répertoire spécifié, le caractère générique astérisque est pris en charge pour la correspondance de motifs.
Voir les notes spéciales concernant les limitations.

Exemple :

```
Include=/absolute/path/to/config/files/*.conf
```

ListenBacklog

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 jusqu'au « maximum spécifié par l'implémentation ».

Par défaut : `SOMAXCONN`
 Plage : 0 - INT_MAX

ListenIP

Une liste d'adresses IP séparées par des virgules sur lesquelles l'agent doit écouter.

Par défaut : `0.0.0.0`

ListenPort

L'agent écoutera sur ce port les connexions provenant du serveur.

Par défaut : 10050
 Plage : 1024-32767

LoadModule

Le module à charger au démarrage de l'agent. Les modules sont utilisés pour étendre les fonctionnalités de l'agent. Le module doit être situé 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 est permis d'inclure plusieurs paramètres LoadModule.

LoadModulePath

Le chemin complet vers l'emplacement des modules de l'agent. La valeur par défaut dépend des options de compilation.

LogFile

Le nom du fichier journal.

Obligatoire : Oui, si LogType est défini sur *file* ; sinon non

LogFileSize

La taille maximale d'un fichier journal en Mo.
0 - désactiver la rotation automatique des journaux.
Remarque : si la limite de taille du fichier journal est atteinte et que la rotation du fichier échoue, pour quelque raison que ce soit, le fichier journal existant est tronqué et redémarré à zéro.

Par défaut : 1
 Plage : 0-1024

LogRemoteCommands

Active la journalisation des commandes shell exécutées en tant qu'avertissements. Les commandes seront journalisées 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

LogType

Le type de la 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`

MaxLinesPerSecond

Le nombre maximal de nouvelles lignes que l'agent enverra par seconde au serveur Zabbix ou au proxy lors du traitement des contrôles actifs « log » et « logrt ». La valeur fournie sera remplacée par le paramètre « maxlines », fourni 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

Le nom du fichier PID.

Par défaut : `/tmp/zabbix_agentd.pid`

RefreshActiveChecks

Fréquence de rafraîchissement de la liste des contrôles actifs, en secondes. Notez qu'après un échec du rafraîchissement des contrôles actifs, la prochaine tentative de rafraîchissement sera effectuée dans 60 secondes.

Par défaut : 5
 Plage : 1-86400

Serveur

Une liste d'adresses IP séparées par des virgules, éventuellement en notation CIDR, ou de noms DNS de serveurs Zabbix et de proxys Zabbix. 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 traitées de manière équivalente 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 déconseillées par 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 à 0

ServerActive

L'adresse du serveur/proxy Zabbix ou la configuration du cluster à partir desquels obtenir les contrôles actifs. L'adresse du serveur/proxy est une adresse IP ou un nom DNS, avec un port optionnel séparé par deux-points.
La configuration du cluster est constituée d'une ou plusieurs adresses de membres d'un groupe de serveurs ou de proxies, 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 proxies, il ne faut pas spécifier plus d'un proxy Zabbix pour chaque serveur/cluster Zabbix. Si un proxy Zabbix est spécifié, le serveur/cluster Zabbix correspondant à ce proxy ne doit pas être spécifié.
Plusieurs adresses séparées par des virgules peuvent être fournies afin d'utiliser en parallèle plusieurs serveurs Zabbix indépendants. 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 un port est spécifié pour cet hôte. Si le port n'est pas spécifié, les crochets pour les adresses IPv6 sont facultatifs.
Si ce paramètre n'est pas spécifié, les contrôles actifs sont désactivés.

Exemple pour un proxy Zabbix :

```
ServerActive=127.0.0.1:10051
```

Exemple pour un groupe de proxies 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

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

Le nombre d'instances pré-forkées de zabbix_agentd qui traitent les contrôles passifs. S'il est défini sur 0, les contrôles passifs sont désactivés et l'agent n'écoutera sur aucun port TCP.

Par défaut : 10
 Plage : 0-100

Délai d'attente

Indique combien de temps attendre (en secondes) pour établir une 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 contrôles actifs ;
- la récupération des données de journal via logfile ;
- l'envoi de messages heartbeat ;
- la durée maximale des contrôles `vfs.*` ;
- l'utilisation par les modules de l'agent Zabbix ;
- l'utilisation comme solution de secours dans les scénarios où un serveur ou un proxy antérieur à la version 7.0 envoie des contrôles sans délais d'attente.

Ce délai d'attente **ne sera pas** utilisé pour les contrôles d'agent qui disposent de paramètres de délai d'attente configurables dans le frontend (au niveau global, du proxy ou de l'élément).

Par défaut : 3
 Plage : 1-30

TLSAccept

Quelles connexions entrantes accepter. 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 pré-partagée (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

Le chemin complet du fichier contenant les certificats de la ou des autorités de certification (CA) de niveau supérieur pour la vérification du certificat du pair, utilisé pour les communications chiffrées entre les composants Zabbix.

TLSCertFile

Le chemin complet du fichier contenant le certificat de l'agent ou la chaîne de certificats, utilisé pour les communications chiffrées avec les composants Zabbix.

TLSCipherAll

La chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplace les critères de sélection par défaut de la suite 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

La chaîne de chiffrement pour OpenSSL 1.1.1 ou version ultérieure en TLS 1.3. Remplace les critères de sélection par défaut des suites de chiffrement pour le chiffrement basé sur des 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

La chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplace les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur des 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

La chaîne de chiffrement pour OpenSSL 1.1.1 ou version ultérieure en TLS 1.3. Remplace les critères de sélection par défaut de la suite cryptographique pour le chiffrement basé sur certificat.

TLSCipherPSK

La chaîne de priorité GnuTLS ou la chaîne de chiffrement OpenSSL (TLS 1.2). Remplace 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

La chaîne de chiffrement pour OpenSSL 1.1.1 ou version plus récente en TLS 1.3. Remplace 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

Comment l'agent doit se connecter au serveur Zabbix ou au proxy. Utilisé pour les contrôles actifs. Une seule valeur peut être spécifiée :
unencrypted - se connecter sans chiffrement (par défaut)
psk - se connecter en utilisant TLS et une clé pré-partagée (PSK)
cert - se connecter en utilisant TLS et un certificat

Obligatoire : oui, si les paramètres de certificat TLS ou de PSK sont définis (même pour une connexion *unencrypted*) ; sinon non

TLSCRLFile

Le chemin complet du fichier contenant les certificats révoqués. Ce paramètre est utilisé pour les communications chiffrées entre les composants Zabbix.

TLSKeyFile

Le chemin complet du fichier contenant la clé privée de l'agent, utilisée pour les communications chiffrées entre les composants Zabbix.

TLSPSKFile

Le chemin complet du fichier contenant la clé pré-partagée de l'agent, utilisée pour les communications chiffrées avec le serveur Zabbix.

TLSPSKIdentity

La chaîne d'identité de clé pré-partagée, utilisée pour les communications chiffrées avec le serveur Zabbix.

TLSServerCertIssuer

L'émetteur autorisé du certificat du serveur (proxy).

TLSServerCertSubject

Le sujet du certificat autorisé du serveur (proxy).

UnsafeUserParameters

Autoriser le passage de tous les caractères 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

Abandonner les privilèges au profit d'un utilisateur spécifique existant sur le système.
N'a d'effet que si le processus est exécuté en tant que « root » et que AllowRoot est désactivé.

Par défaut : `zabbix`

UserParameter

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 une fin de ligne. Les commandes shell peuvent avoir des chemins relatifs si le paramètre UserParameterDir est spécifié.

Exemple :

```
UserParameter=system.test,who|wc -l
UserParameter=check_cpu,./custom_script.sh
```

UserParameterDir

Le chemin de recherche par défaut pour les commandes UserParameter. S'il est utilisé, l'agent changera son répertoire de travail pour celui spécifié ici avant d'exécuter une commande. Ainsi, les commandes UserParameter peuvent avoir 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 contrôles actifs et passifs à partir de la version 2.0.0

4 Zabbix agent 2 (UNIX)

Aperçu

Zabbix agent 2 est une nouvelle génération d'agent Zabbix et peut être utilisé à la place de Zabbix agent.

Les paramètres pris en charge par le fichier de configuration de Zabbix agent 2 (zabbix_agent2.conf) sont répertoriés dans cette section.

Les paramètres sont listés sans informations supplémentaires. Cliquez sur un paramètre pour afficher tous les détails.

Parameter	Description
Alias	Définit un alias pour une clé d'élément.
AllowKey	Autorise l'exécution des clés d'élément correspondant à un modèle.
BufferSend	Ne pas conserver les données plus de N secondes dans le tampon.
BufferSize	Le nombre maximal de valeurs dans le tampon mémoire.
ControlSocket	Le socket de contrôle, utilisé pour envoyer des commandes d'exécution avec l'option '-R'.
DebugLevel	Le niveau de débogage.
DenyKey	Refuse l'exécution des clés d'élément correspondant à un modèle.
EnablePersistentBuffer	Active l'utilisation d'un stockage persistant local pour les éléments actifs.
ForceActiveChecksOnStart	Exécute les contrôles actifs immédiatement après le redémarrage pour la première configuration reçue.
HeartbeatFrequency	La fréquence des messages de pulsation en secondes.
HostInterface	Paramètre facultatif qui définit l'interface de l'hôte.
HostInterfaceItem	Paramètre facultatif qui définit un élément utilisé pour obtenir l'interface de l'hôte.
HostMetadata	Paramètre facultatif qui définit les métadonnées de l'hôte.
HostMetadataItem	Paramètre facultatif qui définit un élément Zabbix agent utilisé pour obtenir les métadonnées de l'hôte.
Hostname	Paramètre facultatif qui définit le nom de l'hôte.
HostnameItem	Paramètre facultatif qui définit un élément Zabbix agent utilisé pour obtenir le nom de l'hôte.
Include	Vous pouvez inclure des fichiers individuels ou tous les fichiers d'un répertoire dans le fichier de configuration.
ListenIP	Liste d'adresses IP séparées par des virgules sur lesquelles l'agent doit écouter.
ListenPort	L'agent 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.
PersistentBufferFile	Le fichier dans lequel Zabbix agent 2 doit conserver la base de données SQLite.
PersistentBufferPeriod	La période pendant laquelle les données doivent être stockées lorsqu'il n'y a pas de connexion au serveur ou au proxy.
PidFile	Le nom du fichier PID.
Plugins.<PluginName>.System.Capacity	La limite de contrôles par plugin pouvant être exécutés en même temps.
Plugins.Log.MaxLinesPerSecond	Le nombre maximal de nouvelles lignes que l'agent enverra par seconde au serveur Zabbix ou au proxy lors du traitement des contrôles actifs 'log' et 'logrt'.
Plugins.SystemRun.LogRemoteCommands	Activer la journalisation des commandes shell exécutées en tant qu'avertissements.
PluginSocket	Le chemin vers le socket UNIX pour les communications avec les plugins chargeables.
PluginTimeout	Le délai d'attente pour les connexions avec les plugins chargeables, en secondes.
RefreshActiveChecks	Fréquence de rafraîchissement de la liste des contrôles actifs.
Server	Liste d'adresses IP séparées par des virgules, éventuellement en notation CIDR, ou de noms DNS des serveurs Zabbix et des proxies Zabbix.
ServerActive	L'adresse du serveur/proxy Zabbix ou la configuration du cluster à partir de laquelle obtenir les contrôles actifs.
SourceIP	L'adresse IP source.
StatusPort	Si défini, l'agent écoutera sur ce port les requêtes d'état HTTP (http://localhost:<port>/status).
Timeout	Spécifie combien de temps attendre (en secondes) pour établir une connexion et échanger des données avec le proxy ou le serveur Zabbix.
TLSAccept	Quelles connexions entrantes accepter.
TLSCAFile	Le chemin complet d'un fichier contenant les certificats de la ou des autorités de certification 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 de l'agent ou la chaîne de certificats, utilisé pour les communications chiffrées entre les composants Zabbix.
TLSCipherAll	La chaîne de chiffrement OpenSSL (TLS 1.2). Remplace les critères de sélection par défaut de la suite cryptographique pour le chiffrement basé sur certificat et PSK.

Parameter	Description
TLSCipherAll13	La chaîne de chiffrement OpenSSL (TLS 1.3) pour OpenSSL 1.1.1 ou version ultérieure. Remplace les critères de sélection par défaut de la suite cryptographique pour le chiffrement basé sur certificat et PSK.
TLSCipherCert	La chaîne de chiffrement OpenSSL (TLS 1.2). Remplace les critères de sélection par défaut de la suite cryptographique pour le chiffrement basé sur certificat.
TLSCipherCert13	La chaîne de chiffrement OpenSSL (TLS 1.3) pour OpenSSL 1.1.1 ou version ultérieure. Remplace les critères de sélection par défaut de la suite cryptographique pour le chiffrement basé sur certificat.
TLSCipherPSK	La chaîne de chiffrement OpenSSL (TLS 1.2). Remplace les critères de sélection par défaut de la suite cryptographique pour le chiffrement basé sur PSK.
TLSCipherPSK13	La chaîne de chiffrement OpenSSL (TLS 1.3) pour OpenSSL 1.1.1 ou version ultérieure. Remplace les critères de sélection par défaut de la suite cryptographique pour le chiffrement basé sur PSK.
TLSConnect	Comment l'agent doit se connecter au serveur ou au proxy Zabbix.
TLSCRLFile	Le chemin complet d'un fichier contenant les certificats révoqués. Ce paramètre est utilisé pour les communications chiffrées entre les composants Zabbix.
TLSKeyFile	Le chemin complet d'un fichier contenant la clé privée de l'agent, utilisé pour les communications chiffrées entre les composants Zabbix.
TLSPSKFile	Le chemin complet d'un fichier contenant la clé pré-partagée de l'agent, utilisé pour les communications chiffrées avec le serveur Zabbix.
TLSPSKIdentity	La chaîne d'identité de la clé pré-partagée, utilisée pour les communications chiffrées avec le serveur Zabbix.
TLSServerCertIssuer	L'émetteur autorisé du certificat du serveur (proxy).
TLSServerCertSubject	Le sujet autorisé du certificat du serveur (proxy).
UnsafeUserParameters	Autorise le passage de tous les caractères dans les arguments des paramètres définis par l'utilisateur.
UserParameter	Paramètre défini par l'utilisateur à surveiller.
UserParameterDir	Le chemin de recherche par défaut pour les commandes UserParameter.

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 les valeurs des fichiers de configuration fournis.
- Les valeurs prennent en charge les variables d'environnement.
- Zabbix ne prend en charge que 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 des paramètres

Alias

Définit 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 référencer la même clé d'élément.
 Les alias peuvent être utilisés dans *HostMetadataItem*, mais pas dans le paramètre *HostnameItem*.

Exemple 1 : Récupération de l'identifiant de l'utilisateur « zabbix ».

```
Alias=zabbix.userid:vfs.file.regexp[/etc/passwd,"^zabbix:.:([0-9]+)",,,,\1]
```

La clé abrégée **zabbix.userid** peut désormais être utilisée pour récupérer des 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 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 désormais 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

Autoriser l'exécution des clés d'élément qui correspondent à un motif. Le motif de clé est une expression générique qui prend en charge le caractère « * » pour correspondre à n'importe quel nombre de caractères quelconques.
Plusieurs règles de correspondance de clés peuvent être définies en combinaison avec DenyKey. Les paramètres sont traités un par un selon leur ordre d'apparition. Voir aussi : Restriction des vérifications de l'agent.

BufferSend

L'intervalle de temps en secondes qui détermine la fréquence à laquelle les valeurs sont envoyées depuis le tampon vers le serveur Zabbix. Notez que si le tampon est plein, les données seront envoyées plus tôt.

Par défaut : 5
 Plage : 1-3600

BufferSize

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

Le socket de contrôle, utilisé pour envoyer des commandes d'exécution avec l'option « -R ».

Par défaut : `/tmp/agent.sock`

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 (produit beaucoup d'informations) ;
5 - débogage étendu (produit encore plus d'informations).

Par défaut : 3
 Plage : 0-5

DenyKey

Refuse l'exécution des clés d'élément qui correspondent à un motif. Le motif de clé est une expression générique qui prend en charge le caractère "*" pour correspondre à n'importe quel nombre de caractères quelconques.
Plusieurs règles de correspondance de clés peuvent être définies en combinaison avec AllowKey. Les paramètres sont traités un par un selon leur ordre d'apparition. Voir aussi : Restriction des vérifications de l'agent.

EnablePersistentBuffer

Active 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

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

La fréquence des messages de pulsation en secondes. Utilisée pour surveiller la disponibilité des contrôles actifs.
0 - messages de pulsation désactivés.

Par défaut : 60
 Plage : 0-3600

HostInterface

Paramètre facultatif qui définit l'interface de l'hôte (adresse IP ou nom DNS) utilisée pendant le processus d'autoregistration de l'hôte. Cette valeur sera utilisée pour renseigner l'interface dans l'hôte nouvellement créé et permet de configurer explicitement soit une adresse IP, soit une adresse DNS. Pour plus de détails, voir Utilisation de DNS comme interface par défaut.

S'il n'est pas défini, 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

Un paramètre facultatif qui définit un élément utilisé pour déterminer l'interface de l'hôte (adresse IP ou nom DNS) pendant le processus d'autoregistration de l'hôte. Cette valeur est utilisée uniquement si HostInterface n'est pas défini. Pour plus de détails, voir Utilisation de DNS comme interface par défaut.

Lors d'une requête d'autoregistration, 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 независимо des paramètres AllowKey/DenyKey.

HostMetadata

Un paramètre facultatif qui définit les métadonnées utilisées pour identifier ou distinguer l'hôte pendant le processus d'enregistrement automatique de l'hôte. HostMetadata permet de distinguer les hôtes au-delà du nom d'hôte.

S'il n'est pas défini, la valeur sera obtenue à partir de HostMetadataItem.

L'agent signalera une erreur et ne démarrera pas si la valeur spécifiée dépasse la limite de 2034 octets ou n'est pas une chaîne UTF-8 valide. Lorsqu'un paramètre attend une adresse IP ou un nom DNS, les valeurs qui sont valides en UTF-8 mais ne sont pas des adresses IP ou des noms DNS valides seront également rejetées et signalées comme invalides.

Plage : 0-2034 octets

HostMetadataItem

Un paramètre facultatif qui définit un élément d'agent Zabbix utilisé pour obtenir les métadonnées de l'hôte. Cette option n'est utilisée que lorsque HostMetadata n'est pas défini.

La valeur de HostMetadataItem est récupérée à chaque tentative d'auto-enregistrement et n'est utilisée que lors du processus d'auto-enregistrement 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 независимо des paramètres AllowKey/DenyKey.

Lors d'une requête d'auto-enregistrement, l'agent consignera un message d'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 ne sont pas des adresses IP ou des noms DNS valides seront également rejetées et signalées comme invalides.

Hostname

Une liste de noms d'hôte uniques, sensibles à la casse, séparés par des virgules. Obligatoire 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

Un paramètre facultatif qui définit un élément utilisé pour obtenir le nom d'hôte. Cette option n'est utilisée que 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/DenyKey.

Par défaut : `system.hostname`

Include

Vous pouvez inclure des fichiers individuels ou tous les fichiers d'un répertoire dans le fichier de configuration. Lors de l'installation, Zabbix créera le répertoire d'inclusion dans /usr/local/etc, sauf s'il a été modifié au moment de la compilation. Le chemin peut être relatif à l'emplacement du fichier *zabbix_agent2.conf*.
Pour n'inclure que les fichiers pertinents dans le répertoire spécifié, le caractère générique astérisque est pris en charge pour la correspondance de motifs.
Voir les notes spéciales concernant les limitations.

Exemple :

```
Include=/absolute/path/to/config/files/*.conf
```

ListenIP

Une liste d'adresses IP séparées par des virgules sur lesquelles l'agent doit écouter. La première adresse IP est envoyée au serveur Zabbix, lors de la connexion à celui-ci, afin de récupérer la liste des contrôles actifs.

Par défaut : `0.0.0.0`

ListenPort

L’agent écoutera sur ce port les connexions provenant du serveur.

Par défaut : 10050
 Plage : 1024-32767

LogFile

Le nom du fichier journal.

Par défaut : `/tmp/zabbix_agent2.log`
 Obligatoire : Oui, si LogType est défini sur *file* ; sinon non

LogFileSize

La taille maximale d’un fichier journal en Mo.
0 - désactiver la rotation automatique des journaux.
Remarque : si la limite de taille du fichier journal est atteinte et que la rotation du fichier échoue, pour quelque raison que ce soit, le fichier journal existant est tronqué et recommencé à zéro.

Par défaut : 1
 Plage : 0-1024

LogType

Le type de la 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`

PersistentBufferFile

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

La période pendant laquelle les données doivent être stockées 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 n’est utilisé que si le tampon persistant est activé (*EnablePersistentBuffer=1*).

Par défaut : `1h`
 Plage : 1m-365d

PidFile

Le nom du fichier PID.

Par défaut : `/tmp/zabbix_agent2.pid`

Plugins.<PluginName>.System.Capacity

La limite de contrôles par plugin `<PluginName>` pouvant être exécutés en même temps.

Par défaut : 1000 Plage : 1-1000

Plugins.Log.MaxLinesPerSecond

Le nombre maximal de nouvelles lignes que l’agent enverra par seconde au serveur ou au proxy Zabbix lors du traitement des contrôles actifs ’log’ et ’logrt’. La valeur fournie sera remplacée par le paramètre ’maxlines’, fourni 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.

Par défaut : 20
 Plage : 1-1000

Plugins.SystemRun.LogRemoteCommands

Activez 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

PluginSocket

Le chemin vers le socket UNIX pour les communications avec les plugins chargeables.

Par défaut : `/tmp/agent.plugin.sock`

PluginTimeout

Le délai d’expiration des connexions avec les plugins chargeables, en secondes.

Par défaut : `Timeout`
 Plage : 1-30

RefreshActiveChecks

Fréquence de rafraîchissement de la liste des contrôles actifs, en secondes. Notez qu'après un échec du rafraîchissement des contrôles actifs, la prochaine tentative de rafraîchissement sera effectuée dans 60 secondes.

Par défaut : 5
 Plage : 1-86400

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. 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 traitées de manière identique 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

L'adresse du serveur/proxy Zabbix ou la configuration du cluster pour obtenir les contrôles actifs. L'adresse du serveur/proxy est une adresse IP ou un nom DNS, avec un port optionnel séparé par deux-points.
La configuration du cluster est constituée d'une ou plusieurs adresses de membres du groupe de serveurs ou de proxys, séparées par des points-virgules. Plusieurs serveurs/clusters Zabbix et proxys Zabbix peuvent être spécifiés, séparés par des virgules. Sauf en cas d'utilisation de groupes de proxys, il ne faut pas spécifier plus d'un proxy Zabbix pour chaque serveur/cluster Zabbix. Si un proxy Zabbix est spécifié, le serveur/cluster Zabbix correspondant à ce proxy ne doit pas être spécifié.
Plusieurs adresses séparées par des virgules peuvent être fournies pour utiliser en parallèle plusieurs serveurs Zabbix indépendants. 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 un port est spécifié pour cet hôte. Si le port n'est pas spécifié, les crochets pour les adresses IPv6 sont facultatifs.
Si ce paramètre n'est pas spécifié, les contrôles actifs sont désactivés.

Exemple pour un proxy Zabbix :

```
ServerActive=127.0.0.1:10051
```

Exemple pour un groupe de proxys 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 de haute disponibilité :

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

Exemple de haute disponibilité avec deux clusters et un serveur :

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

SourceIP

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

Si ce paramètre est défini, l'agent écoutera sur ce port les requêtes de statut HTTP (http://localhost:<port>/status).

Plage : 1024-32767

Délai d'attente

Spécifie combien de temps attendre (en secondes) pour établir une 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 contrôles actifs ;
- la récupération des données de journal via logfile ;
- l'envoi de messages heartbeat ;
- la durée maximale des contrôles `vfs.*` ;

- l'utilisation comme solution de repli dans les cas où un serveur ou un proxy antérieur à la version 7.0 envoie des contrôles sans délais d'attente.

Ce délai d'attente **ne sera pas** utilisé pour les contrôles d'agent qui disposent de paramètres de délai d'attente configurables dans l'interface web (au niveau global, du proxy ou de l'élément).

Par défaut : 3
 Plage : 1-30

TLSAccept

Les connexions entrantes à accepter. Utilisé pour les contrôles passifs. 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 pré-partagée (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

Le chemin complet du fichier contenant les certificats de la ou des autorités de certification (CA) de niveau supérieur pour la vérification du certificat du pair, utilisé pour les communications chiffrées entre les composants Zabbix.

TLSCertFile

Le chemin complet du fichier contenant le certificat de l'agent ou la chaîne de certificats, utilisé pour les communications chiffrées avec les composants Zabbix.

TLSCipherAll

La chaîne de chiffrement OpenSSL (TLS 1.2). Remplace les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur des certificats et sur PSK.

Exemple :

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherAll13

La chaîne de chiffrement OpenSSL (TLS 1.3) pour OpenSSL 1.1.1 ou version ultérieure. Remplace les critères de sélection par défaut des suites cryptographiques 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
```

TLSCipherCert

La chaîne de chiffrement OpenSSL (TLS 1.2). Remplace les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur un certificat.

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

La chaîne de chiffrement OpenSSL (TLS 1.3) pour OpenSSL 1.1.1 ou version ultérieure. Remplace les critères de sélection par défaut de la suite cryptographique pour le chiffrement basé sur un certificat.

Notez que ce paramètre ne peut pas être utilisé avec `TLSAccept=cert,psk` ; pour une connexion par certificat (`TLSConnect=cert`), utilisez plutôt TLSCipherAll13.

TLSCipherPSK

La chaîne de chiffrement OpenSSL (TLS 1.2). Remplace les critères de sélection par défaut de la suite de chiffrement pour le chiffrement basé sur PSK.

Exemple :

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

La chaîne de chiffrement OpenSSL (TLS 1.3) pour OpenSSL 1.1.1 ou version plus récente. Remplace les critères de sélection par défaut de la suite cryptographique pour le chiffrement basé sur PSK.

Exemple :

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Comment l'agent doit se connecter au serveur ou au proxy Zabbix. Utilisé pour les contrôles actifs. Une seule valeur peut être spécifiée :
unencrypted - se connecter sans chiffrement (par défaut)
psk - se connecter à l'aide de TLS et d'une clé pré-partagée (PSK)
cert - se connecter à l'aide de TLS et d'un certificat

Obligatoire : oui, si les paramètres de certificat TLS ou de PSK sont définis (même pour une connexion *unencrypted*) ; sinon non

TLSCRLFile

Le chemin complet du fichier contenant les certificats révoqués. Ce paramètre est utilisé pour les communications chiffrées entre les composants Zabbix.

TLSKeyFile

Le chemin complet du fichier contenant la clé privée de l'agent, utilisée pour les communications chiffrées entre les composants Zabbix.

TLSPSKFile

Le chemin complet du fichier contenant la clé pré-partagée de l'agent, utilisée pour les communications chiffrées avec le serveur Zabbix.

TLSPSKIdentity

La chaîne d'identité de clé pré-partagée, utilisée pour les communications chiffrées avec le serveur Zabbix.

TLSServerCertIssuer

L'émetteur autorisé du certificat du serveur (proxy).

TLSServerCertSubject

Le sujet du certificat autorisé du serveur (proxy).

UnsafeUserParameters

Autoriser le passage de tous les caractères 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

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 une fin de ligne.
Les commandes shell peuvent avoir des chemins relatifs si le paramètre UserParameterDir est spécifié.

Exemple :

```
UserParameter=system.test,who|wc -l
UserParameter=check_cpu,./custom_script.sh
```

UserParameterDir

Le chemin de recherche par défaut pour les commandes UserParameter. S'il est utilisé, l'agent changera son répertoire de travail pour celui spécifié ici avant d'exécuter une commande. Ainsi, les commandes UserParameter peuvent avoir un préfixe relatif `./` au lieu d'un chemin complet.
Une seule entrée est autorisée.

Exemple :

```
UserParameterDir=/opt/myscripts
```

5 Agent Zabbix (Windows)

Aperçu

Les paramètres pris en charge par le fichier de configuration de l'agent Zabbix Windows (zabbix_agentd.conf) sont répertoriés dans cette section.

Les paramètres sont listés sans informations supplémentaires. Cliquez sur un paramètre pour en voir tous les détails.

Parameter	Description
Alias	Définit un alias pour une clé d'élément.
AllowKey	Autorise l'exécution des clés d'élément correspondant à un motif.
BufferSend	Ne pas conserver les données plus de N secondes dans le tampon.
BufferSize	Le nombre maximal de valeurs dans le tampon mémoire.
DebugLevel	Le niveau de débogage.
DenyKey	Refuse l'exécution des clés d'élément correspondant à un motif.
EnableRemoteCommands	Indique si les commandes distantes provenant du serveur Zabbix sont autorisées.
HeartbeatFrequency	La fréquence des messages de pulsation en secondes.
HostInterface	Paramètre facultatif qui définit l'interface de l'hôte.
HostInterfaceItem	Paramètre facultatif qui définit un élément utilisé pour obtenir l'interface de l'hôte.
HostMetadata	Paramètre facultatif qui définit les métadonnées de l'hôte.
HostMetadataItem	Paramètre facultatif qui définit un élément de l'agent Zabbix utilisé pour obtenir les métadonnées de l'hôte.
Hostname	Paramètre facultatif qui définit le nom d'hôte.
HostnameItem	Paramètre facultatif qui définit un élément de l'agent Zabbix utilisé pour obtenir le nom d'hôte.
Include	Vous pouvez inclure des fichiers individuels ou tous les fichiers d'un répertoire dans le fichier de configuration.
ListenBacklog	Le nombre maximal de connexions en attente dans la file TCP.
ListenIP	Une liste d'adresses IP séparées par des virgules sur lesquelles l'agent doit écouter.
ListenPort	L'agent écoutera sur ce port les connexions provenant du serveur.
LogFile	Le nom du fichier journal.
LogFileSize	La taille maximale du fichier journal.
LogRemoteCommands	Active la journalisation des commandes shell exécutées en tant qu'avertissements.
LogType	Le type de sortie du journal.
MaxLinesPerSecond	Le nombre maximal de nouvelles lignes que l'agent enverra par seconde au serveur ou au proxy Zabbix lors du traitement des contrôles actifs 'log' et 'logrt'.
PerfCounter	Définit un nouveau paramètre <parameter_name> qui correspond à la valeur moyenne du compteur de performances système <perf_counter_path> pour la période <period> spécifiée (en secondes).
PerfCounterEn	Définit un nouveau paramètre <parameter_name> qui correspond à la valeur moyenne du compteur de performances système <perf_counter_path> pour la période <period> spécifiée (en secondes). Par rapport à PerfCounter, les chemins perfcounter doivent être en anglais.
RefreshActiveChecks	Fréquence de rafraîchissement de la liste des contrôles actifs.
Server	Une liste d'adresses IP séparées par des virgules, éventuellement en notation CIDR, ou de noms DNS des serveurs Zabbix et des proxies Zabbix.
ServerActive	L'adresse du serveur/proxy Zabbix ou la configuration du cluster à partir de laquelle obtenir les contrôles actifs.
SourceIP	L'adresse IP source.
StartAgents	Le nombre d'instances pré-forkées de zabbix_agentd qui traitent les contrôles passifs.
Timeout	Spécifie combien de temps attendre (en secondes) pour établir une connexion et échanger des données avec le proxy ou le serveur Zabbix.
TLSAccept	Quels types de connexions entrantes accepter.
TLSCAFile	Le chemin complet d'un fichier contenant les certificats de la ou des autorités de certification 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 de l'agent ou la chaîne de certificats, utilisé pour les communications chiffrées entre les composants Zabbix.
TLSConnect	Comment l'agent doit se connecter au serveur ou au proxy Zabbix.
TLSCRLFile	Le chemin complet d'un fichier contenant les certificats révoqués. Ce paramètre est utilisé pour les communications chiffrées entre les composants Zabbix.
TLSKeyFile	Le chemin complet d'un fichier contenant la clé privée de l'agent, utilisé pour les communications chiffrées entre les composants Zabbix.
TLSPSKFile	Le chemin complet d'un fichier contenant la clé pré-partagée de l'agent, utilisé pour les communications chiffrées avec le serveur Zabbix.
TLSPSKIdentity	La chaîne d'identité de la clé pré-partagée, utilisée pour les communications chiffrées avec le serveur Zabbix.
TLSServerCertIssuer	L'émetteur autorisé du certificat du serveur (proxy).
TLSServerCertSubject	Le sujet autorisé du certificat du serveur (proxy).
UnsafeUserParameters	Autorise le passage de tous les caractères dans les arguments des paramètres définis par l'utilisateur.
UserParameter	Paramètre défini par l'utilisateur à surveiller.

Parameter	Description
UserParameterDir	Le chemin de recherche par défaut pour les commandes UserParameter.

Tous les paramètres sont facultatifs, sauf indication explicite qu'un paramètre est obligatoire.

Notez que :

- Les valeurs par défaut reflètent celles du démon, et non les valeurs des fichiers de configuration fournis.
- Les valeurs prennent en charge les variables d'environnement.
- 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 des paramètres

Alias

Définit 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 référencer la même clé d'élément.
 Les alias peuvent être utilisés dans *HostMetadataItem* mais pas dans le paramètre *HostnameItem* ni *PerfCounter*.

Exemple 1 : Récupération du pourcentage d'utilisation du fichier d'échange depuis le serveur.

```
Alias=pg_usage:perf_counter[\Paging File(_Total)\% Usage]
```

La clé abrégée **pg_usage** peut désormais être utilisée pour récupérer les données.

Exemple 2 : Obtention de la charge CPU avec les paramètres par défaut et des paramètres personnalisés.

```
Alias=cpu.load:system.cpu.load
Alias=cpu.load[*]:system.cpu.load[*]
```

Cela permet d'utiliser la clé **cpu.load** pour obtenir la charge CPU avec les paramètres par défaut, ainsi que **cpu.load[percpu,avg15]** pour obtenir des données spécifiques sur la charge CPU.

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 désormais 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

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 quelconques.
Plusieurs règles de correspondance de clés peuvent être définies en combinaison avec DenyKey. Les paramètres sont traités un par un selon leur ordre d'apparition. Voir aussi : Restriction des contrôles de l'agent.

BufferSend

Ne pas conserver les données plus de N secondes dans le tampon.

Par défaut : 5
 Plage : 1-3600

BufferSize

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

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 (produit beaucoup d'informations) ;
5 - débogage étendu (produit encore plus d'informations).

Par défaut : 3
 Plage : 0-5

DenyKey

Refuse l'exécution des clés d'élément qui correspondent à un motif. Le motif de clé est une expression générique qui prend en charge le caractère « * » pour correspondre à n'importe quel nombre de caractères quelconques.
Plusieurs règles de

correspondance de clés peuvent être définies en combinaison avec AllowKey. Les paramètres sont traités un par un selon leur ordre d'apparition. Voir aussi : Restriction des vérifications de l'agent.

EnableRemoteCommands

Indique si les commandes distantes provenant du serveur Zabbix sont autorisées. Ce paramètre est **obsolète** ; utilisez plutôt AllowKey=system.run[*] ou DenyKey=system.run[*].
Il s'agit d'un alias interne des paramètres AllowKey/DenyKey selon la valeur :
0 - DenyKey=system.run[*]
1 - AllowKey=system.run[*]

Par défaut : 0
 Valeurs : 0 - ne pas autoriser, 1 - autoriser

HeartbeatFrequency

La fréquence des messages de pulsation en secondes. Utilisée pour surveiller la disponibilité des contrôles actifs.
0 - messages de pulsation désactivés.

Par défaut : 60
 Plage : 0-3600

HostInterface

Paramètre facultatif qui définit l'interface hôte (adresse IP ou nom DNS) utilisée pendant le processus d'enregistrement automatique. Cette valeur sera utilisée pour renseigner l'interface dans l'hôte nouvellement créé et permet de configurer explicitement une adresse IP ou DNS. Pour plus de détails, voir Utilisation de DNS comme interface par défaut.

S'il n'est pas défini, la valeur sera obtenue à partir de HostInterfaceItem.

L'agent émettra une erreur et ne démarrera pas si la valeur dépasse la limite de 255 caractères.

Plage : 0-255 caractères

HostInterfaceItem

Un paramètre facultatif qui définit un élément utilisé pour déterminer l'interface de l'hôte (adresse IP ou nom DNS) pendant le processus d'autoregistration de l'hôte. Cette valeur est utilisée uniquement si HostInterface n'est pas défini. Pour plus de détails, voir Utilisation de DNS comme interface par défaut.

Lors d'une requête d'autoregistration, 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 независимо des paramètres AllowKey/DenyKey.

HostMetadata

Un paramètre facultatif qui définit les métadonnées utilisées pour identifier ou distinguer l'hôte pendant le processus d'enregistrement automatique de l'hôte (agent actif). HostMetadata permet de distinguer les hôtes au-delà du nom d'hôte.

S'il n'est pas défini, la valeur sera obtenue à partir de HostMetadataItem.

L'agent émettra une erreur et ne démarrera pas si la valeur spécifiée dépasse la limite de 2034 octets ou si elle n'est pas une chaîne UTF-8 valide. Lorsqu'un paramètre attend une adresse IP ou un nom DNS, les valeurs qui sont en UTF-8 valide mais ne sont pas des adresses IP ou des noms DNS valides seront également rejetées et signalées comme invalides.

Les métadonnées sur plusieurs lignes ne sont pas prises en charge — la sortie sera tronquée au premier saut de ligne.

Plage : 0-2034 octets

HostMetadataItem

Un paramètre facultatif qui définit un élément d'agent Zabbix utilisé pour obtenir les métadonnées de l'hôte. Cette option n'est utilisée que lorsque HostMetadata n'est pas défini.

La valeur de HostMetadataItem est récupérée à chaque tentative d'auto-enregistrement et n'est utilisée que lors du processus d'auto-enregistrement 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 независимоamment des paramètres AllowKey/DenyKey.

Lors d'une requête d'auto-enregistrement, l'agent consignera un message d'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 ne sont pas des adresses IP ou des noms DNS valides seront également rejetées et signalées comme invalides.

Les métadonnées sur plusieurs lignes ne sont pas prises en charge — la sortie sera tronquée au premier saut de ligne.

Nom d'hôte

Une liste de noms d'hôte uniques, sensibles à la casse, séparés par des virgules. Obligatoire 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

Un paramètre facultatif qui définit un élément d'agent Zabbix utilisé pour obtenir le nom d'hôte. Cette option n'est utilisée que lorsque Hostname n'est pas défini. Les paramètres utilisateur, les compteurs de performance 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/DenyKey.
Voir aussi une description plus détaillée.

Par défaut : `system.hostname`

Include

Vous pouvez inclure des fichiers individuels ou tous les fichiers d'un répertoire 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 Windows MSI ; situé dans le dossier spécifié lors de l'installation si l'agent Zabbix est installé sous forme d'archive zip). Tous les fichiers inclus doivent avoir une syntaxe correcte, sinon l'agent ne démarrera pas.
Pour n'inclure que les fichiers pertinents dans le répertoire spécifié, le caractère générique astérisque est pris en charge pour la correspondance de motifs.
Consultez les notes spéciales concernant les limitations.

Exemple :

```
Include=C:\Program Files\Zabbix Agent\zabbix_agentd.d\*.conf
```

ListenBacklog

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 jusqu'au « maximum spécifié par l'implémentation ».

Par défaut : `SOMAXCONN`
 Plage : 0 - INT_MAX

ListenIP

Une liste d'adresses IP séparées par des virgules sur lesquelles l'agent doit écouter.

Par défaut : `0.0.0.0`

ListenPort

L'agent écoutera sur ce port les connexions provenant du serveur.

Par défaut : 10050
 Plage : 1024-32767

LogFile

Le nom du fichier journal de l'agent.

Par défaut : `c:\zabbix_agentd.log`
 Obligatoire : Oui, si LogType est défini sur *file* ; sinon non

LogFileSize

La taille maximale d'un fichier journal en Mo.
0 - désactiver la rotation automatique des journaux.
Remarque : si la limite de taille du fichier journal est atteinte et que la rotation du fichier échoue, pour quelque raison que ce soit, le fichier journal existant est tronqué et redémarré à zéro.

Par défaut : 1
 Plage : 0-1024

LogRemoteCommands

Activez la journalisation des commandes shell exécutées en tant qu'avertissements. Les commandes seront journalisées 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

LogType

Le type de la sortie du journal :
file - écrit le journal dans le fichier spécifié par le paramètre LogFile ;
system - écrit le journal dans le journal des événements Windows ;
console - écrit le journal sur la sortie standard.

Par défaut : `file`

MaxLinesPerSecond

Le nombre maximal de nouvelles lignes que l'agent enverra par seconde au serveur Zabbix ou au proxy lors du traitement des contrôles actifs « log », « logrt » et « eventlog ». La valeur fournie sera remplacée par le paramètre « maxlines », fourni 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

PerfCounter

Définit un nouveau paramètre <parameter_name>, qui correspond à la valeur moyenne du compteur de performances système <perf_counter_path> pour la période <period> spécifiée (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 performances. Le nom du paramètre (interrupts) doit être utilisé comme clé d'élément lors de la création d'un élément. Des échantillons pour le calcul de la valeur moyenne seront relevés chaque seconde.
Vous pouvez exécuter « typeperf -qx » pour obtenir la liste de tous les compteurs de performances disponibles dans Windows.

PerfCounterEn

Définit un nouveau paramètre <parameter_name>, qui correspond à la valeur moyenne du compteur de performance système <perf_counter_path> pour la période <period> spécifiée (en secondes). Par rapport à PerfCounter, les chemins des compteurs 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 d'élément lors de la création d'un élément. Des échantillons 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

Fréquence de rafraîchissement de la liste des contrôles actifs, en secondes. Notez qu'après un échec du rafraîchissement des contrôles actifs, la prochaine tentative de rafraîchissement sera effectuée dans 60 secondes.

Par défaut : 5
 Plage : 1-86400

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. 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 traitées de manière identique 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 la 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

L'adresse du serveur/proxy Zabbix ou la configuration du cluster à partir desquels obtenir les contrôles actifs. L'adresse du serveur/proxy est une adresse IP ou un nom DNS, avec un port optionnel séparé par deux-points.
La configuration du cluster est constituée d'une ou plusieurs adresses de membres du groupe de serveurs ou de proxies, 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 proxies, il ne faut pas spécifier plus d'un proxy Zabbix pour chaque serveur/cluster Zabbix. Si un proxy Zabbix est spécifié, le serveur/cluster Zabbix correspondant à ce proxy ne doit pas être spécifié.
Plusieurs adresses séparées par des virgules peuvent être fournies pour utiliser en parallèle plusieurs serveurs Zabbix indépendants. 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 un port est

spécifié pour cet hôte. Si le port n'est pas spécifié, les crochets pour les adresses IPv6 sont facultatifs.
Si ce paramètre n'est pas spécifié, les contrôles actifs sont désactivés.

Exemple pour un proxy Zabbix :

```
ServerActive=127.0.0.1:10051
```

Exemple pour un groupe de proxies 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 : (*)

SourceIP

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

Le nombre d'instances pré-forkées de zabbix_agentd qui traitent les contrôles passifs. S'il est défini sur 0, les contrôles passifs sont désactivés et l'agent n'écoutera sur aucun port TCP.

Par défaut : 10
 Plage : 0-100 (*)

Délai d'attente

Indique combien de temps attendre (en secondes) pour établir une 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'éléments dans les contrôles actifs ;
- la récupération des données de journal via la surveillance de logfile ou du journal des événements Windows ;
- l'envoi de messages heartbeat ;
- la durée maximale des contrôles `vfs.*` ;
- l'utilisation par les modules de l'agent Zabbix ;
- le rôle de solution de secours dans les scénarios où un serveur ou un proxy antérieur à la version 7.0 envoie des contrôles sans délais d'attente.

Ce délai d'attente ne sera **pas** utilisé pour les contrôles d'agent qui disposent de paramètres de délai d'attente configurables dans l'interface web (au niveau global, du proxy ou de chaque élément).

Par défaut : 3
 Plage : 1-30

TLSAccept

Les connexions entrantes à accepter. Utilisé pour les contrôles passifs. 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 pré-partagée (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

Le chemin complet du fichier contenant les certificats de la ou des autorités de certification (CA) de niveau supérieur pour la vérification du certificat du pair, utilisé pour les communications chiffrées entre les composants Zabbix.

TLSCertFile

Le chemin complet du fichier contenant le certificat de l'agent ou la chaîne de certificats, utilisé pour les communications chiffrées avec les composants Zabbix.

TLSConnect

Comment l'agent doit se connecter au serveur Zabbix ou au proxy. Utilisé pour les contrôles actifs. Une seule valeur peut être spécifiée :
unencrypted - se connecter sans chiffrement (par défaut)
psk - se connecter en utilisant TLS et une clé pré-partagée (PSK)
cert - se connecter en utilisant TLS et un certificat

Obligatoire : oui, si les paramètres de certificat TLS ou de PSK sont définis (même pour une connexion *unencrypted*) ; sinon non

TLSCRLFile

Le chemin complet du fichier contenant les certificats révoqués. Ce paramètre est utilisé pour les communications chiffrées entre les composants Zabbix.

TLSKeyFile

Le chemin complet du fichier contenant la clé privée de l'agent, utilisée pour les communications chiffrées entre les composants Zabbix.

TLSPSKFile

Le chemin complet du fichier contenant la clé pré-partagée de l'agent, utilisée pour les communications chiffrées avec le serveur Zabbix.

TLSPSKIdentity

La chaîne d'identité de clé pré-partagée, utilisée pour les communications chiffrées avec le serveur Zabbix.

TLSServerCertIssuer

L'émetteur autorisé du certificat du serveur (proxy).

TLSServerCertSubject

Le sujet du certificat autorisé du serveur (proxy).

UnsafeUserParameters

Autorise le passage de tous les caractères 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

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 une fin de ligne. Les commandes shell peuvent avoir 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

Le chemin de recherche par défaut pour les commandes UserParameter. S'il est utilisé, l'agent changera son répertoire de travail pour celui spécifié ici avant d'exécuter une commande. Ainsi, les commandes UserParameter peuvent avoir un préfixe relatif `./` au lieu d'un chemin complet. Une seule entrée est autorisée.

Exemple :

```
UserParameterDir=C:\opt\myscripts
```

> **Note:**
> (*) Le nombre de serveurs actifs répertoriés dans ServerActive plus le nombre d'instances pré-forkées pour les vérifications passives spécifié dans StartAgents doit être inférieur à 64.

6 Agent Zabbix 2 (Windows)

Aperçu

Zabbix agent 2 est une nouvelle génération d'agent Zabbix et peut être utilisé à la place de Zabbix agent.

Les paramètres pris en charge par le fichier de configuration de l'agent Zabbix agent 2 pour Windows (zabbix_agent2.conf) sont répertoriés dans cette section.

Les paramètres sont listés sans informations supplémentaires. Cliquez sur un paramètre pour voir tous les détails.

Parameter	Description
Alias	Définit un alias pour une clé d'élément.
AllowKey	Autorise l'exécution des clés d'élément correspondant à un modèle.
BufferSend	Ne pas conserver les données plus de N secondes dans le tampon.
BufferSize	Le nombre maximal de valeurs dans le tampon mémoire.
ControlSocket	Le socket de contrôle, utilisé pour envoyer des commandes d'exécution avec l'option '-R'.
DebugLevel	Le niveau de débogage.
DenyKey	Refuse l'exécution des clés d'élément correspondant à un modèle.
EnablePersistentBuffer	Active l'utilisation d'un stockage persistant local pour les éléments actifs.
ForceActiveChecksOnStart	Exécute les contrôles actifs immédiatement après le redémarrage pour la première configuration reçue.
HeartbeatFrequency	La fréquence des messages de pulsation en secondes.
HostInterface	Paramètre facultatif qui définit l'interface de l'hôte.
HostInterfaceItem	Paramètre facultatif qui définit un élément utilisé pour obtenir l'interface de l'hôte.
HostMetadata	Paramètre facultatif qui définit les métadonnées de l'hôte.
HostMetadataItem	Paramètre facultatif qui définit un élément Zabbix agent utilisé pour obtenir les métadonnées de l'hôte.
Hostname	Paramètre facultatif qui définit le nom d'hôte.
HostnameItem	Paramètre facultatif qui définit un élément Zabbix agent utilisé pour obtenir le nom d'hôte.
Include	Vous pouvez inclure des fichiers individuels ou tous les fichiers d'un répertoire dans le fichier de configuration.
ListenIP	Une liste d'adresses IP séparées par des virgules sur lesquelles l'agent doit écouter.
ListenPort	L'agent 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.
PersistentBufferFile	Le fichier dans lequel Zabbix agent 2 doit conserver la base de données SQLite.
PersistentBufferPeriod	La période pendant laquelle les données doivent être stockées lorsqu'il n'y a pas de connexion au serveur ou au proxy.
Plugins.<PluginName>.SystemCapacity	La limite de contrôles par plugin pouvant être exécutés en même temps.
Plugins.Log.MaxLinesPerSecond	Le nombre maximal de nouvelles lignes que l'agent enverra par seconde au serveur Zabbix ou au proxy lors du traitement des contrôles actifs 'log' et 'logrt'.
Plugins.SystemRun.LogRemoteCommands	Activer la journalisation des commandes shell exécutées en tant qu'avertissements.
PluginSocket	Le chemin vers le socket UNIX pour les communications avec les plugins chargeables.
PluginTimeout	Le délai d'attente pour les connexions avec les plugins chargeables, en secondes.
PerfCounter	Définit un nouveau paramètre <parameter_name> qui est la valeur moyenne du compteur de performances système <perf_counter_path> pour la période <period> spécifiée (en secondes).
PerfCounterEn	Définit un nouveau paramètre <parameter_name> qui est la valeur moyenne du compteur de performances système <perf_counter_path> pour la période <period> spécifiée (en secondes). Par rapport à PerfCounter, les chemins perfcounter doivent être en anglais.
RefreshActiveChecks	À quelle fréquence la liste des contrôles actifs est actualisée.
Server	Une liste d'adresses IP séparées par des virgules, éventuellement en notation CIDR, ou de noms DNS de serveurs Zabbix et de proxies Zabbix.
ServerActive	L'adresse du serveur/proxy Zabbix ou la configuration du cluster à partir de laquelle obtenir les contrôles actifs.
SourceIP	L'adresse IP source.
StatusPort	Si défini, l'agent écoutera sur ce port les requêtes d'état HTTP (http://localhost:<port>/status).
Timeout	Spécifie combien de temps attendre (en secondes) pour établir une connexion et échanger des données avec le proxy ou le serveur Zabbix.
TLSAccept	Quelles connexions entrantes accepter.
TLSCAFile	Le chemin complet d'un fichier contenant les certificats de la ou des autorités de certification 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 de l'agent ou la chaîne de certificats, utilisé pour les communications chiffrées entre les composants Zabbix.
TLSCipherAll13	La chaîne de chiffrement OpenSSL (TLS 1.3). Remplace les critères de sélection par défaut de la suite cryptographique pour le chiffrement basé sur certificat et PSK.

Parameter	Description
TLSCipherCert13	La chaîne de chiffrement OpenSSL (TLS 1.3). Remplace les critères de sélection par défaut de la suite cryptographique pour le chiffrement basé sur certificat.
TLSCipherPSK13	La chaîne de chiffrement OpenSSL (TLS 1.3). Remplace les critères de sélection par défaut de la suite cryptographique pour le chiffrement basé sur PSK.
TLSConnect	Comment l'agent doit se connecter au serveur ou proxy Zabbix.
TLSCRLFile	Le chemin complet d'un fichier contenant les certificats révoqués. Ce paramètre est utilisé pour les communications chiffrées entre les composants Zabbix.
TLSKeyFile	Le chemin complet d'un fichier contenant la clé privée de l'agent, utilisé pour les communications chiffrées entre les composants Zabbix.
TLSPSKFile	Le chemin complet d'un fichier contenant la clé pré-partagée de l'agent, utilisé pour les communications chiffrées avec le serveur Zabbix.
TLSPSKIdentity	La chaîne d'identité de la clé pré-partagée, utilisée pour les communications chiffrées avec le serveur Zabbix.
TLSServerCertIssuer	L'émetteur autorisé du certificat du serveur (proxy).
TLSServerCertSubject	Le sujet autorisé du certificat du serveur (proxy).
UnsafeUserParameters	Autorise le passage de tous les caractères dans les arguments des paramètres définis par l'utilisateur.
UserParameter	Un paramètre défini par l'utilisateur à surveiller.
UserParameterDir	Le chemin de recherche par défaut pour les commandes UserParameter.

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 les valeurs des fichiers de configuration fournis ;
- Les valeurs prennent en charge les variables d'environnement ;
- 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 des paramètres

Alias

Définit un alias pour une clé d'élément. Il peut être utilisé pour remplacer une clé d'élément longue et complexe par une autre 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 référencer la même clé d'élément.
 Les alias peuvent être utilisés dans *HostMetadataItem*, mais pas dans le paramètre *HostnameItem*.

Exemple 1 : Récupération du pourcentage d'utilisation du fichier d'échange 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 CPU avec les paramètres par défaut et des paramètres personnalisés.

```
Alias=cpu.load:system.cpu.load
Alias=cpu.load[*]:system.cpu.load[*]
```

Cela permet d'utiliser la clé **cpu.load** pour obtenir la charge CPU avec les paramètres par défaut, ainsi que **cpu.load[percpu,avg15]** pour obtenir des données spécifiques sur la charge CPU.

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 désormais 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

Autoriser l'exécution des clés d'élément qui correspondent à un motif. Le motif 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. Les paramètres sont traités un par un selon leur ordre d'apparition. Voir aussi : Restriction des vérifications de l'agent.

BufferSend

L'intervalle de temps en secondes qui détermine la fréquence à laquelle les valeurs sont envoyées du tampon vers le serveur Zabbix.
Notez que si le tampon est plein, les données seront envoyées plus tôt.

Par défaut : 5
 Plage : 1-3600

BufferSize

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

Le socket de contrôle, utilisé pour envoyer des commandes d'exécution avec l'option « -R ».

Par défaut : `\\.\pipe\agent.sock`

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).

Par défaut : 3
 Plage : 0-5

DenyKey

Refuse l'exécution des clés d'élément qui correspondent à un motif. Le motif 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 AllowKey. Les paramètres sont traités un par un selon leur ordre d'apparition. Voir aussi : Restriction des vérifications de l'agent.

EnablePersistentBuffer

Active 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

ForceActiveChecksOnStart

Effectue les vérifications actives immédiatement après le redémarrage pour la première configuration reçue. Également disponible comme paramètre de configuration par plugin, par exemple : `Plugins.Uptime.System.ForceActiveChecksOnStart=1`

Par défaut : 0
 Valeurs : 0 - désactivé, 1 - activé

HeartbeatFrequency

La fréquence des messages de pulsation en secondes. Utilisée pour surveiller la disponibilité des vérifications actives.
0 - messages de pulsation désactivés.

Par défaut : 60
 Plage : 0-3600

HostInterface

Paramètre facultatif qui définit l'interface de l'hôte (adresse IP ou nom DNS) utilisée pendant le processus d'enregistrement automatique. Cette valeur sera utilisée pour renseigner l'interface dans l'hôte nouvellement créé et permet de configurer explicitement une adresse IP ou DNS. Pour plus de détails, voir Utilisation de DNS comme interface par défaut.

S'il n'est pas défini, la valeur sera obtenue à partir de HostInterfaceItem.

L'agent émettra une erreur et ne démarrera pas si la valeur dépasse la limite de 255 caractères.

Plage : 0 à 255 caractères

HostInterfaceItem

Un paramètre facultatif qui définit un élément utilisé pour déterminer l'interface de l'hôte (adresse IP ou nom DNS) pendant le processus d'autoregistration de l'hôte. Cette valeur est utilisée uniquement si HostInterface n'est pas défini. Pour plus de détails, voir Utilisation de DNS comme interface par défaut.

Lors d'une requête d'autoregistration, 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 независимоамment des paramètres AllowKey/DenyKey.

HostMetadata

Paramètre facultatif qui définit les métadonnées utilisées pour identifier ou distinguer l'hôte pendant le processus d'auto-enregistrement de l'hôte (agent actif). HostMetadata permet de distinguer les hôtes au-delà du nom d'hôte.

S'il n'est pas défini, la valeur sera obtenue à partir de HostMetadataItem.

L'agent signalera une erreur et ne démarrera pas si la valeur spécifiée dépasse la limite de 2034 octets ou s'il s'agit d'une chaîne non UTF-8. Lorsqu'un paramètre attend une adresse IP ou un nom DNS, les valeurs valides en UTF-8 mais qui ne sont pas des adresses IP ou des noms DNS valides seront également rejetées et signalées comme invalides.

Les métadonnées sur plusieurs lignes ne sont pas prises en charge — la sortie sera tronquée au premier saut de ligne.

Plage : 0-2034 octets

HostMetadataItem

Un paramètre facultatif qui définit un élément d'agent Zabbix utilisé pour obtenir les métadonnées de l'hôte. Cette option n'est utilisée que lorsque HostMetadata n'est pas défini.

La valeur de HostMetadataItem est récupérée à chaque tentative d'auto-enregistrement et n'est utilisée que lors du processus d'auto-enregistrement 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 независимоamment des paramètres AllowKey/DenyKey.

Lors d'une requête d'auto-enregistrement, l'agent consignera un message d'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 ne sont pas des adresses IP ou des noms DNS valides seront également rejetées et signalées comme invalides.

Les métadonnées sur plusieurs lignes ne sont pas prises en charge — la sortie sera tronquée au premier saut de ligne.

Hostname

Une liste de noms d'hôte uniques, sensibles à la casse, séparés par des virgules. Obligatoire 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

Paramètre facultatif qui définit un élément utilisé pour obtenir le nom d'hôte. Cette option n'est utilisée que 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 независимо des valeurs de AllowKey/DenyKey.

Par défaut : `system.hostname`

Include

Vous pouvez inclure des fichiers individuels ou tous les fichiers d'un répertoire 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 Windows MSI ; situé dans le dossier spécifié lors de l'installation si l'agent Zabbix est installé sous forme d'archive zip). Tous les fichiers inclus doivent avoir une syntaxe correcte, sinon l'agent ne démarrera pas. Le chemin peut être relatif à l'emplacement du fichier *zabbix_agent2.conf* (par exemple, `Include=.\zabbix_agent2.d\plugins.d\*.conf`).
Pour n'inclure que les fichiers pertinents dans le répertoire spécifié, le caractère générique astérisque est pris en charge pour la correspondance de motifs.
Voir les notes spéciales concernant les limitations.

Exemple :

```
Include=C:\Program Files\Zabbix Agent2\zabbix_agent2.d\*.conf
```

ListenIP

Une liste d'adresses IP séparées par des virgules sur lesquelles l'agent doit écouter. La première adresse IP est envoyée au serveur Zabbix, lors de la connexion à celui-ci, afin de récupérer la liste des contrôles actifs.

Par défaut : `0.0.0.0`

ListenPort

L'agent écoutera sur ce port les connexions provenant du serveur.

Par défaut : 10050
 Plage : 1024-32767

LogFile

Le nom du fichier journal de l'agent.

Par défaut : `c:\zabbix_agent2.log`
 Obligatoire : Oui, si LogType est défini sur *file* ; sinon non

LogFileSize

La taille maximale d'un 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 redémarré à zéro.

Par défaut : 1
 Plage : 0-1024

LogType

Le type de la 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`

PersistentBufferFile

Le fichier dans lequel Zabbix agent 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

La période pendant laquelle les données doivent être stockées 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 n'est utilisé que si le tampon persistant est activé (*EnablePersistentBuffer=1*).

Par défaut : `1h`
 Plage : 1m-365d

Plugins.<PluginName>.System.Capacity

La limite de contrôles par plugin `<PluginName>` pouvant être exécutés en même temps.

Par défaut : 1000 Plage : 1-1000

Plugins.Log.MaxLinesPerSecond

Le nombre maximal de nouvelles lignes que l'agent enverra par seconde au serveur Zabbix ou au proxy lors du traitement des contrôles actifs 'log', 'logrt' et 'eventlog'. La valeur fournie sera remplacée par le paramètre 'maxlines', fourni 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

Activez la journalisation des commandes shell exécutées en tant qu'avertissements. Les commandes seront journalisées 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

Le chemin vers le socket UNIX pour les communications avec les plugins chargeables.

Par défaut : `\\.\pipe\agent.plugin.sock`

PluginTimeout

Le délai d'expiration des connexions avec les plugins chargeables, en secondes.

Par défaut : `Timeout`
 Plage : 1-30

PerfCounter

Définit un nouveau paramètre <parameter_name>, qui correspond à la valeur moyenne du compteur de performance système <perf_counter_path> pour la période <period> spécifiée (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 performance. Le nom du paramètre (interrupts) doit être utilisé comme clé d'élément lors de la création d'un élément. Des échantillons pour le calcul de la valeur moyenne seront relevés chaque seconde.
Vous pouvez exécuter « typeperf -qx » pour obtenir la liste de tous les compteurs de performance disponibles sous Windows.

PerfCounterEn

Définit 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 des compteurs 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 obtenir 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 d'élément lors de la création d'un élément. Des échantillons 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

Fréquence de rafraîchissement de la liste des contrôles actifs, en secondes. Notez qu'après un échec du rafraîchissement des contrôles actifs, la prochaine tentative de rafraîchissement sera effectuée dans 60 secondes.

Par défaut : 5
 Plage : 1-86400

Serveur

Une liste d'adresses IP séparées par des virgules, éventuellement en notation CIDR, ou de noms DNS de serveurs Zabbix ou de proxies Zabbix. 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 traitées de manière identique 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 contrôles passifs sont désactivés 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

L'adresse du serveur/proxy Zabbix ou la configuration du cluster à partir desquels obtenir les contrôles actifs. L'adresse du serveur/proxy est une adresse IP ou un nom DNS, avec un port optionnel séparé par deux-points.
La configuration du cluster est constituée d'une ou plusieurs adresses de membres du groupe de serveurs ou de proxies, 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 proxies, il ne faut pas spécifier plus d'un proxy Zabbix pour chaque serveur/cluster Zabbix. Si un proxy Zabbix est spécifié, le serveur/cluster Zabbix correspondant à ce proxy ne doit pas être spécifié.
Plusieurs adresses séparées par des virgules peuvent être fournies afin d'utiliser en parallèle plusieurs serveurs Zabbix indépendants. 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 un port est spécifié pour cet hôte. Si le port n'est pas spécifié, les crochets pour les adresses IPv6 sont facultatifs.
Si ce paramètre n'est pas spécifié, les contrôles actifs sont désactivés.

Exemple pour un proxy Zabbix :

```
ServerActive=127.0.0.1:10051
```

Exemple pour un groupe de proxies 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

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

Si ce paramètre est défini, l'agent écoutera sur ce port les requêtes d'état HTTP (http://localhost:<port>/status).

Plage : 1024-32767

Délai d'attente

Indique combien de temps attendre (en secondes) pour établir une 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'éléments dans les contrôles actifs ;
- la récupération des données de journal via la surveillance de logfile ou du journal des événements Windows ;
- l'envoi de messages heartbeat ;
- la durée maximale des contrôles `vfs.*` ;
- son utilisation comme solution de secours dans les cas où un serveur ou un proxy antérieur à la version 7.0 envoie des contrôles sans délais d'attente.

Ce délai d'attente **ne sera pas** utilisé pour les contrôles d'agent qui disposent de paramètres de délai d'attente configurables dans l'interface web (au niveau global, du proxy ou de l'élément).

Par défaut : 3
 Plage : 1-30

TLSAccept

Les connexions entrantes à accepter. Utilisé pour les contrôles passifs. 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 pré-partagée (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

Le chemin complet du fichier contenant les certificats de la ou des autorités de certification (CA) de niveau supérieur pour la vérification du certificat du pair, utilisé pour les communications chiffrées entre les composants Zabbix.

TLSCertFile

Le chemin complet du fichier contenant le certificat de l'agent ou la chaîne de certificats, utilisé pour les communications chiffrées avec les composants Zabbix.

TLSCipherAll13

La chaîne de chiffrement OpenSSL (TLS 1.3). Remplace les critères de sélection par défaut des suites cryptographiques 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
```

TLSCipherCert13

La chaîne de chiffrement OpenSSL (TLS 1.3). Remplace les critères de sélection par défaut de la suite cryptographique pour le chiffrement basé sur un certificat.

Notez que ce paramètre ne peut pas être utilisé avec `TLSAccept=cert,psk` ; pour une connexion par certificat (`TLSConnect=cert`), utilisez plutôt TLSCipherAll13.

TLSCipherPSK13

La chaîne de chiffrement OpenSSL (TLS 1.3). Remplace 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

Comment l'agent doit se connecter au serveur Zabbix ou au proxy. Utilisé pour les contrôles actifs. Une seule valeur peut être spécifiée :
unencrypted - se connecter sans chiffrement (par défaut)
psk - se connecter en utilisant TLS et une clé pré-partagée (PSK)
cert - se connecter en utilisant TLS et un certificat

Obligatoire : oui, si les paramètres de certificat TLS ou de PSK sont définis (même pour une connexion *unencrypted*) ; sinon non

TLSCRLFile

Le chemin complet du fichier contenant les certificats révoqués. Ce paramètre est utilisé pour les communications chiffrées entre les composants Zabbix.

TLSKeyFile

Le chemin complet du fichier contenant la clé privée de l'agent, utilisée pour les communications chiffrées entre les composants Zabbix.

TLSPSKFile

Le chemin complet du fichier contenant la clé pré-partagée de l'agent, utilisée pour les communications chiffrées avec le serveur Zabbix.

TLSPSKIdentity

La chaîne d'identité de clé pré-partagée, utilisée pour les communications chiffrées avec le serveur Zabbix.

TLSServerCertIssuer

L'émetteur de certificat autorisé du serveur (proxy).

TLSServerCertSubject

Le sujet du certificat autorisé du serveur (proxy).

UnsafeUserParameters

Autorise le passage de tous les caractères 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

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 une fin de ligne. Les commandes shell peuvent avoir 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

Le chemin de recherche par défaut pour les commandes UserParameter. S'il est utilisé, l'agent changera son répertoire de travail pour celui spécifié ici avant d'exécuter une commande. Ainsi, les commandes UserParameter peuvent avoir un préfixe relatif `.\` au lieu d'un chemin complet.
Une seule entrée est autorisée.

Exemple :

```
UserParameterDir=C:\opt\myscripts
```

7 Configuration parameters of plugins

See configuration parameters for Zabbix agent 2 plugins:

- 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. plugin

Plugin configuration principles

This page describes plugin configuration principles and best practices.

All plugins are configured using *Plugins.** parameter, which can either be part of the Zabbix agent 2 configuration file or a plugin's own configuration file. If a plugin uses a separate configuration file, path to this file should be specified in the Include parameter of Zabbix agent 2 configuration file.

A typical plugin parameter has the following structure:

Plugins.<PluginName>.<Parameter>=<Value>

Additionally, there are two specific groups of parameters:

- *Plugins.<PluginName>.Default.<Parameter>=<Value>* used for defining default parameter values.
- *Plugins.<PluginName>.<SessionName>.<Parameter>=<Value>* used for defining separate sets of parameters for different monitoring targets via named sessions.

All parameter names should adhere to the following requirements:

- it is recommended to capitalize the names of your plugins
- the parameter should be capitalized
- special characters are not allowed
- nesting isn't limited by a maximum level
- the number of parameters is not limited

For example, to perform active checks that do not have *Scheduling* update interval immediately after the agent restart only for the Uptime plugin, set `Plugins.Uptime.System.ForceActiveChecksOnStart=1` in the configuration file. Similarly, to set custom limit for concurrent checks for the CPU plugin, set the `Plugins.CPU.System.Capacity=N` in the configuration file.

Default values

You can set default values for the connection-related parameters (URI, username, password, etc.) in the configuration file in the format:

```
Plugins.<PluginName>.Default.<Parameter>=<Value>
```

For example, `Plugins.Mysql.Default.Username=zabbix`, `Plugins.MongoDB.Default.Uri=tcp://127.0.0.1:27017`, etc.

If a value for such parameter is not provided in an item key or in the named session parameters, the plugin will use the default value. If a default parameter is also undefined, hardcoded defaults will be used.

> **Note:**
> If an item key does not have any parameters, Zabbix agent 2 will attempt to collect the metric using values defined in the default parameters section.

Named sessions

Named sessions represent an additional level of plugin parameters and can be used to specify separate sets of authentication parameters for each of the instances being monitored. Each named session parameter should have the following structure:

```
Plugins.<PluginName>.Sessions.<SessionName>.<Parameter>=<Value>
```

A session name can be used as a connString item key parameter instead of specifying a URI, username, and/or password separately.

In item keys, the first parameter can be either a connString or a URI. If the first key parameter doesn't match any session name, it will be treated as a URI. Note that passing embedded URI credentials in the item key is not supported, use named session parameters instead.

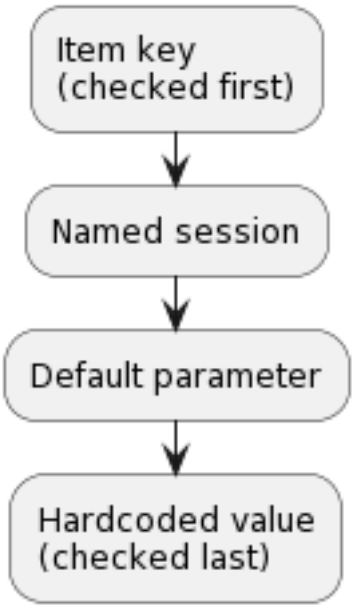
The list of available named session parameters depends on the plugin.

It is possible to override session parameters by specifying new values in the item key parameters (see example).

If a parameter is not defined for the named session, Zabbix agent 2 will use the value defined in the default plugin parameter.

Parameter priority

Zabbix agent 2 plugins search for connection-related parameter values in the following order:



1. The first item key parameter is compared to session names. If no match is found, it is treated as an actual value; in this case, step 3 will be skipped. If a match is found, the parameter value (usually, a URI) must be defined in the named session.
2. Other parameters will be taken from the item key if defined.
3. If an item key parameter (for example, password) is empty, plugin will look for the corresponding named session parameter.
4. If the session parameter is also not specified, the value defined in the corresponding default parameter will be used.
5. If all else fails, the plugin will use the hardcoded default value.

Example 1

Monitoring of two instances “MySQL1” and “MySQL2”.

Configuration parameters:

```
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://192.0.2.0:3306
Plugins.Mysql.Sessions.MySQL2.User=mysql2_user
Plugins.Mysql.Sessions.MySQL2.Password=different_password
```

As a result of this configuration, each session name may be used as a connString in an item key, e.g., `mysql.ping[MySQL1]` or `mysql.ping[MySQL2]`.

Example 2

Providing some of the parameters in the item key.

Configuration parameters:

```
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
```

Item key: `pgsql.ping[session1,new_username,,postgres]`

As a result of this configuration, the agent will connect to PostgreSQL using the following parameters:

- URI from session parameter: `192.0.2.234:5432`
- Username from the item key: `new_username`
- Password from session parameter (since it is omitted in the item key): `session_password`
- Database name from the item key: `postgres`

Example 3

Collecting a metric using default configuration parameters.

