Documentation 8.2

ZABBIX

05.11.2026

Contents

Benutzerhandbuch **5**

Willkommen in der Zabbix-Dokumentation **5**

Erste Schritte mit Zabbix . . . 5
Zabbix Cloud . . . 5
Mobile-App . . . 5
Entwicklerzentrum . . . 5
Community & andere Ressourcen . . . 5
Installation und erste Schritte . . . 6
Installation der Weboberfläche . . . 6
Schnellstartanleitungen . . . 11
Installation: zusätzliche Anleitungen . . . 86
Upgrade . . . 169
Bekannte Probleme . . . 186
Anforderungen . . . 194
Was ist neu und Hinweise zum Upgrade: 8.0 . . . 205
Upgrade-Hinweise . . . 205
Breaking changes . . . 205
Andere . . . 205
Neuerungen und Hinweise zum Upgrade: 8.0.x . . . 205
Zabbix-Appliance . . . 205
Prozesse . . . 209
Server . . . 211
Proxy . . . 239
Agent . . . 286
Agent 2 . . . 317
Java gateway . . . 383
Absender . . . 387
Abrufen . . . 390
JS . . . 390
Webdienst . . . 392
Bridge-Adapter . . . 394
MCP-Server . . . 398
Umgebungsvariablen . . . 406
Konfiguration . . . 407
Hosts und Hostgruppen . . . 419
Datenpunkte . . . 438
Problemerkennung mit Auslösern . . . 669
Visualisierung . . . 723
Vorlagen und Vorlagengruppen . . . 750
Vorlagen out of the box . . . 750
Benachrichtigungen bei Ereignissen . . . 764
Makros . . . 819
Benutzer und Benutzergruppen . . . 834
Geplante Berichte . . . 845
Datenexport . . . 849
Konfigurations-Export/Import . . . 858
Reguläre Ausdrücke . . . 925
Ereignisse und Auflösung . . . 932
Problem-Bestätigung und Unterdrückung . . . 934
Manuelles Schließen von Problemen . . . 938

Wartungszeiträume . . . 939
Tagging . . . 945
Complex event processing . . . 948
Ereigniskorrelation . . . 952
Erkennung . . . 958
Netzwerkerkennung . . . 958
Aktive Agent-Autoregistrierung . . . 967
Low-level-Erkennung . . . 971
Überwachung virtueller Maschinen . . . 996
VMware-Überwachungs-Datenpunkt-Schlüssel . . . 997
Felder des Erkennungs-Schlüssels für virtuelle Maschinen . . . 1016
JSON-Beispiele für VMware-Datenpunkte . . . 1020
Beispiel für die VMware-Überwachungseinrichtung . . . 1024
Webüberwachung . . . 1029
Web-Monitoring-Datenpunkte . . . 1039
Praxisbeispiel . . . 1040
Serviceüberwachung . . . 1047
Service tree . . . 1048
SLA . . . 1052
Einrichtungsbeispiel . . . 1054
Upgrade der Dienstüberwachung . . . 1058
Verschlüsselung und Sicherheit . . . 1059
Verschlüsselung mit Zertifikaten . . . 1067
Verschlüsselung mit vorab gemeinsam genutzten Schlüsseln . . . 1075
Sichere Verbindung zur Datenbank . . . 1078
Sichere Verbindung zwischen Frontend und Server . . . 1091
Zugriffskontrolle . . . 1095
Kryptografie . . . 1099
Absicherung des Webservers . . . 1100
Speicherung von Geheimnissen . . . 1103
Google Chrome TLS-Zertifikatsvertrauen . . . 1112
Sicherheitsbewährte Praktiken . . . 1114
Weboberfläche . . . 1114
Menü . . . 1115
Frontend-Bereiche . . . 1125
Benutzereinstellungen . . . 1406
Globale Suche . . . 1412
Frontend-Wartungsmodus . . . 1414
Seitenparameter . . . 1415
Definitionen . . . 1417
Zusätzliche Frontend-Sprachen . . . 1418
Erstellen Ihres eigenen Themes . . . 1418
Debug-Modus . . . 1420
Von Zabbix verwendete Cookies . . . 1421
Zeitzonen . . . 1421
Passwort zurücksetzen . . . 1422
Zeitraum- und Host-Auswahlfelder . . . 1423
Konfigurationsparameter: Zabbix Frontend . . . 1425
API . . . 1431
Methodenreferenz . . . 1436
Anhang 1. Referenzkommentar . . . 2330
Anhang 2. Änderungen von 7.4 zu 8.0 . . . 2336
Anhang 3. Änderungen in 8.0 . . . 2336
Erweiterungen . . . 2336
Ladbare Module . . . 2339
Frontend-Module . . . 2348
Anhänge . . . 2349
Installation und Einrichtung . . . 2349
Datenpunkte . . . 2397
Unterstützte Funktionen . . . 2438
Makros . . . 2438
Versionskompatibilität . . . 2477
Syntax des Zeitraums . . . 2478

Einheitensymbole . . . 2478
Befehlsausführung . . . 2480
Beispiele für Escaping . . . 2481
Was ist Zabbix . . . 2483

Zabbix Cloud **2487**

Zabbix Cloud **2487**
Zabbix Cloud einrichten und verwalten . . . 2487
Erfahren Sie mehr über Zabbix Cloud . . . 2487
Zabbix in der Cloud bereitstellen . . . 2488
Node-Konfiguration . . . 2497
Übersicht . . . 2498
Registerkarte „Übersicht“ . . . 2499
Registerkarte „Zugriffsfilter“ . . . 2502
Registerkarte „Verschlüsselung“ . . . 2503
Registerkarte „Backups“ . . . 2503
Registerkarte „Historie“ . . . 2504
Registerkarte „Wartung“ . . . 2505
Registerkarte „Upgrade“ . . . 2505
Benutzer hinzufügen . . . 2507
Einführung . . . 2507
Benutzer zu Organisationen hinzufügen . . . 2507
Benutzer zu Nodes hinzufügen . . . 2508
Hauptunterschiede von Zabbix Cloud . . . 2509
Einführung . . . 2509
Auditprotokoll . . . 2511
Einen Zabbix Cloud Proxy einrichten . . . 2513

Mobile-App **2522**

Mobile App **2522**
Zabbix für die App einrichten . . . 2522
Informationen zur App . . . 2522
Zabbix für die App einrichten . . . 2522
App-Übersicht . . . 2523
Sicherer mobiler Zugriff . . . 2531

Entwicklerzentrum **2532**

Entwicklerzentrum **2532**
Auf Zabbix aufbauen . . . 2532
Erstellen eines Plugins . . . 2532
Plugin-Schnittstellen . . . 2538
Python-Bibliothek für Zabbix . . . 2540
Installation . . . 2540
Schnellstartanleitung . . . 2541
Zabbix API verwenden . . . 2544
Daten vom Zabbix Agent erfassen . . . 2547
Daten an Zabbix-Server oder -Proxy senden . . . 2549
Debug-Protokollierung . . . 2552
Module . . . 2553
Struktur der Moduldateien . . . 2554
Widgets . . . 2561
Tutorials . . . 2570
Beispiele . . . 2596
Änderungen an der Erweiterungsentwicklung . . . 2597

Zabbix-Manpages **2597**
zabbix_agent2 . . . 2597
zabbix_agentd . . . 2599
zabbix_get . . . 2600
zabbix_js . . . 2602
zabbix_proxy . . . 2603
zabbix_sender . . . 2606

zabbix_server 2611
zabbix_web_service 2613

Urheberrechtshinweis **2614**

Benutzerhandbuch

Willkommen in der Zabbix-Dokumentation

Ihre zentrale Anlaufstelle für die Arbeit mit dem Zabbix-Monitoring – von grundlegenden Setups bis hin zu erweiterten Konfigurationen. Dieses Handbuch behandelt alles, was für die Installation, Konfiguration und den Betrieb von Zabbix erforderlich ist.

Was ist neu in Zabbix 8.0

Erste Schritte mit Zabbix

Installation Schritt-für-Schritt-Anleitungen zur Installation von Zabbix auf Ihrer bevorzugten Plattform, einschließlich verschiedener Betriebssysteme und Datenbankkonfigurationen.

Anforderungen Eine Liste der unterstützten Plattformen und Softwarevoraussetzungen, die Ihnen dabei hilft, Ihre Umgebung für eine erfolgreiche Zabbix-Bereitstellung vorzubereiten.

Schnellstartanleitungen Knapp gehaltene, aufgabenorientierte Anleitungen, die Sie durch die Grundlagen führen – von den ersten Konfigurationsschritten bis zum Erhalt Ihrer ersten Problemmeldung.

Zabbix Cloud

Erste Schritte mit Zabbix Cloud Schritt-für-Schritt-Anleitungen, um Ihre Zabbix Cloud-Instanz schnell betriebsbereit zu machen, einschließlich der Erstkonfiguration und der Verbindung Ihrer überwachten Geräte.

Zabbix Cloud vs. On-Premises Vergleichen Sie die Unterschiede bei Verwaltung, Skalierbarkeit und Wartung zwischen den beiden Bereitstellungsarten und erfahren Sie, welche Ihre Arbeitsabläufe am nahtlosesten unterstützen kann.

Zabbix Cloud erkunden Verschaffen Sie sich einen Überblick über Zabbix Cloud, einschließlich der wichtigsten Funktionen, Vorteile und dessen, was in einem Abonnement enthalten ist. Häufige Fragen zur Einrichtung und Nutzung finden Sie in den FAQ.

Mobile-App

App herunterladen Installieren Sie die mobile Zabbix-App (demnächst verfügbar) auf Ihrem Gerät und beginnen Sie, Ihre Zabbix-Umgebung unterwegs zu überwachen.

Zabbix für die App einrichten Installieren Sie den Zabbix-Bridge-Adapter, damit mobile Geräte eine Verbindung zu Zabbix herstellen können. Fügen Sie anschließend Ihr Gerät hinzu und aktivieren Sie Push-Benachrichtigungen.

App-Übersicht Erfahren Sie die Grundlagen der Verwendung der mobilen Zabbix-App.

Entwicklerzentrum

Frontend-Module Anleitungen und Referenzen zum Erstellen benutzerdefinierter Module, die das Zabbix-Frontend erweitern oder anpassen, um spezifische Anwendungsfälle zu unterstützen.

Widgets Eine Aufschlüsselung der Widget-Struktur und -Logik mit Anleitungen zum Erstellen benutzerdefinierter Dashboard-Elemente, die auf Ihre Anforderungen zugeschnitten sind.

Plugins Ein Überblick darüber, wie Zabbix-Plugins entwickelt und verwaltet werden, um die Funktionalität zu erweitern oder die Integration mit externen Systemen zu ermöglichen.

Community & andere Ressourcen

Zabbix-Foren Ein Ort, um Ideen auszutauschen, Lösungen zu finden und Erfahrungen auf allen Ebenen der Zabbix-Expertise zu teilen.

Zabbix-Blog Neuigkeiten, Tutorials und Fallstudien, zusammengestellt vom Zabbix-Team und Mitwirkenden aus der Community.

Anleitungsvideos Eine Videobibliothek mit Demonstrationen, Diskussionen und Tipps, die Ihnen helfen, Zabbix optimal zu nutzen.

Installation und erste Schritte

Der folgende Ablauf beschreibt den Zabbix-Einrichtungsprozess:

1. Stellen Sie sicher, dass die erforderliche Software installiert ist und ausgeführt wird.
2. Installieren Sie Zabbix.
3. Installieren Sie die Zabbix-Weboberfläche.
4. Beginnen Sie mit der Überwachung.

Details zu jedem Schritt finden Sie unten.

Erforderliche Software Zabbix funktioniert gut mit der folgenden Software (oft als LAMP-Stack bezeichnet):

- **Linux-Betriebssystem:** Ubuntu, Debian, CentOS oder andere
- **Webserver:** Apache oder NGINX
- **Datenbankserver:** MySQL/MariaDB oder PostgreSQL
- **PHP**

Auf den meisten modernen Linux-Systemen ist PHP möglicherweise bereits installiert. In diesem Fall müssen Sie eventuell nur den Datenbankserver und den Webserver einrichten.

Ausführlichere Informationen, beispielsweise zu den unterstützten Softwareversionen, finden Sie unter Erforderliche Software.

Zabbix installieren Die **einfachste** Möglichkeit, Zabbix zu installieren, besteht in der Verwendung von **Distributionspaketen** (alternativ aus Containern). Die Pakete installieren kleinere Softwareabhängigkeiten.

Präzise, auf Ihr Betriebssystem zugeschnittene Installationsanweisungen finden Sie auf der **Zabbix-Website**.

Wählen Sie Ihre Plattform aus und befolgen Sie die bereitgestellten Anweisungen. Es wird empfohlen, zunächst einen Zabbix-Server, einen Zabbix-Agent 2 und das Frontend (Weboberfläche) zu installieren. Sie können später Zabbix-Proxys und weitere Agents hinzufügen.

Ein Teil des Installationsprozesses besteht darin, eine Datenbank zu erstellen (zum Speichern von Konfigurations- und Verlaufsdaten) und Anfangsdaten in diese zu importieren.

Zabbix-Weboberfläche installieren Führen Sie die Schritte des **Installationsassistenten** für die Weboberfläche aus.

Sie werden nach dem Datenbankpasswort (das im vorherigen Schritt erstellt wurde) gefragt, damit das Frontend mit der Datenbank kommunizieren kann.

Überwachung starten Melden Sie sich bei der Zabbix-Weboberfläche an.

Verwenden Sie **Schnellstartanleitungen**, um mit der Überwachung gängiger Ressourcen (Linux, Apache, MySQL oder anderer) zu beginnen. Diese Schnellstartanleitungen erläutern, mit welchen Vorlagen Sie arbeiten können, um die Überwachung gängiger Ressourcen zu starten.

Sie können mit der Schnellstartanleitung zur Überwachung eines Linux-Servers beginnen.

Installation der Weboberfläche

Dieser Abschnitt enthält Anweisungen zur Installation des Zabbix-Frontends. Führen Sie diese Schritte nach der Installation des Zabbix-Backends durch.

Das Zabbix-Frontend ist in PHP geschrieben und erfordert einen unterstützten Webserver wie Apache oder NGINX mit konfiguriertem PHP. Informationen zur erforderlichen Software und den erforderlichen Bibliotheken finden Sie unter den Voraussetzungen für das Frontend.

Um in der Zabbix-Weboberfläche eine andere Sprache als Englisch zu verwenden, muss das entsprechende Locale auf dem Webserver zusammen mit der PHP-gettext-Erweiterung installiert sein. Beachten Sie, dass eine Änderung der Sprache das Zeit-/Datumsformat ändern kann.

Informationen zum Einrichten von SSL für das Zabbix-Frontend finden Sie unter SSL für das Zabbix-Frontend einrichten.

Installationsassistent

Um das Webinterface zu installieren, öffnen Sie die URL des Zabbix Frontend in einem Browser:

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

Willkommensbildschirm

Der Installationsassistent des Frontend wird auf dem Willkommensbildschirm geöffnet.

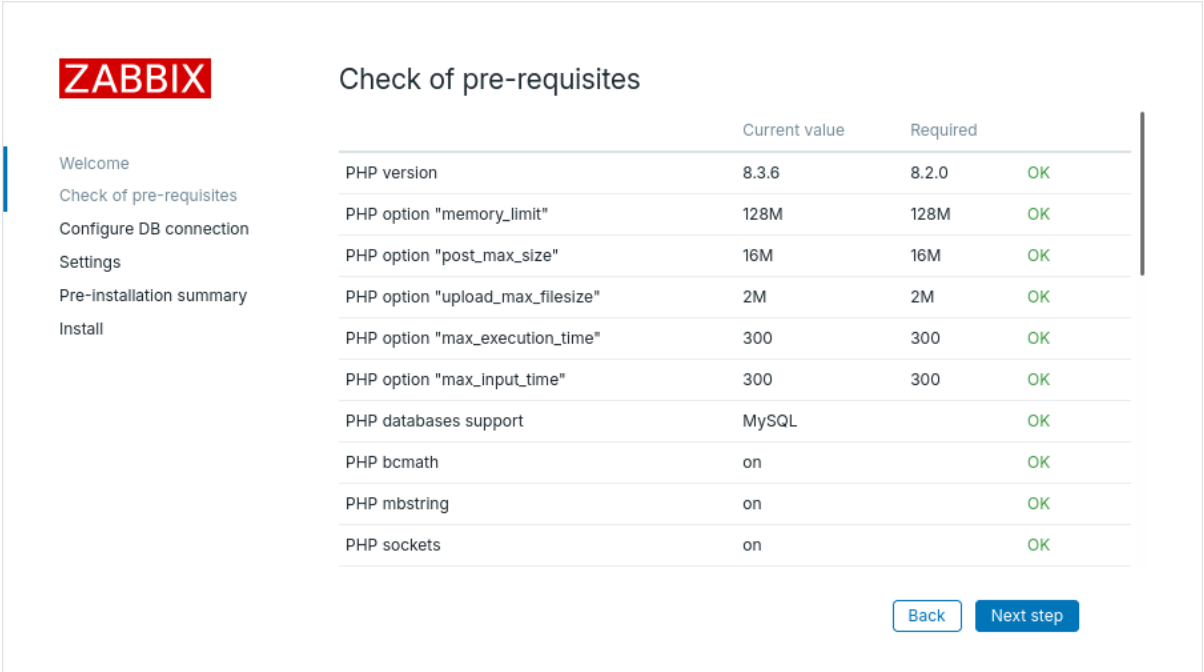


Die systemweite Standardsprache ist Englisch (en_US), mit dem US-amerikanischen Zeit-/Datumsformat. Um die systemweite Standardsprache zu ändern, wählen Sie die gewünschte Sprache aus dem Dropdown-Menü *Standardsprache* aus.

Klicken Sie auf die Schaltfläche *Nächster Schritt*, um fortzufahren.

Prüfung der Voraussetzungen

Stellen Sie sicher, dass alle obligatorischen Voraussetzungen für das Zabbix-Frontend erfüllt sind.



Unter Anforderungen finden Sie eine Liste der erforderlichen und optionalen PHP-Erweiterungen, der erforderlichen PHP-Version und der unterstützten Datenbanken.
Beachten Sie, dass nicht erfüllte optionale Voraussetzungen mit dem Status *Warnung* angezeigt werden. Der Einrichtungsvorgang kann auch fortgesetzt werden, wenn diese nicht erfüllt sind.

Erforderliche Werte für PHP-Konfigurationsdirektiven in der vom Webserver verwendeten Datei `php.ini`:

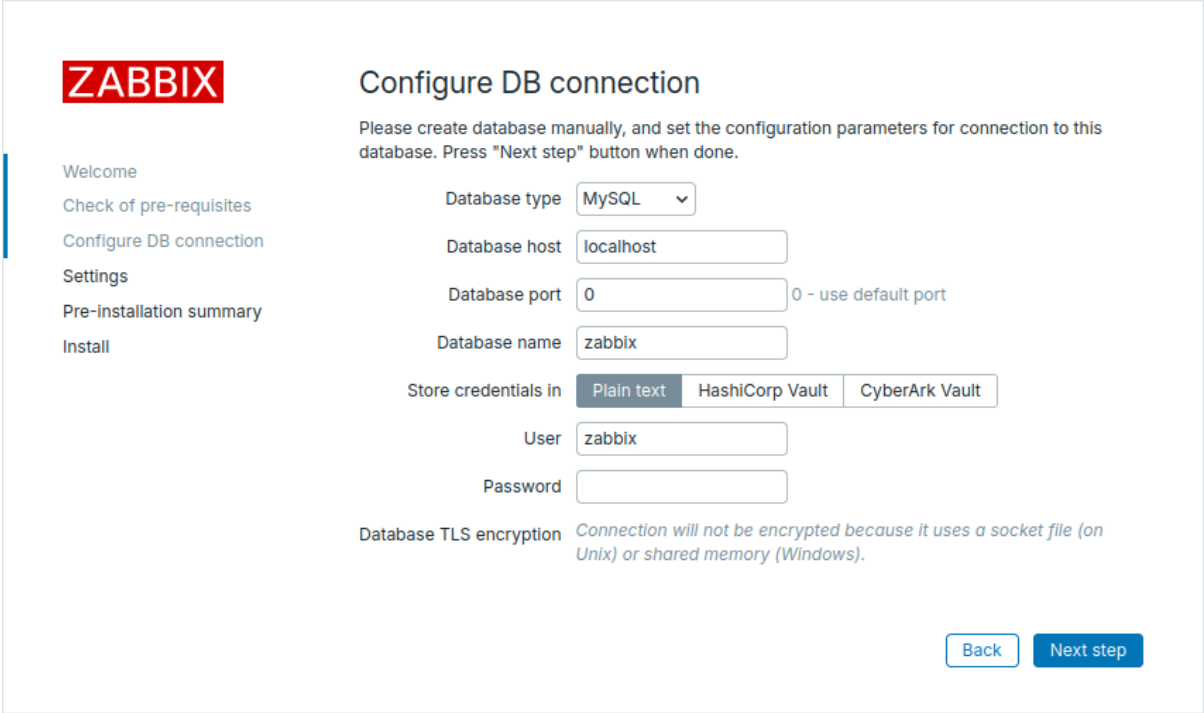
```
memory_limit = 128M ; Mindestwert
post_max_size = 16M ; Mindestwert
upload_max_filesize = 2M ; Mindestwert
max_execution_time = 300 ; Mindestwert, die Werte 0 und -1 sind ebenfalls zulässig
max_input_time = 300 ; Mindestwert, die Werte 0 und -1 sind ebenfalls zulässig
session.auto_start = 0 ; Muss deaktiviert sein
arg_separator.output = "&" ; "&amp;" ist ebenfalls zulässig
```

Attention:
Wenn der Benutzer oder die Gruppe des Webservers geändert werden muss, überprüfen Sie, ob der neue Benutzer oder die neue Gruppe über die erforderlichen Berechtigungen für das PHP-Sitzungsverzeichnis verfügt. Andernfalls kann die Einrichtung von Zabbix möglicherweise nicht fortgesetzt werden.

Wenn alle obligatorischen Voraussetzungen erfüllt sind, fahren Sie mit dem nächsten Schritt fort.

Datenbankverbindung konfigurieren

Geben Sie die Details für die Verbindung zur Datenbank ein.



Wenn das Feld *Datenbank-Host* leer bleibt, wird der standardmäßige Unix-Domain-Socket verwendet.

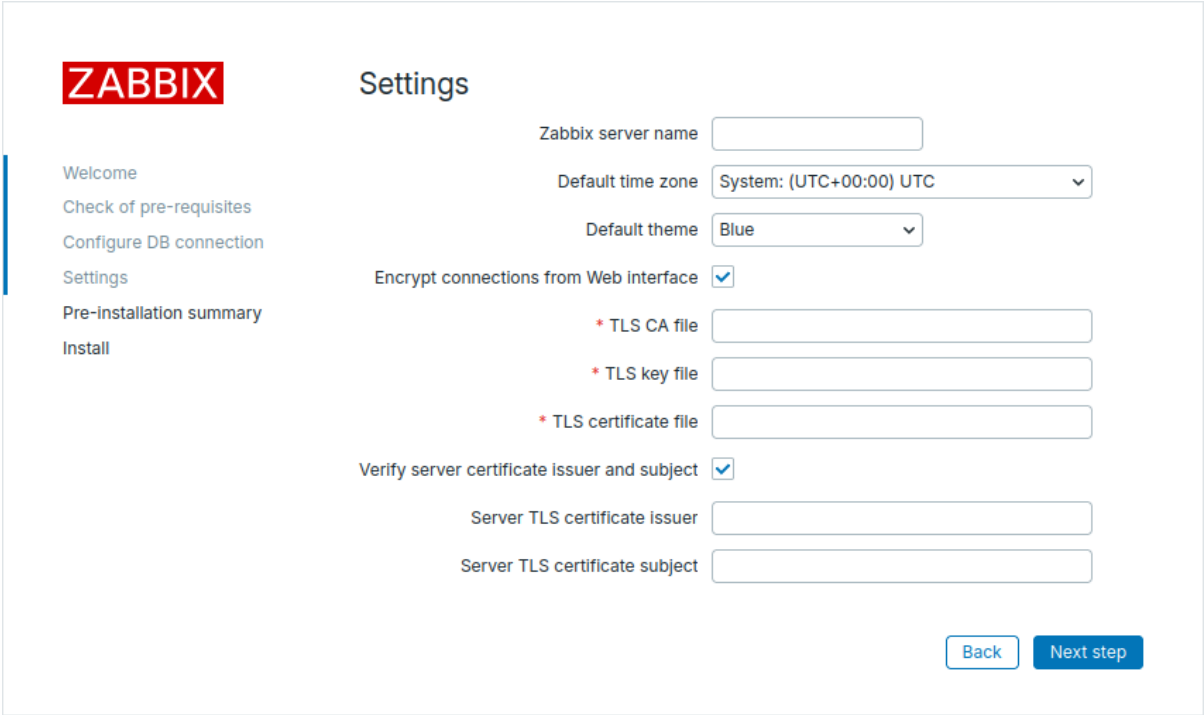
Für MySQL führt auch die Eingabe von `localhost` im Feld *Datenbank-Host* zu einer Verbindung über den standardmäßigen Unix-Domain-Socket. Der Assistent bietet keine Option zum Konfigurieren des Datenbank-Sockets. Um einen benutzerdefinierten Socket zu verwenden, konfigurieren Sie ihn in den Einstellungen des Zabbix Servers (zum Beispiel mit dem Parameter DBSocket= in der Server-Konfigurationsdatei (`zabbix_server.conf`)). Dadurch bleibt das Frontend mit den Einstellungen für die Verbindung vom Server zur Datenbank konsistent.

Für PostgreSQL kann das Feld *Datenbank-Host* mehrere durch Kommas getrennte `host:port`-Einträge enthalten. Die Zeichenfolge wird in Teile aufgeteilt, und jeder Port wird auf dieselbe Weise wie das Feld *Datenbank-Port* validiert. Nach erfolgreicher Validierung wird die Zeichenfolge in Hosts und Ports aufgeteilt, um die Verbindungszeichenfolge zu erstellen. Wenn für einen Host in der durch Kommas getrennten Liste kein Port definiert ist, wird der Wert von *Datenbank-Port* als Standardport verwendet. Wenn ein Socket-Pfad angegeben ist (zum Beispiel `/var/run/pgbouncer`), wird stattdessen dieser Unix-Domain-Socket verwendet.

Um eine sichere Verbindung zur Datenbank zu konfigurieren, aktivieren Sie das Kontrollkästchen *Datenbank-TLS-Verschlüsselung* und fahren Sie mit Frontend zur Datenbank fort, um die Felder zu konfigurieren. Das Kontrollkästchen *Datenbank-TLS-Verschlüsselung* ist in den folgenden Fällen deaktiviert: - Für MySQL ist *Datenbank-Host* auf *localhost* gesetzt. - Für PostgreSQL beginnt der Wert des Feldes *Datenbank-Host* mit einem Schrägstrich oder das Feld ist leer.

Um Zabbix so zu konfigurieren, dass Geheimnisse aus HashiCorp Vault abgerufen werden, wählen Sie unter *Anmeldedaten speichern in* die Option *HashiCorp Vault* und fahren Sie mit HashiCorp-Konfiguration fort, um die erweiterten Felder auszufüllen. Um Zabbix so zu konfigurieren, dass Geheimnisse aus CyberArk Vault abgerufen werden, wählen Sie unter *Anmeldedaten speichern in* die Option *CyberArk Vault* und fahren Sie mit CyberArk-Konfiguration fort, um die erweiterten Felder auszufüllen.

Einstellungen



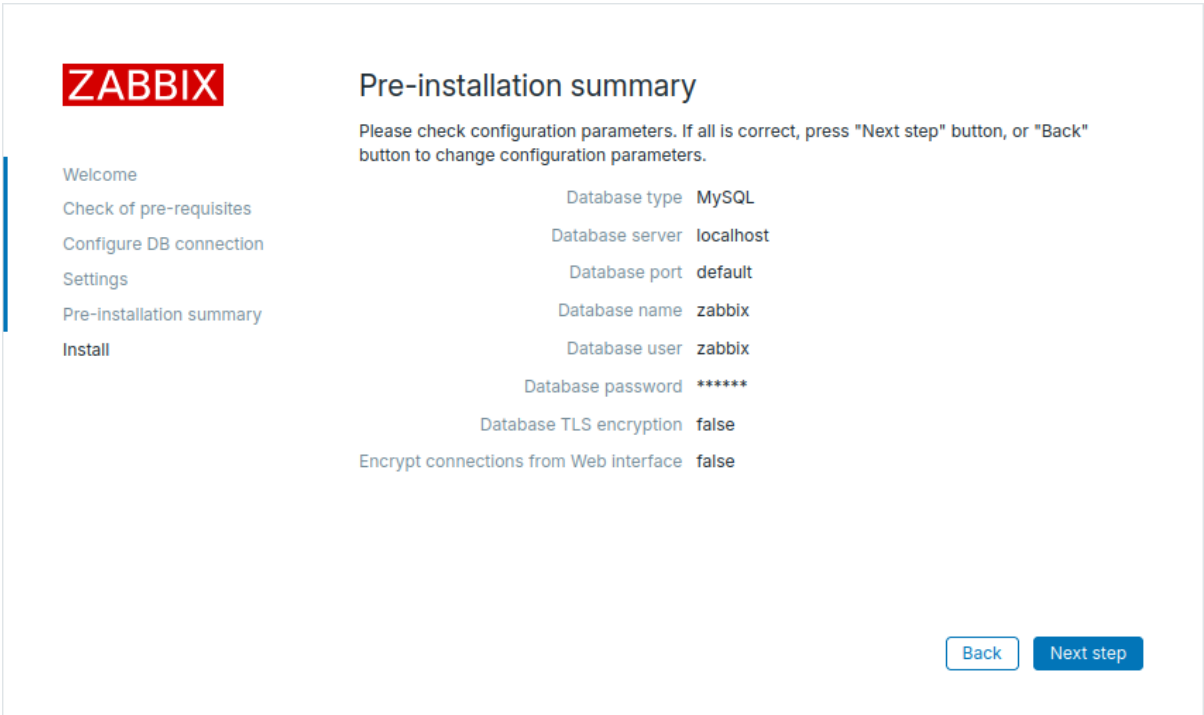
Die Eingabe eines Namens für den Zabbix-Server ist optional. Wenn ein Name angegeben wird, wird er jedoch in der Menüleiste und in den Seitentiteln angezeigt.

Legen Sie die standardmäßige Zeitzone und das Design für das Frontend fest.

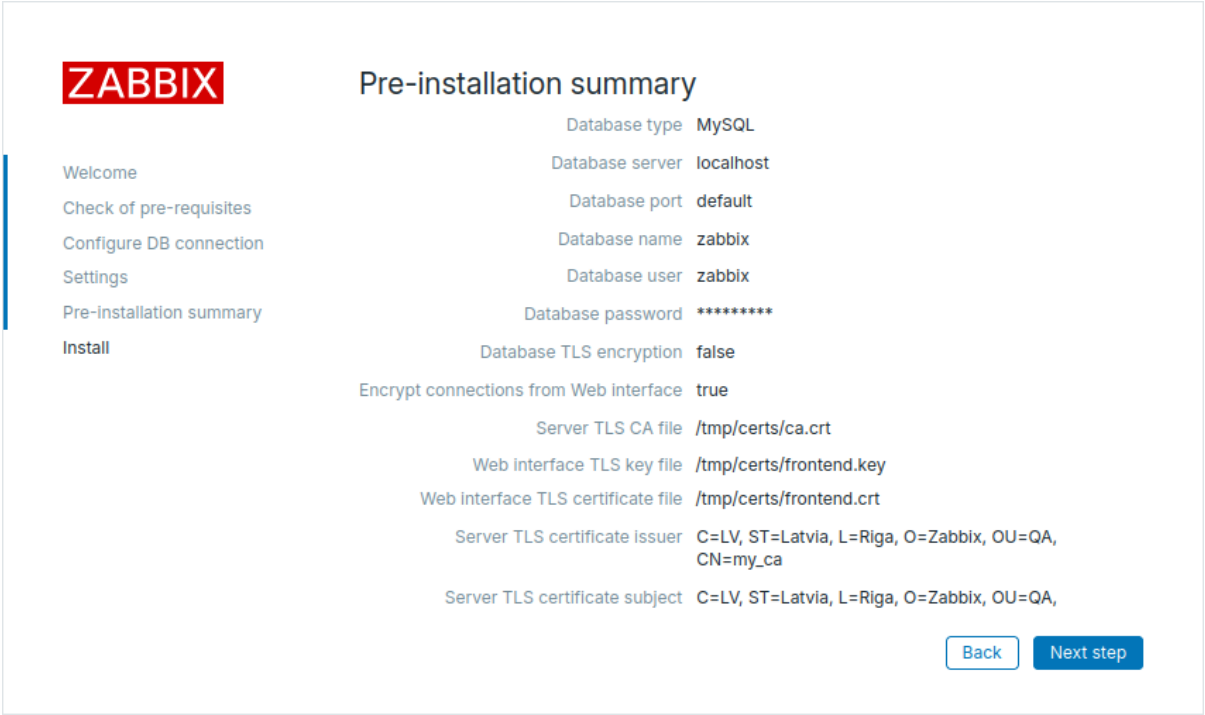
Um TLS-Verbindungen zwischen dem Zabbix-Frontend und dem Zabbix-Server zu konfigurieren, aktivieren Sie das Kontrollkästchen *Verbindungen über die Weboberfläche verschlüsseln* und fahren Sie mit der Konfiguration der TLS-Verbindung fort, um die erweiterten Felder auszufüllen.

Zusammenfassung vor der Installation

Überprüfen Sie eine Zusammenfassung der Einstellungen.



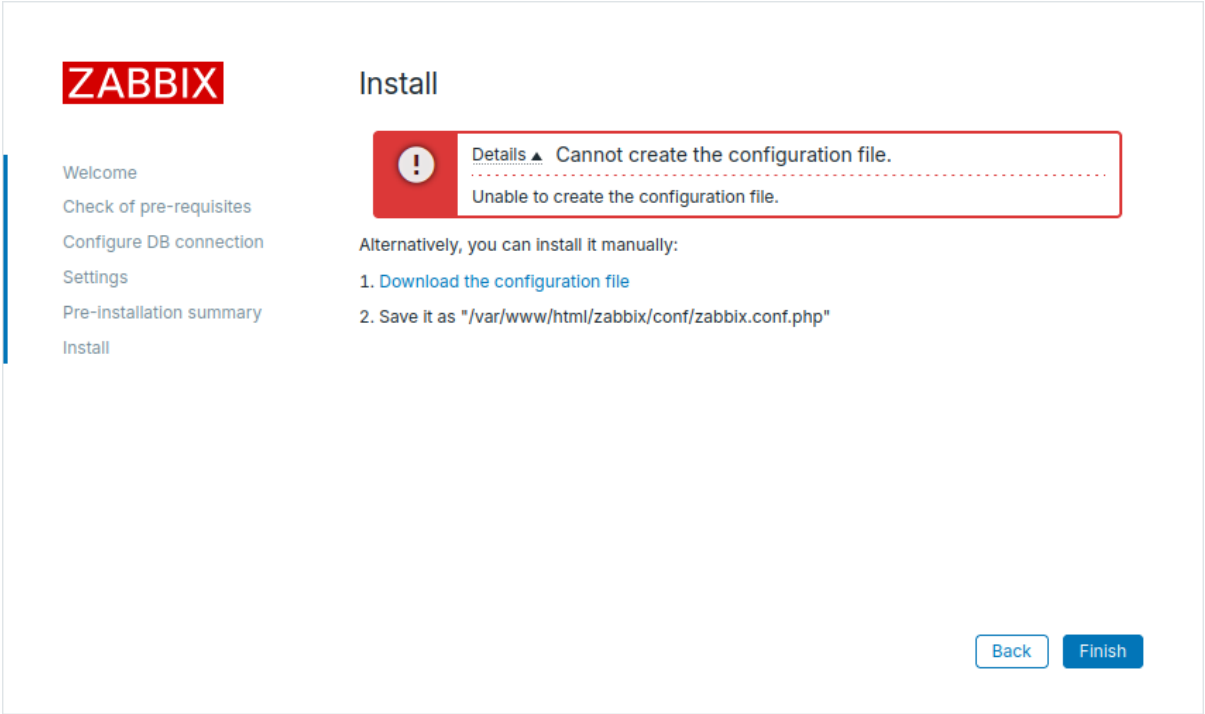
Wenn TLS konfiguriert ist, zeigt der Assistent die entsprechenden Details an.



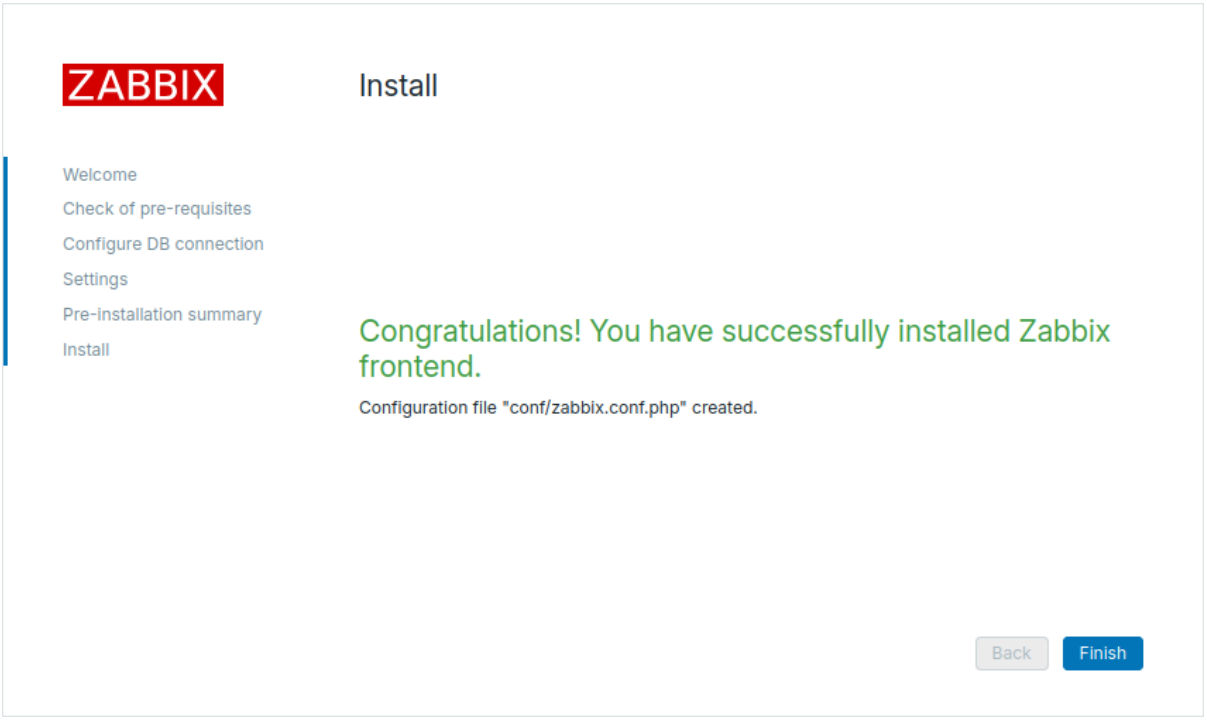
Installation

Wenn der Benutzer des Webservers Schreibzugriff auf das Verzeichnis `conf/` hat, wird die Konfigurationsdatei automatisch gespeichert, und Sie können direkt mit dem nächsten Schritt fortfahren.

Wenn der Benutzer des Webservers keinen Schreibzugriff auf das Verzeichnis `conf/` hat, müssen Sie die Konfigurationsdatei herunterladen und sie im Verzeichnis `conf/` innerhalb des Verzeichnisses mit den HTML-Dokumenten des Webservers ablegen, in das Sie die PHP-Dateien von Zabbix kopiert haben.

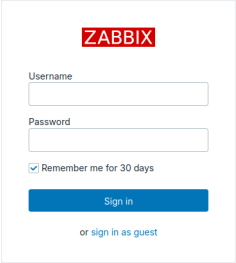


Schließen Sie die Installation ab.



Anmelden

Melden Sie sich mit den Standardanmeldedaten an: - Benutzername: *Admin* - Passwort: *zabbix*



Frontend-Konfigurationsdatei

Nach der Installation können Sie das Zabbix-Frontend neu konfigurieren, indem Sie die Parameter in der Frontend-Konfigurationsdatei `zabbix.conf.php` aktualisieren.

Der Standardpfad zur Konfigurationsdatei hängt vom Betriebssystem und der Installationsmethode von Zabbix ab:

- Bei der Installation aus Paketen: `/usr/share/zabbix/ui/conf/zabbix.conf.php`
- Bei der Installation aus den Quellen: `/var/www/html/zabbix/conf/zabbix.conf.php`
- Für die Appliance: `/etc/zabbix/web/zabbix.conf.php`

Weitere Informationen zur Konfiguration des Zabbix-Frontends finden Sie unter Optionen der Konfigurationsdatei.

Schnellstartanleitungen

Nachdem Sie Zabbix installiert und dessen Weboberfläche eingerichtet haben, können Sie mit der Überwachung beginnen.

Verwenden Sie diese Schnellstartanleitungen, um die Überwachung häufig verwendeter Ressourcen zu starten:

- Überwachung eines Linux-Servers
- Überwachung eines Apache-Webservers
- Überwachung eines MySQL-Servers

Die Schnellstartanleitungen konzentrieren sich auf die Verwendung offizieller Zabbix-Vorlagen für die Überwachung.
Eine Vorlage ist eine vorkonfigurierte Sammlung von Überwachungselementen.
Mit einer Vorlage können Sie die Überwachung skalieren: Sie können eine Vorlage an einem einzelnen Überwachungsziel (Gerät, Anwendung usw.) testen, anpassen und anschließend auf mehrere ähnliche Überwachungsziele anwenden.

Vorlagen eignen sich zwar gut für die Überwachung ohne weitere Konfiguration, möglicherweise möchten Sie ihre Elemente jedoch auch anpassen.

Weitere Informationen zu den grundlegenden Elementen einer Vorlage finden Sie unter Grundlegende Konfiguration.

Grundkonfiguration

Diese Anleitungen führen in die grundlegenden Elemente der Zabbix-Konfiguration ein:

- Anmeldung und Konfiguration des Benutzers
- Neuer Host
- Neuer Datenpunkt
- Neuer Auslöser
- Empfang von Problembenachrichtigungen
- Neue Vorlage

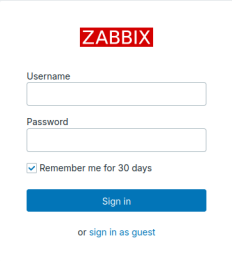
Anmeldung und Konfiguration des Benutzers

Übersicht

In diesem Abschnitt erfahren Sie, wie Sie sich bei Zabbix anmelden und einen Systembenutzer einrichten.

Anmeldung

Öffnen Sie die URL des Zabbix Frontend in einem Browser: - Für Apache: http://<server_ip_or_name>/zabbix - Für NGINX: http://<server_ip_or_name>



Dies ist der Zabbix-Begrüßungsbildschirm. Geben Sie den Benutzernamen **Admin** mit dem Passwort **zabbix** ein, um sich als Zabbix-Superuser anzumelden. Der Zugriff auf alle Menübereiche wird gewährt.

Aus Sicherheitsgründen wird dringend empfohlen, das standardmäßige Passwort für das Admin-Konto unmittelbar nach der ersten Anmeldung zu ändern.

Dauerhafte Anmeldung

Um bis zu 30 Tage angemeldet zu bleiben, wählen Sie vor dem Klicken auf *Anmelden* die Option *30 Tage merken* aus.

30 Tage merken aktiviert:

- Ihre Sitzung bleibt 30 Tage lang aktiv.
- *Automatische Abmeldung* wird außer Kraft gesetzt, sodass Sie bis zum Ablauf des Zeitraums angemeldet bleiben.
- Bei zukünftigen Besuchen innerhalb von 30 Tagen werden Sie automatisch angemeldet, ohne Ihre Zugangsdaten erneut eingeben zu müssen.

30 Tage merken deaktiviert:

- Eine zuvor aktivierte automatische Anmeldung wird entfernt.
- Die Sitzung läuft gemäß dem konfigurierten Intervall für *Automatische Abmeldung* ab.

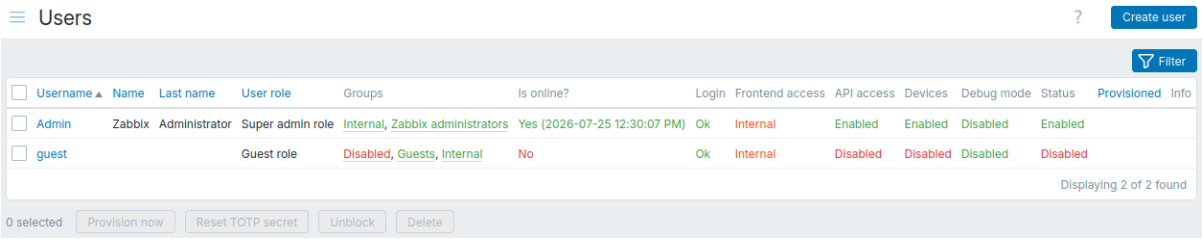
Schutz vor Brute-Force-Angriffen

Bei fünf aufeinanderfolgenden fehlgeschlagenen Anmeldeversuchen pausiert die Zabbix-Oberfläche 30 Sekunden lang, um Brute-Force- und Wörterbuchangriffe zu verhindern.

Die IP-Adresse eines fehlgeschlagenen Anmeldeversuchs wird nach einer erfolgreichen Anmeldung angezeigt.

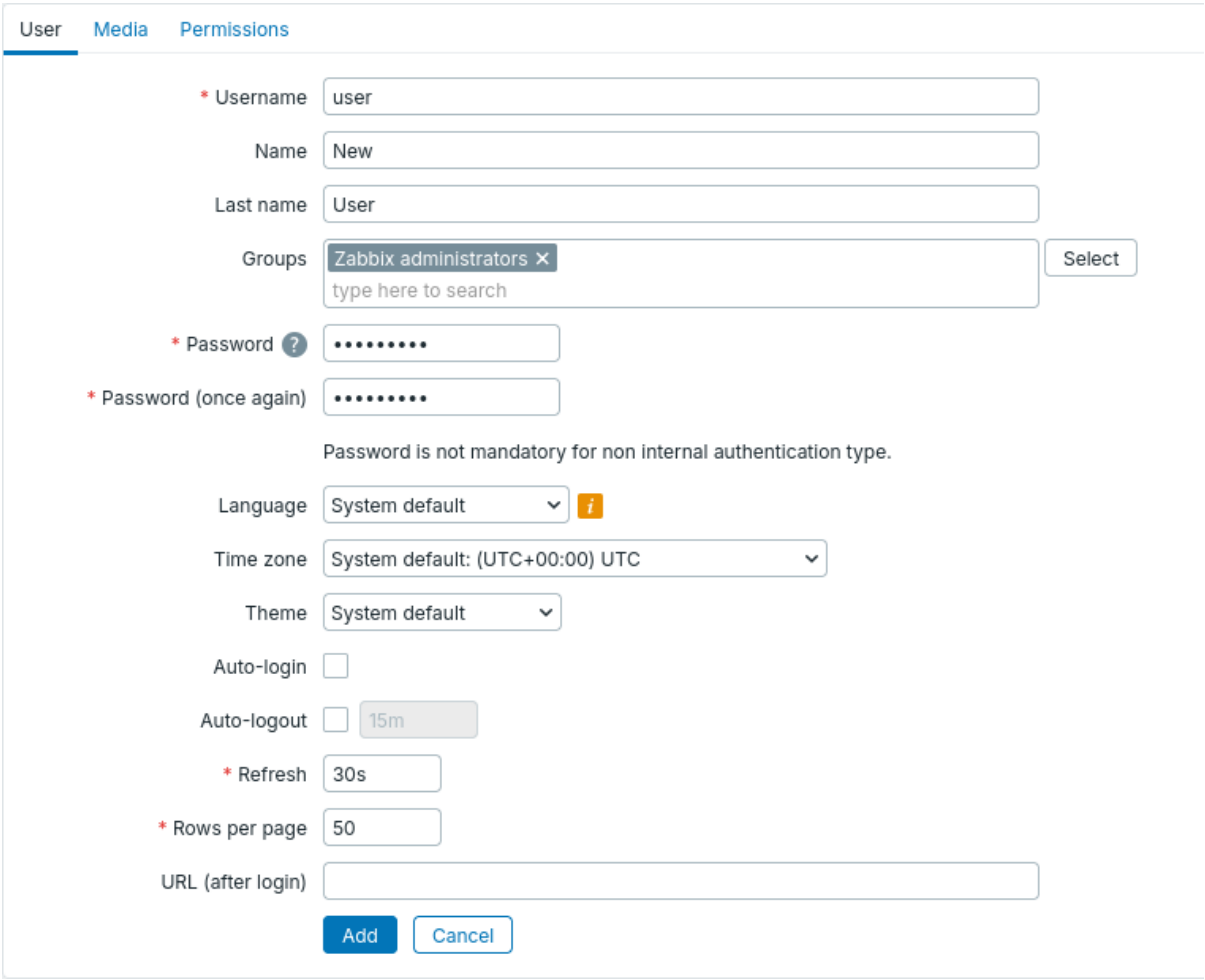
Hinzufügen eines Benutzers

Um Informationen zu Benutzern anzuzeigen, gehen Sie im vertikalen Menü der Seitenleiste zu *Users* > *Users*.



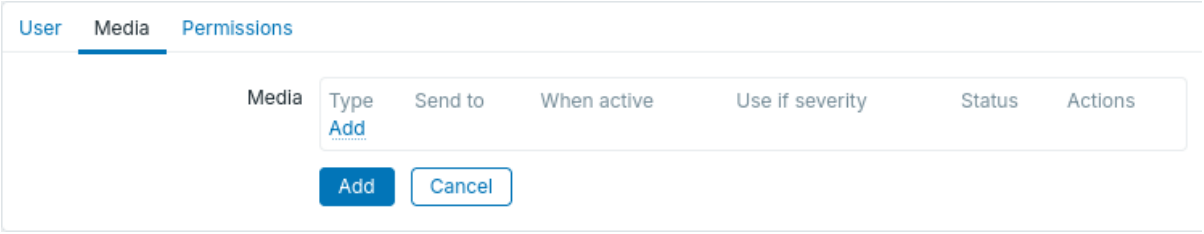
Um einen neuen Benutzer hinzuzufügen, wählen Sie oben rechts *Create user* aus.

Achten Sie im Formular für den neuen Benutzer darauf, den Benutzer einer der vorhandenen Benutzergruppen zuzuordnen, zum Beispiel 'Zabbix administrators'.



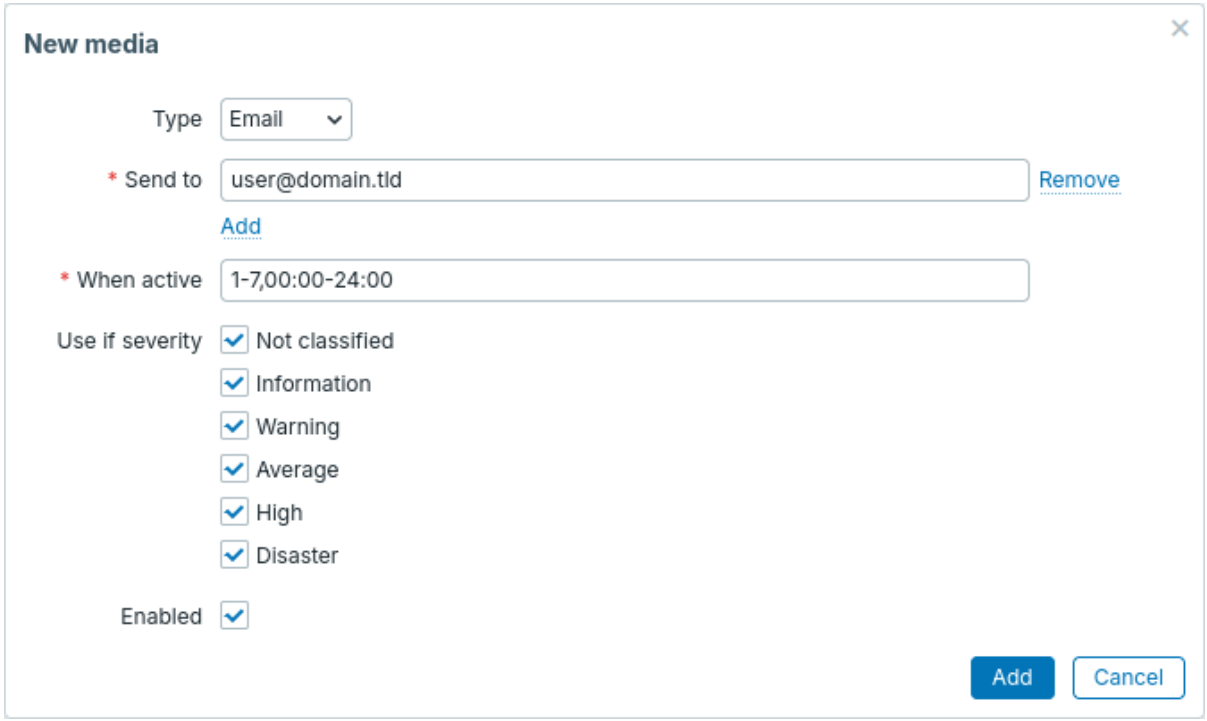
Alle Pflichtfelder sind mit einem roten Sternchen markiert. Weitere Informationen zu den Eingabefeldern in diesem Konfigurationsformular finden Sie auf der Seite Benutzereinstellungen.

Standardmäßig sind für neue Benutzer keine Medien (Methoden zur Zustellung von Benachrichtigungen) definiert. Um eines zu erstellen, gehen Sie zur Registerkarte *Media* und klicken Sie auf *Add*.



Geben Sie im Popup die E-Mail-Adresse des Benutzers ein.
 Alle Medientypen sind standardmäßig deaktiviert und können unter *Alerts* > *Media types* aktiviert werden.

Sie können einen Zeitraum angeben, in dem das Medium aktiv ist (eine Beschreibung des Formats finden Sie auf der Seite Zeitraumspezifikation). Standardmäßig ist ein Medium immer aktiv. Sie können außerdem Auslöser-Schweregrade anpassen, für die das Medium aktiv sein soll, lassen Sie jedoch vorerst alle aktiviert.



Klicken Sie auf *Add*, um das Medium zu speichern, und wechseln Sie dann zur Registerkarte *Permissions*.

Die Registerkarte *Permissions* hat das Pflichtfeld *Role*. Die Rolle bestimmt, welche Elemente der Benutzer im Frontend sehen kann und welche Aktionen er ausführen darf. Klicken Sie auf *Select* und wählen Sie eine der Rollen aus der Liste aus. Wählen Sie beispielsweise *Admin role*, um Zugriff auf alle Zabbix-Frontend-Bereiche außer *Administration* zu erlauben. Später können Sie Berechtigungen ändern oder weitere Benutzerrollen erstellen. Nach der Auswahl einer Rolle werden die Berechtigungen auf derselben Registerkarte angezeigt:

Users

User | Media | Permissions

* Role: Admin role ✕ | Select

User type: Admin

Permissions

User group permissions

Group	Type	Permissions
All groups	Hosts	None
All groups	Templates	None

Proxies: None

Proxy groups: None

Permissions can be assigned for user groups only.

Access to UI elements

Dashboards: Dashboards

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

APM: Traces | Metrics | Logs

Services: Services | SLA | SLA report

Inventory: Overview | Hosts

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

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

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

User profile settings

Redirect URL after login

Access to services

Read-write access to services: None

Read-only access to services: All

Access to modules

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

Access to API

Disabled

Devices

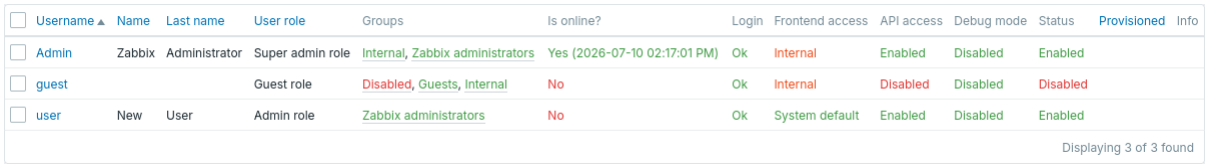
Enabled | Manage own devices

Access to actions

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

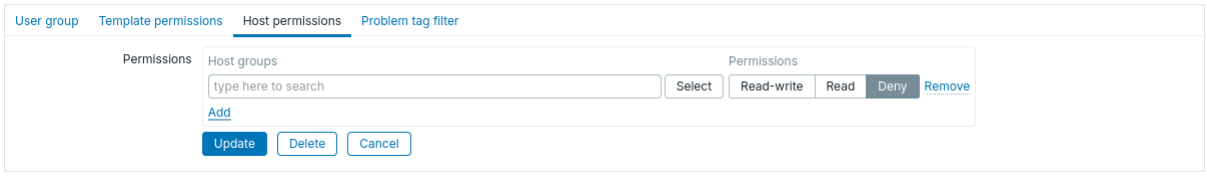
Add | Cancel

Klicken Sie im Formular mit den Benutzereigenschaften auf *Add*, um den Benutzer zu speichern. Der neue Benutzer wird in der Benutzerliste angezeigt.

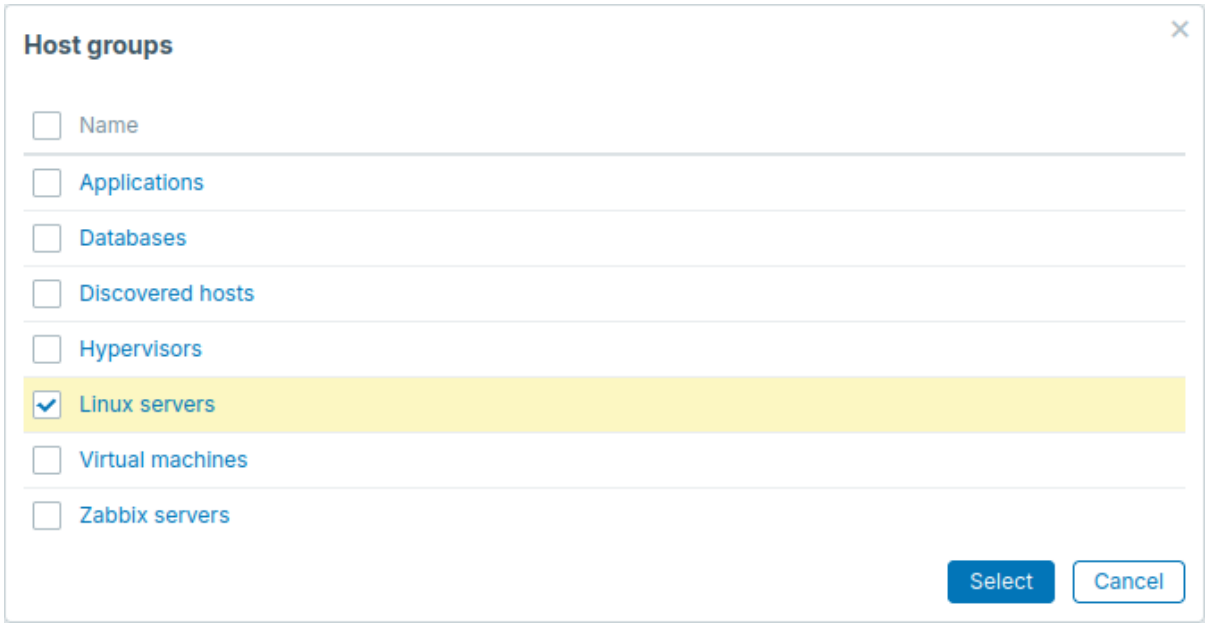


Hinzufügen von Berechtigungen

Standardmäßig hat ein neuer Benutzer keine Berechtigungen für den Zugriff auf Hosts und Vorlagen. Um dem Benutzer Rechte zuzuweisen, klicken Sie in der Spalte *Groups* auf die Gruppe des Benutzers (in diesem Fall - *Zabbix administrators*). Wechseln Sie im Eigenschaftenformular *User groups* zur Registerkarte *Host permissions*, um Berechtigungen für Hostgruppen zuzuweisen. Klicken Sie auf Add, damit das Auswahlfeld für die Hostgruppe angezeigt wird:



Klicken Sie dann neben dem Feld auf *Select*, um die Liste der Hostgruppen anzuzeigen. Dieser Benutzer soll nur Lesezugriff auf die Gruppe *Linux servers* haben. Markieren Sie daher das entsprechende Kontrollkästchen in der Liste und klicken Sie zur Bestätigung Ihrer Auswahl auf *Select*.



Klicken Sie auf die Schaltfläche *Read*, um die Berechtigungsstufe festzulegen, und anschließend auf *Update*, um die an der Konfiguration der Benutzergruppe vorgenommenen Änderungen zu speichern.

Um Berechtigungen für Vorlagen zuzuweisen, müssen Sie zur Registerkarte *Template permissions* wechseln und Vorlagengruppen angeben. Die Schritte sind identisch mit der Zuweisung von Berechtigungen für Hostgruppen. Eine Übersicht über Vorlagen finden Sie im Abschnitt Neue Vorlage dieses Quickstarts.

> **Attention:**
> In Zabbix werden Zugriffsrechte auf Hosts und Vorlagen Benutzergruppen zugewiesen, nicht einzelnen Benutzern.

Fertig! Sie können sich nun mit den Zugangsdaten des neuen Benutzers anmelden.

Neuer Host

Übersicht

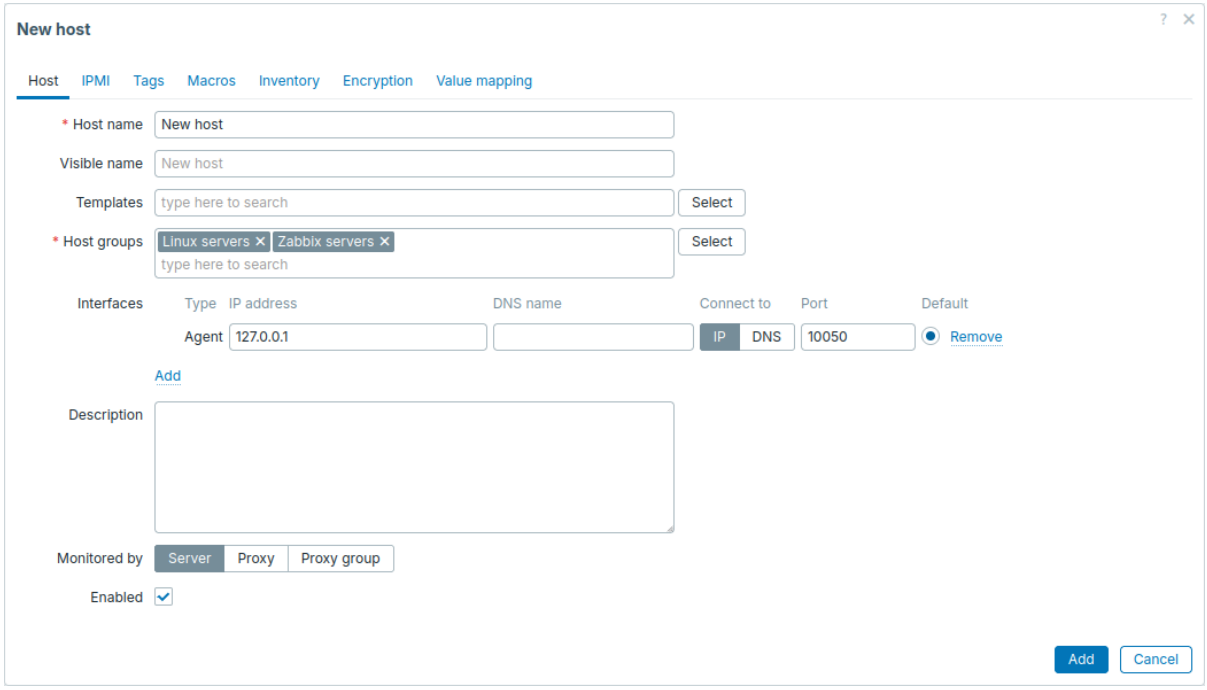
In diesem Abschnitt erfahren Sie, wie Sie einen neuen Host einrichten.

Ein Host in Zabbix ist eine vernetzte Einheit (physisch, virtuell), die Sie überwachen möchten. Die Definition dessen, was in Zabbix ein ”Host” sein kann, ist recht flexibel. Es kann sich um einen physischen Server, einen Netzwerkswitch, eine virtuelle Maschine oder eine Anwendung handeln.

Host hinzufügen

Informationen zu den konfigurierten Hosts in Zabbix sind im Menübereich *Datensammlung* > *Hosts* sowie im Menübereich *Überwachung* > *Hosts* verfügbar. Es ist bereits ein vordefinierter Host mit dem Namen „Zabbix server“ vorhanden, aber wir möchten lernen, wie ein weiterer Host hinzugefügt wird.

Um einen neuen Host hinzuzufügen, klicken Sie auf *Host erstellen*. Ein Host-Konfigurationsformular wird geöffnet.



Alle Pflichtfelder sind mit einem roten Sternchen gekennzeichnet.

Folgende Mindestangaben sind erforderlich:

Hostname

- Geben Sie einen Hostnamen ein. Alphanumerische Zeichen, Leerzeichen, Punkte, Bindestriche und Unterstriche sind zulässig.

Hostgruppen

- Wählen Sie eine oder mehrere vorhandene Gruppen aus, indem Sie auf die Schaltfläche *Auswählen* klicken, oder geben Sie den Namen einer nicht vorhandenen Gruppe ein, um eine neue Gruppe zu erstellen.

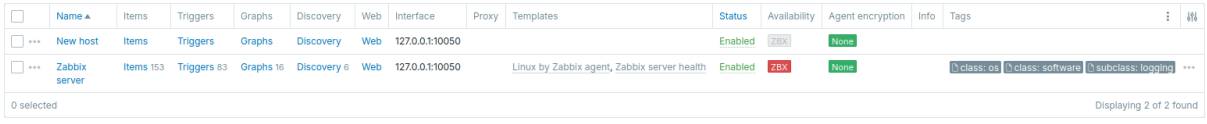
Note:
Alle Zugriffsberechtigungen werden Hostgruppen und nicht einzelnen Hosts zugewiesen. Daher muss ein Host mindestens einer Gruppe angehören.

Schnittstellen: IP-Adresse

- Obwohl dieses Feld technisch nicht erforderlich ist, wird eine Host-Schnittstelle für die Erfassung bestimmter Metriken benötigt. Um passive Prüfungen mit dem Zabbix Agent zu verwenden, geben Sie in diesem Feld die IP-Adresse oder den DNS-Namen des Agents an. Beachten Sie, dass Sie außerdem die IP-Adresse oder den DNS-Namen des Zabbix Servers in der Direktive „Server“ der Agent-Konfigurationsdatei angeben müssen. Wenn Zabbix Agent und Zabbix Server auf demselben Rechner installiert sind, müssen Sie an beiden Stellen dieselbe IP-Adresse bzw. denselben DNS-Namen angeben.

Weitere Optionen können unverändert mit ihren Standardwerten belassen werden.

Klicken Sie anschließend auf *Hinzufügen*. Ihr neuer Host sollte in der Hostliste sichtbar sein.



Die Spalte *Verfügbarkeit* enthält für jede Schnittstelle Indikatoren zur Hostverfügbarkeit. Wenn eine Schnittstelle für den Zabbix Agent definiert ist, können wir anhand des Verfügbarkeitssymbols des Agents (mit *ZBX* gekennzeichnet) die Verfügbarkeit des Hosts bestimmen:

- ZBX - Der Hoststatus wurde noch nicht ermittelt; es wurde noch keine Metrikprüfung durchgeführt.
- ZBX - Der Host ist verfügbar, eine Metrikprüfung war erfolgreich.
- ZBX - Der Host ist nicht verfügbar, eine Metrikprüfung ist fehlgeschlagen (bewegen Sie den Mauszeiger über das Symbol, um die Fehlermeldung anzuzeigen). Möglicherweise liegt ein Kommunikationsfehler vor, der beispielsweise durch falsche Zugangsdaten der Schnittstelle verursacht wurde. Überprüfen Sie, ob der Zabbix Server ausgeführt wird, und versuchen Sie außerdem, die Seite später zu aktualisieren.

Neuer Datenpunkt

Übersicht

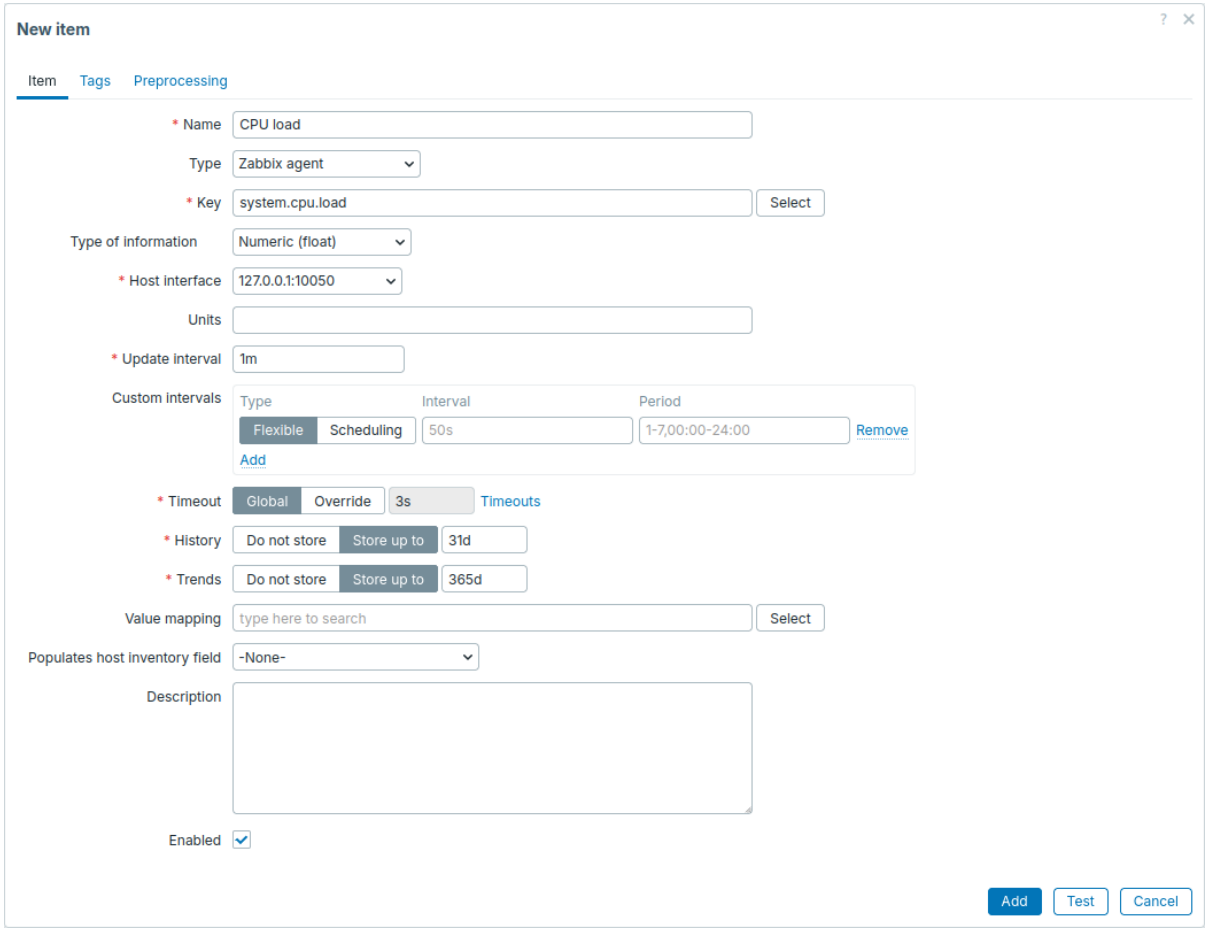
In diesem Abschnitt erfahren Sie, wie Sie einen Datenpunkt einrichten.