Configuration parameters:

```
Plugins.PostgreSQL.Default.Uri=tcp://192.0.2.234:5432
Plugins.PostgreSQL.Default.User=zabbix
Plugins.PostgreSQL.Default.Password=password
```

Item key: `pgsql.ping[,,,postgres]`

As a result of this configuration, the agent will connect to PostgreSQL using the parameters:

- Default URI: `192.0.2.234:5432`
- Default username: `zabbix`
- Default password: `password`
- Database name from the item key: `postgres`

Connections

Some plugins support gathering metrics from multiple instances simultaneously. Both local and remote instances can be monitored. TCP and Unix-socket connections are supported.

It is recommended to configure plugins to keep connections to instances in an open state. The benefits are reduced network congestion, latency, and CPU and memory usage due to the lower number of connections. The client library takes care of this.

> **Note:**
> Time period for which unused connections should remain open can be determined by *Plugins.<PluginName>.KeepAlive* parameter. Example: *Plugins.Memcached.KeepAlive*

1 Ceph plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the Ceph plugin.

It is recommended to specify them in their own configuration file (e.g. `ceph.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Ceph.Default.ApiKey	no			Default API key for connecting to Ceph; used if no value is specified in an item key or named session.
Plugins.Ceph.Default.Mode	no	native / restful	restful (deprecated)	Default mode for connecting to Ceph. The native mode is only supported on Linux and starting with Ceph 16. The restful mode will not work with Ceph version 20 or newer. Note that the user credential set differs for each mode and they are not compatible with each other.
Plugins.Ceph.Default.User	no			Default username for connecting to Ceph; used if no value is specified in an item key or named session.

Parameter	Mandatory	Range	Default	Description
Plugins.Ceph.Default.Uri			https://localhost:8003	Default URI for connecting to Ceph; used if no value is specified in an item key or named session. Should not include embedded credentials (they will be ignored). Must match the URI format. Only `https` scheme is supported; a scheme can be omitted. A port can be omitted (default=8003). Examples: `https://127.0.0.1:8003` `localhost`
Plugins.Ceph.InsecureSkipVerify		false / true	false	Determines whether an http client should verify the server's certificate chain and host name. If *true*, TLS accepts any certificate presented by the server and any host name in that certificate. In this mode, TLS is susceptible to man-in-the-middle attacks (should be used only for testing).
Plugins.Ceph.KeepAlive		60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.Ceph.Sessions.<SessionName>.ApiKey				Named session API key. **<SessionName>** - define name of a session for using in item keys.
Plugins.Ceph.Sessions.<SessionName>.Mode		Native / restful	restful (deprecated)	Named session mode for connecting to Ceph. The native mode is only supported on Linux and starting with Ceph 16. The restful mode will not work with Ceph version 20 or newer. Note that the user credential set differs for each mode and they are not compatible with each other.
Plugins.Ceph.Sessions.<SessionName>.User				Named session username. **<SessionName>** - define name of a session for using in item keys.
Plugins.Ceph.Sessions.<SessionName>.Uri				Connection string of a named session. **<SessionName>** - define name of a session for using in item keys. Should not include embedded credentials (they will be ignored). Must match the URI format. Only `https` scheme is supported; a scheme can be omitted. A port can be omitted (default=8003). Examples: `https://127.0.0.1:8003` `localhost`
Plugins.Ceph.Timeout		1-30	global timeout	Request execution timeout (the duration, in seconds, to wait for a request to complete before shutting it down).

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

2 Docker plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the Docker plugin.

It is recommended to specify them in their own configuration file (e.g. `docker.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;

- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Docker.Endpoint			unix:///var/run/docker.sock	Docker daemon unix-socket location. Must contain a scheme (only `unix://` is supported).
Plugins.Docker.Timeout		1-30	global timeout	Request execution timeout (the duration, in seconds, to wait for a request to complete before shutting it down).

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

3 Ember+ plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the Ember+ plugin.

It is recommended to specify them in their own configuration file (e.g. `emberplus.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

The Ember+ plugin is a loadable plugin and is available and fully described in the Ember+ plugin repository.

This plugin is currently only available to be built from the source (for both Unix and Windows).

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files.
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM.
- Comments starting with "#" are only supported at the beginning of the line.

Options

Parameter	Description
-V --version	Print the plugin version and license information.
-h --help	Print help information (shorthand).
-t, --test <item key>	Launch plugin for testing (plugin config ignored).

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.EmberPlus.Default.Uri			tcp://localhost:9998	The default URI to connect. The only supported schema is `tcp://`. A schema can be omitted. Embedded credentials will be ignored.
Plugins.EmberPlus.KeepAlive		60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.EmberPlus.Sessions.<SessionName>.Uri			tcp://localhost:9998	The URI to connect, for the named session. The only supported schema is `tcp://`. A schema can be omitted. Embedded credentials will be ignored. **<SessionName>** - define name of a session for using in item keys.
Plugins.EmberPlus.System.Path				Path to the Ember+ plugin executable. Example usage: `Plugins.EmberPlus.System.Path=/usr/libexec/zabbix/zabbix`
Plugins.EmberPlus.Timeout		1-30	global timeout	The duration, in seconds, to wait for a server to respond when first connecting and on follow-up operations in the session.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

4 Memcached plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the Memcached plugin.

It is recommended to specify them in their own configuration file (e.g. `memcached.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Memcached.Default.Password				Default password for connecting to Memcached; used if no value is specified in an item key or named session.

Plugins.Memcached.Default.Uri			tcp://localhost:11211	Default URI for connecting to Memcached; used if no value is specified in an item key or named session.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`; a scheme can be omitted.
				A port can be omitted (default=11211).
				Examples: `tcp://localhost:11211`
				`localhost`
				`unix:/var/run/memcached.sock`

Plugins.Memcached.Default.User				Default username for connecting to Memcached; used if no value is specified in an item key or named session.
Plugins.Memcached.KeepAlive		60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.Memcached.Sessions.<SessionName>.Password				Named session password. **<SessionName>** - define name of a session for using in item keys.

Plugins.Memcached.Sessions.<SessionName>.Uri				Connection string of a named session.
				<SessionName> - define name of a session for using in item keys.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`; a scheme can be omitted.
				A port can be omitted (default=11211).
				Examples: `tcp://localhost:11211`
				`localhost`
				`unix:/var/run/memcached.sock`

Plugins.Memcached.Sessions.<SessionName>.User				Named session username. **<SessionName>** - define name of a session for using in item keys.
Plugins.Memcached.Timeout		1-30	global timeout	Request execution timeout (the duration, in seconds, to wait for a request to complete before shutting it down).

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

5 Modbus plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the Modbus plugin.

It is recommended to specify them in their own configuration file (e.g. `modbus.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Modbus.Sessions.<SessionName>.Endpoint				Endpoint is a connection string consisting of a protocol scheme, a host address and a port or serial port name and attributes. **<SessionName>** - define name of a session for using in item keys.
Plugins.Modbus.Sessions.<SessionName>.SlaveID				Slave ID of a named session. **<SessionName>** - define name of a session for using in item keys. Example: `Plugins.Modbus.Sessions.MB1.SlaveID=20` *Note* that this named session parameter is checked only if the value provided in the item key slave ID parameter is empty.
Plugins.Modbus.Sessions.<SessionName>.Timeout				Timeout of a named session in seconds. **<SessionName>** - define name of a session for using in item keys. Example: `Plugins.Modbus.Sessions.MB1.Timeout=2`

If you need to set the request execution timeout (how long to wait for a request to complete before shutting it down), use the item configuration form.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

6 MongoDB plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the MongoDB plugin.

It is recommended to specify them in their own configuration file (e.g., `mongo.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

The MongoDB plugin is a loadable plugin and is available and fully described in the MongoDB plugin repository.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files.
- Values support environment variables.
- Zabbix supports configuration files only in UTF-8 encoding without BOM.
- Comments starting with "#" are only supported at the beginning of the line.

Options

Parameter	Description
-V --version	Print the plugin version and license information.
-h --help	Print help information (shorthand).
-t, --test <item key>	Launch plugin for testing (plugin config ignored).

Parameters

When connecting to MongoDB, the plugin checks its configuration in a specific order, which determines which value to use. The order, from highest to lowest priority, is as follows:

1. Parameters specified in the connection URI (e.g., query parameters defined after ? in `Plugins.MongoDB.Sessions.<session_name` have the highest priority).
2. Parameters in the Zabbix agent 2 item key (e.g., mongodb.collection.stats).
3. Parameters defined in a named session (e.g., `Plugins.MongoDB.Sessions.<SessionName>.User`, `Plugins.MongoDB.Sessions`
4. Default values (e.g., `Plugins.MongoDB.Default.User`, `Plugins.MongoDB.Default.Password`).

For example:

- If you specify the database for authenticating a connection to the MongoDB server in the `*.<SessionName>.Uri` parameter (e.g., `Plugins.MongoDB.Sessions.<session_name>.Uri=mongodb://user:password@127.0.0.1:27017/?authSource=a` then the plugin will consider the `*.<SessionName>.AuthSource` parameter to be set to `admin`, even if the `*.<SessionName>.AuthSource` parameter is set to a different value.
- If you specify the URI in the `*.<SessionName>.Uri` parameter (e.g., `Plugins.MongoDB.Sessions.<session_name>.Uri=mongo` but provide the username and the password in the Zabbix agent 2 item key, the plugin will use the username and password from the item key, even if `Plugins.MongoDB.Sessions.<SessionName>.User` and `Plugins.MongoDB.Sessions.<SessionNam` are configured.

Attention:
If the `mongodb+srv://` scheme is used, the URI may retrieve parameters from a DNS TXT record. Parameters obtained from this TXT record override all other configurations, including URI parameters, Zabbix Agent 2 item key parameters, and session parameters.
 Only one TXT record may exist for the hostname used in the connection string.

Parameter	Mandatory	Range	Default	Description
Plugins.MongoDB.System.Path				Path to the MongoDB plugin executable. Example usage: `Plugins.MongoDB.System.Path=/usr/libexec/zabbix/zabbix-ag`
Plugins.MongoDB.Timeout		1-30	global timeout	Request execution timeout (the duration, in seconds, to wait for a request to complete before shutting it down).
Plugins.MongoDB.KeepAlive		60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.MongoDB.Sessions.<SessionName>.Uri			`tcp://127.0.0.1:27017`	Connection string of a named session. **<SessionName>** - define name of a session for using in item keys. Must match the URI format. Supported schemes: `mongodb://` and `mongodb+srv://`; the `tcp://` scheme is deprecated and kept for backward compatibility with existing configurations. A port can be omitted (default=27017). Examples: `mongodb://127.0.0.1:27017`, `mongodb+srv://example.com`, `localhost`.
Plugins.MongoDB.Sessions.<SessionName>.User				Named session username. **<SessionName>** - define name of a session for using in item keys.
Plugins.MongoDB.Sessions.<SessionName>.Password				Named session password. **<SessionName>** - define name of a session for using in item keys.

Parameter	Mandatory	Range	Default	Description
Plugins.MongoDB.Sessions.<SessionName>.Direct			false	Connection method used to determine whether automatic discovery of MongoDB replica set nodes should be enabled or disabled. **<SessionName>** - define name of a session for using in item keys. Supported values: `true` - the plugin will connect only to the specified server, ignoring other members of the replica set; `false` - the plugin will automatically discover all nodes in the replica set. If the `tcp://` (deprecated) scheme is used, the parameter is set to `true` by default.
Plugins.MongoDB.Sessions.<SessionName>.AuthMechanism				Authentication mechanism for connecting to MongoDB server. **<SessionName>** - define name of a session for using in item keys. Supported values: `SCRAM-SHA-1`, `SCRAM-SHA-256`, and `MONGODB-X509`.
Plugins.MongoDB.Sessions.<SessionName>.AuthSource			admin	Database for authenticating connection to MongoDB server. **<SessionName>** - define name of a session for using in item keys. If the `MONGODB-X509` authentication mechanism is used, this parameter is automatically set to `$external`; modifying it to any other value will cause an error when launching.
Plugins.MongoDB.Sessions.<SessionName>.ReplicaSet				Replica set name, used for automatic discovery of MongoDB replica set nodes. **<SessionName>** - define name of a session for using in item keys. If this parameter is used with the `mongodb+srv://` scheme, it will behave as if multiple URLs have been provided; a DNS server may have a TXT record that can override this (or any other) parameter. Unavailable if the `Plugins.MongoDB.Sessions.<SessionName>.Direct` parameter is set to `true`, the agent will fail to start.
Plugins.MongoDB.Sessions.<SessionName>.ReadPreference				Order for connecting to MongoDB replica set nodes or fallback. **<SessionName>** - define name of a session for using in item keys. Supported values: `PrimaryMode`, `PrimaryPreferredMode`, `SecondaryMode`, `SecondaryPreferredMode`, and `NearestMode`. Unavailable if the `Plugins.MongoDB.Sessions.<SessionName>.Direct` parameter is set to `true`, the agent will fail to start.
Plugins.MongoDB.Sessions.<SessionName>.TLSConnect				Encryption type for communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys. Supported values: `required` - require TLS connection; `verify\_ca` - verify certificates; `verify\_full` - verify certificates and IP address. Supported since plugin version 1.2.1.

Parameter	Mandatory	Range	Default	Description
Plugins.MongoDB.Sessions.<SessionName>.TLSCAFile	Yes, if Plugins.MongoDB.Sessions.<SessionName>.TLSConnect is set to one of: verify_ca, verify_full			Full pathname of a file containing the top-level CA(s) certificates for peer certificate verification, used for encrypted communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys.
Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile	Yes, if Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile is specified			Full pathname of a file containing the agent certificate or certificate chain, used for encrypted communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys.
Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile	Yes, if Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile is specified			Full pathname of a file containing the database private key used for encrypted communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys.
Plugins.MongoDB.Default.Uri			`tcp://127.0.0.1:27017`	Default URI for connecting to MongoDB. Must match the URI format. Supported schemes: `mongodb://` and `mongodb+srv://`. Deprecated scheme: `tcp://` (used as the default for backward compatibility with existing configurations). A port can be omitted (default=27017). Examples: `mongodb://127.0.0.1:27017`, `mongodb+srv://example.com`, `localhost`.
Plugins.MongoDB.Default.User				Default username for connecting to MongoDB.
Plugins.MongoDB.Default.Password				Default password for connecting to MongoDB.
Plugins.MongoDB.Default.Direct			`false`	Default connection method used to determine whether automatic discovery of MongoDB replica set nodes should be enabled or disabled. Supported values: `true` - the plugin will connect only to the specified server, ignoring other members of the replica set; `false` - the plugin will automatically discover all nodes in the replica set. If the `tcp://` (deprecated) scheme is used, the parameter is set to `true` by default.
Plugins.MongoDB.Default.AuthMechanism				Default authentication mechanism for connecting to MongoDB server. Supported values: `SCRAM-SHA-1`, `SCRAM-SHA-256`, and `MONGODB-X509`. If no value is specified either here, in an item key, in a named session, or in the URI, a SCRAM-based mechanism is used (if a username is provided; otherwise, no authentication will be used).
Plugins.MongoDB.Default.AuthSource			`admin`	Default database for authenticating connection to MongoDB server. If the `MONGODB-X509` authentication mechanism is used, this parameter is automatically set to `$external`; modifying it to any other value will cause an error when launching.
Plugins.MongoDB.Default.ReplicaSet				Default replica set name; used for automatic discovery of MongoDB replica set nodes. If this parameter is used with the `mongodb+srv://` scheme, it will behave as if multiple URLs have been provided; a DNS server may have a TXT record that can override this (or any other) parameter. Unavailable if the `Plugins.MongoDB.Default.Direct` parameter is set to `true`, the agent will fail to start.

Parameter	Mandatory	Range	Default	Description
Plugins.MongoDB.Default.ReadPreference				Default order for connecting to MongoDB replica set nodes or fallback. Supported values: `PrimaryMode`, `PrimaryPreferredMode`, `SecondaryMode`, `SecondaryPreferredMode`, and `NearestMode`. Unavailable if the `Plugins.MongoDB.Default.Direct` parameter is set to `true`, the agent will fail to start.
Plugins.MongoDB.Default.TLSConnect				Default encryption type for communications between Zabbix agent 2 and monitored databases. Supported values: `required` - requires TLS connection. `verify\_ca` - verifies certificates. `verify\_full` - verifies certificates and IP address. If no value is specified either here, in an item key, in a named session, or in the URI, a TLS connection will not be established.
Plugins.MongoDB.Default.TLSCAFile				Default pathname of a file containing the top-level CA(s) certificates for peer certificate verification. Used for encrypted communications between Zabbix agent 2 and monitored databases.
Plugins.MongoDB.Default.TLSCertFile				Default pathname of a file containing the MongoDB client certificate or certificate chain. Used for encrypted communications between Zabbix agent 2 and monitored databases.
Plugins.MongoDB.Default.TLSKeyFile				Default pathname of a file containing the MongoDB client private key used for encrypted communications between Zabbix agent 2 and monitored databases.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

7 MQTT plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the MQTT plugin.

It is recommended to specify them in their own configuration file (e.g. `mqtt.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.MQTT.Default.Password				Default password for connecting to MQTT; used if no value is specified in an item key or named session.
Plugins.MQTT.Default.TLSCAFile				Full pathname of a file containing the top-level CA(s) certificates for peer certificate verification for encrypted communications between Zabbix agent 2 and MQTT broker; used if no value is specified in a named session.

Parameter	Mandatory	Range	Default	Description
Plugins.MQTT.Default.TLSCertFile				Full pathname of a file containing the agent certificate or certificate chain for encrypted communications between Zabbix agent 2 and MQTT broker; used if no value is specified in a named session.
Plugins.MQTT.Default.TLSKeyFile				Full pathname of a file containing the MQTT private key for encrypted communications between Zabbix agent 2 and MQTT broker; used if no value is specified in a named session.

Plugins.MQTT.Default.Topic				Default topic for MQTT subscription; used if no value is specified in an item key or named session.
				
				The topic may contain wildcards (”+”,”#”)
				Examples: `path/to/file`
				`path/to/#`
				`path/+/topic`

Plugins.MQTT.Default.Url			tcp://localhost:1883	Default MQTT broker connection string; used if no value is specified in an item key or named session.
				
				Should not include query parameters.
				Must match the URL format.
				Supported schemes: `tcp` (default), `ws`, `tls`; a scheme can be omitted.
				A port can be omitted (default=1883).
				Examples: `tcp://host:1883`
				`localhost`
				`ws://host:8080`

Plugins.MQTT.Default.User				Default username for connecting to MQTT; used if no value is specified in an item key or named session.
Plugins.MQTT.Sessions.<SessionName>.Password				Named session password. **<SessionName>** - define name of a session for using in item keys.
Plugins.MQTT.Sessions.<SessionName>.TLSCAFile				Full pathname of a file containing the top-level CA(s) certificates for peer certificate verification, used for encrypted communications between Zabbix agent 2 and MQTT broker. **<SessionName>** - define name of a session for using in item keys.
Plugins.MQTT.Sessions.<SessionName>.TLSCertFile				Full pathname of a file containing the agent certificate or certificate chain, used for encrypted communications between Zabbix agent 2 and MQTT broker. **<SessionName>** - define name of a session for using in item keys.
Plugins.MQTT.Sessions.<SessionName>.TLSKeyFile				Full pathname of a file containing the MQTT private key used for encrypted communications between Zabbix agent 2 and MQTT broker. **<SessionName>** - define name of a session for using in item keys.

Plugins.MQTT.Sessions.<SessionName>.Topic				Named session topic for MQTT subscription.
				<SessionName> - define name of a session for using in item keys.
				
				The topic may contain wildcards (”+”,”#”)
				Examples: `path/to/file`
				`path/to/#`
				`path/+/topic`

Parameter	Mandatory	Range	Default	Description

Plugins.MQTT.Sessions.<SessionName>.Url				Connection string of a named session.
				<SessionName> - define name of a session for using in item keys.
				
				Should not include query parameters.
				Must match the URL format.
				Supported schemes: `tcp` (default), `ws`, `tls`; a scheme can be omitted.
				A port can be omitted (default=1883).
				Examples: `tcp://host:1883`
				`localhost`
				`ws://host:8080`

Plugins.MQTT.Sessions.<SessionName>.User				Named session username. **<SessionName>** - define name of a session for using in item keys.

If you need to set the request execution timeout (how long to wait for a request to complete before shutting it down), use the item configuration form.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

8 MSSQL plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the MSSQL plugin.

It is recommended to specify them in their own configuration file (e.g. `mssql.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

The MSSQL plugin is a loadable plugin and is available and fully described in the MSSQL plugin repository.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files.
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM.
- Comments starting with "#" are only supported at the beginning of the line.

Options

Parameter	Description
-V --version	Print the plugin version and license information.
-h --help	Print help information (shorthand).
-t, --test <item key>	Launch plugin for testing (plugin config ignored).

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.MSSQL.CustomQueriesDir			/usr/local/share/zabbix/custom-queries/mssql for Unix systems *:\Program Files\Zabbix Agent 2\Custom Queries\MSSQL for Windows systems, where * is the drive name from the ProgramFiles environment variable	Specifies the full path to a directory containing user-defined .sql files with custom queries that the plugin can execute. The plugin loads all available .sql files in the configured directory at startup. This means that any changes to the custom query files will not be reflected until the plugin is restarted. The plugin is started and stopped together with Zabbix agent 2.
Plugins.MSSQL.CustomQueriesEnabled			false	If set, enables the execution of the mssql.custom.query item key. If disabled, no queries will be loaded from the custom query directory path.
Plugins.MSSQL.Default.CACertPath				The default file path to the public key certificate of the certificate authority (CA) that issued the certificate of the MSSQL server. The certificate must be in PEM format.
Plugins.MSSQL.Default.Database				The default database name to connect to.
Plugins.MSSQL.Default.Encrypt				Specifies the default connection encryption type. Possible values are: *true* - data sending between plugin and server is encrypted; *false* - data sending between plugin and server is not encrypted beyond the login packet; *strict* - data sending between plugin and server is encrypted E2E using TDS8; *disable* - data sending between plugin and server is not encrypted.
Plugins.MSSQL.Default.HostNameInCertificate				The common name (CN) of the certificate of the MSSQL server by default.
Plugins.MSSQL.Default.Password				The password to be sent to a protected MSSQL server by default.
Plugins.MSSQL.Default.TLSMinVersion				The minimum TLS version to use by default. Possible values are: 1.0, 1.1, 1.2, 1.3.
Plugins.MSSQL.Default.TrustServerCertificate				Whether the plugin should trust the server certificate without validating it by default. Possible values: true, false.
Plugins.MSSQL.Default.Uri			sqlserver://localhost:1433	The default URI to connect. The only supported schema is sqlserver://. A schema can be omitted. Embedded credentials will be ignored.
Plugins.MSSQL.Default.User				The default username to be sent to a protected MSSQL server.
Plugins.MSSQL.KeepAlive		60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.MSSQL.Sessions.<SessionName>.CACertPath				The file path to the public key certificate of the certificate authority (CA) that issued the certificate of the MSSQL server for the named session. The certificate must be in PEM format. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.Database				The database name to connect to for the named session. **<SessionName>** - define name of a session for using in item keys.

Parameter	Mandatory	Range	Default	Description
Plugins.MSSQL.Sessions.<SessionName>.Encrypt				Specifies the connection encryption type for the named session. Possible values are: *true* - data sending between plugin and server is encrypted; *false* - data sending between plugin and server is not encrypted beyond the login packet; *strict* - data sending between plugin and server is encrypted E2E using TDS8; *disable* - data sending between plugin and server is not encrypted. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.HostNameInCertificate				The common name (CN) of the certificate of the MSSQL server for the named session. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.Password				The password to be sent to a protected MSSQL server for the named session. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.TLSMinVersion				The minimum TLS version to use for the named session. Possible values are: `1.0`, `1.1`, `1.2`, `1.3`. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.TrustServerCertificate				Whether the plugin should trust the server certificate without validating it for the named session. Possible values: `true`, `false`. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.Uri			sqlserver://localhost:1433	The URI to connect, for the named session. The only supported schema is `sqlserver://`. A schema can be omitted. Embedded credentials will be ignored. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.User				The username to be sent to a protected MSSQL server for the named session. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.System.Path				Path to the MSSQL plugin executable. Global setting for the MSSQL plugin. Applied to all connections. Example usage: `Plugins.MSSQL.System.Path=/usr/libexec/zabbix/zabbix-agen`
Plugins.MSSQL.Timeout		1-30	global timeout	The duration, in seconds, to wait for a server to respond when first connecting and on follow-up operations in the session.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

9 MySQL plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the MySQL plugin.

It is recommended to specify them in their own configuration file (e.g. `mysql.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;

- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Mysql.CallTimeout	no	1-30	global timeout	The maximum amount of time in seconds to wait for a request to be done.
Plugins.Mysql.CustomQueriesPath	no		`/usr/local/share/zabbix/custom-queries/mysql` for Unix systems `*:\Program Files\Zabbix Agent 2\Custom Queries\Mysql` for Windows systems, where `*` is the drive name from the `ProgramFiles` environment variable	Full pathname of a directory containing `.sql` files with custom queries.
Plugins.Mysql.CustomQueriesEnabled	no		false	If set, enables the execution of the `mysql.custom.query` item key. If disabled, no queries will be loaded from the custom query directory path.
Plugins.Mysql.Default.Password	no			Default password for connecting to MySQL; used if no value is specified in an item key or named session.
Plugins.Mysql.Default.TLSCAFile	no (yes, if Plugins.Mysql.Default.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the top-level CA(s) certificates for peer certificate verification for encrypted communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session.
Plugins.Mysql.Default.TLSCertFile	no (yes, if Plugins.Mysql.Default.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the agent certificate or certificate chain for encrypted communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session.
Plugins.Mysql.Default.TLSConnect	no			Encryption type for communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session. Supported values: *required* - require TLS connection; *verify_ca* - verify certificates; *verify_full* - verify certificates and IP address.
Plugins.Mysql.Default.TLSKeyFile	no (yes, if Plugins.Mysql.Default.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the database private key for encrypted communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session.

Parameter	Mandatory	Range	Default	Description

Plugins.Mysql.Default.Uri	no		tcp://localhost:3306	Default URI for connecting to MySQL; used if no value is specified in an item key or named session.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`; a scheme can be omitted.
				A port can be omitted (default=3306).
				Examples: `tcp://localhost:3306`
				`localhost`
				`unix:/var/run/mysql.sock`

Plugins.Mysql.Default.User	no			Default username for connecting to MySQL; used if no value is specified in an item key or named session.
Plugins.Mysql.KeepAlive	no	60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.Mysql.Sessions.<SessionName>.Password	no			Named session password. **<SessionName>** - define name of a session for using in item keys.
Plugins.Mysql.Sessions.<SessionName>.TLSCAFile	no (yes, if Plugins.Mysql.Sessions.<SessionName>.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the top-level CA(s) certificates for peer certificate verification, used for encrypted communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys.
Plugins.Mysql.Sessions.<SessionName>.TLSCertFile	yes, if Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile is specified			Full pathname of a file containing the agent certificate or certificate chain, used for encrypted communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys.
Plugins.Mysql.Sessions.<SessionName>.TLSConnect	no			Encryption type for communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys. Supported values: *required* - require TLS connection; *verify_ca* - verify certificates; *verify_full* - verify certificates and IP address.
Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile	yes, if Plugins.Mysql.Sessions.<SessionName>.TLSCertFile is specified			Full pathname of a file containing the database private key used for encrypted communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys.

Plugins.Mysql.Sessions.<SessionName>.Uri	no			Connection string of a named session.
				<SessionName> - define name of a session for using in item keys.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`; a scheme can be omitted.
				A port can be omitted (default=3306).
				Examples: `tcp://localhost:3306`
				`localhost`
				`unix:/var/run/mysql.sock`

Plugins.Mysql.Sessions.<SessionName>.User	no			Named session username. **<SessionName>** - define name of a session for using in item keys.
Plugins.Mysql.Timeout	no	1-30	global timeout	The maximum time in seconds for waiting when a connection has to be established.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

10 NVIDIA GPU plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the NVIDIA GPU plugin.

It is recommended to specify them in their own configuration file (e.g. `nvidiagpu.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

The NVIDIA GPU plugin is a loadable plugin and is available and fully described in the NVIDIA GPU plugin repository.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.NVIDIA.System.Path				Path to the NVIDIA GPU plugin executable. Example usage: `Plugins.NVIDIA.System.Path=/path/to/executable/nvidia`
Plugins.NVIDIA.Timeout		1-30	global timeout	Specifies the maximum time (in seconds) to wait for a server response during connection attempts and subsequent operations in the session. The global item-type timeout or individual item timeout will override this value if greater.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

11 Oracle plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the Oracle plugin.

It is recommended to specify them in their own configuration file (e.g. `oracle.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files.
- Values support environment variables.
- Zabbix supports configuration files only in UTF-8 encoding without BOM.
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Oracle.CallTimeout		1-30	global timeout	The maximum wait time in seconds for a request to be completed.
Plugins.Oracle.ConnectTimeout		1-30	global timeout	The maximum wait time in seconds for a connection to be established.

Parameter	Mandatory	Range	Default	Description
Plugins.Oracle.CustomQueriesPath			/usr/local/share/zabbix/custom_queries/oracle for Unix systems *:\Program Files\Zabbix Agent 2\Custom Queries\Oracle for Windows systems, where * is the drive name from the ProgramFiles environment variable	Full path to a directory containing .sql files with custom queries. Example: /etc/zabbix/oracle/sql
Plugins.Oracle.CustomQueriesEnabled			false	If set, enables the execution of the oracle.custom.query item key. If disabled, no queries will be loaded from the custom query directory path.
Plugins.Oracle.Default.Password				Default password for connecting to Oracle; used if no value is specified in an item key or named session.
Plugins.Oracle.Default.Service				Default service name for connecting to Oracle (SID is not supported); used if no value is specified in an item key or named session.
Plugins.Oracle.Default.Uri				Default URI for connecting to Oracle; used if no value is specified in an item key or named session. Should not include embedded credentials (they will be ignored). Must match the URI format. Only tcp and tcps schemes are supported; a scheme can be omitted. A port can be omitted (default=1521). It is also possible to specify the TNS key or TNS value as the connection string. The TNS value must be composed without whitespaces. Examples: tcp://127.0.0.1:1521 localhost zbx_tns_example (TNS key) (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(POR (TNS value)
Plugins.Oracle.Default.User				Default username for connecting to Oracle; used if no value is specified in an item key or named session.
Plugins.Oracle.KeepAlive		60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.Oracle.ResolveTNS		true / false	true	The option specifies how to interpret the connection string (ConnString) for connecting to the Oracle server. If it is set to false the default scheme and port will be added to ConnString (if absent). If the option is set to true, the default scheme and port will be omitted (unless explicitly specified in ConnString), and ConnString will be passed to the Oracle client as is. If the Oracle client finds ConnString in the tnsnames.ora file, the connection description found will be used to connect to the Oracle server.
Plugins.Oracle.Sessions.<SessionName>.Password				Named session password. **<SessionName>** - define name of a session for using in item keys.

Parameter	Mandatory	Range	Default	Description
Plugins.Oracle.Sessions.<SessionName>.Service	no			Named session service name to be used for connection (SID is not supported). **<SessionName>** - define name of a session for using in item keys.
Plugins.Oracle.Sessions.<SessionName>.Uri	no			Named session connection string for Oracle. **<SessionName>** - define name of a session for using in item keys. Should not include embedded credentials (they will be ignored). Must match the URI format. Only `tcp` and `tcps` schemes are supported; a scheme can be omitted. A port can be omitted (default=1521). It is also possible to specify the TNS key or TNS value as the connection string. The TNS value must be composed without whitespaces. Examples: `tcp://127.0.0.1:1521` `localhost` `zbx_tns_example` (TNS key) `(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(POR` (TNS value)
Plugins.Oracle.Sessions.<SessionName>.User	no			Named session username. **<SessionName>** - define name of a session for using in item keys.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

12 PostgreSQL plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the PostgreSQL plugin.

It is recommended to specify them in their own configuration file (e.g. `postgresql.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

The PostgreSQL plugin is a loadable plugin and is available and fully described in the PostgreSQL plugin repository.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files.
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM.
- Comments starting with "#" are only supported at the beginning of the line.

Options

Parameter	Description
-V --version	Print the plugin version and license information.
-h --help	Print help information (shorthand).
-t, --test <item key>	Launch plugin for testing (plugin config ignored).

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.PostgreSQL.Default.CacheMode			prepare	Cache mode for the PostgreSQL connection. Supported values: *prepare* (default) - will create prepared statements on the PostgreSQL server; *describe* - will use the anonymous prepared statement to describe a statement without creating a statement on the server. Note that ”describe” is primarily useful when the environment does not allow prepared statements such as when running a connection pooler like PgBouncer.
Plugins.PostgreSQL.CallTimeout		1-30	global timeout	Maximum wait time (in seconds) for a request to be completed.
Plugins.PostgreSQL.CustomQueriesPath			`/usr/local/share/zabbix/custom_queries/postgresql` for Unix systems `*:\Program Files\Zabbix Agent 2\Custom Queries\PostgreSQL` for Windows systems, where `*` is the drive name from the `ProgramFiles` environment variable	Full pathname of a directory containing files with custom queries.
Plugins.PostgreSQL.CustomQueriesEnabled			false	If set, enables the execution of the `postgresql.custom.query` item key. If disabled, no queries will be loaded from the custom query directory path.
Plugins.PostgreSQL.Default.Database				Default database for connecting to PostgreSQL; used if no value is specified in an item key or named session.
Plugins.PostgreSQL.Default.Password				Default password for connecting to PostgreSQL; used if no value is specified in an item key or named session.
Plugins.PostgreSQL.Default.TLSCAFile	(yes, if Plugins.PostgreSQL.Default.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the top-level CA(s) certificate for peer certificate verification for encrypted communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session.
Plugins.PostgreSQL.Default.TLSCertFile	(yes, if Plugins.PostgreSQL.Default.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the PostgreSQL certificate or certificate chain for encrypted communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session.
Plugins.PostgreSQL.Default.TLSConnect				Encryption type for communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session. Supported values: *required* - connect using TLS as transport mode without identity checks; *verify_ca* - connect using TLS and verify certificate; *verify_full* - connect using TLS, verify certificate and verify that database identity (CN) specified by DBHost matches its certificate. Undefined encryption type means unencrypted connection.

Parameter	Mandatory	Range	Default	Description
Plugins.PostgreSQL.Default.TLSKeyFile	No (yes, if Plugins.PostgreSQL.Default.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the PostgreSQL private key for encrypted communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session.

Plugins.PostgreSQL.Default.Uri	No			Default URI for connecting to PostgreSQL; used if no value is specified in an item key or named session.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`.
				Examples: `tcp://127.0.0.1:5432`
				`tcp://localhost`
				`unix:/var/run/postgresql/.s.PGSQL.5432`

Plugins.PostgreSQL.Default.User	No			Default username for connecting to PostgreSQL; used if no value is specified in an item key or named session.
Plugins.PostgreSQL.KeepAlive	No	60-900	300	Maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.PostgreSQL.Sessions.<SessionName>.CachePreparedMode	No		prepare	Cache mode for the PostgreSQL connection. **<SessionName>** - define name of a session for using in item keys. Supported values: *prepare* (default) - will create prepared statements on the PostgreSQL server; *describe* - will use the anonymous prepared statement to describe a statement without creating a statement on the server. Note that ”describe” is primarily useful when the environment does not allow prepared statements such as when running a connection pooler like PgBouncer.
Plugins.PostgreSQL.Sessions.<SessionName>.Database	No			Database for session connection. **<SessionName>** - define name of a session for using in item keys.
Plugins.PostgreSQL.Sessions.<SessionName>.Password	No	Must match the password format.		Password for session connection. **<SessionName>** - define name of a session for using in item keys.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSCAFile	No (yes, if Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the top-level CA(s) certificate peer certificate verification. **<SessionName>** - define name of a session for using in item keys.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile	Yes, if Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile is specified			Full pathname of a file containing the PostgreSQL certificate or certificate chain. **<SessionName>** - define name of a session for using in item keys.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect	No			Encryption type for PostgreSQL connection. **<SessionName>** - define name of a session for using in item keys. Supported values: *required* - connect using TLS as transport mode without identity checks; *verify_ca* - connect using TLS and verify certificate; *verify_full* - connect using TLS, verify certificate and verify that database identity (CN) specified by DBHost matches its certificate. Undefined encryption type means unencrypted connection.

Parameter	Mandatory	Range	Default	Description
Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile	yes, if Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile is specified			Full pathname of a file containing the PostgreSQL private key. **<SessionName>** - define name of a session for using in item keys.

Plugins.PostgreSQL.Sessions.<SessionName>.Uri				Connection string of a named session.
				<SessionName> - define name of a session for using in item keys.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`.
				Examples: `tcp://127.0.0.1:5432`
				`tcp://localhost`
				`unix:/var/run/postgresql/.s.PGSQL.5432`

Plugins.PostgreSQL.Sessions.<SessionName>.User				Named session username. **<SessionName>** - define name of a session for using in item keys.
Plugins.PostgreSQL.System.Path	yes			Path to the PostgreSQL plugin executable. Example usage: `Plugins.PostgreSQL.System.Path=/usr/libexec/zabbix/zabbi`
Plugins.PostgreSQL.Timeout		1-30	global timeout	Request execution timeout (the duration, in seconds, to wait for a request to complete before shutting it down).

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

13 Redis plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the Redis plugin.

It is recommended to specify them in their own configuration file (e.g. `redis.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files.
- Values support environment variables.
- Zabbix supports configuration files only in UTF-8 encoding without BOM.
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Redis.Default.Password				Default password for connecting to Redis; used if no value is specified in an item key or named session.
Plugins.Redis.Default.TLSConnect				Client (Zabbix agent 2) TLS verification requirement. Supported values: `required` - encrypted, unverified (only for testing) `verify_ca` - encrypted, server certificate `verify_full` - encrypted, server certificate and server name verified with SAN An empty or unset value is treated as no tls.
Plugins.Redis.Default.TLSCAFile				Default full pathname of a file containing the top-level CA(s) certificates.
Plugins.Redis.Default.TLSCertFile				Default full pathname of a file containing the agent's certificate.

Parameter	Mandatory	Range	Default	Description
Plugins.Redis.Default.TLSKeyFile				Default full pathname of a file containing the agent's private key.

Plugins.Redis.Default.Uri			tcp://localhost:6379	Default URI for connecting to Redis; used if no value is specified in an item key or named session.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`; a scheme can be omitted.
				A port can be omitted (default=6379).
				Examples: `tcp://localhost:6379`
				`localhost`
				`unix:/var/run/redis.sock`

Plugins.Redis.Default.User			default	Default user to send to the protected Redis server; used if no value is specified in an item key or named session.
Plugins.Redis.KeepAlive		60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.Redis.Sessions.<SessionName>.Password				Named session password. **<SessionName>** - define the session name to be used in item keys.
Plugins.Redis.Sessions.<SessionName>.TLSConnect				Client (Zabbix agent 2) TLS verification requirement. **<SessionName>** - define the session name to be used in item keys. Supported values: `required` - encrypted, unverified (only for testing) `verify_ca` - encrypted, server certificate `verify_full` - encrypted, server certificate and server name verified with SAN An empty or unset value is treated as no tls.
Plugins.Redis.Sessions.<SessionName>.TLSCAFile	(yes, if TLSConnect is `verify_ca` or `verify_full`)			Full pathname of a file containing the top-level CA(s) certificates for Redis server certificate verification. **<SessionName>** - define the session name to be used in item keys.
Plugins.Redis.Sessions.<SessionName>.TLSCertFile				Full pathname of a file containing the agent's certificate for client authentication. **<SessionName>** - define the session name to be used in item keys.
Plugins.Redis.Sessions.<SessionName>.TLSKeyFile				Full pathname of a file containing the agent's private key for client authentication. **<SessionName>** - define the session name to be used in item keys.

Plugins.Redis.Sessions.<SessionName>.Uri			localhost:6379	Connection string of a named session.
				<SessionName> - define the session name to be used in item keys.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`; a scheme can be omitted.
				A port can be omitted (default=6379).
				Examples: `tcp://localhost:6379`
				`localhost`
				`unix:/var/run/redis.sock`

Plugins.Redis.Sessions.<SessionName>.User			default	User to send to the protected Redis server. **<SessionName>** - define the session name to be used in item keys.

Parameter	Mandatory	Range	Default	Description
Plugins.Redis.Timeout	no	1-30	global timeout	Request execution timeout (the duration, in seconds, to wait for a request to complete before shutting it down).

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

14 SMART plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the SMART plugin.

It is recommended to specify them in their own configuration file (e.g. `smart.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;
- Values support environment variables;
- The path to the `smartctl` executable must be set either by adding its directory to the system's `PATH` environment variable or by configuring `Plugins.Smart.Path`; this applies to both Linux and Windows;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Smart.Path	no		smartctl	Path to the smartctl executable.
Plugins.Smart.Timeout	no	1-30	global timeout	Request execution timeout (the duration, in seconds, to wait for a request to complete before shutting it down).

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

8 Passerelle Java Zabbix

Si vous utilisez les scripts `startup.sh` et `shutdown.sh` pour démarrer la passerelle Java Zabbix, alors vous pouvez spécifier les paramètres de configuration nécessaires dans le fichier `settings.sh`. Les scripts de démarrage et d'arrêt extraient le fichier de paramètres et prennent en charge la conversion des variables shell (répertoriées dans la première colonne) en propriétés Java (répertoriées dans la deuxième colonne).

Si vous démarrez manuellement la passerelle Java Zabbix en exécutant directement `java`, indiquez les propriétés Java correspondantes sur la ligne de commande.

Variable	Propriété	Obligatoire	Range	Défaut	Description
LISTEN_IP	zabbix.listenIP	non		0.0.0.0	Adresse IP à écouter.
LISTEN_PORT	zabbix.listenPort	non	1024-32767	10052	Port à écouter.
PID_FILE	zabbix.pidFile	non		/tmp/zabbix_java.pid	Nom du fichier PID. S'il est omis, la passerelle Java Zabbix est démarrée en tant qu'application console.

Variable	Propriété	Obligatoire	Range	Défaut	Description
PROPERTIES_FILE	zabbix.propertiesFile	non			Nom du fichier de propriétés. Peut être utilisé pour définir des propriétés supplémentaires à l'aide d'un format clé-valeur de manière à ce qu'elles ne soient pas visibles sur une ligne de commande ou pour écraser celles existantes. Par exemple : "javax.net.ssl.trustStorePassword=<
START_POLLERS	zabbix.startPollers	non	1-1000	5	Nombre de threads de travail à démarrer.
TIMEOUT	zabbix.timeout	non	1-30	3	Combien de temps attendre pour les opérations du réseau.

Warning:
Le port 10052 n'est pas enregistré IANA.

9 Web service Zabbix

Aperçu

Le service web Zabbix est un processus utilisé pour communiquer 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 afficher tous les détails.

Parameter	Description
AllowedIP	Une liste d'adresses IP séparées par des virgules, éventuellement en notation CIDR, ou de noms DNS de serveurs Zabbix et de proxys Zabbix.
DebugLevel	Le niveau de débogage.
IgnoreURLCertErrors	Spécifie la gestion des erreurs de validation du certificat TLS lors de l'accès à l'URL du frontend.
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 à la mise en forme du rapport PDF d'un tableau de bord.
TLSAccept	Les connexions entrantes à accepter.
TLSCAFile	Le chemin complet d'un fichier contenant les certificats de la ou des autorités de certification 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 les valeurs des fichiers de configuration fournis ;
- Les valeurs prennent en charge les variables d'environnement ;
- 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 des paramètres

AllowedIP

Une liste d'adresses IP séparées par des virgules, éventuellement en notation CIDR, ou de noms DNS de serveurs Zabbix et de proxies Zabbix. 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 traitées de manière identique 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 :

`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 sur les erreurs ;
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 du frontend :
0 - ne pas ignorer les erreurs de certificat ;
1 - ignorer les erreurs de certificat.

Par défaut : 0
 Plage : 0-1

Include

Vous pouvez inclure des fichiers individuels ou tous les fichiers d'un répertoire dans le fichier de configuration. Lors de l'installation, Zabbix créera le répertoire d'inclusion dans /usr/local/etc, sauf s'il a été modifié au moment de la compilation. Le chemin peut être relatif à l'emplacement du fichier *zabbix_web_service.conf*.
Pour inclure uniquement les fichiers pertinents dans le répertoire spécifié, le caractère générique astérisque est pris en charge pour la correspondance de motifs.
Voir les notes spéciales concernant les limitations.

Exemple :

```
Include=/absolute/path/to/config/files/*.conf
```

ListenPort

Le service écoutera sur ce port 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 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, pour quelque raison que ce soit, le fichier journal existant est tronqué et redémarré à zéro.

Par défaut : 1
 Plage : 0-1024

LogType

Le type de la 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`

Délai d'expiration

Le temps maximal (en secondes) consacré au formatage du report PDF 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 les certificats de la ou des autorités de certification (CA) de niveau supérieur pour la vérification du certificat du pair, utilisé 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 complet du fichier contenant la clé privée du service, utilisée pour les communications chiffrées entre les composants Zabbix.

10 Variables d'environnement

Aperçu

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 les variables peuvent être transmises au moment de l'exécution pour s'adapter 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 :

- Server
- Proxy
- Agent (UNIX ou Windows)
- Agent 2 (UNIX ou Windows), y compris les plugins
- Web service
- Zabbix sender (lors de l'utilisation de l'option -c, --config)

Remarques 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 des commandes d'exécution (par exemple, pour augmenter le niveau de journalisation de l'agent), toutes les variables d'environnement précédemment utilisées doivent être spécifiées. Cela s'explique par le fait que les composants Zabbix utilisent leur 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. Voir Exemples.
- 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.
- 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 ne peut contenir que des lettres (a-z, A-Z), des traits de soulignement (_), 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 produire 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
```

Lors de l'utilisation des commandes d'exécution (par exemple, pour augmenter le niveau de journalisation de l'agent), toutes les variables d'environnement précédemment utilisées 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.

Sinon, après avoir défini les variables d'environnement dans le fichier de configuration de l'agent, vous pouvez les rendre disponibles aux processus (par exemple, en utilisant la commande `export`). Cela réduit le risque de comportement inattendu dû à des variables manquantes ou incorrectement définies.

```
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 des composants Zabbix (par exemple, l'agent Zabbix), vous pouvez définir des 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 des variables d'environnement :

```
docker run --name my-zabbix-agent -e ZBX_HOSTNAME="new-hostname" -e BUFSZ=1000 -e LISTENPORT=20050 -e _Usr
```

3. Lors de l'utilisation 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 avec des valeurs normales sont rechargés.

11 Inclusion

Vue d'ensemble

Des fichiers ou répertoires supplémentaires peuvent être inclus dans la configuration du serveur, du proxy, de l'agent et du service web à l'aide du paramètre `Include`.

Remarques sur l'inclusion

Si le paramètre `Include` est utilisé pour inclure un fichier, le fichier doit être lisible.

Si le paramètre `Include` est utilisé pour inclure un répertoire : - Tous les fichiers du répertoire doivent être lisibles. - Aucun ordre particulier d'inclusion ne doit être supposé (par exemple, les fichiers ne sont pas inclus dans l'ordre alphabétique). Ne définissez donc pas un paramètre dans plusieurs fichiers "Include" (par exemple pour remplacer un paramètre général par un paramètre spécifique). - Tous les fichiers du répertoire sont inclus dans la configuration. - Méfiez-vous des copies de sauvegarde de fichiers créées automatiquement par certains éditeurs de texte. Par exemple, si l'édition du fichier "include/my_specific.conf" produit une copie de sauvegarde "include/my_specific_conf.BAK" alors les deux fichiers seront inclus. Déplacez "include/my_specific.conf.BAK" hors du répertoire "Include". Sous Linux, le contenu du répertoire "Include" peut être vérifié avec une commande "ls -al" pour les fichiers inutiles.

Si le paramètre `Include` est utilisé pour inclure des fichiers à l'aide d'un modèle :

- `Tous les fichiers correspondant au modèle doivent être lisibles.`
- `Aucun ordre particulier d'inclusion ne doit être supposé (par exemple, les fichiers ne sont pas inclus`

3 Protocoles

Veuillez utiliser la barre latérale pour accéder au contenu de cette section.

1 Protocole d'échange de données serveur-proxy

Aperçu

L'échange de données serveur - proxy est basé sur le format JSON.

Les messages de requête et de réponse doivent commencer par un en-tête et une longueur des données.

Proxy passif

Requête de configuration

Le serveur enverra 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 configuration. Le serveur répond avec les données de configuration qui doivent être mises à jour.

name	value type	description
server→proxy:		

name	value type	description
request	*string*	'proxy config'
proxy→server:		
version	*string*	Version du proxy (<major>.<minor>.<build>).
session	*string*	Jeton de session de configuration du proxy.
config_revision	*number*	Révision de configuration du proxy.
server→proxy:		
full_sync	*number*	1 - si les données de configuration complètes sont envoyées ; absent - sinon (facultatif).
data	*array*	Objet des données de table. Absent si la configuration n'a pas été modifiée (facultatif).
<table>	*object*	Un ou plusieurs objets avec les données de <table> (facultatif, selon les modifications).
fields	*array*	Tableau des noms de champs.
-	*string*	Nom du champ.
data	*array*	Tableau des lignes.
-	*array*	Tableau des colonnes.
-	*string*,*number*	Valeur de colonne avec un type dépendant du type de colonne dans le schéma de base de données.
macro.secrets	*object*	Informations sur les macros secrètes, absent s'il n'y a aucune modification dans les macros du coffre-fort (facultatif).
config_revision	*number*	Révision du cache de configuration - envoyée avec les données de configuration (facultatif).
del_hostids	*array*	Tableau des hostid supprimés (facultatif).
-	*number*	Identifiant d'hôte.
del_macro_hostids	*array*	Tableau des hostid dont toutes les macros ont été supprimées (facultatif).
-	*number*	Identifiant d'hôte.
proxy→server:		
response	*string*	Informations sur le succès de la requête ('success' ou 'failed').
version	*string*	Version du proxy (<major>.<minor>.<build>).

Exemple :

server→proxy:

```
{
  "request":"proxy config"
}
```

proxy→server:

```
{
  "version": "8.0.0",
  "session": "0033124949800811e5686dbfd9bcea98",
  "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, "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "5
]
},
"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
}
```


proxy→server:


```
{
  "response": "success",
  "version": "8.0.0"
}
```


Requête de données

La requête `proxy data` est utilisée pour obtenir depuis le proxy les données de disponibilité des interfaces hôte, les données d'historique, de découverte et d'auto-enregistrement. Cette requête est envoyée toutes les `ProxyDataFrequency` (paramètre de configuration du serveur) secondes.

name	value type	description
server→proxy:		
request	*string*	'proxy data'
proxy→server:		
session	*string*	Jeton de session des données.
interface availability	*array*	*(optional)* Tableau d'objets de données 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 dans une session de données).
timestamp	*number*	*(optional)* Horodatage des éléments de type journal.
source	*string*	*(optional)* Valeur source de l'élément eventlog.
severity	*number*	*(optional)* Valeur de gravité de l'élément eventlog.

name	value type	description
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 de la vérification de découverte ou null pour les données de règle de découverte.
type	*number*	Type de vérification de découverte : **-1** données de règle de découverte **0**, *SVC_SSH* - vérification du service SSH **1**, *SVC_LDAP* - vérification du service LDAP **2**, *SVC_SMTP* - vérification du service SMTP **3**, *SVC_FTP* - vérification du service FTP **4**, *SVC_HTTP* - vérification du service HTTP **5**, *SVC_POP* - vérification du service POP **6**, *SVC_NNTP* - vérification du service NNTP **7**, *SVC_IMAP* - vérification du service IMAP **8**, *SVC_TCP* - vérification 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* - vérification du service HTTPS **15**, *SVC_TELNET* - vérification 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 d'élément pour une vérification 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'auto-enregistrement.
clock	*number*	Horodatage des données d'auto-enregistrement.
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

name	value type	description
status	*number*	État d'exécution de la commande distante : **0**, *ZBX_TM_REMOTE_COMMAND_COMPLETED* - commande distante terminée 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 davantage de 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 commande 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*	*(facultatif)* Horodatage du transfert de données (secondes).
ns	*number*	*(facultatif)* Horodatage du transfert de données (nanosecondes).
server→proxy:		
response	*string*	Information de réussite de la requête ('success' ou '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 un proxy actif afin d'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 sinon (facultatif).
data	*array*	Objet des données de table. Absent si la configuration n'a pas été modifiée (facultatif).
<table>	*object*	Un ou plusieurs objets avec les données de <table> (facultatif, selon les modifications).
fields	*array*	Tableau des noms de champs.
-	*string*	Nom du champ.
data	*array*	Tableau des lignes.
-	*array*	Tableau des colonnes.
-	*string,number*	Valeur de colonne avec un type dépendant du type de colonne dans le schéma de base de données.

name	value type	description
macro.secrets	*object*	Informations sur les macros secrètes, absent s'il n'y a aucune modification dans les macros du coffre-fort (facultatif).
proxy_group	*string*	Nom du groupe de proxys auquel appartient le proxy.
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 d'hôte.
del_macro_hostids	*array*	Tableau des hostids dont toutes les macros ont été supprimées (facultatif).
-	*number*	Identifiant d'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, "", "", "",
```

```
[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
}
```


Requête de données

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'auto-enregistrement. Cette requête est envoyée toutes les `DataSenderFrequency` secondes (paramètre de configuration du proxy). Notez qu'un proxy actif interrogera tout de même 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 des données.
interface availability	*array*	*(facultatif)* Tableau d'objets de données 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*	*(facultatif)* 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*	*(facultatif)* Valeur de l'élément.
id	*number*	Identifiant de la valeur (compteur croissant, unique dans une session de données).
timestamp	*number*	*(facultatif)* Horodatage des éléments de type journal.
source	*string*	*(facultatif)* Valeur source de l'élément eventlog.
severity	*number*	*(facultatif)* Valeur de sévérité de l'élément eventlog.
eventid	*number*	*(facultatif)* Valeur eventid de l'élément eventlog.
state	*string*	*(facultatif)* État de l'élément : **0**, *ITEM_STATE_NORMAL* **1**, *ITEM_STATE_NOTSUPPORTED*
lastlogsize	*number*	*(facultatif)* Dernière taille du journal des éléments de type journal.
mtime	*number*	*(facultatif)* Heure de modification des éléments de type journal.
discovery data	*array*	*(facultatif)* 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 de la vérification de découverte ou null pour les données de règle de découverte.
type	*number*	Type de vérification de découverte : **-1** données de règle de découverte **0**, *SVC_SSH* - vérification du service SSH **1**, *SVC_LDAP* - vérification du service LDAP **2**, *SVC_SMTP* - vérification du service SMTP **3**, *SVC_FTP* - vérification du service FTP **4**, *SVC_HTTP* - vérification du service HTTP **5**, *SVC_POP* - vérification du service POP **6**, *SVC_NNTP* - vérification du service NNTP **7**, *SVC_IMAP* - vérification du service IMAP **8**, *SVC_TCP* - vérification 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* - vérification du service HTTPS **15**, *SVC_TELNET* - vérification de disponibilité Telnet

name	value type	description
ip	*string*	Adresse IP de l'hôte.
dns	*string*	Nom DNS de l'hôte.
port	*number*	*(facultatif)* Numéro de port du service.
key_	*string*	*(facultatif)* Clé d'élément pour une vérification de découverte de type **9** *SVC_AGENT*
value	*string*	*(facultatif)* Valeur reçue du service, peut être vide pour la plupart des services.
status	*number*	*(facultatif)* État du service : **0**, *DOBJECT_STATUS_UP* - service UP **1**, *DOBJECT_STATUS_DOWN* - service DOWN
autoregistration	*array*	*(facultatif)* Tableau d'objets de données d'auto-enregistrement.
clock	*number*	Horodatage des données d'auto-enregistrement.
host	*string*	Nom de l'hôte.
ip	*string*	*(facultatif)* Adresse IP de l'hôte.
dns	*string*	*(facultatif)* Nom DNS résolu à partir de l'adresse IP.
port	*string*	*(facultatif)* Port de l'hôte.
host_metadata	*string*	*(facultatif)* Métadonnées d'hôte envoyées par l'agent (sur la base du paramètre de configuration de l'agent HostMetadata ou HostMetadataItem).
tasks	*array*	*(facultatif)* 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 terminée avec succès **1**, *ZBX_TM_REMOTE_COMMAND_FAILED* - échec de la commande distante
error	*string*	*(facultatif)* Message d'erreur.
parent_taskid	*number*	ID de la tâche parente.
more	*number*	*(facultatif)* 1 - il y a davantage de données d'historique à envoyer
clock	*number*	*(facultatif)* Horodatage du transfert de données (secondes).
ns	*number*	*(facultatif)* 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 du téléversement des données historiques (historique, auto-enregistrement, disponibilité des hôtes, découverte réseau). Valeurs possibles : **enabled** - fonctionnement normal **disabled** - le serveur n'accepte pas les données (éventuellement en raison d'un dépassement de la limite du cache interne)
tasks	*array*	*(facultatif)* 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*	*(facultatif)* Port pour les commandes Telnet et SSH.
authtype	*number*	*(facultatif)* Type d'authentification pour les commandes SSH.
username	*string*	*(facultatif)* Nom d'utilisateur pour les commandes Telnet et SSH.
password	*string*	*(facultatif)* Mot de passe pour les commandes Telnet et SSH.
publickey	*string*	*(facultatif)* Clé publique pour les commandes SSH.
privatekey	*string*	*(facultatif)* 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
}]
}
```


2 Protocole Zabbix agent/agent2

Veuillez consulter la page Vérifications passives et actives de l'agent pour plus d'informations sur les protocoles de Zabbix agent et Zabbix agent 2.

4 Protocole du plug-in agent Zabbix 2

Le protocole de l'agent Zabbix 2 est basé sur le code, la taille et le modèle de données.

Code

Type	Size	Commentaires
Byte	4	Type de charge utile, actuellement seul JSON est pris en charge.

Taille

Type	Taille	Commentaires
Octet	4	Taille de la charge utile actuelle en octets.

Données de charge utile

Type	Taille	Commentaires
Octet	Défini par le champ *Taille*	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 incrémentiel utilisé pour associer les requêtes aux réponses. Il est unique dans une direction de requête donnée (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

Une requête envoyée par un plugin pour écrire un message de journal dans le fichier journal de l'agent.

direction	plugin → agent
response	non

Paramètres spécifiques aux requêtes de journalisation :

Name	Type	Comments
severity	uint32	La sévérité du message (niveau de journalisation).
message	string	Le message à journaliser.

Exemple :

```
{"id":0,"type":1,"severity":3,"message":"message"}
```

Requête d'enregistrement

Une 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.
metrics	array of strings (optional)	Les métriques avec leurs descriptions telles qu'utilisées dans le plugin. Renvoie 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"}
```

Requête de démarrage

Une requête pour exécuter la fonction Start de l'interface Runner.

direction	agent → plugin
response	no

La requête n'a pas de paramètres spécifiques, elle contient uniquement les paramètres des données communes.

Exemple :

```
{"id":3,"type":4}
```

Requête de terminaison

Une requête envoyée par l'agent pour arrêter un plugin.

direction	agent → plugin
response	no

La requête n'a pas de paramètres spécifiques, elle contient uniquement les paramètres des données communes.

Exemple :

```
{"id":3,"type":5}
```

Requête d'export

Une requête pour exécuter la fonction Export de l'interface Exporter.

direction	agent → plugin
response	non

Paramètres spécifiques aux requêtes 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. Absent 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 elles sont 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	JSON object (optional)	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 de *Validate* :

Name	Type	Comments
error	string (optional)	Message d'erreur renvoyé si la fonction Validate ne s'exécute pas correctement. Absent si l'exécution réussit.

Exemple :

```
{"id":8,"type":10}
```

ou

```
{"id":8,"type":10,"error":"error message"}
```

5 Protocole Zabbix sender

Vue d'ensemble

Zabbix server et Zabbix proxy 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 le mode trapping activé.

Les messages de requête et de réponse doivent commencer par un en-tête et une longueur de données.

Requête Zabbix sender

```
{
    "request": "sender data",
    "data": [
        {
            "host": "<hostname>",
            "key": "trap",
            "value": "test value"
        }
    ]
}
```

Réponse du serveur Zabbix

```
{
    "response": "success",
    "info": "traité : 1 ; échec : 0 ; total : 1 ; secondes écoulées : 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"
}
```

6 En-tête

Aperçu

L'en-tête est présent dans tous les messages de requête et de réponse entre les composants Zabbix. Il est nécessaire de déterminer la longueur du message, s'il est compressé ou non, s'il s'agit d'un gros paquet ou non.

Le protocole de communication Zabbix a 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 au proxy Zabbix, la limite de taille de paquet est augmentée à 4 Go pour permettre la synchronisation de grandes configurations. Lorsque la longueur des données avant compression dépasse 4 Go, le serveur Zabbix démarre automatiquement en utilisant le grand format de paquet (flag 0x04) qui augmente la limite de taille de paquet à 16 Go.

Notez que bien qu'un grand format de paquet puisse être utilisé pour envoyer des données, actuellement seul le synchroniseur de configuration du proxy Zabbix peut gérer les paquets supérieurs à 1 Go.

Structure

L'en-tête se compose de quatre champs. Tous les nombres de l'en-tête sont formatés en little-endian.

Champ	Taille	Taille (paquet volumineux)	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

Voici quelques extraits de code montrant comment ajouter un en-tête de protocole Zabbix aux données que vous souhaitez envoyer dans l'ordre pour obtenir le paquet que vous devez envoyer à Zabbix afin qu'il soit interprété correctement. Ces extraits de code supposent que les données ne dépassent pas 1 Go, le format de gros paquets n'est donc 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"
```

7 Protocole d'export JSON délimité par des sauts de ligne

Cette section présente les détails du protocole d'export au format JSON délimité par des sauts de ligne, utilisé dans :

- l'exportation des données vers des fichiers
- le streaming vers des systèmes externes

Les éléments suivants peuvent être exportés :

- événements de déclencheur
- valeurs d'élément
- tendances (exportation vers des fichiers uniquement)

Tous les fichiers ont l'extension .ndjson. Chaque ligne du fichier d'export est un objet JSON.

Événements de déclencheur

Les informations suivantes sont exportées pour un événement de problème :

Champ		Type	Description
clock		nombre	Nombre de secondes depuis Epoch jusqu'au moment où le problème a été détecté (partie entière).
ns		nombre	Nombre de nanosecondes à ajouter à `clock` pour obtenir une heure précise de détection du problème.
valeur		nombre	1 (toujours).
eventid		nombre	ID d'événement de problème.
name		chaîne	Nom de l'événement de problème.
severity		nombre	Gravité de l'événement de problème (0 - Non classifié, 1 - Information, 2 - Avertissement, 3 - Moyen, 4 - Haut, 5 - Désastre).
hosts		tableau	Liste des hôtes impliqués dans l'expression du déclencheur ; il doit y avoir au moins un élément dans le tableau.
-		objet	
	host	chaîne	Nom d'hôte.
	name	chaîne	Nom d'hôte visible.
groups		tableau	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.
-		chaîne	Nom du groupe d'hôtes.
tags		tableau	Liste des tags problématiques (peut être vide).
-		objet	
	tag	chaîne	Nom du tag.
	value	chaîne	Valeur du tag (peut être vide).

Les informations suivantes sont exportées pour un événement de récupération :

Champ	Type	Description
clock	nombre	Nombre de secondes depuis Epoch jusqu'au moment où le problème a été résolu (partie entière).
ns	nombre	Nombre de nanosecondes à ajouter à `clock` pour obtenir un temps de résolution précis du problème.
value	nombre	0 (toujours).
eventid	nombre	ID d'événement de récupération.
p_eventid	nombre	ID d'événement de problème.

Exemples

Problème :

```
{"clock":1519304285,"ns":123456789,"value":1,"name":"Either Zabbix agent is unreachable on Host B or polle
```

Récupération :

```
{"clock":1519304345,"ns":987654321,"value":0,"eventid":43,"p_eventid":42}
```

Problème (génération multiple d'événements 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
```

Récupération :

```
{"clock":1519304346,"ns":987654321,"value":0,"eventid":44,"p_eventid":43}
```

```
{"clock":1519304346,"ns":987654321,"value":0,"eventid":44,"p_eventid":42}
```

Valeurs d'élément

Les informations suivantes sont exportées pour une valeur d'élément collectée :

Champ	Type	Description
host	object	Nom d'hôte de l'élément hôte.
host	string	Nom d'hôte.
name	string	Nom d'hôte visible.
groups	array	Liste des groupes d'hôtes de l'élément hôte ; le tableau doit contenir au moins un élément.
-	string	Nom du groupe d'hôtes.
item_tags	array	Liste des tags d'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.
clock	number	Nombre de secondes écoulées depuis l'époque Unix jusqu'au moment où la valeur a été collectée (partie entière).
ns	number	Nombre de nanosecondes à ajouter à `clock` pour obtenir l'heure précise de collecte de la valeur.
timestamp (*Log* uniquement)	number	0 si non disponible.
source (*Log* uniquement)	string	Chaîne vide si non disponible.
severity (*Log* uniquement)	number	0 si non disponible.

Champ	Type	Description
eventid (*Log* uniquement)	number	0 si non disponible.
value	number (pour les éléments numériques) ou string (pour les éléments texte)	Valeur de l'élément collectée.
type	number	Type de valeur collectée : 0 - numérique à virgule flottante, 1 - caractère, 2 - journal, 3 - numérique non signé, 4 - texte, 5 - binaire, 6 - JSON

Exemples

Valeur numérique (non signée) :

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Valeur numérique (à virgule flottante) :

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Valeur de type 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":[{"
```

Tendances

Les informations suivantes sont exportées pour une valeur de tendance calculée :

Field	Type	Description
host	object	Nom d'hôte de l'hôte de l'élément.
host	string	Nom d'hôte.
name	string	Nom d'hôte visible.
groups	array	Liste des groupes d'hôtes de l'hôte de l'élément ; le tableau doit contenir au moins un élément.
-	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.
clock	number	Nombre de secondes depuis l'époque jusqu'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.
type	number	Type de valeur : 0 - numérique à virgule flottante, 3 - numérique non signé

Exemples

Valeur numérique (non signée) :

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Valeur numérique (à virgule flottante) :

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

4 Éléments

Veuillez utiliser la barre latérale pour accéder au contenu de cette section.

1 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 (elle ne peut pas être relue à partir de 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 toujours rester en RAM. Elle n'est jamais déplacée sur le disque.

Warning:
Certains de ces paramètres sont spécifiques à une 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 minimal 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 attribués instantanément aux processus qui demandent davantage de mémoire.

Ainsi, la mémoire inactive est à la fois utilisée et disponible. Pour cette raison, l'élément *vm.memory.size[used]* est conçu à des fins d'information uniquement, tandis que *vm.memory.size[available]* est conçu pour être utilisé dans les déclencheurs.

Voir également

1. Détails supplémentaires sur le calcul de la mémoire dans différents systèmes d'exploitation

2 Vérifications passives et actives de l'agent

Aperçu

Cette section fournit des détails sur les contrôles passifs et actifs effectués 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 sont 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 requête de données. Le serveur ou le proxy Zabbix 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 de démarrer d'autres vérifications. La résolution DNS est également asynchrone.

L'agent poller tentera de se connecter à toutes les adresses renvoyées par la recherche DNS. Cela garantit que si une adresse IP est inaccessible, le poller essaiera l'adresse disponible suivante, augmentant ainsi la probabilité d'une connexion réussie. Cette amélioration s'applique à la fois au serveur et au proxy Zabbix.

La concurrence maximale des vérifications asynchrones est de 1000 (définie par MaxConcurrentChecksPerPoller).

Le nombre d'agent pollers asynchrones est défini par le paramètre StartAgentPollers.

Requête du serveur

Pour la définition de l'en-tête et de la longueur des données, veuillez consulter les détails du protocole.

```
{
  "request": "passive checks",
  "data": [
    {
      "key": "agent.version",
      "timeout": 3
    }
  ]
}
```

Champ		Type	Obligatoire	Valeur
request		*string*	oui	`"passive checks"`
data		*array of object*	oui	Élément de vérification passive.
	key	*string*	oui	Clé d'élément avec macros développées.
	timeout	*number*	oui	Délai d'expiration de la communication.

Réponse de l'agent

```
{
  "version": "8.0.0",
  "variant": 2,
  "data": [
    {
      "value": "8.0.0"
    }
  ]
}
```

Champ	Type	Obligatoire	Valeur
version	*string*	oui	Le numéro de version de l'agent.
variant	*number*	oui	La variante de l'agent (*1* - agent Zabbix, *2* - agent Zabbix 2).
data	*array of object*	oui	Contient le résultat de la vérification.
value	*string*	non	La valeur de l'élément si la vérification a réussi.
error	*string*	non	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 un protocole en texte brut, un basculement vers l'ancien protocole a été mis en place.

Les contrôles passifs sont effectués à l'aide du protocole JSON (7.0 et versions ultérieures) après un redémarrage ou lorsqu'une configuration d'interface est modifiée. Si aucune réponse JSON valide n'est reçue (l'agent a envoyé « ZBX_NOTSUPPORTED »), Zabbix mettra l'interface en cache comme utilisant l'ancien protocole et **réessaiera** le contrôle en envoyant uniquement la clé de l'élément.

Notez que toutes les heures, le serveur/proxy Zabbix essaiera de 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 depuis le serveur/proxy une liste d'éléments et/ou de commandes distantes pour un traitement indépendant.

Les serveurs/proxys à partir desquels obtenir les vérifications actives sont listés 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 ce même fichier de configuration. Cependant, si l'actualisation des vérifications actives échoue, une nouvelle tentative est effectuée après 60 secondes codées 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), au cours 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(x) serveur(s). Si l'agent a reçu des commandes distantes à exécuter, le résultat de l'exécution sera également envoyé. Notez que l'exécution de commandes distantes 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 contrôles actifs, 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 contrôles actifs est utilisée pour obtenir les contrôles actifs à traiter par l'agent.
Cette requête est envoyée par l'agent au démarrage, puis selon les intervalles de 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"
}
```

Champ	Type	Obligatoire	Valeur
request	*string*	oui	`active checks`
host	*string*	oui	Nom d'hôte.
host_metadata	*string*	non	Le paramètre de configuration HostMetadata ou la valeur de la métrique HostMetadataItem.
interface	*string*	non	Le paramètre de configuration HostInterface ou la valeur de la métrique HostInterfaceItem.
ip	*string*	non	La première IP du paramètre de configuration ListenIP, s'il est défini.
port	*number*	non	La valeur du paramètre de configuration ListenPort, s'il est défini et différent du port d'écoute par défaut de l'agent.
version	*string*	oui	Le numéro de version de l'agent.
variant	*number*	oui	La variante de l'agent (*1* - Zabbix agent, *2* - Zabbix agent 2).
config_revision	*number*	non	Identifiant de configuration pour la synchronisation incrémentielle de la configuration.
session	*string*	non	Identifiant de session pour la synchronisation incrémentielle de la configuration.

Réponse du serveur

La réponse des contrôles actifs est renvoyée par le serveur à l'agent après traitement de la requête de contrôles actifs.