Datenpunkte sind die Grundlage für das Sammeln von Daten in Zabbix. Ohne Datenpunkte gibt es keine Daten - denn nur ein Datenpunkt definiert eine einzelne Metrik oder die Art der Daten, die von einem Host erfasst werden sollen.

Hinzufügen eines Datenpunkts

Alle Datenpunkte sind um Hosts herum gruppiert. Deshalb gehen wir zur Konfiguration eines Beispieldatenpunkts zu *Datenerfassung* > *Hosts* und suchen den von uns erstellten ”New host”.

Klicken Sie in der Zeile von ”New host” auf den Link *Datenpunkte* und anschließend auf *Datenpunkt erstellen*. Ein Konfigurationsformular für den Datenpunkt wird geöffnet.



Alle Pflichtfelder sind mit einem roten Sternchen markiert.

Für unseren Beispieldatenpunkt sind die folgenden Angaben erforderlich:

Name

- Geben Sie als Wert *CPU load* ein. Dies ist der Name des Datenpunkts, der in Listen und an anderen Stellen angezeigt wird.

Key

- Geben Sie manuell als Wert *system.cpu.load* ein. Dies ist der technische Name eines Datenpunkts, der den Typ der zu erfassenden Informationen identifiziert. Dieser spezielle Key ist nur einer der vordefinierten Keys, die mit dem Zabbix Agent geliefert werden.

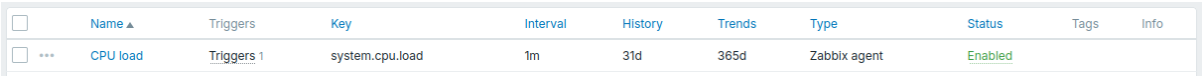
Type of information

- Dieses Attribut definiert das Format der erwarteten Daten. Für den Key *system.cpu.load* wird dieses Feld automatisch auf *Numeric (float)* gesetzt.

Note:
Möglicherweise möchten Sie auch die Anzahl der Tage reduzieren, für die der Datenpunkthistorie gespeichert wird, auf 7 oder 14. Dies ist eine gute Praxis, um die Datenbank von der Speicherung vieler historischer Werte zu entlasten.

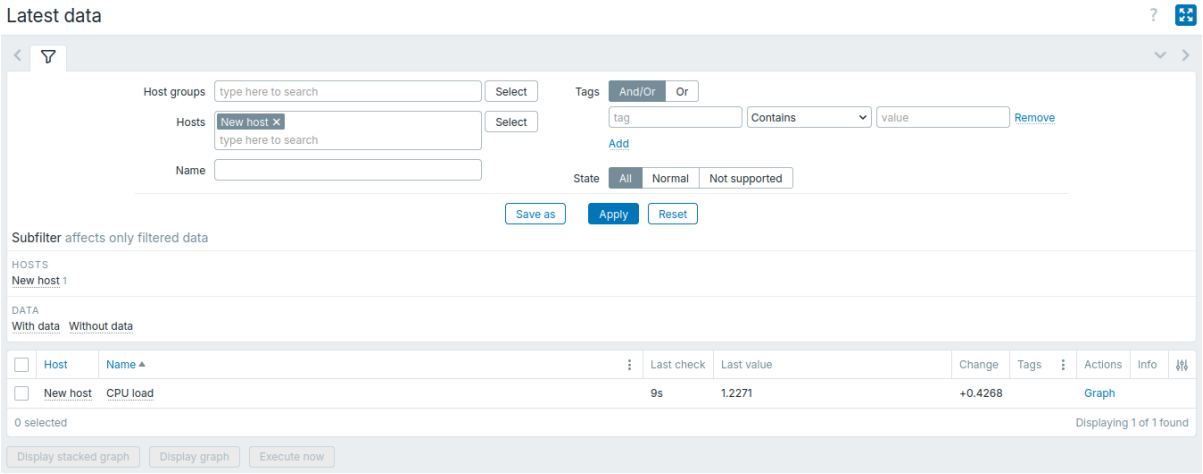
Andere Optionen können unverändert mit ihren Standardwerten belassen werden.

Wenn Sie fertig sind, klicken Sie auf *Add*. Der neue Datenpunkt sollte in der Datenpunktliste erscheinen.



Daten anzeigen

Wenn ein Datenpunkt definiert ist, möchten Sie vielleicht prüfen, ob er tatsächlich Daten sammelt. Gehen Sie dazu zu *Monitoring* > *Neueste Daten*, wählen Sie im Filter 'New host' aus und klicken Sie auf *Anwenden*.



Beachten Sie, dass es bis zu 60 Sekunden dauern kann, bis die ersten Daten eintreffen. So oft liest der Server standardmäßig Konfigurationsänderungen ein und übernimmt neue Datenpunkte zur Ausführung.

Wenn in der Spalte 'Change' kein Wert angezeigt wird, wurde möglicherweise bisher nur ein einziger Wert empfangen. Warten Sie 30 Sekunden, bis ein weiterer Wert eintrifft.

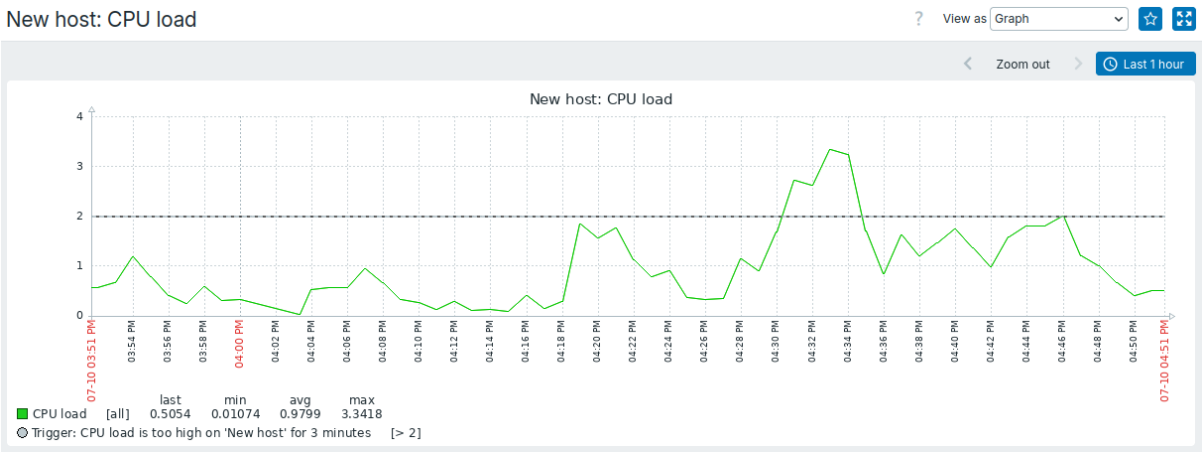
Wenn Sie keine Informationen zum Datenpunkt wie im Screenshot sehen, stellen Sie sicher, dass:

- Sie die Felder 'Key' und 'Type of information' des Datenpunkts genau wie im Screenshot ausgefüllt haben;
- sowohl der Agent als auch der Server ausgeführt werden;
- der Host-Status 'Monitored' ist und das Verfügbarkeits-Symbol grün ist;
- der im Host-Filter ausgewählte Host korrekt ist;
- der Datenpunkt aktiviert ist.

Graphen

Wenn der Datenpunkt eine Weile funktioniert, ist es möglicherweise an der Zeit, etwas Visuelles zu sehen. Einfache Graphen sind für jeden überwachten numerischen Datenpunkt ohne zusätzliche Konfiguration verfügbar. Diese Graphen werden zur Laufzeit generiert.

Um den Graphen anzuzeigen, gehen Sie zu *Überwachung* > *Neueste Daten* und klicken Sie auf den Link *Graph* neben dem Datenpunkt.



Neuer Auslöser

Übersicht

In diesem Abschnitt erfahren Sie, wie Sie einen Auslöser einrichten.

Datenpunkte erfassen nur Daten. Um eingehende Daten automatisch auszuwerten, müssen wir Auslöser definieren. Ein Auslöser enthält einen Ausdruck, der einen Schwellenwert dafür festlegt, was für die Daten ein akzeptables Niveau ist.

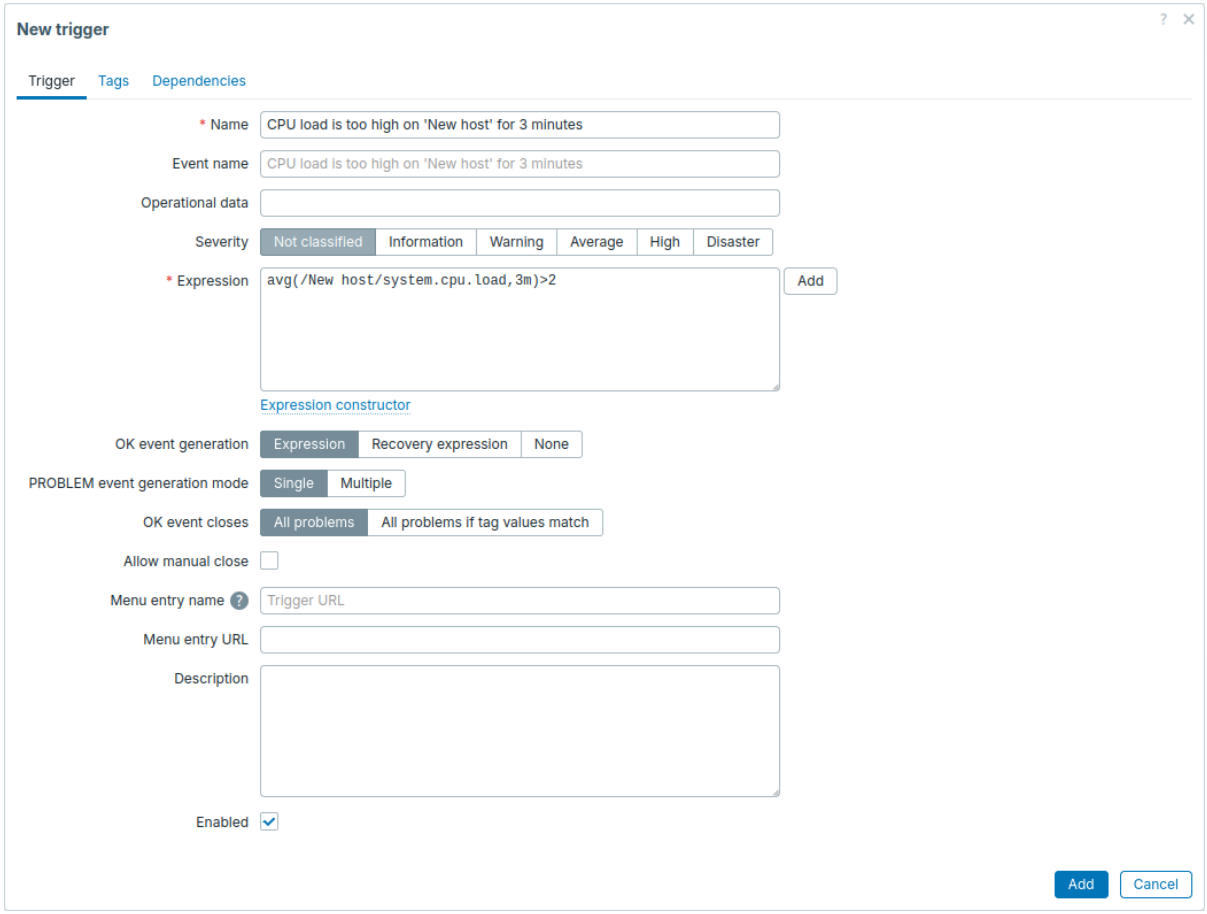
Wenn dieser Wert durch die eingehenden Daten überschritten wird, wird ein Auslöser ”ausgelöst” oder wechselt in den Status ’Problem’ - dies weist darauf hin, dass etwas passiert ist, das möglicherweise Aufmerksamkeit erfordert. Wenn der Wert wieder akzeptabel ist, kehrt der Auslöser in den Status ’Ok’ zurück.

Hinzufügen eines Auslösers

Um einen Auslöser für unseren Datenpunkt zu konfigurieren:

1. Gehen Sie zu *Datenverarbeitung* > *Hosts*.
2. Klicken Sie in der Zeile von ’New host’ auf den Link *Auslöser*.
3. Klicken Sie auf *Auslöser erstellen*.

Ein Formular zur Konfiguration des Auslösers wird geöffnet.



Für unseren Auslöser sind hier die folgenden wesentlichen Informationen einzugeben:

Name

- Geben Sie als Wert *CPU load is too high on 'New host' for 3 minutes* ein. Dies ist der Name des Auslösers, der in Listen und an anderen Stellen angezeigt wird.

Ausdruck

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

Dies ist der Ausdrück des Auslösers. Stellen Sie sicher, dass der Ausdruck korrekt eingegeben wird, bis hin zum letzten Symbol. Der Datenpunkt-Schlüssel hier (`system.cpu.load`) wird verwendet, um auf den Datenpunkt zu verweisen. Dieser spezielle Ausdruck besagt im Wesentlichen, dass der Problem-Schwellenwert überschritten wird, wenn der Durchschnittswert der CPU-Auslastung über 3 Minuten größer als 2 ist. Weitere Informationen finden Sie in der Syntax von Ausdrücken für Auslöser.

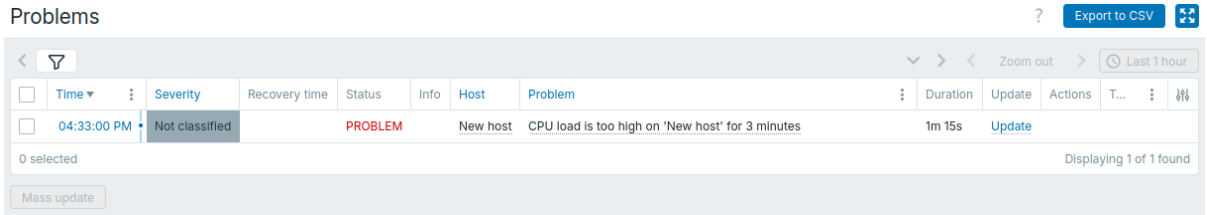
Andere Optionen können unverändert mit ihren Standardwerten belassen werden.

Wenn Sie fertig sind, klicken Sie auf *Hinzufügen*. Der neue Auslöser sollte in der Auslöserliste erscheinen.

Anzeigen des Auslöserstatus

Wenn ein Auslöser definiert ist, möchten Sie möglicherweise seinen Status anzeigen.

Wenn die CPU-Auslastung den in dem Auslöser definierten Schwellenwert überschritten hat, wird das Problem unter *Monitoring* > *Probleme* angezeigt.



Das Blinken in der Statusspalte weist auf eine kürzliche Änderung des Auslöserstatus hin, die in den letzten 30 Minuten stattgefunden hat.

Empfang einer Problembenachrichtigung

Übersicht

In diesem Abschnitt erfahren Sie, wie Sie Benachrichtigungen in Zabbix einrichten.

Da Datenpunkte Daten sammeln und Auslöser so konzipiert sind, dass sie bei Problemsituationen ”auslösen”, wäre es auch nützlich, einen Benachrichtigungsmechanismus einzurichten, um über wichtige Ereignisse zu informieren, selbst wenn das Zabbix Frontend nicht aktiv überprüft wird.

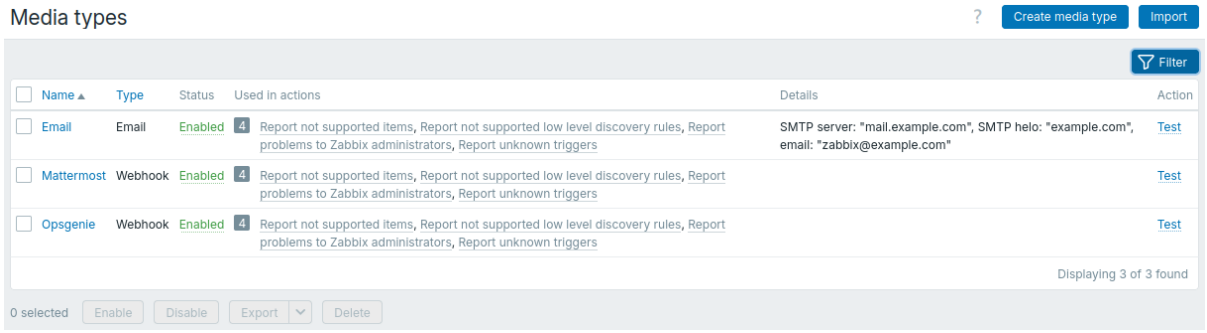
Genau das leisten Benachrichtigungen.

Da E-Mail die beliebteste Zustellmethode für Problembenachrichtigungen ist, lernen wir, wie man eine E-Mail-Benachrichtigung einrichtet.

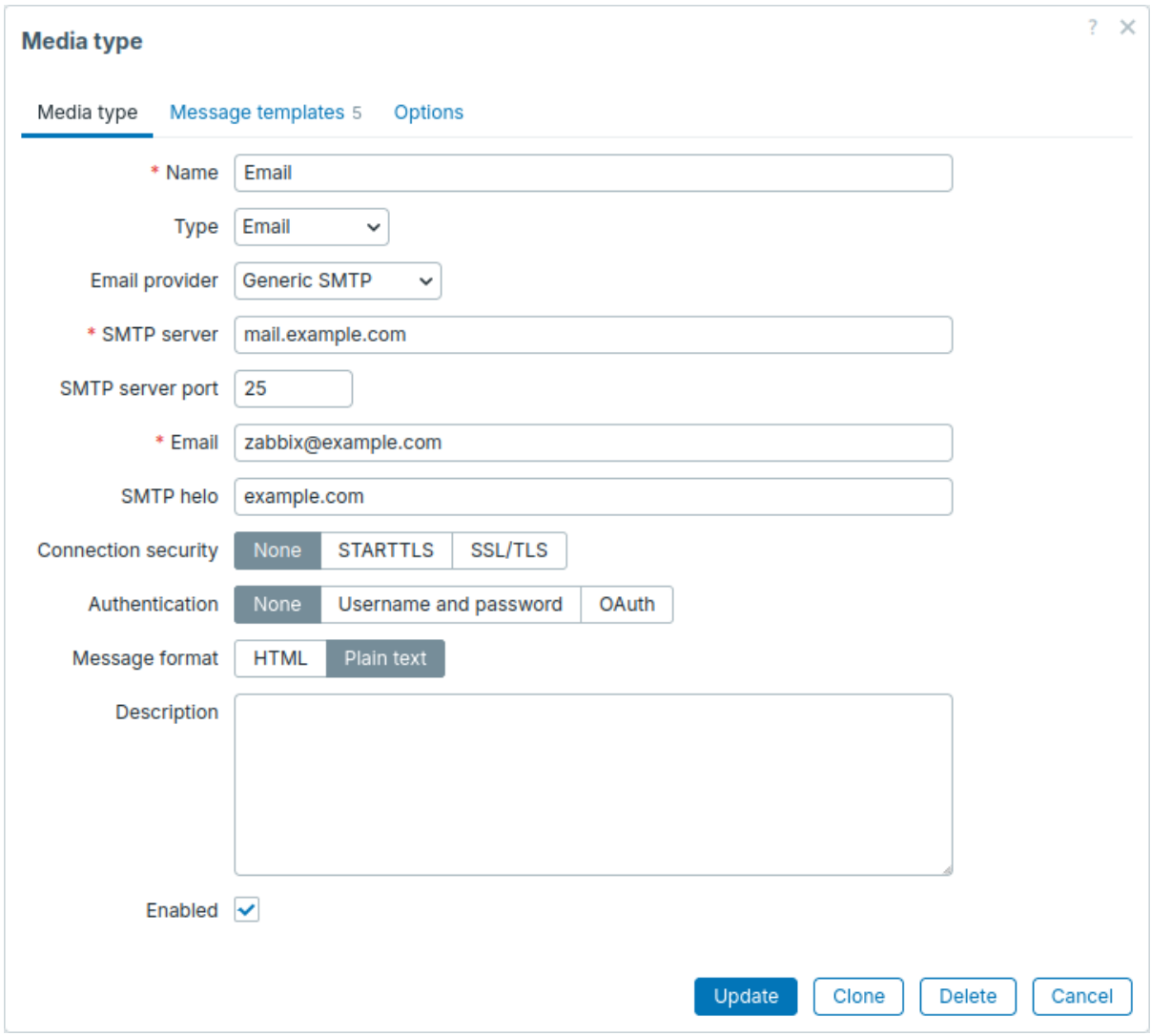
E-Mail-Einstellungen

Zunächst gibt es in Zabbix mehrere vordefinierte Benachrichtigungs-Übermittlungsarten. E-Mail ist eine davon.

Um die E-Mail-Einstellungen zu konfigurieren, gehen Sie zu *Alerts* > *Media types* und klicken Sie in der Liste der vordefinierten Medientypen auf *Email*.



Ein Konfigurationsformular für die E-Mail-Einstellungen wird geöffnet.



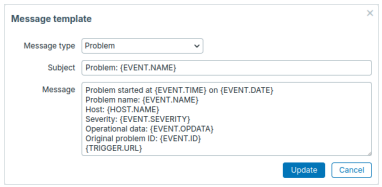
Alle Pflichtfelder sind mit einem roten Sternchen markiert.

Legen Sie im Tab *Media type* die Werte für *SMTP server*, *SMTP helo* und *Email* passend zu Ihrer Umgebung fest.

> **Note:**
> Der Wert im Feld *Email* wird als 'From'-Adresse für die von Zabbix gesendeten Benachrichtigungen verwendet.

Als Nächstes muss der Inhalt der Problemnachricht definiert werden. Der Inhalt wird mithilfe einer Nachrichtenvorlage festgelegt, die im Tab *Message templates* konfiguriert wird.

Klicken Sie auf *Add*, um eine Nachrichtenvorlage zu erstellen, und wählen Sie *Problem* als Nachrichtentyp aus.

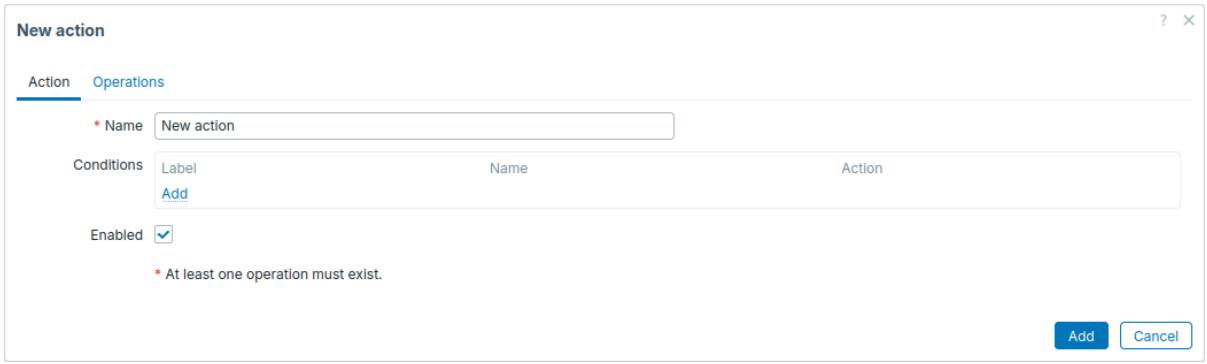


Klicken Sie auf *Add*, wenn Sie fertig sind, und speichern Sie das Formular.

Nun haben Sie *Email* als funktionierenden Medientyp konfiguriert. Der Medientyp muss außerdem Benutzern zugeordnet werden, indem spezifische Zustelladressen definiert werden (wie wir es bei der Konfiguration eines neuen Benutzers getan haben), andernfalls wird er nicht verwendet.

Neue Aktion

Das Senden von Benachrichtigungen ist eine der Aufgaben, die Aktionen in Zabbix übernehmen. Um eine Benachrichtigung einzurichten, gehen Sie daher zu *Alerts* > *Actions* > *Trigger actions* und klicken Sie auf *Create action*.

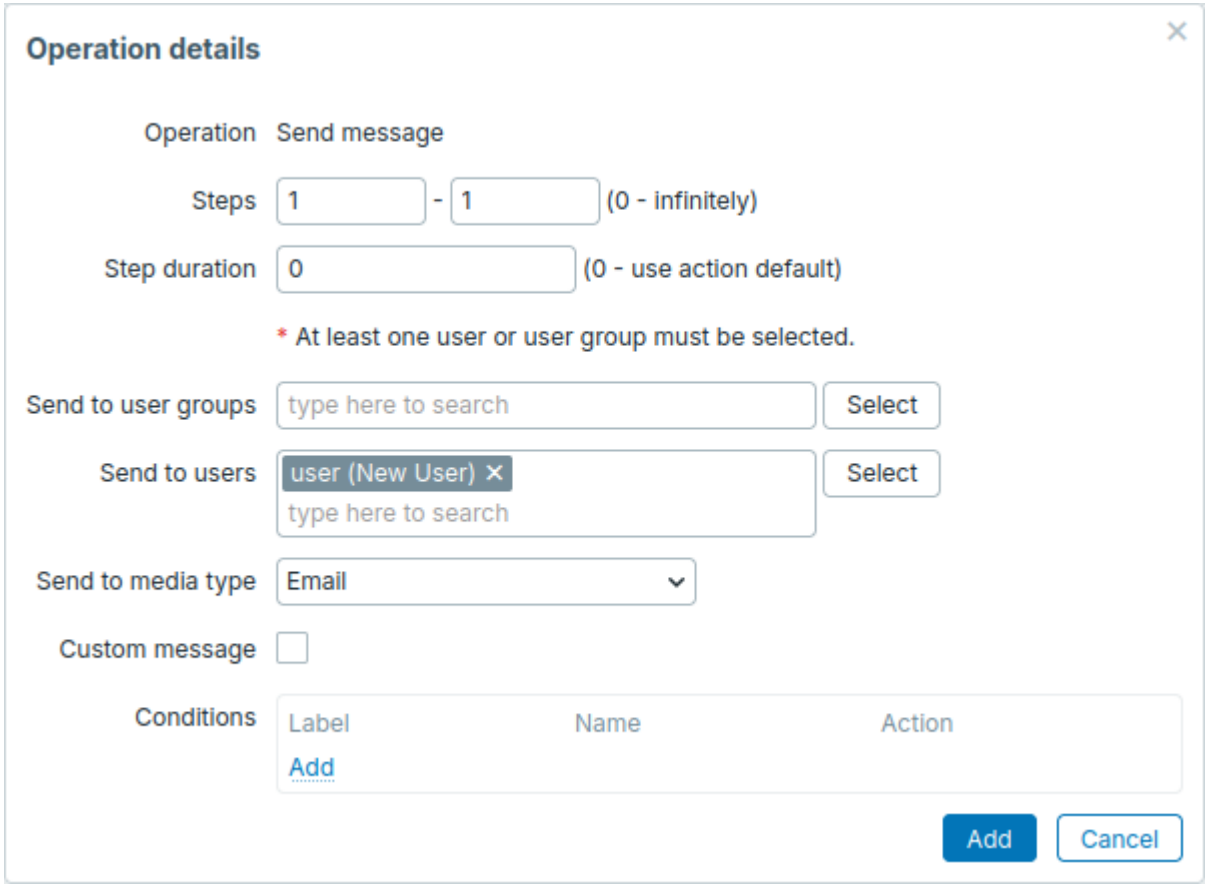


Alle Pflichtfelder sind mit einem roten Sternchen markiert.

Geben Sie in diesem Formular einen Namen für die Aktion ein.

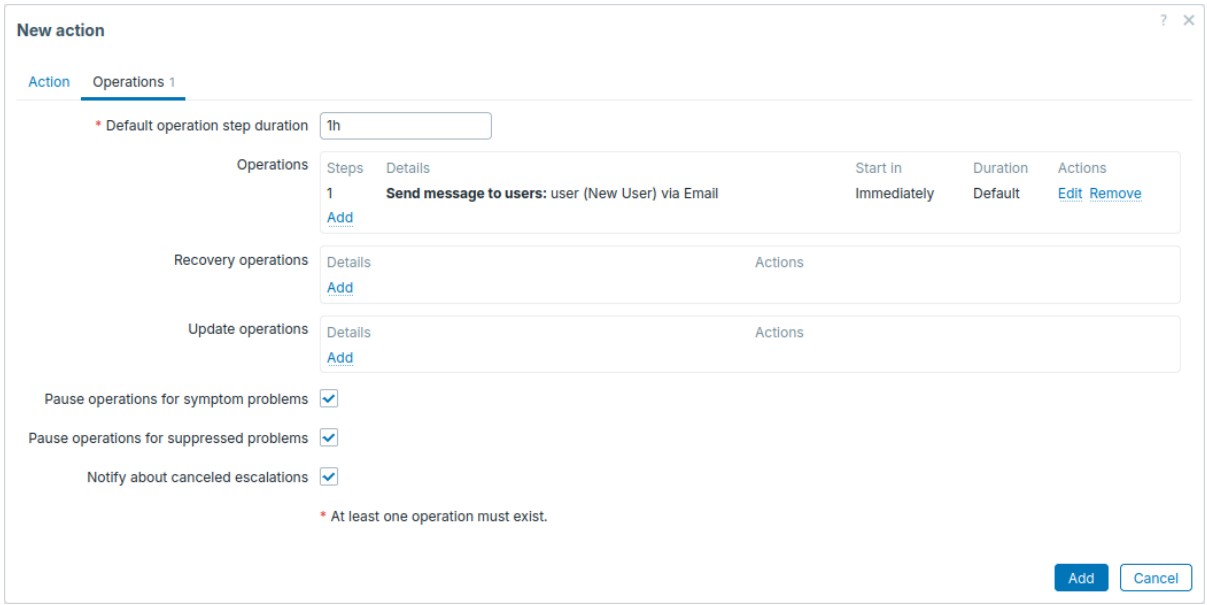
Im einfachsten Fall, wenn wir keine weiteren spezifischen Bedingungen hinzufügen, wird die Aktion bei jeder Auslöser-Änderung von 'Ok' zu 'Problem' ausgeführt.

Wir müssen noch festlegen, was die Aktion tun soll - und das geschieht im Tab *Operations*. Klicken Sie im Block *Operations* auf *Add*, wodurch ein neues Operationsformular geöffnet wird.



Alle Pflichtfelder sind mit einem roten Sternchen markiert.

Klicken Sie hier im Block *Send to Users* auf *Select* und wählen Sie den von uns definierten Benutzer ('user') aus. Wählen Sie "Email" als Wert für *Send to media type*. Wenn Sie damit fertig sind, klicken Sie auf *Add*, und die Operation sollte hinzugefügt werden:



Das ist alles für eine einfache Aktionskonfiguration, also klicken Sie im Aktionsformular auf *Add*.

Empfang von Benachrichtigungen

Jetzt, da das Senden von Benachrichtigungen konfiguriert ist, wäre es schön, tatsächlich eine zu erhalten. Dazu können wir absichtlich die Last auf unserem Host erhöhen - damit unser Auslöser "auslöst" und wir eine Problembenachrichtigung erhalten.

Öffnen Sie die Konsole auf Ihrem Host und führen Sie Folgendes aus:

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

Sie können einen oder mehrere dieser Prozesse ausführen.

Gehen Sie nun zu *Monitoring* > *Latest data* und sehen Sie, wie die Werte von 'CPU Load' gestiegen sind. Denken Sie daran: Damit

unser Auslöser ”auslöst”, muss der Wert von ’CPU Load’ 3 Minuten lang über ’2’ liegen. Sobald das der Fall ist:

- unter *Monitoring* > *Problems* sollten Sie den Auslöser mit einem blinkenden `Problem`-Status sehen;
- Sie sollten eine Problembenachrichtigung per E-Mail erhalten.

Attention:
Wenn Benachrichtigungen nicht funktionieren:
- überprüfen Sie noch einmal, ob sowohl die E-Mail-Einstellungen als auch die Aktion korrekt konfiguriert wurden
- stellen Sie sicher, dass der von Ihnen erstellte Benutzer mindestens Leserechte auf dem Host hat, der das Ereignis erzeugt hat, wie im Schritt *Benutzer hinzufügen* beschrieben. Der Benutzer, der Teil der Benutzergruppe ’Zabbix administrators’ ist, muss mindestens Lesezugriff auf die Hostgruppe ’Linux servers’ haben, zu der unser Host gehört.
- Zusätzlich können Sie das Aktionsprotokoll unter *Reports* > *Action log* einsehen.

Neue Vorlage

Übersicht

In diesem Abschnitt erfahren Sie, wie Sie eine Vorlage einrichten.

Zuvor haben wir gelernt, wie man einen Datenpunkt und einen Auslöser einrichtet und wie man eine Problembenachrichtigung für den Host erhält.

Obwohl all diese Schritte an sich eine große Flexibilität bieten, kann es nach vielen Schritten aussehen, wenn sie beispielsweise für tausend Hosts erforderlich sind. Eine gewisse Automatisierung wäre hilfreich.