```
{
  "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
    }
  ]
}
```

Champ	Type	Obligatoire	Valeur
response	*string*	oui	`success` | `failed`
info	*string*	non	Informations sur l'erreur en cas d'échec.
data	*array of objects*	non	Éléments de contrôle actif. Omis si la configuration de l'hôte est inchangée.
key	*string*	non	Clé d'élément avec macros développées.
itemid	*number*	non	Identifiant de l'élément.
delay	*string*	non	Intervalle de mise à jour de l'élément. Les intervalles flexibles/de planification sont pris en charge à la fois par Zabbix agent et Zabbix agent 2 depuis Zabbix 7.0.
lastlogsize	*number*	non	lastlogsize de l'élément.
mtime	*number*	non	mtime de l'élément.
timeout	*string*	non	Délai d'expiration de l'élément.
refresh_unsupported	*number*	non	Intervalle de rafraîchissement des éléments non pris en charge.
regexp	*array of objects*	non	Expressions régulières globales.
name	*string*	non	Nom de l'expression régulière globale.
expression	*string*	non	Expression régulière globale.
expression_type	*number*	non	Type d'expression régulière globale.
exp_delimiter	*string*	non	Délimiteur de l'expression régulière globale.
case_sensitive	*number*	non	Paramètre de sensibilité à la casse de l'expression régulière globale.
commands	*array of objects*	non	Commandes distantes à exécuter. Incluses si l'exécution de commandes distantes a été déclenchée par une operation 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 toute commande distante incluse dans la réponse du serveur aux contrôles actifs.
command	*string*	non	Commande distante.
id	*number*	non	Identifiant de la commande distante.
wait	*number*	non	Mode d'exécution de la commande distante ("0" (nowait) pour les commandes provenant des operations d'action ; "1" (wait) pour les commandes provenant de l'exécution manuelle d'un script).
config_revision	*number*	non	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 contrôles
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 démarre 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'un contrôle actif. Cela est possible parce que n'importe qui peut se faire passer pour un agent actif et demander les données de configuration des éléments ; aucune authentification n'a lieu, sauf si vous utilisez les options de chiffrement.

Envoi des données collectées

Envoi par l'agent

La requête de données de l'agent contient les valeurs d'éléments collectées ainsi que 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
}
```

Champ	Type	Obligatoire	Valeur
request	*string*	oui	`agent data`
data	*array of objects*	oui	Valeurs des éléments.
id	*number*	oui	Identifiant de la valeur (compteur incrémental utilisé pour vérifier les valeurs dupliquées en cas de problèmes réseau).
itemid	*number*	oui	Identifiant de l'élément.
value	*string*	non	Valeur de l'élément.
lastlogsize	*number*	non	lastlogsize de l'élément.
mtime	*number*	non	mtime de l'élément.
state	*number*	non	État de l'élément.
source	*string*	non	Source du journal des événements de la valeur.
eventid	*number*	non	eventid du journal des événements de la valeur.
severity	*number*	non	Gravité du journal des événements de la valeur.
timestamp	*number*	non	Horodatage du journal des événements de la valeur.
clock	*number*	oui	Horodatage de la valeur (secondes depuis l'époque Unix).
ns	*number*	oui	Nanosecondes de l'horodatage de la valeur.

Champ	Type	Obligatoire	Valeur
commands	*array of objects*	non	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 anciens agents actifs ignoreront toute commande distante incluse dans la réponse du serveur de contrôles actifs.
id	*number*	non	Identifiant de la commande distante.
value	*string*	non	Résultat de l'exécution de la commande distante si l'exécution a réussi.
error	*string*	non	Message d'erreur d'exécution de la commande distante si l'exécution a échoué.
session	*string*	oui	Identifiant de session unique généré à chaque démarrage de l'agent.
host	*string*	oui	Nom d'hôte.
version	*string*	oui	Numéro de version de l'agent.
variant	*number*	oui	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 dupliquées qui peuvent être envoyées dans des environnements à connectivité dégradée.

Réponse du serveur

La réponse de données de l'agent est renvoyée par le serveur à l'agent après traitement de la requête de données de l'agent.

```
{
  "response": "success",
  "info": "processed: 2; failed: 0; total: 2; seconds spent: 0.003534"
}
```

Champ	Type	Obligatoire	Valeur
response	*string*	oui	`success` | `failed`
info	*string*	oui	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 caractères 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 Zabbix agent/ agent 2).

Il est utilisé pour surveiller la disponibilité des contrôles actifs.

```
{
  "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 d'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* - Zabbix agent, *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 contrôles actifs suivants) vers une autre instance de proxy ou de serveur.

```
{
  "response": "failed",
  "redirect": {
    "revision": 2,
    "address": "192.0.2.0:10055"
  }
}
```

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 prendra jusqu'à 16 Mo de données codées en XML Base64, mais une seule valeur décodée ne devrait pas dépasser 64 Ko, sinon elle sera tronquée à 64 Ko lors du décodage.

3 Niveau d'autorisation minimal 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 d'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

Clé d’élément	Groupe d’utilisateurs	
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’éléments spécifiques à Windows

Clé d’élément	Groupe d’utilisateurs	
	Recommandé	Minimum techniquement acceptable (la fonctionnalité est limitée)
eventlog	Event Log Readers	Guests
net.if.list	Guests	Guests
perf_counter	Performance Monitor Users	Performance Monitor Users
proc_info	Administrators	Guests
service.discovery	Guests	Guests
service.info	Guests	Guests
services	Guests	Guests
wmi.get	Administrators	Guests

Clé d'élément	Groupe d'utilisateurs	
vm.vmemory.size	Guests	Guests

4 Encodage des valeurs renvoyées

Le serveur Zabbix s'attend à ce que chaque valeur textuelle renvoyée soit encodée en UTF8. Cela concerne tous les types de contrôles : agent Zabbix, SSH, Telnet, etc.

Différents systèmes/appareils surveillés et différents contrôles 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 renvoyée n'est pas encodée en UTF8 et qu'elle 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 avec un encodage non UTF8, ce caractère et les suivants seront ignorés lorsque la base de données stockera cette valeur. Aucun message d'avertissement ne sera écrit dans *zabbix_server.log*.
Pertinent pour au moins MySQL version 5.1.61

PostgreSQL

Si une valeur contient un caractère non-ASCII avec un encodage non UTF8, une requête SQL échouée (PGRES_FATAL_ERROR:ERROR invalid byte sequence for encoding) et les données ne seront pas stockées. Un message d'avertissement approprié sera écrit dans *zabbix_server.log*.
Pertinent pour au moins la version 9.1.3 de PostgreSQL.

5 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 Go 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 un Solaris 32 bits en raison de son incompatibilité avec procfs et swapctl.

6 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]`.

7 Remarques sur le paramètre memtype dans les éléments proc.mem

Aperçu

Le paramètre **memtype** est pris en charge sur les plateformes Linux, AIX, FreeBSD et Solaris.

Trois valeurs communes de 'memtype' sont prises en charge sur toutes ces plates-formes : `pmem`, `rss` et `vsize`. De plus, les valeurs 'memtype' spécifiques à la plate-forme sont prises en charge sur certaines plates-formes.

AIX

Vous trouverez les valeurs prises en charge pour le paramètre 'memtype' sous AIX dans le tableau suivant.

Valeur prise en charge	Description	Source dans la structure procentry64	Essaye 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	"ps gvw" SIZE column
dsize	Taille des données	pi_dsize	
tsize	Taille du texte (code)	pi_tsize	"ps gvw" Colonne TSIZ
sdsize	Taille des données de la bibliothèque partagée	pi_sdsize	
drss	Taille de l'ensemble résident des données	pi_drss	
trss	Taille de l'ensemble résident du texte	pi_trss	

Remarques pour AIX :

1. Lors du choix des paramètres pour la clé d'élément proc.mem[] sur AIX, essayez de spécifier des critères de sélection de processus précis. Sinon, il existe un risque que des processus non souhaités soient 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 faire correspondre) entraîne une auto-comptabilisation de l'agent Zabbix, ce qui n'est probablement pas ce que vous souhaitez.

2. N'utilisez pas `ps -ef` pour parcourir les processus : il n'affiche que les processus non noyau. Utilisez `ps -Af` pour voir tous les processus qui seront vus par l'agent Zabbix.
3. Examinons l'exemple de `topasrec` pour voir comment l'agent Zabbix proc.mem[] sélectionne les processus.

```
$ 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[] comporte les arguments suivants :

`proc.mem[<name>,<user>,<mode>,<cmdline>,<memtype>]`

Le 1er critère est un nom de processus (argument <name>). Dans notre exemple, l'agent Zabbix le verra comme « topasrec ». Pour qu'il y ait correspondance, vous devez soit spécifier « topasrec », soit le laisser vide. Le 2e critère est un nom d'utilisateur (argument <user>). Pour qu'il y ait correspondance, vous devez soit spécifier « root », soit le laisser vide. Le 3e critère utilisé dans la sélection du 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 qu'il y ait correspondance, vous devez soit spécifier une expression régulière qui correspond à cette chaîne, soit le laisser vide.

Les arguments <mode> et <memtype> sont appliqués après l'utilisation des trois critères mentionnés ci-dessus.

FreeBSD

Vous trouverez les valeurs prises en charge pour le paramètre 'memtype' sous FreeBSD dans le tableau suivant.

Valeur prise en charge	Description	Source dans la structure kinfo_proc	Essaye d'être compatible avec
vsize	Taille de la mémoire virtuelle	kp_eproc.e_vm.vm_map_size ou ki_size	ps -o vsz
pmem	Pourcentage de 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

Vous trouverez les valeurs prises en charge pour le paramètre 'memtype' sous Linux dans le tableau suivant.

Valeur prise en charge	Description	Source dans le fichier /proc/<pid>/status
vsize [1]	Taille de la mémoire virtuelle	VmSize
pmem	Pourcentage de 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 la table des pages	VmPTE
taille	Taille du code de processus + données + segments de pile	VmExe + VmData + VmStk
stk	Taille du segment de pile	VmStk
swap	Taille de l'espace de swap 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 remarqué que l'autosurveillance de l'agent Zabbix actif vérifiant le processus avec `proc.mem[,,,,data]` affiche une valeur de 4 ko de plus de ce qui est indiqué par la ligne "VmData" dans le fichier de l'agent /proc/<pid>/status. Au moment de l'auto-mesure, le segment de données de l'agent augmente de 4 ko, puis revient à la taille précédente.

Solaris

Voir les valeurs prises en charge pour le paramètre 'memtype' sur Solaris dans le tableau.

Valeur supportée	Description	Source dans la structure psinfo	Essaie 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 Il peut être sous-estimé - voir la description rss dans "man ps".	pr_rssize	ps -o rss

Notes de bas de page

[1] Valeur par défaut.

8 Remarques 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.

9 Détails d'implémentation des contrôles net.tcp.service et net.udp.service

L'implémentation des contrôles net.tcp.service et net.udp.service est détaillée sur cette page pour les différents 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.

10 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 parent	ppid : PID parent	processes : nombre de processus
name : nom du processus	name : nom du processus	vsize : taille de la mémoire virtuelle

mode=process	mode=thread	mode=summary
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 — divisez par `getconf CLK_TCK` ou `sysconf(_SC_CLK_TCK)` pour obtenir des 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 — divisez par `getconf CLK_TCK` ou `sysconf(_SC_CLK_TCK)` pour obtenir des secondes)	pte : taille des entrées de table de pages
data : taille du segment de données	state : état du thread	size : taille du code du processus + segments de données + pile
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 — divisez par `getconf CLK_TCK` ou `sysconf(_SC_CLK_TCK)` pour obtenir des secondes)
lib : taille des bibliothèques partagées		cputime_system : temps CPU total passé en mode système (valeur exprimée en tops d’horloge — divisez par `getconf CLK_TCK` ou `sysconf(_SC_CLK_TCK)` pour obtenir des 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 de pages		page_faults : nombre de défauts de page
size : taille du code du processus + segments de données + pile		pss : mémoire de taille d’ensemble proportionnelle
stk : taille du segment de pile		
swap : taille de l’espace d’échange utilisé		
cputime_user : temps CPU total passé en mode utilisateur (valeur exprimée en tops d’horloge — divisez par `getconf CLK_TCK` ou `sysconf(_SC_CLK_TCK)` pour obtenir des secondes)		

mode=process	mode=thread	mode=summary
cputime_system : temps CPU total passé en mode système (valeur exprimée en tops d'horloge — divisez par `getconf CLK_TCK` ou `sysconf(_SC_CLK_TCK)` pour obtenir des 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 : mémoire de taille d'ensemble proportionnelle		

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
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

mode=process	mode=thread	mode=summary
cputime_system: temps CPU total passé en mode système (indiqué en secondes) 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		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 ppid: PID parent name: nom du processus user: utilisateur sous lequel le processus s'exécute sid: SID de l'utilisateur vmsize: taille de la mémoire virtuelle wkset: taille de l'ensemble de travail du processus cputime_user: temps CPU total passé en mode utilisateur (indiqué en millisecondes) cputime_system: temps CPU total passé en mode système (indiqué en millisecondes) threads: nombre de threads page_faults: nombre de défauts de page handles: nombre de handles io_read_b: octets d'E/S lus io_write_b: octets d'E/S écrits io_read_op: opérations de lecture d'E/S io_write_op: opérations d'écriture d'E/S	pid: PID ppid: PID parent name: nom du processus user: utilisateur sous lequel le processus s'exécute sid: SID de l'utilisateur tid: ID du thread	name: nom du processus processes: nombre de processus vmsize: taille de la mémoire virtuelle wkset: taille de l'ensemble de travail du processus cputime_user: temps CPU total passé en mode utilisateur (indiqué en millisecondes) cputime_system: temps CPU total passé en mode système (indiqué en millisecondes) threads: nombre de threads page_faults: nombre de défauts de page handles: nombre de handles io_read_b: octets d'E/S lus io_write_b: octets d'E/S écrits io_read_op: opérations de lecture d'E/S 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

mode=process	mode=thread	mode=summary
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		

11 Paramètres d'interface d'hôte injoignable/indisponible

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 hôte devient inaccessible.

Interface inaccessible

Une interface d'hôte est considérée comme inaccessible après un contrôle échoué (erreur réseau, délai d'attente) par les agents Zabbix, SNMP, IPMI ou JMX. Depuis Zabbix 6.2.0, les contrôles actifs de l'agent Zabbix affectent également la disponibilité de l'interface. Lorsque les contrôles actifs deviennent indisponibles, ils contribuent à l'état global de disponibilité de l'interface de l'agent.

À partir du moment où une interface devient inaccessible, **UnreachableDelay** définit à quelle fréquence 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 contrôles IPMI).
Par défaut, l'intervalle entre deux contrôles consécutifs d'accessibilité est de 15 secondes.

Attention:
Les contrôles effectués par des pollers asynchrones ne sont pas transférés aux pollers d'inaccessibilité.

Dans le journal du serveur Zabbix, l'inaccessibilité est indiquée par des messages tels que :

```
L'élément d'agent Zabbix "system.cpu.load[percpu,avg1]" sur l'hôte "New host" a échoué : première erreur ré
L'élément d'agent Zabbix "system.cpu.load[percpu,avg15]" sur l'hôte "New host" a échoué : autre erreur rése
```

Les messages du journal précisent l'élément exact qui a échoué ainsi que son type (agent Zabbix).

Note:
Le paramètre *Timeout* affecte également le délai avant qu'une interface soit revérifiée pendant son inaccessibilité. Si Timeout est défini à 20 secondes et UnreachableDelay à 30 secondes, le prochain contrôle 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 contrôles actifs 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.

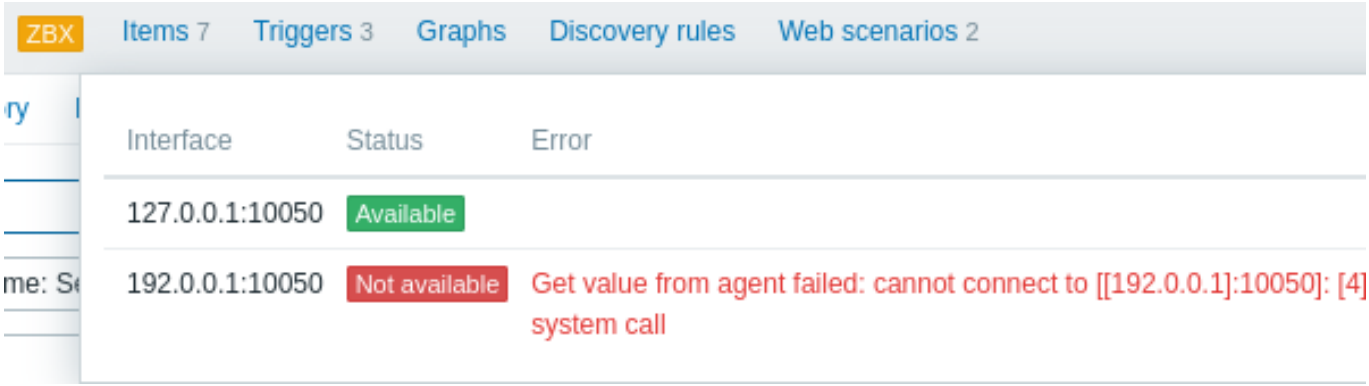
Interface indisponible

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
```

12 Surveillance à distance des statistiques de Zabbix

Aperçu

Il est possible de rendre quelques métriques internes du serveur et du proxy Zabbix accessibles à distance par une autre instance Zabbix ou un outil tiers. Cela peut être utile pour que les services de supports ou de fournisseurs puissent surveiller les serveurs/proxy Zabbix à distance ou, dans les organisations où Zabbix n'est pas le principal outil de surveillance, pour que les métriques internes de Zabbix puissent être surveillées par un système tiers dans une configuration de surveillance parapluie.

Les statistiques internes de Zabbix sont exposées à un ensemble configurable d'adresses listé dans le nouveau paramètre serveur/proxy 'StatsAllowedIP'. Les demandes seront acceptées uniquement à partir de ces adresses.

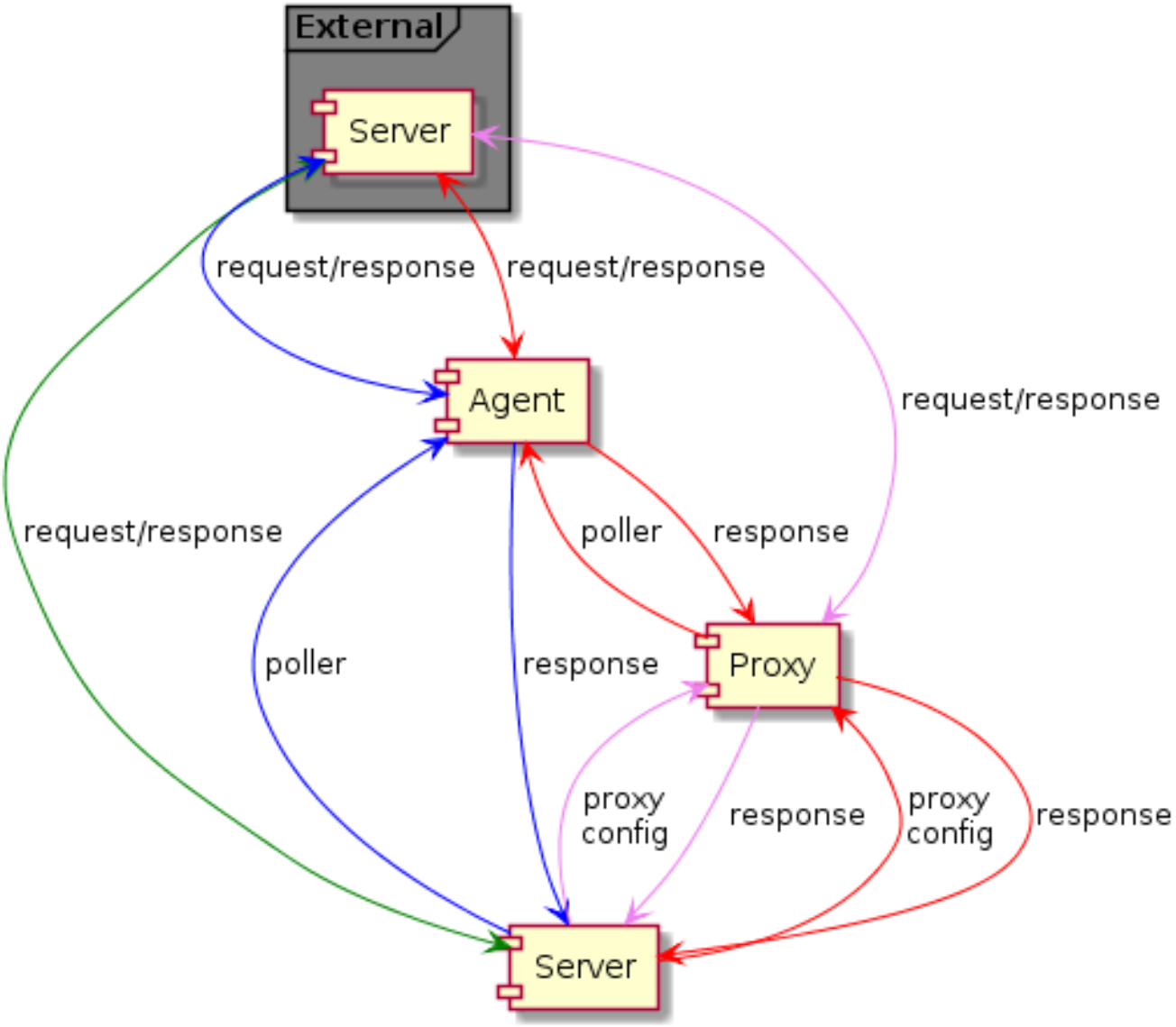
Eléments

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 la file d'attente interne

Il existe également deux autres éléments permettant spécifiquement d'interroger à distance les statistiques de la file d'attente interne sur une autre instance Zabbix :

- Elément interne `zabbix[stats,<ip>,<port>,queue,<from>,<to>]` - pour les requêtes de file d'attente internes directes vers le serveur/proxy Zabbix distant
- Elément agent `zabbix.stats[<ip>,<port>,queue,<from>,<to>]` - pour les requêtes de file d'attente interne basées sur l'agent vers le serveur/proxy Zabbix distant

Voir également : Eléments internes, éléments agent Zabbix

13 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. Installez le KDC et les utilitaires client :

```
sudo apt update
sudo apt install krb5-kdc krb5-admin-server krb5-user
```

Pendant l'installation des paquets, répondez 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. Associez un nom d'hôte convivial (facultatif, pour les tests locaux).

Modifiez /etc/hosts et ajoutez une entrée pour votre DC et votre serveur web si vous n'avez pas de DNS :

```
sudo vi /etc/hosts
```

Exemple de ligne que vous pouvez ajouter :

```
192.168.1.100  dc01.example.com dc01
```

3. Configurez 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. Des incohérences ici provoquent des erreurs `Server not found in Kerberos database`.

4. Initialisez la base de données Kerberos (une seule fois, sur l'hôte KDC). Définissez un mot de passe maître sécurisé lorsque vous y êtes invité :

```
sudo krb5_newrealm
```

5. Créez le principal `HTTP/host.fqdn@REALM` en utilisant le nom d'hôte exact que les clients utiliseront ; privilégiez les minuscules (par exemple `HTTP/web.example.com@EXAMPLE.COM`). Une différence de casse/nom provoque `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
```

Déplacez le keytab vers l'hôte web (ou conservez-le localement s'il s'agit de la même machine) et définissez des permissions utilisables par Apache :

```
chown www-data:www-data /etc/apache2/http.keytab
chmod 600 /etc/apache2/http.keytab
#### verify
sudo -u www-data -k /etc/apache2/http.keytab
```

6. Installez et activez le module Apache GSSAPI :

```
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'`, supprimez la directive non prise en charge ou mettez à niveau le module.

7. Configurez un VirtualHost (ajustez `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>
```

Redémarrez Apache :

```
sudo systemctl restart apache2
```

8. Activez/démarrez les services KDC et vérifiez les ports d'écoute (hôte KDC) :

```
sudo systemctl enable --now krb5-kdc krb5-admin-server
ss -tnlp | grep :80    # or: sudo netstat -tnlp | grep :80
```

9. Obtenez un TGT pour les tests (exécutez la commande en tant qu'utilisateur qui utilisera le ticket).

Vous devez voir krbtgt/EXAMPLE.COM@EXAMPLE.COM dans la liste des tickets. Exécutez kinit avec le même utilisateur du système d'exploitation qui a besoin du ticket (par exemple zabbix pour les contrôles web ou www-data/Apache pour les tests SSO interactifs dans le navigateur). Les tickets émis pour 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. Testez l'échange SPNEGO avec curl (depuis un client disposant d'un TGT valide). Un 200 OK (ou une redirection vers l'application) indique que SPNEGO a réussi :

```
curl -v --negotiate -u : http://dc01.example.com/
```

11. Facultativement, si l'interface Zabbix doit accepter les connexions authentifiées par HTTP, activez l'authentification HTTP dans le front-end Zabbix (ui/conf/zabbix.conf.php) :

```
$ALLOW_HTTP_AUTH = true;
```

Dans l'interface web, allez dans *Users* > *Authentication* puis ouvrez l'onglet *HTTP settings*. Cochez la case *Enable HTTP authentication* et cliquez sur *Ok* dans la fenêtre contextuelle. Sélectionnez "HTTP login form" dans la liste déroulante *Default login form*. Déterminez si *Case-sensitive login* correspond à la politique de votre annuaire. Cliquez sur le bouton *Update* pour terminer.

12. Configuration du navigateur (Firefox est utilisé comme exemple) : définissez network.negotiate-auth.trusted-uris sur le ou les hôtes effectuant Negotiate (dc01.example.com) afin que le navigateur envoie automatiquement les jetons Kerberos.

Dans about:config :

```
network.negotiate-auth.trusted-uris = dc01.example.com
```

Désormais, en visitant http://dc01.example.com, vous devriez être connecté directement à Zabbix sans le formulaire.

13. Gardez les clés/tickets à jour. La durée de vie par défaut d'un ticket Kerberos est d'environ 10 heures. Ajoutez un cron/minuteur systemd pour éviter les expirations :

```
####for the web service
kinit -kt /etc/apache2/http.keytab HTTP/dc01.example.com@EXAMPLE.COM
####for the monitoring user
kinit -kt /var/lib/zabbix/kerb.keytab kerb-admin@EXAMPLE.COM
```

14. Vérifications de maintenance :

- klist -k /etc/apache2/http.keytab — vérifiez que le principal de service est présent dans le keytab.
- sudo tail -f /var/log/apache2/error.log — surveillez les erreurs GSSAPI (gss_acquire_cred[_from]() failed to get server creds signifie keytab/permissions incorrects ou principal manquant).
- curl --negotiate renvoyant 401/403 signifie souvent un principal incorrect, l'absence de ticket, une non-correspondance de l'en-tête d'hôte ou un problème de permissions du système de fichiers ; vérifiez les journaux et les mappages de domaine dans /etc/krb5.conf.

Remarques sur la sécurité et les permissions de 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 d'utilisateur zabbix, ou 0440 et root:www-data pour un keytab Apache.

Évitez de stocker des mots de passe en clair à longue durée de vie 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 de nouveau le propriétaire et les permissions après l'opération — le rejet des informations d'identification par un serveur web est souvent dû à un problème de permissions de fichier.

14 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

15 Création de noms de compteurs de performance personnalisés pour VMware

Aperçu

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*

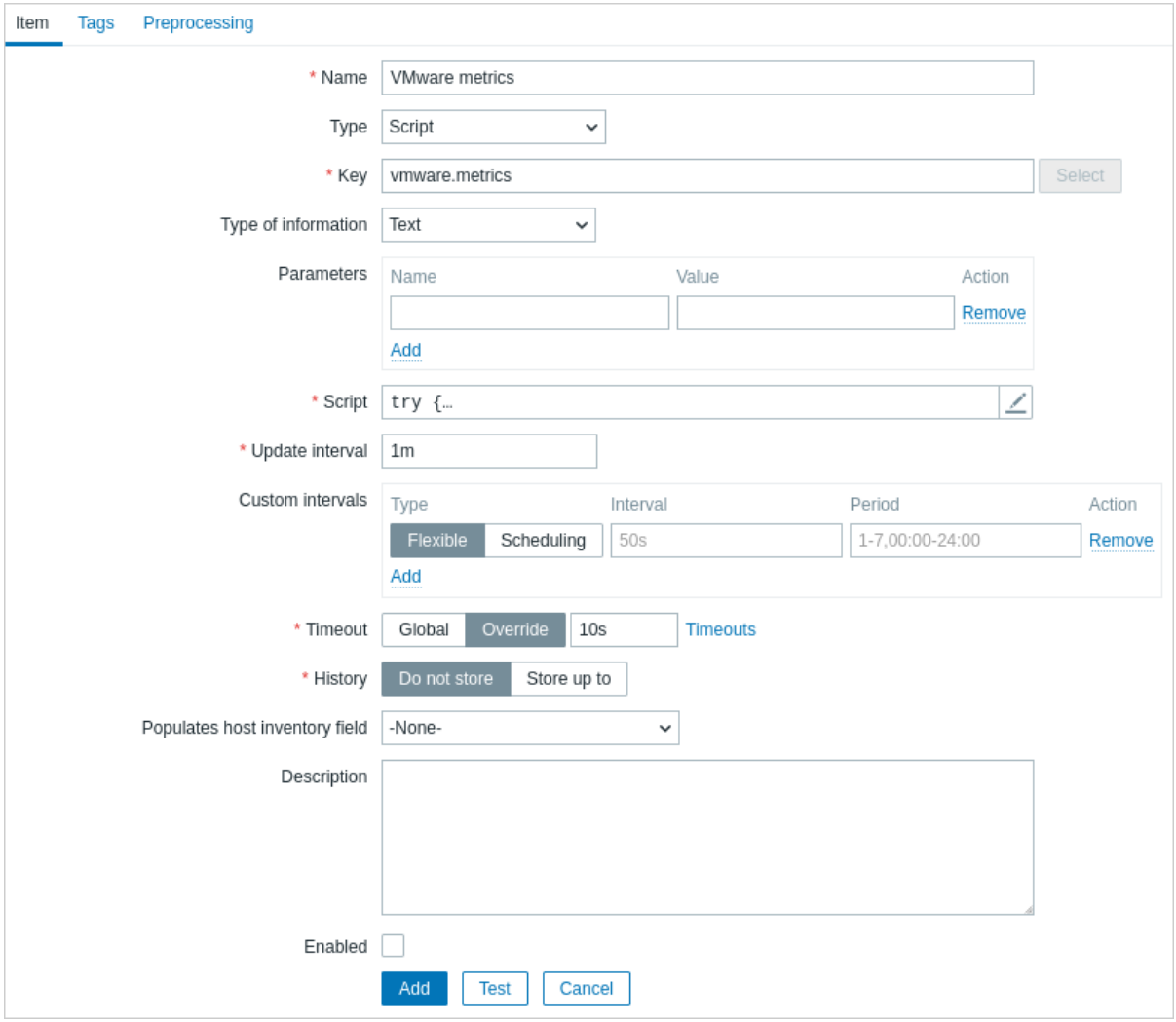
Ainsi, l'exemple ci-dessus donnerait le chemin de compteur suivant : `cpu/usagemhz[average]`

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 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 :



- *Nom* : VMware metrics
- *Type* : Script
- *Clé* : vmware.metrics
- *Type d'information* : Texte
- *Script* : copiez et collez le script fourni ci-dessous
- *Timeout* : 10
- *Historique* : Ne pas stocker
- *Activé* : non coché

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, appuyez sur le bouton *Test*, puis sur *Get value*.



Copiez le XML reçu dans un formateur XML quelconque et recherchez 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 :

champ	xPath	valeur
groupe	//groupInfo[../key=6]/key	cpu
compteur	//nameInfo[../key=6]/key	usagemhz
rollup	//rollupType[../key=6]	average

Le chemin du compteur de performance obtenu dans ce cas est : `cpu/usagemhz[average]`

16 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 clés 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": ""
        }
    }
]
```

17 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 les enregistrements DNS en fonction des 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[192.0.2.0,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[192.0.2.0,example.com]` peut renvoyer une erreur :


```
{
    "zbx_error_code": -1,
    "zbx_error_msg": "Communication error: read udp 192.0.2.0:12345->192.0.2.0:53: i/o timeout"
}
```


Les types de codes d'erreur suivants sont possibles :

Scénario	"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 lors de l'analyse du JSON	-2	"Received unexpected response"

Avec des paramètres supplémentaires, l'élément `net.dns.get[192.0.2.0,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": "192.0.2.0"
            },
            "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

18 Remarques 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
```

19 Prise en charge des grandes valeurs JSON

Pour garantir un traitement fiable des grandes valeurs avec le type de données JSON (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 Mo sur certains systèmes) :

```
[mysqld]
max_allowed_packet=128M

[mysqldump]
max_allowed_packet=1G
```

2. Ajustez d'autres variables système du serveur MySQL/MariaDB et variables système InnoDB selon les besoins. 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 d'expiration de communication entre le serveur/proxy Zabbix et l'agent (par exemple, les délais d'expiration des éléments ou la valeur du paramètre `Timeout` dans les fichiers de configuration du serveur/proxy et de l'agent Zabbix). Sinon, les messages échangés entre eux risquent de ne pas être transmis complètement, et vous pouvez voir des erreurs telles que *"message length does not match expected length"*. Examinez également les autres délais d'expiration selon les besoins, tels que le délai d'expiration du script ou le délai d'expiration du test d'élément.

5 Fonctions prises en charge

Voir Expression de déclencheur.

6 Macros

Il est possible d'utiliser les macros prises en charge et les macros utilisateur prises en charge par emplacement.

1 Macros prises en charge par l'emplacement

Vue d'ensemble

Le tableau des macros prises en charge par emplacement contient une référence de toutes les macros (intégrées, d'expression et définissables par l'utilisateur) prises en charge dans 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 tous les détails des macros.

La liste des 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, raccourcir ou extraire des sous-chaînes spécifiques), vous pouvez utiliser les fonctions de macro.

Macros prises en charge par emplacement

Emplacement	Macros prises en charge	Commentaires
Action		
Condition de période, Durée par défaut de l'étape d'opération, Durée de l'étape	Macros définies par l'utilisateur (globales uniquement)	Une seule macro doit remplir tout le champ ; plusieurs macros/un mélange avec du texte ne sont pas pris en charge
Notifications et commandes		

Emplacement	Macros prises en charge	Commentaires
Basées sur un déclencheur	Macros d'action Macros de date/heure {ESC.HISTORY} Macros d'événement, macros d'événement 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 d'hôte cible (commandes uniquement) Macros d'inventaire d'hôte Macros d'élément (sauf {ITEM.STATE}, {ITEM.STATE.ERROR}) Macros de 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 définies par l'utilisateur	Voir aussi : Utilisation des macros dans les messages
Mise à jour de problème	Macros d'action Macros de date/heure {ESC.HISTORY} Macros d'événement, macros d'événement cause/symptôme, macros d'événement de rétablissement Macros de mise à jour d'événement : toutes (sauf {EVENT.UPDATE.NSEVERITY}, {EVENT.UPDATE.SEVERITY}) Macro d'expression : {?EXPRESSION} Macros de fonction Macros d'hôte : {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros d'hôte cible (commandes uniquement) Macros d'inventaire d'hôte Macros d'élément (sauf {ITEM.STATE}, {ITEM.STATE.ERROR}) Macros de 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 définies par l'utilisateur	
Rétablissement de problème	Macros d'événement de rétablissement {FUNCTION.RECOVERY.VALUE} {TRIGGER.EXPRESSION.RECOVERY.EXPLAIN}	

Emplacement	Macros prises en charge	Commentaires
Découverte	Macros d'action Macros de date/heure Macros de découverte Macros d'événement : {EVENT.AGE}, {EVENT.DATE}, {EVENT.ID}, {EVENT.OBJECT}, {EVENT.SOURCE}, {EVENT.TIME}, {EVENT.TIMESTAMP} Macros d'hôte cible (commandes uniquement) Macros de proxy	
Auto-enregistrement	Macros d'action Macros de 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 d'hôte cible (commandes uniquement) Macros de proxy	
Basées sur un service	Macros d'action Macros de date/heure {ESC.HISTORY} Macros d'événement : toutes sauf {EVENT.ACK.STATUS}, {EVENT.OPDATA} Macros de service Macros définies par l'utilisateur	
Rétablissement de service	Macros d'événement (sauf {EVENT.ACK.STATUS}, {EVENT.OPDATA}), macros d'événement de rétablissement	
Mise à jour de service	Macros d'action Macros de date/heure {ESC.HISTORY} Macros d'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 de service Macros définies par l'utilisateur	
Notifications internes	Macros d'action Macros de date/heure {ESC.HISTORY} Macros d'événement (sauf {EVENT.NSEVERITY}, {EVENT.SEVERITY}), macros d'événement de rétablissement Macros d'hôte : {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros d'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 de proxy	
Basées sur un élément	Macros d'élément : {ITEM.STATE}, {ITEM.STATE.ERROR}	
Basées sur une règle LLD	Macros de notification de découverte de bas niveau	

Emplacement	Macros prises en charge	Commentaires
Basées 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 définies par l'utilisateur	
Paramètres de script d'alerte	Macros d'alerte Macros définies 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 définies par l'utilisateur (globales uniquement)	
Champ Timeout	Macros définies par l'utilisateur (globales uniquement)	Une seule macro doit remplir tout le champ ; plusieurs macros/un mélange avec du texte ne sont pas pris en charge.
Nom du graphique	Macro d'expression : {?EXPRESSION}	Seule une fonction **avg**, **last**, **max** ou **min** unique avec des secondes comme paramètre (les suffixes de temps peuvent être utilisés) est prise en charge dans cette macro à cet emplacement. La macro {HOST.HOST<1-9>} peut être utilisée comme hôte dans la macro (voir aussi les 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 définies par l'utilisateur [1]	[1] Pour l'adresse IP, une seule macro doit remplir tout le champ ; plusieurs macros/un mélange avec du texte ne sont pas pris en charge

Emplacement	Macros prises en charge	Commentaires
Port de l'interface	Macros définies par l'utilisateur	Une seule macro doit remplir tout le champ ; plusieurs macros/un mélange avec du texte ne sont pas pris en charge
Communauté SNMP *Nom de contexte SNMPv3, Nom de sécurité, Phrase secrète d'authentification, Phrase secrète de confidentialité* *Nom d'utilisateur, mot de passe IPMI*	Macros définies par l'utilisateur	
Élément, prototype d'élément, règle LLD		
Nom	Macros définies par l'utilisateur	Non pris en charge dans les noms de règles LLD.
Description	Macros définies par l'utilisateur	
Paramètres de clé	Macros d'hôte : {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros définies 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 Zabbix agent, SNMP, JMX ou IPMI de l'hôte dans cet ordre de priorité, ou vers « UNKNOWN » si l'hôte n'a aucune interface.
Champs Intervalle de mise à jour, Intervalles personnalisés, Timeout (disponible pour les types d'élément pris en charge), Conserver jusqu'à (pour l'historique/les tendances)	Macros définies par l'utilisateur	Une seule macro doit remplir tout le champ ; plusieurs macros/un mélange avec du texte ne sont pas pris en charge.

Emplacement	Macros prises en charge	Commentaires
Champs Supprimer les ressources perdues, Désactiver les ressources perdues	Macros définies par l'utilisateur	Ces champs sont pris en charge uniquement pour les règles LLD. Une seule macro doit remplir tout le champ ; plusieurs macros/un mélange avec du texte ne sont pas pris en charge.
Paramètres d'étape de prétraitement (y compris les scripts personnalisés), *paramètres personnalisés de gestion des erreurs* (*Définir la valeur sur* et champs *Définir l'erreur sur*)	Macros définies par l'utilisateur	
Expressions régulières de filtre	Macros d'hôte : {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros définies par l'utilisateur	Ces champs sont pris en charge uniquement pour les règles LLD.
Expressions régulières de filtre de surcharge	Macros définies par l'utilisateur	Ces champs sont pris en charge uniquement pour les règles LLD.
Opérations de surcharge : *Intervalle de mise à jour*, *Période de conservation de l'historique*, *Période de conservation des tendances* (pour les prototypes d'élément)	Macros définies par l'utilisateur	Ces champs sont pris en charge uniquement pour les règles LLD. Une seule macro doit remplir tout le champ ; plusieurs macros/un mélange avec du texte ne sont pas pris en charge.
Types d'élément		
Browser item *noms et valeurs des paramètres*	Macros d'hôte : {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros d'inventaire d'hôte Macros d'élément : {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} Macros définies par l'utilisateur	
Script item *noms et valeurs des paramètres*		
Formule d'élément **calculé/agrégé** (*constantes d'expression*, *paramètres de fonction*, *paramètres de clé d'élément*, (élément agrégé uniquement) *conditions de filtre* (nom du groupe d'hôtes et nom du tag))	Macros définies par l'utilisateur macros	

Emplacement	Macros prises en charge	Commentaires
Élément **Database monitor** *champs Nom d'utilisateur, Mot de passe, Requête SQL*	Macros définies par l'utilisateur	
Élément **HTTP agent** *champs URL, Requête, Corps de 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 définies par l'utilisateur	Les URL contenant une macro secrète ne fonctionneront pas, car la macro qu'elles contiennent sera résolue en « ****** »
Élément **HTTP agent** *champ Proxy HTTP*	Macros d'élément : {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} Macros définies par l'utilisateur	
Élément **HTTP agent** *champs Codes d'état requis, Nom d'utilisateur/mot de passe d'authentification HTTP, Mot de passe de clé SSL*	Macros définies par l'utilisateur	
Élément **HTTP agent** *champ Hôtes autorisés*	Macros d'hôte : {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros définies par l'utilisateur	
Élément **JMX agent** *champ Endpoint*	Macros d'hôte : {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.PORT} Macros définies par l'utilisateur	
Élément **Trapper** *champ Hôtes autorisés*		
Élément **SNMP agent** *champ OID SNMP*	Macros définies par l'utilisateur	
Élément **SSH and Telnet** *script*	Macros d'hôte : {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros définies par l'utilisateur	
Élément **SSH agent** *champs Nom d'utilisateur, Fichier de clé publique, Fichier de clé privée, Mot de passe*	Macros définies par l'utilisateur	
Élément **Telnet agent** *champs Nom d'utilisateur, Mot de passe*	Macros définies 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 d'inventaire d'hôte Macros de carte {TRIGGER.ID}	

Emplacement	Macros prises en charge	Commentaires
Libellé d'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 d'inventaire d'hôte Macros de 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** unique avec des secondes comme paramètre (les suffixes de temps peuvent être utilisés) est prise en charge dans cette macro à 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 de lien	Macro d'expression : {?EXPRESSION} [1]	
Libellé de forme	Macro d'expression : {?EXPRESSION} [1]	
Champ texte dans les formes de carte	{MAP.NAME}	
Découverte réseau		
Champ Intervalle de mise à jour	Macros définies par l'utilisateur (globales uniquement)	Une seule macro doit remplir tout le champ ; plusieurs macros/un mélange avec du texte ne sont pas pris en charge.
Champs Communauté SNMP, OID SNMP Champs Nom de contexte SNMPv3, Nom de sécurité, Phrase secrète d'authentification, Phrase secrète de confidentialité	Macros définies par l'utilisateur (globales uniquement)	
Proxy		
Adresse pour les agents actifs > *Port* (si le proxy appartient à un groupe) Pour le mode proxy passif : champs *Adresse* et *Port* de l'interface Champs *Timeouts for item types* en mode surcharge	Macros définies par l'utilisateur (globales uniquement)	Une seule macro doit remplir tout le champ ; plusieurs macros/un mélange avec du texte ne sont pas pris en charge.
Groupe de proxys		

Emplacement	Macros prises en charge	Commentaires
Champs Période de basculement, Nombre minimal de proxys	Macros définies par l'utilisateur (globales uniquement)	Une seule macro doit remplir tout le champ ; plusieurs macros/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 d'inventaire d'hôte [2] {MANUALINPUT} Macros de nom d'utilisateur Macros définies 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 de date/heure Macros d'événement, macros d'événement cause/symptôme, macros d'événement de rétablissement (si le rétablissement 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 d'inventaire d'hôte Macros d'élément (sauf {ITEM.STATE}, {ITEM.STATE.ERROR}) {MANUALINPUT} Macros de 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 de nom d'utilisateur Macros définies par l'utilisateur	
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 d'inventaire d'hôte {MANUALINPUT} Macros de nom d'utilisateur Macros définies par l'utilisateur	
URL, texte de confirmation (Type : URL, **action d'hôte/d'événement** manuelle)		
Nom et valeur du tag	Macros d'hôte : {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros d'inventaire d'hôte Macros d'élément : {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Macros de déclencheur : {TRIGGER.ID} (valeur du tag de déclencheur uniquement) Macros définies par l'utilisateur	Les macros de nom et de valeur du tag sont résolues uniquement pendant le processus de génération d'événement.

Emplacement	Macros prises en charge	Commentaires
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 définies par l'utilisateur	
Nom d'événement	Macros de date/heure : {TIME}, {TIMESTAMP} Macro d'expression : {?EXPRESSION} Macros de 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 définies par l'utilisateur {TRIGGER.VALUE}	
Expression	Macros définies par l'utilisateur [1]	[1] Uniquement dans les constantes et les paramètres de fonction ; les macros secrètes ne sont pas prises en charge
Champs Tag pour la correspondance, Nom d'entrée de menu, URL d'entrée de menu	Macros définies 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 définies par l'utilisateur {EVENT.ID}	
URL	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 définies par l'utilisateur	
Champ Intervalle de mise à jour	Macros définies par l'utilisateur	Une seule macro doit remplir tout le champ ; plusieurs macros/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 définies par l'utilisateur	

Emplacement	Macros prises en charge	Commentaires
Champs Utilisateur, Mot de passe, Mot de passe de clé SSL	Macros définies par l'utilisateur	
Étape de scénario web *champs Nom, Variables (valeurs uniquement)*	Macros définies par l'utilisateur	
Étape de scénario web *champs Timeout, Codes d'état requis*	Macros définies par l'utilisateur	Une seule macro doit remplir tout le champ ; plusieurs macros/un mélange avec du texte ne sont pas pris en charge.
Étape de 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 définies par l'utilisateur	
Étape de 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 d'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/un mélange avec du texte ne sont pas pris en charge.
Script JavaScript, nom de paramètre du *script JavaScript*	Macros définies par l'utilisateur (globales uniquement)	
Nom d'entrée de menu, URL d'entrée de menu	{EVENT.TAGS.<tag name>}	
Widgets		
Widget **Gauge** *champ Description*	Macros d'hôte : {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros d'inventaire d'hôte Macros d'élément (sauf {ITEM.STATE.ERROR}) Macros définies par l'utilisateur (globales uniquement)	
Widget **Honeycomb** *champ Libellé principal/secondaire*		
Widget **Item value** *champ Description*		
Widget **Top hosts** *colonne Données texte*	Macros d'hôte : {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros d'inventaire d'hôte Macros définies par l'utilisateur (globales uniquement)	
Widget **URL** (dynamique) *champ URL*	Macros d'hôte : {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macros définies par l'utilisateur macros	
Autres emplacements		

Emplacement	Macros prises en charge	Commentaires
Champ Temps de travail	Macros définies par l'utilisateur (globales uniquement)	Une seule macro doit remplir tout le champ ; plusieurs macros/un mélange avec du texte ne sont pas pris en charge.
Champ Timeouts for item types (paramètre global)		
Média utilisateur *champ Quand actif*	Macros définies par l'utilisateur (globales uniquement)	
Type de média e-mail *champs Nom d'utilisateur, Mot de passe*	Macros définies par l'utilisateur (globales uniquement)	Les macros secrètes sont recommandées.
Rapports planifiés *champs Sujet, Message*	Macros de 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.
 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 balises d’événement.
Développé en tableau vide si aucune balise n’existe.

{EVENT.TAGS.<tag name>}

La valeur du tag d’événement référencée par le nom du tag.
 Un nom de tag 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 tag entre guillemets doivent être échappés avec une barre oblique inverse.

{EVENT.TIME}

L’heure de l’événement qui a déclenché une action.
 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.
 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 un tableau vide si aucune balise n'existe.
 Utilisé dans le contexte d'un événement symptôme.

{EVENT.CAUSE.TAGS.<tag name>}

La valeur du tag de l'événement cause référencée par le nom du tag.
 Un nom de tag 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 tag 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.
 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 symptômes. Inclut les détails suivants : nom de l'hôte, nom de l'événement, gravité, ancienneté, tags de service et valeurs.
 Cette macro est utilisée dans le contexte de l'événement cause et renvoie des informations sur les événements symptômes.

Macros d'événement de récupération

{EVENT.RECOVERY.DATE}

La date de l'événement de rétablissement.

{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. Développé 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 tableau vide si aucune balise n'existe.

{EVENT.RECOVERY.TIME}

L'heure de l'événement de récupération.

{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 par l'humain de la ou des actions effectuées lors de la mise à jour du problème.
 Se résout aux valeurs suivantes : *acknowledged*, *unacknowledged*, *commented*, *changed severity from (original severity) to (updated severity)* et *closed* (selon le nombre d'actions effectuées dans une même mise à jour).

{EVENT.UPDATE.ACTIONJSON}

Un tableau JSON contenant des détails sur la ou les action(s) effectuée(s) lors de la mise à jour du 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.).

{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 (accusé de réception, etc.).

{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 Nième fonction basée sur un élément dans l'expression du déclencheur au moment de l'événement.
 Seules les fonctions ayant `/host/key` comme premier paramètre 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 Nième fonction basée sur un élément dans l'expression de récupération au moment de l'événement.*
 Seules les fonctions ayant `/host/key` comme premier paramètre sont comptabilisées.

 Cette macro peut être utilisée avec un index numérique, par exemple {FUNCTION.RECOVERY.VALUE<1-9>}, pour désigner la première, la deuxième, la troisième, etc. fonction dans une expression de déclencheur. Voir macros indexées.

Macros d'hôte

{HOST.CONN}

L'adresse IP ou le nom DNS de l'hôte, selon les paramètres de l'hôte.
 Dans les scripts globaux, les champs IP/DNS d'interface et les scénarios web, la macro sera résolue vers l'interface agent principale. Si une interface agent n'est pas définie, l'interface SNMP principale sera utilisée. Si une interface SNMP n'est pas non plus 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.CONN<1-9>} pour 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 faire référence au premier, deuxième, troisième, etc. hôte dans une expression de déclencheur. Voir indexed macros.

{HOST.DNS}

Le nom DNS de l'hôte.
 Dans les scripts globaux, les champs IP/DNS d'interface et les scénarios web, la macro sera résolue vers l'interface agent principale. Si une interface agent n'est pas définie, l'interface SNMP principale sera utilisée. Si une interface SNMP n'est pas non plus 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.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 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 {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 faire référence au premier, deuxième, 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 d'interface et les scénarios web, la macro sera résolue vers l'interface agent principale. Si une interface agent n'est pas définie, l'interface SNMP principale sera utilisée. Si une interface SNMP n'est pas non plus définie, l'interface JMX principale sera utilisée. Si une interface JMX n'est pas définie non plus, l'interface IPMI principale sera utilisée. Si l'hôte n'a aucune interface, la macro est résolue en « UNKNOWN ».

 Cette macro peut être utilisée avec un index numérique comme {HOST.IP<1-9>} pour faire référence au premier, deuxième, 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 d'interface et les scénarios web, la macro se résoudra vers l'interface agent principale. Si aucune interface agent n'est définie, l'interface SNMP principale sera utilisée. Si aucune interface SNMP n'est non plus définie, l'interface JMX principale sera utilisée. Si aucune interface JMX n'est non plus définie, l'interface IPMI principale sera utilisée. Si l'hôte ne possède aucune interface, la macro se résout 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 macros indexées.

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 indexed macros.

{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 hôte, etc., dans une expression de déclencheur. Voir indexed macros.

{INVENTORY.CHASSIS}

Le champ *Châssis* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.CHASSIS<1-9>}, pour pointer vers 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 faire référence au premier, deuxième, troisième, etc. hôte dans une expression de déclencheur. Voir les 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 pointer vers le premier, le deuxième, le troisième hôte, etc., dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.DEPLOYMENT.STATUS}

Le champ *Statut de déploiement* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.DEPLOYMENT.STATUS<1-9>}, pour faire référence au premier, deuxième, troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.HARDWARE}

Le champ *Matériel* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.HARDWARE<1-9>}, pour désigner le premier, le deuxième, le troisième hôte, etc., 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 hôte, etc., 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 pointer vers le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.HOST.NETWORKS}

Le champ *Réseaux de l'hôte* 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 hôte, etc., dans une expression de déclencheur. Voir les macros indexées.

{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 pointer vers le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.HW.ARCH}

Le champ *Architecture matérielle* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.HW.ARCH<1-9>}, pour faire référence au premier, au deuxième, au troisième hôte, etc., 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 les 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 faire référence au premier, deuxième, 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 pointer vers 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 hôte, etc., dans une expression de déclencheur. Voir indexed macros.

{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 ex. {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 pointer vers le premier, le deuxième, le troisième hôte, etc., 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 pointer vers 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 hôte, etc., 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 les 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 hôte, etc., dans une expression de déclencheur. Voir indexed macros.

{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 faire référence au premier, deuxième, 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 hôte, etc., 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 les 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 pointer vers le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.OS.SHORT}

Le champ *OS (Short)* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.OS.SHORT<1-9>}, pour faire référence au premier, deuxième, troisième, etc. hôte dans une expression de déclencheur. Voir indexed macros.

{INVENTORY.POC.PRIMARY.CELL}

Le champ *Cellulaire principal du point de contact (POC)* 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 hôte, etc., dans une expression de déclencheur. Voir indexed macros.

{INVENTORY.POC.PRIMARY.EMAIL}

Le champ *E-mail principal du point de contact (POC)* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.POC.PRIMARY.EMAIL<1-9>}, pour faire référence au premier, deuxième, troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{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 les macros indexées.

{INVENTORY.POC.PRIMARY.NOTES}

Le champ *Notes du contact 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 hôte, etc., 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 faire référence au premier, au deuxième, au troisième hôte, etc., dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.POC.PRIMARY.PHONE.B}

Le champ *Téléphone B du point de 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 faire référence au premier, deuxième, troisième, etc. hôte dans une expression de déclencheur. Voir les macros indexées.

{INVENTORY.POC.PRIMARY.SCREEN}

Le champ *Nom de l'écran principal du POC* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.POC.PRIMARY.SCREEN<1-9>}, pour faire référence au premier, deuxième, troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.POC.SECONDARY.CELL}

Le champ *Cellulaire du 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 secondaire du point de contact* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.POC.SECONDARY.EMAIL<1-9>}, pour faire référence au premier, au deuxième, au troisième hôte, etc., dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.POC.SECONDARY.NAME}

Le champ *Nom du point de 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 hôte, etc., dans une expression de déclencheur. Voir les macros indexées.

{INVENTORY.POC.SECONDARY.NOTES}

Le champ *Notes du point de 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 faire référence au premier, deuxième, troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.POC.SECONDARY.PHONE.A}

Le champ *Téléphone A du point de contact secondaire* 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 pointer vers le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.POC.SECONDARY.PHONE.B}

Le champ *Téléphone B du point de contact secondaire* 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 faire référence au premier, deuxième, troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.POC.SECONDARY.SCREEN}

Le champ *Nom d'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 faire référence au premier, deuxième, 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 pointer vers le premier, le deuxième, le troisième hôte, etc., 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 pointer vers le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.SITE.ADDRESS.A}

Le champ *Adresse du site 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 pointer vers 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 pointer vers 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 ex. {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 pointer vers 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 *Notes du site* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.SITE.NOTES<1-9>}, pour faire référence au premier, deuxième, 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 pointer vers 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 pointer vers le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir les macros indexées.

{INVENTORY.SOFTWARE}

Le champ *Logiciel* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.SOFTWARE<1-9>}, pour pointer vers le premier, le deuxième, le troisième hôte, etc., 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 pointer vers le premier, le deuxième, le troisième hôte, etc., dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.SOFTWARE.APP.B}

Le champ *Application logicielle 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 pointer vers le premier, le deuxième, le troisième hôte, etc., dans une expression de déclencheur. Voir macros indexées.

{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 pointer vers le premier, le deuxième, le troisième hôte, etc., dans une expression de déclencheur. Voir macros indexées.

{INVENTORY.SOFTWARE.APP.D}

Le champ *Application logicielle 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 pointer vers le premier, le deuxième, le troisième hôte, etc., 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 pointer vers le premier, le deuxième, le troisième hôte, etc., 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 pointer vers le premier, le deuxième, le troisième hôte, etc., 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 pointer vers le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir les 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 les macros indexées.

{INVENTORY.TYPE.FULL}

Le champ *Type (détails complets)* dans l'inventaire de l'hôte.

 Cette macro peut être utilisée avec un index numérique, par exemple {INVENTORY.TYPE.FULL<1-9>}, pour faire référence au premier, deuxième, 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 faire référence au premier, au deuxième, au troisième hôte, etc., 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 faire référence au premier, deuxième, troisième, etc. hôte dans une expression de déclencheur. Voir les 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 faire référence au premier, deuxième, troisième, etc. hôte dans une expression de déclencheur. Voir les 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 Nième 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<1-9>} pour désigner le premier, le deuxième, le troisième hôte, etc. dans une expression de déclencheur. Voir macros indexées.

{ITEM.DESCRIPTION.ORIG}

La description d'origine (avec les macros non résolues) du Nième 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'ID numérique du Nième 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 pointer vers le premier, le deuxième, le troisième, etc. hôte dans une expression de déclencheur. Voir macros indexées.

{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 macros indexées.

{ITEM.KEY.ORIG}

La clé d'origine (avec les macros non résolues) du Nième élément dans l'expression du déclencheur ayant 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 Nième élément dans l'expression de déclencheur qui a provoqué une notification.
 Elle sera résolue en *UNKNOWN* dans l'interface web si la dernière valeur d'historique a été collectée il y a plus longtemps que la durée *Max history display period* (définie dans la section de menu Administration → General).
 Lorsqu'elle est utilisée dans le nom du problème, la macro ne sera pas résolue en la dernière valeur de l'élément lors de l'affichage des événements de problème ; à la place, elle conservera 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 en la dernière valeur de l'élément au moment où le déclencheur s'est activé. Par exemple, si un élément reçoit rapidement deux valeurs, « A » et « B », et que le déclencheur s'active 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 activé et celui où la notification a été créée. Pour éviter cela, vous pouvez utiliser la macro {ITEM.VALUE}, qui est résolue en la valeur au moment où le déclencheur s'active, garantissant 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 texte/log est tronquée à 20 caractères par l'interface web aux emplacements suivants :
- Données opérationnelles ;
- Description du déclencheur ;
- URL du déclencheur ;
- Libellés d'URL du déclencheur ;
- Description du widget de valeur d'élément.
Pour obtenir une valeur complète, vous pouvez utiliser les 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 ex. {ITEM.LASTVALUE<1-9>}, pour désigner le premier, le deuxième, le troisième, etc. élément dans une expression de déclencheur. Voir indexed macros.

{ITEM.LASTVALUE.AGE}

Le temps écoulé entre la collecte de la dernière valeur de l'élément et l'évaluation de la macro.
 Affiché dans un format lisible par l'humain (par ex. 1m 45s).

 Cette macro peut être utilisée avec un index numérique, par ex. {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 déclencheurs (dans Monitoring -> Problems list), les noms d'événements, 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 déclencheurs (dans Monitoring -> Problems list), les noms d'événements, 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.TIME<1-9>}, pour faire référence au premier, au deuxième, au troisième, etc. élément dans une expression de déclencheur. Voir macros indexées.

{ITEM.LASTVALUE.TIMESTAMP}

L'horodatage UNIX du moment où 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 pointer vers le premier, le deuxième, le troisième, etc. élément dans une expression de déclencheur. Voir macros indexées.

{ITEM.LOG.AGE}

L'âge 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 pointer vers 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 du journal a été écrite dans le journal.

 Cette macro peut être utilisée avec un index numérique, par exemple {ITEM.LOG.DATE<1-9>}, pour pointer vers 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 pointer vers 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 pointer vers 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 faire référence au premier, deuxième, troisième, etc. élément dans une expression de déclencheur. Voir macros indexées.

{ITEM.LOG.TIME}

L'heure à laquelle l'entrée du journal a été écrite dans le journal.

 Cette macro peut être utilisée avec un index numérique, par exemple {ITEM.LOG.TIME<1-9>}, pour pointer vers le premier, le deuxième, le troisième, etc. élément dans une expression de déclencheur. Voir macros indexées.

{ITEM.LOG.TIMESTAMP}

L'horodatage UNIX indiquant quand 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 pointer vers 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 d'origine (avec les macros non résolues) de l'élément.

 Cette macro peut être utilisée avec un index numérique, par exemple {ITEM.NAME.ORIG<1-9>}, pour désigner le premier, le deuxième, le troisième, etc. élément dans une expression de déclencheur. Voir les macros indexées.

{ITEM.STATE}

Le dernier état du Nième élément dans l'expression de déclencheur ayant provoqué une notification.
 Valeurs possibles : *Non pris en charge*, *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 macros indexées.

{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 :

- La valeur historique (au moment de l'événement) du Nième 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 des événements ou de l'envoi de notifications ;
- La dernière valeur du Nième élément dans l'expression du déclencheur, si elle est utilisée hors du contexte d'un changement d'état du déclencheur, par exemple lors de l'affichage d'une liste de déclencheurs dans une fenêtre contextuelle de sélection. Dans ce cas, elle fonctionne de la même manière que {ITEM.LASTVALUE}.

Dans le premier cas, elle sera résolue en *UNKNOWN* si la valeur d'historique a déjà été supprimée ou n'a jamais été stockée.
Dans le second cas, et uniquement dans le frontend, elle sera résolue en *UNKNOWN* si la dernière valeur d'historique a été collectée il y a plus longtemps que la durée de *Max history display period* (définie dans la section de menu Administration→General).
La valeur résolue pour les éléments de type texte/log est tronquée à 20 caractères par le frontend aux emplacements suivants :
- Données opérationnelles ;
- Description du déclencheur ;
- URL du déclencheur ;
- Libellés d'URL du déclencheur ;
- Description du widget de valeur d'élément.
Pour obtenir une valeur complète, vous pouvez utiliser les 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 ex. {ITEM.VALUE<1-9>}, pour pointer vers le premier, le deuxième, le troisième, etc. élément dans une expression de déclencheur. Voir 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'humain (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 déclencheurs (dans Monitoring -> Problems list), les noms d'événements, 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.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 Monitoring -> Problems list), 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.VALUE.TIME<1-9>}, pour faire référence au premier, au deuxième, au troisième, etc. élément dans une expression de déclencheur. Voir macros indexées.

{ITEM.VALUE.TIMESTAMP}

L'horodatage UNIX du moment où 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 Nième élément dans l'expression de déclencheur qui a provoqué une notification.
 Valeurs possibles : *0* - Numérique flottant, *1* - Caractère, *2* - Journal, *3* - Numérique 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 : *Non pris en charge*, *Normal*.

 Cette macro peut être utilisée avec un index numérique, par exemple {LLDRULE.STATE<1-9>}, pour pointer vers 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.
 Se résout en l'un des éléments suivants :

- Le proxy du Nième élément dans l'expression du déclencheur (dans les notifications basées sur un déclencheur). Vous pouvez utiliser ici des macros indexées ;
- Le proxy qui a exécuté la découverte (dans les notifications de découverte). Utilisez ici {PROXY.DESCRIPTION}, sans indexation ;
- Le proxy auprès duquel un agent actif s'est enregistré (dans les notifications d'auto-enregistrement). Utilisez ici {PROXY.DESCRIPTION}, 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 macros indexées.

{PROXY.NAME}

Le nom du proxy.
 Se résout en l'un des éléments suivants :

- Le proxy du Nième élément dans l'expression du déclencheur (dans les notifications basées sur les déclencheurs). Vous pouvez utiliser ici des macros indexées ;
- Le proxy qui a exécuté la découverte (dans les notifications de découverte). Utilisez ici {PROXY.NAME}, sans indexation ;

- Le proxy auprès duquel un agent actif s'est enregistré (dans les notifications d'auto-enregistrement). Utilisez ici {PROXY.NAME}, sans indexation.

Cette macro peut être utilisée avec un index numérique, par exemple {PROXY.NAME<1-9>}, pour faire référence au premier, deuxième, 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 tags d'événement de service séparés par des virgules.
 Les tags d'événement de service peuvent être définis dans la section des tags de la configuration du service. Développé en une chaîne vide si aucun tag 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 la configuration du service. Développé en un tableau vide si aucune balise n'existe.

{SERVICE.TAGS.<tag name>}

La valeur du tag d'événement de service référencée par le nom du tag.
 Les tags d'événement de service peuvent être définis dans la section des tags de la configuration du service.
 Un nom de tag 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 tag entre guillemets doivent être échappés avec une barre oblique inverse.

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 les 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 le débogage des 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 les 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 utilisé pour le débogage des 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, la N-ième fonction ou le N-ième hôte dans le champ *Expression* d'un déclencheur :

- **{HOST.IP1}, {HOST.IP2}, {HOST.IP3}** correspondent aux adresses IP du premier, du deuxième et du troisième hôte dans l'expression du déclencheur (si présents).
- **{ITEM.VALUE1}, {ITEM.VALUE2}, {ITEM.VALUE3}** correspondent aux 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}** correspondent aux valeurs de la première, de la deuxième et de la troisième fonction 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 la *Recovery expression*. Par exemple, lors d'un événement de récupération, **{ITEM.VALUE2}** correspondra à 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 graphiques**. Par exemple, `{?func(/{HOST.HOST2}/key,param)}` dans un nom de graphique correspondra à l'hôte du deuxième élément du graphique.

Warning:
Les macros indexées ne seront pas résolues dans un autre contexte que les cas mentionnés ici. Pour les autres contextes, utilisez des macros sans index (**{HOST.HOST}, {HOST.IP}**, etc.).

2 Macros utilisateurs supportées 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'élément

Dans la configuration d'un élément ou d'un prototype d'élément, les macros utilisateur peuvent être utilisées dans les champs suivants :

Emplacement	Macros multiples/mélange avec du texte[1]
Nom de l'élément	oui
Paramètres de clé d'élément	oui
Intervalle de mise à jour	non

Emplacement	Macros multiples/mélange avec du texte[1]
Intervalles personnalisés	non
Délai d'expiration (disponible pour les types d'élément pris en charge)	non
Conserver 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 d'élément ; (*élément agrégé uniquement*) conditions de filtre (nom du groupe d'hôtes et nom du tag))	oui
Surveillance 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 script	
Noms et valeurs des paramètres	oui

Emplacement	Macros multiples/mélange avec du texte[1]
Élément navigateur	
Noms et valeurs des paramètres	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 TELNET	
Nom d'utilisateur	oui
Mot de passe	oui
Script	oui
Zabbix trapper	
Hôtes autorisés	oui
Tags[2]	
Noms des tags	oui
Valeurs des tags	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
Intervalle de mise à jour	non
Intervalle personnalisé	non

Emplacement	Macros multiples/mélange avec du texte[1]
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
Supervision de base de don-nées	
Nom d'utilisateur	oui
Mot de passe	oui
Requête SQL	oui
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

Emplacement	Macros multiples/mélange avec du texte[1]
Mot de passe pour l'authentification HTTP	oui
Filtres	
Expression régulière	oui
Substitutions	
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 :

Emplacement	Macros multiples/mélange avec du texte[1]
Adresse pour les agents actifs > *Port* (si le proxy appartient à un groupe)	non
Adresse et port de l'interface (pour un proxy passif)	non

Emplacement	Macros multiples/mélange avec du texte[1]
Délais d'expiration pour les types d'éléments	non

Groupes de proxys

Dans une configuration de groupe de proxys, les macros utilisateur peuvent être utilisées dans les champs suivants :

Emplacement	Macros multiples/mélange avec du texte[1]
Période de basculement	non
Nombre minimal de proxys	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

Emplacement	Macros multiples/mélange avec du texte[1]
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	oui
Webhooks	
Script JavaScript	non
Nom du paramètre du script JavaScript	non
Valeur du paramètre du script JavaScript	oui
Tableaux de bord	
Colonne du type de données *Texte* dans le widget de tableau de bord *Top hosts*	oui
Paramètre *Description* dans les widgets de tableau de bord *Item value* et *Gauge*	oui
Paramètre *Text* de l'étiquette principale/secondaire dans le widget de tableau de bord *Honeycomb*	oui
Paramètre *URL*[3] dans le widget de tableau de bord *URL*	oui
Utilisateurs → *Utilisateurs* → *Média*	
Période d'activité	oui
Administration → *Général* → *Interface graphique*	
Temps de travail	non
Administration → *Général* → *Délais d'expiration*	
Délais d'expiration pour les types d'éléments	non
Administration → *Général* → *Connecteurs*	

Emplacement	Macros multiples/mélange avec du texte[1]
URL	oui
Nom d'utilisateur	oui
Mot de passe	oui
Jeton Bearer	oui
Délai d'expiration	non
Proxy HTTP	oui
Fichier de certificat SSL	oui
Fichier de clé SSL	oui
Mot de passe de la clé SSL	oui
Alertes → Types de média → Modèles de message	
Sujet	oui
Message	oui
Alertes → Types de média → Script	
Paramètres du script	oui
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" ; les macros secrètes sont recommandées)	oui

Pour une liste complète de toutes les macros prises en charge dans Zabbix, voir les macros prises en charge.

Notes de bas de page

[1] Si plusieurs macros dans un champ ou des macros mélangées à du texte ne sont pas prises en charge pour l'emplacement, une seule macro doit remplir tout le champ.

[2] Les macros utilisées dans les noms et les valeurs des tags sont résolues uniquement lors du processus de génération d'événements.

[3] Les URL qui contiennent une macro secrète ne fonctionneront pas, car la macro qu'elles contiennent sera résolue comme "******".

7 Symboles des unités

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 web de Zabbix prend en charge les suffixes de temps suivants dans la configuration des entités :

- **s** - secondes *(lorsqu'il est utilisé, fonctionne de la même manière que la valeur brute)*
- **m** - minutes
- **h** - heures
- **d** - jours
- **w** - semaines
- **M** - mois *(fonctions de tendances uniquement)*
- **y** - années *(fonctions de tendances uniquement)*

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 le sont pas ; utilisez 90m à la place.

Suffixes de taille de mémoire

Zabbix prend en charge les suffixes de taille de mémoire suivants :

- **K** - kilo-octets
- **M** - mégaoctets
- **G** - gigaoctets
- **T** - téraoctets

Suffixes des valeurs d'élément

Les suffixes peuvent également être utilisés pour afficher les valeurs numériques des éléments dans un format lisible par l'humain.

Pour activer cela, 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 plus grandes unités de temps non nulles
- **uptime** - temps écoulé au format hh:mm:ss ou N jours, hh:mm:ss
- **unixtime** - horodatage Unix, formaté comme yyyy.mm.dd hh:mm:ss

Les règles supplémentaires suivantes s'appliquent à la façon 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 plus grandes unités de temps non nulles sont affichées (par exemple, 1M 10d 4h).
  - Si une unité vaut zéro et se trouve entre deux unités non nulles, elle est omise (par exemple, 10d 56m au lieu de 10d 0h 56m).

Lorsque *Units* est utilisé, les suffixes multiplicateurs suivants sont automatiquement appliqués aux valeurs des éléments :

- **K**, **M**, **G**, **T** - kilo, méga, giga, téra
- **P**, **E**, **Z**, **Y** - péta, exa, zetta, yotta *(ceux-ci sont appliqués uniquement dans le frontend)*

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 codée en dur d'unités à exclure de la conversion, composée de `ms`, `rpm`, `RPM`, `%`. Cette liste a été abandonnée ; la méthode correcte pour empêcher la conversion de telles unités est donc `!ms`, `!rpm`, `!RPM`, `!%`.

8 Syntaxe des périodes 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
```

9 Exécution de commande

Zabbix utilise des fonctionnalités communes pour les vérifications externes, les paramètres utilisateur, les éléments system.run, les scripts d'alerte personnalisés, les commandes à distance et les scripts utilisateur.

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 distantes 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 ayant échoué est collecté et disponible dans le frontend (où le résultat de l'exécution est affiché).
- Une entrée de journal supplémentaire peut être créée pour les commandes distantes exécutées sur l'agent/proxy Zabbix en activant le paramètre LogRemoteCommands sur agent/proxy.

Messages du frontend et entrées de journal possibles pour les commandes/scripts ayant échoué :

- Contenu de la sortie d'erreur standard et de la sortie standard pour les exécutions ayant échoué (le cas échéant).
- "Le processus s'est terminé avec le code : N." (pour une sortie vide et un code de sortie différent de 0).
- "Le processus a été arrêté par le signal : N." (pour un processus terminé par un signal, sous Linux uniquement).
- "Le processus s'est terminé de manière inattendue." (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

10 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 1.4 et ne doit pas être plus récent que 8.0.

Il peut être nécessaire de revoir la configuration des anciens agents, car certains paramètres ont changé, par exemple les paramètres liés à la journalisation pour les versions antérieures à 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 le dernier agent pris en charge.

Remarques pour Windows XP

- Sur Windows XP 32 bits, n'utilisez pas d'agents Zabbix plus récents que 6.0.x ;
- Sur Windows XP/Server 2003, n'utilisez pas de modèles d'agent plus récents que Zabbix 4.0.x. Les nouveaux modèles utilisent des compteurs de performance 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

Version du serveur	Version de proxy *actuelle*	Version de proxy *obsolète*	Version de proxy *non prise en charge*
8.0	8.0	7.0, 7.2, 7.4	Antérieure à 7.0 ; supérieure à 8.0

Fonctionnalités prises en charge par les proxys :

Version du proxy	Mise à jour des données	Mise à jour de la configuration	Tâches
Actuelle	Oui	Oui	Oui
Obsolète	Oui	Non	Commandes distantes (par ex., scripts shell) ; Vérifications immédiates des valeurs d'élément (c.-à-d. *Exécuter maintenant*) ; Remarque : les tests de prétraitement avec une valeur réelle ne sont pas pris en charge.
Non prise en charge	Non	Non	Non

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.

12 Bibliothèque de liens dynamiques Zabbix sender pour Windows

Aperçu

Dans un environnement Windows, les applications peuvent envoyer des données au serveur/proxy Zabbix en utilisant la bibliothèque de liens dynamiques 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, veillez à sélectionner « No encryption » dans *Encryption* et « Archive » dans *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 liens dynamiques 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-le avec la bibliothèque zabbix_sender.lib.

Voir aussi

- un exemple d'un utilitaire Zabbix sender simple implémenté avec la bibliothèque de liens dynamiques Zabbix sender pour illustrer l'utilisation de la bibliothèque ;
- le fichier zabbix_sender.h pour les fonctions d'interface de la bibliothèque de liens dynamiques Zabbix sender. Ce fichier contient une documentation expliquant l'objectif de chaque fonction d'interface, ses arguments et sa valeur de retour.

13 Mise à niveau de la surveillance des services

Aperçu Dans Zabbix 6.0, la fonctionnalité surveillance des services a été retravaillée de manière significative (voir Quoi de neuf dans Zabbix 6.0.0 pour la liste des modifications).

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 : logicielles et matérielles. Après une mise à niveau, toutes les dépendances deviendront égales.

Si un service "Service enfant" a été précédemment lié au "Service parent 1" via une dépendance matérielle et en plus au "Service parent 2" via une dépendance logicielle, après une mise à niveau, le "Service enfant" aura deux services parents "Service parent 1" et "Service parent 2".

Le mappage basé sur les déclencheurs entre les problèmes et les services a été remplacé par le mappage basé sur les tags. Dans Zabbix 6.0 et versions ultérieures, le formulaire de configuration de service a un nouveau paramètre *Tags de problème*, qui permet de spécifier une ou plusieurs paires de noms et de valeurs de tags pour la correspondance des problèmes. Les déclencheurs qui ont été liés à un service recevront un nouveau tag `ServiceLink` : `<trigger ID>:<trigger name>` (la valeur du tag sera tronquée à 32 caractères). Les services liés recevront un tag 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 calendrier 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 les tags de service.

Lors d'une mise à niveau :

- Les SLA identiques définis pour chaque service seront regroupés et un SLA sera créé pour chaque groupe.
- Chaque service concerné recevra un tag spécial `SLA:<ID>` et le même tag sera spécifié dans le paramètre *Service tags* du SLA correspondant.
- La date de création du service, une nouvelle métrique dans les rapports SLA, sera définie sur 01/01/2000 00:00 pour les services existants.

14 Autres problèmes

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 ex. via SSH) à l'hôte exécutant Zabbix. Cela peut provoquer le plantage du démon Zabbix lors de la déconnexion. Dans ce cas, vous obtiendrez 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 à cause du paramètre systemd par défaut `RemoveIPC=yes` configuré dans `/etc/systemd/logind.conf`. Lorsque vous vous déconnectez du système, les sémaphores créés précédemment par Zabbix sont supprimés, ce qui provoque le plantage.

Citation de la documentation systemd :

RemoveIPC=

Contrôle si les objets IPC System V et POSIX appartenant à l'utilisateur doivent être supprimés lorsque l'utilisateur se déconnecte complètement. Prend un argument booléen. Si activé, l'utilisateur ne peut pas après la fin de la dernière session de l'utilisateur. Cela couvre les sémaphores System V, la mémoire part et les files de messages, ainsi que la mémoire partagée POSIX et les files de messages. Notez que les obje l'utilisateur root et des autres utilisateurs système sont exclus de l'effet de ce paramètre. La valeur pa

Il existe 2 solutions à ce problème :

1. (recommandé) Cesser d'utiliser le compte *zabbix* pour autre chose que les processus Zabbix, et créer un compte dédié pour les autres usages.
2. (non recommandé) Définir `RemoveIPC=no` dans `/etc/systemd/logind.conf` puis redémarrer le système. Notez que `RemoveIPC` est un paramètre global au système ; le modifier affectera l'ensemble du système.

Utilisation de l'interface Zabbix derrière le 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 du 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
# ..
```

16 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.

21 Guides de référence rapide

Aperçu

Cette section de la documentation contient des procédures rapides pour configurer Zabbix afin de répondre à certains objectifs de supervision couramment requis.

Elle est conçue pour les nouveaux utilisateurs de Zabbix et peut être utilisée comme guide de navigation vers d'autres sections de la documentation contenant les informations nécessaires à la réalisation de la tâche.

Les guides de référence rapide suivants sont disponibles :

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

1 Surveiller Linux avec l'agent Zabbix

Introduction Cette page vous guide à travers les étapes requises pour démarrer la surveillance de base des machines Linux avec Zabbix. Les étapes décrites dans ce guide peuvent être appliquées à n'importe quel système d'exploitation basé sur Linux.

À qui s'adresse ce guide

Ce guide est conçu pour les nouveaux utilisateurs de Zabbix et contient l'ensemble minimum d'étapes requises pour permettre la surveillance de base de Linux. Si vous recherchez des options de personnalisation approfondies ou si vous avez besoin d'une configuration plus avancée, voir la section Configuration du manuel Zabbix.

Prérequis

Avant de poursuivre ce guide, vous devez télécharger et installer Zabbix server et Zabbix frontend conformément aux instructions correspondant à votre système d'exploitation.

Installer l'agent Zabbix L'agent Zabbix est le processus chargé de collecter les données.

Vérifiez la version de votre serveur Zabbix :

```
zabbix_server -V
```

Installez l'agent Zabbix de la même version (recommandé) sur la machine Linux que vous souhaitez superviser. 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 paquets 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 surveillance L'agent Zabbix peut collecter des métriques en mode actif ou passif (simultanément).

> **Note:**
> Une vérification passive est une simple demande de données. Le serveur ou proxy Zabbix demande certaines données (par exemple, la charge du processeur) et l'agent renvoie le résultat au serveur. Les contrôles actifs nécessitent un traitement plus complexe. L'agent doit d'abord récupérer à partir du ou des serveurs, une liste d'éléments pour un traitement indépendant, puis renvoyer les données en masse. Voir Contrôles d'agents passifs et actifs pour plus d'informations.

Les modèles de surveillance fournis par Zabbix offrent 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 surveillance identiques, mais en utilisant des protocoles de communication différents.

La configuration ultérieure de Zabbix dépend si vous sélectionnez un modèle pour active ou passive pour la vérifications de l'agent Zabbix.

Contrôle passifs 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 éventuellement le port) de votre serveur Zabbix au paramètre server. Par exemple :

```
Server=192.0.2.0:10051
```

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

3. Redémarrez l'agent Zabbix.

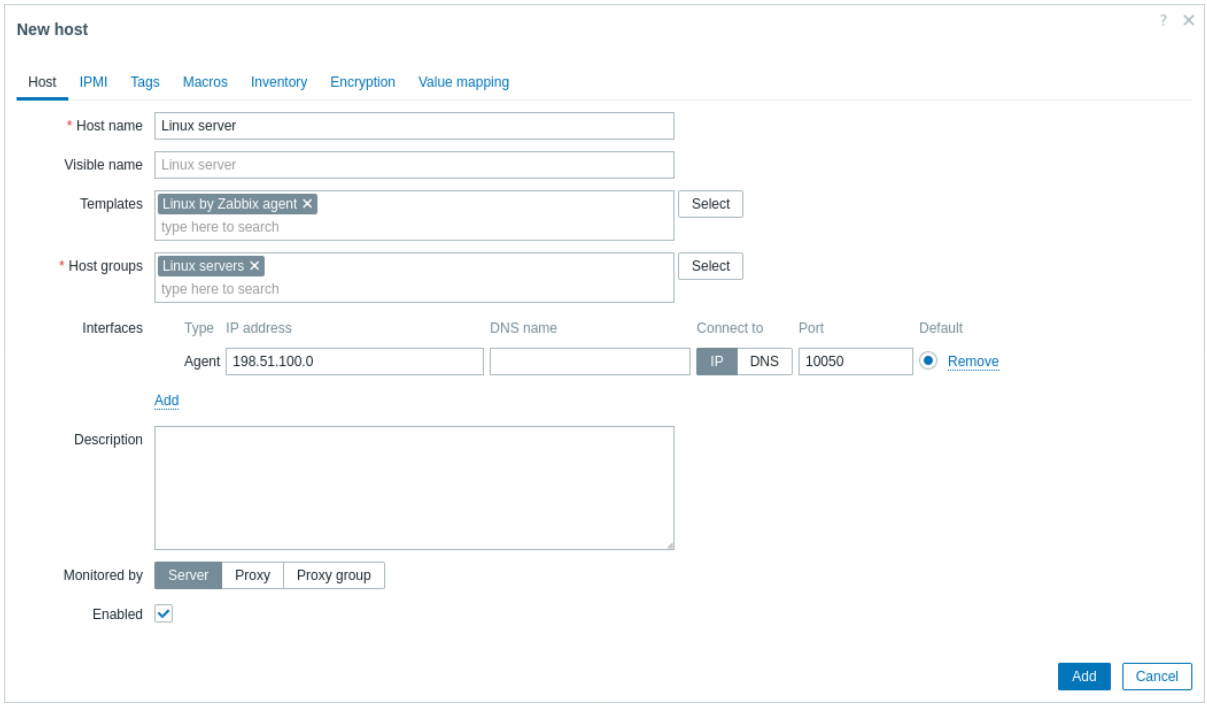
```
systemctl restart zabbix-agent
```

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.

Contrôle actifs 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 éventuellement le port) de votre serveur Zabbix au paramètre ServerActive. Par exemple :

```
ServerActive=192.0.2.0:10051
```

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 le frontend 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 initie une connexion au serveur et s'identifie à l'aide de la valeur Hostname. Le serveur fournit ensuite la configuration de supervision pour 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.

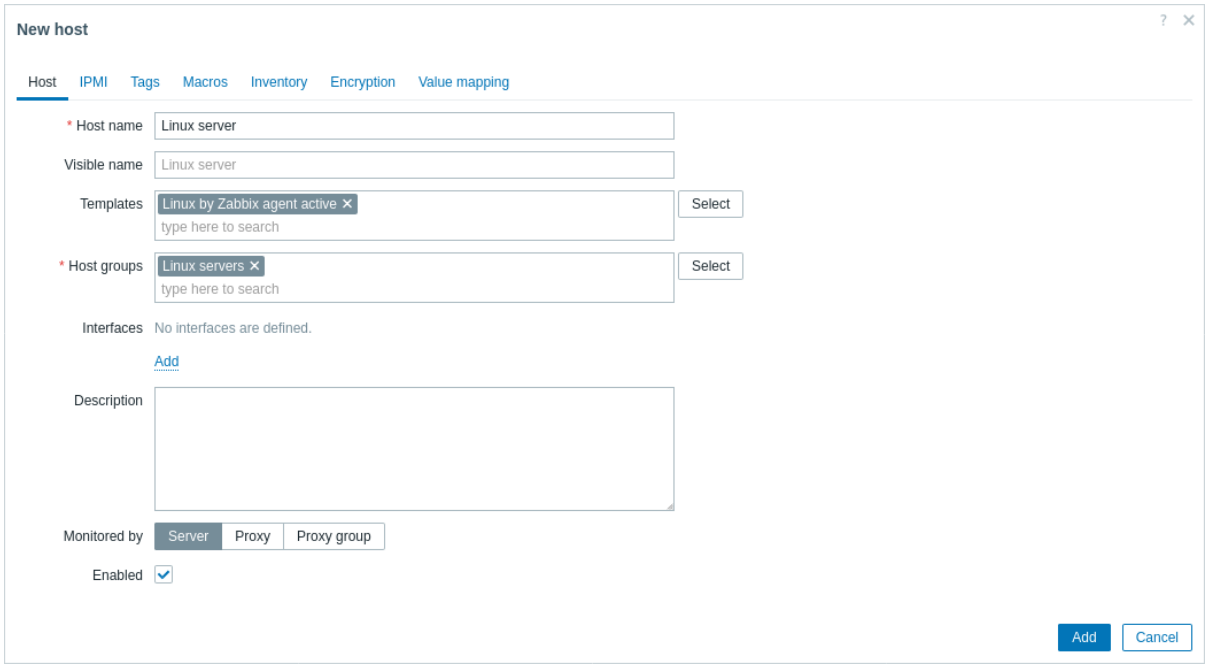
```
systemctl restart zabbix-agent
```

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 contrôles actifs, car l'agent initie la connexion au serveur au lieu d'attendre une connexion provenant du serveur.



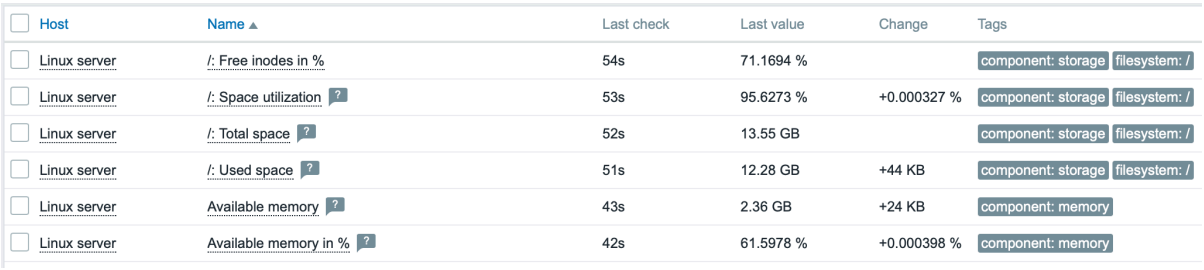
3. Cliquez sur *Add* pour ajouter l'hôte. Cet hôte représentera la machine Linux surveillée.

Afficher les métriques collectées Toutes nos félicitations! À ce stade, Zabbix surveille déjà votre machine Linux.

Pour afficher les métriques collectées, ouvrez *Surveillance->Hôtes* section de menu et cliquez sur *Dernières données* à côté de l'hôte.



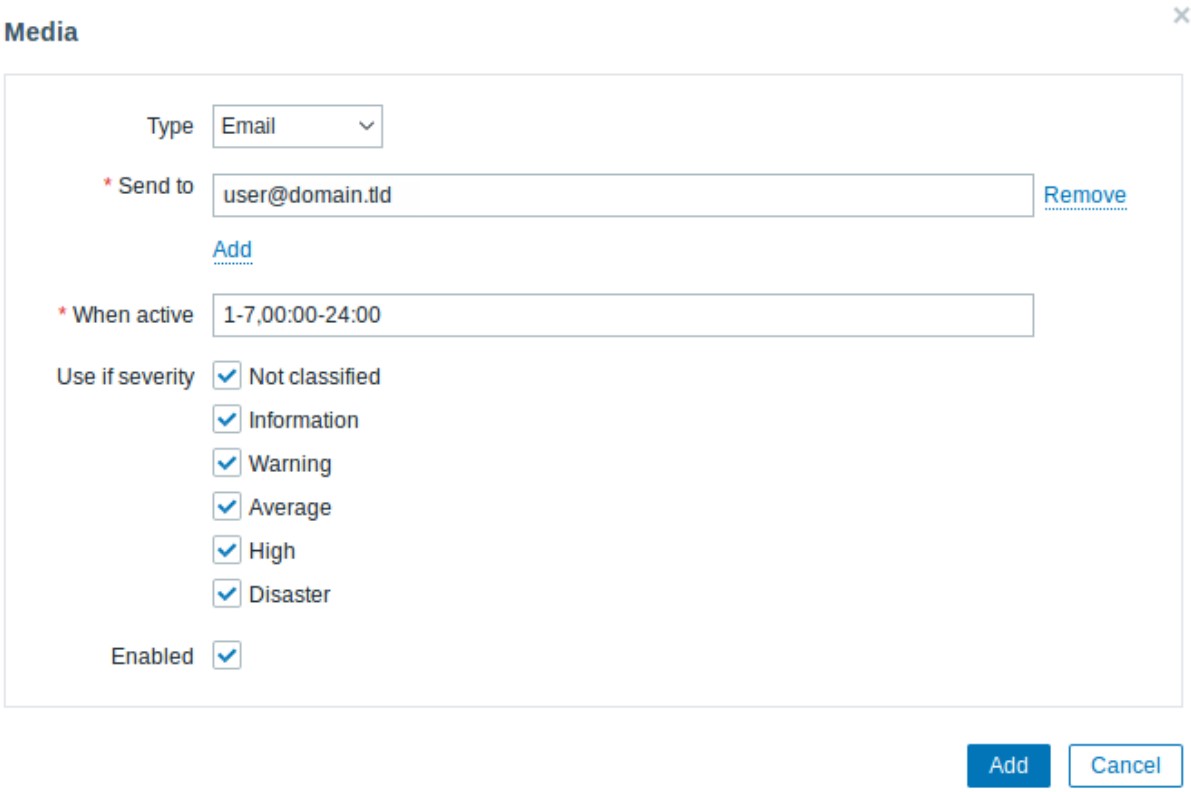
Cette action ouvrira une liste de toutes les dernières métriques collectées auprès de l'hôte du serveur Linux.



Host	Name ▲	Last check	Last value	Change	Tags
Linux server	/: Free inodes in %	54s	71.1694 %		component: storage filesystem: /
Linux server	/: Space utilization	53s	95.6273 %	+0.000327 %	component: storage filesystem: /
Linux server	/: Total space	52s	13.55 GB		component: storage filesystem: /
Linux server	/: Used space	51s	12.28 GB	+44 KB	component: storage filesystem: /
Linux server	Available memory	43s	2.36 GB	+24 KB	component: memory
Linux server	Available memory in %	42s	61.5978 %	+0.000398 %	component: memory

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

1. Accédez à *User settings -> Profile*, passez à l'onglet *Media* et ajoutez votre adresse e-mail.



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

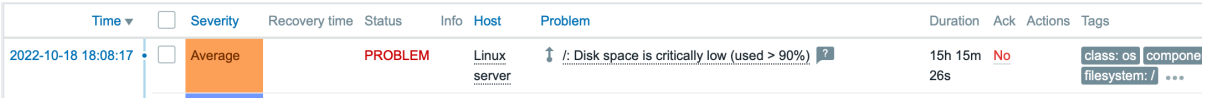
La prochaine fois que Zabbix détectera un problème, vous devriez recevoir une alerte par e-mail.

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

```
chat /dev/urandom | somme md5
```

Vous devrez peut-être exécuter plusieurs processus md5sum pour que la charge du processeur dépasse le seuil.

Lorsque Zabbix détecte le problème, il apparaîtra dans la section *Surveillance->Problèmes*.



Time ▼	Severity	Recovery time	Status	Info	Host	Problem	Duration	Ack	Actions	Tags
2022-10-18 18:08:17	Average		PROBLEM		Linux server	/: Disk space is critically low (used > 90%)	15h 15m 26s	No		class: os compone filesystem: /

Si les alertes sont [configurées] (#set-up-problem-alerts), vous recevrez également la notification du problème.

Voir aussi :

- Création d'un élément - comment commencer à surveiller des métriques supplémentaires (surveillance personnalisée sans modèles).
- Éléments d'agent Zabbix, Éléments d'agent Zabbix pour Windows - liste complète des métriques que vous pouvez surveiller à l'aide de l'agent Zabbix sous Windows.

- 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).
- Installation à partir de paquets - comment installer les composants Zabbix à l'aide des paquets RPM et DEB officiels pour diverses distributions Linux, afin de garantir l'accès aux dernières fonctionnalités et corrections de bogues.

2 Surveiller Windows avec l'agent Zabbix

Introduction Cette page vous guide à travers les étapes nécessaires pour démarrer la surveillance de base des machines Windows avec Zabbix.

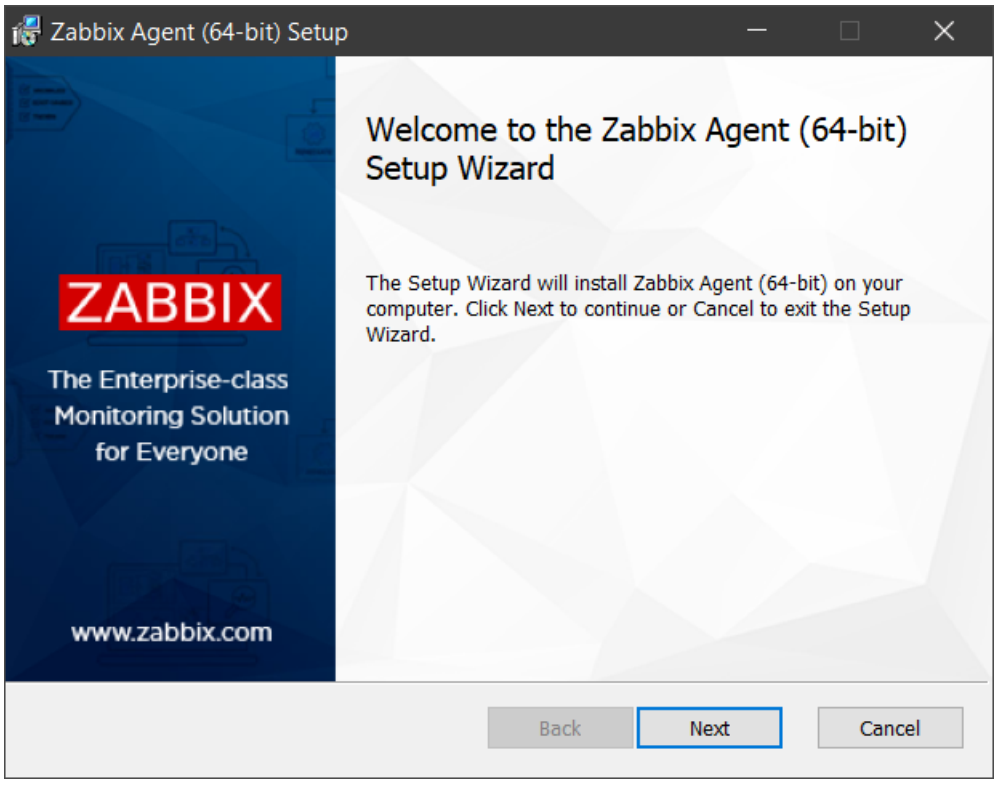
À qui s'adresse ce guide

Ce guide est conçu pour les nouveaux utilisateurs de Zabbix et contient l'ensemble minimal d'étapes nécessaires pour activer la surveillance de base de votre machine Windows.
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 avec ce guide d'installation, vous devez télécharger et installer le serveur Zabbix et le frontend Zabbix conformément aux instructions correspondant à votre système d'exploitation.

Installer l'agent Zabbix L'agent Zabbix est le processus chargé de collecter les données. Vous devez l'installer sur la machine Windows que vous souhaitez surveiller. Suivez les instructions d'installation de l'agent Zabbix 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 demande de données. Le serveur ou le proxy Zabbix 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 auprès du ou des serveurs une liste d'éléments à traiter de manière indépendante, puis renvoyer 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 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. Ces modèles fourniront des résultats de supervision identiques, mais en utilisant des protocoles de communication différents.

La configuration supplémentaire de Zabbix dépend de votre choix d'un modèle pour les vérifications de l'agent Zabbix actives ou passives.

Vérifications passives Interface web Zabbix

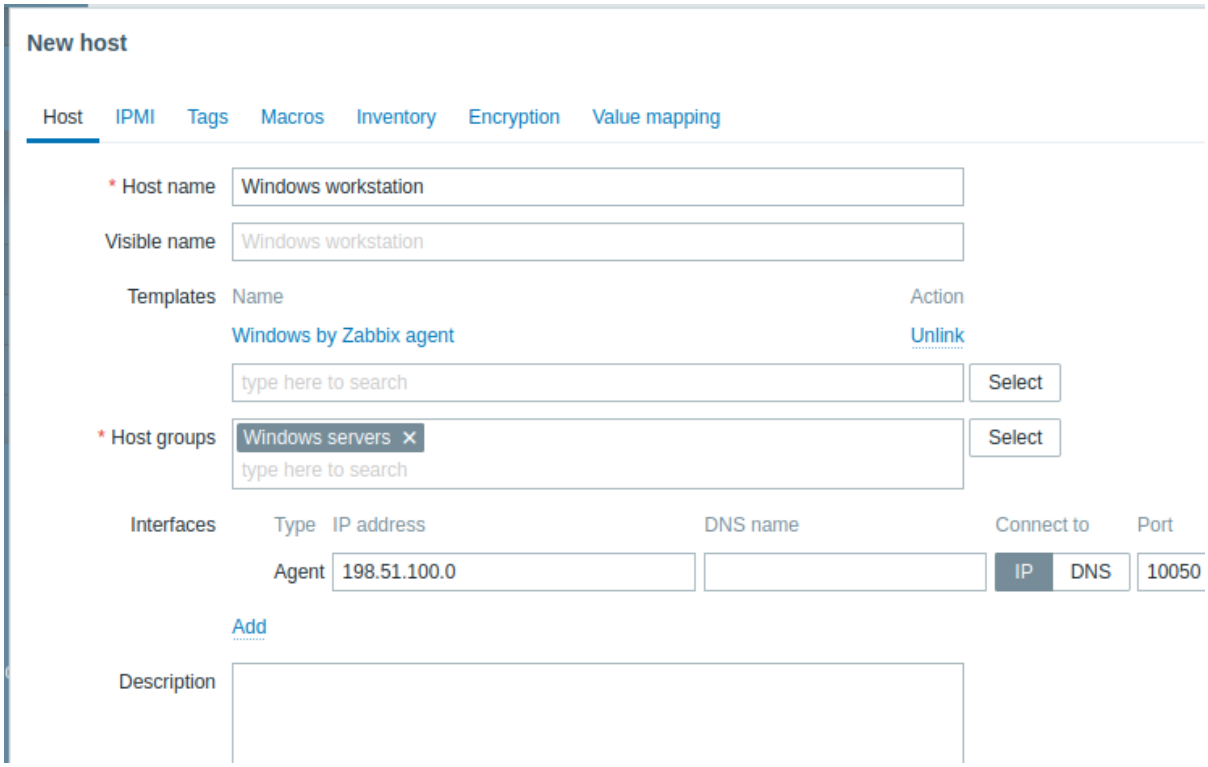
1. Connectez-vous à l'interface web Zabbix.

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 *Templates*, saisissez ou sélectionnez *Windows by Zabbix agent*.



Agent Zabbix

Pour les contrôles passifs, 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 est déjà mis à jour. Sinon, vous devez le spécifier manuellement. Accédez au dossier `C:\Program files\Zabbix Agent`, ouvrez le fichier *zabbix_agentd.conf* et ajoutez l'IP/le DNS de votre serveur Zabbix au paramètre *Server*.

Exemple :

```
Server=192.0.2.22
```

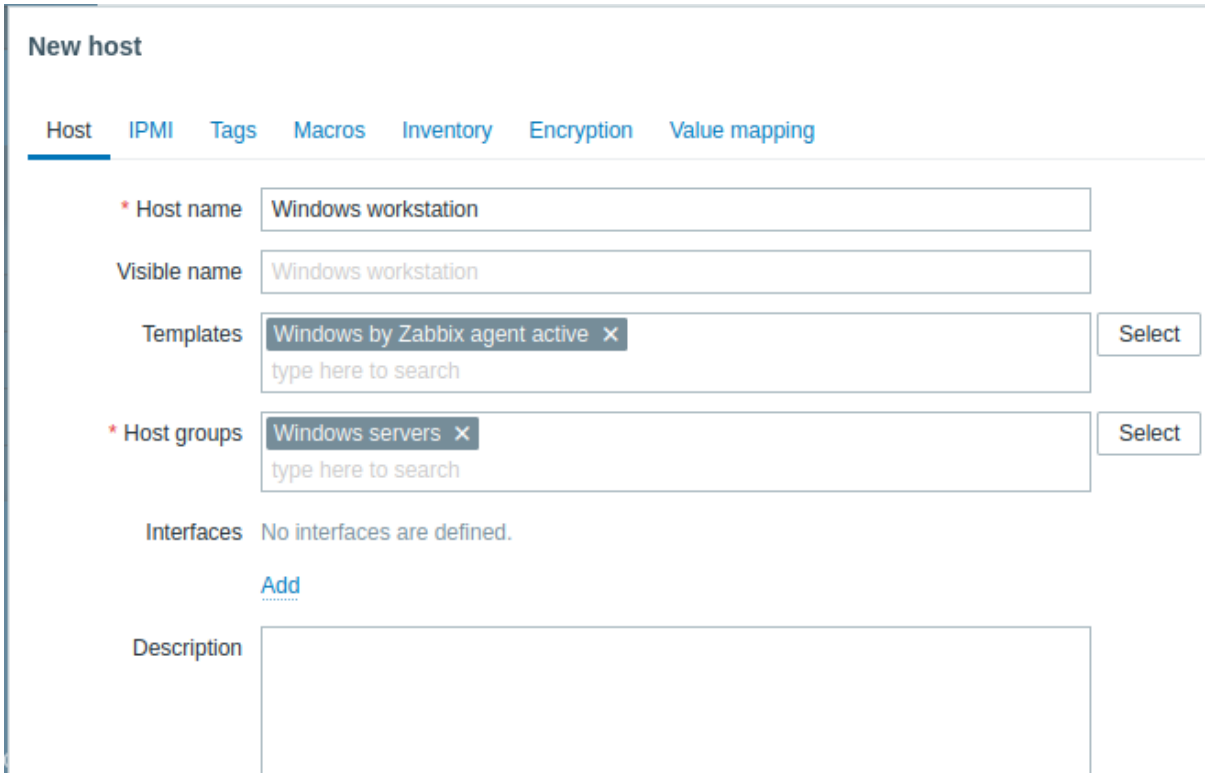
Vérifications actives Interface web Zabbix

1. Connectez-vous à l'interface web 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 :

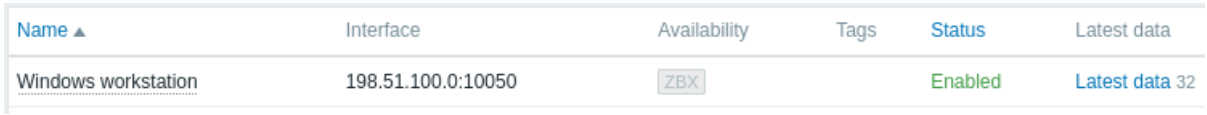
- Le nom de l'hôte que vous avez créé dans l'interface web de Zabbix au paramètre *Hostname*.
- L'adresse IP ou le nom DNS de votre machine serveur Zabbix au paramètre *ServerActive* (peut être prérempli si vous l'avez fourni lors de l'installation de l'agent Zabbix).

Exemple :

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

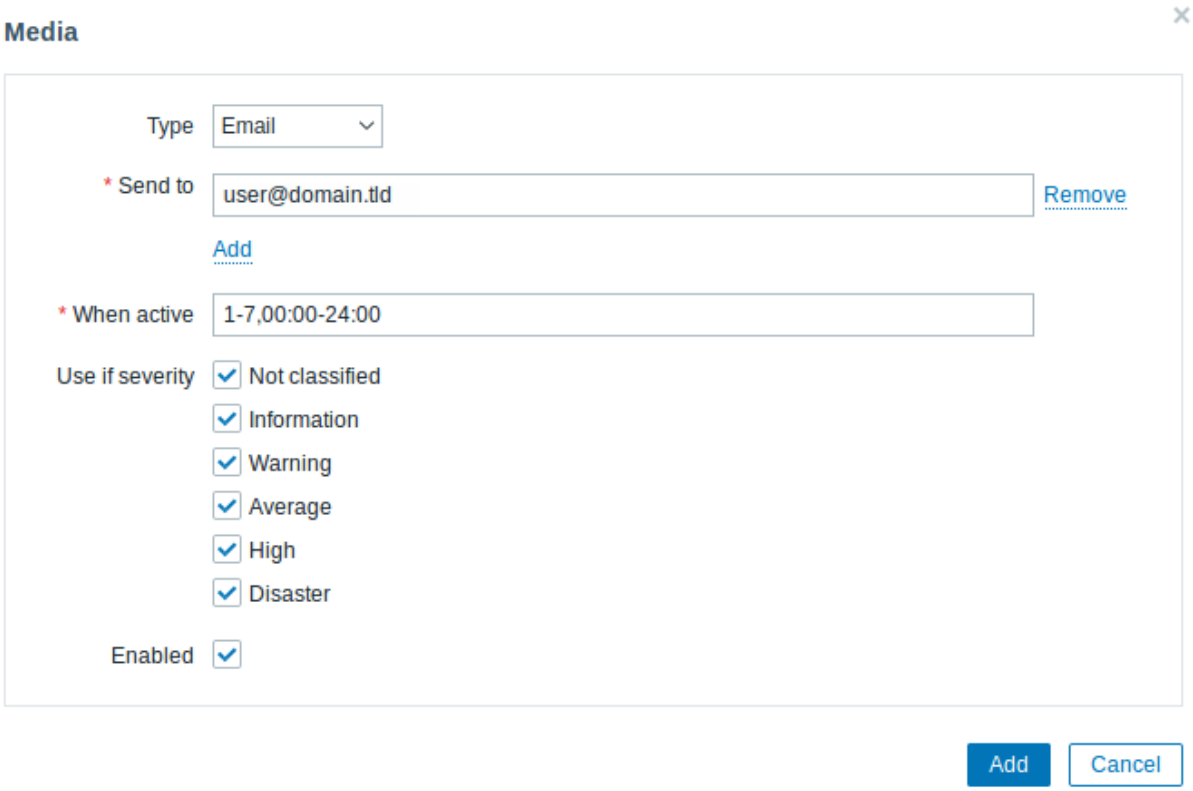
Voir les métriques collectées Félicitations ! À ce stade, Zabbix surveille déjà votre machine Windows.

Pour voir 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 dans votre infrastructure à l'aide de différentes méthodes. Ce guide fournit les étapes de configuration pour l'envoi d'alertes par e-mail.

1. Accédez à *User settings -> Profile*, passez à 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.

Note:
Sous Windows, vous pouvez utiliser l'utilitaire CpuStres pour simuler une charge CPU élevée et ainsi recevoir une alerte 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 problèmes - comment créer des scénarios d'alerte en plusieurs étapes (par ex., 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).

3 Surveiller Apache via HTTP

Introduction Cette page montre un moyen simple et rapide de démarrer la surveillance d'un serveur Web Apache sans installer de logiciel supplémentaire.

À qui s'adresse ce guide

Ce guide est conçu pour les nouveaux utilisateurs de Zabbix et contient l'ensemble minimum des étapes requises pour activer la surveillance de base de votre installation Apache. Si vous recherchez des options de personnalisation approfondies ou avez besoin d'une configuration plus avancée, voir la section Configuration du manuel Zabbix.

Prérequis

Avant de procéder à ce guide d'installation, vous devez télécharger et installer Zabbix serveur et l'interface Zabbix selon les instructions de 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 d'état 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 à partir de 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 (**replace 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. Redémarrer Apache

Sur un système basé sur RHEL, exécutez :

```
sudo systemctl restart httpd
```

Sur Debian/Ubuntu, exécutez :

```
sudo systemctl restart apache2
```

5. Pour vérifier si tout est correctement configuré, exécutez (**replace 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.

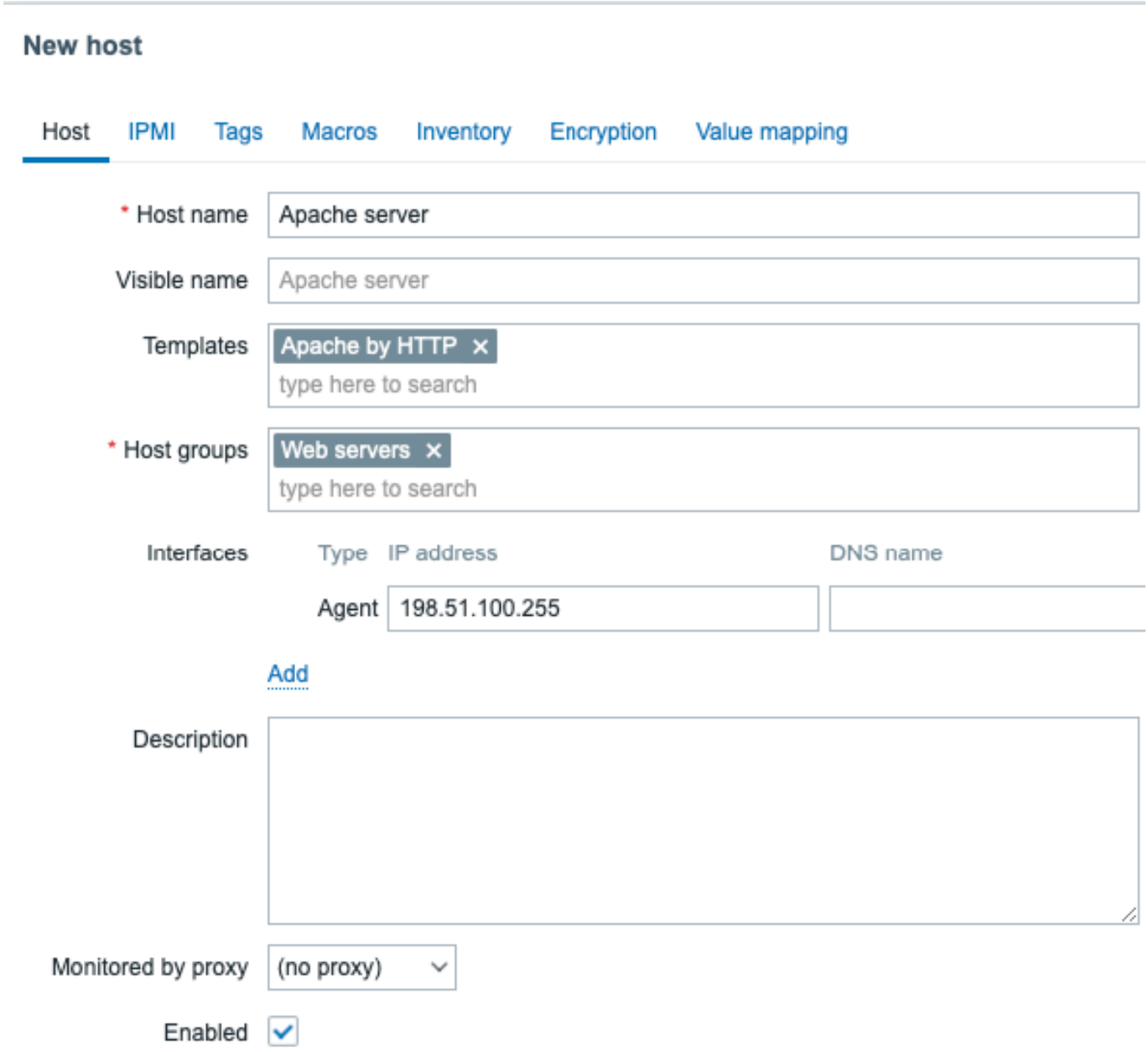
Configurer Zabbix pour la surveillance 1. Connectez-vous à l'interface Zabbix.

2. Créer un hôte dans l'interface Web Zabbix.

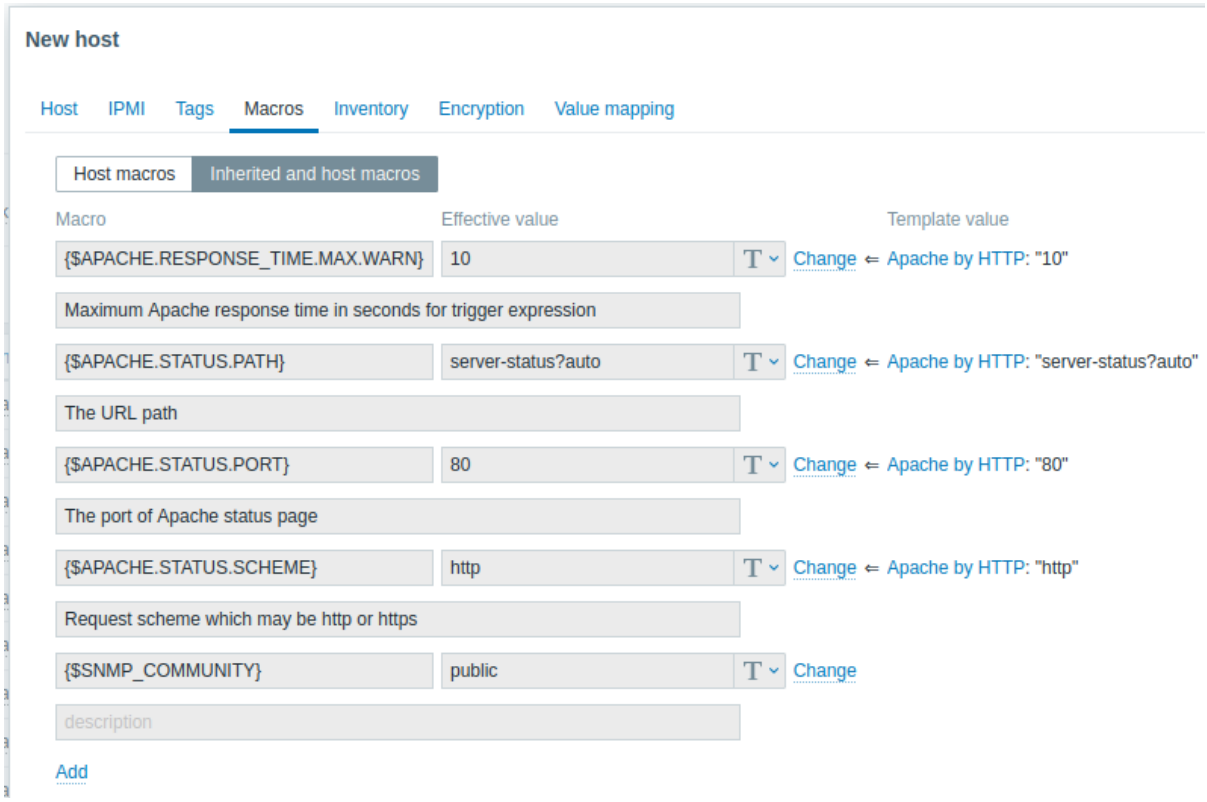
Cet hôte représentera votre serveur Apache.

3. Dans le paramètre *Interfaces*, ajoutez l'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 de 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 *Macros héritées et de l'hôte*. Vérifiez que les valeurs des macros {$APACHE.STATUS.PORT} et {$APACHE.STATUS.SCHEME} correspondent à vos paramètres d'installation. Par défaut, le port est 80 et le schéma est http. Modifiez les valeurs de la macro si vous utilisez un port et/ou un schéma différent.



Afficher les métriques collectées Toutes nos félicitations! À ce stade, Zabbix surveille déjà votre serveur Web Apache.

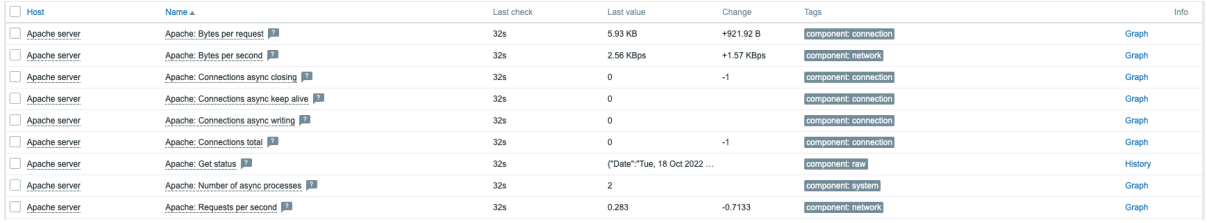
Pour afficher les métriques collectées, ouvrez *Surveillance->Hôtes* section menu et cliquez sur *Tableaux de bord* à côté de l'hôte.



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

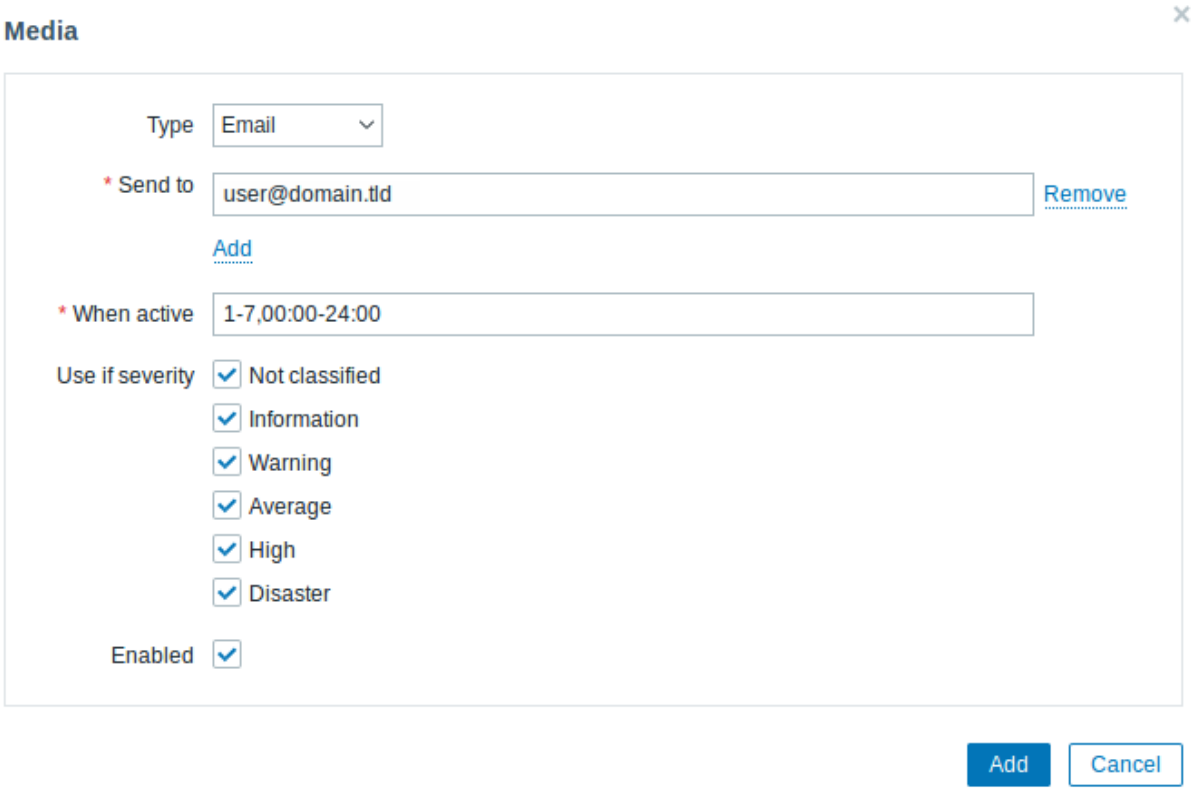


Alternativement, à partir de *Surveillance->Hôtes*, vous pouvez cliquer sur *Dernières données* pour afficher toutes les dernières données collectées sous forme de liste.



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

1. Accédez à *User settings -> Profile*, passez à 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 simuler un problème réel et recevoir une alerte de problème de test :

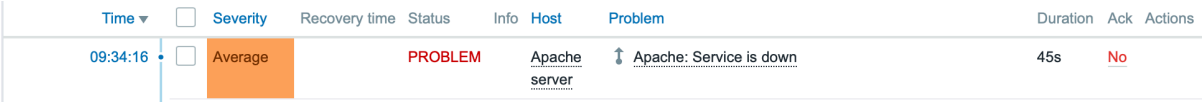
1. Ouvrez la configuration de l'hôte *Serveur apache* dans Zabbix.

2. Basculez vers l'onglet Macros et sélectionnez *Macros héritées et hôtes*.

3. Appuyez sur *Modifier* à côté de la macro {$APACHE.STATUS.PORT} et définissez un autre port.

4. Appuyez sur Mettre à jour pour enregistrer la configuration de l'hôte.

5. Dans quelques minutes, Zabbix détectera le problème *Le service Apache est en panne*, car il ne peut plus se connecter à l'instance. Il apparaîtra dans la section Surveillance->Problèmes.



Si les alertes sont [configurées] (#set-up-problem-alerts), vous recevrez également la notification du problème.

6. Modifiez à nouveau la valeur de la macro pour résoudre le problème et continuez à surveiller 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).

4 Surveiller MySQL avec l'agent Zabbix 2

Introduction

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

Pour surveiller un serveur MySQL, plusieurs approches sont possibles : Zabbix agent, Zabbix agent 2 ou la norme Open Database Connectivity (ODBC). Ce guide se concentre principalement sur la surveillance d'un serveur MySQL avec Zabbix agent 2, qui constitue l'approche **recommandée** grâce à sa configuration fluide dans divers environnements. Cependant, cette page propose également des instructions pour les autres approches ; n'hésitez donc pas à choisir celle qui correspond le mieux à vos besoins.

À qui s'adresse ce guide

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

Prérequis

Avant de poursuivre ce guide, vous devez télécharger et installer Zabbix server, Zabbix frontend et Zabbix agent 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 doit y avoir accès 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 possède plus de privilèges que nécessaire pour une simple surveillance (privilèges pour DROP des bases de données, DELETE des entrées de tables, etc.). Il est donc nécessaire de 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;
```

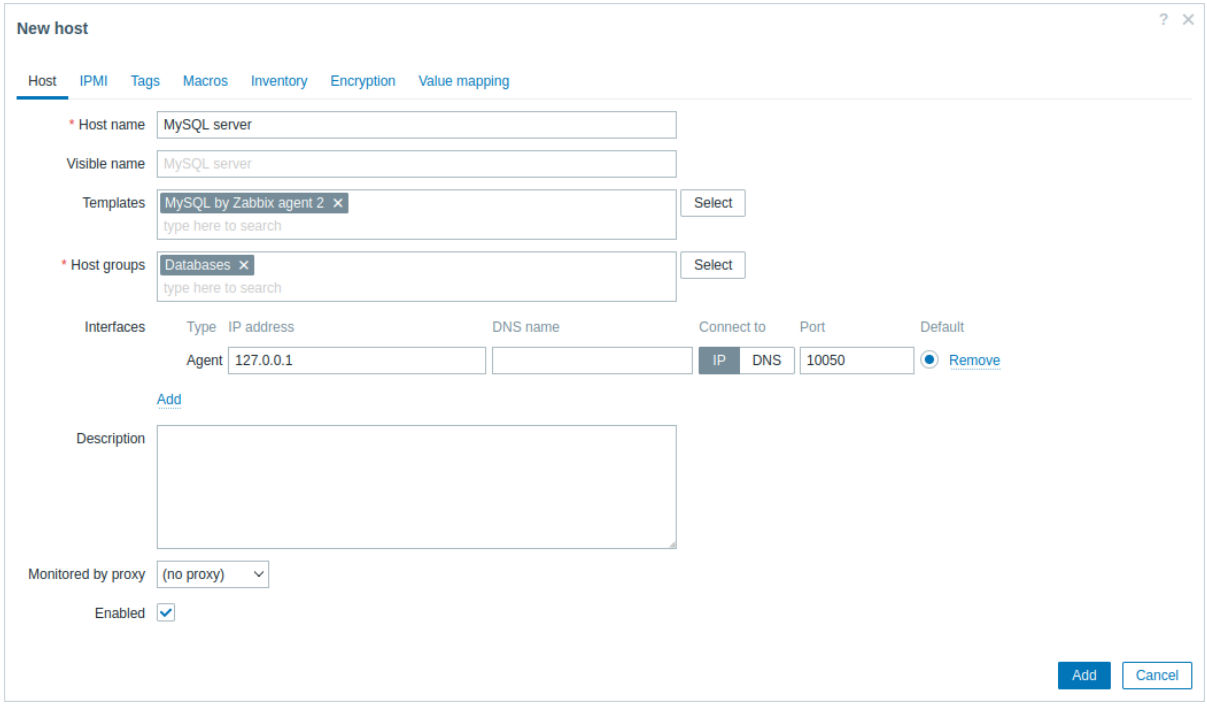
Une fois l'utilisateur créé, vous pouvez passer à l'étape suivante.

Configurer l'interface web Zabbix

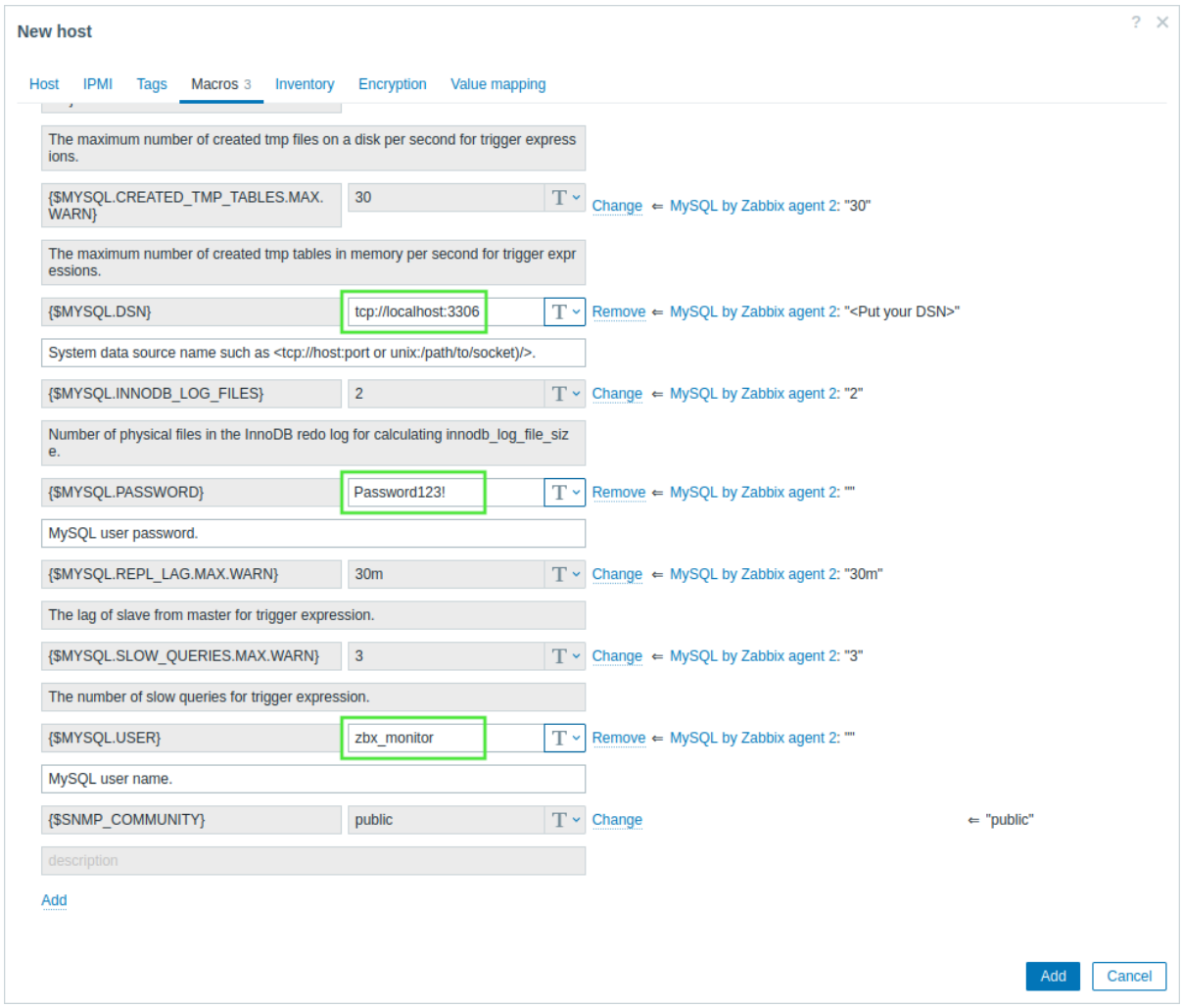
1. Connectez-vous à l'interface web Zabbix.

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 Zabbix agent 2.



- 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 :
  - {$MYSQL.DSN} - définissez la source de données du serveur MySQL (la chaîne de connexion d'une session nommée du fichier de configuration du plugin MySQL de Zabbix agent 2). Ce guide utilise la source de données par défaut « tcp://localhost:3306 » pour surveiller un serveur MySQL installé sur la même machine que le serveur Zabbix et Zabbix agent 2.
  - {$MYSQL.PASSWORD} - définissez le mot de passe de l'utilisateur MySQL « zbx_monitor » précédemment créé.
  - {$MYSQL.USER} - définissez le nom de l'utilisateur MySQL « zbx_monitor » précédemment créé.

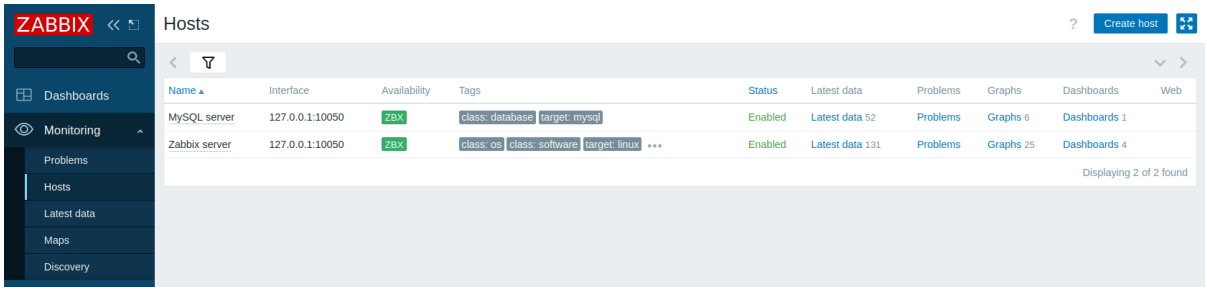


3. Cliquez sur *Add* pour ajouter l'hôte. Cet hôte représentera votre serveur MySQL.

Voir les métriques collectées

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

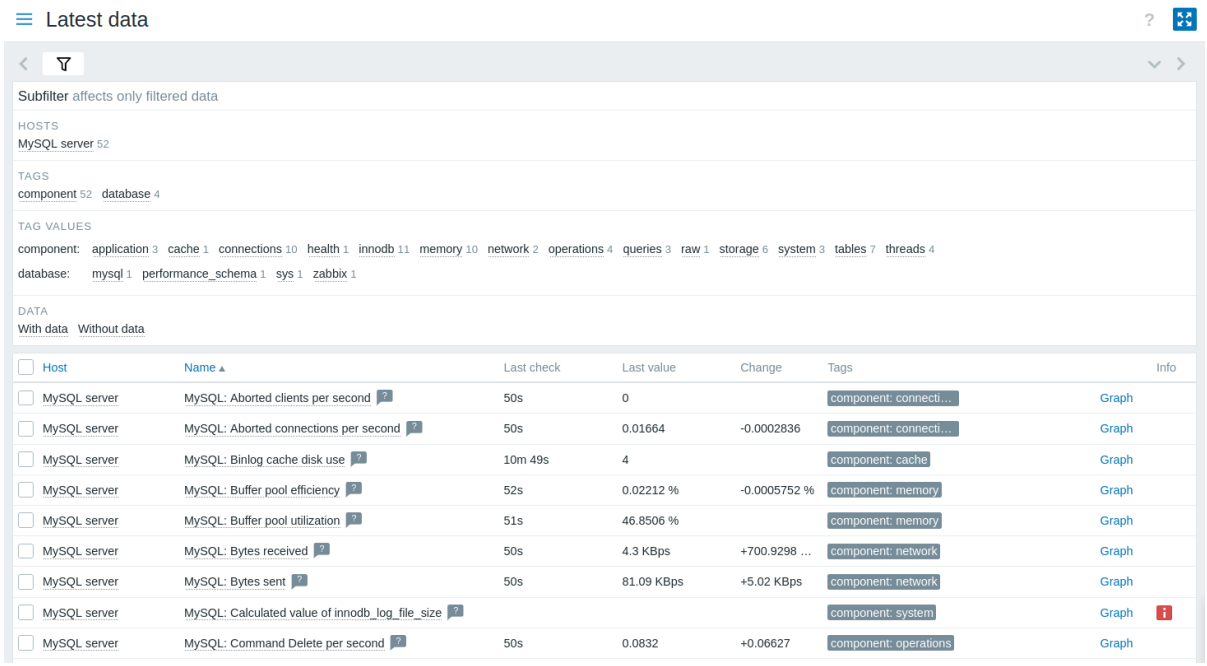
Pour voir les métriques collectées, accédez à la section de menu *Surveillance → Hôtes* et cliquez sur *Tableaux de bord* à côté de l'hôte.



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.



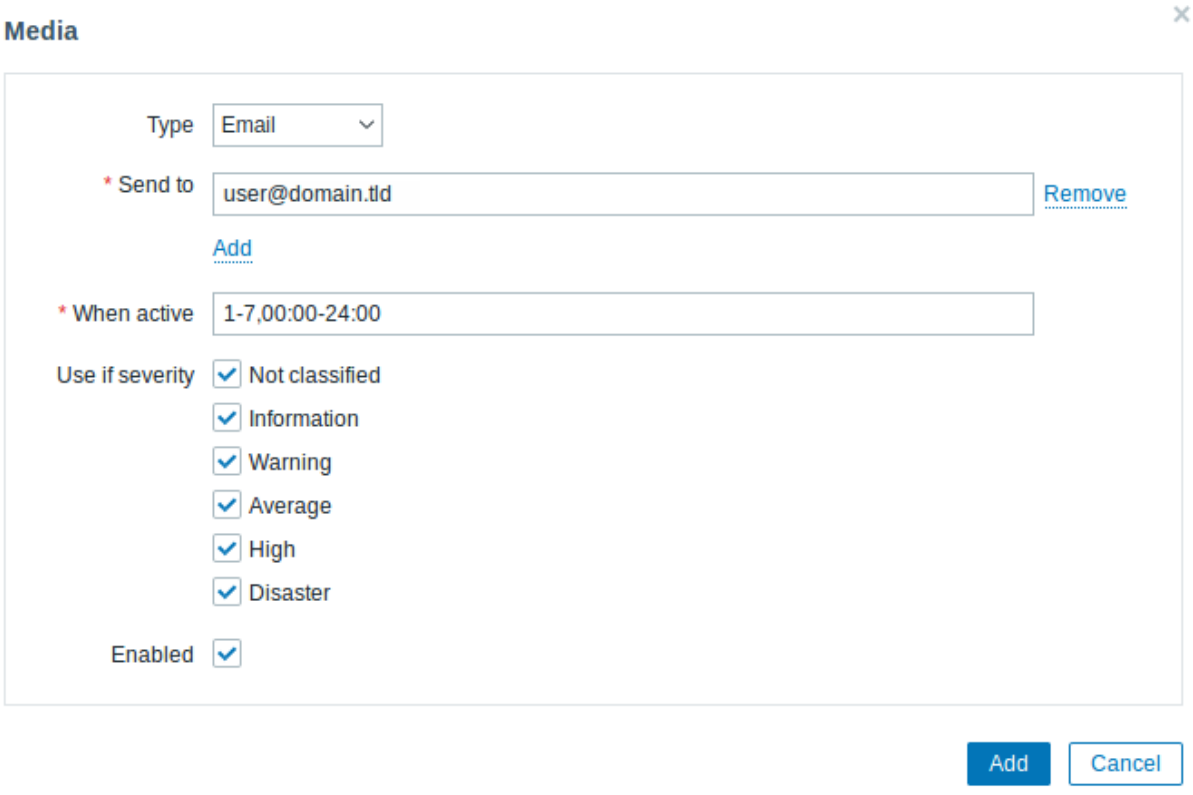
Vous pouvez également, depuis la section de menu *Surveillance → Hôtes*, cliquer sur *Dernières données* pour afficher sous forme de liste toutes les dernières métriques collectées. Notez que l'élément *MySQL: Calculated value of innodb_log_file_size* ne doit pas contenir de données, car la valeur sera calculée à partir des données de la dernière heure.



Configurer les alertes de problème

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

1. Accédez à *User settings* → *Profile*, 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.

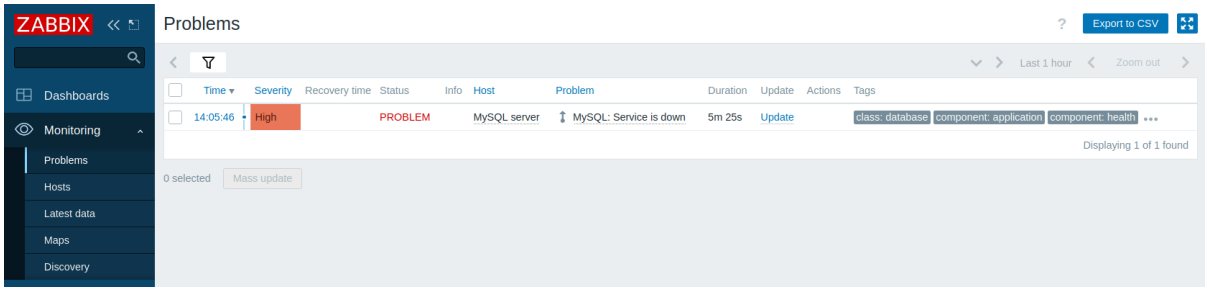
1. Ouvrez la configuration de votre hôte serveur MySQL dans Zabbix.

2. Passez 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 et 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 poursuivre la surveillance du 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 le standard 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 sa fonctionnalité intégrée, 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 la surveillance d'un serveur MySQL avec Zabbix agent 2, il existe quelques différences importantes : vous devez configurer Zabbix agent ou ODBC afin de pouvoir surveiller un serveur MySQL. Les instructions suivantes vous guident à travers ces **différences**.

Surveiller MySQL avec l'agent Zabbix

Pour surveiller un serveur MySQL avec l'agent Zabbix, vous devez télécharger et installer le serveur Zabbix, le frontend Zabbix et l'agent Zabbix conformément aux instructions correspondant à 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 l'agent Zabbix 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 contrôles d'agent personnalisés, ainsi que la fourniture à l'agent Zabbix 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 /usr/local/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 (*zabbix_agentd.conf*). Selon votre système d'exploitation et votre installation de Zabbix, ce répertoire peut se trouver à un emplacement différent de celui indiqué dans ce guide. Pour les emplacements par défaut, consultez le paramètre `Include` dans le fichier de configuration de l'agent Zabbix.

Au lieu de définir tous les paramètres utilisateur nécessaires à la surveillance du serveur MySQL dans le fichier de configuration de l'agent Zabbix, ces paramètres seront définis dans un fichier distinct situé dans le 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 web Zabbix et tester votre configuration

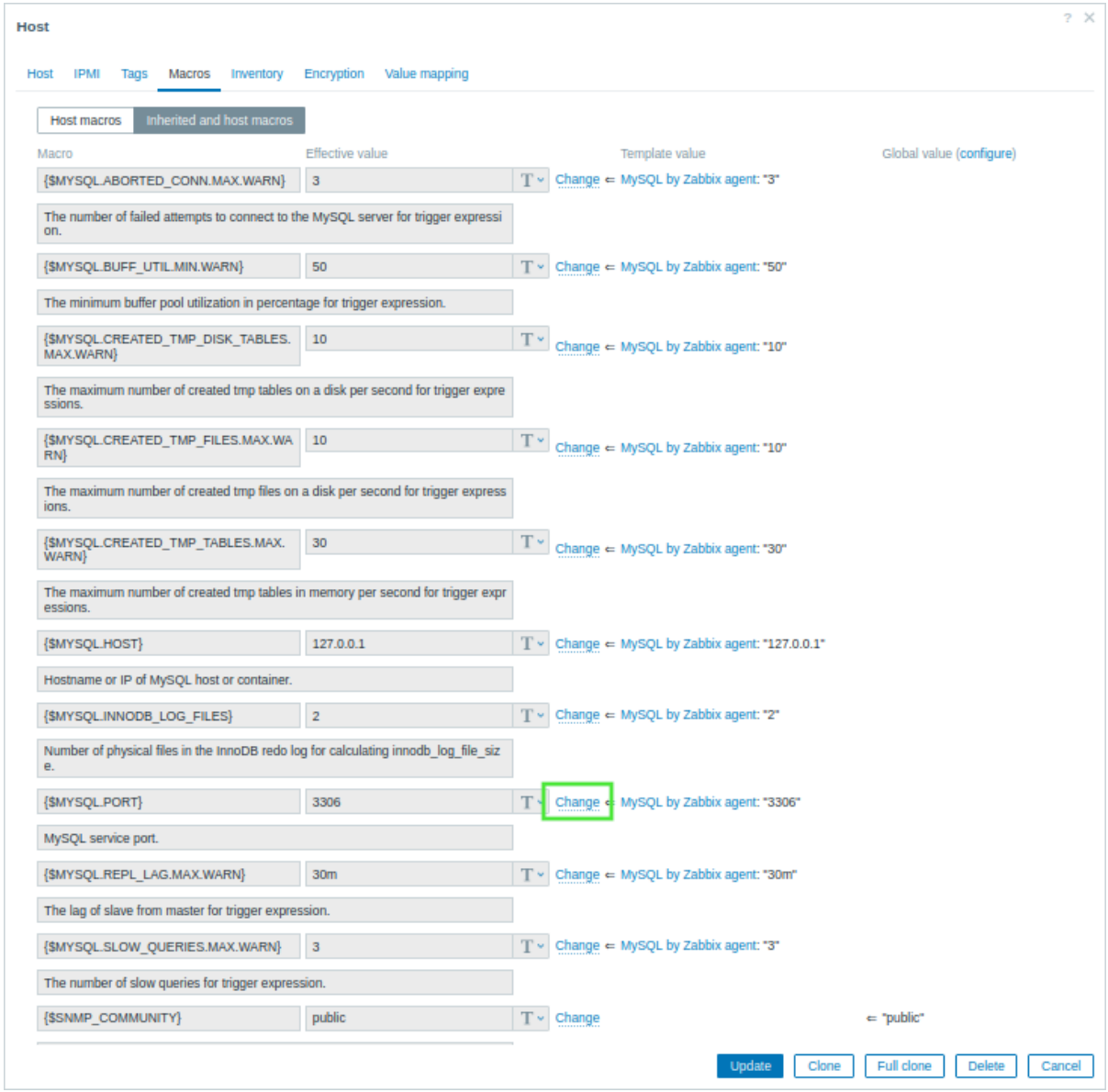
Pour configurer l'interface web Zabbix, suivez les instructions de la section *Configurer l'interface web Zabbix* avec les ajustements suivants :

- Dans le champ *Templates*, 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 web Zabbix configurée, vous pouvez afficher les métriques collectées et configurer des alertes de problème.

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

- Dans la section *Inherited and host macros* de la configuration de l'hôte du serveur MySQL, cliquez sur *Change* à côté de la valeur de 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 server et Zabbix frontend.

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 de l'API ODBC les plus couramment utilisées - 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. Même si vous utilisez 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 afin de surveiller une base de données MySQL, vous avez besoin du nom du pilote, qui se trouve dans le fichier *odbcinst.ini*. Dans l'exemple suivant du fichier *odbcinst.ini*, 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 source de données (DSN) "test", qui sera requis lors de la configuration du frontend Zabbix.

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

Configurer l'interface web Zabbix et tester votre configuration

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

- Dans le champ *Templates*, 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 *Inherited and host macros* de la configuration de l'hôte du serveur MySQL, doit être définie sur le nom DSN du fichier *odbc.ini*.

Une fois l'interface web Zabbix configurée, vous pouvez afficher 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 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).
- Supervision 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*.

5 Surveiller VMware avec Zabbix

Introduction

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

À qui s'adresse ce guide

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

Prérequis

Avant de poursuivre ce guide, vous devez télécharger et installer le serveur Zabbix et le frontend 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 la manière dont la surveillance VMware est effectuée, consultez Surveillance des machines virtuelles.

1. Ouvrez le fichier de configuration du serveur Zabbix.

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

2. Recherchez et définissez le paramètre `StartVMwareCollectors` dans le fichier de configuration du serveur Zabbix 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
```

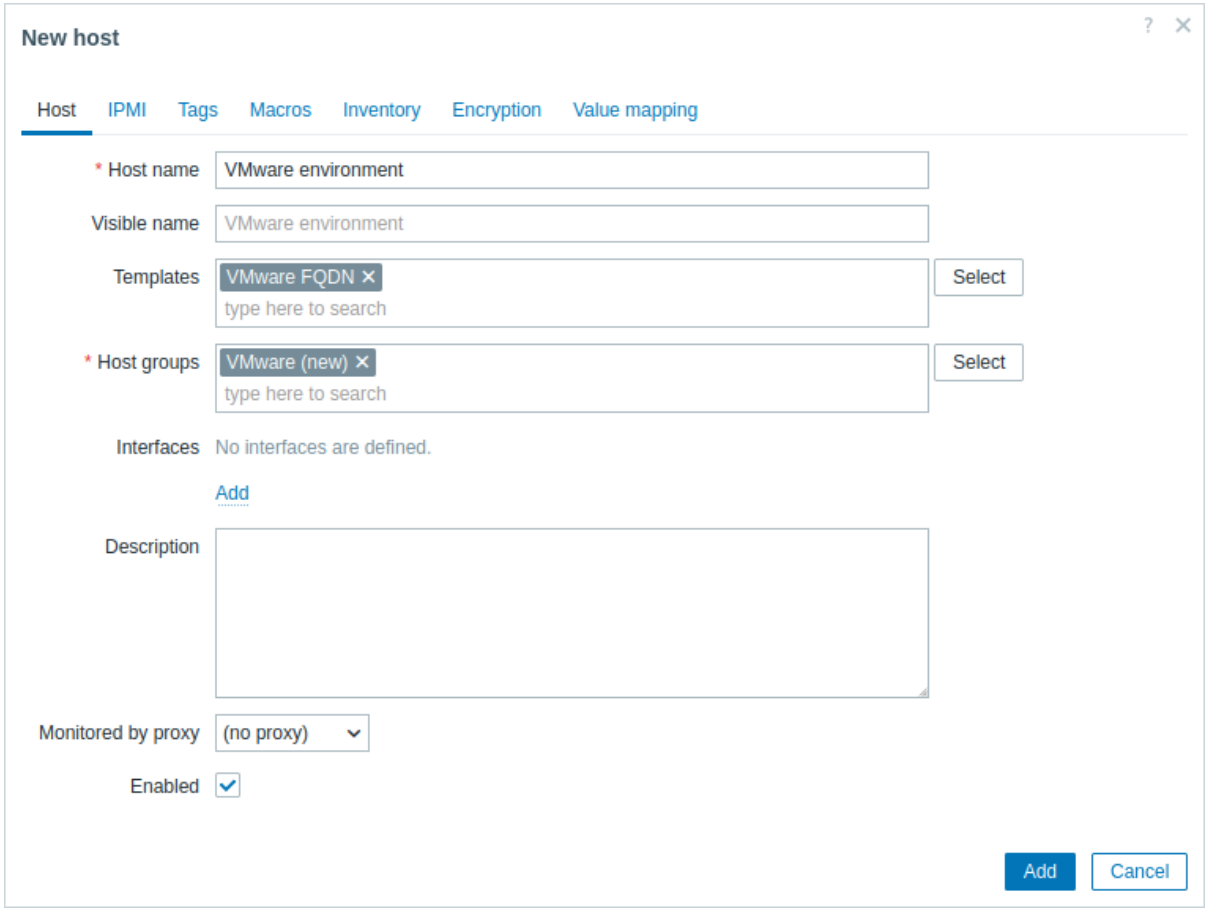
Une fois les processus *vmware collector* démarrés, passez à l'étape suivante.