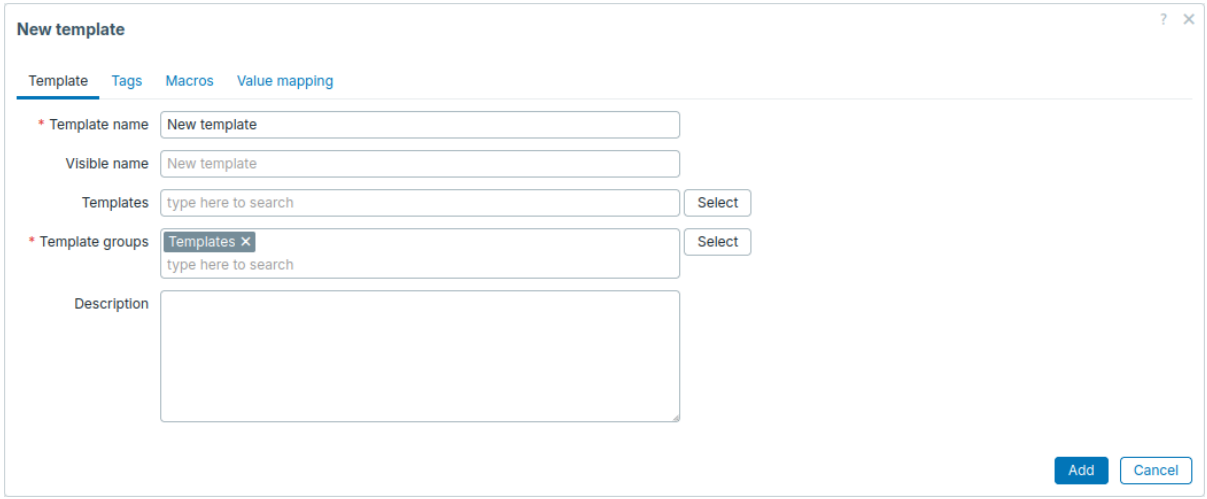
Hier kommen Vorlagen ins Spiel.

Vorlagen ermöglichen es Ihnen, nützliche Datenpunkte, Auslöser und andere Entitäten zu gruppieren, sodass diese immer wieder verwendet werden können, indem sie in einem einzigen Schritt auf Hosts angewendet werden.

Wenn eine Vorlage mit einem Host verknüpft wird, übernimmt der Host alle Entitäten der Vorlage. Somit kann im Grunde eine vorbereitete Sammlung von Prüfungen sehr schnell angewendet werden.

Vorlage hinzufügen

Um mit Vorlagen zu arbeiten, müssen wir zunächst eine erstellen. Dazu klicken Sie unter *Datenerfassung* > *Vorlagen* auf *Vorlage erstellen*. Daraufhin wird ein Konfigurationsformular für die Vorlage angezeigt.



Alle Pflichtfelder sind mit einem roten Sternchen markiert.

Die hier einzugebenden erforderlichen Parameter sind:

Vorlagenname

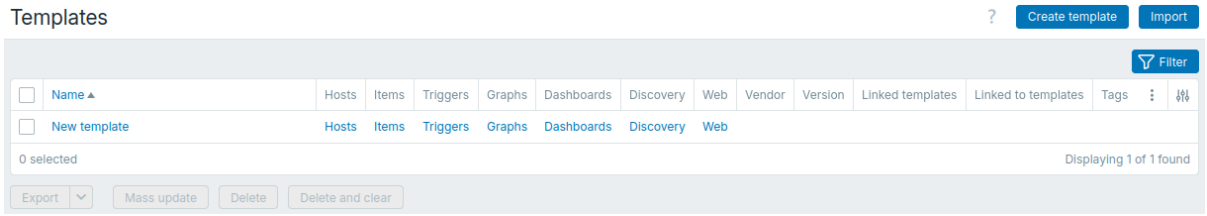
- Geben Sie einen Vorlagennamen ein. Alphanumerische Zeichen, Leerzeichen und Unterstriche sind zulässig.

Vorlagengruppen

- Wählen Sie eine oder mehrere Gruppen aus, indem Sie auf die Schaltfläche *Auswählen* klicken. Die Vorlage muss zu einer Gruppe gehören.

Note:
Zugriffsberechtigungen für Vorlagengruppen werden in der Konfiguration der Benutzergruppe auf der Registerkarte **Vorlagenberechtigungen** auf dieselbe Weise wie Host-Berechtigungen zugewiesen. Alle Zugriffsberechtigungen werden Gruppen und nicht einzelnen Vorlagen zugewiesen, deshalb ist es zwingend erforderlich, die Vorlage mindestens einer Gruppe hinzuzufügen.

Wenn Sie fertig sind, klicken Sie auf *Hinzufügen*. Ihre neue Vorlage sollte in der Vorlagenliste sichtbar sein. Sie können auch den Filter verwenden, um Ihre Vorlage zu finden.

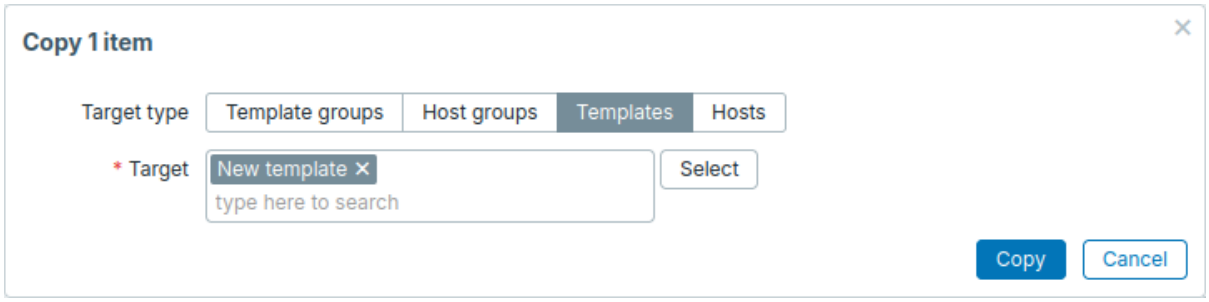


Wie Sie sehen, ist die Vorlage vorhanden, enthält jedoch noch nichts - keine Datenpunkte, Auslöser oder andere Elemente.

Datenpunkt zur Vorlage hinzufügen

So fügen Sie einen Datenpunkt zur Vorlage hinzu:

1. Gehen Sie zu *Datensammlung* > *Hosts*.
2. Suchen Sie in der Liste nach „New host“ und klicken Sie auf den Hyperlink *Datenpunkte*.
3. Aktivieren Sie in der Liste das Kontrollkästchen des Datenpunkts „CPU Load“.
4. Klicken Sie unterhalb der Liste auf *Kopieren*.
5. Wählen Sie die Registerkarte *Vorlagen* aus.
6. Wählen Sie die Vorlage aus, in die der Datenpunkt kopiert werden soll.



Alle Pflichtfelder sind mit einem roten Sternchen gekennzeichnet. 7. Klicken Sie auf *Kopieren*.

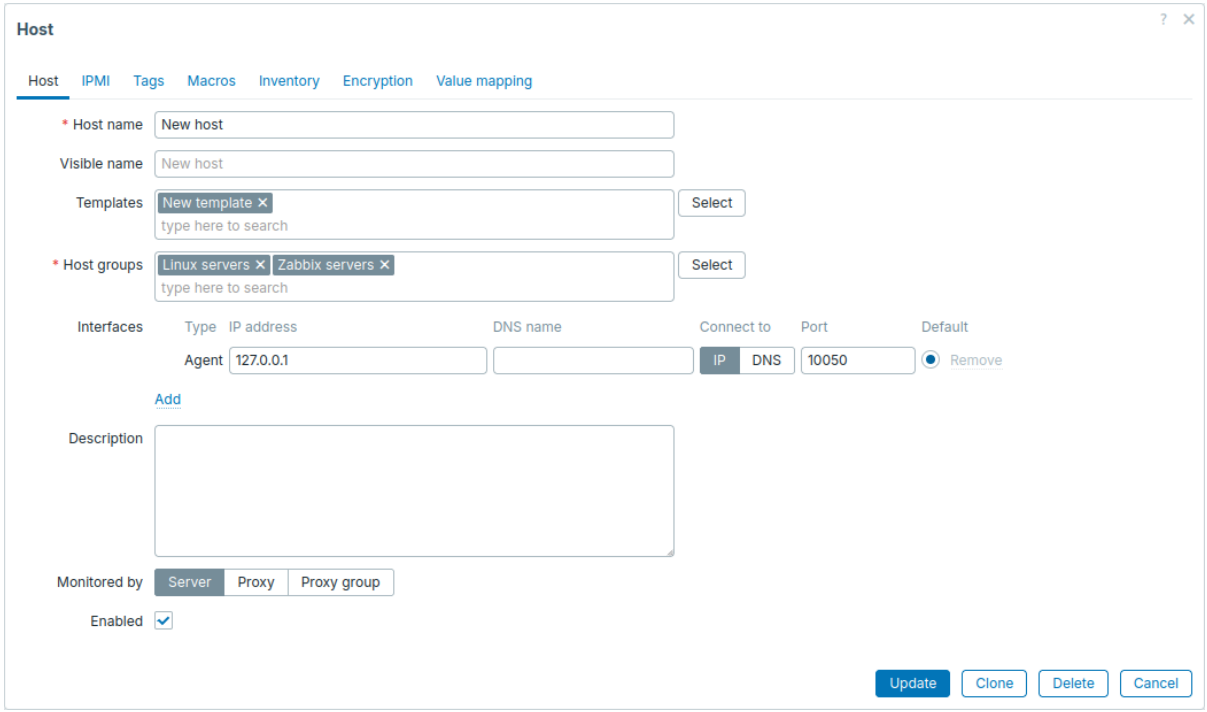
Wenn Sie nun zu *Datensammlung* > *Vorlagen* wechseln, sollte „New template“ einen neuen Datenpunkt enthalten.

Wir beschränken uns vorerst auf einen Datenpunkt. Auf ähnliche Weise können Sie jedoch weitere Datenpunkte, Auslöser oder andere Entitäten zur Vorlage hinzufügen, bis sie eine weitgehend vollständige Sammlung von Entitäten für den jeweiligen Zweck darstellt (Überwachung des Betriebssystems, Überwachung einer einzelnen Anwendung).

Vorlage mit Host verknüpfen

Nachdem eine Vorlage erstellt wurde, muss sie nur noch einem Host hinzugefügt werden. Gehen Sie dazu wie folgt vor:

1. Gehen Sie zu *Datensammlung* > *Hosts*.
2. Klicken Sie auf „Neuer Host“, um das Konfigurationsformular zu öffnen, und suchen Sie das Feld *Vorlagen*.
3. Beginnen Sie im Feld *Vorlagen* mit der Eingabe von *Neue Vorlage*. Der Name der neu erstellten Vorlage sollte in der Dropdown-Liste angezeigt werden.
4. Scrollen Sie nach unten, um sie auszuwählen. Sie wird nun im Feld *Vorlagen* angezeigt.



5. Klicken Sie im Formular auf *Aktualisieren*, um die Änderungen zu speichern.

Die Vorlage wurde nun zusammen mit allen darin enthaltenen Entitäten zum Host hinzugefügt.

Auf diese Weise kann sie auch auf jeden anderen Host angewendet werden. Alle Änderungen an Datenpunkten, Auslösern und anderen Entitäten auf Vorlagenebene werden an die Hosts weitergegeben, mit denen die Vorlage verknüpft ist.

Verknüpfen vordefinierter Vorlagen mit Hosts

Wie Sie vielleicht bemerkt haben, wird Zabbix mit einer Reihe vordefinierter Vorlagen für verschiedene Betriebssysteme, Geräte und Anwendungen ausgeliefert. Um sehr schnell mit dem Monitoring zu beginnen, können Sie die passende davon mit einem Host verknüpfen, sollten jedoch beachten, dass diese Vorlagen an Ihre Umgebung angepasst werden müssen. Einige Prüfungen sind möglicherweise nicht erforderlich, und die Abfrageintervalle können deutlich zu häufig sein.

Weitere Informationen zu Vorlagen sind verfügbar.

Apache web server überwachen

Um mit der Überwachung des Apache web server zu beginnen, folgen Sie diesem Ablauf:

1. Stellen Sie sicher, dass Sie Zugriff auf Apache-Statistiken haben.
2. Wenden Sie die Apache-Überwachungsvorlage an.
3. Sehen Sie sich die neuesten Daten an.

Um diese Anleitung zu verwenden, müssen Sie den Zabbix Server und das Web-Frontend installiert haben. Sie müssen in Zabbix als Standardbenutzer *Admin* angemeldet sein.

Zugriff auf Apache-Statistiken

Das Ziel dieses Schritts ist sicherzustellen, dass die Statistiken des Apache-Webservers von Zabbix gelesen werden können.

In Apache werden Echtzeitinformationen über die Leistung und Aktivität des Servers durch das integrierte Modul `mod_status` bereitgestellt. Wenn es aktiviert ist, erstellt es eine über das Web zugängliche Seite (normalerweise unter `/server-status`), auf der Leistungs- und Aktivitätsdaten angezeigt werden.

Um zu überprüfen, ob `mod_status` für den Apache-Webserver aktiviert ist, führen Sie Folgendes aus:

Auf Debian/Ubuntu	Auf RHEL-basierten Systemen
`apache2ctl -M | grep status`	`httpd -M | grep status`

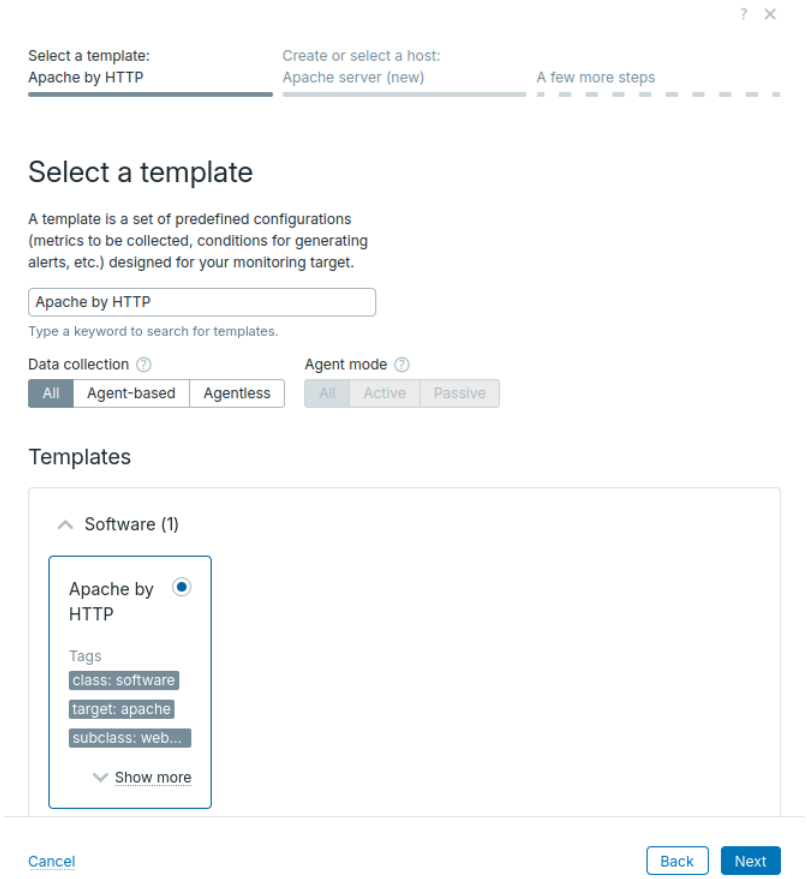
Wenn Sie "`status_module (shared)`" in der Ausgabe sehen, ist das Statusmodul bereits aktiviert. Andernfalls müssen Sie das Statusmodul aktivieren.

Apache-Überwachungsvorlage anwenden

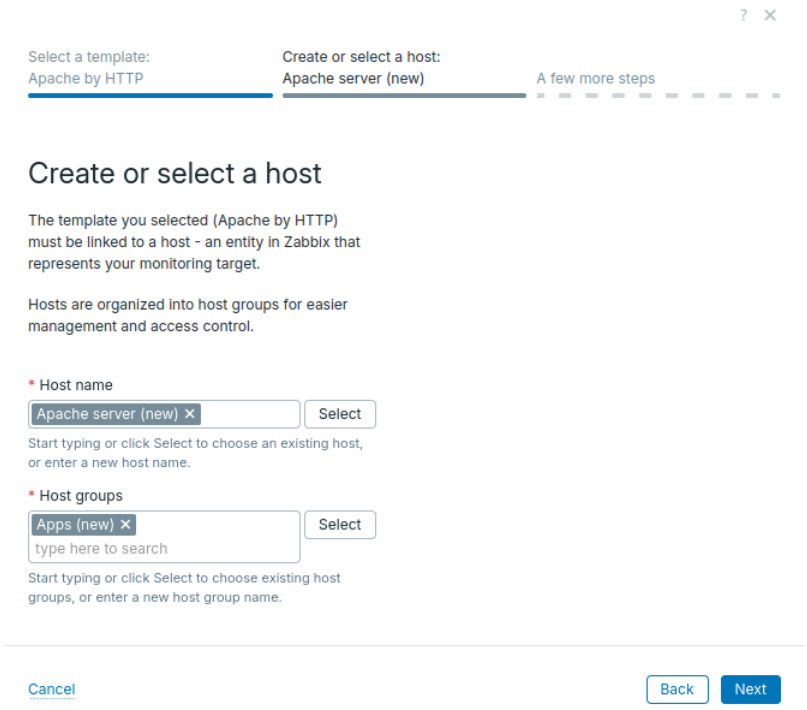
In Zabbix ist diese Vorlage sofort einsatzbereit. Sie enthält vorkonfigurierte Elemente für Datenerfassung, Visualisierung und Analyse.

Um die Vorlage anzuwenden:

1. Starten Sie den Host-Assistenten unter *Datenerfassung* > *Hosts*.
2. Wählen Sie die Vorlage *Apache by HTTP* aus.

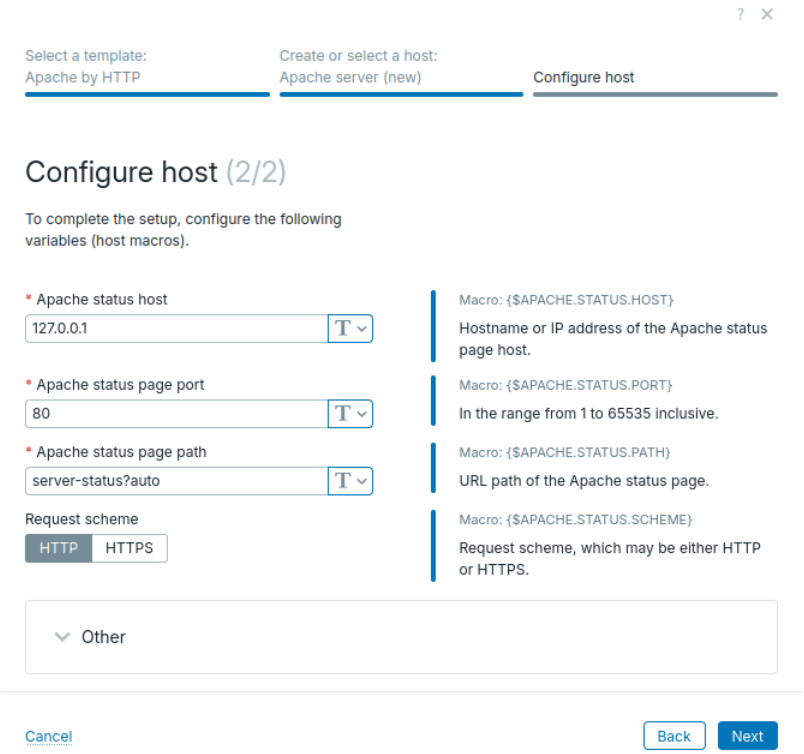


3. Erstellen Sie einen Apache-Host und fügen Sie ihm eine Gruppe hinzu. Dieser virtuelle Host enthält Apache-bezogene Daten.



4. Konfigurieren Sie den Host.
   Wichtig ist, dass Sie die Adresse des Apache-Webservers eingeben (127.0.0.1 auf einem lokalen Rechner, IP-Adresse auf

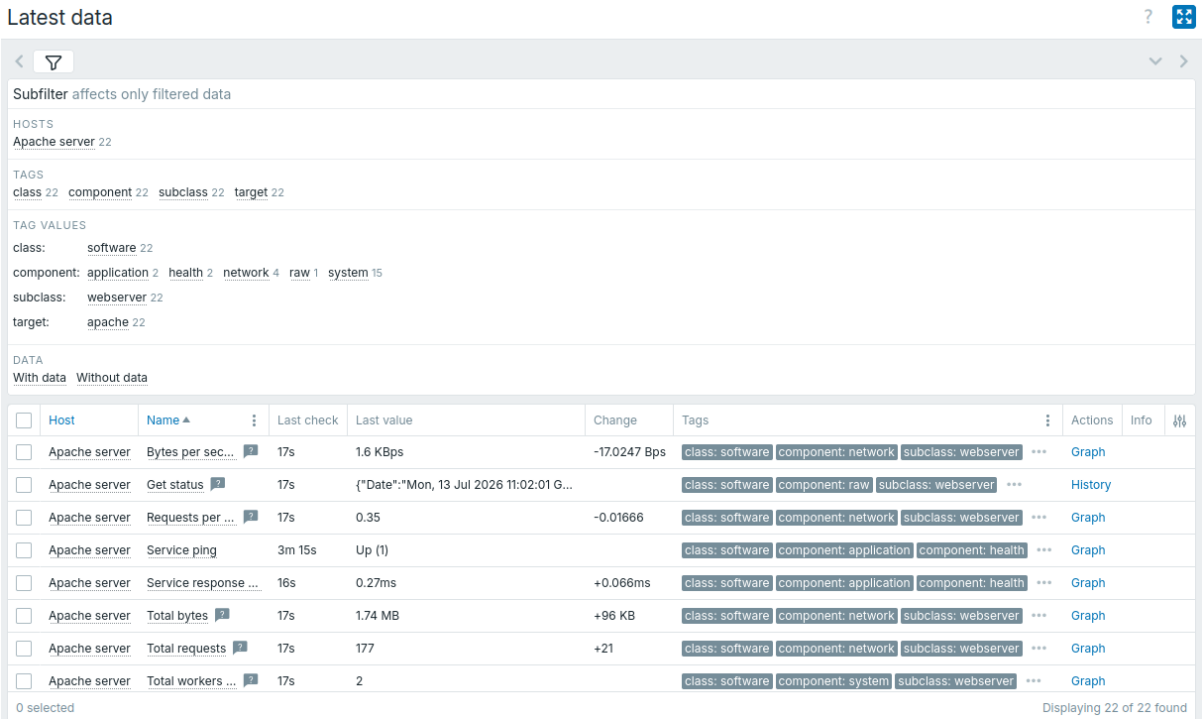
einem entfernten Rechner).



Die übrigen Einstellungen können Sie auf den Standardwerten belassen.

5. Um den Host-Assistenten abzuschließen, klicken Sie auf *Create* und *Finish*.

Blick auf die neuesten Daten



Glückwunsch, das Monitoring des Apache-Webservers wurde gestartet!

Zabbix bietet Visualisierungsoptionen und sendet Benachrichtigungen, wenn Probleme auftreten. Informationen zum Konfigurieren von E-Mail-Benachrichtigungen finden Sie unter Empfangen von Problembenachrichtigungen.

Linux-Server überwachen

Um mit der Überwachung eines Linux-Servers zu beginnen, gehen Sie wie folgt vor:

1. Wenden Sie die Linux-Überwachungsvorlage an.

2. Installieren Sie den Zabbix Agent auf dem Server.
3. Sehen Sie sich die neuesten Daten an.

Um mit der Überwachung eines nicht lokalen Linux-Servers zu beginnen, müssen Sie seine IP-Adresse und die IP-Adresse des Zabbix-Servers kennen.

Auf dem Linux-Server wird ein Zabbix Agent verwendet, um ihn zu überwachen. Anweisungen zur Installation des Agenten werden im Verlauf des Prozesses bereitgestellt.

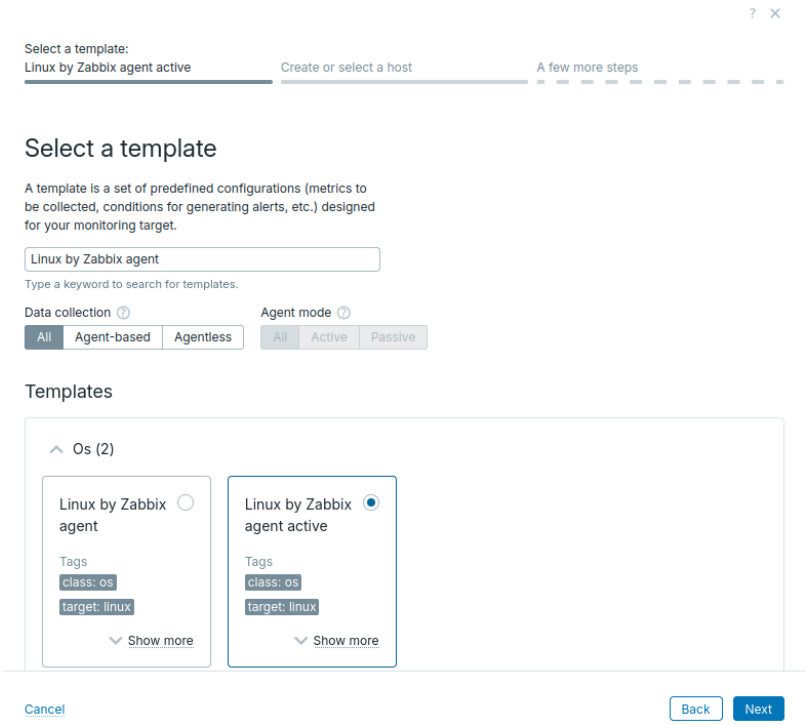
Um diese Anleitung zu verwenden, müssen Sie Zabbix-Server und die Weboberfläche installiert haben. Sie müssen in Zabbix als Standardbenutzer *Admin* angemeldet sein.

Linux-Monitoring-Vorlage anwenden

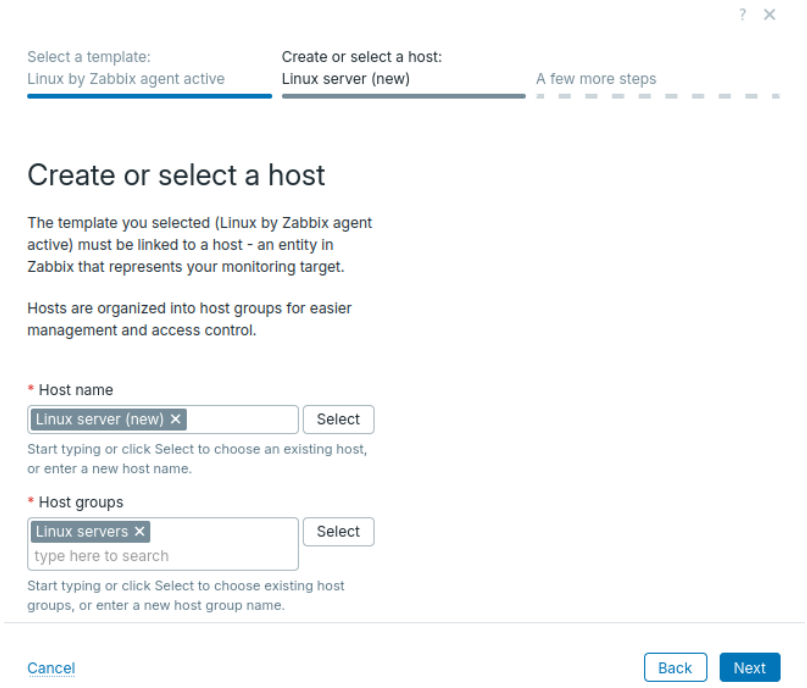
In Zabbix ist diese Vorlage sofort einsatzbereit. Sie enthält vorkonfigurierte Elemente für Datenerfassung, Visualisierung und Analyse.

Um die Vorlage anzuwenden:

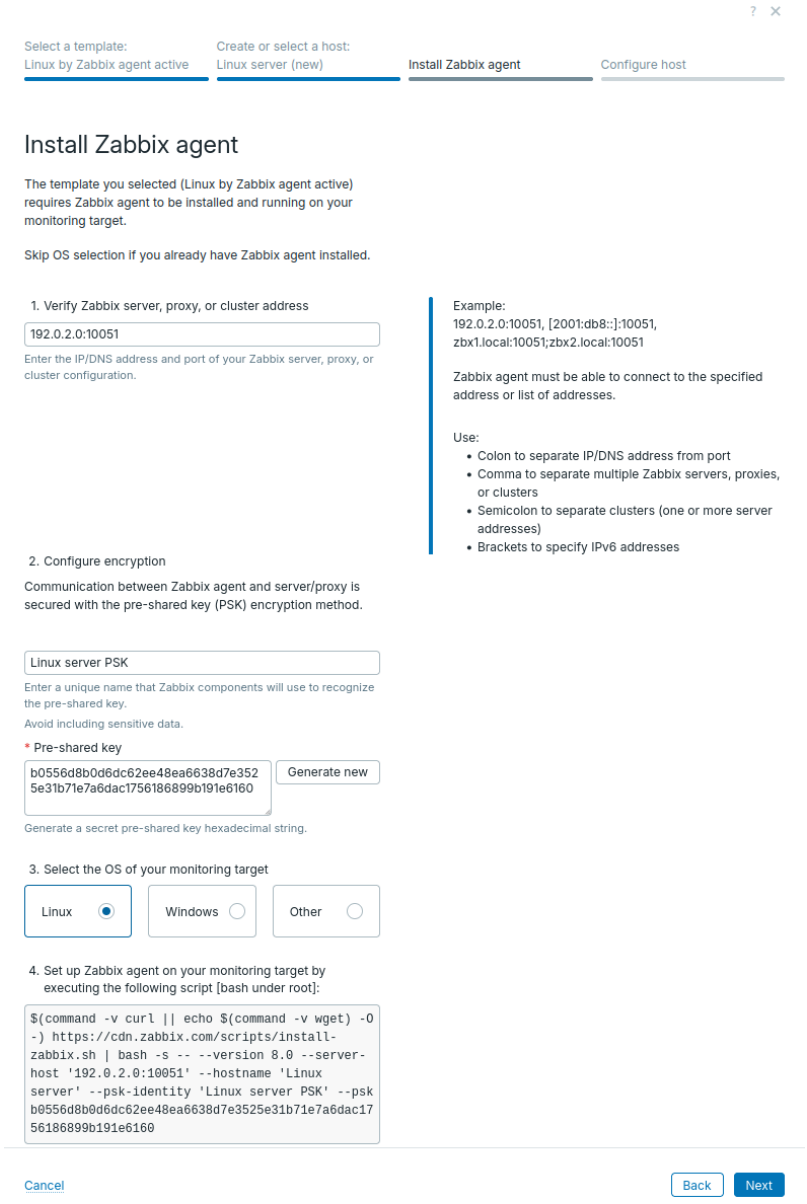
1. Starten Sie den Host-Assistenten unter *Datenerfassung* > *Hosts*.
2. Wählen Sie die Vorlage *Linux by Zabbix agent active* aus.



3. Erstellen Sie einen Host für den Linux-Server und fügen Sie ihm eine Gruppe hinzu. Dieser virtuelle Host enthält alle Daten, die mit der Überwachung des Linux-Servers zusammenhängen.



4. Konfigurieren und installieren Sie den Agent.



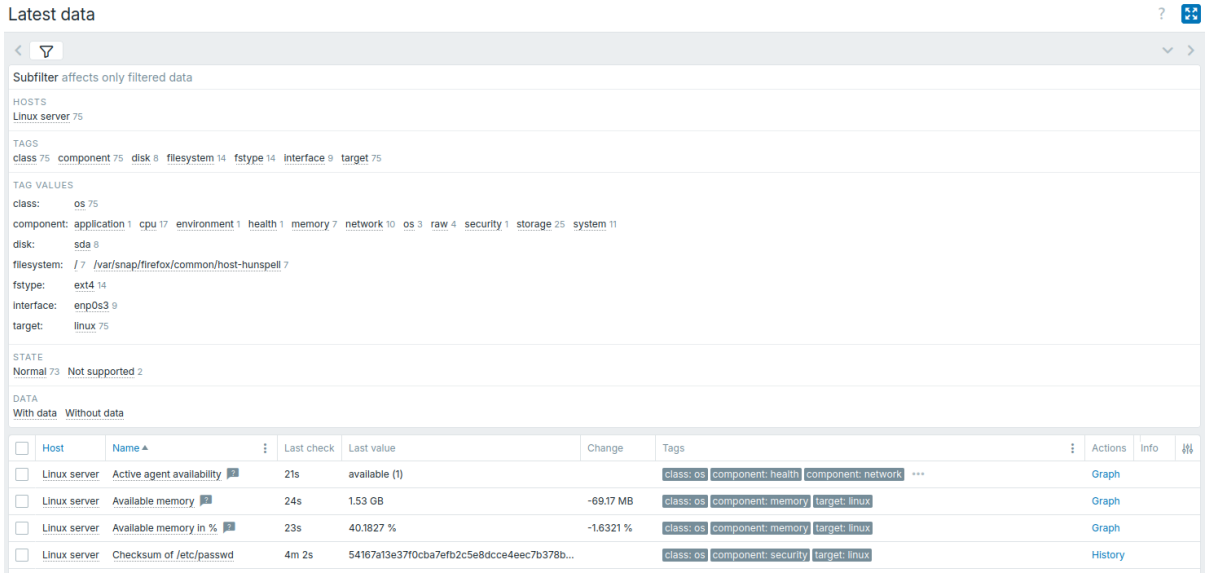
Um den Agent zu installieren, führen Sie den bereitgestellten Befehl auf dem Linux-Server aus. Dadurch wird der Agent installiert

und die Verbindung des Agent zu Zabbix Server mit Verschlüsselung konfiguriert.
 Wenn der Agent bereits installiert ist, können Sie diesen Schritt überspringen (klicken Sie auf *Weiter*). Sie werden jedoch aufgefordert, die Identität des vorab gemeinsam genutzten Schlüssels und den vorab gemeinsam genutzten Schlüssel anzugeben, um eine Verbindung zum Agent herzustellen.

Sie können die übrigen Einstellungen auf den Standardwerten belassen.

5. Schließen Sie die Host-Konfiguration ab: Klicken Sie auf *Erstellen* und *Fertigstellen*.

Sehen Sie sich die neuesten Daten an



Glückwunsch, die Überwachung des Linux-Servers wurde gestartet!

Zabbix bietet Visualisierungsoptionen und sendet Benachrichtigungen, wenn Probleme auftreten. Informationen zum Konfigurieren von E-Mail-Benachrichtigungen finden Sie unter Empfangen von Problembenachrichtigungen.

MySQL-Server überwachen

Um mit der Überwachung eines MySQL-Servers zu beginnen, folgen Sie diesem Ablauf:

1. Erstellen Sie einen eingeschränkten MySQL-Benutzer.
2. Wenden Sie die MySQL-Überwachungsvorlage an.
3. Sehen Sie sich die neuesten Daten an.

Um mit der Überwachung eines MySQL-Servers zu beginnen, müssen Sie dessen DSN und die IP-Adresse des Zabbix-Servers kennen.

Für die Überwachung des MySQL-Servers wird ein Zabbix Agent verwendet. Anweisungen zur Installation des Agent werden im Verlauf des Prozesses bereitgestellt.

Um diese Anleitung zu verwenden, müssen Sie Zabbix-Server und die Weboberfläche installiert haben. Sie müssen in Zabbix als Standardbenutzer *Admin* angemeldet sein.

Eingeschränkten MySQL-Benutzer erstellen

Es wird empfohlen, einen MySQL-Benutzer zu erstellen, dessen Berechtigungen auf die für die Überwachung von MySQL erforderlichen Berechtigungen beschränkt sind.
Ersetzen Sie `<password>` durch ein starkes, eindeutiges Passwort.