Configurer l'interface web Zabbix

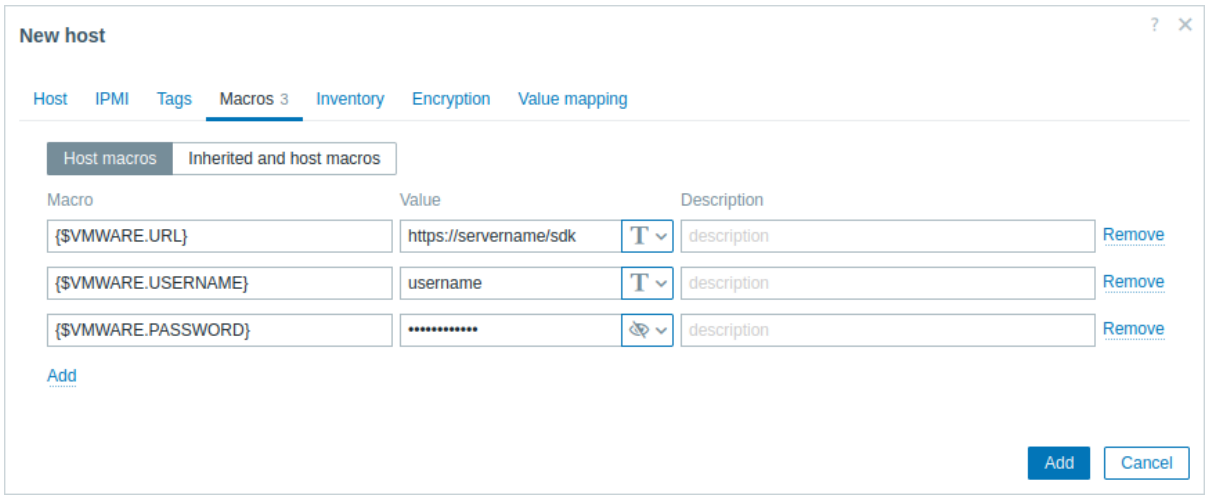
1. Connectez-vous à l'interface web Zabbix.

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 :
  - {$VMWARE.URL} - URL du SDK du service VMware (vCenter ou hyperviseur ESXi) (https://servername/sdk)
  - {$VMWARE.USERNAME} - nom d'utilisateur du service VMware
  - {$VMWARE.PASSWORD} - mot de passe de l'utilisateur {$VMWARE.USERNAME} du service VMware



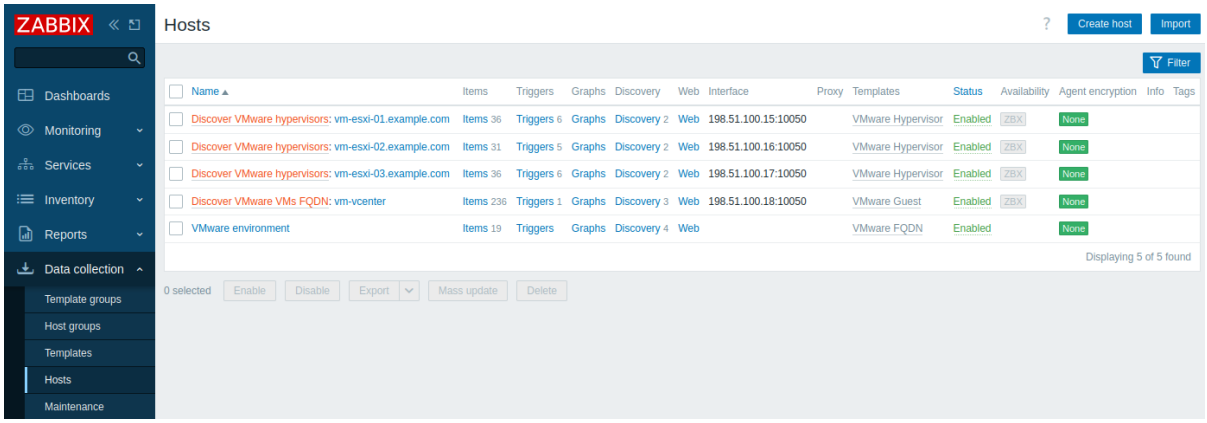
3. Cliquez sur le bouton *Add* pour créer l'hôte. Cet hôte représentera votre environnement VMware.

Afficher les métriques collectées

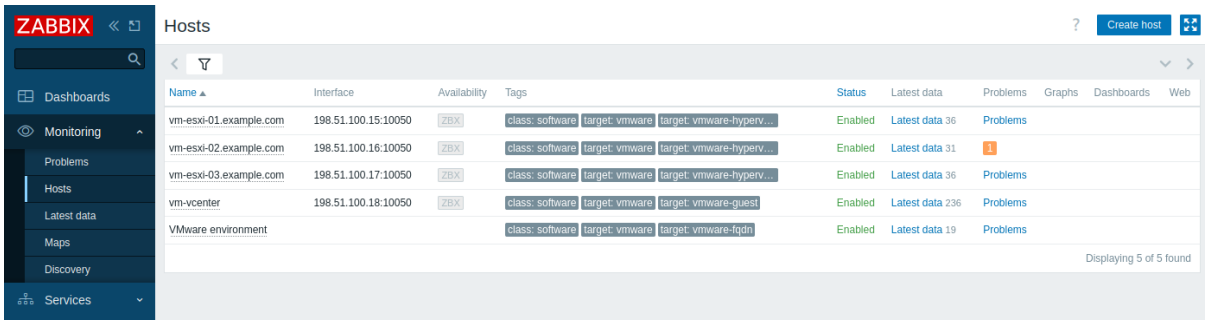
Félicitations ! À ce stade, Zabbix surveille déjà votre environnement VMware.

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.

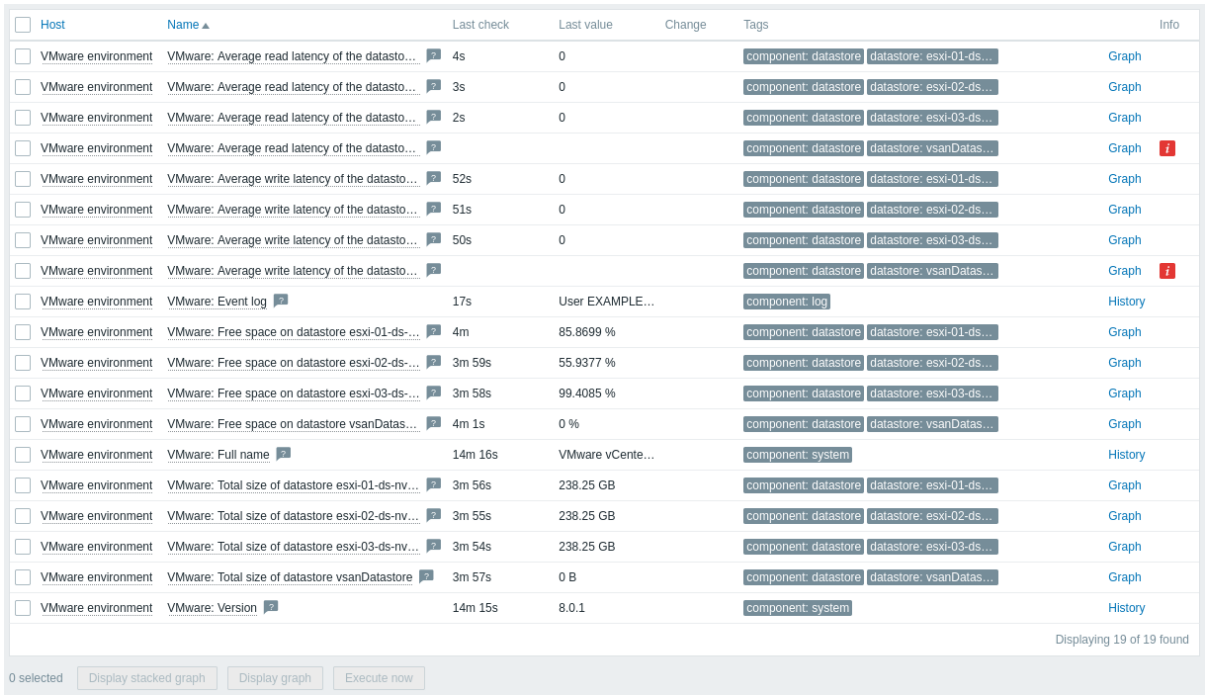
Pour afficher les hôtes créés, accédez à la section de menu *Collecte de données → Hôtes*.



Pour afficher les métriques collectées, accédez à la section de menu *Surveillance → Hôtes* et cliquez sur *Dernières données* à côté de l'hôte « VMware environment » créé 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.

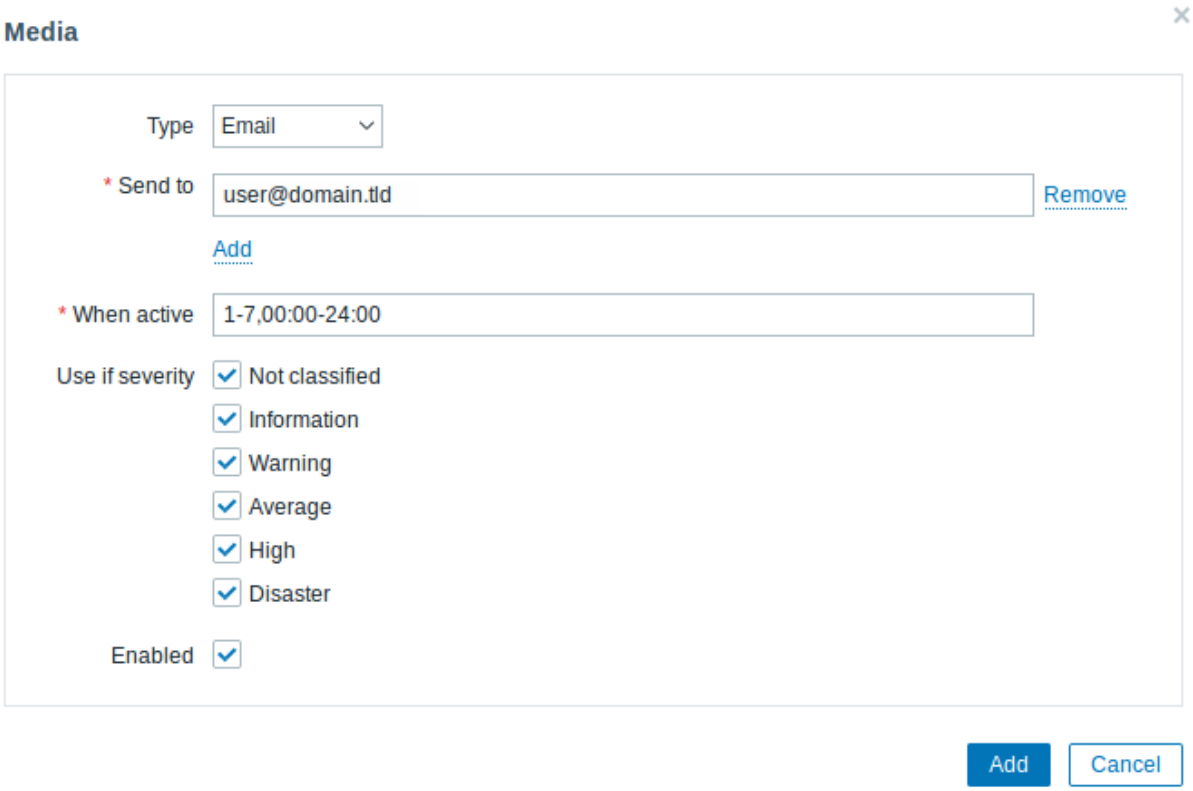


Notez que certains éléments n'ont pas de données et sont dans l'état *Non pris en charge*. Cela s'explique par le fait que Zabbix ne peut pas trouver de compteurs de performance valides sur le datastore spécifique, car ils ne sont pas activés dans l'environnement VMware surveillé.

Configurer les alertes de problème

Zabbix peut vous notifier d'un problème dans votre infrastructure à l'aide de différentes 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 *Médias* 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.

Voir aussi

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

6 Surveiller le trafic réseau avec Zabbix

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

À qui s'adresse ce guide

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

Prérequis

Avant de poursuivre ce guide, vous devez télécharger et installer Zabbix server, Zabbix frontend et Zabbix agent conformément aux instructions pour votre système d'exploitation. Notez que vous devez installer Zabbix agent sur la machine qui nécessite la surveillance du trafic. Il peut s'agir soit du même hôte sur lequel Zabbix server est installé, soit d'un hôte différent.

Ce guide fournit les instructions pour configurer 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 passifs

sera décrite.

Configuration de l'agent Zabbix

1. Ouvrez le fichier de configuration de l'agent sur la machine où l'agent est installé (par défaut, le chemin est */usr/local/etc/zabbix_agentd.conf* :

```
sudo vi /usr/local/etc/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
```

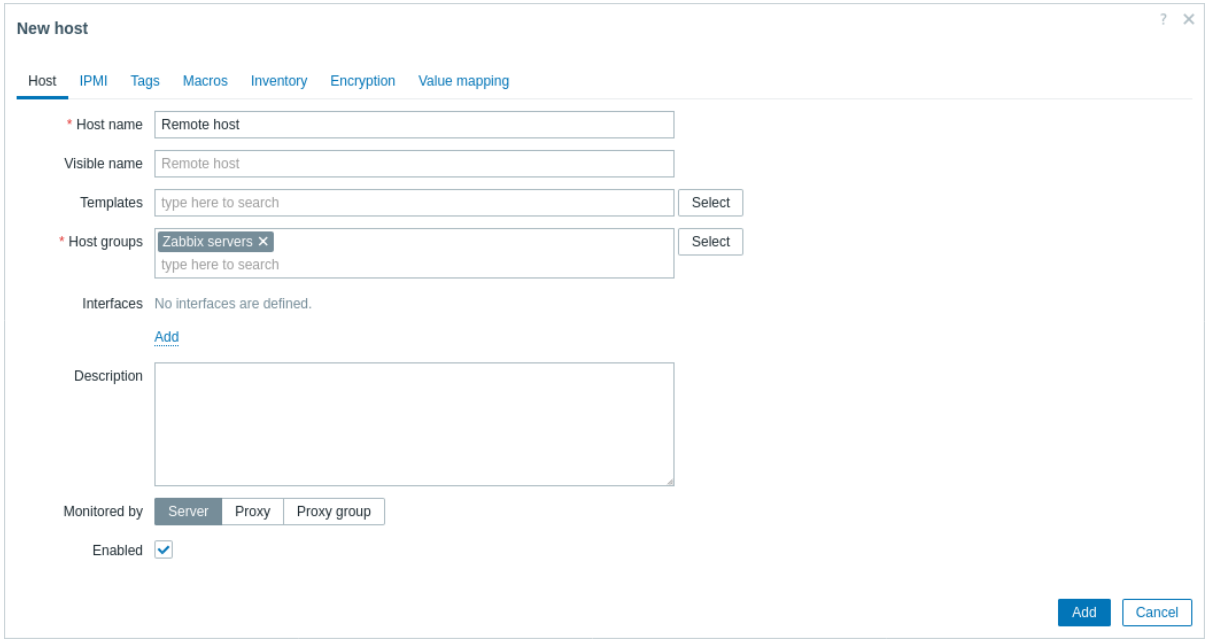
3. Redémarrez l'agent Zabbix :

```
systemctl restart zabbix-agent
```

Interface web Zabbix

1. Connectez-vous à l'interface web Zabbix.

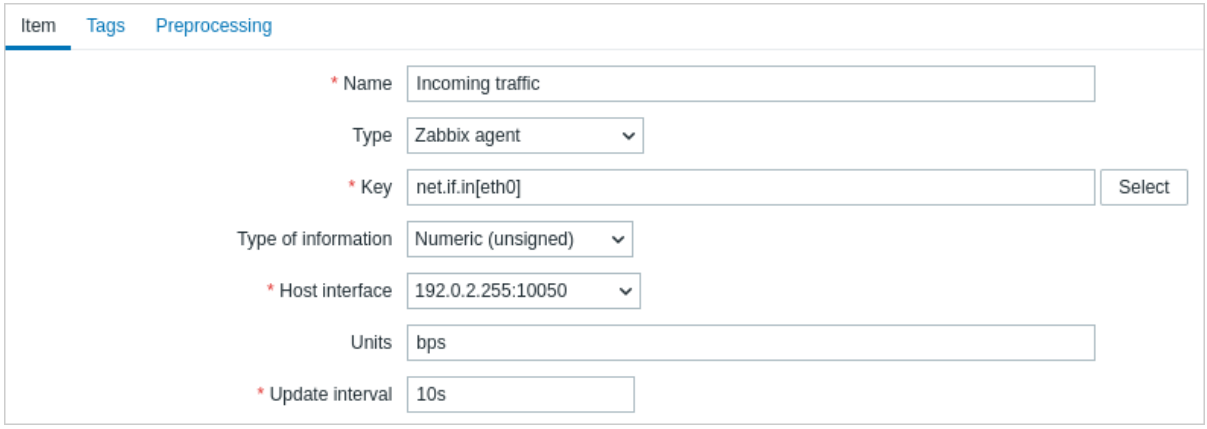
2. Créez un hôte dans l'interface web Zabbix, en indiquant l'adresse IP ou le nom DNS de la machine sur laquelle l'agent est installé.



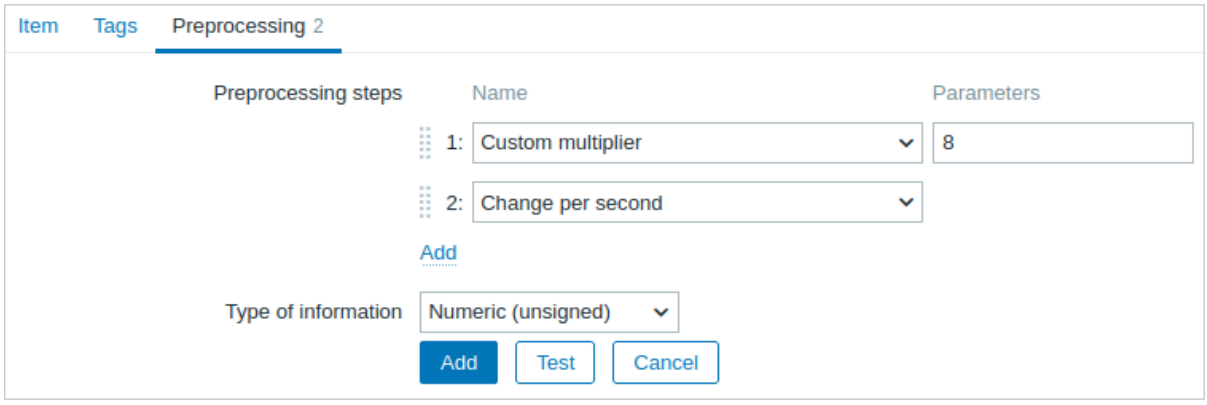
Créer des éléments Suivez les instructions de création d'un élément pour ajouter les éléments de surveillance du trafic, à savoir :

- Trafic entrant
- Trafic sortant
- Trafic total

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

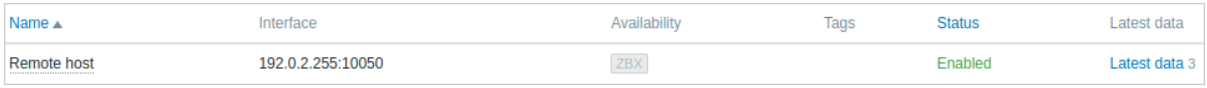


Pour rendre les données collectées adaptées à une utilisation pratique, vous pouvez définir certaines étapes de prétraitement lors de la création des éléments. Dans le cas présent, il peut s'agir d'une multiplication par 8 (pour convertir les octets en bits) et d'un affichage sous forme de variation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

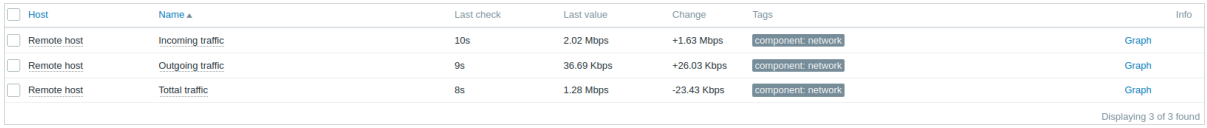


Voir les données collectées Félicitations ! À ce stade, Zabbix surveille déjà votre trafic réseau.

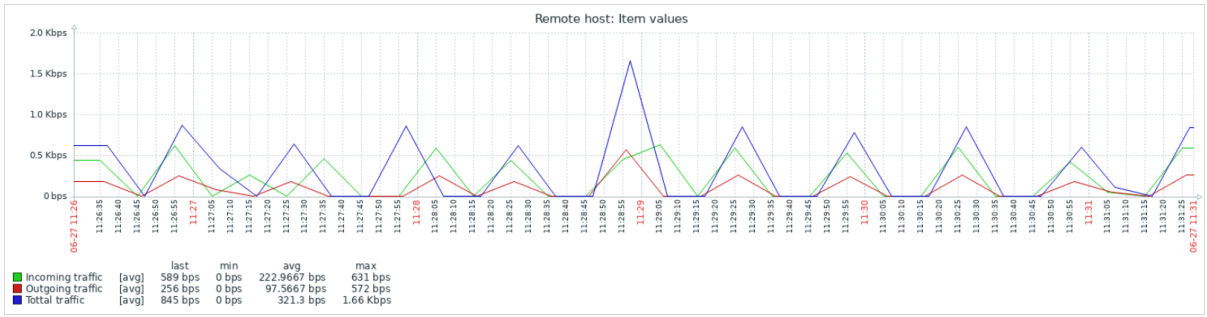
Pour afficher les métriques collectées, ouvrez la section de menu *Monitoring → Hosts* et cliquez sur *Latest data* dans la ligne de l'hôte.



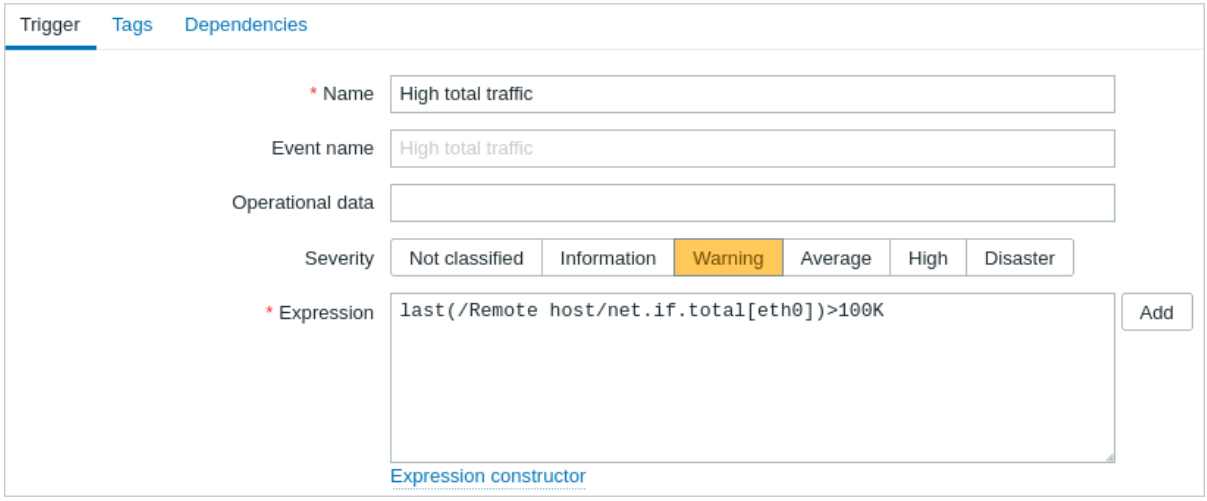
Vous verrez les données de trafic.



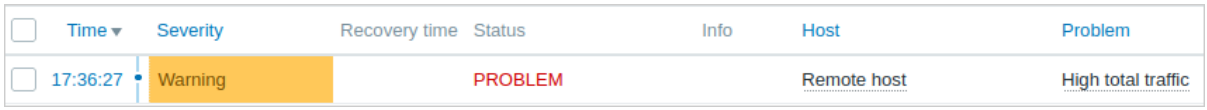
Voir les graphiques Les données collectées peuvent être affichées sous forme de graphiques. Pour les consulter, dans la section *Dernières données*, cliquez sur *Graphique* dans la ligne de l'élément ou sélectionnez les éléments requis et cliquez sur *Afficher le graphique* ci-dessous.



Configurer des 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 indiquant que le trafic total est trop élevé, par exemple :



Faites maintenant en sorte que le trafic dépasse le seuil que vous avez défini dans l'expression du déclencheur, puis accédez à *Surveillance* → *Problèmes* 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 plus populaire, suivez les instructions pour configurer une notification de problème par e-mail. Vous pouvez également choisir d'autres types de média à utiliser pour l'envoi des notifications.

Voir aussi :

- 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).
- Acquittement de problème - comment indiquer que le problème est connu, ajouter des commentaires sur sa résolution, supprimer ou clôturer le problème.
- Surveiller Linux avec l'agent Zabbix - comment démarrer une surveillance de base des éléments les plus importants en liant un modèle préconfiguré.

7 Surveiller le trafic réseau à l'aide de contrôles actifs

Introduction Cette page vous guide à travers les étapes nécessaires pour démarrer une surveillance de base de votre trafic réseau avec Zabbix en utilisant des contrôles actifs.

À qui s'adresse ce guide

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

Prérequis

Avant de poursuivre avec ce guide, vous devez télécharger et installer le serveur Zabbix, le frontend Zabbix et l'agent Zabbix conformément aux instructions pour 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 soit du même hôte sur lequel le serveur Zabbix est installé, soit d'un hôte différent.

Ce guide fournit les instructions pour configurer la surveillance du trafic réseau de l'interface *eth0* sur une machine distincte nommée *Hôte distant*.

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

Configuration de l'agent Zabbix

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

Si vous utilisez l'agent Zabbix :

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

Si vous utilisez l'agent Zabbix 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=192.0.2.0:10051
```

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 le frontend Zabbix. Dans notre exemple, il s'agit de :

```
Hostname=Remote host
```

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 initie une connexion au serveur et s'identifie à l'aide de la valeur Hostname. Le serveur fournit ensuite la configuration de supervision pour 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.

Si vous utilisez l'agent Zabbix :

```
systemctl restart zabbix-agent
```

Si vous utilisez l'agent Zabbix 2 :

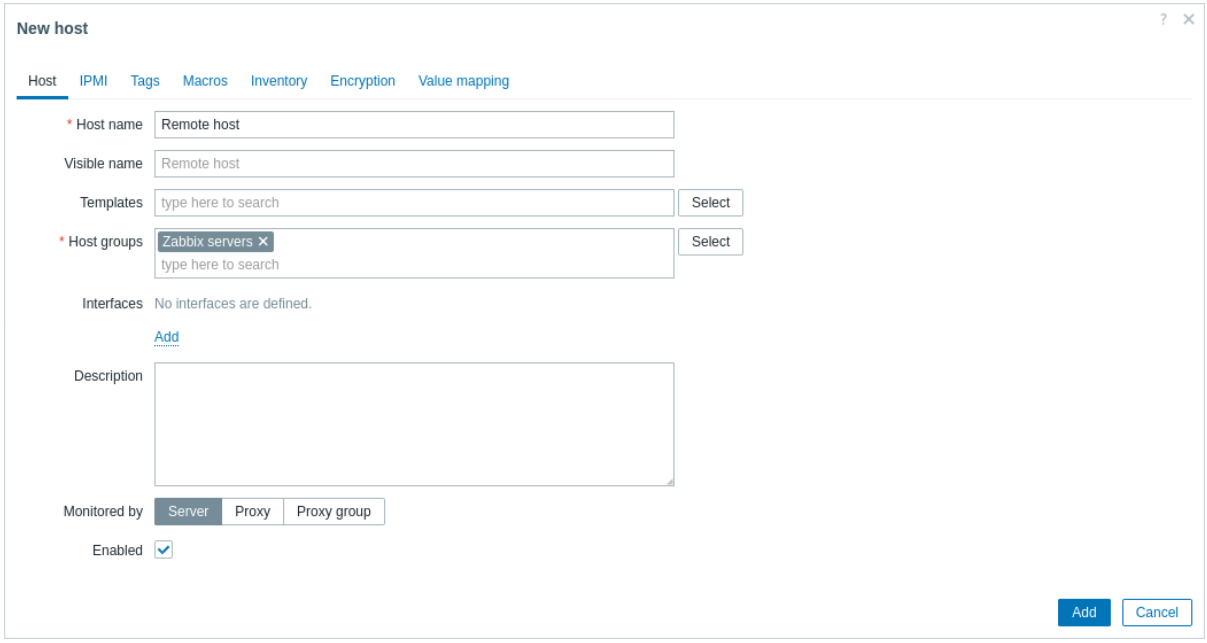
```
systemctl restart zabbix-agent2
```

Interface web Zabbix

1. Connectez-vous à l'interface web 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 contrôles actifs, car l'agent initie la connexion au serveur au lieu d'attendre une connexion provenant du serveur.

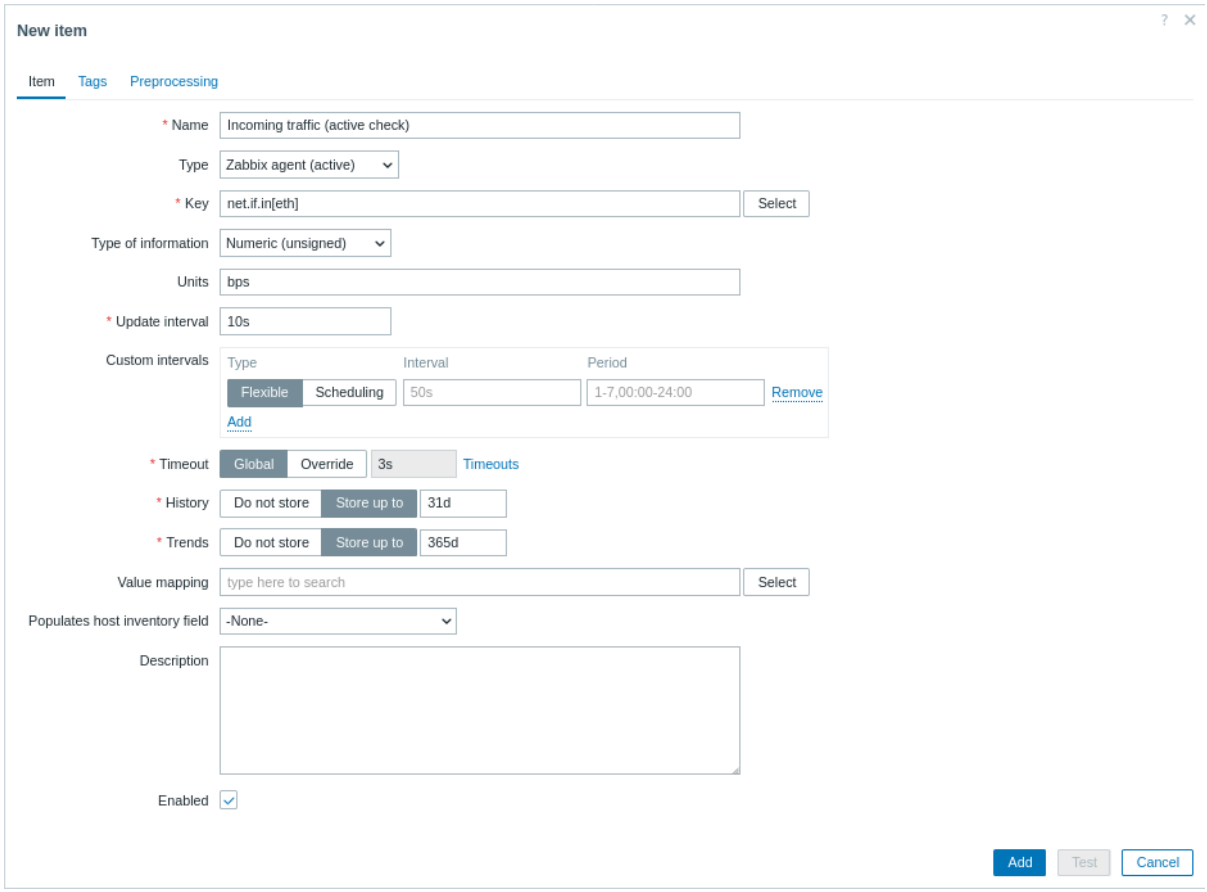


3. Cliquez sur *Add* pour ajouter l'hôte. Cet hôte représentera la machine Linux surveillée.

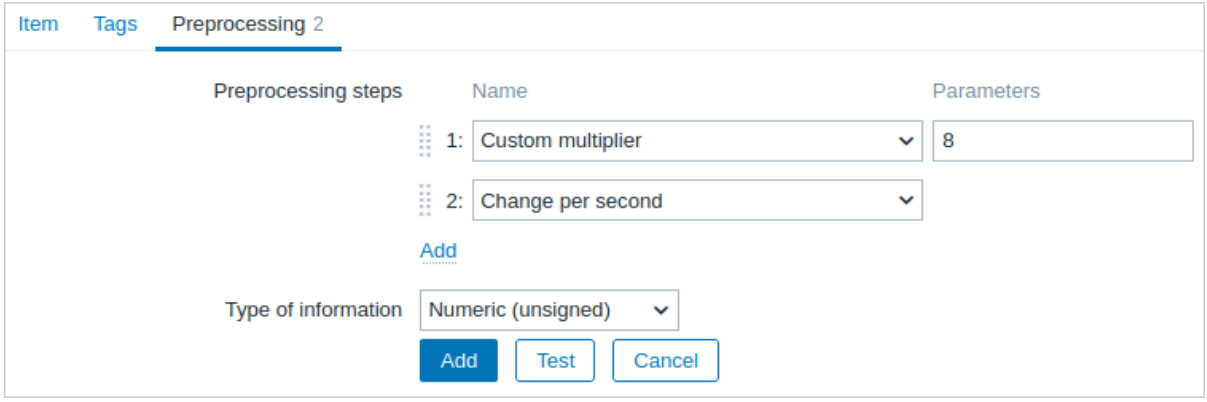
Créer des éléments Suivez les instructions sur la création d'un élément pour ajouter les éléments de surveillance du trafic, à savoir :

- Trafic entrant
- Trafic sortant
- Trafic total

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

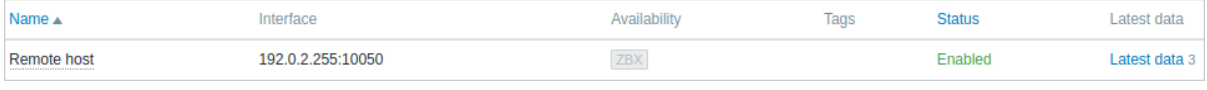


Pour rendre les données collectées adaptées à une utilisation pratique, vous pouvez définir certaines étapes de prétraitement lors de la création des éléments. Dans le cas présent, il peut s'agir d'une multiplication par 8 (pour convertir les octets en bits) et d'un affichage sous forme de variation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

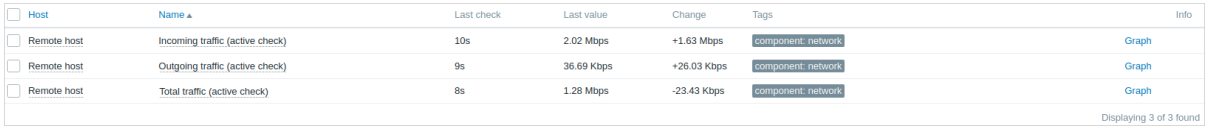


Voir les données collectées Félicitations ! À ce stade, Zabbix surveille déjà votre trafic réseau.

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.



Host	Name ▲	Last check	Last value	Change	Tags	Info
Remote host	Incoming traffic (active check)	10s	2.02 Mbps	+1.63 Mbps	component: network	Graph
Remote host	Outgoing traffic (active check)	9s	36.69 Kbps	+26.03 Kbps	component: network	Graph
Remote host	Total traffic (active check)	8s	1.28 Mbps	-23.43 Kbps	component: network	Graph

Displaying 3 of 3 found

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 des situations problématiques.
- 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).
- Acquittement de problème - comment indiquer que le problème est connu, ajouter des commentaires sur sa résolution, supprimer ou clôturer le problème.
- Surveiller Linux avec l'agent Zabbix - comment démarrer une surveillance de base des éléments les plus importants en liant un modèle préconfiguré.
- Installation à partir de paquets - comment installer les composants Zabbix à l'aide des paquets RPM et DEB officiels pour diverses distributions Linux, en garantissant l'accès aux dernières fonctionnalités et corrections de bogues.

8 Surveiller des sites web avec des éléments Browser

Introduction

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

À qui s'adresse ce guide

Ce guide est conçu pour les nouveaux utilisateurs de Zabbix et contient l'ensemble minimal d'étapes nécessaires pour activer la surveillance de base des sites web avec des éléments Browser. Si vous recherchez des options de personnalisation poussées ou avez besoin d'une configuration plus avancée, 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 le frontend 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 et interagit avec un navigateur, 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 Install Docker Engine on Ubuntu.

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

- **docker run --name browser** - exécute un nouveau conteneur Docker nommé « browser » ;
- **-p 4444:4444** - mappe le port 4444 de votre machine hôte vers le port 4444 du conteneur (il s'agit du 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 (il s'agit du port utilisé par le serveur Virtual Network Computing (VNC), vous permettant d'afficher 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 afin d'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 with Chrome.

```
docker run --name browser \
-p 4444:4444 \
-p 7900:7900 \
```

```
--shm-size="2g" \
-d selenium/standalone-chrome:latest
```

2. Assurez-vous 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, reportez-vous à la documentation Docker.

Configurer le serveur Zabbix

Les éléments de navigateur sont exécutés et traités par les processus Zabbix *browser poller*, 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
###URL HTTP[S] de l'interface WebDriver. Par exemple, http://localhost:4444 utilisé avec le serveur autono
###
### Mandatory: no
### Default:
### WebDriverURL=

WebDriverURL=192.0.2.1:4444
```

3. Redémarrez le serveur Zabbix.

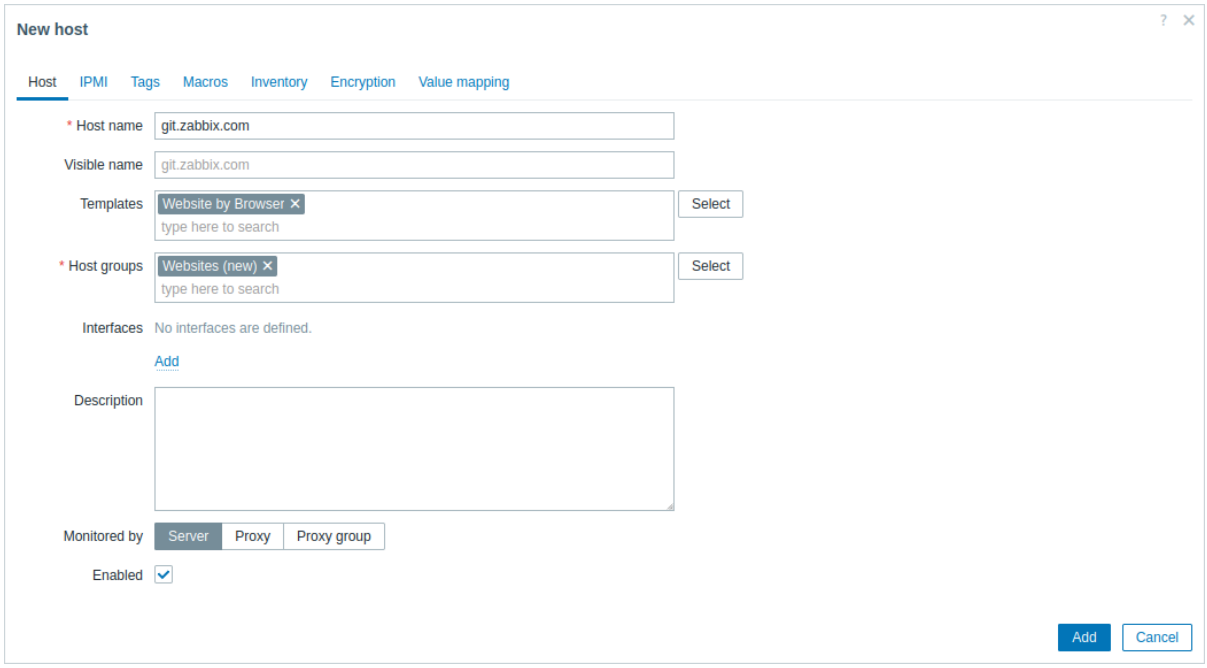
```
systemctl restart zabbix-server
```

Configurer l'interface web Zabbix

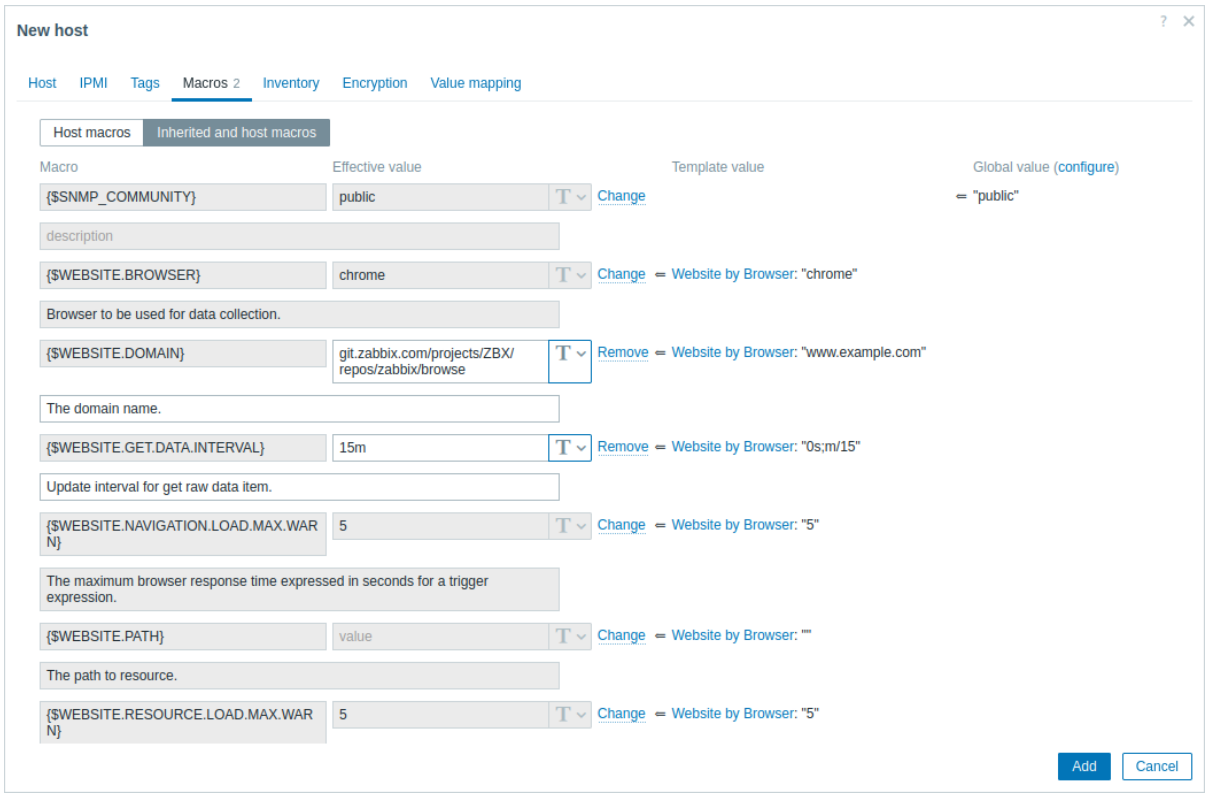
1. Connectez-vous à l'interface web Zabbix.

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*, saisissez ou sélectionnez le modèle « Website by Browser ». Pour plus d'informations sur ce modèle, consultez Website by Browser.
   - Dans le champ *Host groups*, saisissez 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)

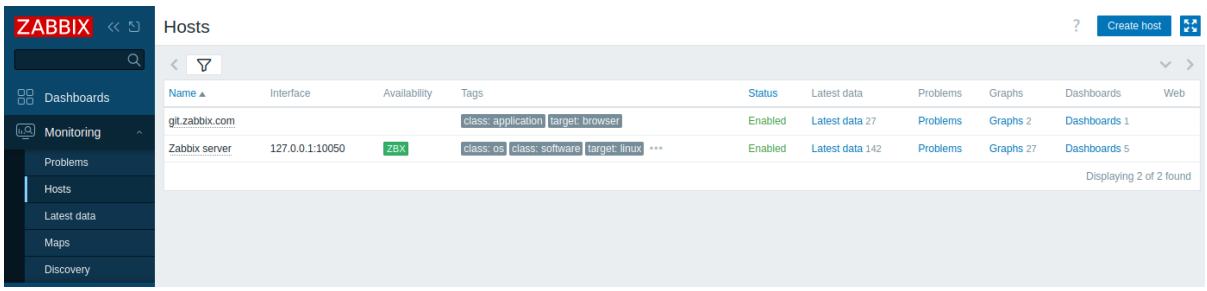


3. Cliquez sur le bouton *Add* pour créer l'hôte. Cet hôte représentera le site web que vous souhaitez surveiller.

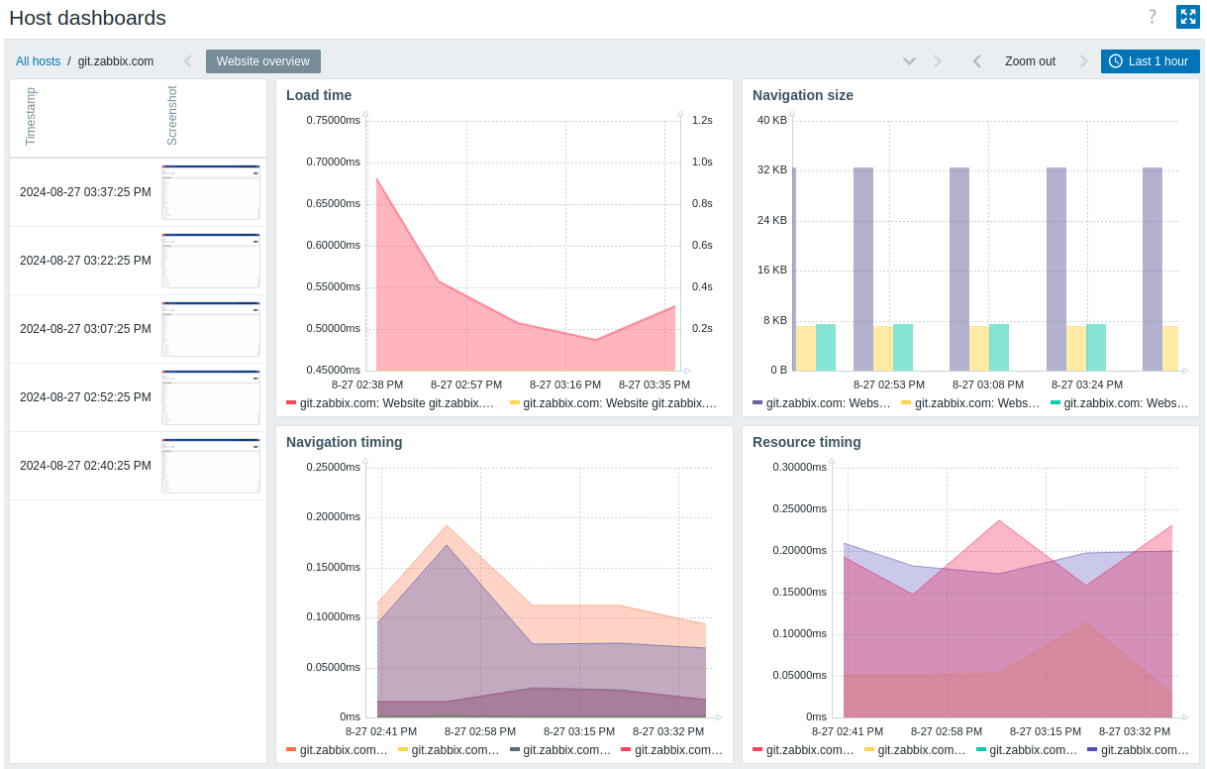
Voir les métriques collectées

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

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



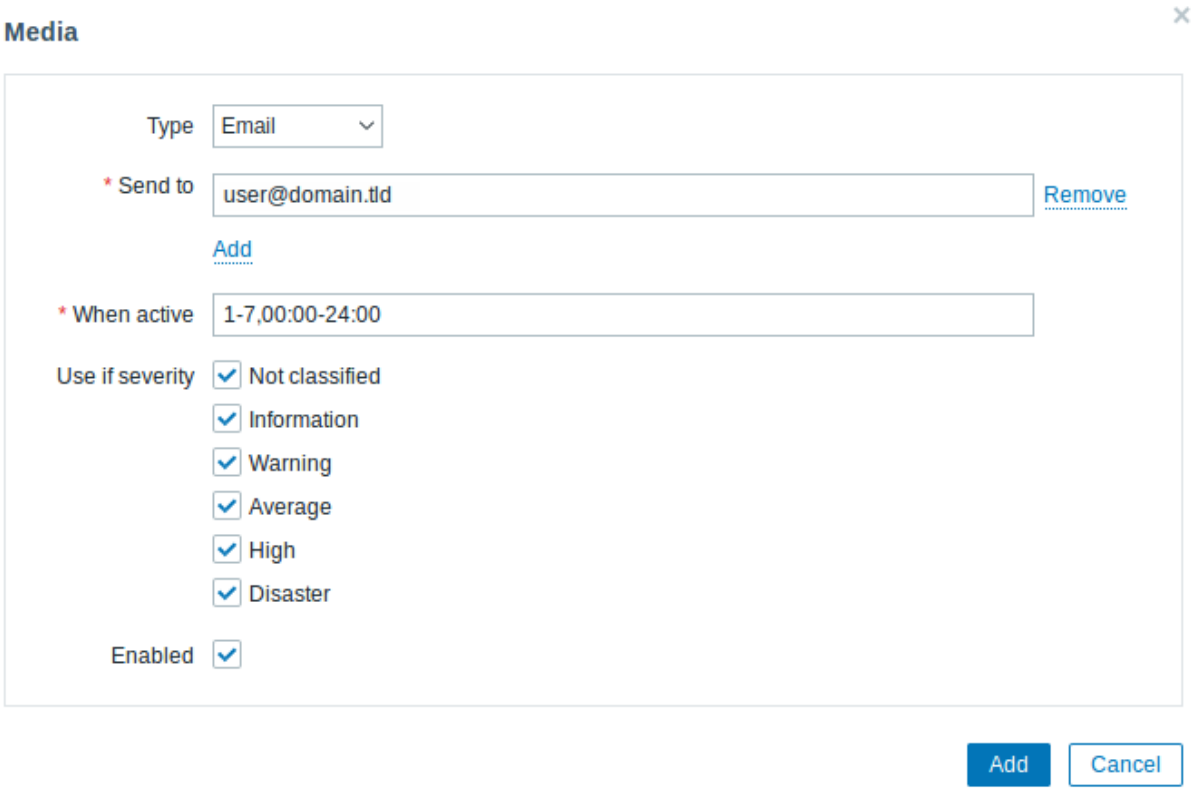
Cette action vous amènera au tableau de bord de l'hôte (configuré au niveau du modèle) avec les métriques les plus importantes collectées à partir du site web.



Configurer les alertes de problème

Zabbix peut vous informer d'un problème dans votre infrastructure à l'aide de différentes 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 *Médias* 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. Passez à l'onglet *Macros* et sélectionnez *Inherited and host macros*.

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

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

5. Dans 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 *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 afin de résoudre le problème et de continuer à surveiller le site web.

Voir aussi

- Création d'un élément - comment commencer à surveiller des métriques supplémentaires.
- 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).
- Éléments Browser - comment configurer les éléments Browser.
- Modèle Website by Browser - informations supplémentaires sur le modèle *Website by Browser*.

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

Introduction

Ce guide fournit une vue d'ensemble complète de la manière de configurer et de surveiller les certificats SSL/TLS à l'aide de la clé `web.certificate.get` dans l'agent Zabbix 2. Il est conçu pour simplifier la surveillance des certificats pour un ou plusieurs sites web, en veillant à ce que les administrateurs puissent identifier rapidement les problèmes potentiels, tels que des certificats expirés ou non valides.