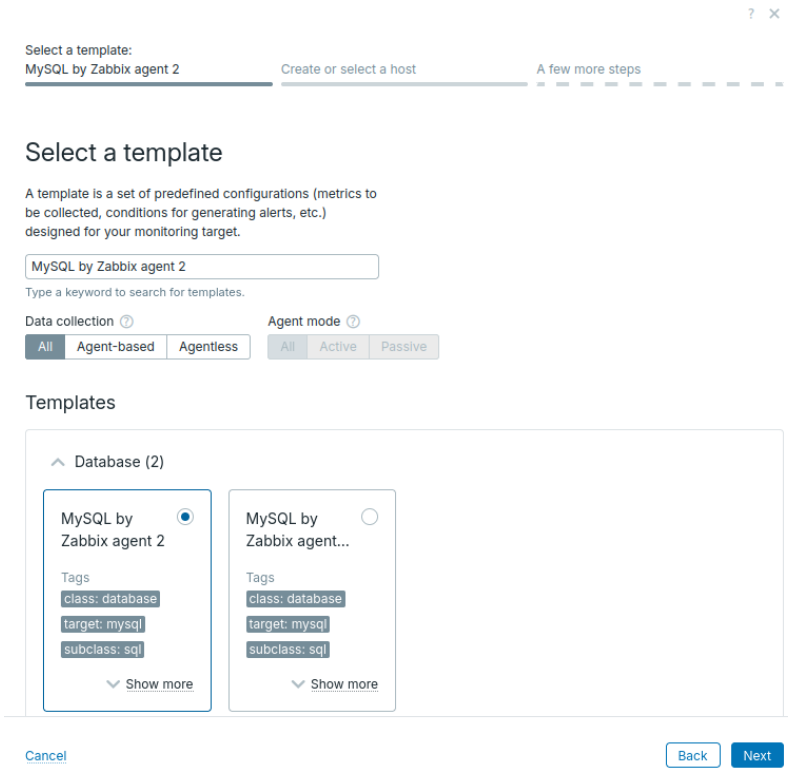
```
mysql> CREATE USER 'zbx_monitor'@'%' IDENTIFIED BY '<password>';
mysql> GRANT REPLICATION CLIENT,PROCESS,SHOW DATABASES,SHOW VIEW ON *.* TO 'zbx_monitor'@'%';
```

MySQL-Monitoring-Vorlage anwenden

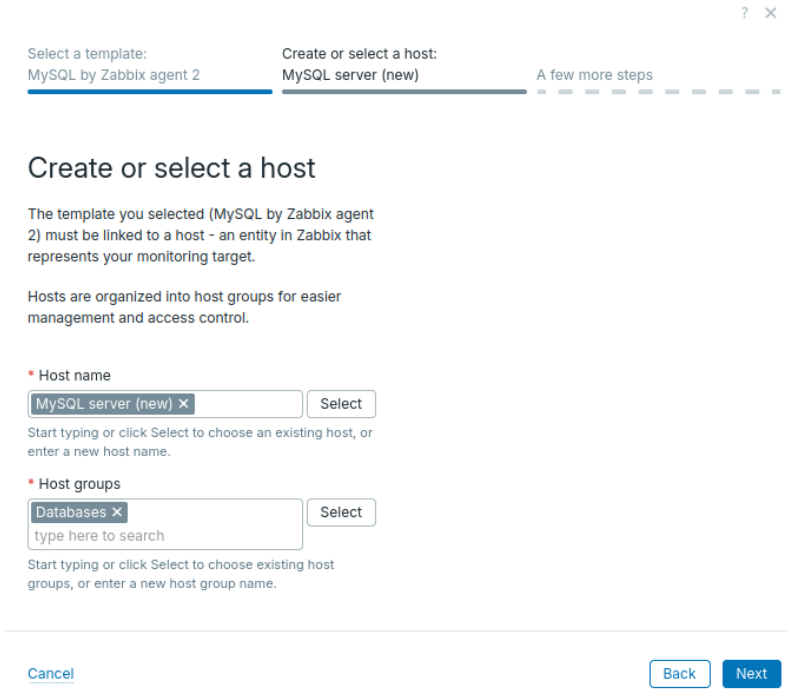
In Zabbix ist diese Vorlage sofort einsatzbereit. Sie enthält vorkonfigurierte Elemente für Datenerfassung, Visualisierung und Analyse.

Um die Vorlage anzuwenden:

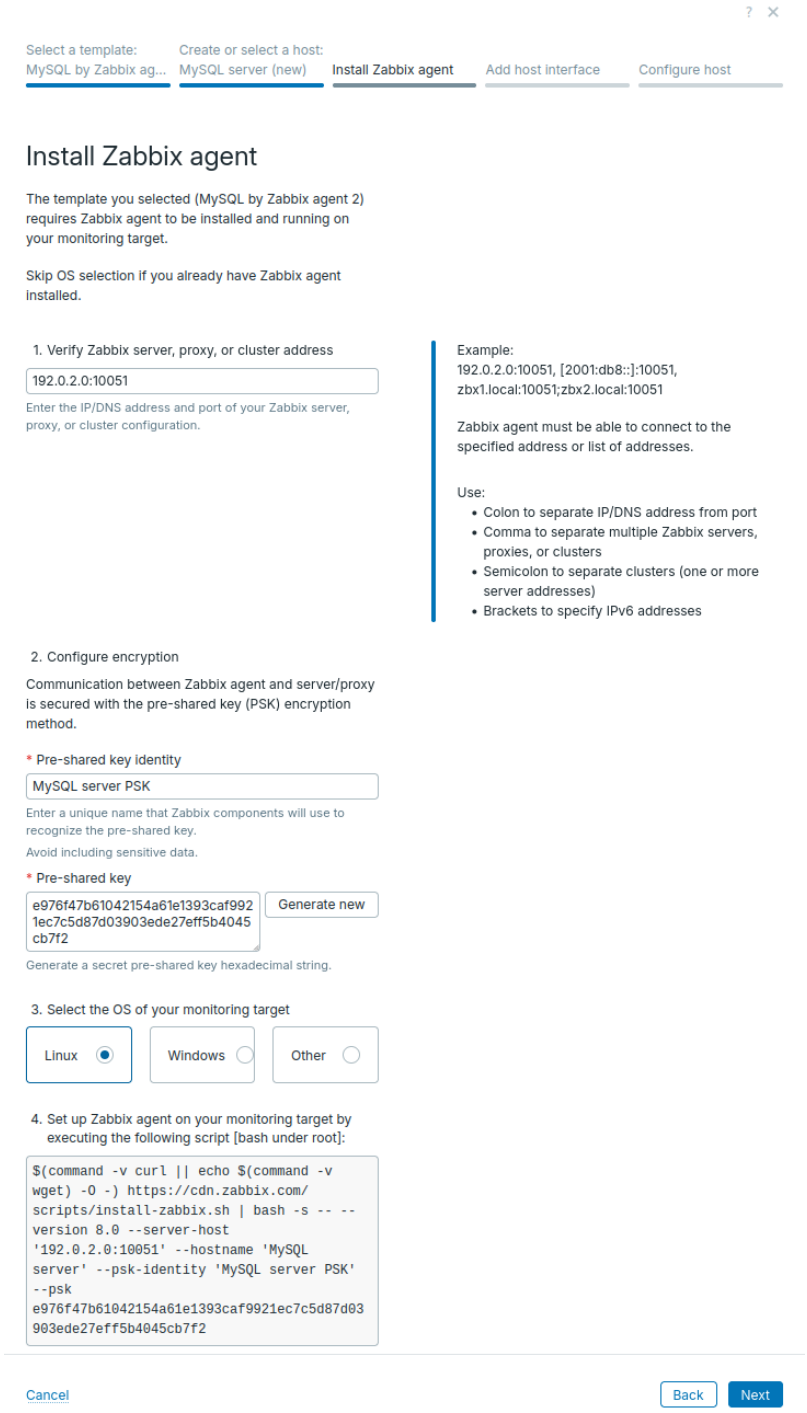
1. Starten Sie den Host-Assistenten unter *Datenerfassung* > *Hosts*.
2. Wählen Sie die Vorlage *MySQL by Zabbix agent 2* aus.



3. Erstellen Sie einen MySQL-Server-Host und fügen Sie eine Gruppe hinzu. Dieser virtuelle Host enthält alle Daten, die mit der Überwachung des MySQL-Servers zusammenhängen.

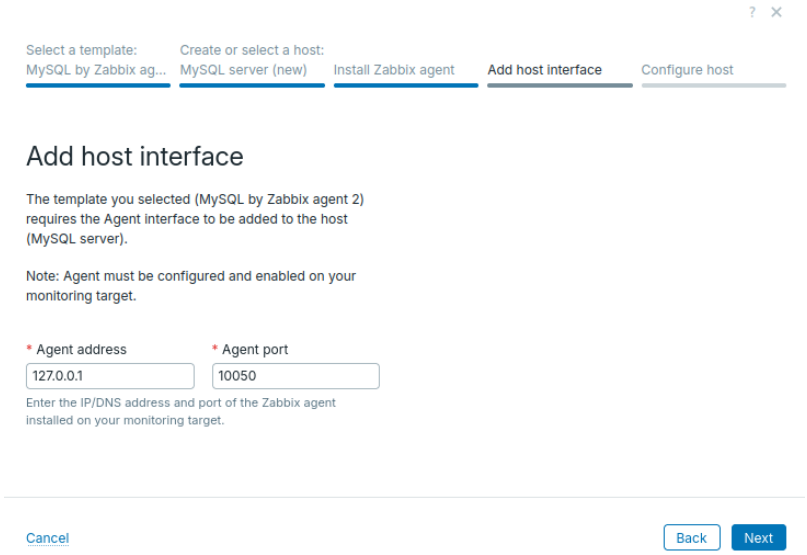


4. Konfigurieren und installieren Sie Agent 2 (Zabbix agent 2 bietet MySQL-Überwachung standardmäßig an).

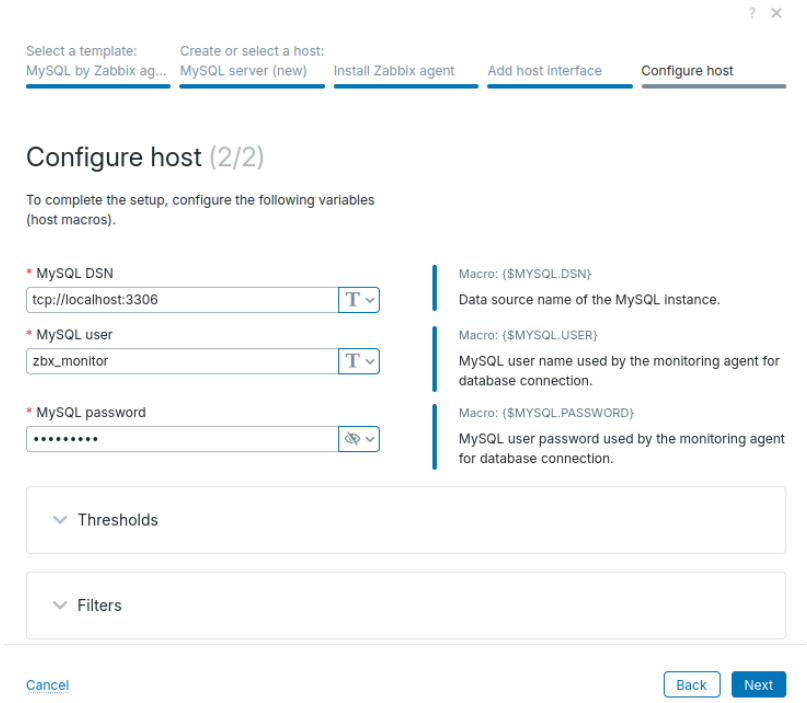


Um den Agent zu installieren, führen Sie den bereitgestellten Befehl auf dem Rechner aus, auf dem der MySQL-Server gehostet wird. Dadurch wird der Agent installiert und die Verbindung des Agenten zum Zabbix Server mit Verschlüsselung konfiguriert.
 Wenn Agent 2 bereits installiert ist, können Sie diesen Schritt überspringen (klicken Sie auf *Weiter*). Sie werden jedoch aufgefordert, die Identität des Pre-Shared Keys und den Pre-Shared Key anzugeben, um eine Verbindung zum Agenten herzustellen.

5. Geben Sie die IP-Adresse und den Port des Agenten (Linux-Server) ein.



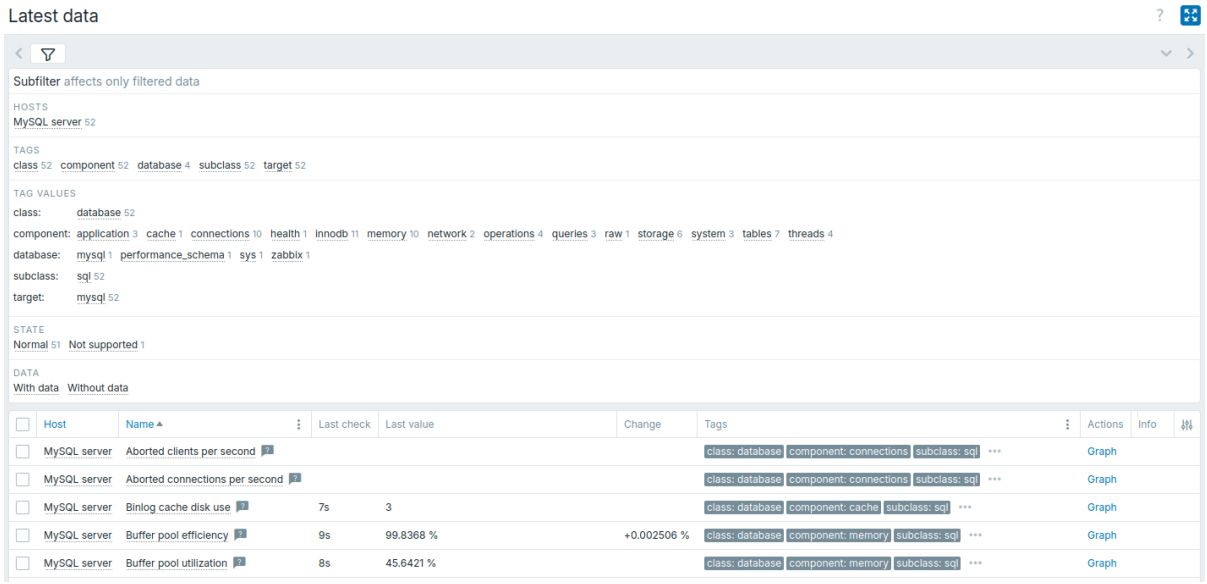
6. Schließen Sie die Konfiguration des Hosts ab. Wichtig ist, dass Sie den DSN des MySQL-Servers sowie den MySQL-Benutzer und das Passwort eingeben müssen. Geben Sie den Benutzer `zbx_monitor` mit dem von Ihnen erstellten Passwort an.



Sie können die übrigen Einstellungen auf den Standardwerten belassen.

6. Um den Host-Assistenten abzuschließen, klicken Sie auf *Erstellen* und *Fertigstellen*.

Neueste Daten anzeigen



Herzlichen Glückwunsch, die Überwachung des MySQL-Servers wurde gestartet!

Zabbix bietet Visualisierungsoptionen und sendet Benachrichtigungen, wenn Probleme auftreten. Informationen zum Konfigurieren von E-Mail-Benachrichtigungen finden Sie unter Empfangen von Problembenachrichtigungen.

Weitere Anleitungen

Dieser Dokumentationsabschnitt enthält kurze Anleitungen zum Einrichten von Zabbix für einige häufig benötigte Überwachungsziele.

Er ist für neue Zabbix-Benutzer gedacht und kann als Navigator durch andere Dokumentationsabschnitte verwendet werden, die die für die Lösung der Aufgabe erforderlichen Informationen enthalten.

Die folgenden Kurzanleitungen sind verfügbar:

- Linux mit Zabbix Agent überwachen
- Windows mit Zabbix Agent überwachen
- Apache über HTTP überwachen
- MySQL mit Zabbix Agent 2 überwachen
- VMware mit Zabbix überwachen
- Netzwerkverkehr mit Zabbix überwachen
- Netzwerkverkehr mit Zabbix mithilfe aktiver Prüfungen überwachen
- Websites mit Browser-Datenpunkten überwachen
- Website-Zertifikate mit Zabbix Agent 2 (passiv) überwachen
- Einen Netzwerkswitch oder Router mit Zabbix überwachen
- Windows-Ereignisprotokoll mithilfe aktiver Prüfungen überwachen

Linux mit Zabbix Agent überwachen

Einführung

Diese Seite führt Sie durch die Schritte, die erforderlich sind, um die grundlegende Überwachung von Linux-Maschinen mit Zabbix zu starten.
Die in diesem Tutorial beschriebenen Schritte können auf jedes Linux-basierte Betriebssystem angewendet werden.

Für wen dieser Leitfaden gedacht ist

Dieser Leitfaden ist für neue Zabbix-Benutzer gedacht und enthält die minimale Anzahl an Schritten, die erforderlich sind, um die grundlegende Überwachung Ihres Linux-Systems zu aktivieren. Wenn Sie nach umfangreichen Anpassungsoptionen suchen oder eine fortgeschrittenere Konfiguration benötigen, lesen Sie den Abschnitt Konfiguration im Zabbix-Handbuch.

Voraussetzungen

Bevor Sie mit dieser Anleitung fortfahren, müssen Sie den Zabbix Server und das Zabbix Frontend gemäß den Anweisungen für Ihr Betriebssystem herunterladen und installieren.

Installieren von Zabbix Agent

Zabbix Agent ist der Prozess, der für das Sammeln von Daten verantwortlich ist.

Prüfen Sie die Version Ihres Zabbix Servers:

```
zabbix_server -V
```

Installieren Sie Zabbix Agent derselben Version (empfohlen) auf dem Linux-Rechner, den Sie überwachen möchten.
Je nach Ihren Überwachungsanforderungen kann dies derselbe Rechner sein, auf dem der Zabbix Server installiert ist, oder ein völlig anderer Rechner.

Wählen Sie die am besten geeignete Installationsmethode:

- Als Docker-Container ausführen - siehe die Liste der verfügbaren Images im Zabbix Docker repository.
- Aus Zabbix Paketen installieren (verfügbar für Alma Linux, CentOS, Debian, Oracle Linux, Raspberry Pi OS, RHEL, Rocky Linux, SUSE Linux Enterprise Server, Ubuntu).
- Aus den Quellen kompilieren.

Zabbix für die Überwachung konfigurieren

Der Zabbix Agent kann Metriken im aktiven oder passiven Modus erfassen (gleichzeitig).

> **Note:**
> Eine passive Prüfung ist eine einfache Datenanforderung. Der Zabbix Server oder Proxy fragt bestimmte Daten ab (zum Beispiel die CPU-Auslastung), und der Zabbix Agent sendet das Ergebnis an den Server zurück. Aktive Prüfungen erfordern eine komplexere Verarbeitung. Der Agent muss zunächst vom Server bzw. von den Servern eine Liste von Datenpunkten zur eigenständigen Verarbeitung abrufen und die Daten dann gesammelt zurücksenden. Weitere Informationen finden Sie unter Passive and active agent checks.

Die von Zabbix bereitgestellten Überwachungsvorlagen bieten in der Regel zwei Alternativen - eine Vorlage für den Zabbix Agent und eine Vorlage für den Zabbix Agent (aktiv). Bei der ersten Option erfasst der Agent Metriken im passiven Modus. Solche Vorlagen liefern identische Überwachungsergebnisse, verwenden jedoch unterschiedliche Kommunikationsprotokolle.

Die weitere Zabbix-Konfiguration hängt davon ab, ob Sie eine Vorlage für aktive oder passive Zabbix Agent-Prüfungen auswählen.

Passive Checks

Zabbix Agent

1. Öffnen Sie die Agent-Konfigurationsdatei auf dem Rechner, auf dem der Agent installiert ist.

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

2. Fügen Sie dem Parameter `Server` die IP-Adresse oder den DNS-Namen (und optional den Port) Ihres Zabbix-Servers hinzu. Beispiel:

```
Server=192.0.2.1:10051
```

Der Zabbix Agent verwendet diese Adresse, um eingehende Verbindungen nur von den angegebenen Zabbix-Servern oder Proxys zu akzeptieren und Daten abzurufen.

3. Starten Sie den Zabbix Agent neu.

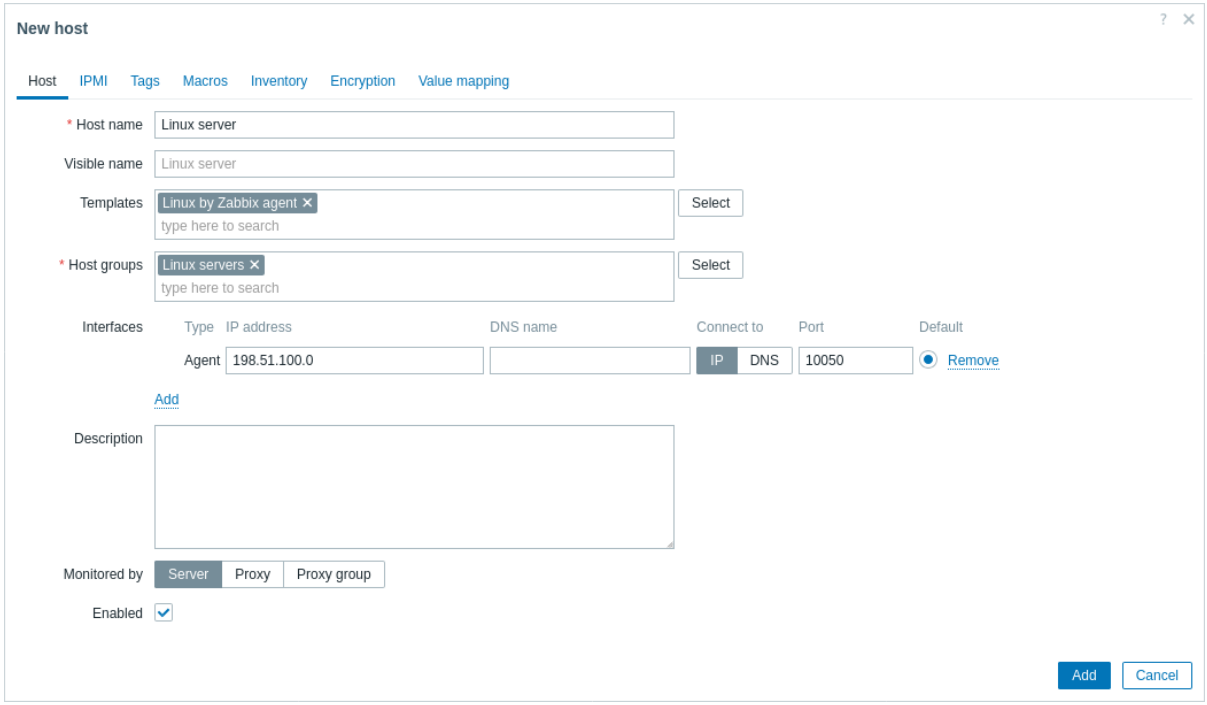
```
systemctl restart zabbix-agent
```

Zabbix Frontend

1. Melden Sie sich im Zabbix Frontend an.

2. Erstellen Sie einen Host in der Zabbix-Weboberfläche.

- Geben Sie im Feld *Host name* einen Hostnamen ein (z. B. ”Linux server”).
- Geben Sie im Feld *Templates* die Vorlage ”Linux by Zabbix agent” ein oder wählen Sie sie aus, die mit dem Host verknüpft wird.
- Geben Sie im Feld *Host groups* eine Hostgruppe ein oder wählen Sie sie aus (z. B. ”Linux servers”).
- Fügen Sie im Parameter *Interfaces* eine *Agent*-Schnittstelle hinzu und geben Sie die IP-Adresse oder den DNS-Namen des Linux-Rechners an, auf dem der Agent installiert ist.



3. Klicken Sie auf *Add*, um den Host hinzuzufügen. Dieser Host repräsentiert den überwachten Linux-Rechner.

Aktive Prüfungen

Zabbix Agent

1. Öffnen Sie die Agent-Konfigurationsdatei auf dem Rechner, auf dem der Agent installiert ist:

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

2. Fügen Sie dem Parameter ServerActive die IP-Adresse oder den DNS-Namen (und optional den Port) Ihres Zabbix Servers hinzu. Beispiel:

```
ServerActive=192.0.2.1:10051
```

Der Zabbix Agent verwendet diese Adresse, um eine Verbindung zum Trapper-Port des Zabbix Servers (Standard: 10051) herzustellen und die Konfigurationsdaten für aktive Prüfungen anzufordern.

3. Definieren Sie den Parameter Hostname. Dieser muss mit dem Hostnamen übereinstimmen, der im Zabbix Frontend definiert wird. In unserem Beispiel lautet er:

```
Hostname=Linux server
```

Der Wert von Hostname muss übereinstimmen, da der Zabbix Agent ihn bei aktiven Prüfungen verwendet, um die korrekte Host-Konfiguration vom Server abzurufen. Der Agent stellt insbesondere eine Verbindung zum Server her und identifiziert sich anhand des Werts von Hostname. Der Server stellt anschließend die Überwachungskonfiguration für diesen Host bereit. Wenn diese Werte voneinander abweichen, erhält der Agent nicht die passende Konfiguration, was zu fehlenden Metriken oder Problemen bei der Überwachung führt.

4. Starten Sie den Zabbix Agent neu.

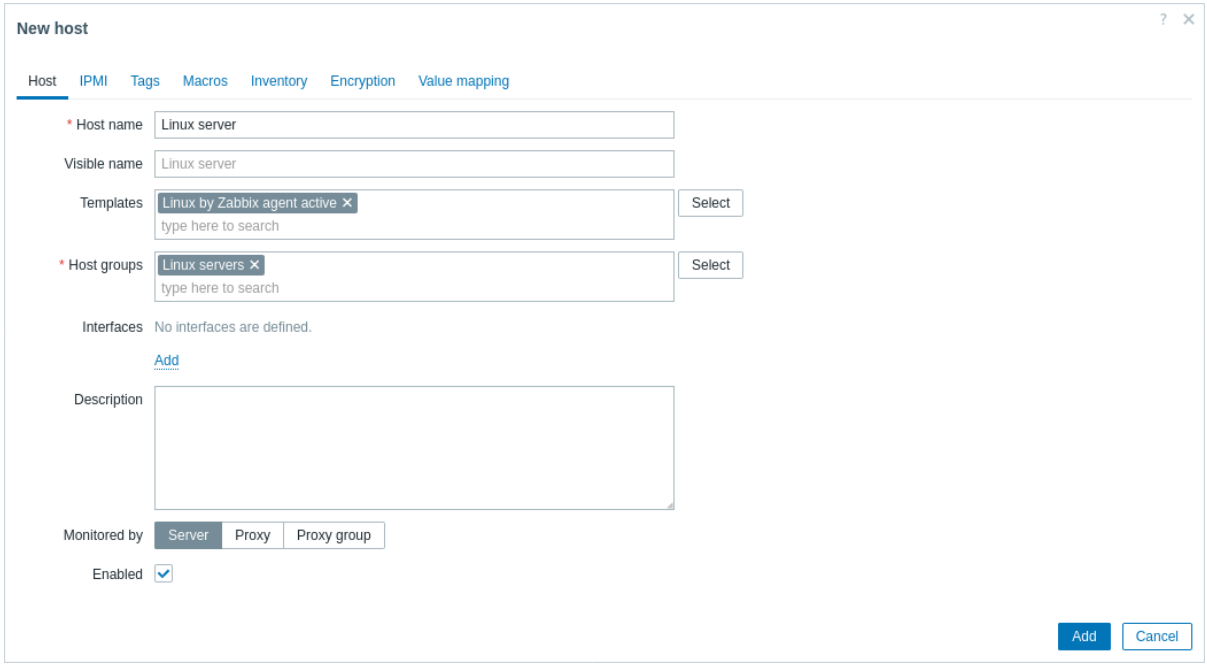
```
systemctl restart zabbix-agent
```

Zabbix Frontend

1. Melden Sie sich im Zabbix Frontend an.

2. Erstellen Sie einen Host in der Zabbix-Weboberfläche.

- Geben Sie im Feld *Host name* einen Hostnamen ein (z. B. ”Linux server”), der mit dem zuvor in der Konfigurationsdatei des Agent definierten Wert des Parameters *Hostname* übereinstimmt.
- Geben Sie im Feld *Templates* die Vorlage ”Linux by Zabbix agent active” ein oder wählen Sie sie aus, die mit dem Host verknüpft wird.
- Geben Sie im Feld *Host groups* eine Hostgruppe ein oder wählen Sie sie aus (z. B. ”Linux servers”).
- Lassen Sie *Interfaces* undefiniert; für aktive Prüfungen ist keine Schnittstelle erforderlich, da der Agent die Verbindung zum Server initiiert, anstatt auf eine Verbindung vom Server zu warten.



3. Klicken Sie auf *Add*, um den Host hinzuzufügen. Dieser Host wird den überwachten Linux-Rechner repräsentieren.

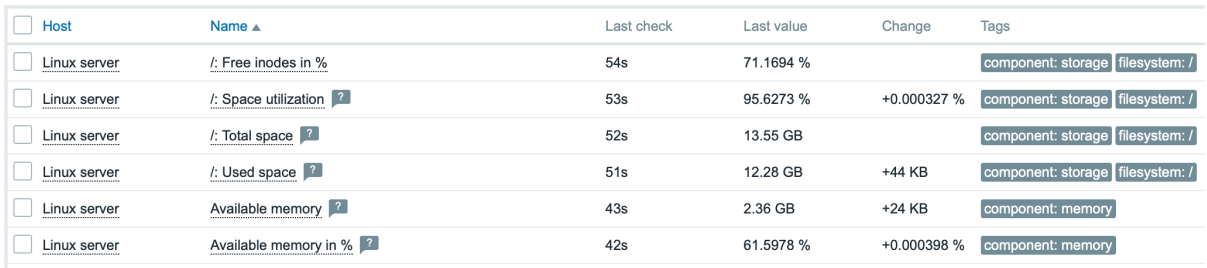
Gesammelte Metriken anzeigen

Glückwunsch! Zu diesem Zeitpunkt überwacht Zabbix Ihren Linux-Rechner bereits.

Um die gesammelten Metriken anzuzeigen, öffnen Sie den Menübereich *Monitoring* > *Hosts* Menüabschnitt und klicken Sie neben dem Host auf *Latest data*.



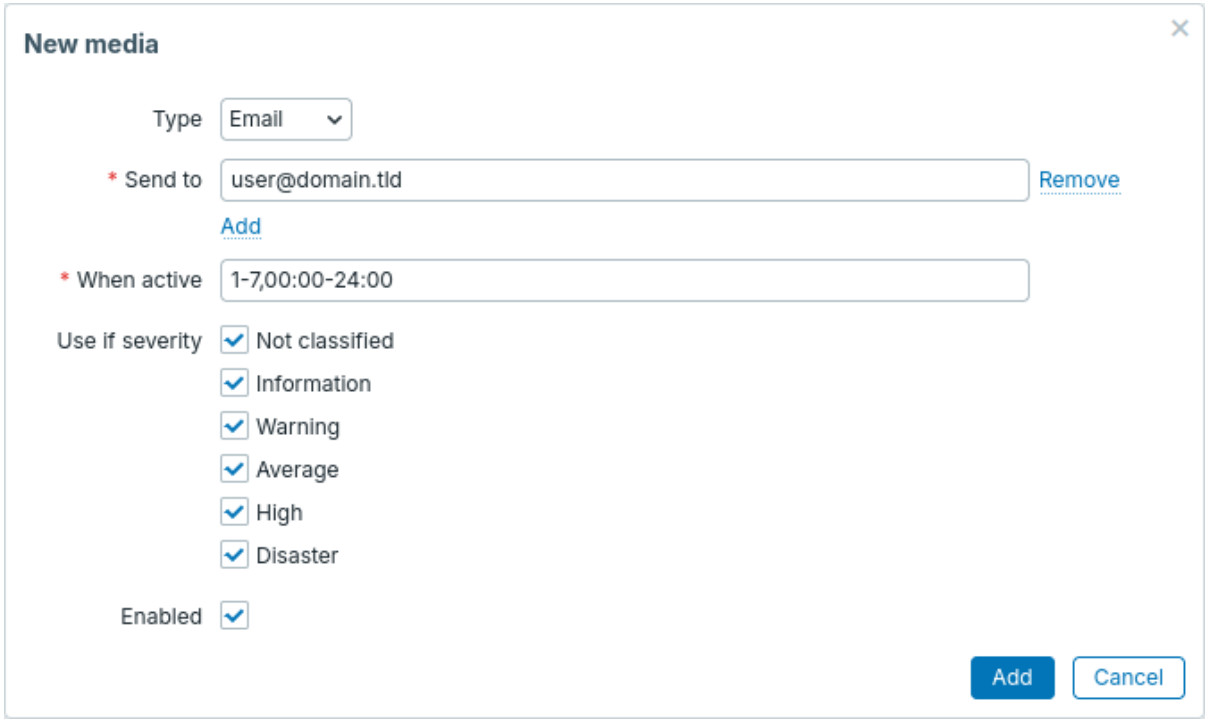
Dadurch wird eine Liste aller neuesten Metriken geöffnet, die vom Linux-Server-Host gesammelt wurden.



Problembenachrichtigungen einrichten

Zabbix kann Sie auf verschiedene Arten über ein Problem mit Ihrer Infrastruktur benachrichtigen.
Diese Anleitung enthält Konfigurationsschritte zum Senden von E-Mail-Benachrichtigungen.

1. Gehen Sie zu *Benutzereinstellungen* > *Profil*, wechseln Sie zur Registerkarte *Medien* und fügen Sie Ihre E-Mail-Adresse hinzu.



2. Folgen Sie der Anleitung zum Empfangen von Problembenachrichtigungen.

Wenn Zabbix das nächste Mal ein Problem erkennt, sollten Sie eine Benachrichtigung per E-Mail erhalten.

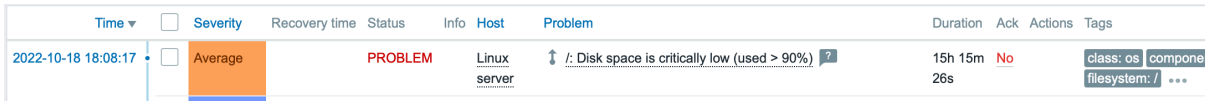
Testen Sie Ihre Konfiguration

Unter Linux können Sie eine hohe CPU-Last simulieren und dadurch eine Problemwarnung erhalten, indem Sie Folgendes ausführen:

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

Möglicherweise müssen Sie mehrere md5sum-Prozesse ausführen, damit die CPU-Last den Schwellenwert überschreitet.

Wenn Zabbix das Problem erkennt, wird es im Abschnitt *Monitoring* > *Problems* angezeigt.



Wenn die Benachrichtigungen konfiguriert sind, erhalten Sie außerdem die Problembenachrichtigung.

Siehe auch:

- Ein Datenpunkt erstellen - wie Sie mit der Überwachung zusätzlicher Metriken beginnen (benutzerdefinierte Überwachung ohne Vorlagen).
- Zabbix-Agent-Datenpunkte, Zabbix-Agent-Datenpunkte für Windows - vollständige Liste der Metriken, die Sie mit dem Zabbix Agent unter Windows überwachen können.
- Problem-Eskalationen - wie Sie mehrstufige Alarmszenarien erstellen (z. B. zuerst eine Nachricht an den Systemadministrator senden und dann, falls ein Problem innerhalb von 45 Minuten nicht behoben wird, eine Nachricht an den Rechenzentrum-sleiter senden).
- Installation aus Paketen - wie Sie Zabbix-Komponenten mit den offiziellen RPM- und DEB-Paketen für verschiedene Linux-Distributionen installieren und so Zugriff auf die neuesten Funktionen und Fehlerbehebungen sicherstellen.