À qui s'adresse ce guide

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

Prérequis

Avant de poursuivre ce guide, vous devez télécharger et installer le serveur Zabbix, le frontend Zabbix et l'agent Zabbix 2 conformément aux instructions correspondant à 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 Zabbix agent (chemin par défaut : `/etc/zabbix/zabbix_agent2.conf`) :

```
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 l'installation et la configuration de 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 séparée n'est donc requise.

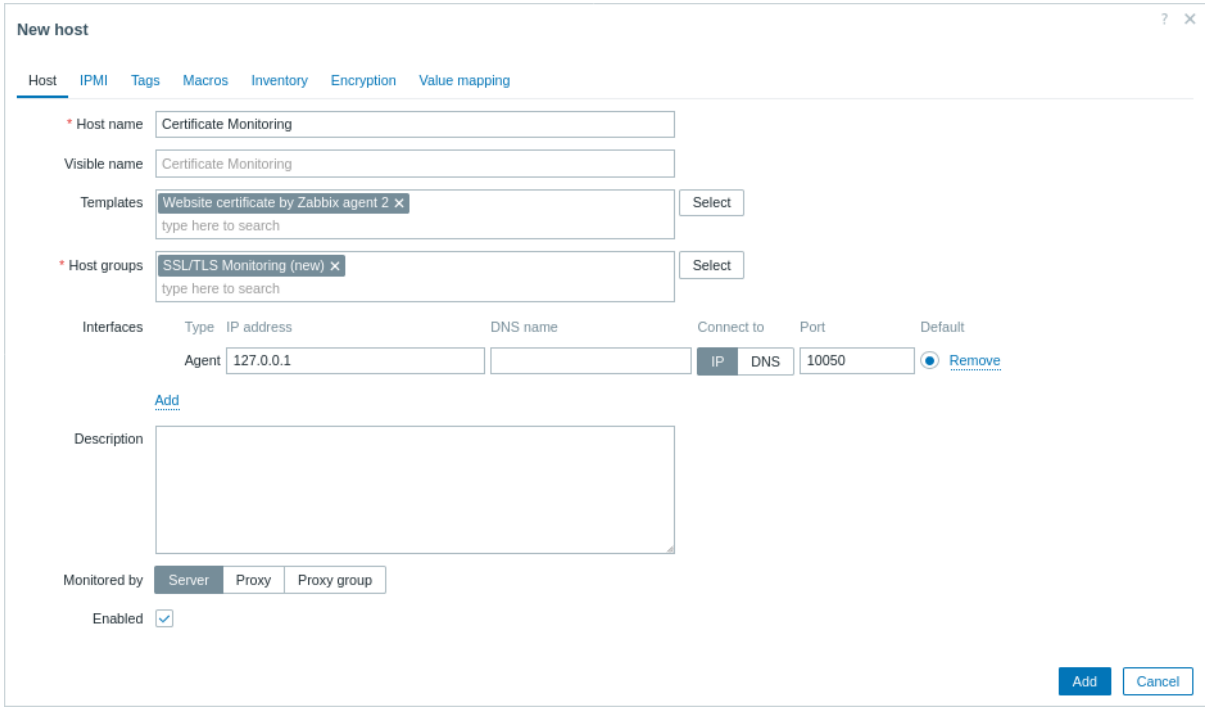
Configurer l'interface web Zabbix

1. Connectez-vous à l'interface web Zabbix.

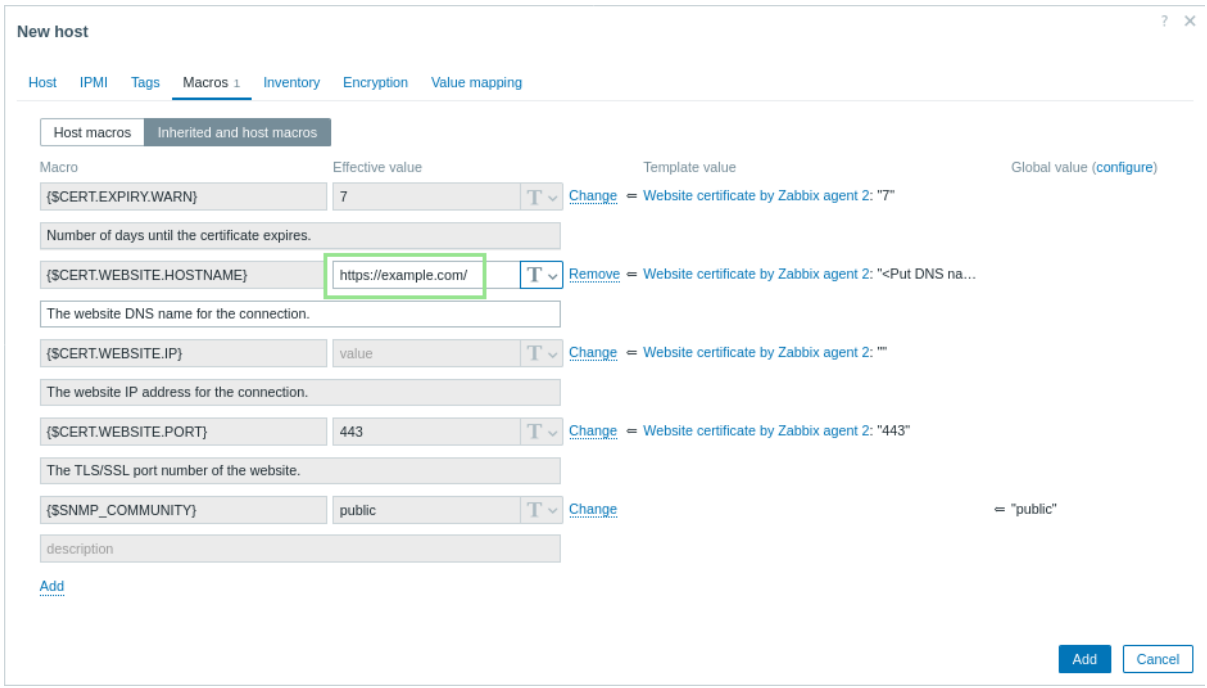
2. Accédez à *Monitoring* > *Hosts*.

3. Cliquez sur un hôte existant sur lequel vous souhaitez surveiller les certificats de sites 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 indiquez 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 `{$CERT.WEBSITE.HOSTNAME}`. Facultativement, vous pouvez également fournir une liste de ports séparés par des virgules dans la macro `{$CERT.WEBSITE.PORT}`, où chaque port correspond au nom d'hôte respectif dans le même 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

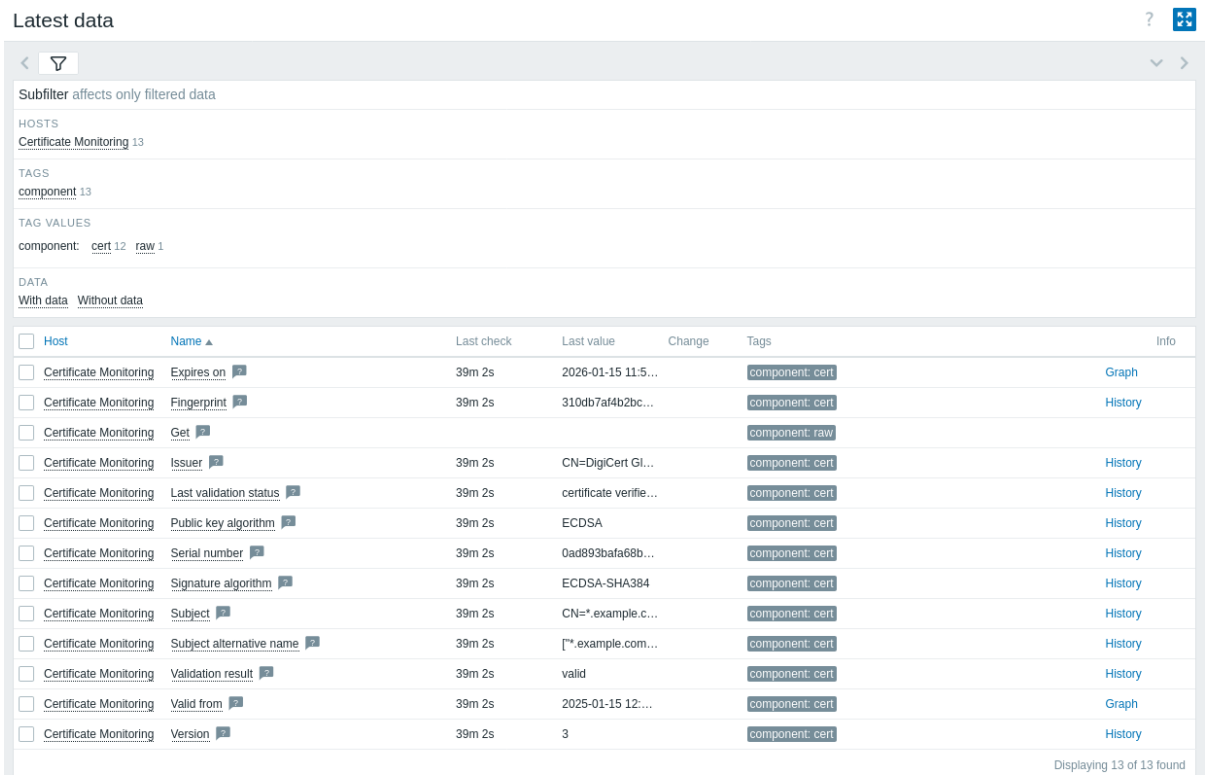
Pour chaque site web spécifié dans la macro `{$CERT.WEBSITE.HOSTNAME}`, Zabbix créera un ensemble correspondant d'éléments et de déclencheurs. Cela permet une surveillance et des alertes individuelles pour le certificat SSL de chaque site web.

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:…

Voir les métriques collectées

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

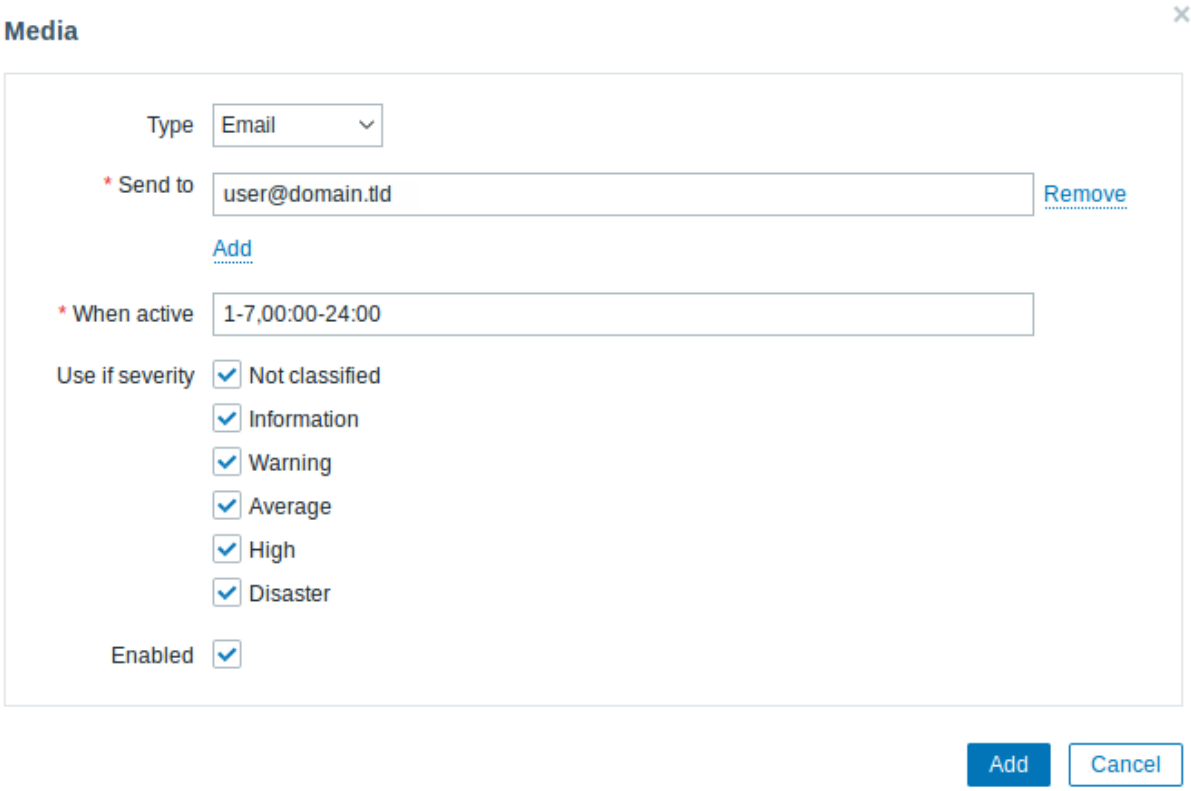
Pour voir les métriques collectées, accédez à la section de menu *Surveillance > Hôtes* et cliquez sur *Dernières données* à côté de l'hôte pour afficher toutes les dernières métriques collectées dans une liste, comme la date d'expiration, l'émetteur et le sujet.



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 *Médias* 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.

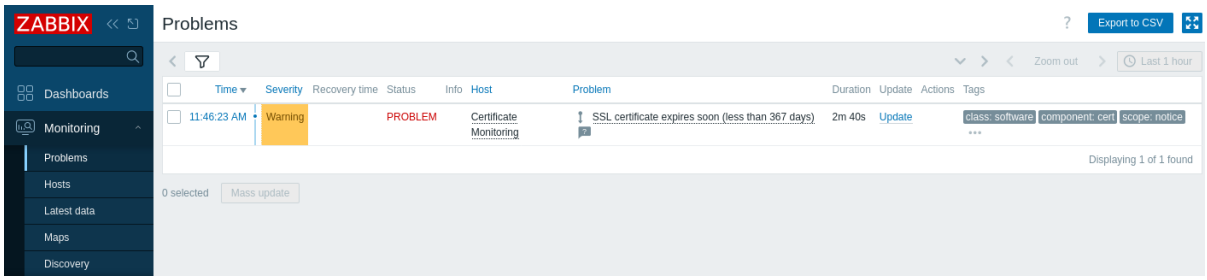
1. Ouvrez la configuration de votre hôte ”Certificate Monitoring” dans Zabbix.

2. Passez 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 (supérieur à 365 jours devrait être suffisant) 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. Remettez 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 une surveillance de base des sites web avec des éléments Browser.

10 Surveiller un commutateur réseau ou un routeur avec Zabbix

Introduction Ce guide vous accompagne à travers les étapes nécessaires pour démarrer 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 équipement 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 équipements 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 avec ce guide, assurez-vous de disposer des éléments suivants :

- Zabbix server et Zabbix frontend installés : effectuez l’installation conformément aux instructions correspondant à votre système d’exploitation (voir Installation à partir de paquets et Installation de l’interface web).
- Zabbix agent installé si vous surveillez des métriques réseau locales.
- Équipement 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 MIB appropriée, vous risquez de ne voir que des valeurs numériques, ce qui complique la configuration des éléments et le dépannage.

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/mibs/.
- 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 chemin correct.

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

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

Pour des instructions détaillées, reportez-vous à 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 toutes 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 la surveillance d'un équipement réseau Cisco Catalyst 3750V2-24FS.

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

Configurer l'équipement réseau (exemple de routeur Cisco) Pour la supervision via SNMP, vous devez configurer votre équipement 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.

Pour des raisons de sécurité, il est recommandé de limiter l'accès SNMP uniquement aux appareils nécessaires. 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 par rapport à 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 les tutoriels externes de configuration SNMP Cisco. Ce guide fournit toutefois les étapes de base pour activer la supervision SNMP.

Configurer l'interface web de Zabbix

Créer un hôte dans l'interface web Zabbix 1. Connectez-vous à l'interface web Zabbix.

2. Ajoutez un nouvel hôte.

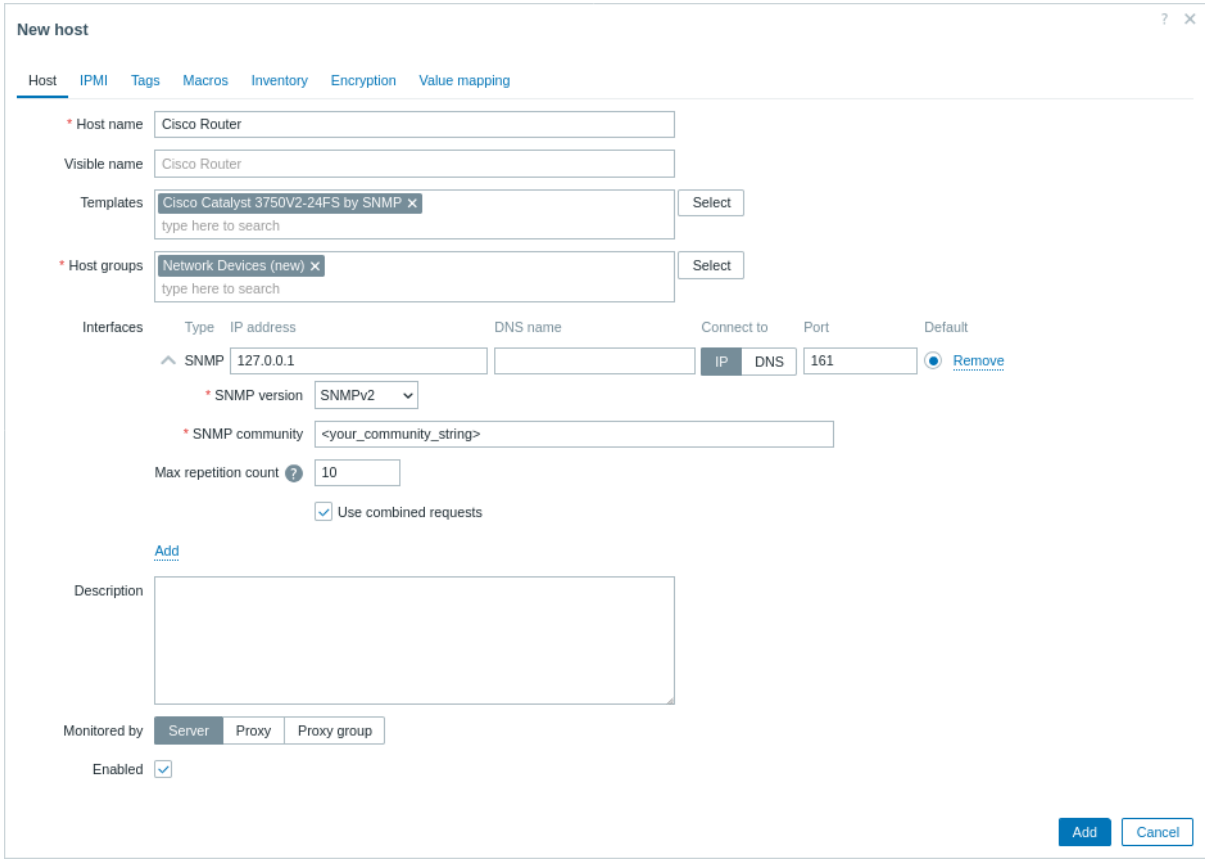
Accédez à *Collecte de données* > *Hôtes* et cliquez sur *Créer un hôte*.

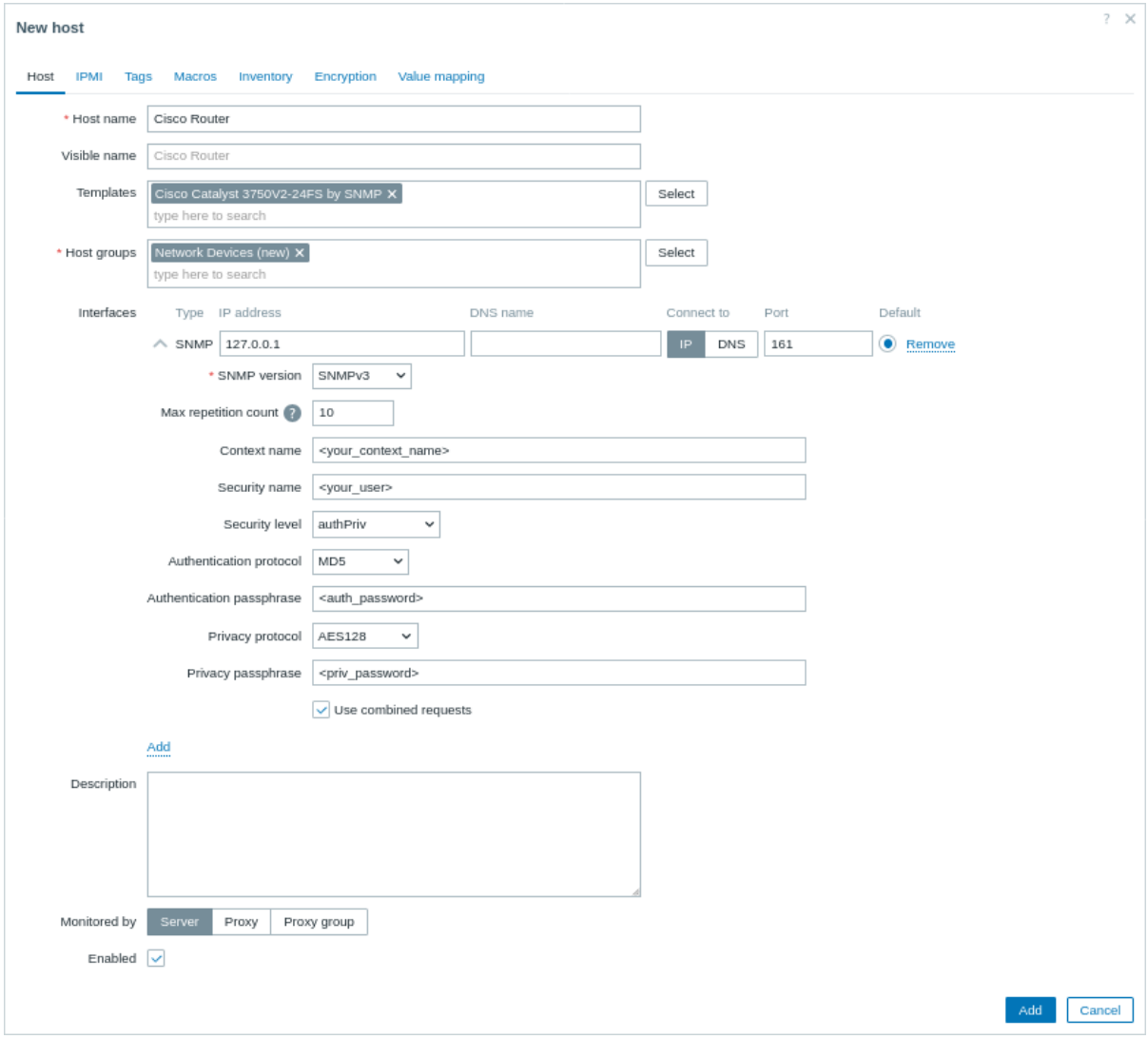
- *Nom de l'hôte* : saisissez un nom pour votre appareil (par exemple, « Cisco Router »).
- Groupes d'hôtes : sélectionnez un groupe existant ou créez-en un nouveau, par exemple « Network Devices ».
- Interfaces :
  - Cliquez sur *Ajouter* sous Interfaces.
  - Choisissez *SNMP* comme type d'interface.
  - 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.) 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 (tel que Cisco IOS SNMP ou Cisco Catalyst 3750<modèle d'appareil> SNMP).

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

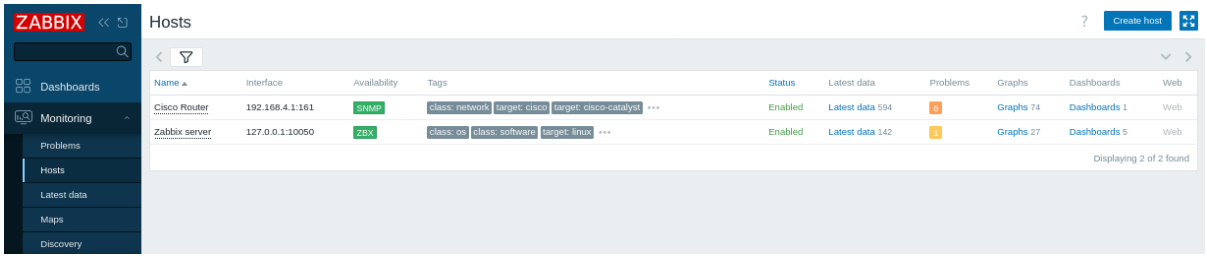




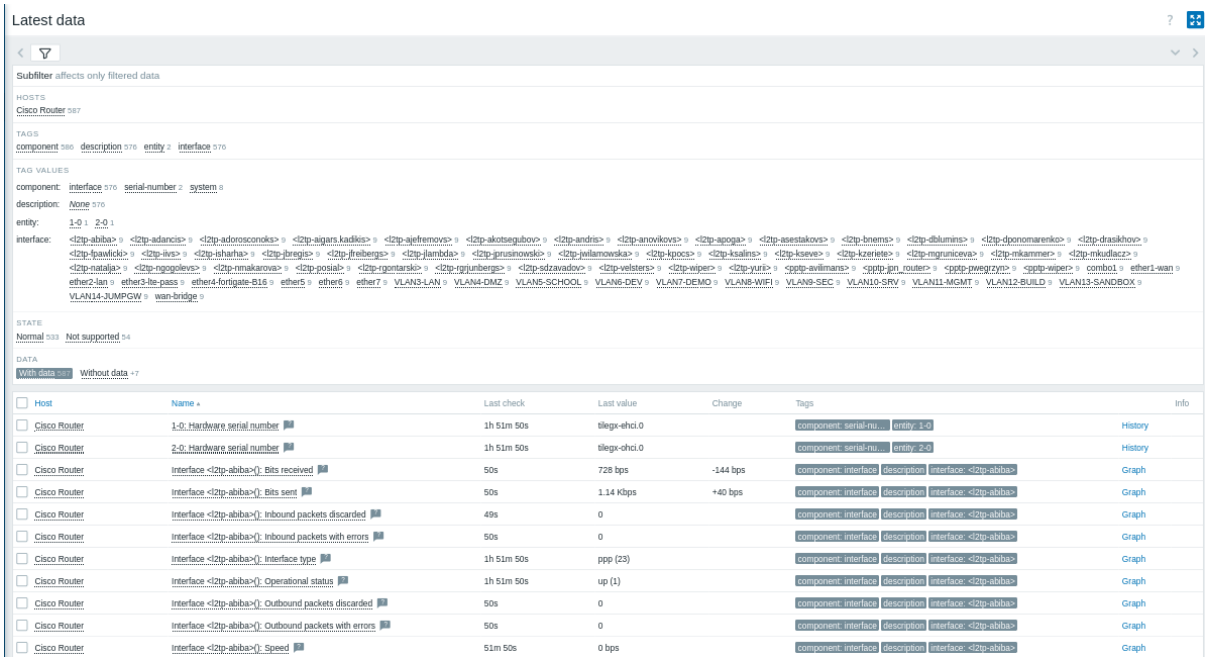
Afficher les métriques collectées Félicitations ! Zabbix est maintenant configuré pour surveiller votre équipement réseau.

Dernières données :

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



- Sélectionnez votre hôte « Cisco Router » (ou les hôtes découverts) pour afficher des métriques telles que le temps de fonctionnement du matériel et du 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 pour voir des métriques détaillées.

Comme étape suivante, vous pouvez :

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

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. Identifier l'OID SNMP :

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

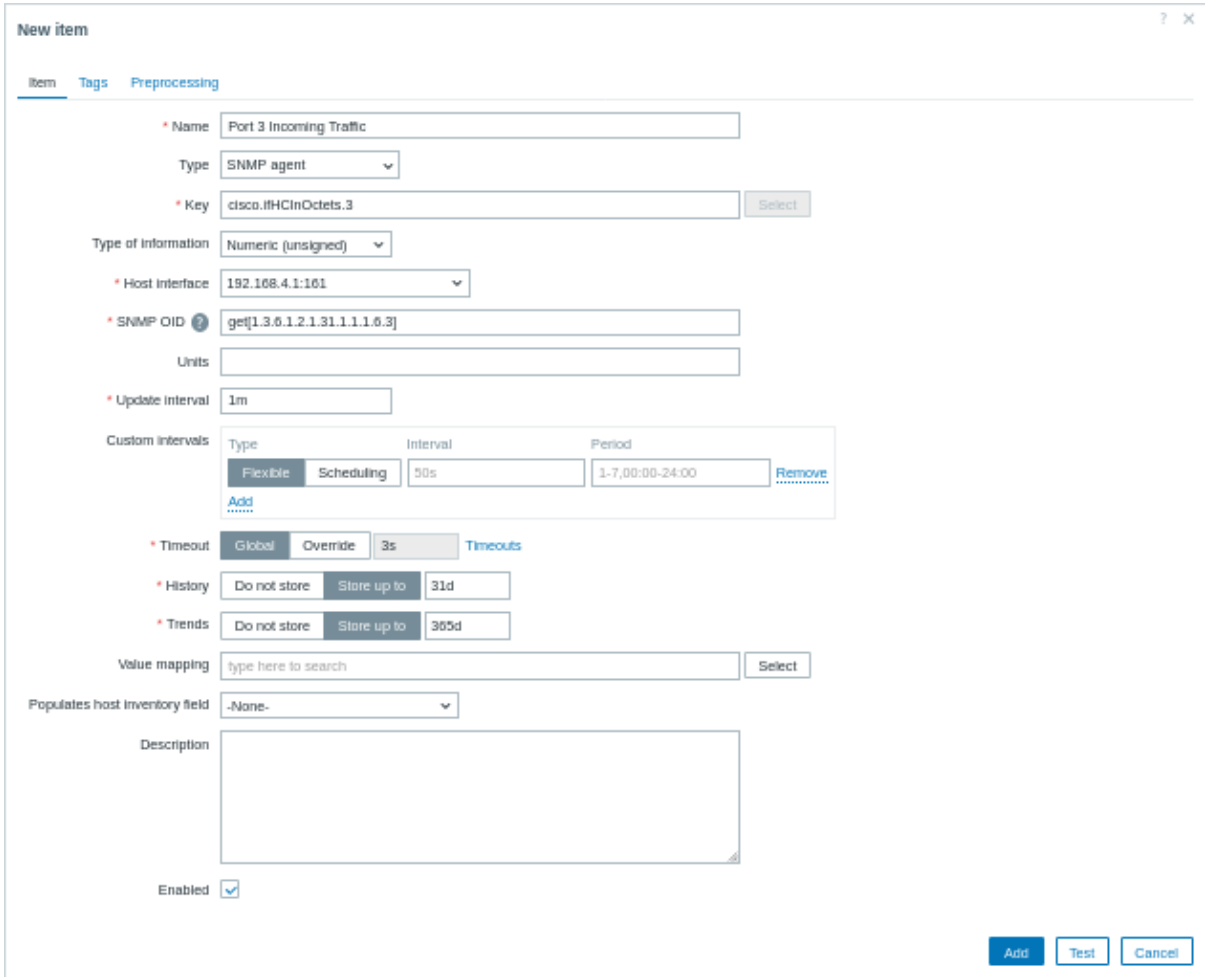
```
snmpwalk -v 2c -c <your_community_string> <device_IP> .
```

Recherchez l'OID de 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 :

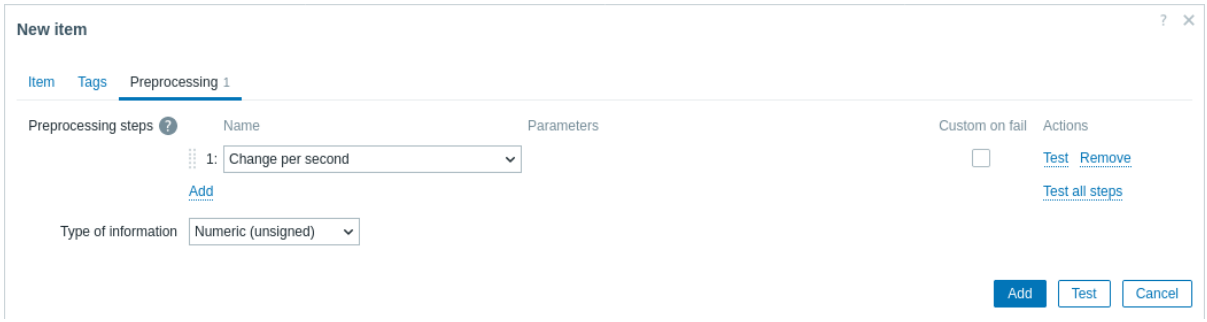
```
snmpget -v 2c -c <your_community_string> -On <device_IP> IF-MIB::ifHCInOctets.3
```

2. Créer un élément SNMP :

- Accédez à *Collecte de données* > *Hôtes*, puis cliquez sur l'onglet *Éléments* de votre hôte SNMP et cliquez sur *Créer un élément*.
- *Nom* : saisissez un nom descriptif (par exemple, « Trafic entrant du port 3 »).
- *Type* : sélectionnez *Agent SNMP*.
- *Clé* : indiquez 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 un sous-arbre de valeurs de manière asynchrone.



- *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 « Changement par seconde » pour calculer le débit.



Pour récupérer plusieurs valeurs en une seule transaction SNMP, vous pouvez spécifier plusieurs OID en utilisant la syntaxe `walk[OID1,OID2,...]`.

Traduction des OID entre les valeurs numériques et les 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.

- Traduction d'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 produire :

```
.1.3.6.1.2.1.31.1.1.1.6.3
```

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

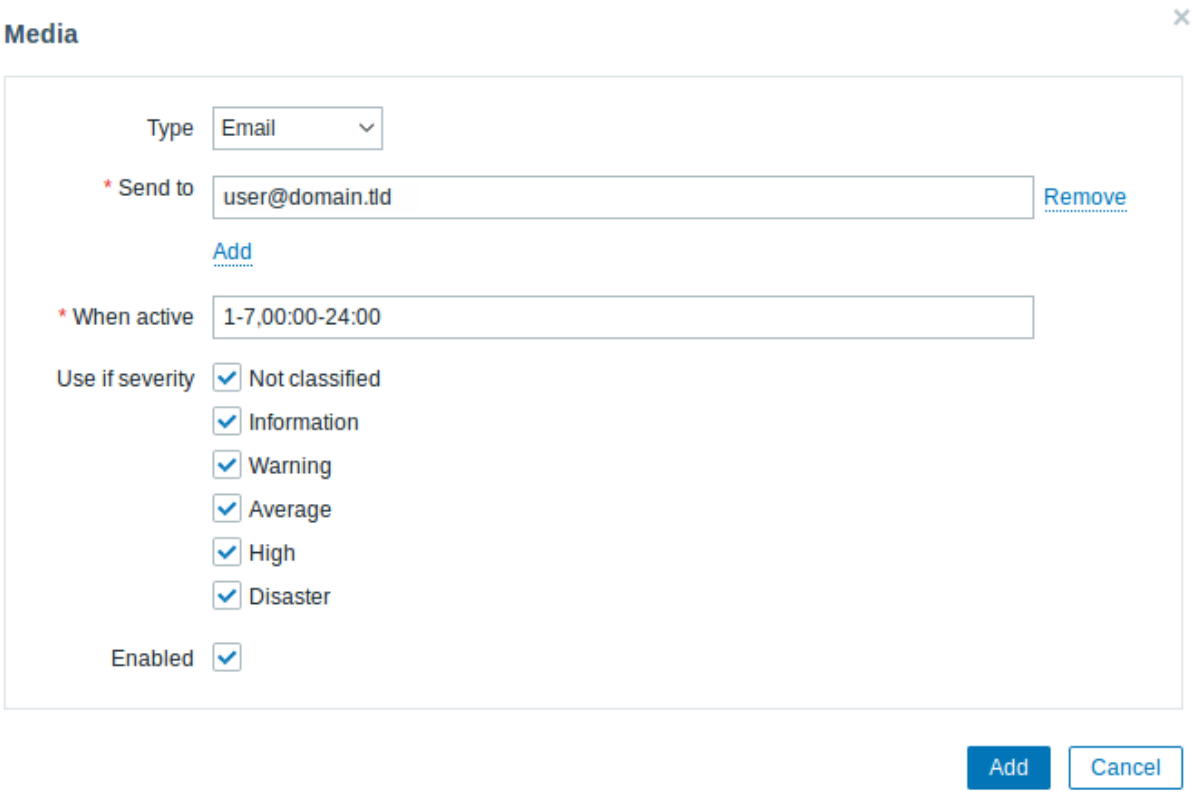
```
snmptranslate -IR -On .1.3.6.1.2.1.31.1.1.1.6.3
```

Cette commande peut produire :

```
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*, passez à l'onglet *Média* et ajoutez votre adresse e-mail.

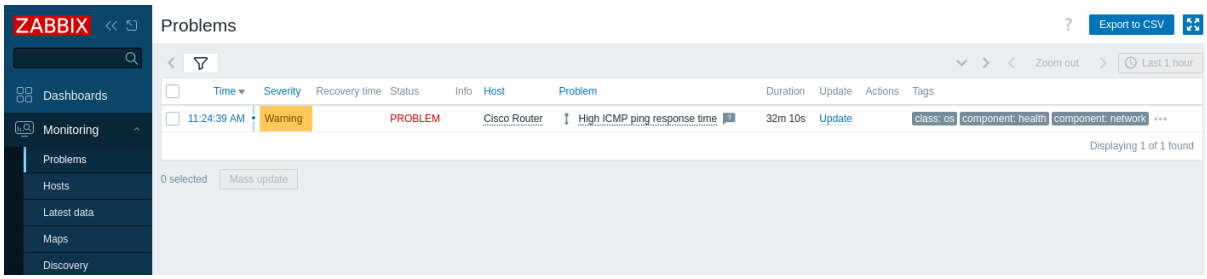


2. Suivez le guide pour Recevoir une notification de problème.

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 similaire de seuil de temps de réponse).
4. Définissez une valeur très basse (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 des alertes sont configurées, vous devriez également recevoir une alerte de problème.

8. Rétablissez la valeur d'origine de la macro et cliquez sur *Update* pour enregistrer les modifications.

9. Confirmez que le problème est résolu et disparaît de la section *Problems*.

Dépannage de la supervision SNMP Si vous remarquez que l'icône SNMP dans l'interface web de 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 vous en assurer, la commande suivante ne doit pas renvoyer d'erreur lorsque vous interrogez un équipement réseau :

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

Cela doit renvoyer des OID traduits sans erreurs.

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 détails de l'utilisateur SNMPv3 sont corrects.

4. Vérifiez que SNMP est correctement activé sur votre équipement 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 ou problème réseau ne bloque le trafic SNMP (généralement sur le port 161) entre le serveur Zabbix et l'appareil. Vous pouvez tester la connectivité avec :

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'aider à 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 :

- Création d'un élément - découvrez comment ajouter des métriques supplémentaires.
- Agent SNMP - informations supplémentaires sur la supervision SNMP avec Zabbix.
- Modèles standardisés pour les équipements 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 manière de configurer une règle de découverte réseau utilisée par Zabbix pour découvrir des hôtes et des services.

11 Surveiller le journal des événements Windows à l'aide de contrôles actifs

Introduction

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

À qui s'adresse ce guide

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 avec ce guide, vous devez télécharger et installer Zabbix server et Zabbix frontend conformément aux instructions correspondant à votre système d'exploitation. Vous devez également avoir Zabbix agent téléchargé et installé sur la machine Windows que vous souhaitez surveiller.

Configurer l'agent Zabbix pour la surveillance du journal des événements Windows

1. Ouvrez `zabbix_agentd.conf` (chemin par défaut : `C:\Program Files\Zabbix Agent\zabbix_agentd.conf`) sur votre hôte Windows et assurez-vous 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 le frontend Zabbix. Cela permet à l'agent de demander des contrôles actifs pour son hôte et depuis le serveur Zabbix spécifié. Par exemple :

```
ServerActive=192.0.2.0
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 :

- Assurez-vous 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 :

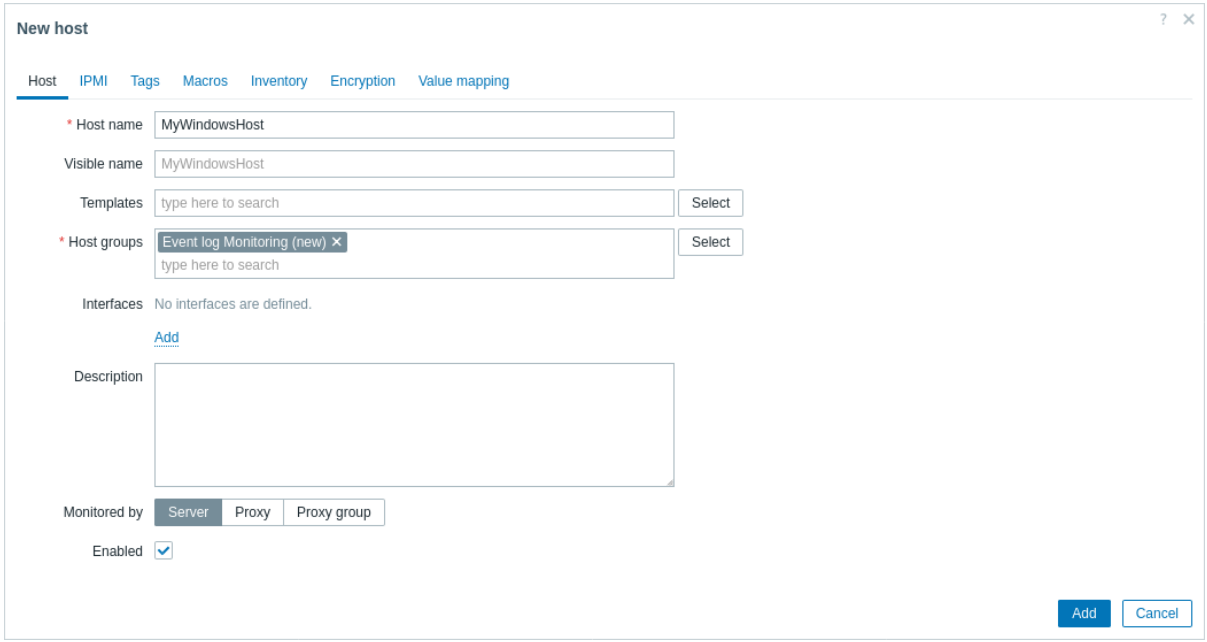
```
Test-NetConnection -ComputerName <Zabbix-server-IP> -Port 10051
```

Configurer le frontend Zabbix

1. Accédez à *Collecte de données* > *Hôtes* et créez un hôte :

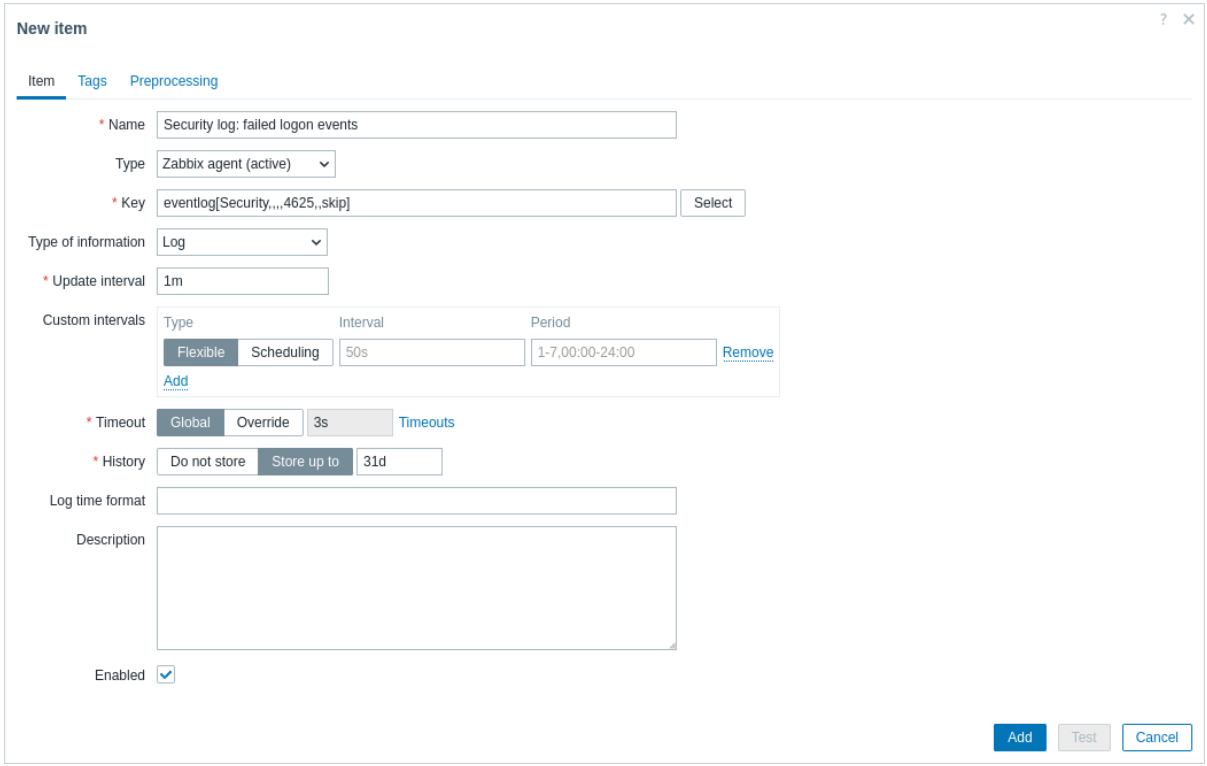
- Dans le champ *Nom de l'hôte*, saisissez un nom d'hôte (par exemple, "MyWindowsHost").
- Dans le champ *Groupes d'hôtes*, saisissez ou sélectionnez un groupe d'hôtes (par exemple, "Event log Monitoring").
- Appuyez sur *Ajouter* pour enregistrer l'hôte configuré.

> **Note:**
> Dans le champ *Modèles*, vous pouvez ajouter le modèle "Windows by Zabbix agent active" pour vous aider à résoudre les problèmes en observant si d'autres éléments actifs sur le même hôte sont en cours de mise à jour.



2. Créez un nouvel élément avec les paramètres suivants :

- Dans le champ *Nom*, 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 des événements).
- Dans le champ *Clé*, utilisez la clé d'élément eventlog. Par exemple, pour surveiller les tentatives de connexion échouées (ID d'événement : 4625) dans le journal Security et ignorer les entrées plus anciennes que la dernière vérification de l'élément (à l'aide du paramètre `skip`), saisissez la clé d'élément suivante : `eventlog[Security,,,,4625,,skip]`
- Dans la liste déroulante *Type d'information*, sélectionnez "Log".



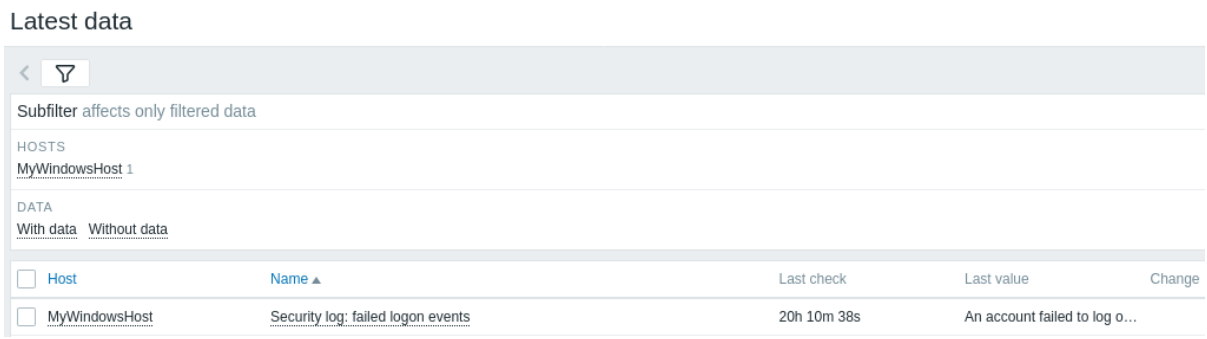
3. Cliquez sur *Ajouter* pour enregistrer l'élément.

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 reconnecter avec des identifiants incorrects.

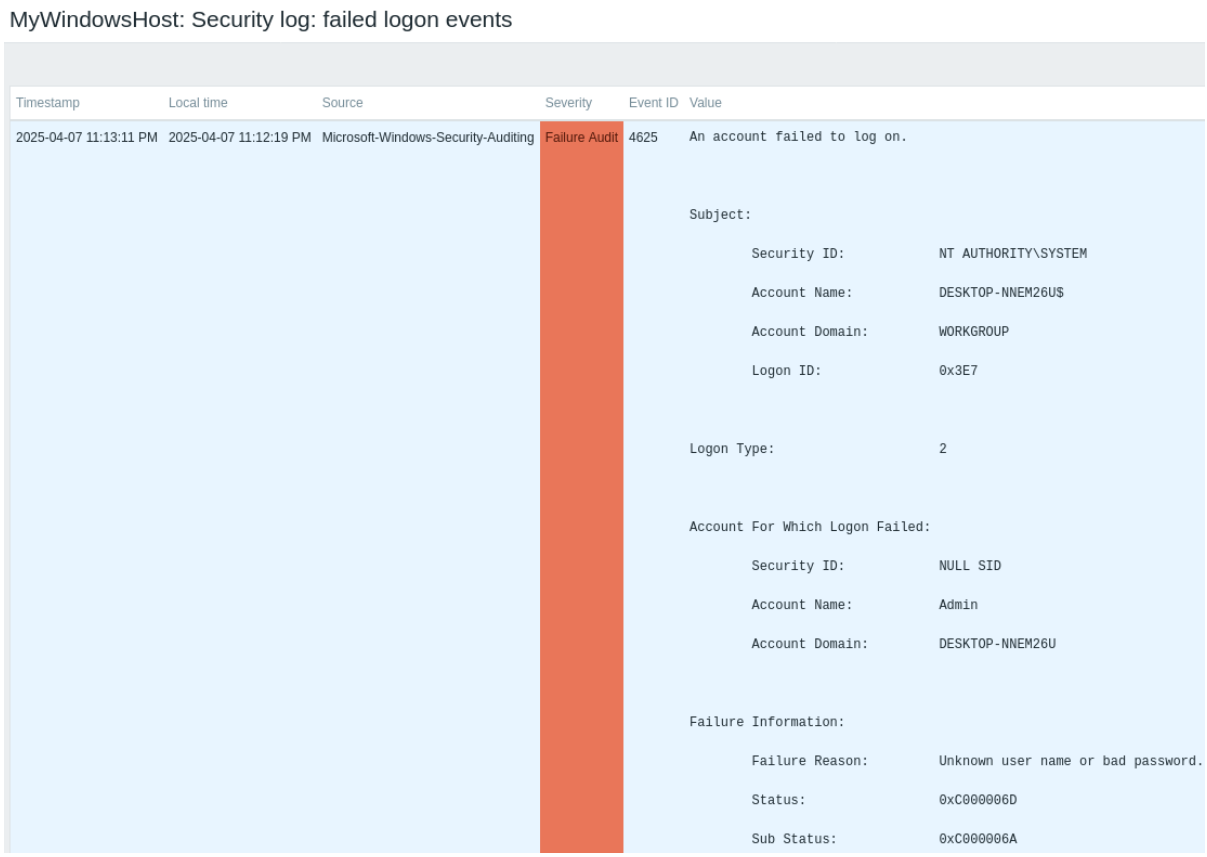
Ensuite, affichez les journaux collectés dans le frontend Zabbix :

1. Accédez à *Monitoring* > *Latest data* dans le frontend Zabbix.



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

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



4. Si aucune valeur de journal n’est affichée, passez à la section Dépannage du guide.

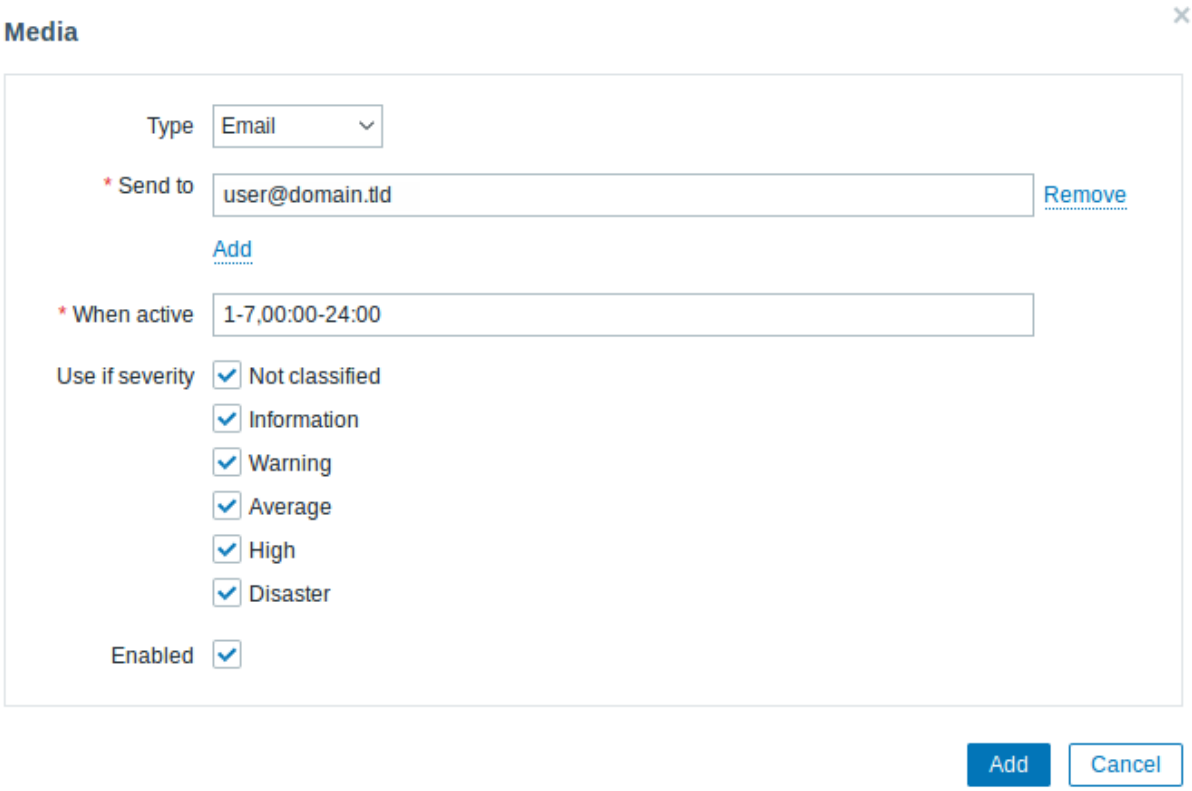
Configurer les alertes de problème

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

1. Accédez à Data collection > Hosts pour définir un déclencheur qui se déclenche lorsque votre élément de journal des événements enregistre le modèle 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 *User settings > Profile*, ouvrez l’onglet *Media* et ajoutez votre adresse e-mail.



3. Suivez le guide 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 ex. skip) et les autres paramètres exactement comme indiqué dans les clés d'élément spécifiques à Windows.

Voir aussi :

- Création d'un élément - découvrez comment ajouter des métriques supplémentaires.
- Agent Zabbix sur Microsoft Windows - instructions d'installation détaillées.
- Surveiller Windows avec l'agent Zabbix - guide complet pour configurer une surveillance 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 surveillance du journal des événements.
- Surveillance des fichiers journaux - instructions pour configurer Zabbix afin d'assurer une surveillance centralisée et une analyse des fichiers journaux, applicable aux journaux des événements Windows.

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 surveille de nombreux paramètres d'un réseau ainsi que l'état et l'intégrité des serveurs, des machines virtuelles, des applications, des services, des bases de données, des sites web, du 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 de serveur. Zabbix offre d'excellentes fonctionnalités de reporting et de visualisation des données basées sur les données stockées. Cela fait de Zabbix un outil idéal pour la planification de capacité.

Zabbix prend en charge à la fois l'interrogation et la réception passive de données. Tous les rapports et statistiques de Zabbix, ainsi que les paramètres de configuration, sont accessibles via une interface web. Une interface web permet d'évaluer l'état de votre réseau et la santé de vos serveurs 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 gratuit. Zabbix est écrit et distribué sous la licence AGPL-3.0. Cela signifie que son code source est librement distribué et accessible au grand public.

Le support commercial est disponible et fourni par Zabbix Company et ses partenaires dans le monde entier.

Pour en savoir plus sur les fonctionnalités de Zabbix.

Utilisateurs de Zabbix De nombreuses organisations de tailles diverses à travers le monde s'appuient sur Zabbix comme plateforme principale de supervision.

Architecture Zabbix se compose de plusieurs composants logiciels majeurs. Leurs responsabilités sont décrites ci-dessous.

Serveur

Le serveur Zabbix est le composant central auquel les agents signalent les informations de disponibilité et d'intégrité ainsi que les statistiques. 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 le serveur est exécuté.

Proxy

Le proxy Zabbix peut collecter des données de performance et de disponibilité pour le compte du serveur Zabbix. Un proxy est un composant facultatif d'un 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 locales et les applications, puis 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 En outre, 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. En passant à l'autre extrémité du spectre de Zabbix, vous devez d'abord disposer d'un élément pour créer un déclencheur. Vous devez disposer d'un déclencheur pour créer une action. Ainsi, si vous souhaitez recevoir une alerte indiquant que la charge CPU est trop élevée sur *Serveur X*, vous devez d'abord créer une entrée d'hôte pour *Serveur X*, suivie d'un élément pour surveiller son CPU, puis d'un déclencheur qui s'active si la charge CPU est trop élevée, suivi d'une action qui vous envoie un e-mail. Bien que cela puisse sembler représenter beaucoup d'étapes, avec l'utilisation des modèles, ce n'est en réalité 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 multitude de fonctionnalités dans un seul package.

Collecte de données

- contrôles de disponibilité et de performance
- prise en charge de SNMP (à la fois le trapping et le polling), IPMI, JMX, la supervision VMware
- contrôles personnalisés
- collecte des données souhaitées à des intervalles personnalisés
- effectuée par le serveur/proxy et par les agents

Définitions flexibles des seuils

- vous pouvez définir des seuils de problème très flexibles, appelés déclencheurs, en référençant des valeurs de la base de données backend

Alertes hautement configurables

- l'envoi des notifications peut être personnalisé selon le calendrier d'escalade, le destinataire et le type de média
- les notifications peuvent être rendues explicites et utiles à l'aide de variables de macro
- les actions automatiques incluent des commandes distantes

Création de graphiques en temps réel

- les éléments supervisés sont immédiatement représentés sous forme de graphiques à l'aide de la fonctionnalité de création de graphiques intégrée

Capacités de supervision web

- Zabbix peut suivre un parcours simulé de clics de souris sur un site web et vérifier le bon fonctionnement et le temps de réponse

Options de visualisation étendues

- possibilité de créer des graphiques personnalisés pouvant combiner plusieurs éléments dans une vue unique
- 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 nettoyage intégrée

Configuration facile

- ajouter des équipements supervisés en tant qu'hôtes
- les hôtes sont pris en compte pour la supervision une fois dans la base de données
- appliquer des modèles aux équipements supervisés

Utilisation de modèles

- regroupement des contrôles dans des modèles
- les modèles peuvent hériter d'autres modèles

Découverte réseau

- découverte automatique des équipements réseau
- auto-enregistrement des agents
- découverte des systèmes de fichiers, des interfaces réseau et des OID SNMP

Interface web rapide

- une interface web en PHP
- accessible depuis n'importe où
- navigation intuitive par clic
- journal d'audit

API Zabbix

- l'API Zabbix fournit une interface programmable vers Zabbix pour les manipulations de masse, l'intégration de logiciels tiers et d'autres usages.

Système d'autorisations

- 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 de hautes performances et une faible empreinte mémoire
- facilement portables

Prêt pour les environnements complexes

- la supervision à distance est facilitée par l'utilisation d'un proxy Zabbix

Glossaire Dans cette section, vous pouvez découvrir la signification de certains termes couramment utilisés dans Zabbix.

hôte

- tout dispositif physique ou virtuel, application, service ou toute autre collection logiquement liée 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 est utilisée pour « évaluer » les données reçues dans les éléments.

Lorsque les données reçues sont au-dessus du seuil, les déclencheurs passent de l'état « Ok » à l'état « Problem ». Lorsque les données reçues sont en dessous du seuil, les déclencheurs restent dans/reviennent à l'état « Ok ».

modèle

- un ensemble d'entités (éléments, déclencheurs, graphiques, règles de découverte de bas niveau, scénarios web) prêt à être appliqué à un ou plusieurs hôtes.

Le rôle des modèles est d'accélérer le déploiement des tâches de supervision sur un hôte ; ils facilitent également l'application de modifications de masse aux tâches de supervision. Les modèles sont liés directement 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 de quelque chose qui mérite attention, comme un déclencheur changeant d'état ou une découverte/autoregistration d'agent en cours.

tag d'événement

- un marqueur prédéfini pour l'événement. Il peut être utilisé dans la corrélation d'événements, la granularité des permissions, etc.

corrélation d'événements

- une méthode permettant de corréler les problèmes à leur résolution de manière souple 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 est dans l'état « Problem ».

mise à jour de problème

- options de gestion des problèmes fournies par Zabbix, telles que l'ajout d'un commentaire, l'acquittement, le changement de gravité ou la fermeture manuelle.

action

- un moyen prédéfini de réagir à un événement.

Une action se compose d'opérations (par ex. l'envoi d'une notification) et de conditions (*quand* l'opération est effectuée)

escalade

- un scénario personnalisé pour exécuter des opérations dans le cadre d'une action ; une séquence d'envoi de notifications/d'exécution de commandes distantes.

média

- un moyen de distribuer des notifications ; canal de diffusion.

notification

- un message concernant un événement, envoyé à un utilisateur via le canal de média choisi.

commande distante

- une commande prédéfinie qui est automatiquement exécutée sur un hôte surveillé lorsqu'une certaine condition est remplie.

scénario web

- une ou plusieurs requêtes HTTP pour vérifier la disponibilité d'un site web.

frontend

- l'interface web fournie avec Zabbix.

tableau de bord

- section personnalisable de l'interface web affichant des résumés et des visualisations d'informations importantes dans des unités visuelles appelées widgets.

widget

- unité visuelle affichant des informations d'un certain type et d'une certaine source (un résumé, une carte, un graphique, l'horloge, etc.), utilisée dans le tableau de bord.

Zabbix API

- Zabbix API vous permet d'utiliser le protocole JSON RPC pour créer, mettre à jour et récupérer des objets Zabbix (comme des hôtes, des éléments, des graphiques et autres) ou effectuer toute autre tâche personnalisée.

Zabbix server

- un processus central du logiciel Zabbix qui effectue la supervision, interagit avec les proxys et agents Zabbix, calcule les déclencheurs, envoie des notifications ; un référentiel central de données.

Zabbix proxy

- un processus qui peut collecter des données pour le compte de Zabbix server, en prenant une partie de la charge de traitement du serveur.

Zabbix agent

- un processus déployé sur les cibles de supervision pour surveiller activement les ressources locales et les applications.

Zabbix agent 2

- une nouvelle génération de Zabbix agent pour surveiller activement les ressources locales et les applications, permettant l'utilisation de plugins personnalisés pour la supervision.

> **Attention:**
> Comme Zabbix agent 2 partage de nombreuses fonctionnalités avec Zabbix agent, le terme « Zabbix agent » dans la documentation désigne les deux - Zabbix agent et Zabbix agent 2, si le comportement fonctionnel est identique. Zabbix agent 2 n'est spécifiquement nommé que lorsque sa fonctionnalité diffère.

chiffrement

- prise en charge des communications chiffrées entre les composants Zabbix (server, proxy, agent, zabbix_sender et zabbix_get utilities) à l'aide du protocole Transport Layer Security (TLS).

autoregistration d'agent

- processus automatisé par lequel un Zabbix agent lui-même est enregistré en tant qu'hôte et la supervision est démarrée.

découverte réseau

- découverte automatisée des périphériques réseau.

découverte de bas niveau

- découverte automatisée d'entités de bas niveau sur un périphérique particulier (par ex. systèmes de fichiers, interfaces réseau, etc.).

règle de découverte de bas niveau

- ensemble de définitions pour la découverte automatisée d'entités de bas niveau sur un périphérique.

prototype d'élément

- *une métrique avec certains paramètres comme variables, prête pour la découverte de bas niveau*. Après la découverte de bas niveau, les variables sont automatiquement remplacées par les paramètres réellement découverts et la métrique commence automatiquement à collecter des données.

prototype de déclencheur

- *un déclencheur avec certains paramètres comme variables, prêt pour la découverte de bas niveau*. Après la découverte de bas niveau, les variables sont automatiquement remplacées par les paramètres réellement découverts et le déclencheur commence automatiquement à évaluer les données.

Des *prototypes* de certaines autres entités Zabbix sont également utilisés dans la découverte de bas niveau - prototypes de graphiques, prototypes d'hôtes, 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.

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.

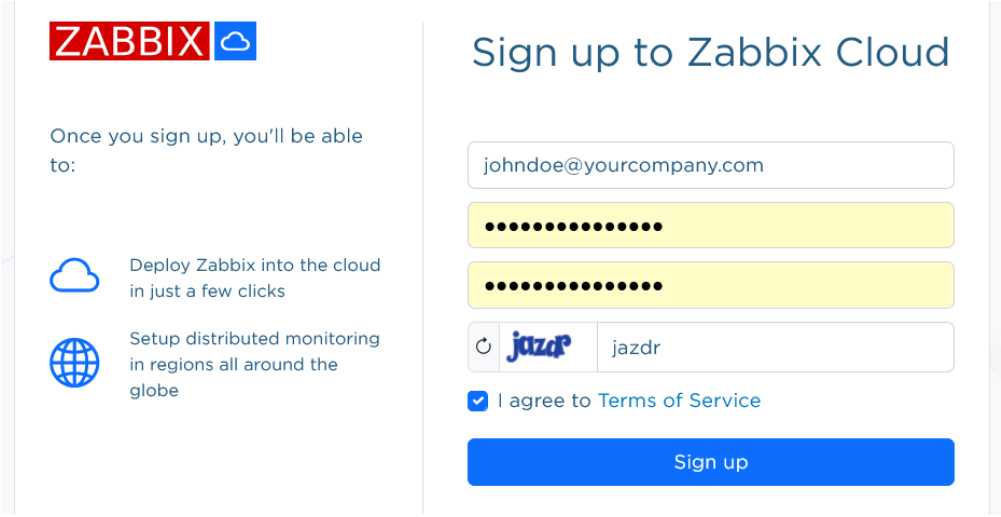
1 Déployer Zabbix dans le cloud

Pour démarrer Zabbix dans le cloud, suivez ce processus :

- Connectez-vous ou inscrivez-vous sur cloud.zabbix.com
- Créez un nœud cloud
- 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 sign up pour vous inscrire.



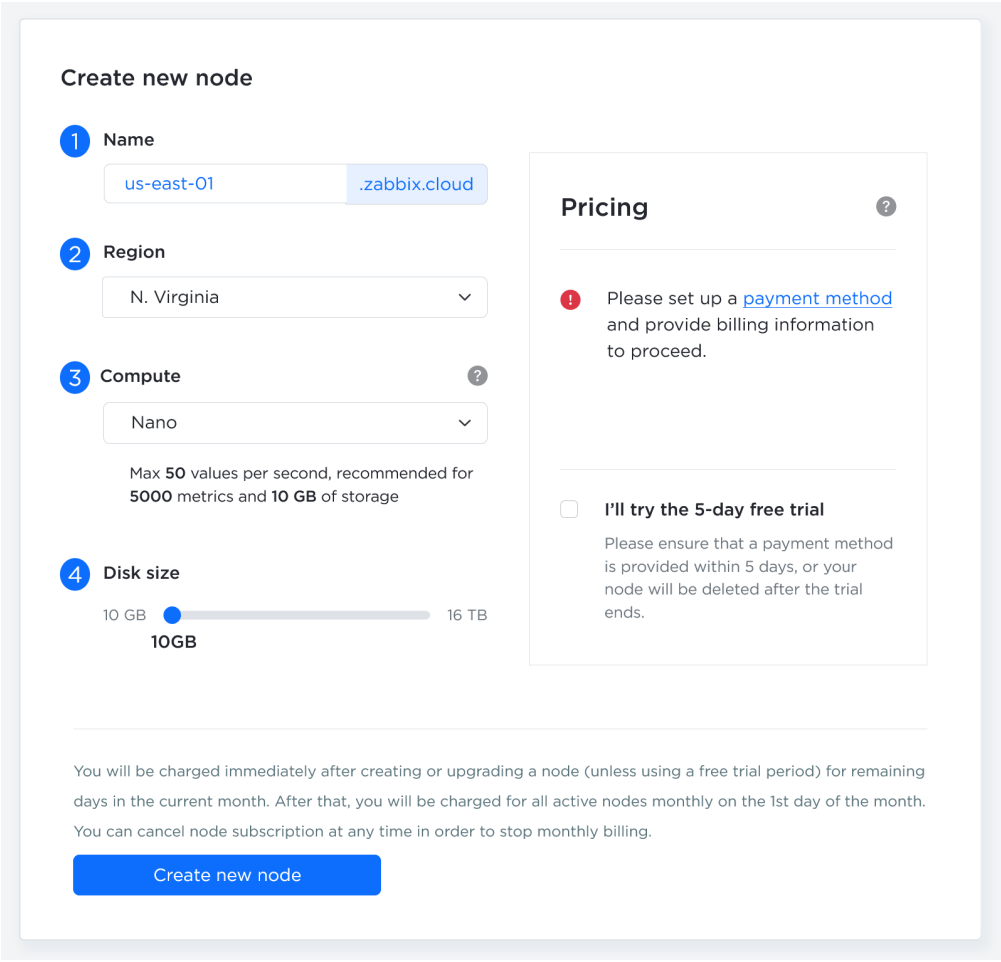
Après votre inscription, vous recevrez dans votre e-mail une confirmation de votre nouveau compte.

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, hébergée dans le cloud. Il comprend le serveur Zabbix, la base de données et l'interface web.

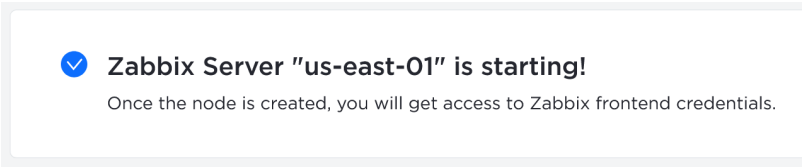
Une fois connecté, vous verrez qu'une organisation *Private* existe déjà par défaut pour votre compte. Vous pourrez renommer l'organisation par défaut plus tard.

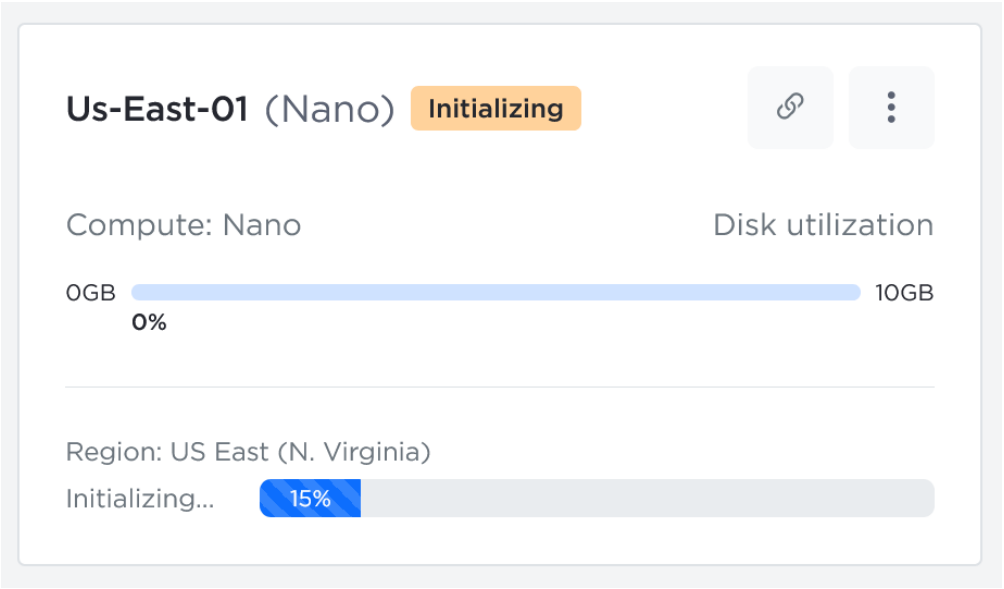
Cliquez sur *Create New Node*. Un formulaire de configuration du nœud s'ouvrira.

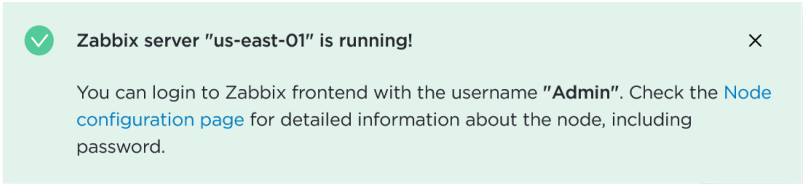


Parameter	Description
Name	Sélectionnez un nom de nœud unique. Le nom du nœud forme une URL unique pour le serveur/l'interface cloud, au format `<your-name>.zabbix.cloud`.
Region	Sélectionnez l'emplacement du centre de données le plus proche afin de minimiser la latence réseau.
Compute	Sélectionnez la puissance de calcul du nœud. Pour déterminer le niveau de ressources dont vous avez besoin, consultez les détails tarifaires. Vous pourrez toujours effectuer une mise à niveau plus tard si vous avez besoin de plus de puissance. Un mode de facturation sera requis avant de créer un nœud non 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 montant ne vous sera facturé pendant la période d'essai, mais une fois celle-ci terminée, 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 selon le niveau. Augmentez-le si vous stockez de grands volumes d'historique ou de tendances. Notez que des disques plus grands 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.

Après avoir rempli le formulaire, cliquez sur *Create New Node*. Pendant l'initialisation du nœud, un message correspondant s'affiche et le nœud apparaît déjà sous Organizations.





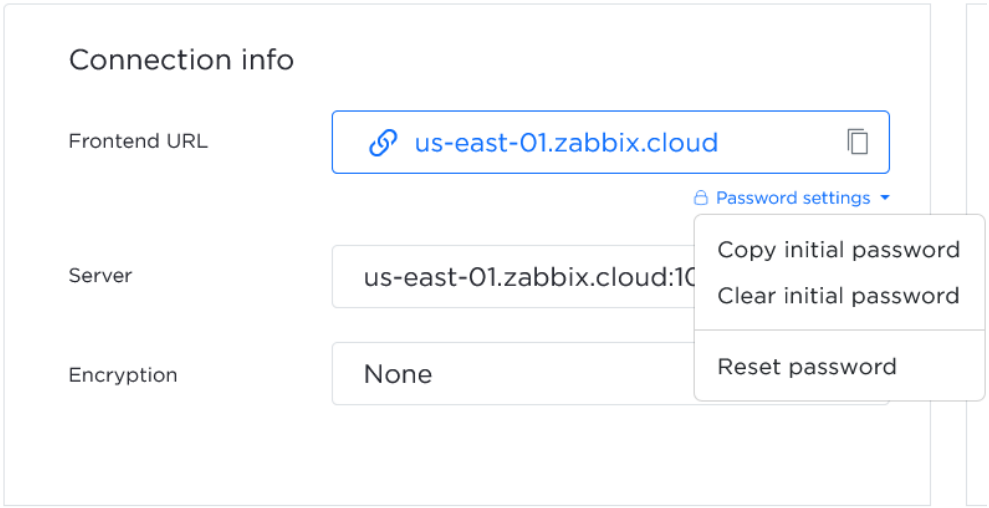


Commencer à utiliser Zabbix

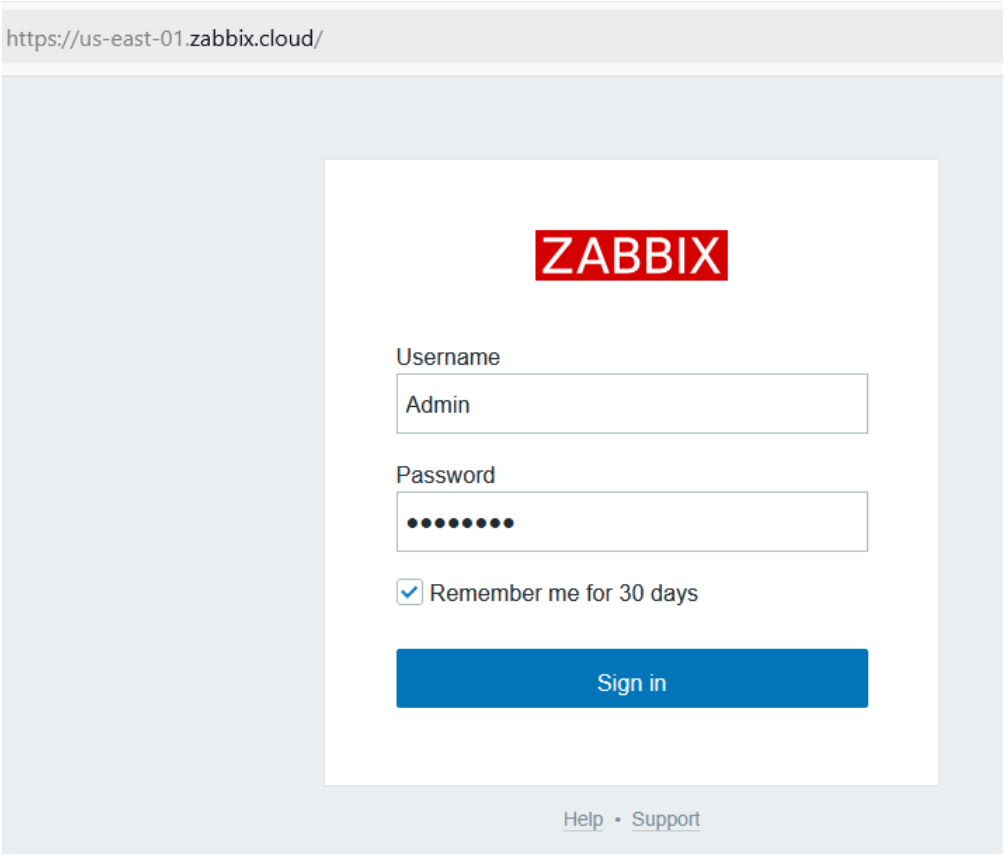
Lorsque le statut du nœud passe de *Initializing* à *Running*, vous pouvez vous connecter à l'interface web de Zabbix.

Pour accéder au mot de passe initial de l'interface web, 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**, cliquez sur *Password settings* et sélectionnez *Copy initial password* dans la liste déroulante.



Cliquez sur l'URL de l'interface web pour ouvrir l'interface web de Zabbix (`us-east-01.zabbix.cloud`).



Connectez-vous à l'aide des identifiants suivants :

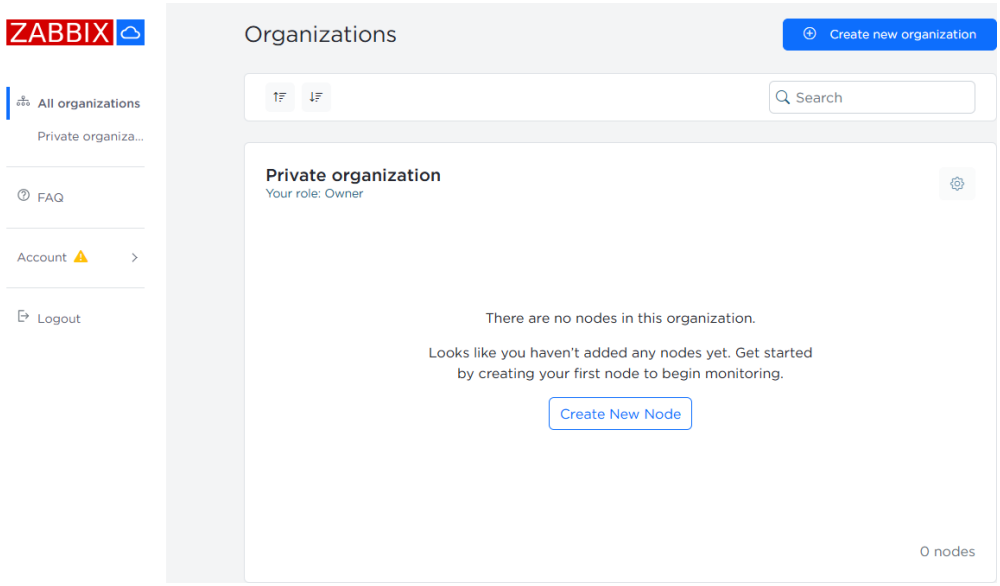
- *Username* - `Admin`
- *Password* - indiquez le mot de passe initial (par exemple, HzTG9t7Y)

Pour commencer la supervision, vous devez définir des filtres d'accès : une ou plusieurs adresses IP autorisées à se connecter au serveur cloud Zabbix. Au moins une 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 des paramètres pour accéder aux options de gestion.

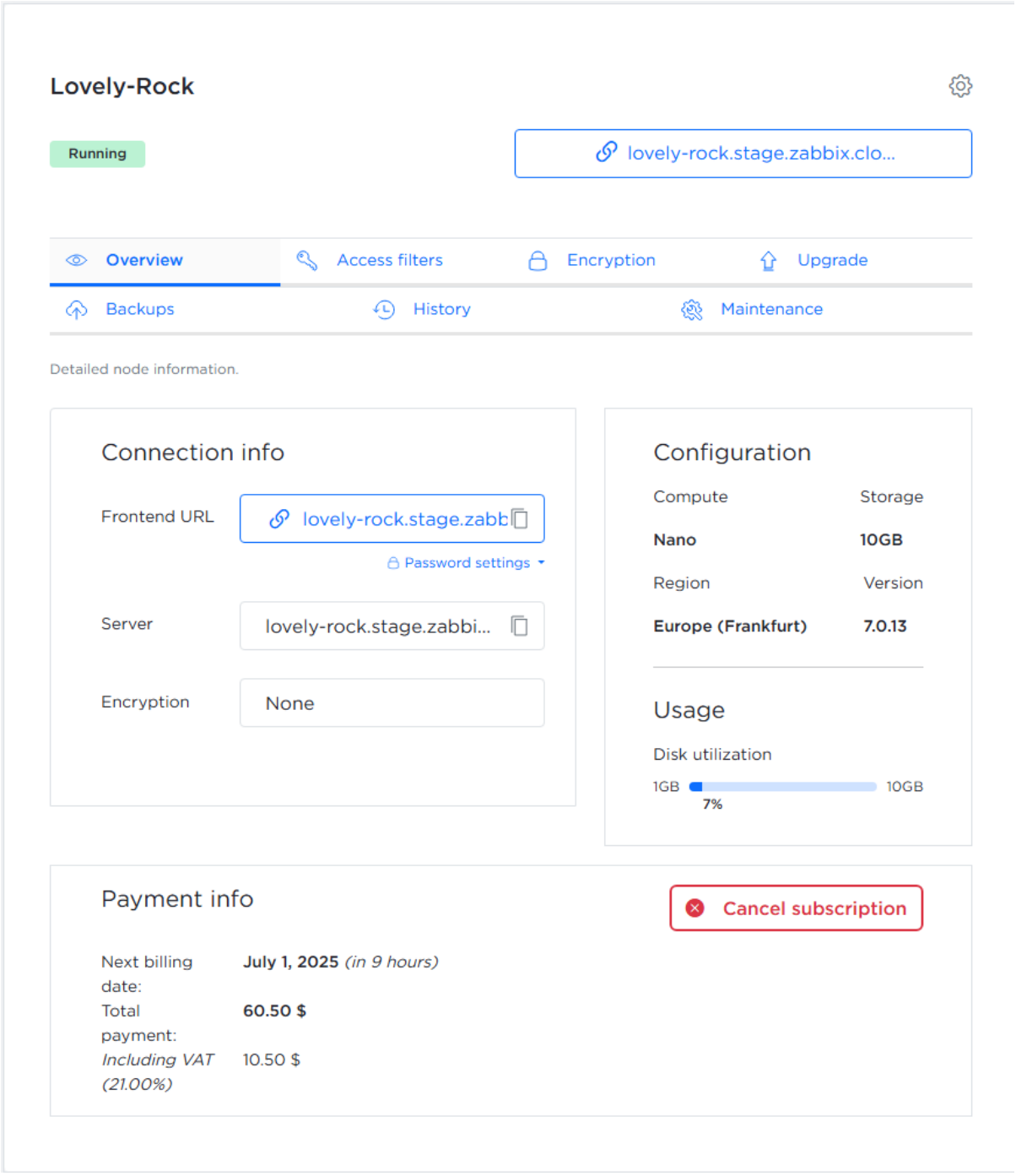
2 Configuration du nœud

Aperçu

Pour accéder à la page de configuration du nœud, cliquez sur le nœud soit dans *Toutes les organisations*, depuis votre menu d'organisation, soit en utilisant le lien *Page de configuration du nœud* affiché après le lancement d'un nœud.

Dans la vue de configuration du nœud, chaque onglet vous permet de gérer un aspect spécifique de votre nœud :

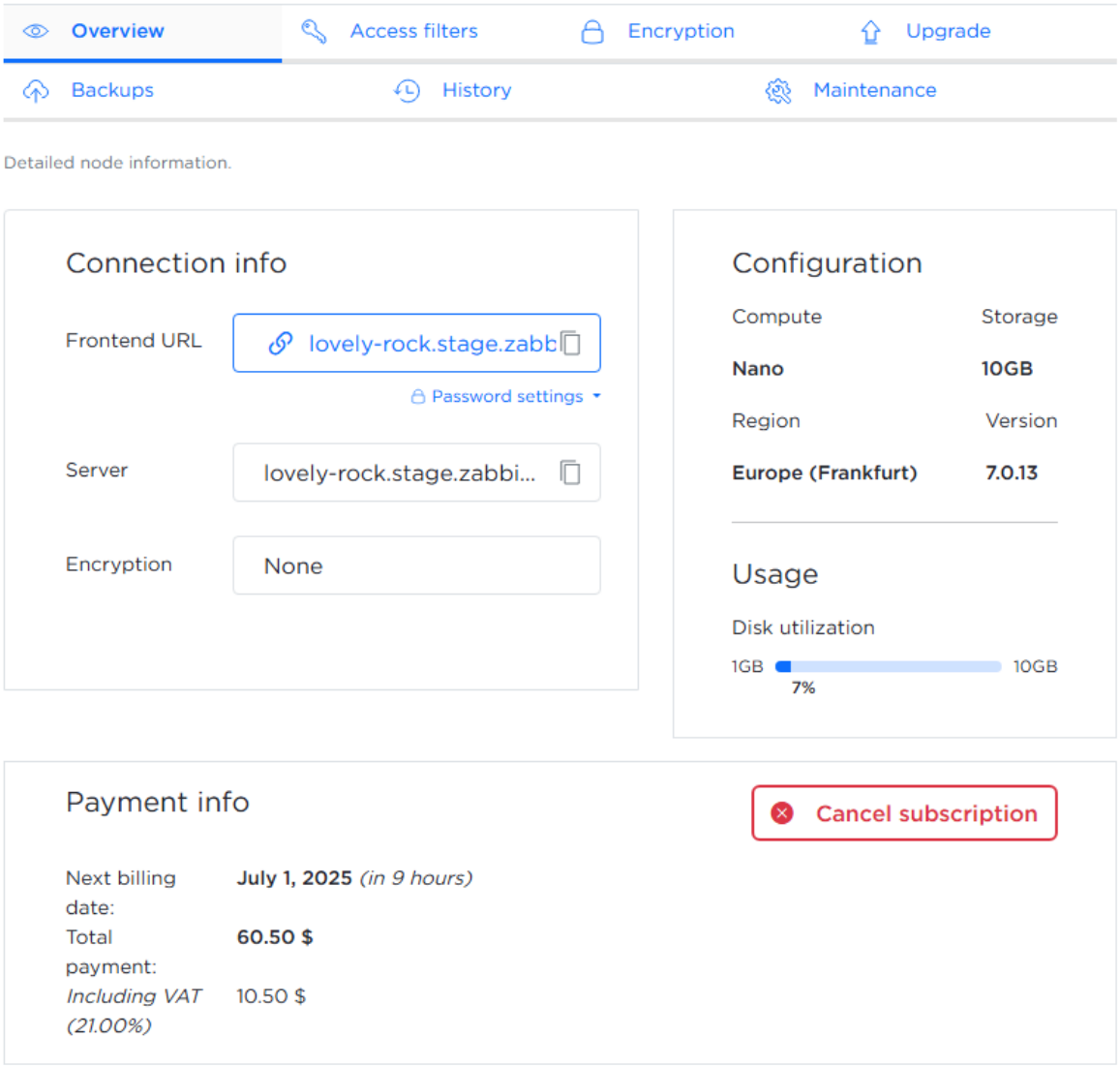
- Aperçu - afficher les détails du nœud, ouvrir son interface web et gérer les utilisateurs du nœud.
- Filtres d'accès - contrôler l'accès par adresse IP ou sous-réseau.
- Chiffrement - configurer TLS en téléversant des certificats, des clés et, éventuellement, des listes de révocation de certificats (CRL).
- 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 par fuseau horaire, jour et heure.
- Mise à niveau - ajuster les ressources de calcul/stockage ou gérer le statut de votre essai/paiement.



Note:
Certains onglets peuvent être indisponibles selon le type d'utilisateur qui consulte la page et le statut du nœud (essai/non-essai).

Onglet Vue d'ensemble

L'onglet *Vue d'ensemble* fournit des informations détaillées sur le nœud.



Il est divisé en trois sections :

Informations de connexion :

- URL du frontend cliquable avec option de copie dans le presse-papiers
- Paramètres du mot de passe
- Méthode de chiffrement actuelle

Configuration et utilisation :

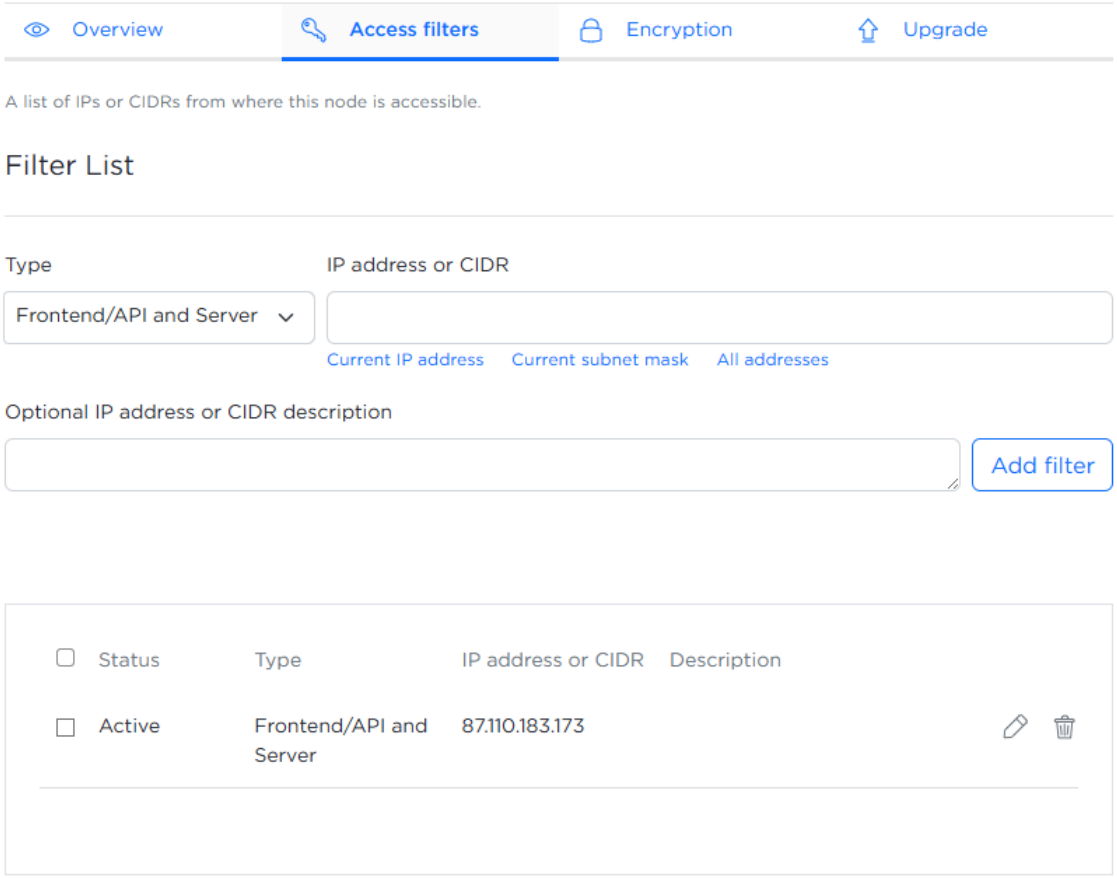
- Niveau de calcul, stockage maximal, région
- Version des composants Zabbix
- Utilisation du disque

Informations sur le nœud/Informations de paiement :

- Date d'expiration (pour les nœuds d'essai et prépayés)
- Un bouton spécifique au contexte s'affiche selon le type de nœud une fois l'initialisation du nœud terminée : - *Passer à une offre payante* - disponible pour les nœuds en essai gratuit - *Annuler l'abonnement* - disponible pour les nœuds payants avec un abonnement actif - *Se réabonner à une offre payante* - affiché si un abonnement a été annulé - Aucun bouton d'action n'est affiché pour les nœuds prépayés, qui ont une longue durée de validité et ne peuvent pas être annulés

Onglet 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* - utilisez la liste déroulante pour choisir quel composant Zabbix limiter : frontend/API uniquement, serveur uniquement, ou les deux.
- *Adresse IP ou CIDR* - saisissez l'adresse IP ou le CIDR manuellement, ou renseignez automatiquement le champ en sélectionnant l'option appropriée.
- *Description facultative de l'adresse IP ou du CIDR* - ajoutez une description facultative pour chaque 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 statut, leur type, leur adresse IP ou sous-réseau, ainsi que 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 ensuite depuis une autre adresse IP ou un autre réseau, vous devez ajouter cette adresse ici, sinon l'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 entre les composants Zabbix en téléversant les fichiers de certificat requis.

Pour activer une communication sécurisée à l'aide de certificats, sélectionnez *CERTIFICATE* et téléversez les fichiers suivants depuis votre ordinateur :

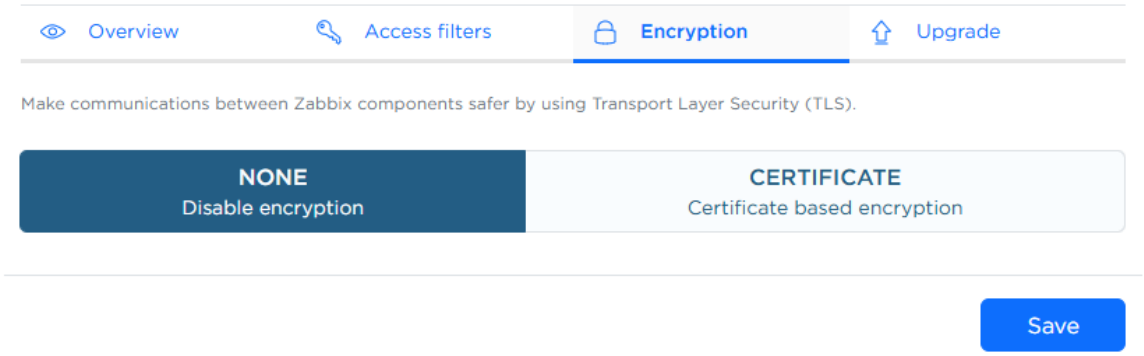
- *Certificats CA racine* - utilisés pour vérifier l'authenticité des certificats des pairs.
  Cliquez sur *Choose file* et sélectionnez le fichier contenant les certificats CA racine de confiance (généralement un fichier `.crt` ou `.pem`, encodé en PEM).

- *Chaîne de certificats* - représente la chaîne de confiance de votre propre certificat.
  Cliquez sur *Choose file* et sélectionnez le fichier encodé en PEM (par exemple, `.crt` ou `.pem`) qui inclut votre certificat ainsi que tous les certificats CA intermédiaires.

- *Clé privée* - la clé privée correspondant à votre certificat.
  Cliquez sur *Choose file* et sélectionnez le fichier `.key` encodé en 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 considérés comme fiables.

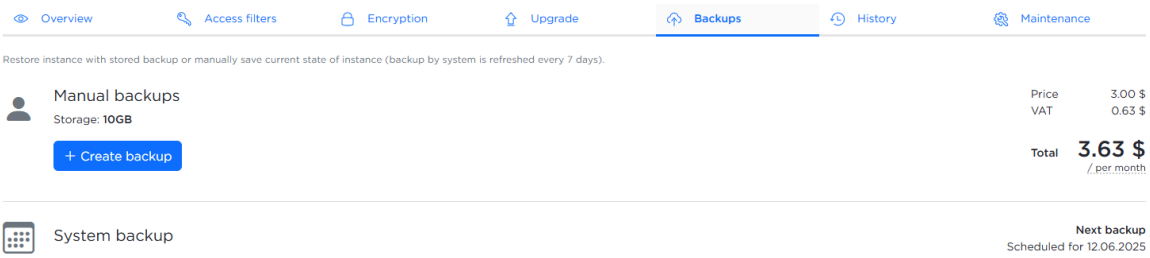
Après avoir sélectionné les fichiers, cliquez sur *Save*. Les fichiers seront téléversés, validés et appliqués pour configurer le chiffrement TLS entre les composants Zabbix.



Onglet Sauvegardes

L'onglet *Sauvegardes* 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.



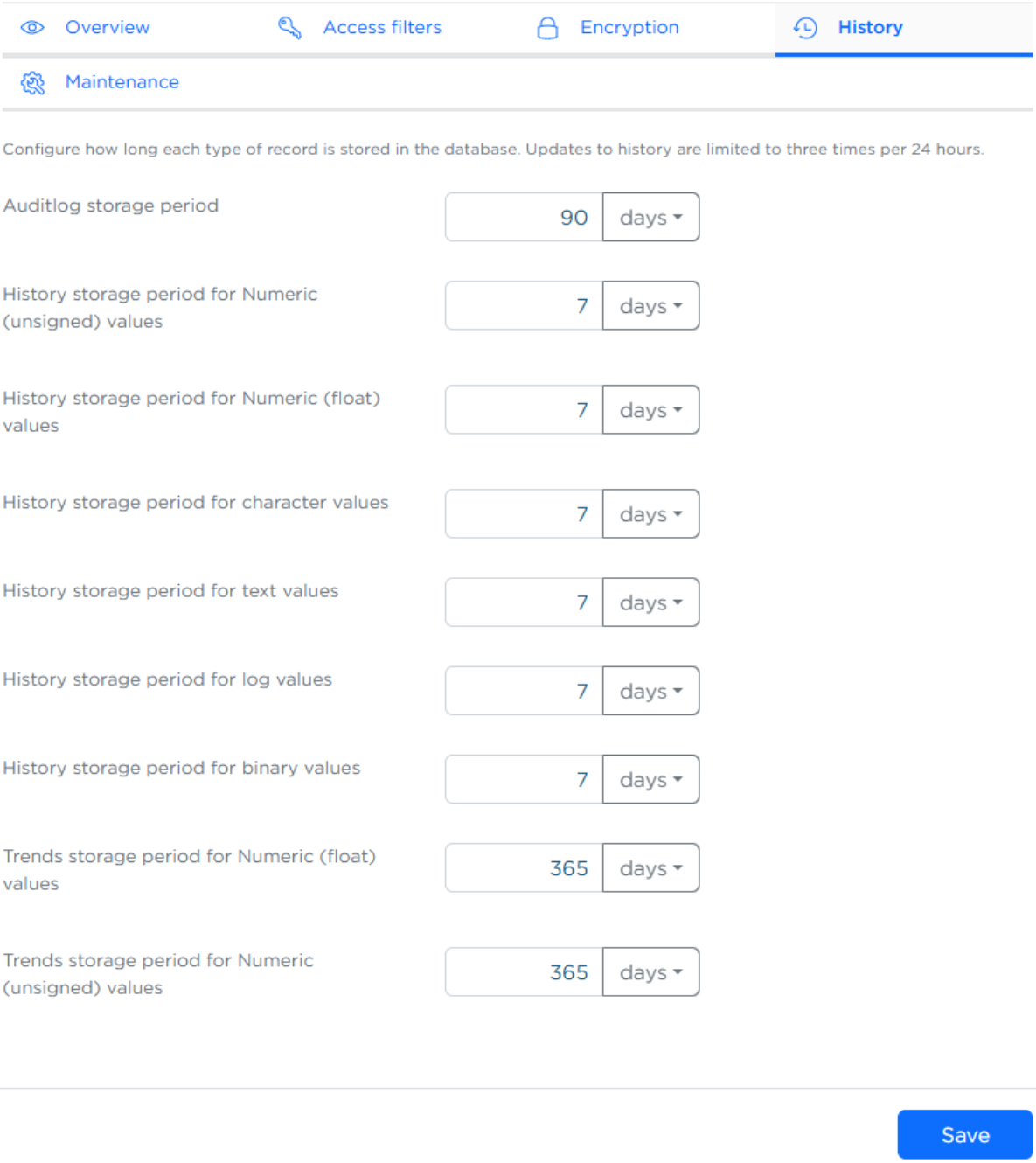
Pour créer une sauvegarde :

1. Cliquez sur le bouton *Créer une sauvegarde*.
2. Saisissez une description facultative dans la fenêtre contextuelle.
3. Cliquez sur *Créer* 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.



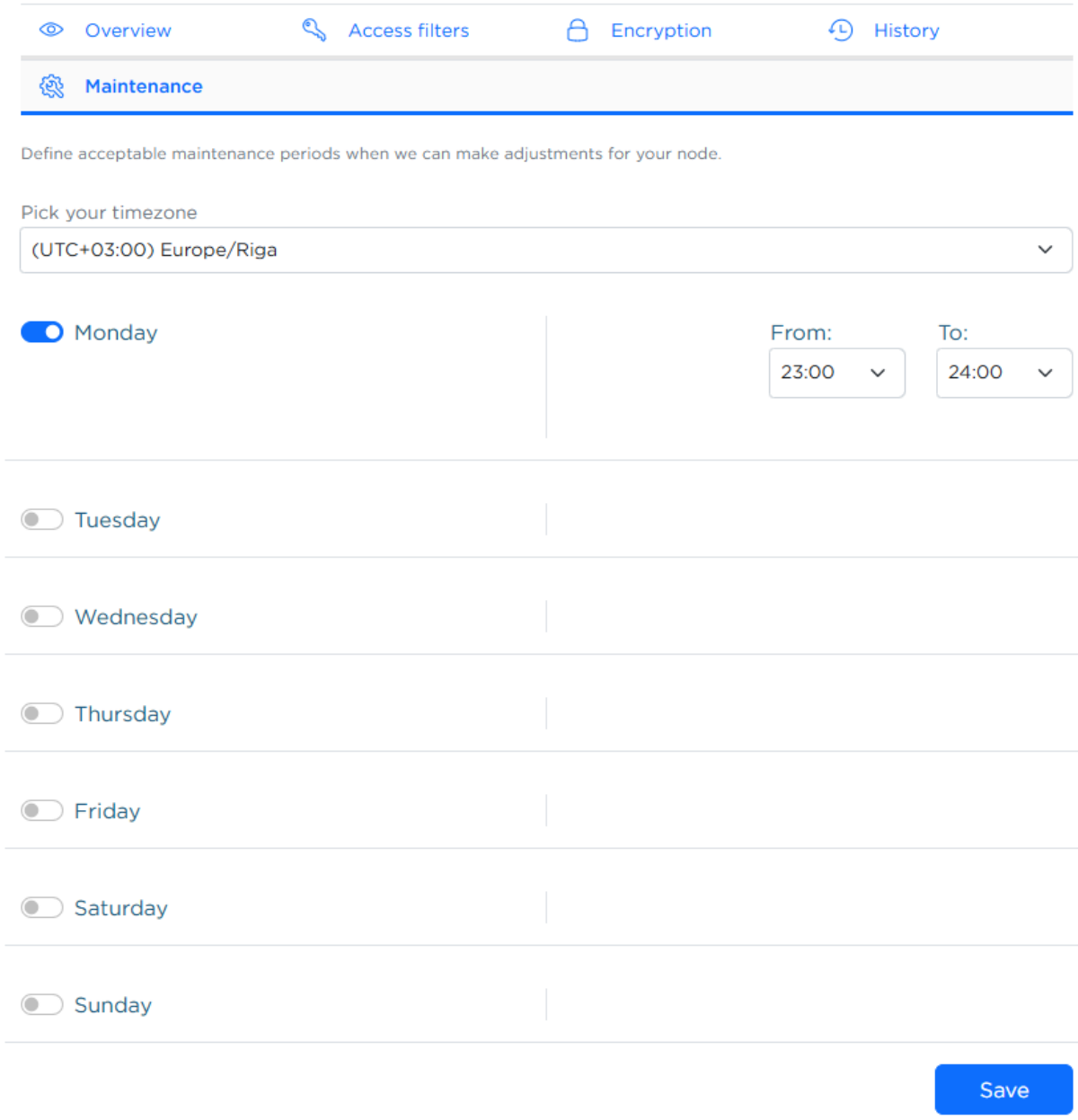
Vous pouvez définir la période de stockage pour chaque type de données (par ex. numériques, texte, journaux) et choisir l'unité (jours, mois, années). Les modifications peuvent être effectuées jusqu'à trois fois toutes les 24 heures.

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 avoir un impact sur les performances. Il est recommandé de configurer les périodes de conservation en fonction de vos besoins réels en matière de données et d'analyse.
 Zabbix Cloud utilise le partitionnement de base de données pour gérer le stockage des données au lieu du housekeeping Zabbix traditionnel. Cela permet d'obtenir de meilleures performances et une gestion de la conservation plus prévisible.

Onglet Maintenance

L'onglet *Maintenance* vous permet de définir des fenêtres de maintenance acceptables pendant lesquelles des mises à jour ou des ajustements peuvent être effectués sur votre nœud.

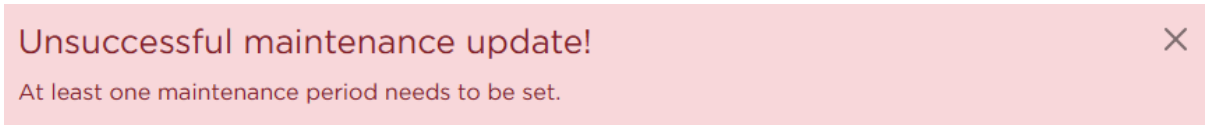


Vous devez :

- Sélectionner votre fuseau horaire dans la liste déroulante.
- Choisir au moins un jour et un créneau horaire pour la maintenance.

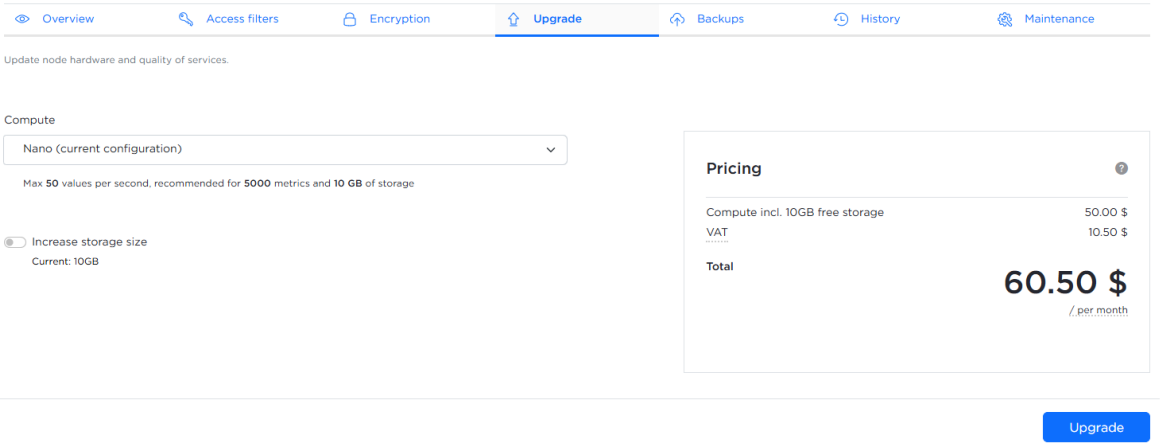
Cliquez sur *Save* pour appliquer les modifications.

Si aucun jour ou créneau horaire n'est sélectionné, une erreur s'affichera.



Onglet Mise à niveau

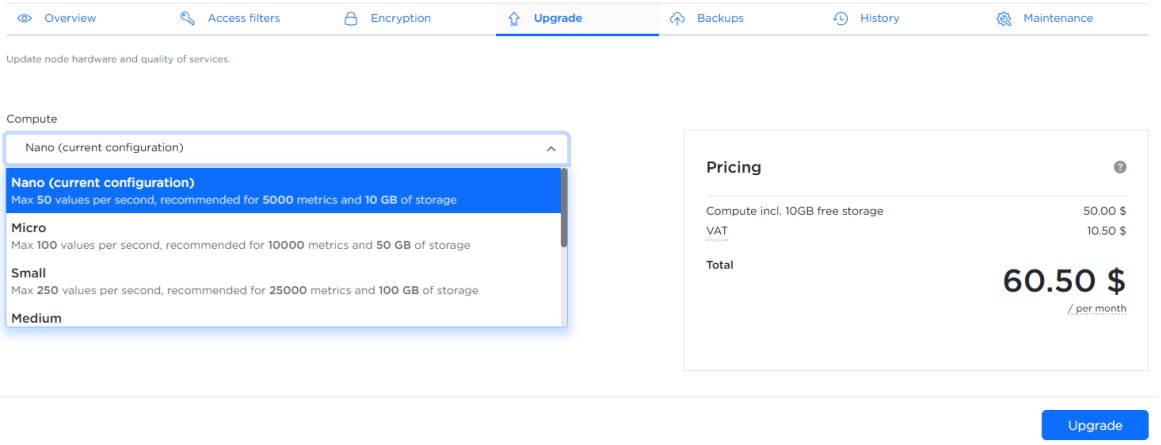
L'onglet *Mise à niveau* vous permet d'augmenter les ressources de votre nœud ou de gérer son abonnement.



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 utilisateur (contactez l'assistance Zabbix pour toute demande de réduction).
Les nœuds en essai gratuit peuvent être convertis en abonnements payants.

Pour augmenter le niveau de calcul :

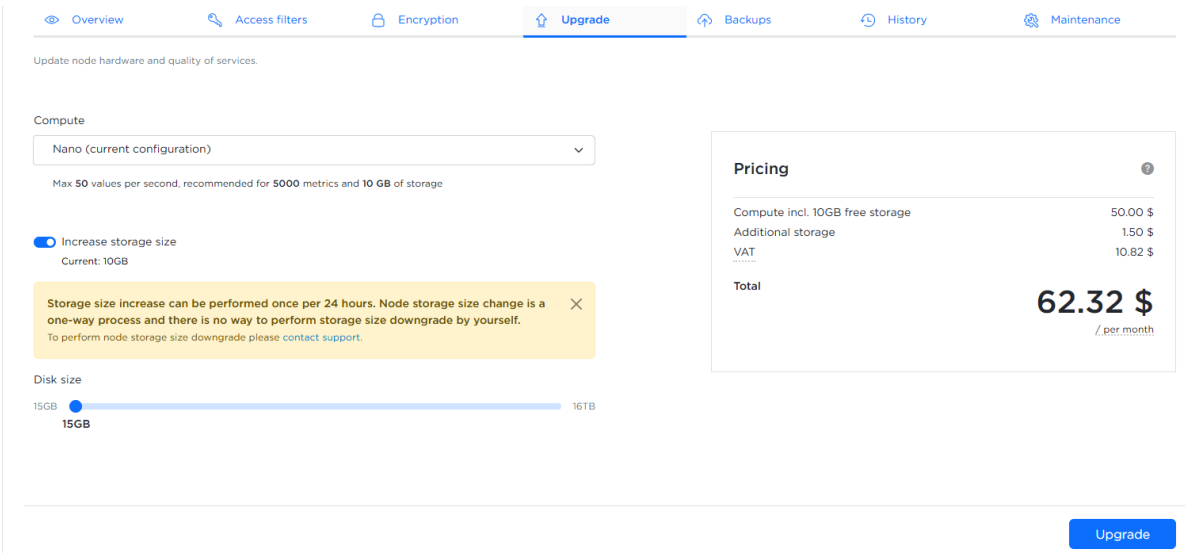
1. Cliquez sur la liste déroulante *Compute*.



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 le bouton *Upgrade*.

Note:
Le nouveau prix mensuel total est affiché dans le tableau *Pricing* avant de confirmer les modifications.

3 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.

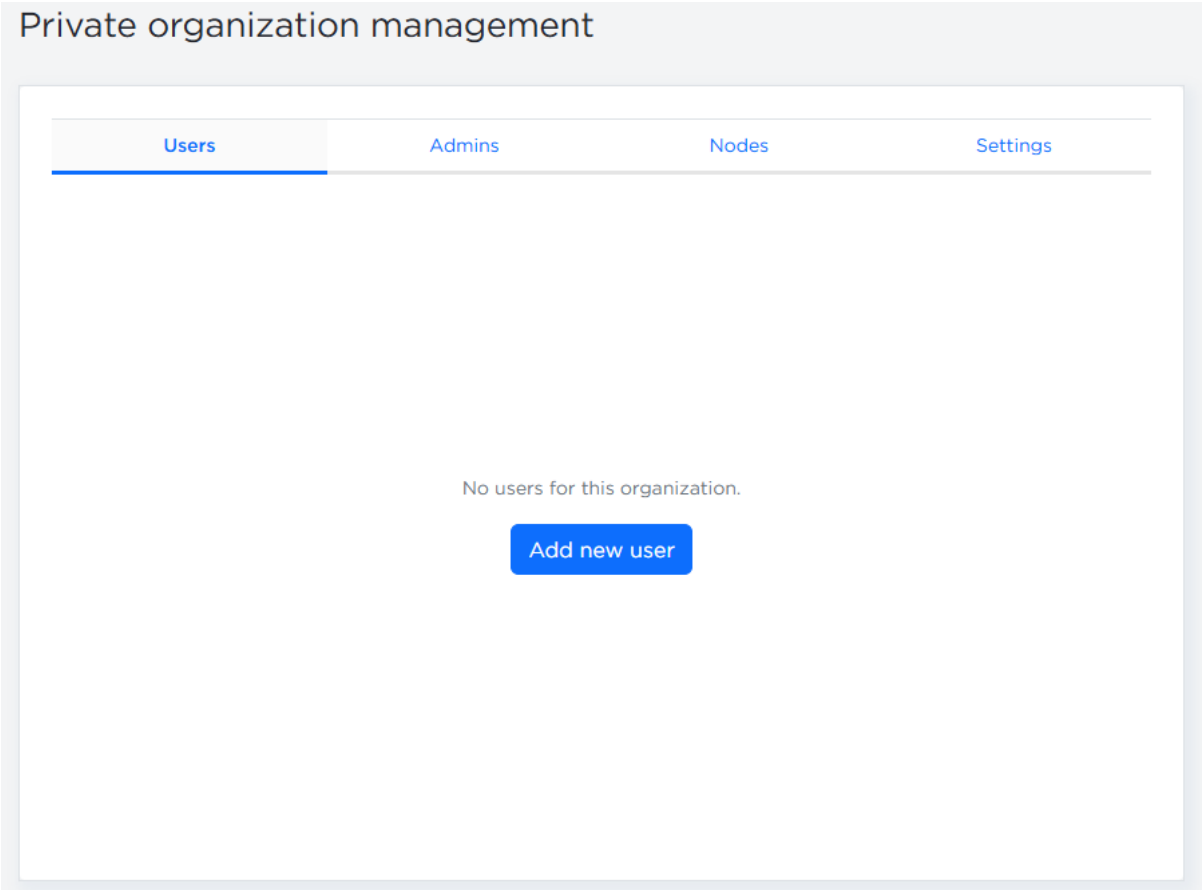
Ajout d'utilisateurs aux organisations

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*.

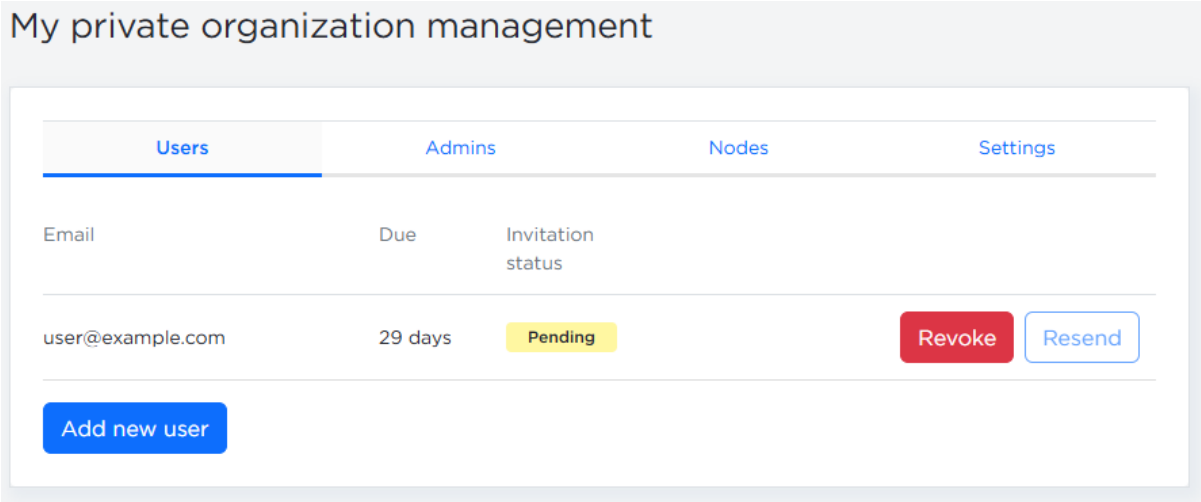
Notez que la sélection du rôle *User* ajoute l'utilisateur à l'organisation, mais ne lui accorde pas l'accès à des nœuds.

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.

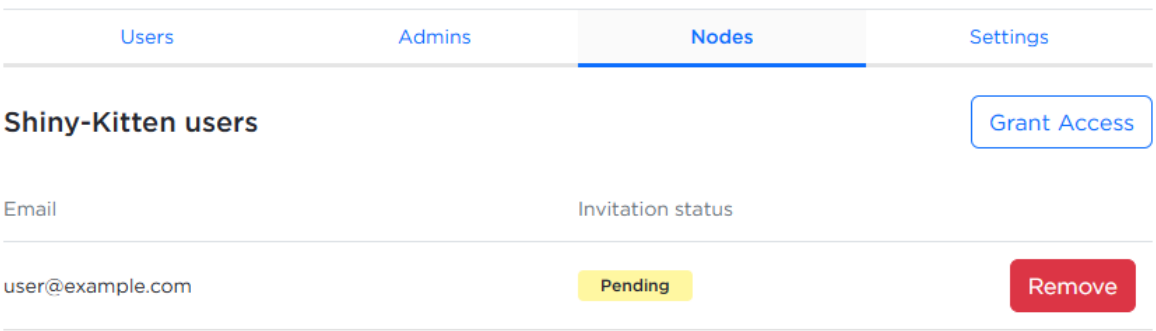
L'utilisateur invité sera ajouté avec le statut *Pending*. Vous pouvez révoquer l'invitation ou la renvoyer (le renvoi est possible une fois toutes les 10 minutes — le bouton sera désactivé si cette limite est atteinte).



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.

4 Principales différences de Zabbix Cloud

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 fourni ci-dessous. Les sections suivantes décrivent certains éléments en détail.

Fonctionnalité / Aspect	Zabbix Cloud	Zabbix sur site
Cycle de publication	Aligné sur les versions LTS	Vous choisissez votre version de Zabbix (stable, bêta, builds personnalisés)
Facilité d'utilisation	Plateforme prête à l'emploi, entièrement gérée par Zabbix	Nécessite une installation, une configuration et une maintenance
Correctifs de vulnérabilités	Correctifs automatiques de l'OS/Zabbix	Application manuelle des correctifs par l'utilisateur
Certificat HTTPS	Certificat valide prêt à l'emploi (sans Let's Encrypt)	Auto-signé par défaut (Let's Encrypt ou une autre AC peuvent être configurés)
Performances de la BD (INSERT/UPDATE/SELECT)	Partitionnement automatique pour une vitesse maximale ; pas de DELETE de longue durée	Partitionnement et nettoyage manuels (transactions DELETE longues possibles)
Réglage des performances	Préconfiguré par l'équipe Zabbix Cloud (pas de journaux des modifications)	Visibilité et contrôle complets sur les paramètres de réglage
Contrôle du pare-feu	Géré par l'équipe Zabbix Cloud ; les ports GUI/trapper ne sont pas exposés publiquement	Contrôle total des règles de pare-feu et de l'exposition des ports
Isolation et protection de la BD	Isolation élevée, pas d'accès direct à la BD/SSH	Dépend de votre réseau et de la sécurité de l'hôte
Haute disponibilité (HA)	Fonctionne 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 (pas de récepteur de traps en HA/avec équilibrage de charge)	Directement ou via un proxy, HA/équilibrage de charge possibles
Interrogation SNMP	Difficile sans proxys (chaque appareil nécessite un NAT/port personnalisé)	Interrogation SNMP native ; proxy facultatif
Paramètres de rétention	Historique/Tendances/Audit uniquement via l'interface utilisateur (pas d'API ni de fichier de configuration)	Configurables dans le fichier de configuration du serveur Zabbix ou via l'API
Scripts personnalisés	`AlertScriptsPath`, `ExternalScripts`, frontend et modules communautaires non pris en charge	Entièrement pris en charge (chemin des scripts, modules, intégrations)
Supervision ODBC	PostgreSQL uniquement (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 ajustables
Téléversement de certificat SAML	Non pris en charge	Pris en charge via l'interface utilisateur ou l'API
Rapports planifiés	Vous devez créer votre propre média Email (média script non pris en charge)	Prend en charge à la fois les médias script et Email prêts à l'emploi

Fonctionnalité / Aspect	Zabbix Cloud	Zabbix sur site
Interface hôte pour les contrôles actifs	Adresse IP émise par la plateforme (créée automatiquement)	Interfaces gérées par vous ; 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.

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 maintenir la cohérence, vous pouvez supprimer manuellement l'interface d'hôte après sa création ou l'ajuster.

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 Email.

2. Assignez le média e-mail à votre utilisateur dans *Paramètres utilisateur* > *Médias*.
3. Sélectionnez ce type de média lors de la configuration des rapports planifiés.

Centre de développement

Centre de développement

Commencez à développer des extensions personnalisées pour Zabbix grâce à des conseils sur les modules, les widgets et les plugins.

Développer sur Zabbix

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.

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.

Plugins

Présentation Les plugins chargeables personnalisés étendent les fonctionnalités de l'agent Zabbix 2. Ils sont compilés séparément, mais utilisent un package partagé avec l'agent Zabbix 2.

Chaque plugin est un package *Go* qui définit la structure et implémente une ou plusieurs interfaces de plugin (*Exporter*, *Configurator*, *Runner*).

Aller à :

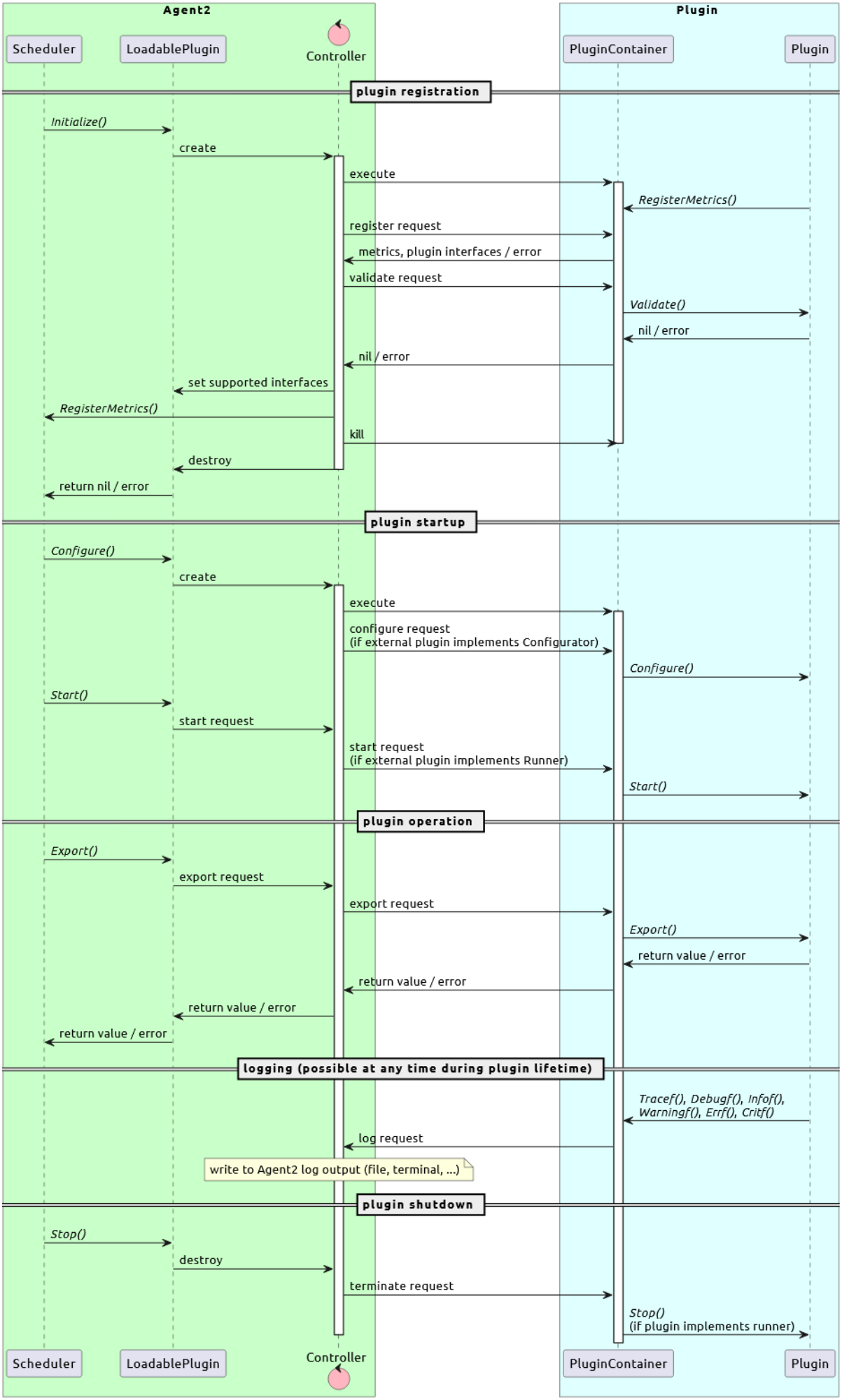
- Écrivez votre premier plugin
- Interfaces de plugin

Voir aussi :

- Exemple de plugin pour l'agent Zabbix 2

Diagramme de connexion L'agent Zabbix 2 se connecte de manière bidirectionnelle aux plugins à l'aide de sockets UNIX sous Linux et de Named Pipes sous Windows.

Le diagramme de connexion ci-dessous illustre le processus de communication entre l'agent Zabbix 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

What you'll create This is ste-by-step tutorial for creating a simple loadable plugin named **MyIP**. This plugin will implement a single item key (`myip`) returning the external IP address of the Zabbix agent host.

Step 1: Setup

1. A plugin is a standard Go module. Start by initializing the `go.mod` file in the plugin directory to track plugin dependencies:

```
cd path/to/plugins/myip # Switch to your plugin directory
go mod init myip
```

2. Install the mandatory dependency Zabbix Go SDK (`golang.zabbix.com/sdk`):

```
go get golang.zabbix.com/sdk@$LATEST_COMMIT_HASH
```

Replace `$LATEST_COMMIT_HASH` with the latest `HEAD` commit hash from the `golang.zabbix.com/sdk` repository in the appropriate release branch. For example:

```
go get golang.zabbix.com/sdk@af85407
```

Note that `golang.zabbix.com/sdk` versioning is currently not supported, but this may change in the future.

Additional dependencies can be installed as needed using `go get`.

3. Create an empty `main.go` file for the plugin source code:

```
touch main.go
```

Now the initial setup is complete, and the plugin is ready for development.

Step 2: Plugin structure The `golang.zabbix.com/sdk` module, installed in the previous step, provides the necessary framework for plugin development and ensures all plugins have a consistent structure.

1. Set up basic execution flow.

Start by defining the main execution flow of the plugin. Add the following code to `main.go`:

```
package main

func main() {
    err := run()
    if err != nil {
        panic(err)
    }
}

func run() error {
    return nil
}
```

This establishes the basic execution flow for the plugin. The run function will later contain the core logic of the plugin.

2. Explore the plugin interfaces.

A Zabbix agent 2 plugin shall be represented by a struct that implements interfaces from the `golang.zabbix.com/sdk/plugin` package:

- *Accessor* - defines essential methods all plugins must implement, such as setting the plugin name and handling item key timeouts.
- One or more of the following functional plugin interfaces:
  - *Exporter* - performs a poll and returns a value (or values), nothing, or an error; often used alongside the *Collector* interface.
  - *Collector* - manages the periodic collection of data.
  - *Runner* - defines plugin startup and shutdown procedures.
  - *Watcher* - allows to implement independent metric polling, bypassing the agent's internal scheduler; useful for trap-based or event-driven monitoring.
  - *Configurator* - defines how the plugin reads and applies its configuration settings.

You can either implement these interfaces yourself or use the default ones provided by the Zabbix Go SDK, modifying them as needed. This tutorial uses the default implementations.

3. Create the plugin struct.

Now, import the *plugin* package and create a `myIP` struct that embeds the `plugin.Base` struct:

```
import "golang.zabbix.com/sdk/plugin"

type myIP struct {
    plugin.Base
}
```

The myIP struct currently satisfies the Accessor interface. A method for implementing one of the functional plugin interfaces, the `Exporter`, will be added later in the tutorial.

Step 3: Define item keys Your plugin needs the item keys to collect data and provide it to Zabbix server or proxy.

1. Import *errs* package for error handling:

```
import "golang.zabbix.com/sdk/errs"
```

2. Register item keys using the `plugin.RegisterMetrics()` function within the `run()` function:

```
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
}
```

To register several item keys, repeat the parameters *metric name* and *description* for each metric. For example:

```
plugin.RegisterMetrics(&impl, "Myip", "metric.one", "Metric one description.", "metric.two", "Metric two d
```

Step 4: Set up the handler The handler facilitates communication between the agent and the plugin.

1. Import the *container* package:

```
import "golang.zabbix.com/sdk/plugin/container"
```

2. Indise the `run()` function add code to create and set up a 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
}
```

Step 5: Implement data collection Data collection is done via the Exporter interface, which describes the `Export` method:

```
func Export(
  key string,             // The item key to collect.
  params []string,        // Arguments passed to the item key (`myip[arg1, arg2]`).
  context ContextProvider // Metadata about the item key data collection.
) (any, error)
```

1. Import the required packages for HTTP requests and response reading:

```
import (
    "io"
    "net/http"
)
```

2. Implement the `Export` method for the `myIP` struct:

```
func (p *myIP) Export(
    key string, params []string, context plugin.ContextProvider,
) (any, error) {
    // The plugin can use different data collection logic based on the `key` parameter.
    // This implementation only verifies that the provided `key` is supported.
    if key != "myip" {
        return nil, errs.Errorf("unknown item key %q", key)
    }

    // The log will get forwarded to the agent 2 log.
    p.Infof(
        "received request to handle %q key with %d parameters",
        key,
        len(params),
    )

    // Collect the data and return it.

    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
}
```

Step 6: Build and configure the plugin

1. To build the plugin, run:

```
go mod tidy
go build
```

This should create an executable `myip` in the current directory.

2. Configure Zabbix agent 2 to use the plugin:

```
echo "Plugins.MyIP.System.Path=$PATH_TO_THE_MYIP_PLUGIN_EXECUTABLE" > /etc/zabbix_agent2.d/plugins.d/myip.
```

Replace `$PATH_TO_THE_MYIP_PLUGIN_EXECUTABLE` with the path to the `myip` created in step 5.

The plugin name in the configuration parameter name (*MyIP* in this tutorial) must match the plugin name defined in the *plugin.RegisterMetrics()* function.

3. To test the plugin and its `myip` item, run:

```
zabbix_agent2 -c /etc/zabbix_agent2.conf -t myip
```

The output should contain an external IP address of your host and look similar to this:

```
myip                                          [s|192.0.2.0]
```

With that, you have created a simple loadable plugin for Zabbix agent 2. Congrats!

```go
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{}

    // 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 from the agent is received.
    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) {
    // The plugin can use different data collection logic based on the `key` parameter.
    // This implementation only verifies that the provided `key` is supported.
    if key != "myip" {
        return nil, errs.Errorf("unknown item key %q", key)
    }

    // The log will get forwarded to the agent 2 log.
    p.Infof(
        "received request to handle %q key with %d parameters",
        key,
        len(params),
    )

    // Collect the data and return it.

    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
}
```

Complete source code

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

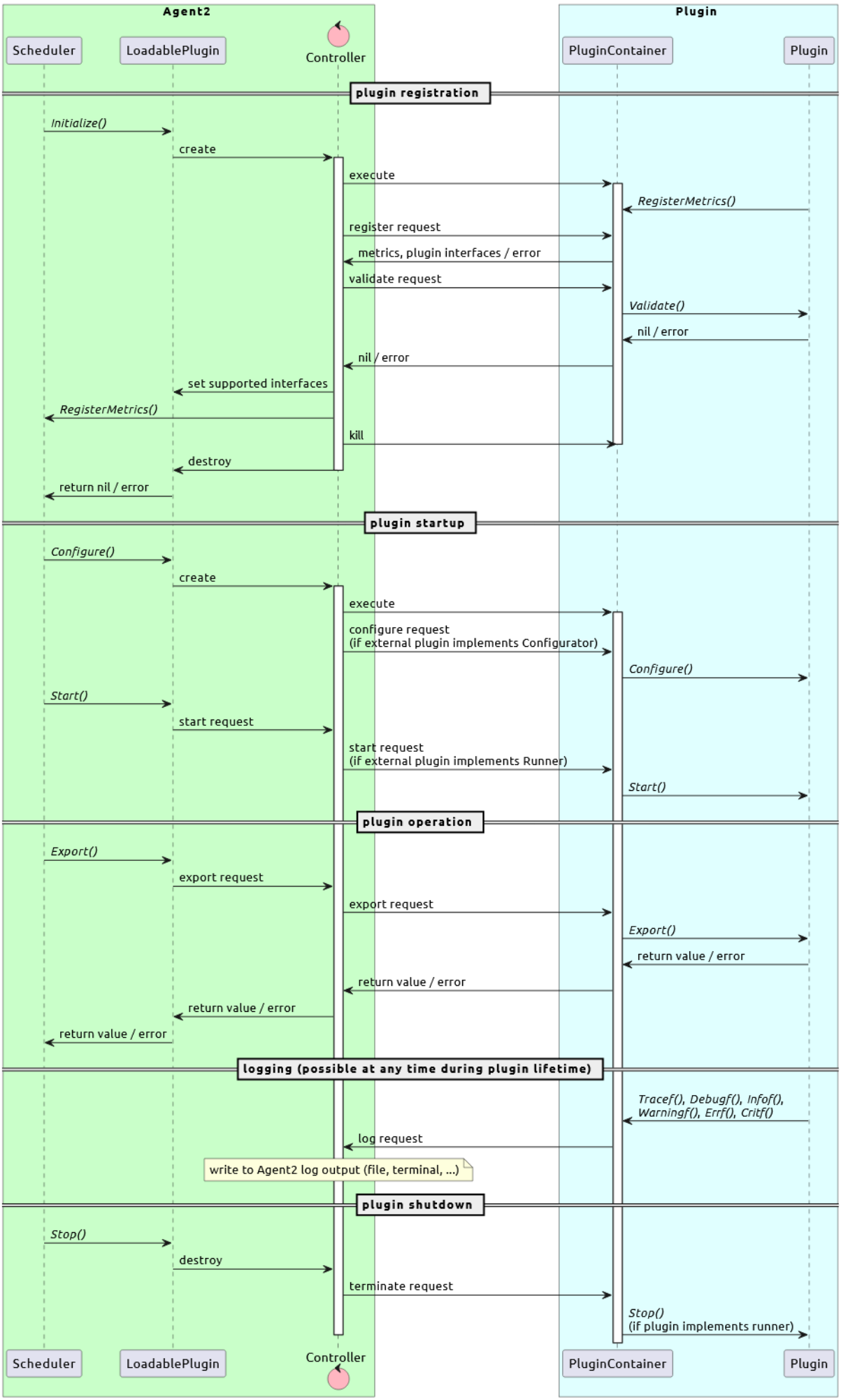
plugin est exclusif et aucune méthode ne peut être appelée si un plugin exécute déjà une tâche. En outre, il existe une limite de 100 appels *Export()* simultanés maximum par plugin, qui peut être réduite en fonction des exigences de chaque plugin.

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.

Connection diagram Zabbix agent 2 connects bidirectionally to the plugins using UNIX sockets on Linux and Named Pipes on Windows.

The connection diagram below illustrates the communication process between Zabbix agent 2 and a loadable plugin and the metrics collection process.


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 de `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='192.0.2.0', 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` que le `Getter` utilisera lors de la connexion à l'agent Zabbix :

```
agent = Getter(
    host='192.0.2.0',
    port=10050,
    source_ip='10.10.7.1'
)
```

Utilisation du délai d'expiration

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='192.0.2.0',
    port=10050,
    timeout=30
)
```

Utilisation du chiffrement

Le `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):
    # ...
    # Implémentation du wrapper TLS PSK pour le socket
    # ...

agent = Getter(
    host='192.0.2.0',
    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 l'arrivée de chacune d'elles. 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 `asyncio` de Python (vous devez d'abord installer les dépendances requises).
- Importez `AsyncGetter` au lieu de `Getter`.
- Écrivez votre code dans 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='192.0.2.0', 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 Zabbix ou de l'agent 2. Dans ce cas, vous n'avez pas besoin de spécifier les paramètres de connexion lors de la création d'une instance de `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 dans chaque cluster. Si `ServerActive` n'est pas défini, l'adresse de `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

Ecrire un module Un exemple de processus d'écriture de module comprend les étapes suivantes (le cas échéant, 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 de module.
4. Créez le dossier *actions* et définissez une ou plusieurs actions de 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'actifs requis dans le 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 être situé dans le répertoire principal du module (Par exemple, *zabbix/ui/modules/module_name/manifest.json*).

Au strict minimum, manifest.json doit spécifier ces champs :

```
{
    "manifest_version": 2.0,
    "id": "mon_adresse_ip",
    "name": "Mon adresse IP",
    "namespace": "MonAdresseIP",
    "version": "1.0"
}
```

Paramètres pris en charge dans **manifest.json** (appuyez sur le nom du paramètre pour une description détaillée) :

Paramètre	Description	Obligatoire
manifest_version	Version manifeste du module.	Oui
id	ID de module unique.	

Paramètre	Description	Obligatoire
name	Nom du module qui sera affiché dans la section Administration.	
namespace	Espace de noms PHP pour les classes de modules.	
version	Version du module.	
type	Type du module. Pour le widget doit être défini sur *widget*	Oui pour les widgets, sinon non
widget	Configuration des widgets. Utilisé uniquement pour les widgets.	
actions	Actions pour s'inscrire au module.	
assets	Styles CSS et fichiers JavaScript à inclure.	Non
author	Auteur du module.	
config	Valeurs par défaut pour les options de module personnalisées.	
description	Description du module.	
url	Un 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.

Type : Chaîne

Exemple:

```
"url": "http://exemple.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;
}
```

```php
/**
 * 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 doAction(): 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 frontend 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
<?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 "thème" pour définir un style différent pour un thème frontal spécifique.

Thèmes disponibles et leurs valeurs d'attribut :

- **Bleu** - [theme='blue-theme']
- **Sombre** - [theme='dark-theme']
- **Lumière 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 appuyez sur *Analyser le répertoire*.



Localisez votre module dans la liste et activez-le.

Pour activer un module, appuyez sur le lien hypertexte *Désactivé* - l'état du module passera à *Activé*.

Appuyez sur le nom du module pour afficher des informations supplémentaires sur le module, telles que l'auteur, la version ou une brève description (si définie 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 spécifier la logique de présentation des champs définis dans la classe *WidgetForm*, déterminer leur apparence et leur comportement lors du rendu dans la vue de configuration.

La classe *CWidgetFormView* prend en charge les méthodes suivantes :

- *addField()* - reçoit une instance de la classe *CWidgetFieldView* en paramètre ; chaques classes *CWidgetField* , possède une classe *CWidgetFieldView* à utiliser dans la vue de configuration du widget.
- *includeJsFile()* - permet d'ajouter un fichier JavaScript à la vue de configuration du widget.

- *addJavaScript()* - permet d'ajouter du JavaScript en ligne qui sera exécuté dès que la vue de configuration du widget sera chargée.

La classe *CWidgetFormView* doit être située dans le répertoire *views*.

views/widget.edit.php example

```
<?php

/**
 * My custom widget form view.
 *
 * @var CView $this
 * @var array $data
 */

use Zabbix\Widgets\Fields\CWidgetFieldGraphDataSet;

(new CWidgetFormView($data))
    ->addField(
        new CWidgetFieldMultiSelectItemView($data['fields']['itemid'], $data['captions']['items']['itemid'
    )
    ->addField(
        new CWidgetFieldTextBoxView($data['fields']['description']),
        'js-advanced-configuration'
    )
    ->addField(
        new CWidgetFieldColorView($data['fields']['chart_color']),
        'js-advanced-configuration'
    )
    ->includeJsFile('widget.edit.js.php')
    ->addJavaScript('my_widget_form.init('.json_encode([
        'color_palette' => CWidgetFieldGraphDataSet::DEFAULT_COLOR_PALETTE
    ], JSON_THROW_ON_ERROR).');')
    ->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 toutes les classes de champs de formulaire (*CWidgetFieldCheckBox*, *CWidgetFieldTextArea*, *CWidgetFieldRadioButtonList*, etc.) sont héritées. Les classes étendant *CWidgetField* sont responsables de la réception, de l'enregistrement et de la validation des valeurs de configuration du widget.

Les classes *CWidgetField* suivantes sont disponibles.

Classe CWidgetField	Type de champ de base de données	Description
CWidgetFieldCheckBox	int32	case à cocher unique.
CWidgetFieldCheckBoxList	tableau de 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.
CWidgetFieldHostPatternSelect	string	Champ Multiselect qui permet de sélectionner un ou plusieurs hôtes. Prend en charge la définition de modèles de nom d'hôte (tous les hôtes correspondants seront sélectionnés).
CWidgetFieldIntegerBox	int32	Champ pour saisir un entier. Peut être utilisé pour configurer les 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.
CWidgetFieldMultiSelectAction	ID	Champ multisélection de sélection d'actions (dans la liste des actions définies dans *Alertes → Actions*).
CWidgetFieldMultiSelectGraph	ID	Champ multisélection pour sélectionner des graphiques personnalisés.
CWidgetFieldMultiSelectGraphPrototype	ID	Champ multisélection pour sélectionner des prototypes de graphiques personnalisés.
CWidgetFieldMultiSelectGroup	ID	Champ multisélection pour la sélection des groupes d'hôtes.
CWidgetFieldMultiSelectHost	ID	Champ multisélection pour la sélection des hôtes.
CWidgetFieldMultiSelectItem	ID	Champ multisélection pour la sélection d'éléments.
CWidgetFieldMultiSelectItemPattern	ID	Champ multisélection pour sélectionner des modèles d'éléments.
CWidgetFieldMultiSelectItemPrototype	ID	Champ multisélection pour la sélection des prototypes d'éléments.
CWidgetFieldMultiSelectMap	ID	Champ multisélection pour la sélection de cartes.
CWidgetFieldMultiSelectMediaType	ID	Champ multisélection pour sélectionner les types de médias.
CWidgetFieldMultiSelectOverrideHost	ID	Champ multisélection permettant de sélectionner une source de données (tableau de bord ou autre widget) contenant un hôte pour lequel le widget peut afficher des données.
CWidgetFieldMultiSelectService	ID	Champ multisélection pour la sélection des services.
CWidgetFieldMultiSelectSla	ID	Champ multisélection pour la sélection des SLA.
CWidgetFieldMultiSelectUser	ID	Champ multisélection pour la sélection des utilisateurs.
CWidgetFieldNumericBox	string	Champ pour saisir un nombre flottant.
CWidgetFieldRadioButtonList	int32	Groupe de boîtiers radio composé d'un ou plusieurs boîtiers radio.
CWidgetFieldRangeControl	int32	Curseur pour sélectionner une valeur de type entier.
CWidgetFieldReference	string	Crée un identifiant unique pour ce widget sur le tableau de bord. Il est utilisé pour référencer ce widget à partir d'autres widgets.
CWidgetFieldSelect	int32	Boîte de sélection déroulante.
CWidgetFieldSeverities	tableau de préréglages int32	*CWidgetFieldCheckBoxList* avec sévérités de déclenchement.
CWidgetFieldTags	tableau de (string, int32, string)	Permet de configurer une ou plusieurs lignes de filtre de balises.
CWidgetFieldTextArea	string	Zone de texte pour saisir du texte multiligne.
CWidgetFieldTextBox	string	Zone de texte pour saisir du texte sur une seule ligne.
CWidgetFieldTimePeriod	tableau de chaînes	Champ de sélection de période.
CWidgetFieldTimeZone	string	Déroulante avec les fuseaux horaires.
CWidgetFieldThresholds	tableau de (string, string)	Permet de configurer des paires de couleurs et de nombres.
CWidgetFieldUrl	string	Zone de texte permettant de saisir des URL.

Les classes *CWidgetField* suivantes ont été créées pour des widgets particuliers. Ces classes ont des cas d'utilisation très spécifiques, mais elles peuvent également être réutilisées si nécessaire.

Classe CWidgetField	Type de champ de base de données	Description
CWidgetFieldColumnsList	tableau de (plusieurs mélanges)	Pour le widget *Top hosts*. Créez un tableau avec des colonnes personnalisées de types autorisés.
CWidgetFieldNavTree	string	Pour le widget *Arbre de navigation cartographique*. Remplace la vue widget en mode édition par l'arborescence de sélection de carte.

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 *CControllerDashboardWidgetView*.

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

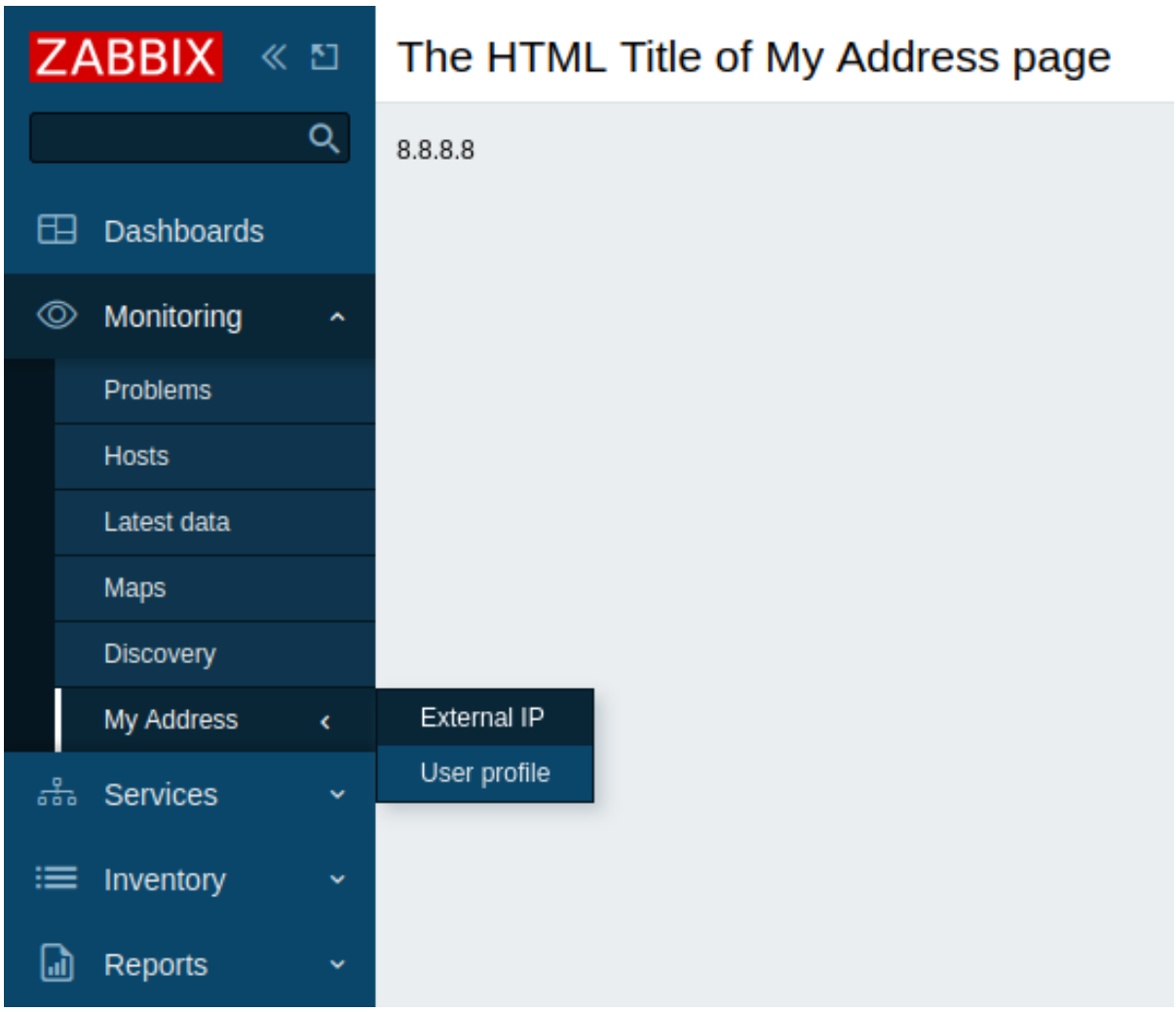
Cette section contient des tutoriels pratiques étape par étape pour illustrer comment créer un module personnalisé et un widget dans Zabbix.

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 construire

Au cours de ce tutoriel, vous allez d'abord créer un module frontend qui ajoute une nouvelle section de menu *Mon adresse* puis convertit en un module frontal plus avancé qui envoie une requête HTTP à *https://api.seeip.org* et affiche la réponse - l'adresse IP de votre ordinateur - sur une nouvelle page dans la section de menu *Mon adresse* nouvellement créée. Voici à quoi ressemblera le module terminé :



Partie I - Nouvelle section de menu

Ajouter un module vide au frontend Zabbix

1. Créez un répertoire *MyAddress* dans le répertoire *modules* de votre installation du frontend 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).

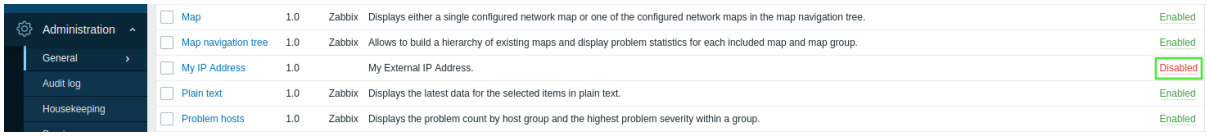
ui/modules/MyAddress/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "my-address",
    "name": "My IP Address",
    "version": "1.0",
    "namespace": "MyAddress",
    "description": "My External IP Address."
}
```

3. Dans le frontend Zabbix, accédez à 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 l'hyperlien "Disabled" pour changer le statut du module de "Disabled" à "Enabled" (si le module n'est pas répertorié, consultez la section résolution des problèmes).



Le module est maintenant enregistré dans le frontend. Cependant, il n'est pas encore visible, car vous devez encore définir la fonctionnalité du module. Une fois que vous aurez ajouté du contenu au répertoire du module, vous verrez immédiatement les changements dans le frontend Zabbix après avoir actualisé la page.

Créer une section de menu

1. Créez un fichier *Module.php* dans le répertoire *MonAdresse*.

Ce fichier implémente une nouvelle classe *Module* qui étend la classe *CModule* par défaut. La classe *Module* insérera une nouvelle section de menu *My Address* dans le menu principal.

La méthode *setAction()* spécifie une action à exécuter en cliquant 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 didacticiel, vous apprendrez à créer une action personnalisée.

ui/modules/MonAdresse/Module.php

```
<?php

namespace Modules\MonAdresse ;

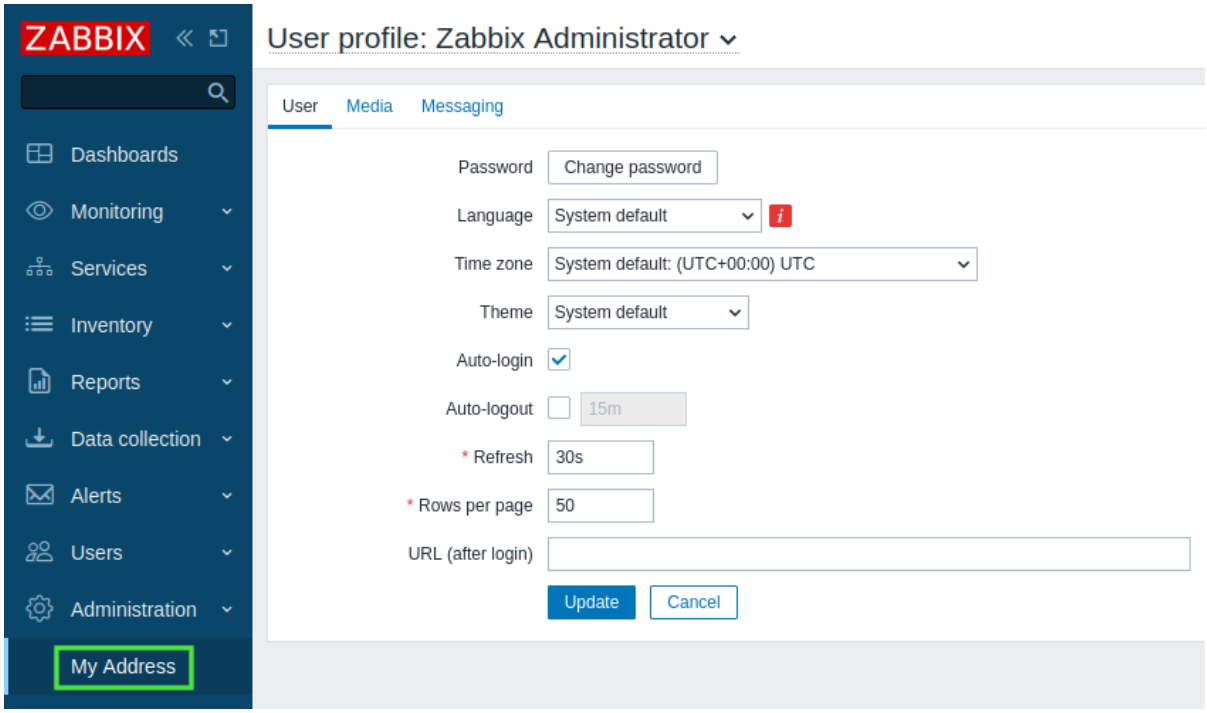
use Zabbix\Core\CModule,
    APP,
    CMenuItem ;

class Module extends CModule {

    public function init(): void {
        APP::Component()->get('menu.main')
            -->add((new CMenuItem(_('Mon adresse')))
            ->setAction('userprofile.edit'));
    }
}
```

Note:
Vous pouvez remplacer `userprofile.edit'` par d'autres actions, par exemple,charts.view'(ouvre des graphiques personnalisés), ``problems.view'` (ouvre *Surveillance* → *Problèmes*) ou 'report .status' (ouvre le rapport *Informations système*).