Windows mit Zabbix Agent überwachen

Einführung

Diese Seite führt Sie durch die erforderlichen Schritte, um die grundlegende Überwachung von Windows-Computern mit Zabbix zu starten.

Für wen ist diese Anleitung gedacht?

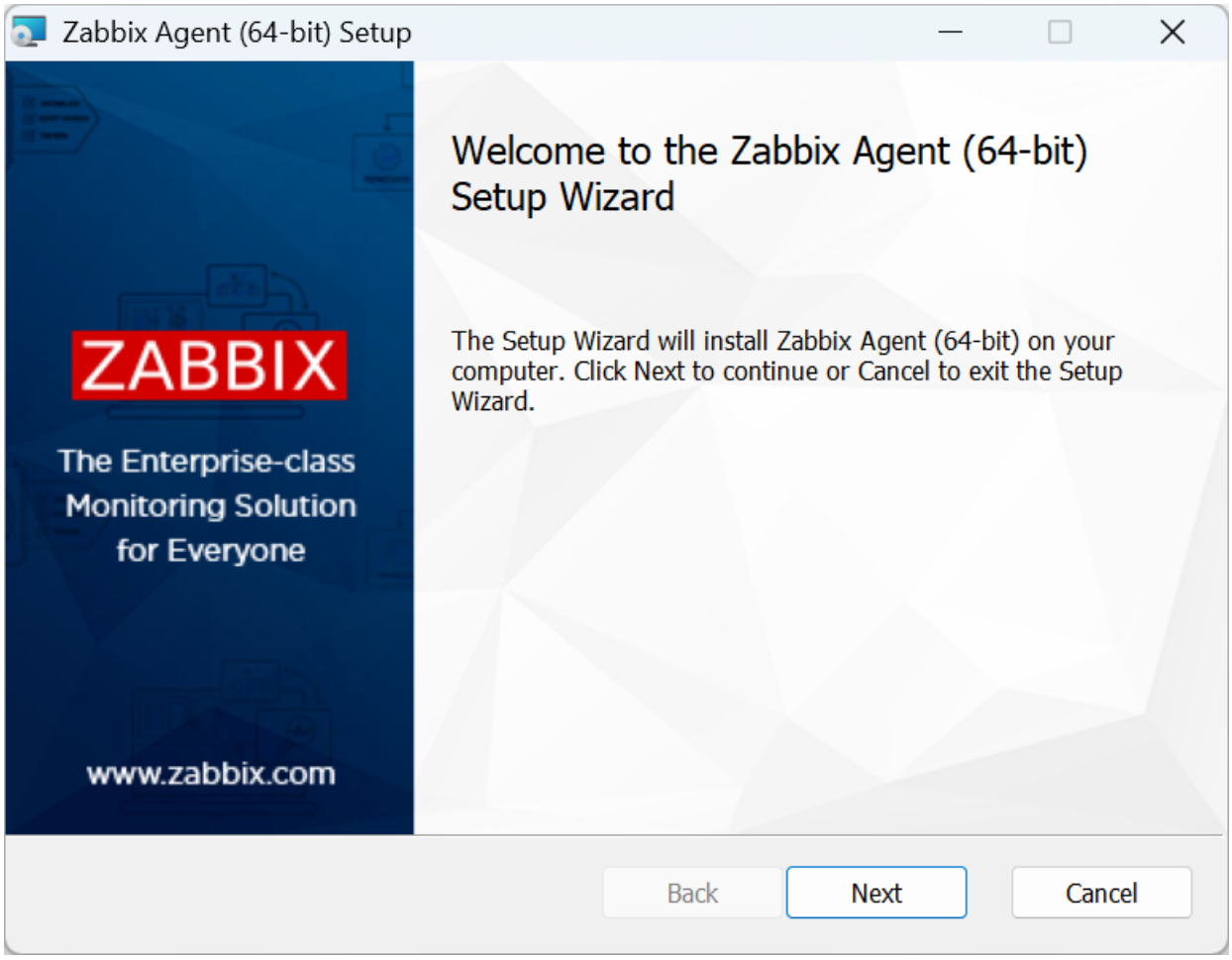
Diese Anleitung richtet sich an neue Zabbix-Benutzer und enthält die Mindestanzahl an Schritten, die erforderlich sind, um die grundlegende Überwachung Ihres Windows-Computers zu aktivieren. Wenn Sie nach umfangreichen Anpassungsmöglichkeiten suchen oder eine fortgeschrittenere Konfiguration benötigen, lesen Sie den Abschnitt Konfiguration im Zabbix-Handbuch.

Voraussetzungen

Bevor Sie mit dieser Installationsanleitung fortfahren, müssen Sie den Zabbix-Server und das Zabbix-Frontend gemäß den Anweisungen für Ihr Betriebssystem herunterladen und installieren.

Zabbix Agent installieren

Der Zabbix Agent ist der Prozess, der für das Sammeln von Daten verantwortlich ist. Sie müssen ihn auf dem Windows-Rechner installieren, den Sie überwachen möchten. Folgen Sie den Installationsanweisungen für den Zabbix Agent unter Windows.



Zabbix für die Überwachung konfigurieren

Der Zabbix Agent kann Metriken im aktiven oder passiven Modus erfassen (gleichzeitig).

Note:
Eine passive Prüfung ist eine einfache Datenanforderung. Der Zabbix Server oder Proxy fragt bestimmte Daten ab (zum Beispiel die CPU-Auslastung), und der Zabbix Agent sendet das Ergebnis an den Server zurück. Aktive Prüfungen erfordern eine komplexere Verarbeitung. Der Agent muss zunächst vom Server bzw. von den Servern eine Liste von Datenpunkten zur unabhängigen Verarbeitung abrufen und dann die Daten gesammelt zurücksenden. Weitere Informationen finden Sie unter Passive and active agent checks.

Die von Zabbix bereitgestellten Überwachungsvorlagen bieten in der Regel zwei Alternativen - eine Vorlage für den Zabbix Agent und eine Vorlage für den Zabbix Agent (aktiv). Bei der ersten Option erfasst der Agent Metriken im passiven Modus. Solche Vorlagen liefern identische Überwachungsergebnisse, verwenden jedoch unterschiedliche Kommunikationsprotokolle.

Die weitere Zabbix-Konfiguration hängt davon ab, ob Sie eine Vorlage für aktive oder passive Zabbix Agent-Prüfungen auswählen.

Passive Checks

Zabbix Frontend

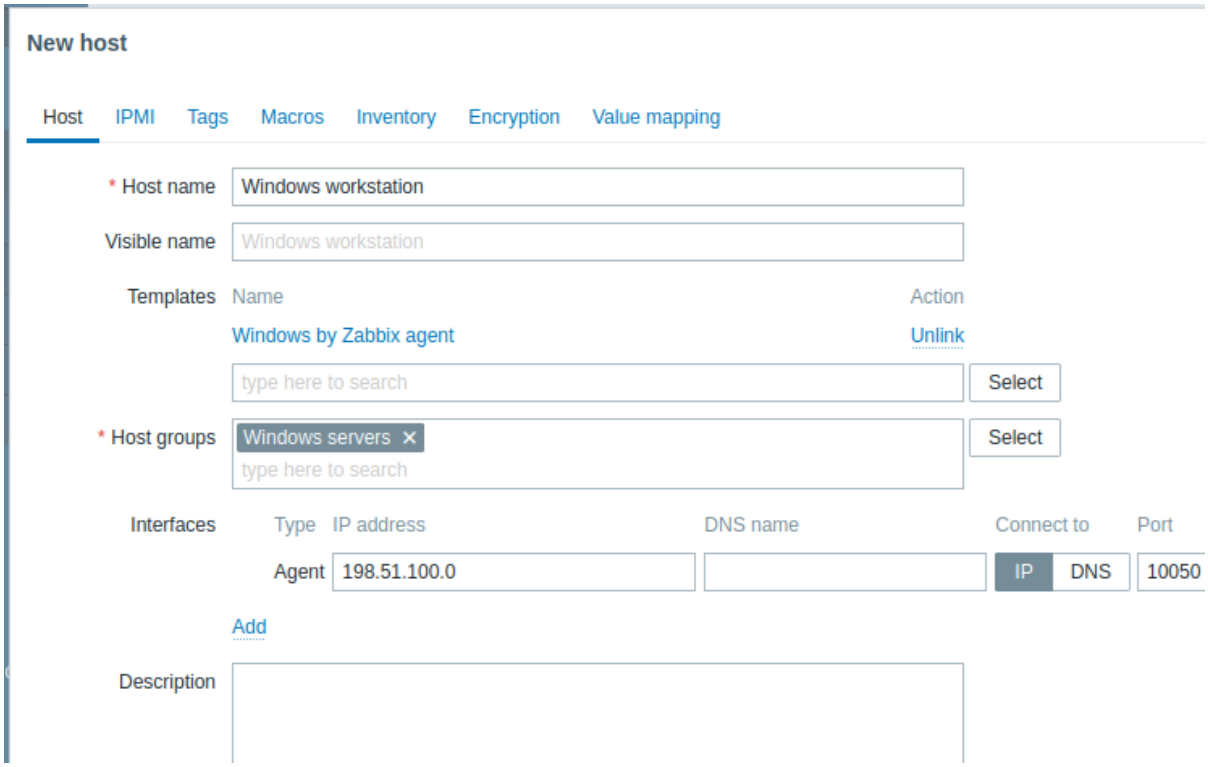
1. Melden Sie sich im Zabbix Frontend an.

2. Erstellen Sie einen Host in der Zabbix-Weboberfläche.

Dieser Host wird Ihren Windows-Computer repräsentieren.

3. Fügen Sie im Parameter *Interfaces* eine *Agent*-Schnittstelle hinzu und geben Sie die IP-Adresse oder den DNS-Namen des Windows-Computers an, auf dem der Agent installiert ist.

4. Geben Sie im Parameter *Templates Windows by Zabbix agent* ein oder wählen Sie es aus.



Zabbix Agent

Für passive Prüfungen muss der Zabbix Agent die IP-Adresse oder den DNS-Namen des Zabbix Servers kennen. Wenn Sie während der Installation des Agents die richtigen Informationen angegeben haben, wurde die Konfigurationsdatei bereits aktualisiert. Andernfalls müssen Sie diese manuell angeben. Öffnen Sie die Konfigurationsdatei des Agents und fügen Sie die IP-Adresse bzw. den DNS-Namen Ihres Zabbix Servers zum Parameter `Server` hinzu.

Beispiel:

```
Server=192.0.2.22
```

Aktive Prüfungen

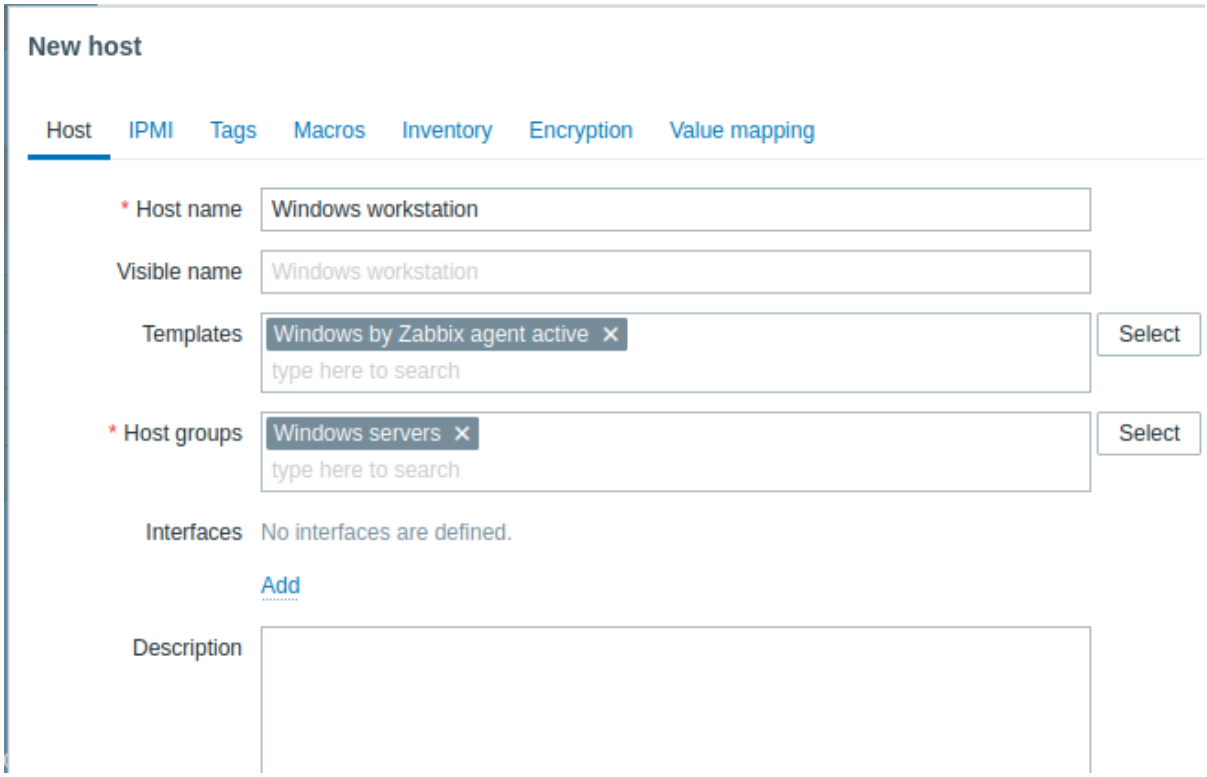
Zabbix Frontend

1. Melden Sie sich im Zabbix Frontend an.

2. Erstellen Sie einen Host in der Zabbix-Weboberfläche.

Dieser Host wird Ihren Windows-Computer repräsentieren.

3. Geben Sie im Parameter *Templates Windows by Zabbix agent active* ein oder wählen Sie ihn aus.



Zabbix Agent

Öffnen Sie im Ordner `C:\Program files\Zabbix Agent` die Datei `zabbix_agentd.conf` und fügen Sie Folgendes hinzu:

- Den Namen des Hosts, den Sie im Zabbix-Web-Frontend erstellt haben, zum Parameter *Hostname*.
- Die IP-Adresse oder den DNS-Namen Ihres Zabbix-Server-Systems zum Parameter *ServerActive* (dies ist möglicherweise bereits vorausgefüllt, wenn Sie ihn während der Zabbix-Agent-Installation angegeben haben).

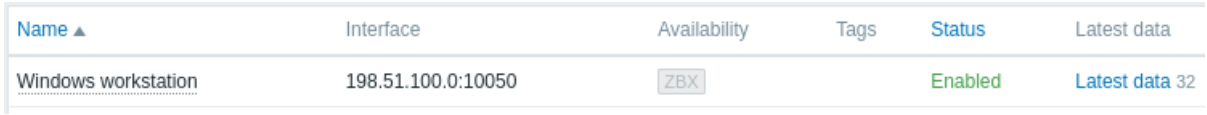
Beispiel:

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

Gesammelte Metriken anzeigen

Glückwunsch! Zu diesem Zeitpunkt überwacht Zabbix Ihren Windows-Rechner bereits.

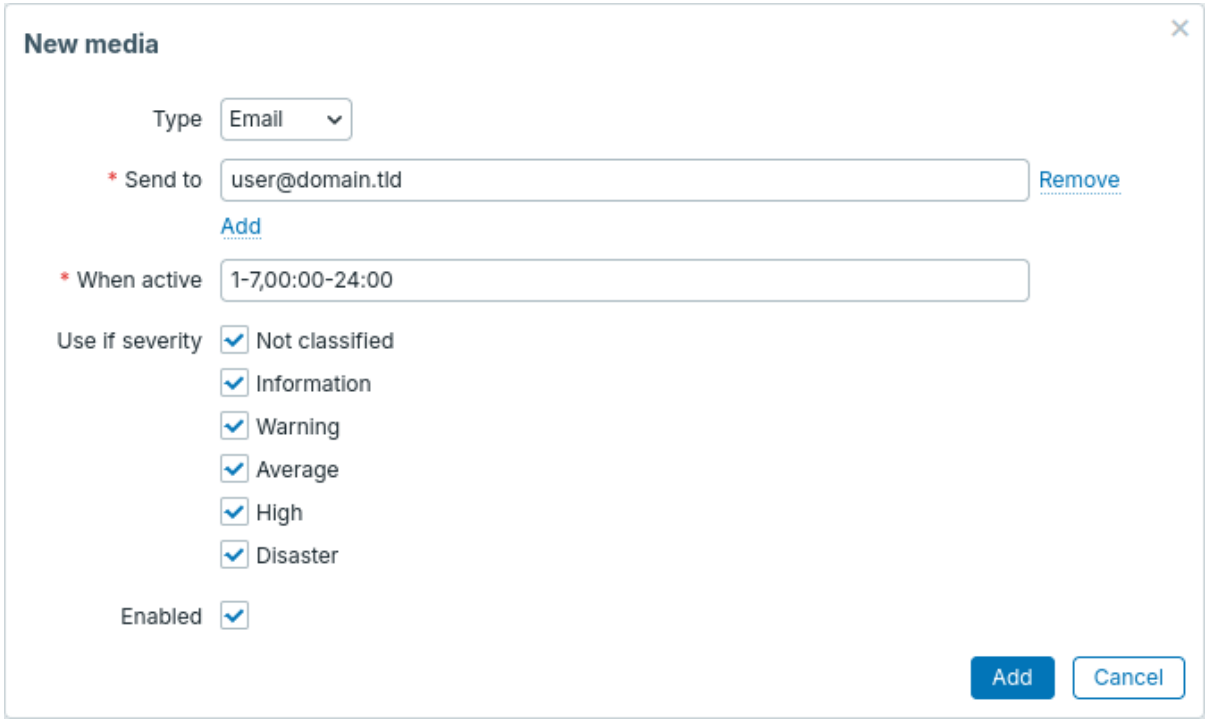
Um die gesammelten Metriken anzuzeigen, öffnen Sie den Menübereich *Monitoring* > *Hosts* Menüabschnitt und klicken Sie auf *Latest data* neben dem Host.



Problemalarme einrichten

Zabbix kann Sie mit verschiedenen Methoden über ein Problem in Ihrer Infrastruktur benachrichtigen. Diese Anleitung enthält die Konfigurationsschritte zum Senden von E-Mail-Benachrichtigungen.

1. Gehen Sie zu *Benutzereinstellungen* > *Profil*, wechseln Sie zur Registerkarte *Medien* und fügen Sie Ihre E-Mail-Adresse hinzu.



2. Folgen Sie der Anleitung zum Empfangen von Problembenachrichtigungen.

Beim nächsten Mal, wenn Zabbix ein Problem erkennt, sollten Sie eine Benachrichtigung per E-Mail erhalten.

> **Note:**
> Unter Windows können Sie das Dienstprogramm CpuStres verwenden, um eine hohe CPU-Last zu simulieren und dadurch eine Problembenachrichtigung zu erhalten.

Siehe auch:

- Einen Datenpunkt erstellen - wie Sie mit der Überwachung zusätzlicher Metriken beginnen (benutzerdefinierte Überwachung ohne Vorlagen).
- Zabbix agent items, Zabbix agent items for Windows - vollständige Liste der Metriken, die Sie mit dem Zabbix Agent unter Windows überwachen können.
- Problem-Eskalationen - wie Sie mehrstufige Alarmszenarien erstellen (z. B. zuerst eine Nachricht an den Systemadministrator senden und dann, falls ein Problem nach 45 Minuten nicht behoben ist, eine Nachricht an den Rechenzentrumsleiter senden).

Apache über HTTP überwachen

Einführung

Diese Seite zeigt eine schnelle und einfache Möglichkeit, mit der Überwachung eines Apache-Webservers zu beginnen, ohne zusätzliche Software zu installieren.

Für wen dieser Leitfaden gedacht ist

Dieser Leitfaden richtet sich an neue Zabbix-Benutzer und enthält die minimale Anzahl an Schritten, die erforderlich sind, um die grundlegende Überwachung Ihrer Apache-Installation zu aktivieren. Wenn Sie nach umfangreichen Anpassungsoptionen suchen oder eine fortgeschrittenere Konfiguration benötigen, lesen Sie den Abschnitt Konfiguration im Zabbix-Handbuch.

Voraussetzungen

Bevor Sie mit dieser Installationsanleitung fortfahren, müssen Sie den Zabbix Server und das Zabbix Frontend gemäß den Anweisungen für Ihr Betriebssystem herunterladen und installieren.

Apache vorbereiten

1. Prüfen Sie, welche Apache-Version Sie verwenden:

Auf einem RHEL-basierten System führen Sie aus:

```
httpd -v
```

Auf Debian/Ubuntu führen Sie aus:

```
apache2 -v
```

2. Stellen Sie sicher, dass das Statusmodul in Ihrer Apache-Instanz aktiviert ist.

Auf einem RHEL-basierten System führen Sie aus:

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

Auf Debian/Ubuntu führen Sie aus:

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

Wenn Sie status_module nicht in der Liste sehen, aktivieren Sie das Modul, indem Sie Folgendes ausführen:

Auf einem RHEL-basierten System führen Sie aus:

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

Auf Debian/Ubuntu führen Sie aus:

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

3. Bearbeiten Sie die Apache-Konfigurationsdatei, um den Zugriff auf Statusberichte von der IP-Adresse des Zabbix Servers zu erlauben.

Auf einem RHEL-basierten System: `/etc/httpd/conf.modules.d/status.conf`:

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

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

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

Fügen Sie der Datei die folgenden Zeilen hinzu (**ersetzen Sie 198.51.100.255** durch die IP-Adresse Ihres Zabbix Servers):

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

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

4. Starten Sie Apache neu

Auf einem RHEL-basierten System führen Sie aus:

```
sudo systemctl restart httpd
```

Auf Debian/Ubuntu führen Sie aus:

```
sudo systemctl restart apache2
```

5. Um zu prüfen, ob alles korrekt konfiguriert ist, führen Sie Folgendes aus (**ersetzen Sie 198.51.100.255** durch die IP-Adresse Ihres Zabbix Servers):

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

Die Antwort sollte Apache-Webserver-Statistiken enthalten.

Zabbix für die Überwachung konfigurieren

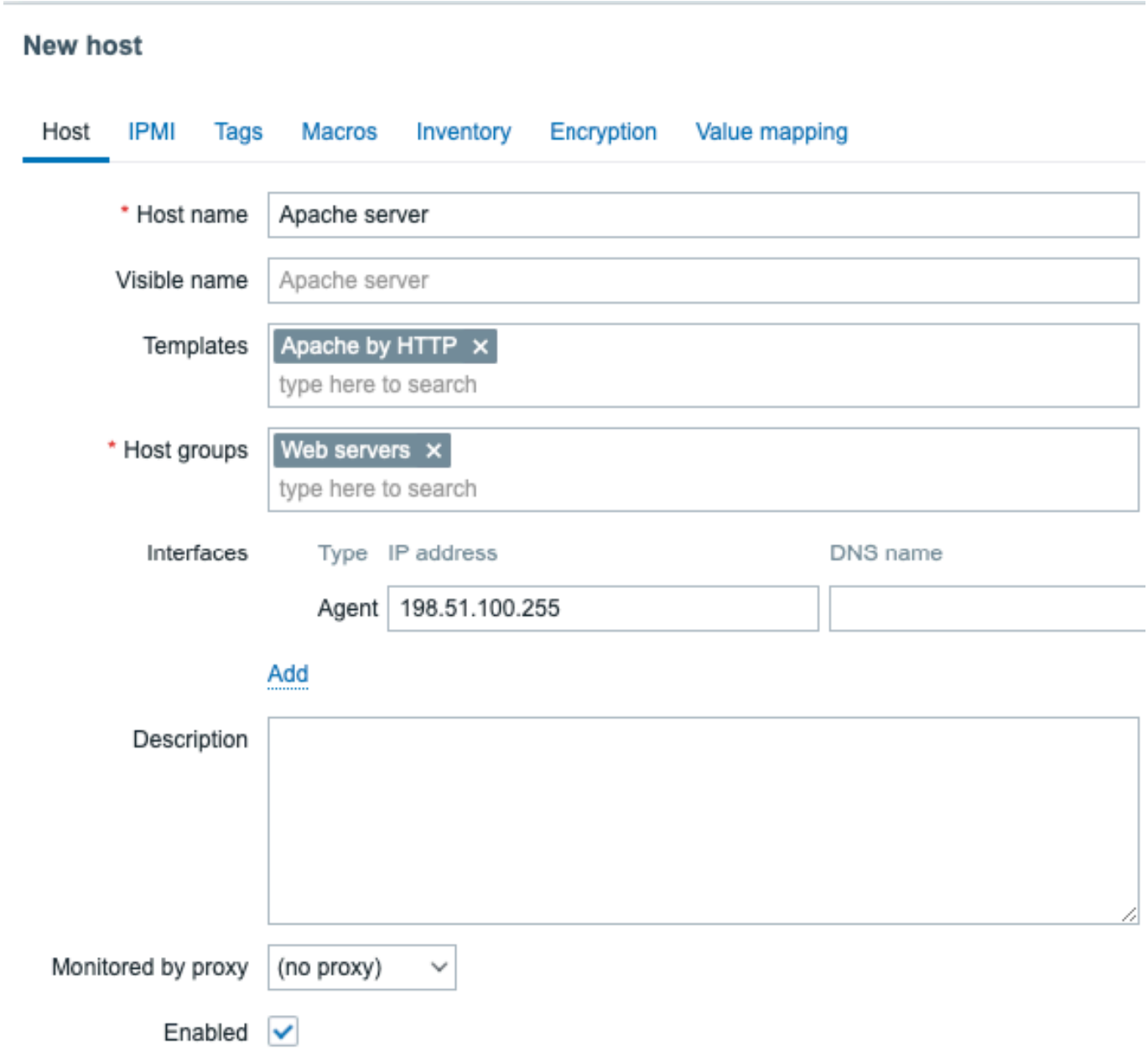
1. Melden Sie sich im Zabbix-Frontend an.

2. Erstellen Sie einen Host in der Zabbix-Weboberfläche.

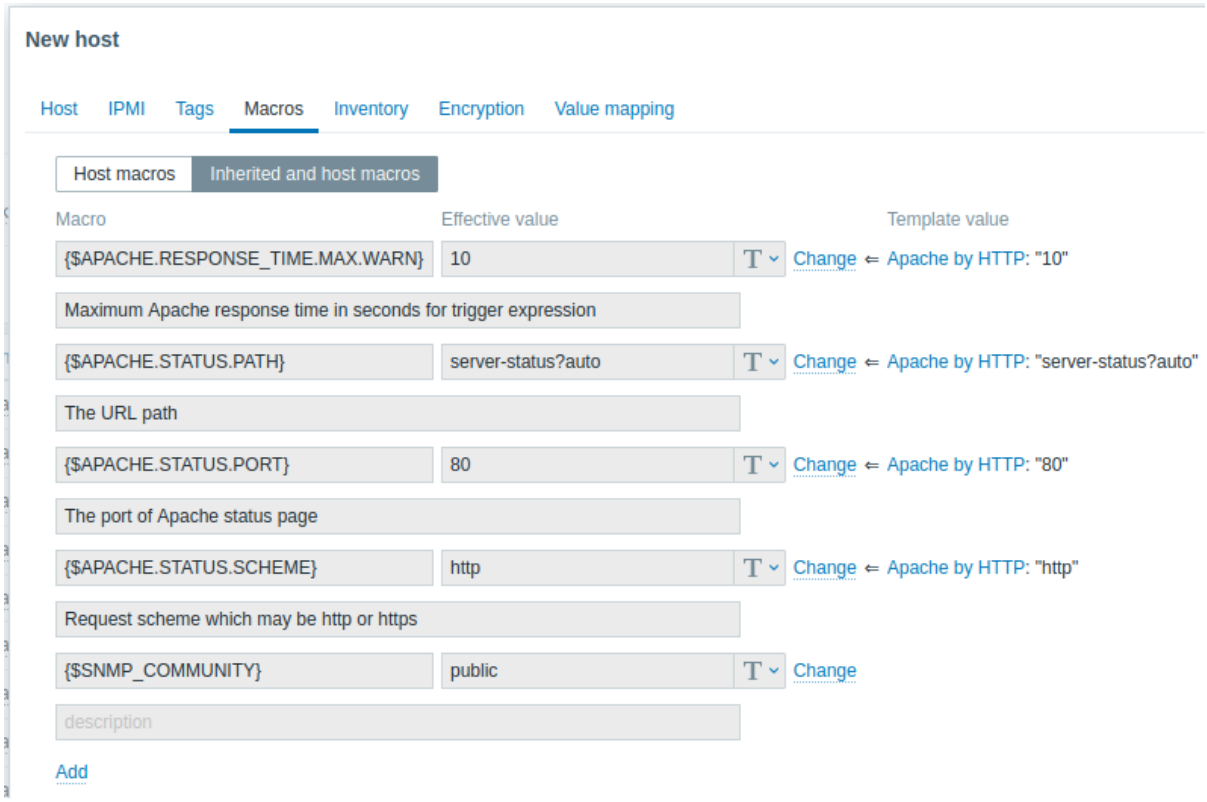
Dieser Host repräsentiert Ihren Apache-Server.

3. Fügen Sie im Parameter *Interfaces* eine *Agent*-Schnittstelle hinzu und geben Sie die IP-Adresse Ihrer Apache-Instanz an. **Sie müssen auf dem Rechner keinen Zabbix-Agent installieren**, die Schnittstelle wird nur zur Auflösung des Makros {HOST.CONN} verwendet. Dieses Makro wird in Vorlagen-Datenpunkten verwendet, um die Apache-Instanz zu lokalisieren.

4. Geben Sie im Parameter *Templates Apache by HTTP* ein oder wählen Sie es aus.



5. Wechseln Sie zur Registerkarte **Macros** und wählen Sie den Modus *Inherited and host macros* aus. Prüfen Sie, ob die Werte der Makros {$APACHE.STATUS.PORT} und {$APACHE.STATUS.SCHEME} zu Ihren Installationseinstellungen passen. Standardmäßig ist der Port 80 und das Schema http. Ändern Sie die Makrowerte, wenn Sie einen anderen Port und/oder ein anderes Schema verwenden.



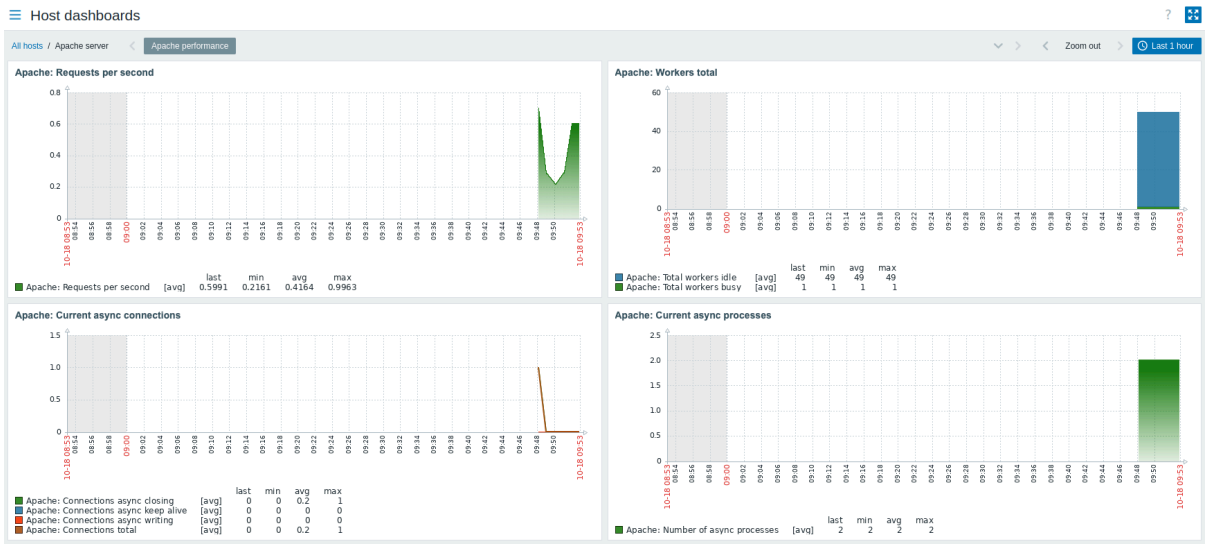
Gesammelte Metriken anzeigen

Glückwunsch! Zu diesem Zeitpunkt überwacht Zabbix bereits Ihren Apache-Webserver.

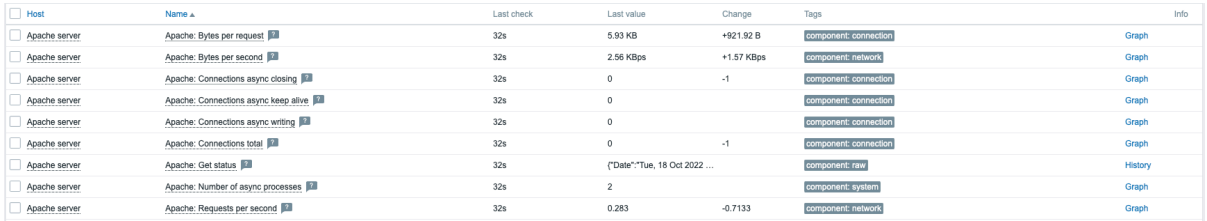
Um die gesammelten Metriken anzuzeigen, öffnen Sie den Menüabschnitt *Monitoring* > *Hosts* Menüabschnitt und klicken Sie auf *Dashboards* neben dem Host.



Diese Aktion führt Sie zum Host-Dashboard mit den wichtigsten Metriken, die von der Apache-Seite /server-status erfasst wurden.



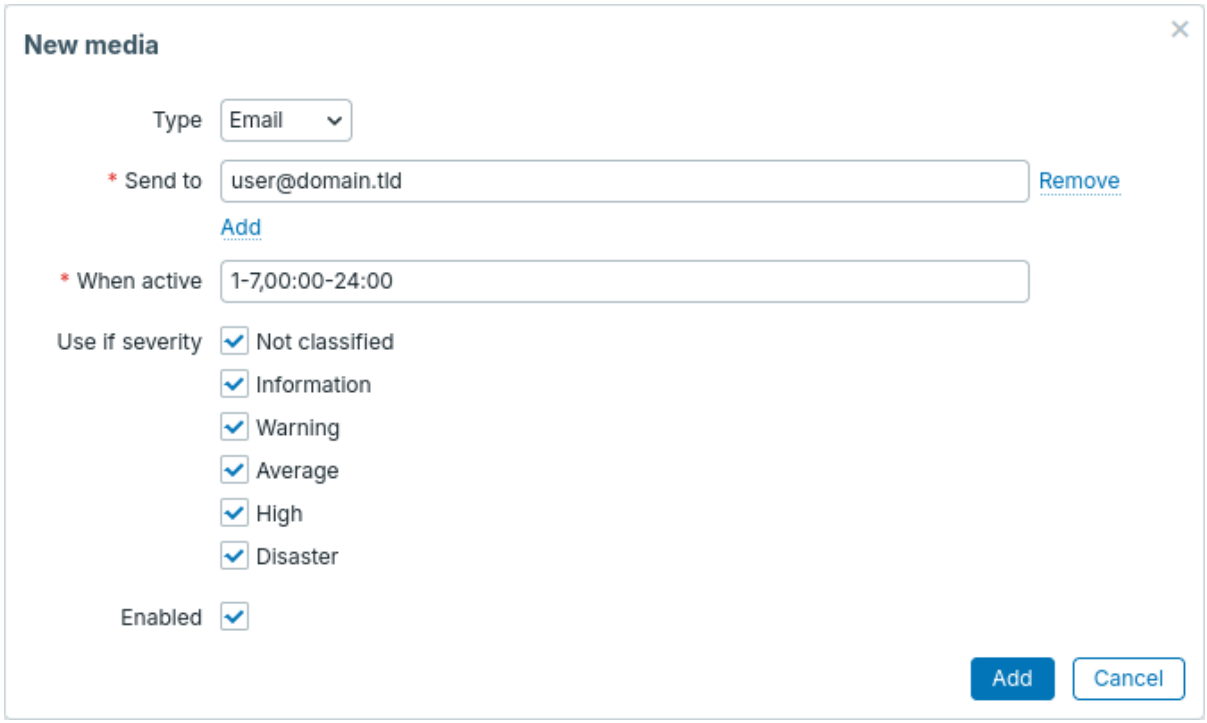
Alternativ können Sie unter *Monitoring* > *Hosts* auf *Latest data* klicken, um alle zuletzt erfassten Metriken in einer Liste anzuzeigen.



Problembenachrichtigungen einrichten

Zabbix kann Sie mit verschiedenen Methoden über ein Problem in Ihrer Infrastruktur benachrichtigen.
Diese Anleitung enthält Konfigurationsschritte zum Senden von E-Mail-Benachrichtigungen.

1. Gehen Sie zu *Benutzereinstellungen* > *Profil*, wechseln Sie zur Registerkarte *Medien* und fügen Sie Ihre E-Mail-Adresse hinzu.



2. Folgen Sie der Anleitung zum Empfangen von Problembenachrichtigungen.

Wenn Zabbix das nächste Mal ein Problem erkennt, sollten Sie eine Benachrichtigung per E-Mail erhalten.

Testen Sie Ihre Konfiguration

Um ein echtes Problem zu simulieren und eine Test-Problembenachrichtigung zu erhalten:

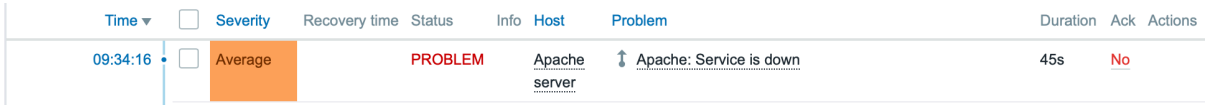
1. Öffnen Sie die Host-Konfiguration *Apache server* in Zabbix.

2. Wechseln Sie zur Registerkarte *Macros* und wählen Sie *Inherited and host macros* aus.

3. Klicken Sie neben dem Makro *{$APACHE.STATUS.PORT}* auf *Change* und legen Sie einen anderen Port fest.

4. Klicken Sie auf *Update*, um die Host-Konfiguration zu speichern.

5. In wenigen Minuten erkennt Zabbix das Problem *Apache service is down*, da keine Verbindung mehr zur Instanz hergestellt werden kann. Es wird im Abschnitt *Monitoring* > *Problems* angezeigt.



Wenn die Benachrichtigungen konfiguriert sind, erhalten Sie außerdem die Problembenachrichtigung.

6. Ändern Sie den Makrowert wieder zurück, um das Problem zu beheben und die Überwachung von Apache fortzusetzen.

Siehe auch:

- Absicherung des Webservers - empfohlene Einstellungen für eine höhere Sicherheit des Webservers.
- Erstellen eines Datenpunkts - wie Sie mit der Überwachung zusätzlicher Metriken beginnen.
- HTTP-Datenpunkte - wie Sie benutzerdefinierte Metriken mit dem HTTP-Agent überwachen.
- Problem-Eskalationen - wie Sie mehrstufige Alarmszenarien erstellen (z. B. zuerst eine Nachricht an den Systemadministrator senden und dann, wenn ein Problem nach 45 Minuten nicht behoben ist, eine Nachricht an den Leiter des Rechenzentrums senden).

MySQL mit Zabbix Agent 2 überwachen

Einführung

Diese Seite führt Sie durch die Schritte, die erforderlich sind, um mit der grundlegenden Überwachung eines MySQL-Servers zu beginnen.

Für die Überwachung eines MySQL-Servers gibt es mehrere Ansätze: Zabbix Agent, Zabbix Agent 2 oder den Standard Open Database Connectivity (ODBC). Der Schwerpunkt dieses Leitfadens liegt auf der Überwachung eines MySQL-Servers mit Zabbix Agent 2, was aufgrund der nahtlosen Konfiguration in verschiedenen Umgebungen der **empfohlene** Ansatz ist. Diese Seite enthält jedoch auch Anweisungen für die anderen Ansätze, sodass Sie denjenigen auswählen können, der Ihren Anforderungen am besten entspricht.

Für wen dieser Leitfaden gedacht ist

Dieser Leitfaden ist für neue Zabbix-Benutzer gedacht und enthält die minimale Anzahl an Schritten, die erforderlich sind, um die grundlegende Überwachung eines MySQL-Servers zu aktivieren. Wenn Sie nach umfangreichen Anpassungsoptionen suchen oder eine fortgeschrittenere Konfiguration benötigen, lesen Sie den Abschnitt Konfiguration im Zabbix-Handbuch.

Voraussetzungen

Bevor Sie mit diesem Leitfaden fortfahren, müssen Sie den Zabbix Server, das Zabbix Frontend und den Zabbix Agent 2 gemäß den Anweisungen für Ihr Betriebssystem herunterladen und installieren.

Je nach Ihrer Umgebung können einige Schritte in diesem Leitfaden leicht abweichen. Dieser Leitfaden basiert auf einer Umgebung mit Ubuntu.

MySQL-Benutzer erstellen

Um einen MySQL-Server zu überwachen, benötigt Zabbix Zugriff darauf und auf seine Prozesse. Ihre MySQL-Installation verfügt bereits über einen Benutzer mit dem erforderlichen Zugriffslevel (den Benutzer ”zabbix”, der bei der Installation von Zabbix erstellt wurde), allerdings hat dieser Benutzer mehr Berechtigungen als für eine einfache Überwachung notwendig sind (Berechtigungen zum DROP von Datenbanken, DELETE von Einträgen aus Tabellen usw.). Daher muss ein MySQL-Benutzer ausschließlich für die Überwachung des MySQL-Servers erstellt werden.

1. Verbinden Sie sich mit dem MySQL-Client, erstellen Sie einen Benutzer ”zbx_monitor” (ersetzen Sie *<password>* für den Benutzer ”zbx_monitor” durch ein Passwort Ihrer Wahl), und GRANT Sie dem Benutzer die erforderlichen Berechtigungen:

```
mysql -u root -p
# Passwort eingeben:

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

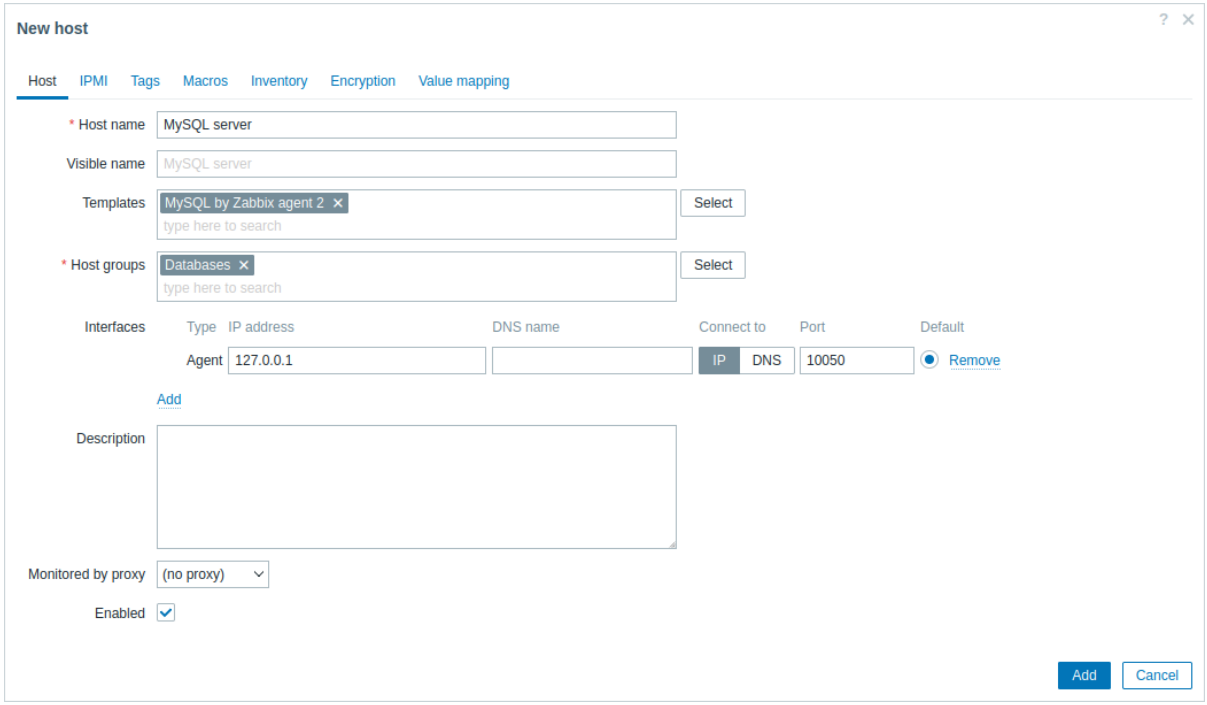
Sobald der Benutzer erstellt wurde, können Sie mit dem nächsten Schritt fortfahren.

Zabbix Frontend konfigurieren

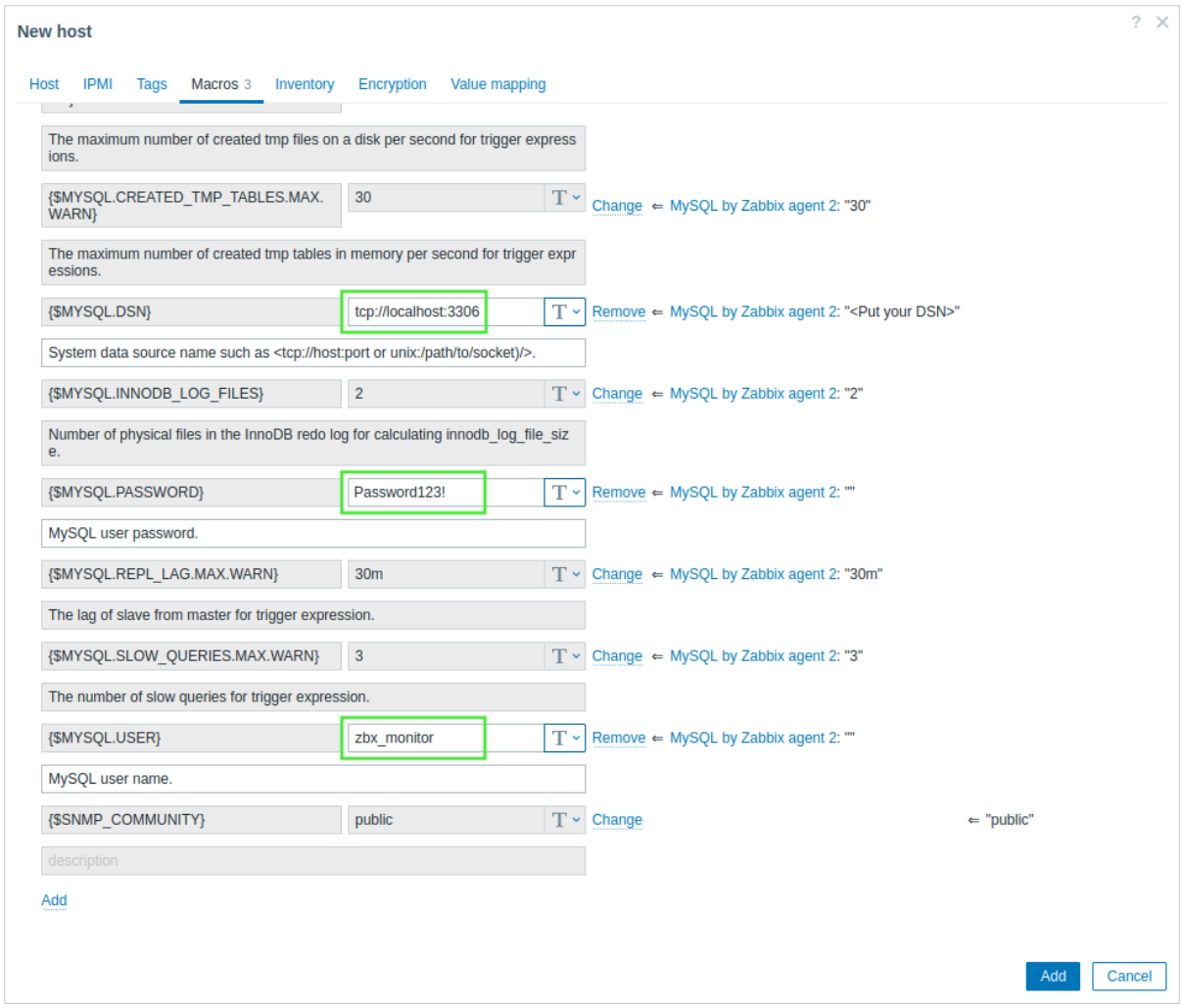
1. Melden Sie sich im Zabbix Frontend an.

2. Erstellen Sie einen Host in der Zabbix-Weboberfläche:

- Geben Sie im Feld *Host name* einen Hostnamen ein (z. B. ”MySQL server”).
- Geben Sie im Feld *Templates* die Vorlage ”MySQL by Zabbix agent 2” ein oder wählen Sie sie aus, die mit dem Host verknüpft wird.
- Geben Sie im Feld *Host groups* eine Hostgruppe ein oder wählen Sie sie aus (z. B. ”Databases”).
- Fügen Sie im Feld *Interfaces* eine Schnittstelle vom Typ ”Agent” hinzu und geben Sie die IP-Adresse Ihres MySQL-Servers an. In diesem Leitfaden wird ”127.0.0.1” (localhost) verwendet, um einen MySQL-Server zu überwachen, der auf demselben Rechner wie der Zabbix Server und Zabbix agent 2 installiert ist.



- Wechseln Sie auf der Registerkarte *Macros* zu *Inherited and host macros*, suchen Sie nach den folgenden Makros und klicken Sie neben dem Makrowert auf *Change*, um ihn zu aktualisieren:
  - {$MYSQL.DSN} - legen Sie die Datenquelle des MySQL-Servers fest (die Verbindungszeichenfolge einer benannten Sitzung aus der Konfigurationsdatei des MySQL-Zabbix-agent-2-Plugins). In diesem Leitfaden wird die Standard-Datenquelle ”tcp://localhost:3306” verwendet, um einen MySQL-Server zu überwachen, der auf demselben Rechner wie der Zabbix Server und Zabbix agent 2 installiert ist.
  - {$MYSQL.PASSWORD} - legen Sie das Passwort des zuvor erstellten MySQL-Benutzers ”zbx_monitor” fest.
  - {$MYSQL.USER} - legen Sie den Namen des zuvor erstellten MySQL-Benutzers ”zbx_monitor” fest.

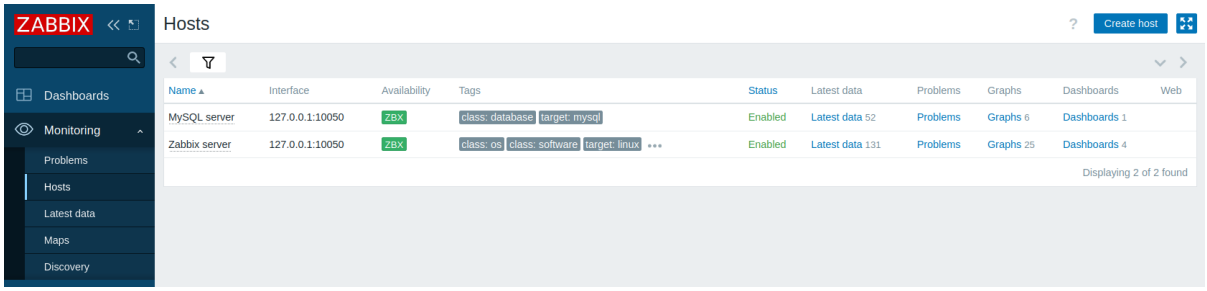


3. Klicken Sie auf *Add*, um den Host hinzuzufügen. Dieser Host wird Ihren MySQL-Server repräsentieren.

Erfasste Metriken anzeigen

Glückwunsch! Zu diesem Zeitpunkt überwacht Zabbix bereits Ihren MySQL-Server.

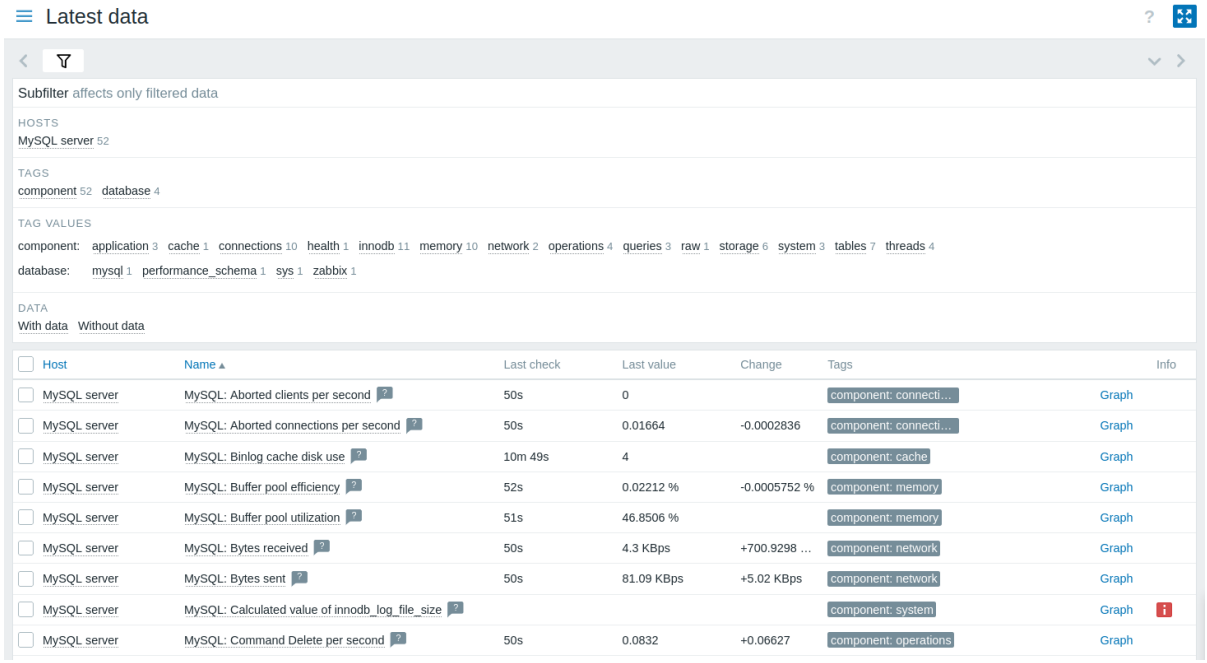
Um die erfassten Metriken anzuzeigen, navigieren Sie zum Menübereich *Monitoring > Hosts* und klicken Sie neben dem Host auf *Dashboards*.



Diese Aktion führt Sie zum Host-Dashboard (auf Vorlagenebene konfiguriert) mit den wichtigsten vom MySQL-Server erfassten Metriken.



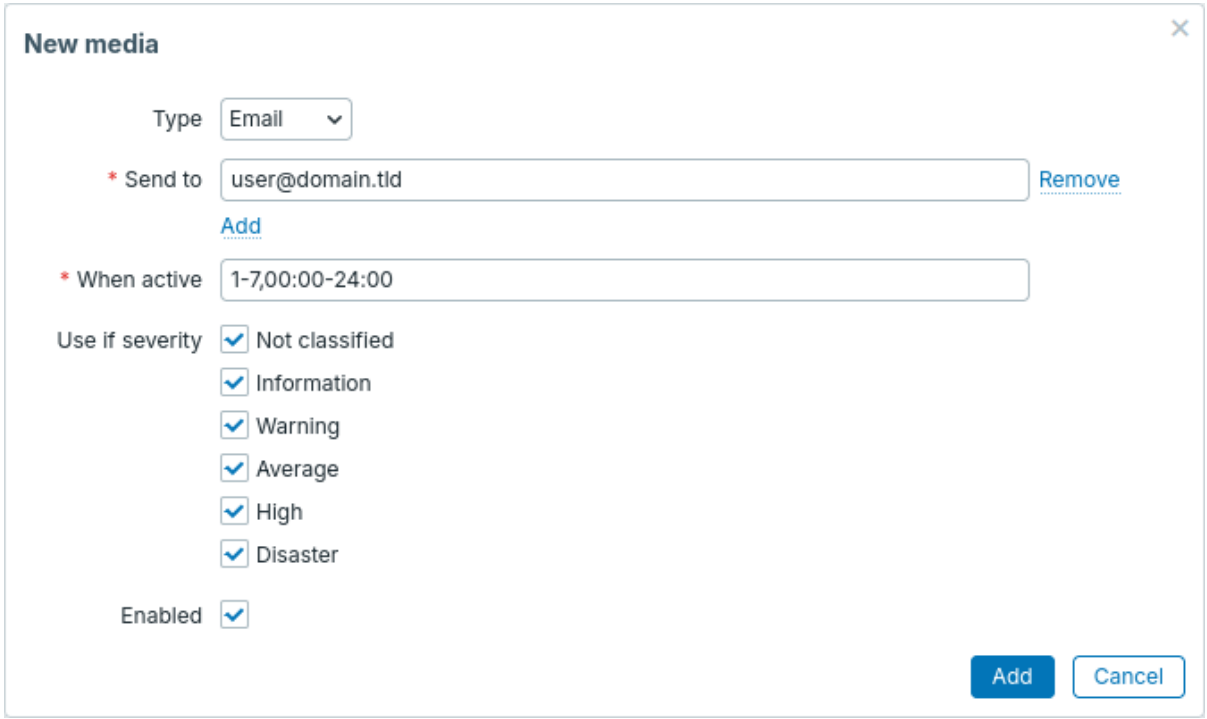
Alternativ können Sie im Menübereich *Monitoring > Hosts* auf *Neueste Daten* klicken, um alle zuletzt erfassten Metriken in einer Liste anzuzeigen. Beachten Sie, dass der Datenpunkt *MySQL: Calculated value of innodb_log_file_size* erwartungsgemäß keine Daten enthält, da der Wert aus den Daten der letzten Stunde berechnet wird.



Problembenachrichtigungen einrichten

Zabbix kann Sie mit verschiedenen Methoden über ein Problem in Ihrer Infrastruktur benachrichtigen. Diese Anleitung enthält grundlegende Konfigurationsschritte zum Senden von E-Mail-Benachrichtigungen.

1. Navigieren Sie zu *Benutzereinstellungen > Profil*, wechseln Sie zur Registerkarte *Medien* und fügen Sie Ihre E-Mail-Adresse hinzu.



2. Folgen Sie der Anleitung zum Empfangen einer Problembenachrichtigung.

Wenn Zabbix das nächste Mal ein Problem erkennt, sollten Sie eine Benachrichtigung per E-Mail erhalten.

Testen Sie Ihre Konfiguration

Um Ihre Konfiguration zu testen, können wir ein echtes Problem simulieren, indem wir die Host-Konfiguration im Zabbix Frontend aktualisieren.

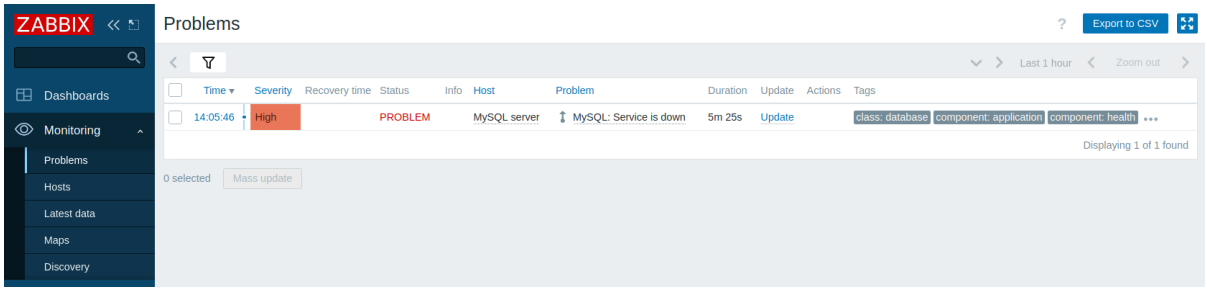
1. Öffnen Sie die Host-Konfiguration Ihres MySQL-Servers in Zabbix.

2. Wechseln Sie zur Registerkarte *Macros* und wählen Sie *Inherited and host macros* aus.

3. Klicken Sie neben dem beispielsweise zuvor konfigurierten {$MYSQL.USER}-Makrowert auf *Change* und legen Sie einen anderen MySQL-Benutzernamen fest.

4. Klicken Sie auf *Update*, um die Host-Konfiguration zu aktualisieren.

5. Nach einigen Augenblicken erkennt Zabbix das Problem "MySQL: Service is down", da keine Verbindung zum MySQL-Server hergestellt werden kann. Das Problem wird unter *Monitoring > Problems* angezeigt.



Wenn Benachrichtigungen konfiguriert sind, erhalten Sie außerdem die Problembenachrichtigung.

6. Setzen Sie den Makrowert wieder auf den vorherigen Wert zurück, um das Problem zu beheben und die Überwachung des MySQL-Servers fortzusetzen.

Andere Ansätze zur Überwachung von MySQL

Anstatt einen MySQL-Server mit Zabbix Agent 2 zu überwachen, können Sie auch Zabbix Agent oder den Standard Open Database Connectivity (ODBC) verwenden. Obwohl die Verwendung von Zabbix Agent 2 empfohlen wird, gibt es möglicherweise einige Setups, die Zabbix Agent 2 nicht unterstützen oder einen benutzerdefinierten Ansatz erfordern.

Der wesentliche Unterschied zwischen Zabbix Agent und ODBC liegt in der Methode der Datenerfassung - Zabbix Agent wird direkt auf dem MySQL-Server installiert und erfasst Daten mithilfe seiner integrierten Funktionalität, während ODBC auf einen ODBC-Treiber angewiesen ist, um eine Verbindung zum MySQL-Server herzustellen und Daten mithilfe von SQL-Abfragen abzurufen.

Obwohl viele der Konfigurationsschritte denen zur Überwachung eines MySQL-Servers mit Zabbix Agent 2 ähneln, gibt es einige wesentliche Unterschiede - Sie müssen Zabbix Agent oder ODBC konfigurieren, damit ein MySQL-Server überwacht werden kann. Die folgenden Anweisungen führen Sie durch diese **Unterschiede**.

MySQL mit Zabbix Agent überwachen

Um einen MySQL-Server mit Zabbix Agent zu überwachen, müssen Sie Zabbix Server, Zabbix Frontend und Zabbix Agent gemäß den Anweisungen für Ihr Betriebssystem herunterladen und installieren.

Nachdem Sie die erforderlichen Zabbix-Komponenten erfolgreich installiert haben, müssen Sie einen MySQL-Benutzer erstellen, wie im Abschnitt *MySQL-Benutzer erstellen* beschrieben.

Nachdem Sie den MySQL-Benutzer erstellt haben, müssen Sie Zabbix Agent so konfigurieren, dass eine Verbindung zum MySQL-Server hergestellt und dieser überwacht werden kann. Dazu gehört die Konfiguration mehrerer UserParameter für die Ausführung benutzerdefinierter Agent-Prüfungen sowie die Bereitstellung der erforderlichen Anmeldedaten für Zabbix Agent, um eine Verbindung zum MySQL-Server als der zuvor erstellte Benutzer "zbx_monitor" herzustellen.

Zabbix Agent konfigurieren

1. Navigieren Sie zum Verzeichnis für zusätzliche Konfigurationen des Zabbix Agent.

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

> **Attention:**
> Das Verzeichnis für zusätzliche Konfigurationen des Zabbix Agent sollte sich im selben Verzeichnis wie Ihre Agent-Konfigurationsdatei befinden. Der Standardpfad hängt vom Betriebssystem und der Installationsmethode von Zabbix ab.

Anstatt alle erforderlichen Benutzerparameter für die Überwachung des MySQL-Servers in der Konfigurationsdatei des Zabbix Agent zu definieren, werden diese Parameter in einer separaten Datei im Verzeichnis für zusätzliche Konfigurationen definiert.

2. Erstellen Sie eine Datei `template_db_mysql.conf` im Verzeichnis für zusätzliche Konfigurationen des Zabbix Agent.

```
vi template_db_mysql.conf
```

3. Kopieren Sie den Inhalt der Datei `template_db_mysql.conf` (die sich im Zabbix-Repository befindet) in die von Ihnen erstellte Datei `template_db_mysql.conf` und speichern Sie die Datei.

4. Starten Sie den Zabbix Agent neu, um seine Konfiguration zu aktualisieren.

```
systemctl restart zabbix-agent
```

Nachdem Sie die Benutzerparameter des Zabbix Agent konfiguriert haben, können Sie mit der Konfiguration der Zugangsdaten fortfahren, die dem Zabbix Agent den Zugriff auf den MySQL-Server ermöglichen.

5. Navigieren Sie zum Home-Verzeichnis des Zabbix Agent (falls es auf Ihrem System nicht vorhanden ist, müssen Sie es erstellen; Standard: */var/lib/zabbix*).

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

6. Erstellen Sie eine Datei *.my.cnf* im Home-Verzeichnis des Zabbix Agent.

```
vi .my.cnf
```

7. Kopieren Sie den folgenden Inhalt in die Datei *.my.cnf* (ersetzen Sie *<password>* durch das Passwort des Benutzers "zbx_monitor").

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

Zabbix Frontend konfigurieren und Ihre Konfiguration testen

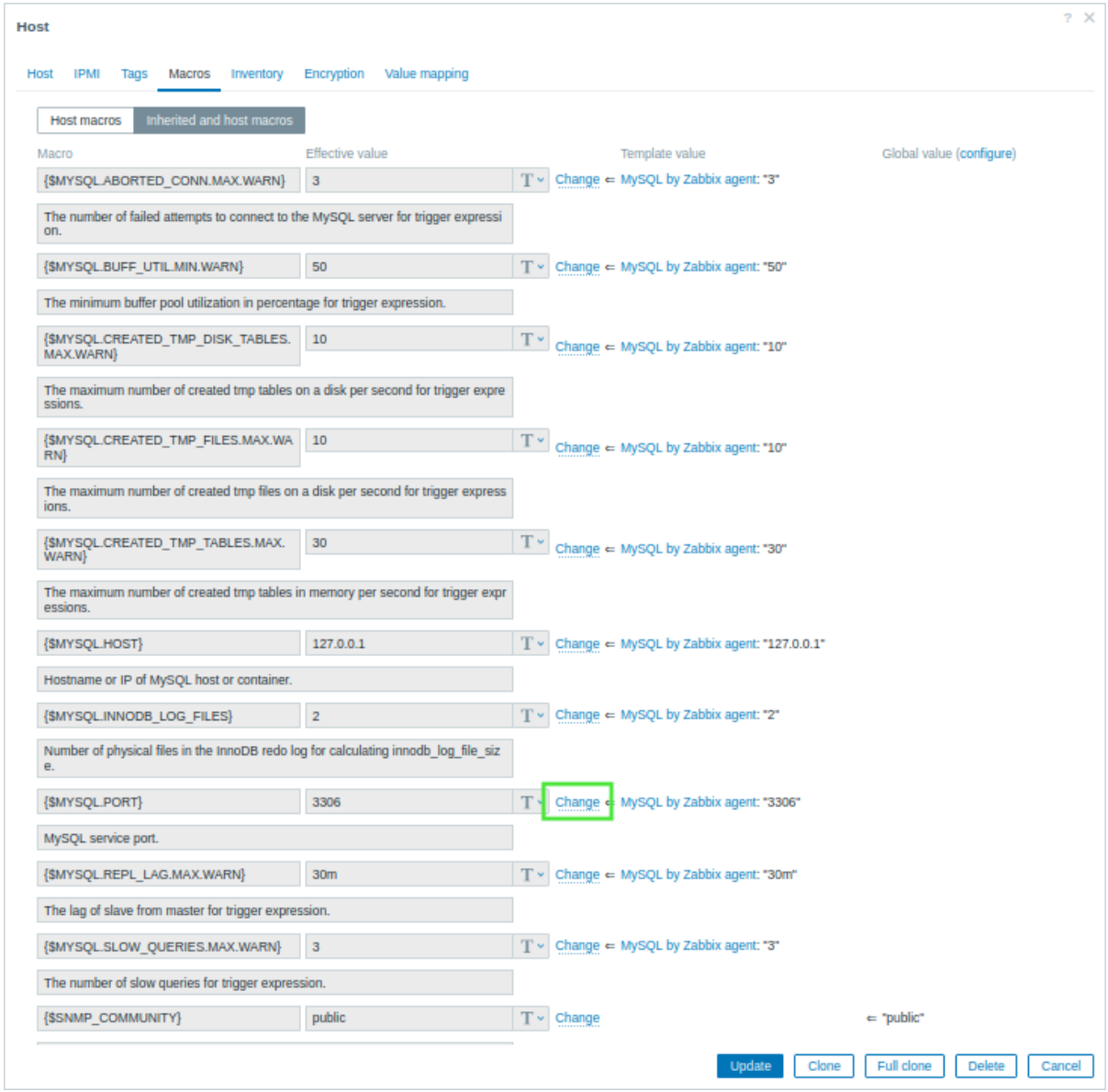
Um das Zabbix Frontend zu konfigurieren, folgen Sie den Anweisungen im Abschnitt *Zabbix Frontend konfigurieren* mit den folgenden Anpassungen:

- Geben Sie im Feld *Vorlagen* die Vorlage "MySQL by Zabbix agent" ein oder wählen Sie sie aus, die mit dem Host verknüpft wird.
- Die Konfiguration von *Makros* ist nicht erforderlich.

Nachdem Sie das Zabbix Frontend konfiguriert haben, können Sie gesammelte Metriken anzeigen und Problemalarme einrichten.

Um Ihre Konfiguration zu testen, folgen Sie den Anweisungen im Abschnitt *Ihre Konfiguration testen* mit den folgenden Anpassungen:

- Klicken Sie im Abschnitt *Vererbte und Host-Makros* der Host-Konfiguration des MySQL-Servers neben dem Wert des Makros {$MYSQL.PORT} auf *Ändern* und legen Sie einen anderen Port fest (z. B. "6033").



MySQL mit ODBC überwachen

Um einen MySQL-Server mit ODBC zu überwachen, müssen Sie den Zabbix-Server und das Zabbix-Frontend herunterladen und installieren.

Nachdem Sie die erforderlichen Zabbix-Komponenten erfolgreich installiert haben, müssen Sie einen MySQL-Benutzer erstellen, wie im Abschnitt *MySQL-Benutzer erstellen* beschrieben.

Nachdem Sie den MySQL-Benutzer erstellt haben, müssen Sie ODBC einrichten. Dazu gehört die Installation einer der am häufigsten verwendeten Open-Source-Implementierungen der ODBC-API - unixODBC - und eines unixODBC-Treibers sowie die Bearbeitung der Konfigurationsdatei des ODBC-Treibers.

ODBC konfigurieren

1. Installieren Sie unixODBC. Die empfohlene Methode zur Installation von unixODBC ist die Verwendung der Standard-Paketquellen des Linux-Betriebssystems.