3. Actualisez l'interface Zabbix. Il y a maintenant une nouvelle section *Mon adresse* en bas du menu principal de Zabbix. Cliquez sur *Mon adresse* pour ouvrir la page *Profil utilisateur*.



Partie II - Changement d'emplacement de la section de menu

Dans cette partie, vous allez déplacer la section du menu *Mon adresse* vers la section *Surveillance*. En conséquence, les utilisateurs pourront accéder à deux pages de sous-menu à partir de la section de menu *Surveillance → Mon adresse*.

1. Ouvrez et modifiez le fichier *Module.php*.

ui/modules/MonAdresse/Module.php

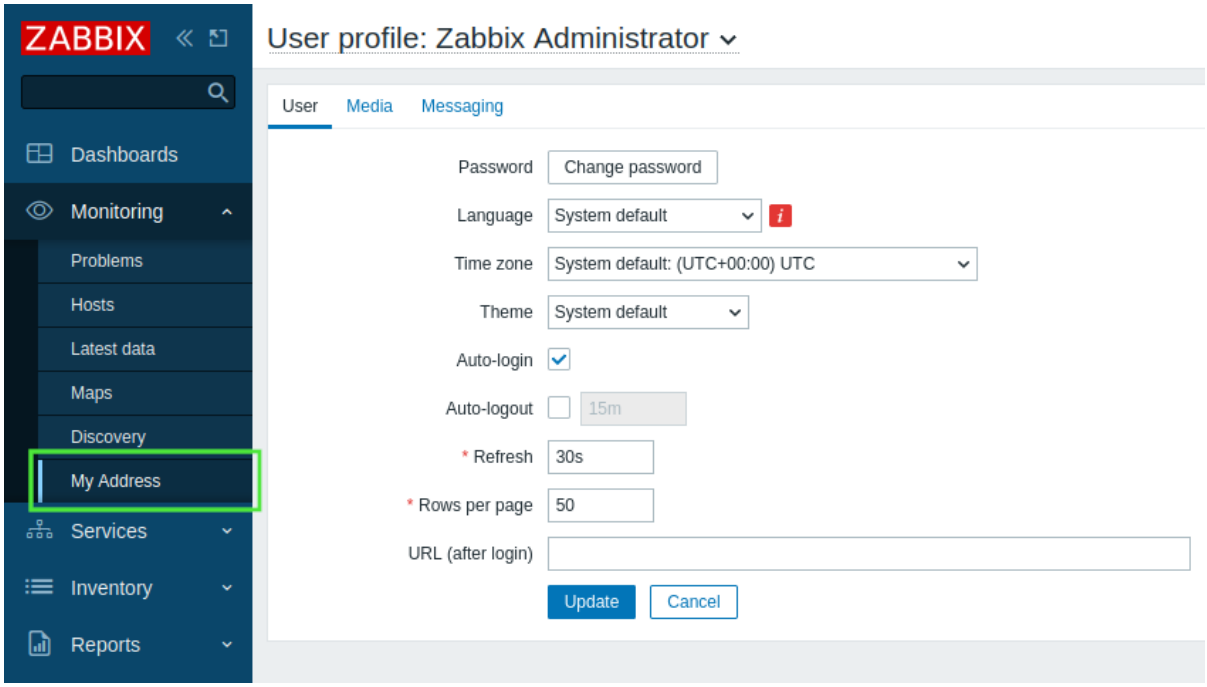
```
<?php

namespace Modules\MyAddress;
    APP,
    CMenuItem ;

class Module extends CModule {

    class Module extends CModule {
        APP::Component()->get('menu.main')
            ->findOrAdd(_('Surveillance'))
                ->getSous-menu()
                    ->insertAfter(_('Discovery'),
                        (new CMenuItem(_('My Address')))->setAction('userprofile.edit')
                    );
    }
}
```

2. Actualisez l'interface Zabbix. Développez la section du menu *Surveillance* et observez que la section *Mon adresse* se trouve désormais sous la section *Découverte*.



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*** s'occupe de 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()*. Le frontend 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 la définir spécifiquement. Le framework passe automatiquement le tableau associatif à la vue.

ui/modules/MyAddress/views/my.address.php

```
<?php

(new CHtmlPage())
->setTitle(_('The HTML Title of My Address Page'))
->addItem(new CDiv($data['my-ip']))
->show();
```

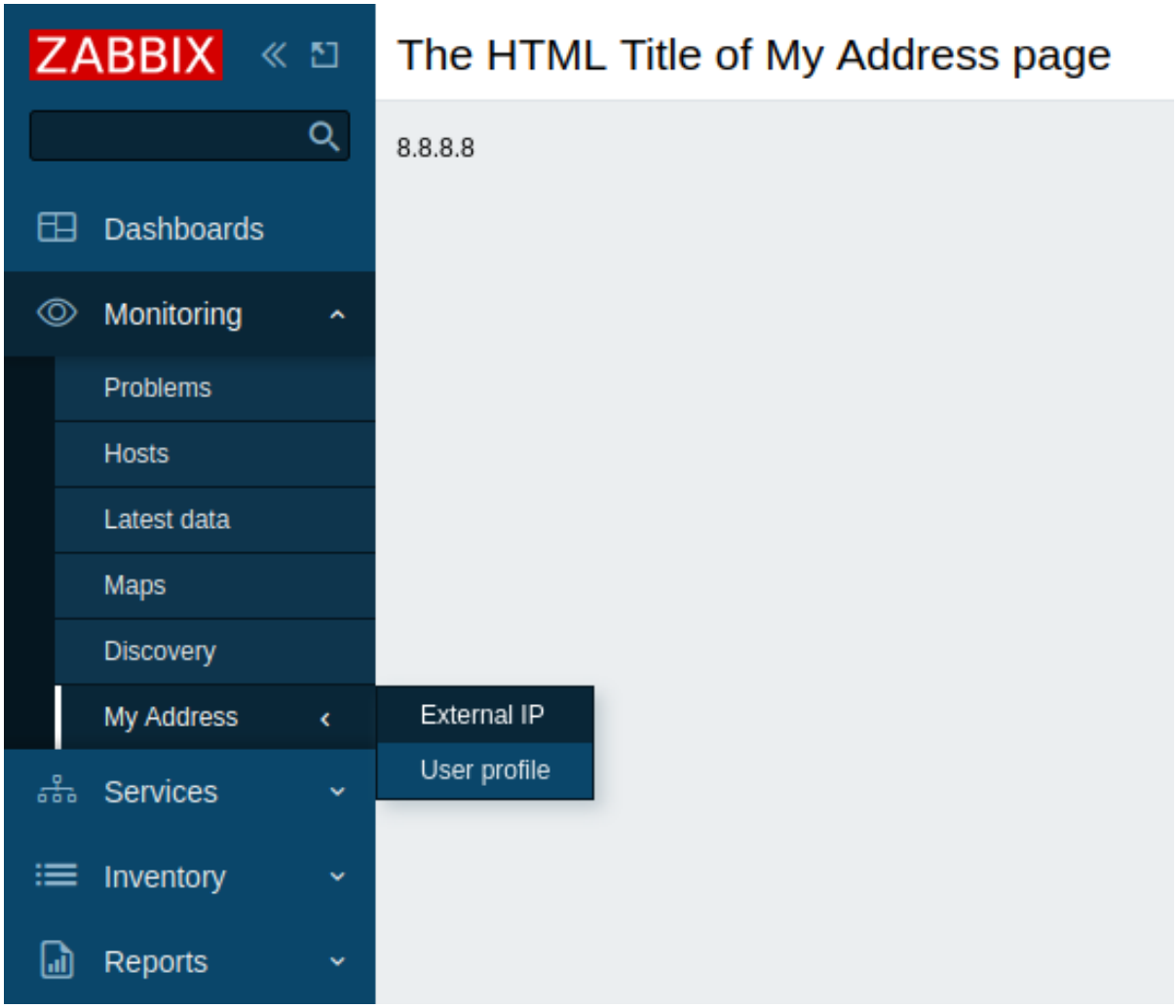
5. L'action du module doit être enregistrée dans le fichier *manifest.json*. Ouvrez *manifest.json* et ajoutez un nouvel objet `actions` qui contient :

- la clé d'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 pour la clé `class` de l'objet `my.address` ;
- le nom de la vue d'action (`my.address`) comme valeur pour 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"
}
}
}
```

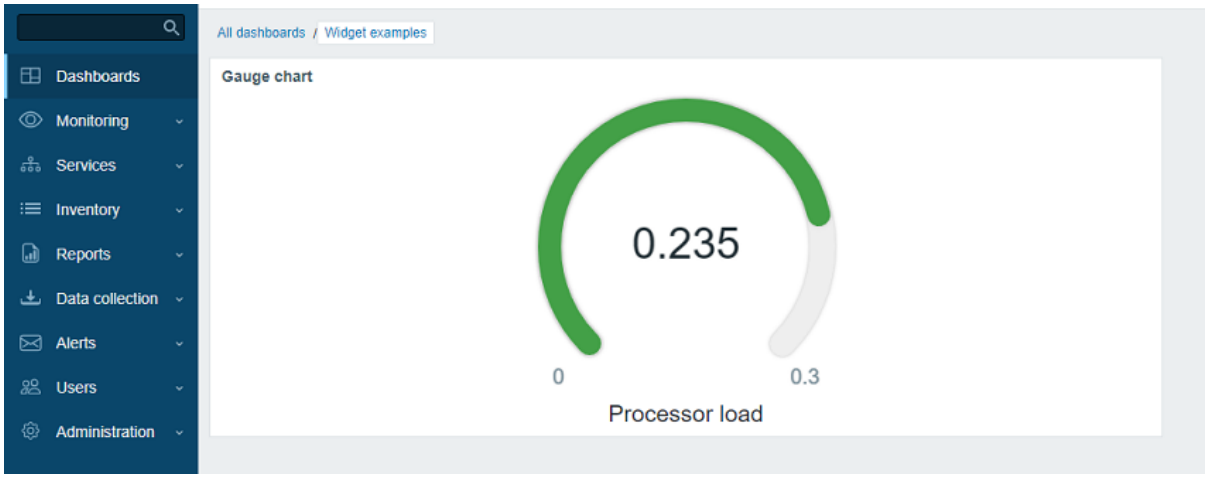
6. Actualisez l'interface Zabbix. Cliquez sur *Mon adresse → IP externe* pour voir l'adresse IP de votre ordinateur.



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 construirez Au cours de ce didacticiel, vous allez d'abord créer un basic "Hello, world!" widget, puis convertissez-le en un widget plus avancé qui affiche la valeur d'un élément sous forme de graphique à jauge. Voici à quoi ressemblera le widget fini :



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 au frontend Zabbix

1. Créez un répertoire *lesson_gauge_chart* dans le répertoire *modules* de votre installation du frontend 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 du frontend Zabbix (par exemple, *zabbix/ui/modules*). Le répertoire *zabbix/ui/widgets* est réservé aux widgets intégrés de Zabbix et est mis à jour avec l'interface utilisateur de 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"
}
```

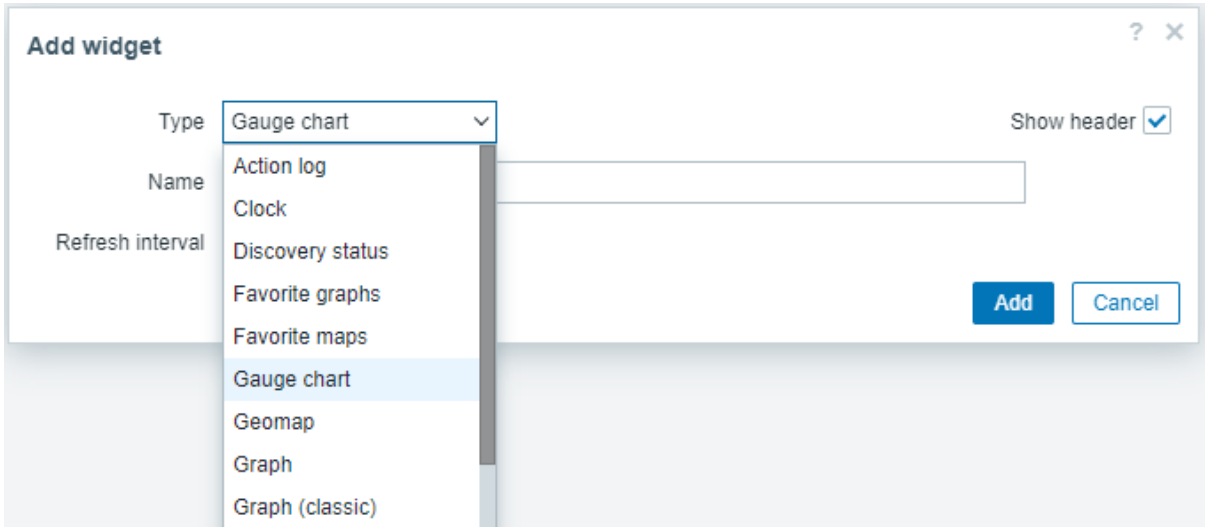
3. Dans le frontend Zabbix, accédez à la section *Administration → General → Modules* et cliquez sur le bouton *Scan directory*.



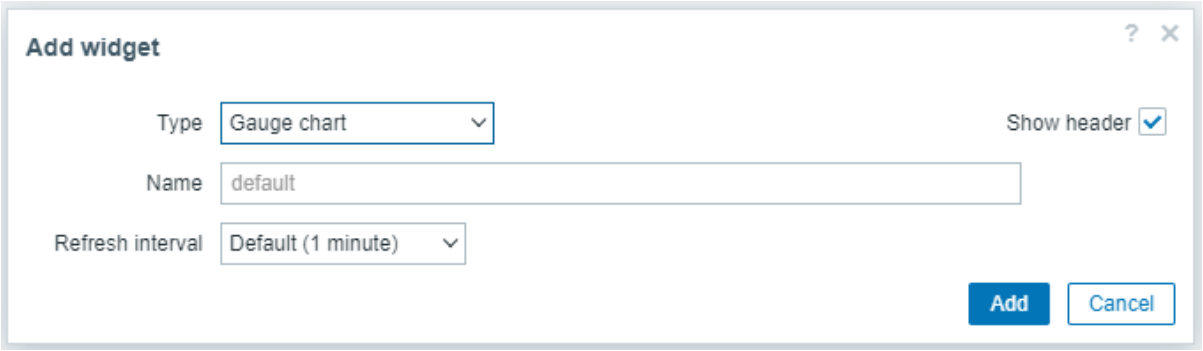
4. Repérez le nouveau module *Gauge chart* dans la liste et cliquez sur l'hyperlien "Disabled" pour changer le statut du module de "Disabled" à "Enabled" (si le module n'est pas répertorié, consultez la section dépannage).



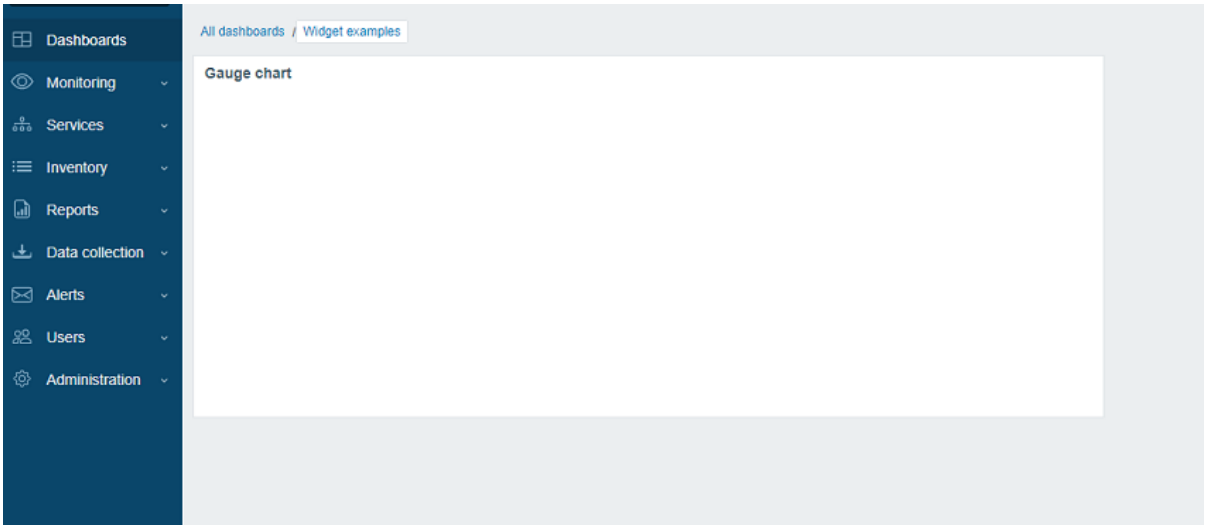
5. Ouvrez un tableau de bord, passez-le en mode édition et ajoutez un nouveau widget. Dans le champ *Type*, sélectionnez "Gauge chart".



6. À ce stade, la configuration du widget *Gauge chart* contient uniquement 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 de **vue** du widget doit être situé 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 porte 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*.

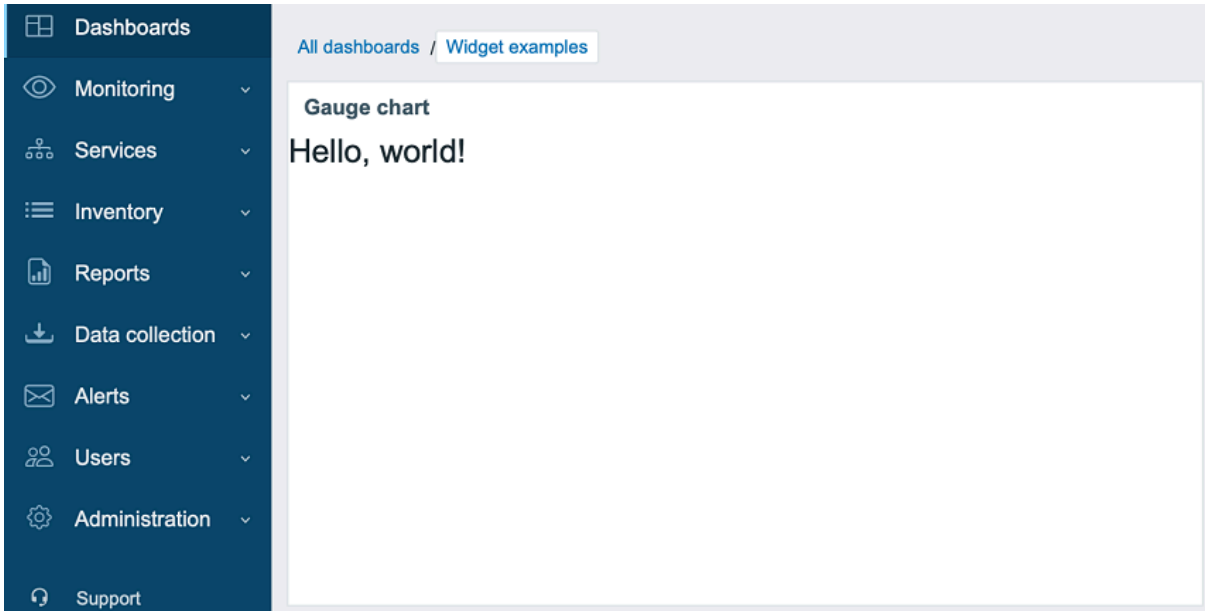
ui/modules/lesson_gauge_chart/views/widget.view.php

```
<?php

/**
 * Gauge chart widget view.
 *
 * @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 étendra *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*) correspondant à chaque type d'élément d'entrée convertit la valeur dans un format unique afin de la stocker dans la base de données.

> **Note:**
> Le fichier **form** du widget doit être placé 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 nom différent, 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 de texte standard dans lequel un 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 du widget *widget.edit.php* et ajoutez une vue pour le nouveau champ *Description*. Pour la classe de champ *CWidgetFieldTextBox*, la vue est *CWidgetFieldTextBoxView*.

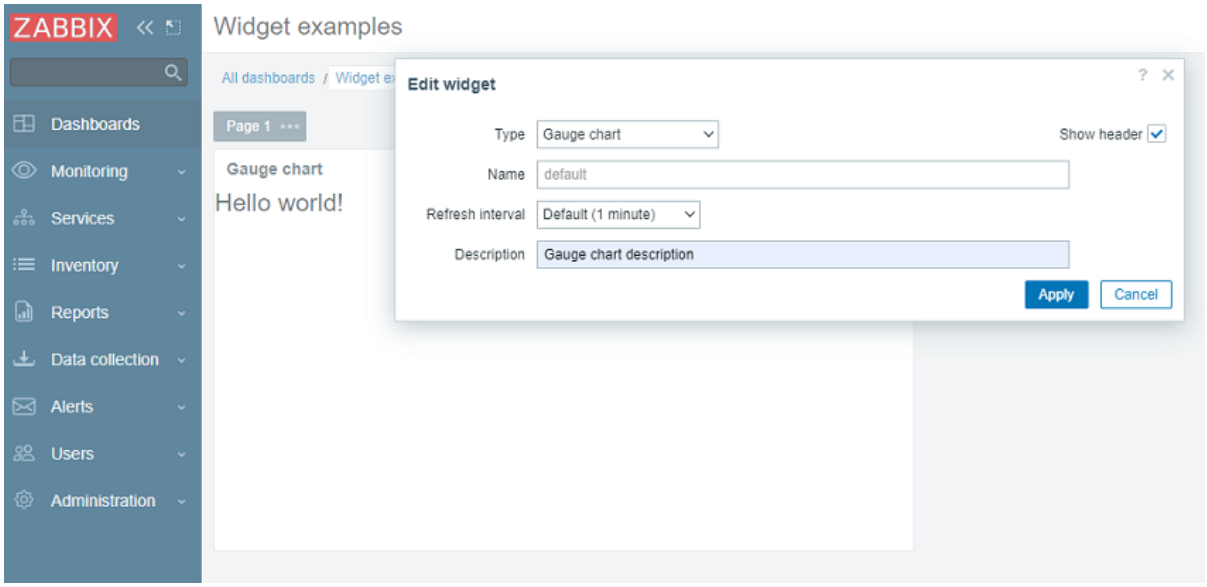
ui/modules/lesson_gauge_chart/views/widget.edit.php

```
<?php

/**
 * Gauge chart widget form view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetFormView($data))
    ->addField(
        new CWidgetFieldTextBoxView($data['fields']['description'])
    )
    ->show();
```

5. Accédez au tableau de bord et cliquez sur l'icône d'engrenage du widget pour ouvrir le formulaire de configuration du widget.
6. Le formulaire de configuration du widget contient maintenant un nouveau champ de texte *Description*. Saisissez une valeur quelconque, 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* étendra 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.view*.

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 de 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']*.

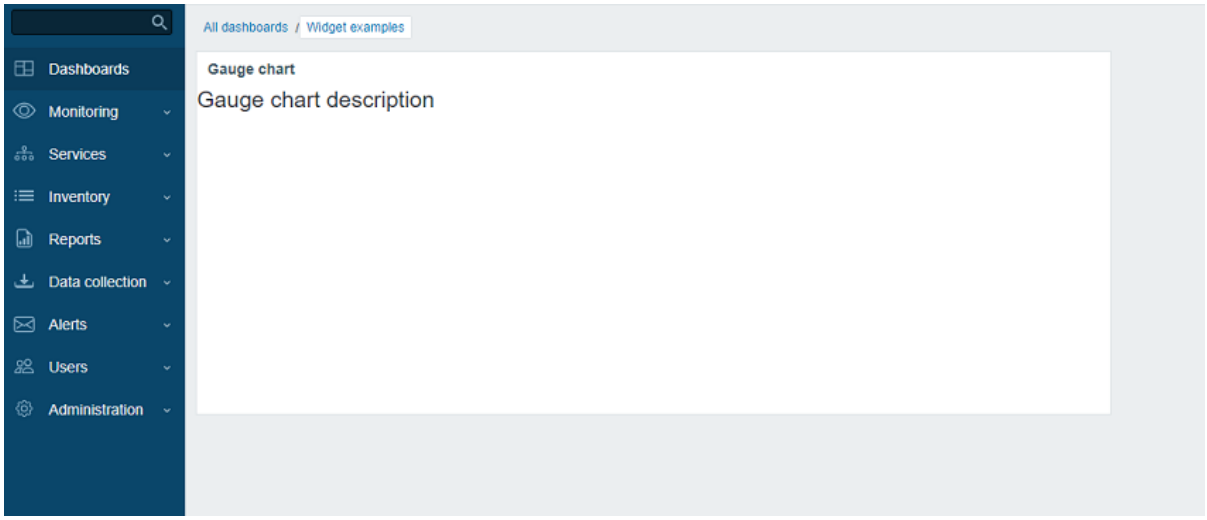
ui/modules/lesson_gauge_chart/views/widget.view.php

```
<?php

/**
 * Gauge chart widget view.
 *
 * @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 une valeur d'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', _('Item')))
                    ->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.

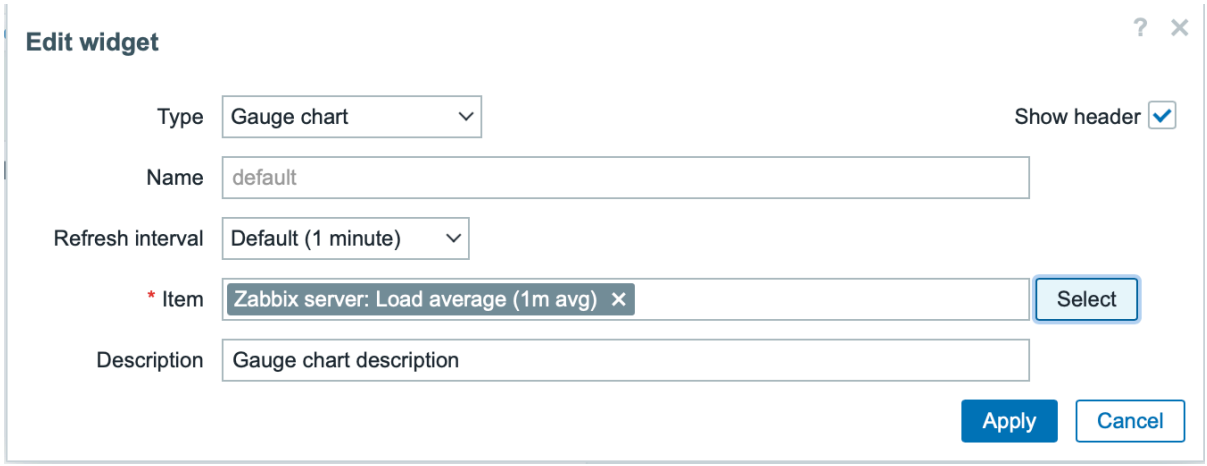
ui/modules/lesson_gauge_chart/views/widget.edit.php

```php
<?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 en forme 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 *Item*. Sélectionnez l'hôte « Zabbix server » et l'élément « Load average (1m avg) ».



5. 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.
6. Ouvrez et modifiez *actions/WidgetView.php*.

À partir de maintenant, l'ID de l'élément sera disponible dans le contrôleur du widget dans *$this->fields_values['itemid']*. La méthode de 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
<?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.

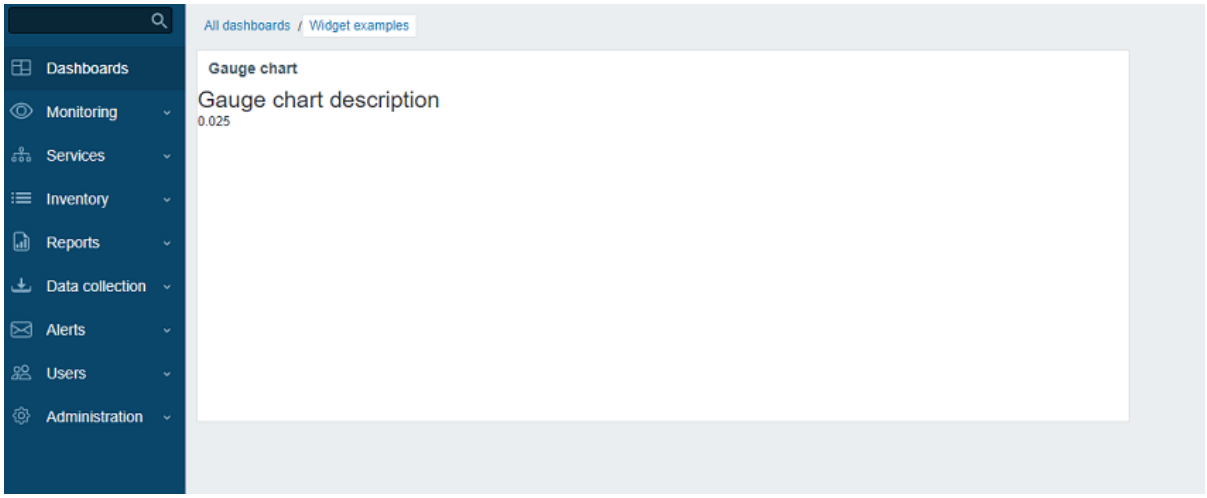
ui/modules/lesson_gauge_chart/views/widget.view.php

```
<?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, ainsi que le champ *Description* créé précédemment.

1. Créez un fichier *Widget.php* dans le répertoire principal du widget *lesson_gauge_chart* afin de créer une nouvelle classe *Widget*.

La classe *Widget* étendra la classe de base *CWidget* afin d'ajouter/remplacer les paramètres par défaut du widget (dans ce cas, les traductions). Le code JavaScript fourni ci-dessous affiche la chaîne « No data » en cas d'absence de données. 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 également recommandé de les spécifier 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* (liste de sélection), 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
};

/**
 * Formulaire du widget Gauge chart.
 */
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

/**
 * Vue du formulaire du widget Gauge chart.
 *
```

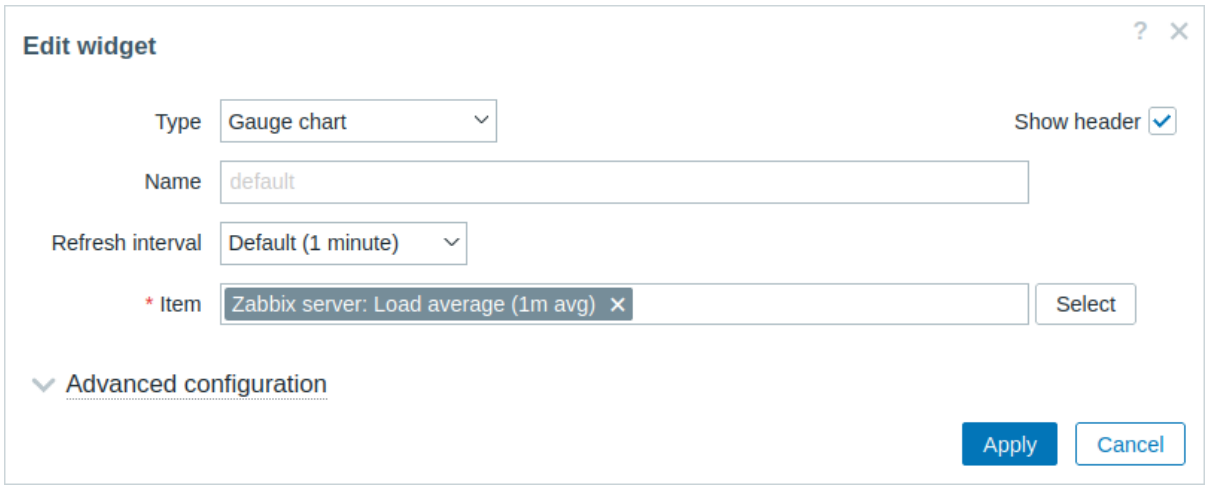
```
 * @var CView $this
 * @var array $data
 */

$lefty_units = new CWidgetFieldSelectView($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();
```

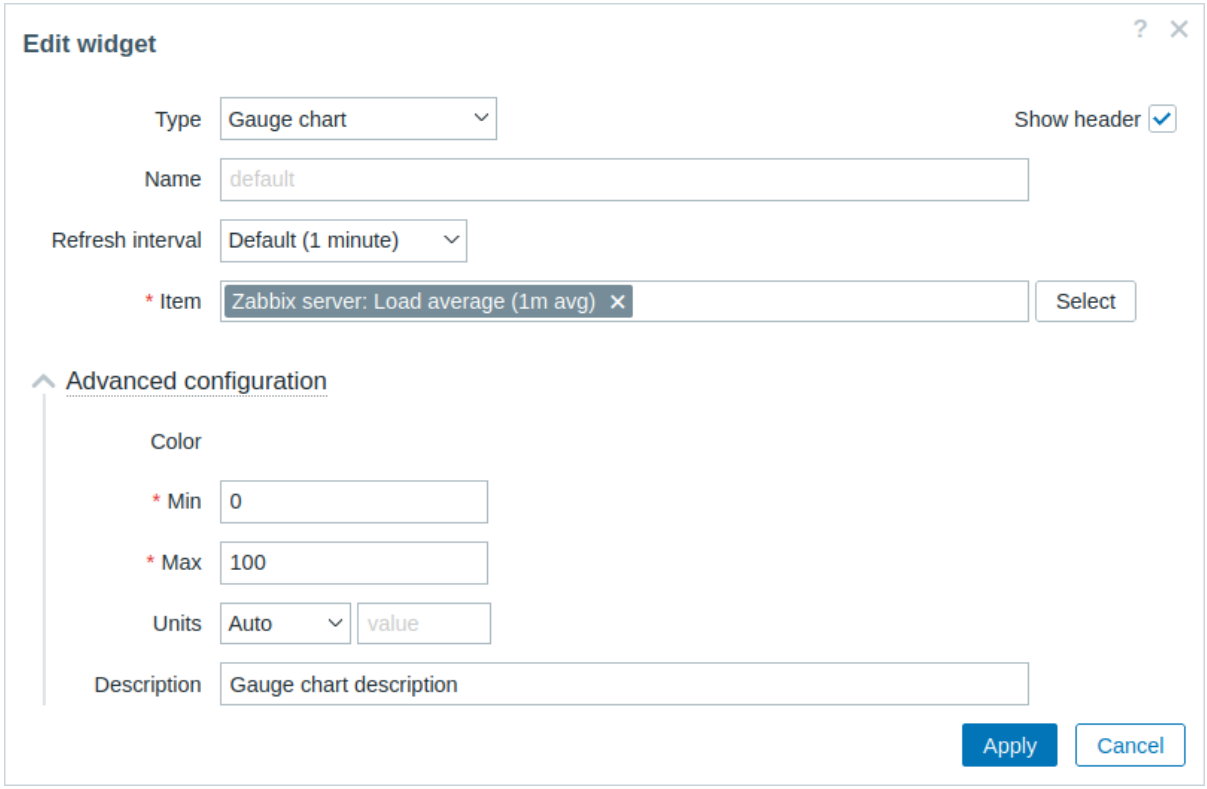
Note:
La méthode *addField()* de la classe *CWidgetFormView* prend une chaîne de classe CSS comme deuxième paramètre.

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 *Advanced configuration*.



5. Développez la section *Advanced configuration* pour voir les champs de configuration supplémentaires du widget. Notez que le champ *Color* ne dispose pas encore de sélecteur de couleur. Cela s'explique par le fait que le sélecteur de couleur doit

être initialisé avec JavaScript, qui sera ajouté dans la section suivante : *Add JavaScript to the widget*.



Ajouter du JavaScript au widget

Dans cette section, vous apprendrez à ajouter un graphique de jauge, créé à l'aide de JavaScript, qui indique si la dernière valeur est normale, trop élevée ou trop basse.

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* contenant le JavaScript à la vue de configuration. Pour ce faire, utilisez la méthode *includeJsFile()*. Pour ajouter du JavaScript en ligne, utilisez la méthode *addJavaScript()*.

ui/modules/lesson_gauge_chart/views/widget.edit.php

```
<?php

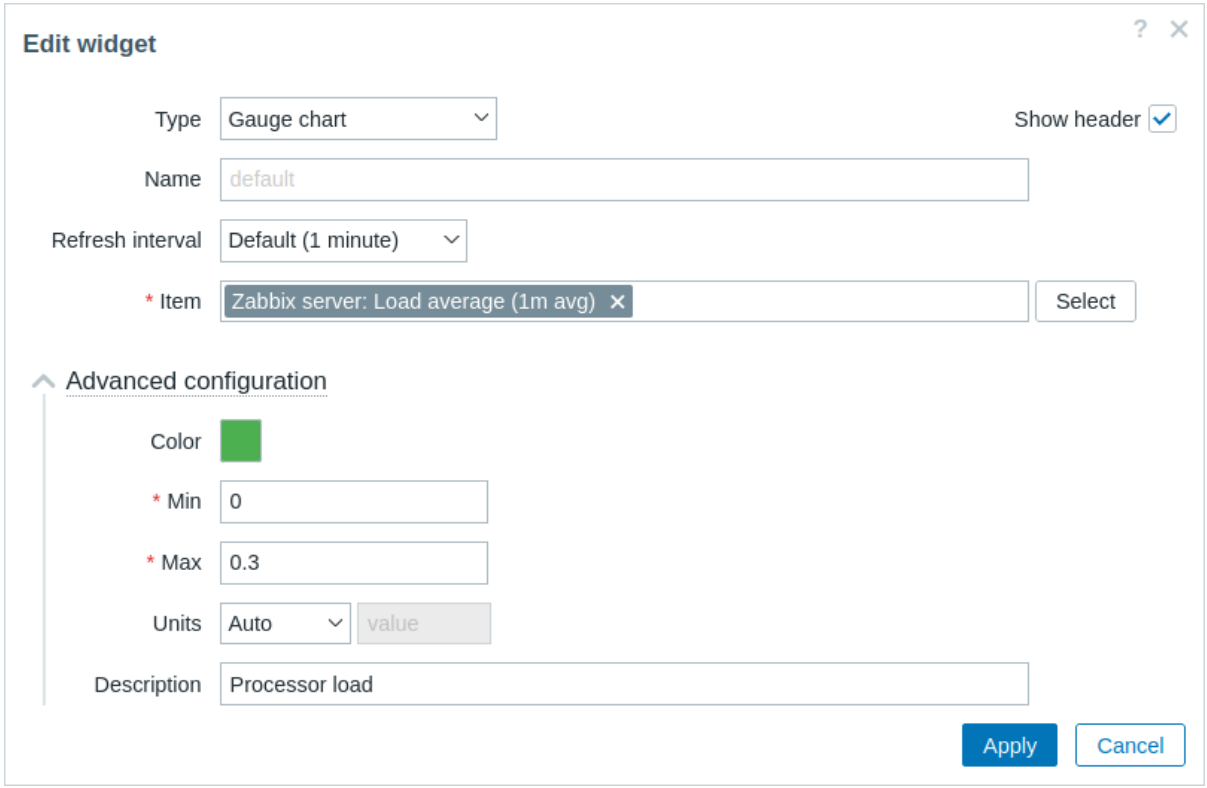
/**
 * Vue du formulaire du widget de graphique de jauge.
 *
 * @var CView $this
 * @var array $data
 */

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, puis cliquez sur l'icône en forme d'engrenage dans le widget pour ouvrir le formulaire de configuration du widget. Développez maintenant la section *Advanced configuration* pour voir le sélecteur de couleur initialisé. Renseignez les champs avec des valeurs et sélectionnez une couleur pour le graphique de 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 afin de permettre le passage de la configuration et de 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 de 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

/**
 * Vue du widget de graphique de jauge.
 *
 * @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 sera utilisé pour stocker JavaScript, CSS et éventuellement d'autres ressources, telles que 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 de widget JavaScript étendra la classe JavaScript de base de tous les widgets de tableau de bord, *CWidget*.

Le tableau de bord s'appuie sur une implémentation correcte d'un widget et communique au widget 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 chargée de définir l'état initial du widget (voir la méthode *onInitialize()*) ;
- l'affichage du contenu du widget (c'est-à-dire le dessin du graphique de 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 de jauge, l'implémentation par défaut du comportement du widget sera utilisée. Pour en savoir plus sur les méthodes JavaScript de la classe *CWidget*, voir : 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 de 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 de la section *assets/js* ;
- le nom de la classe (*WidgetLessonGaugeChart*) au paramètre *js_class* de 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 afin de rendre vos Widgets plus attrayant.

1. Pour les styles de Widgets, 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 a 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;
}
```

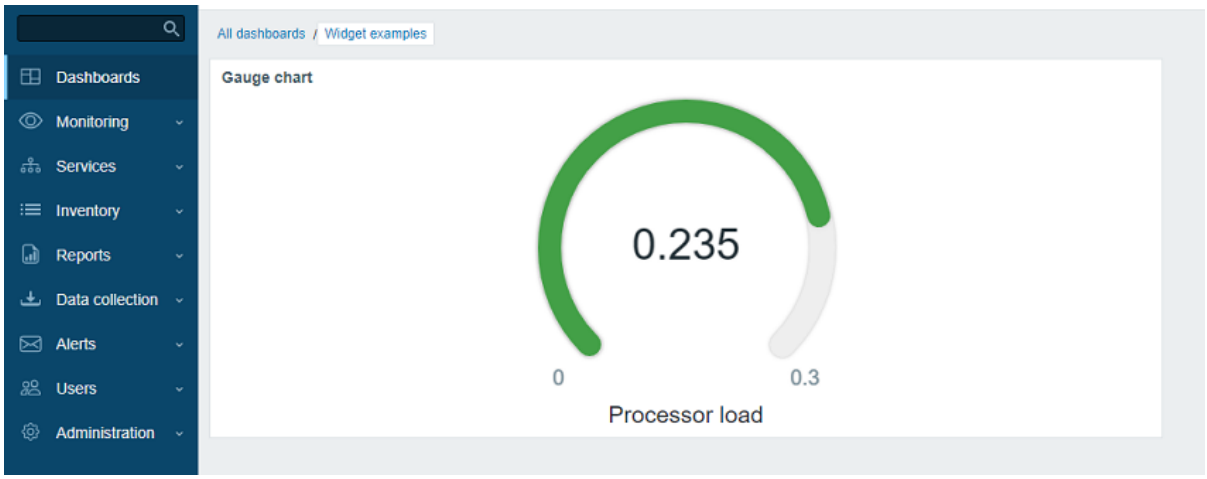
3. Ouvrir *manifest.json* et ajoutez le nom du fichier CSS (*widget.css*) au tableau dans la section *assets/css*. Cela permettra aux styles CSS définis dans *widget.css* de se charger avec la page du tableau de bord.

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 SIA",
    "actions": {
        "widget.lesson_gauge_chart.view": {
            "class": "WidgetView"
        }
    },
    "widget": {
        "js_class": "WidgetLessonGaugeChart"
    },
    "assets": {
        "css": ["widget.css"],
        "js": ["class.widget.js"]
    }
}
```

4. Rafraichir la page du tableau de bord afin de 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

Cette page répertorie toutes les modifications (le cas échéant) apportées au développement d'extensions Zabbix personnalisées.

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

Section : Commandes de maintenance (8)
Mise à jour : 2019-01-29
Index Return to Main Contents

NOM

zabbix_agent2 - Zabbix agent 2

RÉSUMÉ

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 les valeurs par défaut génériques sont utilisées, soit des valeurs par défaut spécifiques aux tests sont fournies. Ces valeurs par défaut sont indiquées 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 Zabbix ou zabbix_get lors de l'interrogation d'un démon d'agent en cours d'exécution, car les autorisations 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 d'autorisation 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 Zabbix agent 2 (s'il n'a pas été modifié lors de la compilation).

VOIR ÉGALEMENT

Documentation https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(8), **zabbix_js**(8), **zabbix_proxy**(8), **zabbix_sender**(8), **zabbix_server**(8)

Index

NOM

SYNOPSIS

DESCRIPTION

OPTIONS

FICHIERS

VOIR AUSSI

AUTEUR

Ce document a été créé le : 14:07:57 GMT, 22 novembre 2021

zabbix_agentd

Section : Commandes de maintenance (8)
Mise à jour : 2019-01-29
Index Retour au sommaire principal

NOM

zabbix_agentd - Démon de l'agent Zabbix

RÉSUMÉ

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 pour la surveillance de divers paramètres du serveur.

OPTIONS

-c, **--config** *config-file*
Utiliser le *config-file* 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 selon *runtime-option*.

Options de contrôle d'exécution userparameter_reload
Recharger les paramètres utilisateur à partir du 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 de contrôle du niveau de journalisation *type-de-processus*
Tous les processus du type spécifié (contrôles actifs, collecteur, listener)

type-de-processus,N
Type de processus et numéro (par ex., listener,3)

pid
Identifiant du processus, jusqu'à 65535.
Pour des valeurs plus grandes, spécifiez la cible comme « type-de-processus,N »

-p, **--print**
Afficher les éléments connus et quitter.
Pour chaque élément, soit les valeurs par défaut génériques sont utilisées, soit des valeurs par défaut spécifiques pour les tests sont fournies.
Ces valeurs par défaut sont indiquées 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 Zabbix ou zabbix_get lors de l'interrogation d'un démon d'agent en cours d'exécution, car les autorisations 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 dû à 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 d'autorisation 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 modifié lors de la compilation).

VOIR EGALEMENT

Documentation https://www.zabbix.com/manuals

zabbix_agent2(8), **zabbix_get**(1), **zabbix_js**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_server**(8)

Index

NOM

SYNOPSIS

DESCRIPTION

OPTIONS

FICHIERS

VOIR AUSSI

AUTEUR

Ce document a été créé le : 20:50:13 GMT, 22 novembre 2021

zabbix_get

Section : Commandes utilisateur (1)
Mis à jour : 2022-01-06
Index Retour au sommaire principal

NOM

zabbix_get - Utilitaire Zabbix get

SYNOPSIS

zabbix_get -s *host-name-or-IP* [**-p** *port-number*] [**-I** *IP-address*] [**-t** *timeout*] **-k** *item-key*
zabbix_get -s *host-name-or-IP* [**-p** *port-number*] [**-I** *IP-address*] [**-t** *timeout*] **--tls-connect cert --tls-ca-file** *CA-file* [**--tls-crl-file** *CRL-file*] [**--tls-agent-cert-issuer** *cert-issuer*] [**--tls-agent-cert-subject** *cert-subject*] **--tls-cert-file** *cert-file* **--tls-key-file** *key-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] **-k** *item-key*
zabbix_get -s *host-name-or-IP* [**-p** *port-number*] [**-I** *IP-address*] [**-t** *timeout*] **--tls-connect psk --tls-psk-identity** *PSK-identity* **--tls-psk-file** *PSK-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] **-k** *item-key*
zabbix_get -h
zabbix_get -V

DESCRIPTION

zabbix_get est un utilitaire en ligne de commande permettant d'obtenir des données depuis l'agent Zabbix.

OPTIONS

-s, **--host** *nom-d-hôte-ou-IP*
Spécifiez le nom d'hôte ou l'adresse IP d'un hôte.

-p, **--port** *numéro-de-port*
Spécifiez le numéro de port de l'agent exécuté sur l'hôte.
La valeur par défaut est 10050.

-I, **--source-address** *adresse-IP*
Spécifiez l'adresse IP source.

-t, **--timeout** *secondes*
Spécifiez le délai d'expiration.
Plage valide : 1 à 600 secondes (par défaut : 30)

-k, **--key** *clé-d-élément*
Spécifiez la clé de l'élément dont la valeur doit être récupérée.

-P, **--protocol** *valeur*
Protocole utilisé pour communiquer avec l'agent.
Valeurs :

auto connexion à l'aide du protocole JSON, avec repli et nouvelle tentative en utilisant le protocole en texte brut (par défaut)

json connexion à l'aide du protocole JSON

plaintext connexion à l'aide du 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 :

sans chiffrement
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 *CA-file*
Chemin complet vers un fichier contenant les certificats de la ou des autorités de certification racines pour la vérification du certificat du pair.

--tls-crl-file *CRL-file*
Chemin complet vers 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 vers un fichier contenant le certificat ou la chaîne de certificats.

--tls-key-file *key-file*
Chemin complet vers un fichier contenant la clé privée.

--tls-psk-identity *PSK-identity*
Chaîne d'identité PSK.

--tls-psk-file *PSK-file*
Chemin complet vers un fichier contenant la clé pré-partagée.

--tls-cipher13 *cipher-string*
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 cryptographique.
Cette option n'est pas disponible si la version d'OpenSSL est inférieure à 1.1.1.

--tls-cipher *cipher-string*
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 par défaut de la suite cryptographique.

-h, **--help**
Afficher cette aide et quitter.

-V, **--version**
Afficher les informations de version et quitter.

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 ÉGALEMENT

Documentation https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_server**(8), **zabbix_js**(1), **zabbix_agent2**(8), **zabbix_web_service**(8)

Index

NOM

SYNOPSIS

DESCRIPTION

OPTIONS

EXEMPLES

VOIR AUSSI

AUTEUR

Ce document a été créé le : 08:42:29 GMT, 11 juin 2021

zabbix_js

Section : Commandes utilisateur (1)
Mis à jour : 2022-01-06
Index Retour au sommaire principal

NOM

zabbix_js - Utilitaire Zabbix JS

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 embarqués.

OPTIONS

-s, **--script** *script-file*
Spécifie le nom de fichier du script à exécuter. Si « - » est indiqué comme nom de fichier, le script sera lu depuis stdin.

-p, **--param** *input-param*
Spécifie le paramètre d'entrée.

-i, **--input** *input-file*
Spécifie le nom de fichier du paramètre d'entrée. Si « - » est indiqué comme nom de fichier, l'entrée sera lue depuis stdin.

-w, **--webdriver** *url*
Spécifie l'URL du webdriver.

-l, **--loglevel** *log-level*
Spécifie le niveau de journalisation.

-t, **--timeout** *timeout*
Spécifie 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 ÉGALEMENT

Documentation https://www.zabbix.com/manuals

zabbix_agent2(8), **zabbix_agentd**(8), **zabbix_get**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_server**(8)

Indice

NOM

SYNOPSIS

DESCRIPTION

OPTIONS

EXEMPLES

VOIR EGALEMENT

Ce document a été créé le : 21:23:35 GMT, le 18 mars 2020

zabbix_proxy

Section : Commandes de maintenance (8)
Mise à jour : 2020-09-04
Index Retour au sommaire principal

NOM

zabbix_proxy - Démon proxy Zabbix proxy

RÉSUMÉ

zabbix_proxy [**-c** *config-file*]
zabbix_proxy [**-c** *config-file*] **-R** *runtime-option*
zabbix_proxy [**-c** *config-file*] **-T**
zabbix_proxy -h
zabbix_proxy -V

DESCRIPTION

zabbix_proxy est un démon qui collecte les données de supervision des appareils et les envoie au serveur Zabbix.

OPTIONS

-c, **--config** *config-file*
Utiliser le *config-file* alternatif au lieu de celui par défaut.

-f, **--foreground**
Exécuter le proxy Zabbix au premier plan.

-R, **--runtime-control** *runtime-option*
Exécuter des fonctions administratives selon *runtime-option*.

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 le signal sera envoyé au processus indiqué dans le fichier PID.

snmp_cache_reload
Recharger le cache SNMP. Notez que Zabbix met en cache les correspondances EngineID→IP de SNMPv3 et peut réutiliser automatiquement les EngineID afin de réduire la charge d'interrogation.

housekeeper_execute
Exécuter le housekeeper. Ignoré si le housekeeper est en cours d'exécution.

diaginfo[=*section*]
Consigner dans le journal 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 dans le journal.

log_level_increase[=*target*]
Augmenter le niveau de journalisation ; affecte tous les processus si la cible n'est pas spécifiée.

log_level_decrease[=*target*]
Diminuer le niveau de journalisation ; affecte tous les processus si la cible n'est pas spécifiée.

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, housekeeper, http poller, icmp pinger, ipmi manager, ipmi poller, java poller, odbc poller, poller, agent poller, http agent poller, snmp poller, preprocessing manager, self-monitoring, snmp trapper, task manager, trapper, unreachable poller, vmware collector)

process-type,N
Type de processus et numéro (par ex., poller,3)

pid
Identifiant du processus, jusqu'à 65535. Pour des valeurs plus élevées, spécifiez la cible comme « process-type,N »

Cibles de contrôle du profilage

process-type
Tous les processus du type spécifié (configuration syncer, data sender, discovery manager, history syncer, housekeeper, http poller, preprocessing manager, icmp pinger, ipmi manager, ipmi poller, java poller, poller, agent poller, http agent poller, snmp poller, self-monitoring, snmp trapper, task manager, trapper, unreachable poller, vmware collector, history poller, availability manager, odbc poller)

process-type,N
Type de processus et numéro (par ex., history syncer,1)

pid
Identifiant du processus, jusqu'à 65535. Pour des valeurs plus élevées, spécifiez la cible comme « process-type,N »

scope
La portée du profilage (rwlock, mutex, processing) peut être utilisée avec le type de processus (par ex., 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 (s'il n'a pas été modifié lors de la compilation).

VOIR EGALEMENT

Documentation https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(1), **zabbix_sender**(1), **zabbix_server**(8), **zabbix_js**(1), **zabbix_agent2**(8)

Index

NOM

SYNOPSIS

DESCRIPTION

OPTIONS

FICHIERS

VOIR AUSSI

AUTEUR

Ce document a été créé le : 16:12:22 GMT, 04 septembre 2020

zabbix_sender

Section : Commandes utilisateur (1)
Mise à jour : 2021-06-01
Index Retour au sommaire principal

NOM

zabbix_sender - Utilitaire Zabbix sender

SYNOPSIS

zabbix_sender [**-v**] **-z** *serveur* [**-p** *port*] [**-I** *adresse-IP*] [**-t** *délai d'expiration*] **-s** *hôte* **-k** *clé* **-o** *valeur*
zabbix_sender [**-v**] **-z** *serveur* [**-p** *port*] [**-I** *adresse-IP*] [**-t** *délai d'expiration*] [**-s** *hôte*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *fichier-d'entrée*
zabbix_sender [**-v**] **-c** *fichier-de-configuration* [**-z** *serveur*] [**-p** *port*] [**-I** *adresse-IP*] [**-t** *délai d'expiration*] [**-s** *hôte*] **-k** *clé* **-o** *valeur*
zabbix_sender [**-v**] **-c** *fichier-de-configuration* [**-z** *serveur*] [**-p** *port*] [**-I** *adresse-IP*] [**-t** *délai d'expiration*] [**-s** *hôte*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *fichier-d'entrée*
zabbix_sender [**-v**] **-z** *serveur* [**-p** *port*] [**-I** *adresse-IP*] [**-t** *délai d'expiration*] **-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-certificat* **--tls-key-file** *fichier-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 d'expiration*] [**-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-certificat* **--tls-key-file** *fichier-clé* [**--tls-cipher13** *chaîne-de-chiffrement*] [**--tls-cipher** *chaîne-de-chiffrement*]

[**-T**] [**-N**] [**-r**] [**-g**] **-i** *fichier-d'entrée*
zabbix_sender [**-v**] **-c** *fichier-de-configuration* [**-z** *serveur*] [**-p** *port*] [**-I** *adresse-IP*] [**-t** *délai d'expiration*] [**-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-certificat* **--tls-key-file** *fichier-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 d'expiration*] [**-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-certificat* **--tls-key-file** *fichier-clé* [**--tls-cipher13** *chaîne-de-chiffrement*] [**--tls-cipher** *chaîne-de-chiffrement*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *fichier-d'entrée*
zabbix_sender [**-v**] **-z** *serveur* [**-p** *port*] [**-I** *adresse-IP*] [**-t** *délai d'expiration*] **-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 d'expiration*] [**-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**] [**-g**] **-i** *fichier-d'entrée*
zabbix_sender [**-v**] **-c** *fichier-de-configuration* [**-z** *serveur*] [**-p** *port*] [**-I** *adresse-IP*] [**-t** *délai d'expiration*] [**-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 d'expiration*] [**-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**] [**-g**] **-i** *fichier-d'entrée*
zabbix_sender -h
zabbix_sender -V

DESCRIPTION

zabbix_sender est un utilitaire de ligne de commande pour envoyer des données de surveillance au serveur ou au proxy Zabbix. Sur le serveur Zabbix, un élément de type **trappeur Zabbix** doit être créé avec la clé correspondante. Notez que les valeurs entrantes ne seront acceptées qu'à partir des hôtes spécifiés dans le champ **Hôtes autorisés** pour cet élément.

OPTIONS

-c, **--config** *fichier-config*
Utiliser *fichier-config*. **Zabbix sender** lit les détails du serveur à partir du fichier de configuration 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 nom d'hôte défini via le paramètre **HostnameItem** ne sera pas repris ; dans ce cas, le nom d'hôte doit être spécifié 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 de données par lot 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. Lorsqu'elle est utilisée avec **--config**, cette option remplace les entrées du paramètre **ServerActive** spécifiées dans le fichier de configuration agentd.

-p, **--port** *port*
Spécifier le numéro de port du trapper du serveur Zabbix exécuté sur le serveur. La valeur par défaut est 10051. Lorsqu'elle est utilisée avec **--config**, cette option remplace les entrées de port du paramètre **ServerActive** spécifiées dans le fichier de configuration agentd.

-I, **--source-address** *adresse-IP*
Spécifier l'adresse IP source. Lorsqu'elle est utilisée avec **--config**, cette option remplace le paramètre **SourceIP** spécifié dans le fichier de configuration agentd.

-t, **--timeout** *secondes*
Spécifier le délai d'expiration. Plage valide : 1-300 secondes (par défaut : 60)

-s, **--host** *hôte*
Spécifier le nom d'hôte auquel l'élément appartient (tel qu'enregistré dans le frontend Zabbix). L'adresse IP de l'hôte et le nom DNS ne fonctionneront pas. Lorsqu'elle est utilisée avec **--config**, cette option remplace le paramètre **Hostname** spécifié dans le fichier de configuration agentd.

-k, **--key** *clé*
Spécifier la clé de l'élément auquel envoyer la valeur.

-o, **--value** *valeur*
Spécifier la valeur de l'élément.

-i, **--input-file** *fichier-entrée*
Charger les valeurs depuis un fichier d'entrée. Spécifiez **-** comme **<input-file>** pour lire les valeurs depuis l'entrée standard. Chaque ligne du fichier contient des champs séparés par des espaces : **<hostname> <key> <value>**. Chaque valeur doit être spécifiée sur sa propre ligne. Chaque ligne doit contenir 3 entrées séparées par des espaces : **<hostname> <key> <value>**, où "hostname" est le nom de l'hôte supervisé tel qu'enregistré dans le frontend Zabbix, "key" est la clé de l'élément cible et "value" la valeur à envoyer. Spécifiez **-** comme **<hostname>** pour utiliser le nom d'hôte du fichier de configuration de l'agent ou de l'argument **--host**.

Exemple de ligne dans un fichier d'entrée :

"Linux DB3" db.connections 43

Le type de valeur doit être correctement défini dans la configuration de l'élément du frontend Zabbix.
Zabbix sender enverra jusqu'à 250 valeurs dans une seule connexion.
La limite de taille pour l'envoi de valeurs depuis un fichier d'entrée dépend de la taille décrite dans le protocole de communication Zabbix.
Le contenu du fichier d'entrée doit être encodé en UTF-8.
Toutes les valeurs du fichier d'entrée sont envoyées séquentiellement 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 citation.
- Les entrées contenant des espaces doivent être 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 de 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 : **<hostname> <key> <timestamp> <value>**.
L'horodatage doit être spécifié 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 des événements ne sera pas correct.

Exemple de ligne dans le fichier d'entrée :

"Linux DB3" db.connections 1429533600 43

Pour plus de détails, veuillez consulter l'option **--input-file**.

Si une valeur horodatée est envoyée pour un hôte qui est dans une maintenance de type « aucune donnée », cette valeur sera ignorée ; cependant, il est possible d'envoyer une valeur horodatée pour 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 : **<hostname> <key> <timestamp> <ns> <value>**.

Exemple de ligne dans le fichier d'entrée :

"Linux DB3" db.connections 1429533600 7402561 43

Pour plus de détails, veuillez consulter l'option **--input-file**.

-r, **--real-time**
Envoyer les valeurs une par une dès leur réception. Cela peut être utilisé lors de la lecture depuis l'entrée standard.

-g, **--group**
Regrouper les valeurs par hôtes et les envoyer à chaque hôte dans un lot séparé.

--tls-connect *valeur*
Comment se connecter au serveur ou au proxy. Valeurs :

unencrypted
se connecter sans cryptage (par défaut)

psk
se connecter en utilisant TLS et une clé pré-partagée

cert
se connecter à l'aide de TLS et d'un certificat

--tls-ca-file *CA-file*
Chemin d'accès complet d'un fichier contenant les certificats des autorités de certification de niveau supérieur pour la vérification des certificats homologues.

--tls-crl-file *CRL-file*
Chemin d'accès complet d'un fichier contenant des certificats révoqués.

--tls-server-cert-issuer *cert-issuer*
Émetteur de certificat de serveur autorisé.

--tls-server-cert-subject *cert-subject*
Sujet du certificat de serveur autorisé.

--tls-cert-file *cert-file*
Chemin d'accès complet d'un fichier contenant le certificat ou la chaîne de certificats.

--tls-key-file *key-file*
Chemin d'accès complet d'un fichier contenant la clé privée.

--tls-psk-identity *PSK-identity*
Chaîne d'identité PSK.

--tls-psk-file *PSK-file*
Chemin d'accès complet d'un fichier contenant la clé pré-partagée.

--tls-cipher13 *cipher-string*
Chaîne de chiffrement pour OpenSSL 1.1.1 ou plus récent pour TLS 1.3. Remplacer les critères de sélection de suite de chiffrement par défaut. Cette option n'est pas disponible si la version d'OpenSSL est inférieure à 1.1.1.

--tls-cipher *cipher-string*
Chaîne de priorité GnuTLS (pour TLS 1.2 et versions ultérieures) ou chaîne de chiffrement OpenSSL (uniquement pour TLS 1.2). Remplacer les critères de sélection de suite de chiffrement par défaut.

-v, **--verbose**
Mode verbeux, **-vv** pour plus de détails.

-h, **--help**
Afficher cette aide et quitter.

-V, **--version**
Afficher les informations de version et quitter.

ETAT DE SORTIE

L'état de sortie est 0 si les valeurs ont été envoyées et qu'elles ont toutes été traitées avec succès par le serveur. Si des données ont été envoyées, mais que le traitement d'au moins une des valeurs a échoué, l'état de sortie est 2. Si l'envoi des données a échoué, l'état de sortie est 1.

EXEMPLES

zabbix_sender -c /etc/zabbix/zabbix_agentd.conf -k mysql.queries -o 342.45

Envoyer **342.45** comme valeur pour l'élément **mysql.queries** de l'hôte surveillé. Utiliser l'hôte surveillé 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 pour l'élément **mysql.queries** de l'hôte **Monitored Host** 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 avec l'IP **192.168.1.113**. Les noms d'hôte 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, spécifiée 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 à partir du même fichier de configuration.

echo '"Zabbix server" trapper.item ""' | zabbix_sender -z 192.168.1.113 -p 10000 -i -

Envoyer la valeur vide d'un élément au serveur Zabbix avec l'adresse IP **192.168.1.113** sur le port **10000** à partir de 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

Envoyez **342.45** comme valeur pour l'élément **mysql.queries** de l'hôte **Monitored Host** au serveur avec l'IP **192.168.1.113** en utilisant TLS avec 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

Envoyez **342.45** comme valeur pour l'élément **mysql.queries** de l'hôte **Monitored Host** au serveur avec l'IP **192.168.1.113** en utilisant TLS avec la clé pré-partagée (PSK).

VOIR EGALEMENT

Documentation https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(1), **zabbix_proxy**(8), **zabbix_server**(8), **zabbix_js**(1), **zabbix_agent2**(8), **zabbix_web_service**(8)

Index

NOM

SYNOPSIS

DESCRIPTION

OPTIONS

ETAT DE SORTIE

EXEMPLES

VOIR EGALEMENT

AUTEUR

Ce document a été créé le : 08:42:39 GMT, June 11, 2021

zabbix_server

Section : Commandes de maintenance (8)
Mise à jour : 2020-09-04
Index Retour au sommaire principal

NOM

zabbix_server - Démon serveur Zabbix

RÉSUMÉ

zabbix_server [**-c** *config-file*]
zabbix_server [**-c** *config-file*] **-R** *runtime-option*
zabbix_server [**-c** *config-file*] **-T**
zabbix_server -h
zabbix_server -V

DESCRIPTION

zabbix_server est le démon principal du logiciel Zabbix.

OPTIONS

-c, **--config** *config-file*
Utiliser le *config-file* alternatif au lieu de celui par défaut.

-f, **--foreground**
Exécuter le serveur Zabbix au premier plan.

-R, **--runtime-control** *runtime-option*
Exécuter des fonctions administratives conformément à *runtime-option*.

-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.

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 D'EXÉCUTION

Options de contrôle d'exécution :

config_cache_reload
Recharge le cache de configuration.
Ignoré si le cache est en cours de chargement.
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
Recharge le cache SNMP, efface les propriétés SNMP (engine time, engine boots, engine id, credentials) pour tous les hôtes.
Notez que Zabbix met en cache les correspondances SNMPv3 EngineID→IP et peut réutiliser automatiquement les EngineID afin de réduire la surcharge d'interrogation.

housekeeper_execute
Exécute le housekeeper.
Ignoré si le housekeeper est en cours d'exécution.

trigger_housekeeper_execute
Exécute le housekeeper des déclencheurs (supprime les problèmes des déclencheurs supprimés).
Ignoré si le housekeeper des déclencheurs pour les services est en cours d'exécution.

diaginfo[=*section*]
Consigne dans le journal les informations de diagnostic internes de la section spécifiée.
La section peut être *historycache*, *preprocessing*, *alerting*, *lld*, *valuecache*, *locks*.
Par défaut, les informations de diagnostic de toutes les sections sont consignées.

ha_status
Consigne dans le journal l'état du cluster de haute disponibilité (HA).

ha_remove_node[=*target*]
Supprime le nœud de haute disponibilité (HA) spécifié par son nom ou son ID.
Notez que les nœuds actifs/de secours ne peuvent pas être supprimés.

ha_set_failover_delay[=*delay*]
Définit 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*] Recharge le cache de configuration du proxy.

secrets_reload
Recharge les secrets depuis Vault.

service_cache_reload
Recharge le cache du gestionnaire de services.

prof_enable[=*target*]
Active le profilage.
Affecte tous les processus si target n'est pas spécifié.
Le profilage activé fournit des détails sur tous les rwlocks/mutexes par nom de fonction.

prof_disable[=*target*]
Désactive le profilage.
Affecte tous les processus si target n'est pas spécifié.

log_level_increase[=*target*]
Augmente le niveau de journalisation, affecte tous les processus si target n'est pas spécifié

log_level_decrease[=*target*]
Diminue 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 (alerter, gestionnaire d'alertes, gestionnaire de disponibilité, browser poller, configuration syncer, worker de configuration syncer, gestionnaire de connecteurs, worker de connecteur, gestionnaire de découverte, escalator, gestionnaire HA, history poller, history syncer, housekeeper, http poller, icmp pinger, internal poller, gestionnaire IPMI, ipmi poller, java poller, odbc poller, poller, agent poller, http agent poller, snmp poller, gestionnaire de prétraitement, gestionnaire de groupes de proxys, proxy poller, auto-surveillance, gestionnaire de services, snmp trapper, gestionnaire de tâches, timer, trapper, unreachable poller, vmware collector)

process-type,N
Type de processus et numéro (par ex., poller,3)

pid
Identifiant du processus, jusqu'à 65535. Pour des valeurs plus grandes, spécifiez la cible comme "process-type,N"

Cibles de contrôle du profilage

process-type
Tous les processus du type spécifié (alerter, gestionnaire d'alertes, gestionnaire de disponibilité, browser poller, configuration syncer, worker de configuration syncer, gestionnaire de connecteurs, worker de connecteur, gestionnaire de découverte, escalator, gestionnaire HA, history poller, history syncer, housekeeper, http poller, icmp pinger, internal poller, gestionnaire IPMI, ipmi poller, java poller, odbc poller, poller, agent poller, http agent poller, snmp poller, gestionnaire de prétraitement, gestionnaire de groupes de proxys, proxy poller, auto-surveillance, gestionnaire de services, snmp trapper, gestionnaire de tâches, timer, trapper, unreachable poller, vmware collector)

process-type,N
Type de processus et numéro (par ex., history syncer,1)

pid
Identifiant du processus, jusqu'à 65535. Pour des valeurs plus grandes, spécifiez la cible comme "process-type,N"

scope
La portée du profilage (rwlock, mutex, processing) peut être utilisée avec le type de processus (par ex., history syncer,1,processing)

FICHIERS

/usr/local/etc/zabbix_server.conf
Emplacement par défaut du fichier de configuration du serveur Zabbix (s'il n'a pas été modifié lors de la compilation).

VOIR EGALEMENT

Documentation https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_js**(1), **zabbix_agent2**(8)

Index

NOM

SYNOPSIS

DESCRIPTION

OPTIONS

FICHIERS

VOIR AUSSI

AUTEUR

Ce document a été créé le : 16:12:14 GMT, 04 septembre 2020

zabbix_web_service

Section : Commandes de maintenance (8)
Mise à jour : 2019-01-29
Index Retour au sommaire principal

NOM

zabbix_web_service - Service web Zabbix

RÉSUMÉ

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 *config-file* alternatif 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 EGALEMENT

Documentation https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_server**(8), **zabbix_js**(1), **zabbix_agent2**(8)

Indice

NOM

SYNOPSIS

DESCRIPTION

OPTIONS

FICHIERS

VOIR EGALEMENT

AUTEUR

Ce document a été créé le : 12:58:30 GMT, le 11 juin 2021

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.