```
apt install unixodbc
```

2. Installieren Sie den MariaDB unixODBC-Datenbanktreiber. Obwohl Sie eine MySQL-Datenbank haben, wird der MariaDB unixODBC-Treiber aus Kompatibilitätsgründen verwendet.

```
apt install odbc-mariadb
```

3. Prüfen Sie den Speicherort der ODBC-Konfigurationsdateien *odbcinst.ini* und *odbc.ini*.

```
odbcinst -j
```

Das Ergebnis der Ausführung dieses Befehls sollte in etwa wie folgt aussehen.

```
unixODBC 2.3.9
```

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

4. Um den ODBC-Treiber für die Überwachung einer MySQL-Datenbank zu konfigurieren, benötigen Sie den Treibernamen, der in der Datei *odbcinst.ini* angegeben ist. Im folgenden Beispiel der Datei *odbcinst.ini* lautet der Treibername "MariaDB Unicode".

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

5. Kopieren Sie den folgenden Inhalt in die Datei *odbc.ini* (ersetzen Sie *<password>* durch das Passwort des Benutzers "zbx_monitor"). Diese Anleitung verwendet "127.0.0.1" (localhost) als MySQL-Serveradresse, um einen MySQL-Server zu überwachen, der auf demselben Rechner wie der ODBC-Treiber installiert ist. Beachten Sie den Datenquellennamen (DSN) "test", der benötigt wird, wenn Sie Zabbix Frontend konfigurieren.

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

Zabbix Frontend konfigurieren und Ihre Konfiguration testen

Um das Zabbix Frontend zu konfigurieren, folgen Sie den Anweisungen im Abschnitt *Zabbix Frontend konfigurieren* mit den folgenden Anpassungen:

- Geben Sie im Feld *Vorlagen* die Vorlage "MySQL by ODBC" ein oder wählen Sie sie aus, die mit dem Host verknüpft wird.
- Die Konfiguration von *Schnittstellen* ist nicht erforderlich.
- Der Wert des Makros {$MYSQL.DSN} im Abschnitt *Geerbte und Host-Makros* der Host-Konfiguration des MySQL-Servers sollte auf den DSN-Namen aus der Datei *odbc.ini* gesetzt werden.

Nachdem Sie das Zabbix Frontend konfiguriert haben, können Sie gesammelte Metriken anzeigen, Problemalarme einrichten und Ihre Konfiguration testen.

Siehe auch

- Ein Datenpunkt erstellen - wie Sie mit der Überwachung zusätzlicher Metriken beginnen.
- Problem-Eskalationen - wie Sie mehrstufige Alarmszenarien erstellen (z. B. zuerst eine Nachricht an den Systemadministrator senden und dann, wenn ein Problem nach 45 Minuten nicht behoben ist, eine Nachricht an den Rechenzentrumsleiter senden).
- ODBC-Überwachung - wie Sie ODBC auf anderen Linux-Distributionen einrichten und wie Sie mit ODBC die Überwachung zusätzlicher datenbankbezogener Metriken starten.
- Vorlage *MySQL by Zabbix Agent* - zusätzliche Informationen zur Vorlage *MySQL by Zabbix Agent*.
- Vorlage *MySQL by Zabbix Agent 2* - zusätzliche Informationen zur Vorlage *MySQL by Zabbix Agent 2*.
- Vorlage *MySQL by ODBC* - zusätzliche Informationen zur Vorlage *MySQL by ODBC*.

VMware mit Zabbix überwachen

Einführung

Diese Seite führt Sie durch die Schritte, die erforderlich sind, um die grundlegende Überwachung von VMware zu starten.

Für wen dieser Leitfaden gedacht ist

Dieser Leitfaden ist für neue Zabbix-Benutzer gedacht und enthält die minimale Anzahl an Schritten, die erforderlich sind, um die grundlegende Überwachung von VMware zu aktivieren. Wenn Sie nach umfangreichen Anpassungsoptionen suchen oder eine fortgeschrittenere Konfiguration benötigen, lesen Sie den Abschnitt Überwachung virtueller Maschinen oder den Abschnitt Konfiguration im Zabbix-Handbuch.

Voraussetzungen

Bevor Sie mit diesem Leitfaden fortfahren, müssen Sie den Zabbix Server und das Zabbix Frontend gemäß den Anweisungen für Ihr Betriebssystem herunterladen und installieren.

Je nach Ihrer Umgebung können einige Schritte in diesem Leitfaden leicht abweichen. Dieser Leitfaden basiert auf einer Umgebung mit Ubuntu.

Es wird davon ausgegangen, dass VMware bereits konfiguriert ist. Dieser Leitfaden behandelt nicht die Konfiguration von VMware.

Zabbix Server konfigurieren

Für die Überwachung von VMware müssen die Zabbix-Prozesse des *vmware collectors* aktiviert werden. Weitere Informationen zur Durchführung der VMware-Überwachung finden Sie unter Überwachung virtueller Maschinen.

1. Öffnen Sie die Konfigurationsdatei des Zabbix Servers.

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

2. Suchen Sie den Parameter `StartVMwareCollectors` in der Konfigurationsdatei des Zabbix Servers und setzen Sie ihn auf 2 oder höher (der Standardwert ist 0).

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

StartVMwareCollectors=2
```

3. Starten Sie den Zabbix Server neu.

```
systemctl restart zabbix-server
```

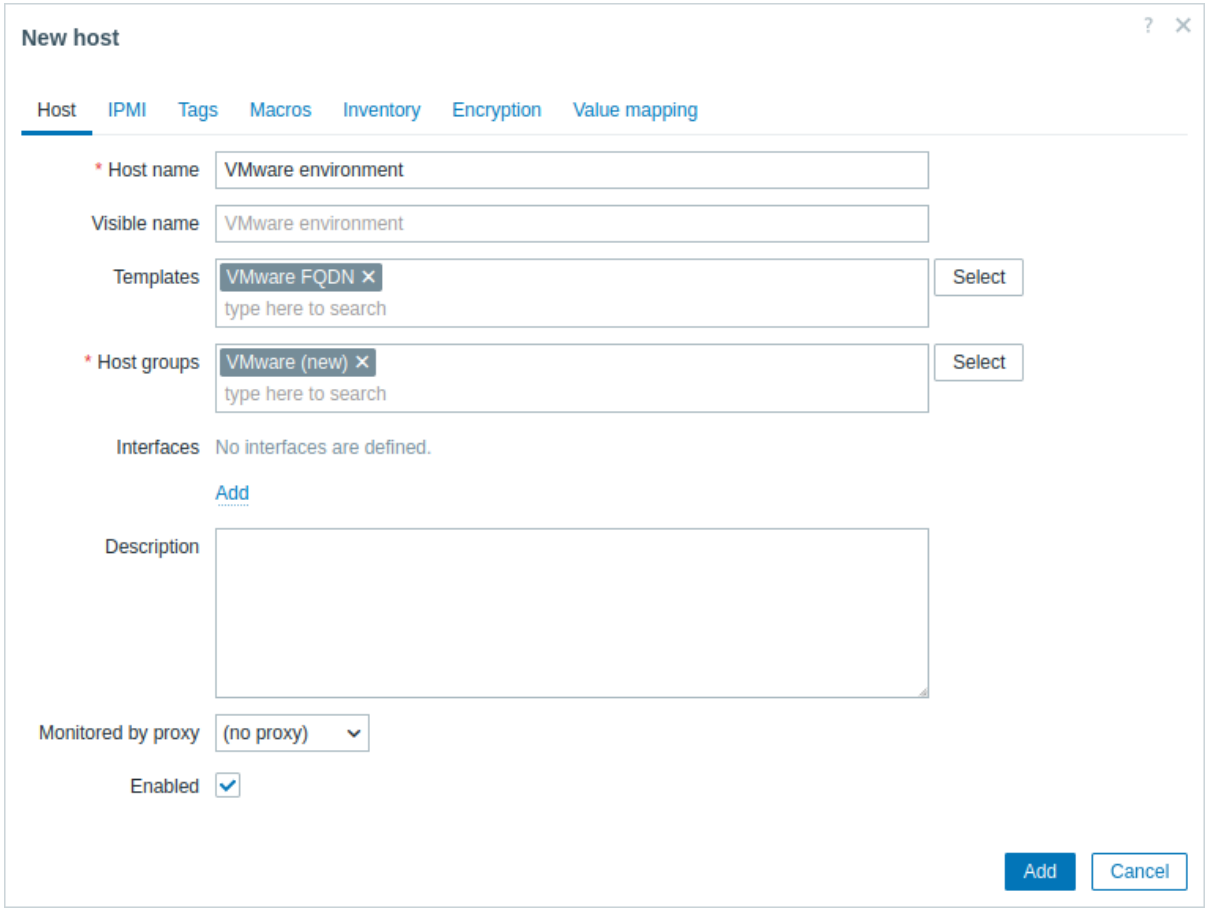
Sobald die Prozesse des *vmware collectors* gestartet wurden, fahren Sie mit dem nächsten Schritt fort.

Zabbix Frontend konfigurieren

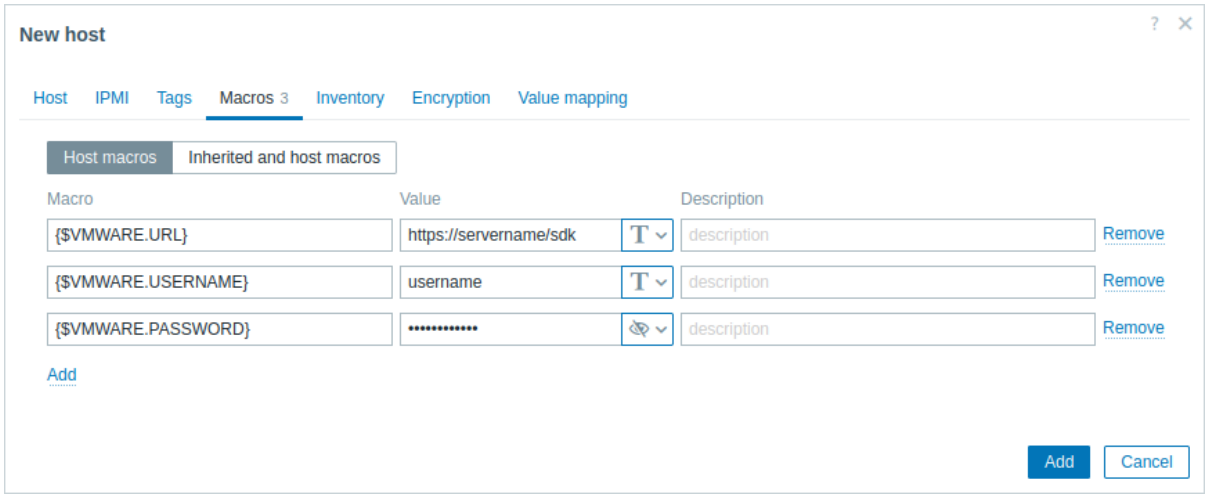
1. Melden Sie sich im Zabbix Frontend an.

2. Erstellen Sie einen Host in der Zabbix-Weboberfläche:

- Geben Sie im Feld *Host name* einen Hostnamen ein (zum Beispiel ”VMware environment”).
- Geben Sie im Feld *Templates* die Vorlage ”VMware FQDN” (oder ”VMware”) ein oder wählen Sie sie aus. Weitere Informationen zu diesen Vorlagen finden Sie unter Virtual machine monitoring.
- Geben Sie im Feld *Host groups* eine Hostgruppe ein oder wählen Sie sie aus (zum Beispiel eine neue Hostgruppe ”VMware”).



- Legen Sie auf der Registerkarte *Macros* die folgenden Host-Makros fest:
  - {$VMWARE.URL} - VMware-Dienst- (vCenter oder ESXi-Hypervisor) SDK-URL (https://servername/sdk)
  - {$VMWARE.USERNAME} - Benutzername des VMware-Dienstes
  - {$VMWARE.PASSWORD} - Passwort des VMware-Dienstes für {$VMWARE.USERNAME}



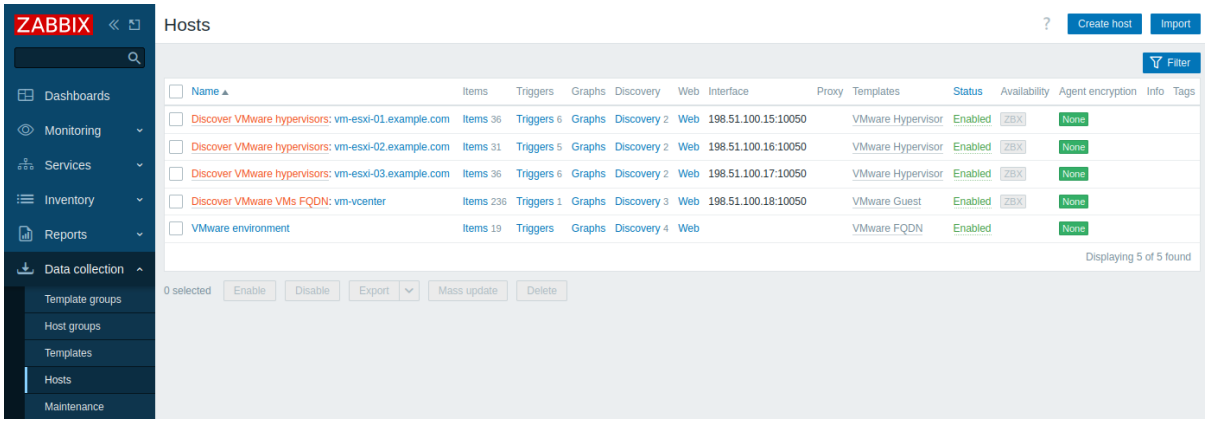
3. Klicken Sie auf die Schaltfläche *Add*, um den Host zu erstellen. Dieser Host wird Ihre VMware-Umgebung repräsentieren.

Erfasste Metriken anzeigen

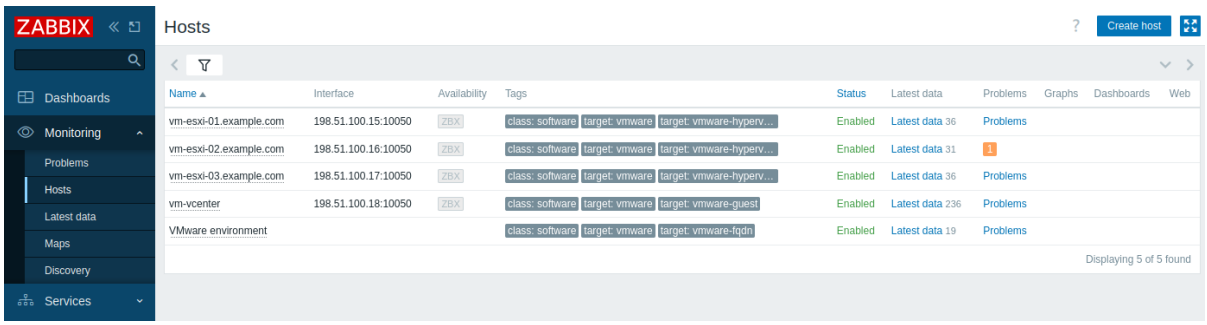
Glückwunsch! Zu diesem Zeitpunkt überwacht Zabbix Ihre VMware-Umgebung bereits.

Abhängig von der Konfiguration Ihrer VMware-Umgebung kann Zabbix die Erkennung durchführen und anschließend Hosts für die erkannten Entitäten erstellen. Beachten Sie, dass die Erkennung und Erstellung von Hosts bei Bedarf auch manuell ausgeführt werden kann.

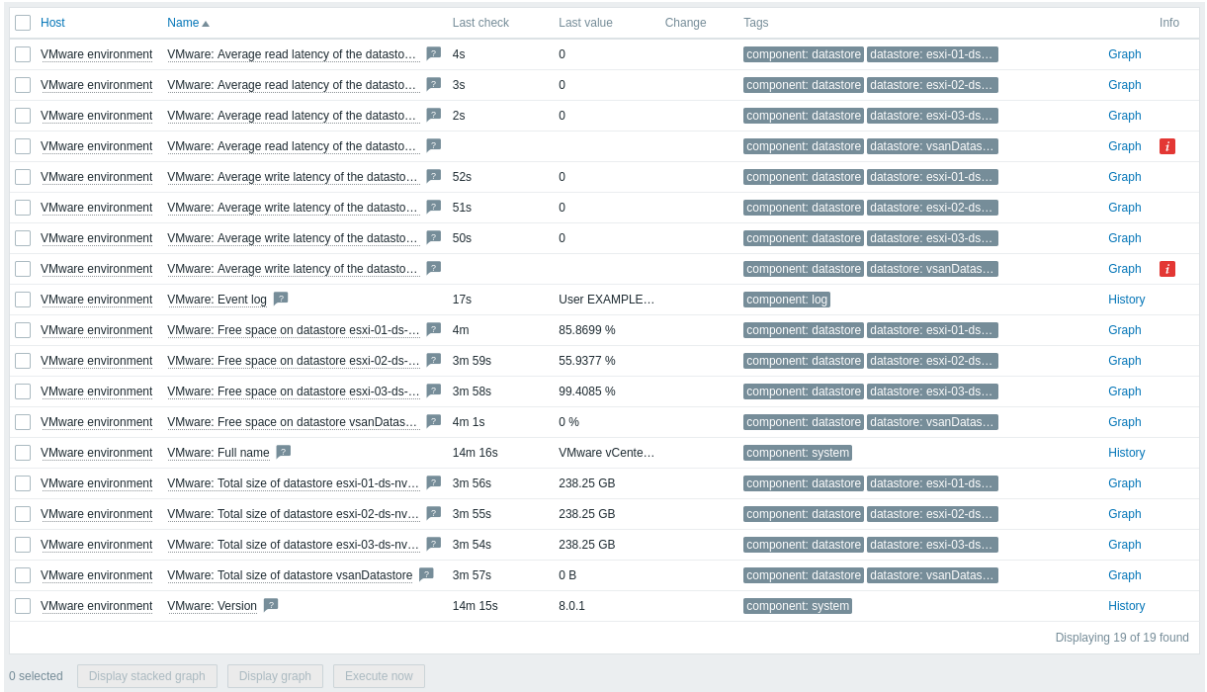
Um die erstellten Hosts anzuzeigen, navigieren Sie zum Menübereich *Datenerfassung > Hosts*.



Um die erfassten Metriken anzuzeigen, navigieren Sie zum Menübereich *Überwachung > Hosts* und klicken Sie neben dem erstellten Host "VMware environment" oder neben einem der für die erkannten Entitäten erstellten Hosts auf *Neueste Daten*.



Dadurch wird eine Liste aller zuletzt vom ausgewählten Host erfassten Metriken geöffnet.

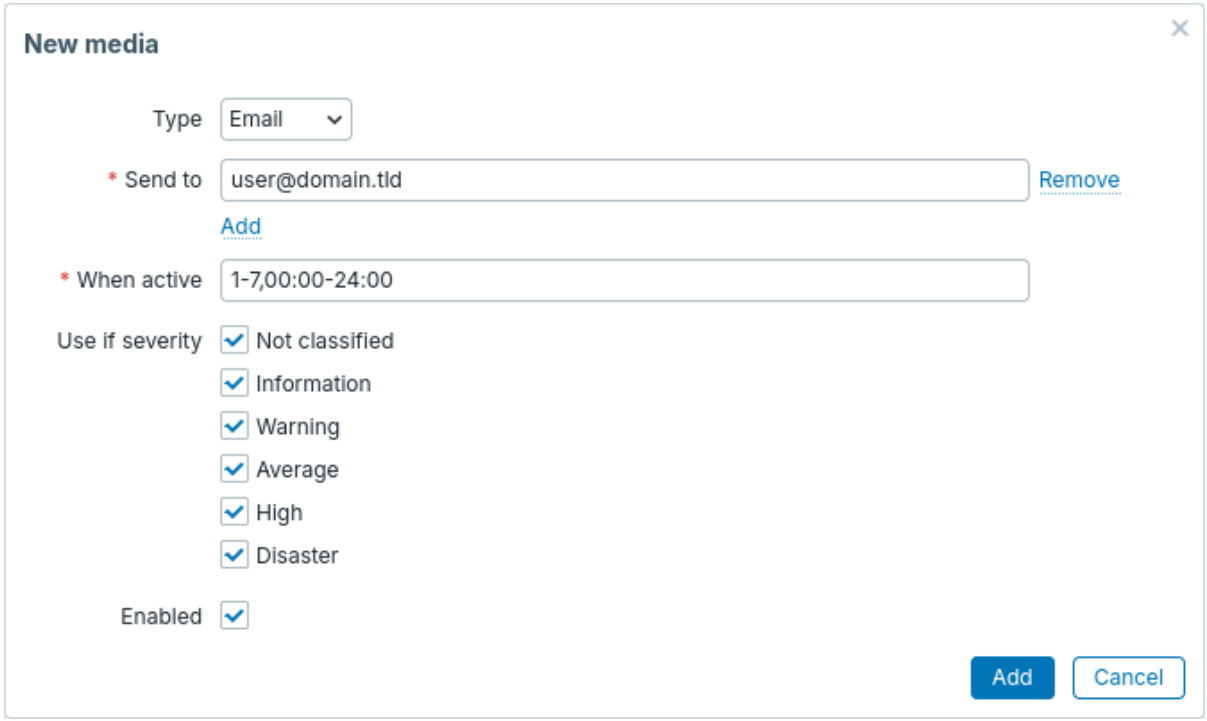


Beachten Sie, dass einige Datenpunkte keine Daten haben und sich im Status *Nicht unterstützt* befinden. Der Grund dafür ist, dass Zabbix auf dem jeweiligen Datenspeicher keine gültigen Leistungszähler finden kann, da dieser in der überwachten VMware-Umgebung nicht aktiviert ist.

Problemalarme einrichten

Zabbix kann Sie mit verschiedenen Methoden über ein Problem in Ihrer Infrastruktur benachrichtigen. Diese Anleitung enthält grundlegende Konfigurationsschritte zum Senden von E-Mail-Benachrichtigungen.

1. Navigieren Sie zu *Benutzereinstellungen > Profil*, wechseln Sie zur Registerkarte *Medien* und fügen Sie Ihre E-Mail-Adresse hinzu.



2. Folgen Sie der Anleitung zum Empfangen einer Problembenachrichtigung.

Wenn Zabbix das nächste Mal ein Problem erkennt, sollten Sie eine Benachrichtigung per E-Mail erhalten.

Siehe auch

- Erstellen eines Datenpunkts - wie Sie mit der Überwachung zusätzlicher Metriken beginnen.
- Problem-Eskalationen - wie Sie mehrstufige Alarmszenarien erstellen (z. B. zuerst eine Nachricht an den Systemadministrator senden und dann, wenn ein Problem innerhalb von 45 Minuten nicht behoben ist, eine Nachricht an den Rechenzentrumsleiter senden).
- Überwachung virtueller Maschinen - zusätzliche Informationen zur VMware-Überwachung (Datenerfassungsprozess, Konfigurationsoptionen des Servers, Hinweise zur Fehlerbehebung usw.).
- Datenpunktschlüssel für VMware-Überwachung - eine vollständige Liste der VMware-Metriken, die mit Zabbix überwacht werden können.
- Vorlage *VMware* - zusätzliche Informationen zur Vorlage *VMware*.
- Vorlage *VMware FQDN* - zusätzliche Informationen zur Vorlage *VMware FQDN*.

Netzwerkverkehr mit Zabbix überwachen

Einführung

Diese Seite führt Sie durch die Schritte, die erforderlich sind, um mit Zabbix die grundlegende Überwachung Ihres Netzwerkverkehrs zu starten.

Für wen dieser Leitfaden gedacht ist

Dieser Leitfaden ist für neue Zabbix-Benutzer gedacht und enthält die minimale Anzahl an Schritten, die erforderlich sind, um die grundlegende Überwachung Ihres Netzwerkverkehrs zu aktivieren. Wenn Sie nach umfangreichen Anpassungsoptionen suchen oder eine fortgeschrittenere Konfiguration benötigen, lesen Sie den Abschnitt Konfiguration im Zabbix-Handbuch.

Voraussetzungen

Bevor Sie mit dieser Anleitung fortfahren, müssen Sie den Zabbix Server, das Zabbix Frontend und den Zabbix Agent gemäß den Anweisungen für Ihr Betriebssystem herunterladen und installieren. Beachten Sie, dass Sie den Zabbix Agent auf dem Rechner installieren sollten, der die Überwachung des Datenverkehrs benötigt. Dies kann entweder derselbe Host sein, auf dem der Zabbix Server installiert ist, oder ein anderer Host.

Diese Anleitung enthält die Schritte zur Konfiguration der Überwachung des Netzwerkverkehrs der Schnittstelle *eth0* auf einem separaten Rechner mit dem Namen *Remote host*.

Zabbix für die Überwachung konfigurieren

Der Zabbix Agent kann Metriken im aktiven oder passiven Modus (gleichzeitig) erfassen. Weitere Informationen finden Sie unter Passive and active agent checks. In diesem Leitfaden wird die Überwachung mithilfe passiver Prüfungen beschrieben.

Zabbix Agent konfigurieren

1. Öffnen Sie die Agent-Konfigurationsdatei auf dem Rechner, auf dem der Agent installiert ist:

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

2. Fügen Sie dem Parameter *Server* die IP-Adresse oder den DNS-Namen Ihres Zabbix-Servers hinzu. Beispiel:

```
Server=192.0.2.22
```

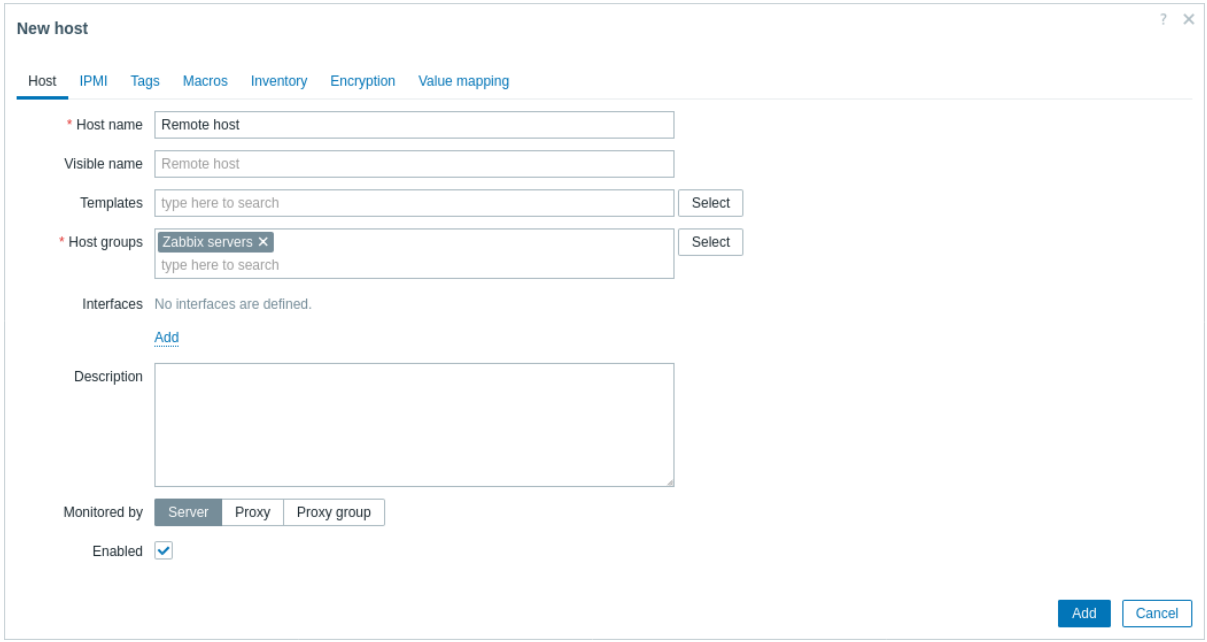
3. Starten Sie den Zabbix Agent neu:

```
systemctl restart zabbix-agent
```

Zabbix Frontend

1. Melden Sie sich im Zabbix Frontend an.

2. Erstellen Sie einen Host in der Zabbix-Weboberfläche und geben Sie die IP-Adresse oder den DNS-Namen des Computers an, auf dem der Agent installiert ist.

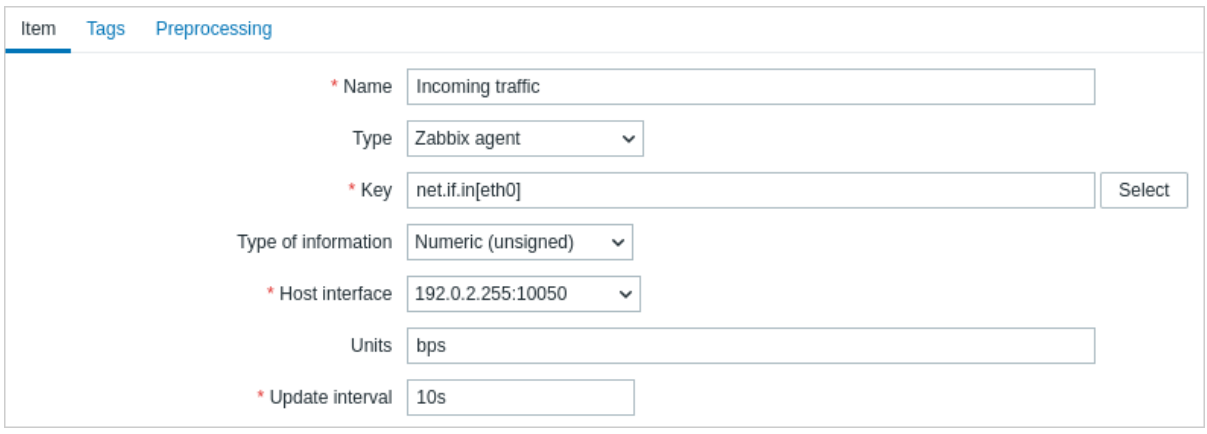


Datenpunkte erstellen

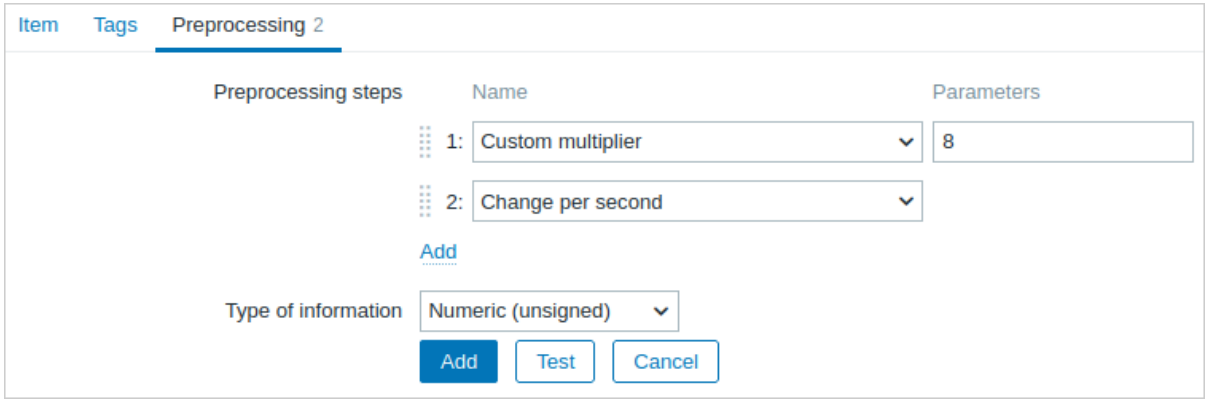
Befolgen Sie die Anweisungen unter einen Datenpunkt erstellen, um die Datenpunkte für die Verkehrsüberwachung hinzuzufügen, und zwar:

- Eingehender Verkehr
- Ausgehender Verkehr
- Gesamtverkehr

Eine einfache Konfiguration für den Datenpunkt zur Überwachung des eingehenden Verkehrs würde wie folgt aussehen:



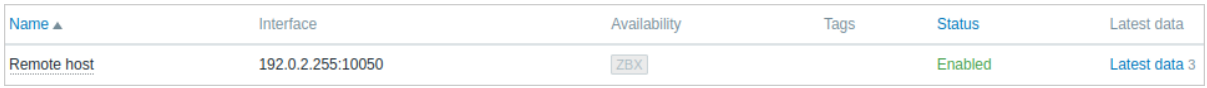
Um die erfassten Daten für den praktischen Einsatz geeignet zu machen, können Sie beim Erstellen der Datenpunkte einige Vorverarbeitung-Schritte festlegen. Im vorliegenden Fall können diese aus einer Multiplikation mit 8 (zur Umrechnung von Bytes in Bits) und der Darstellung als Änderung pro Sekunde bestehen.



Gesammelte Daten anzeigen

Herzlichen Glückwunsch! Zu diesem Zeitpunkt überwacht Zabbix bereits Ihren Netzwerkverkehr.

Um die gesammelten Metriken anzuzeigen, öffnen Sie den Menübereich *Monitoring > Hosts* und klicken Sie in der Zeile des Hosts auf *Latest data*.



Sie sehen die Verkehrsdaten.

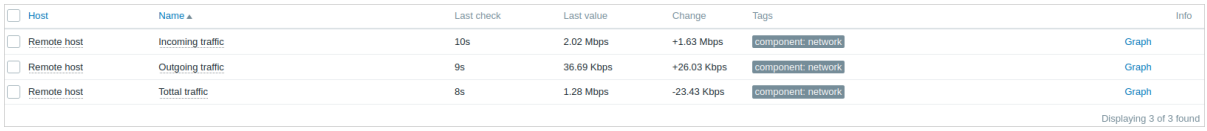
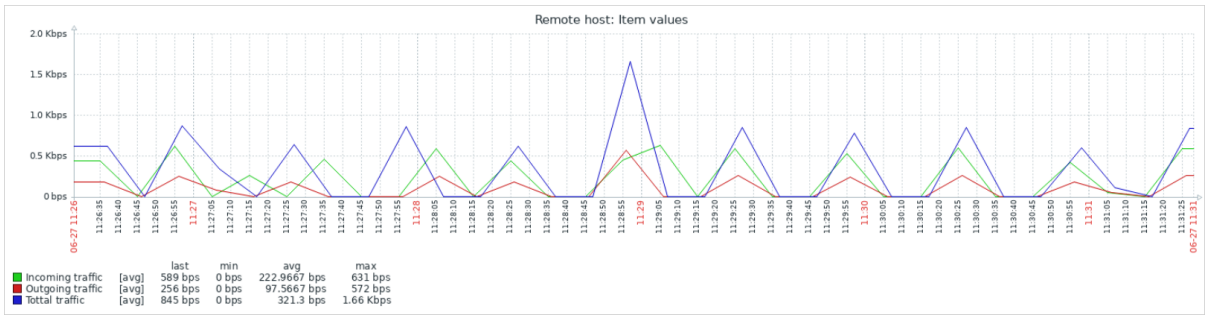


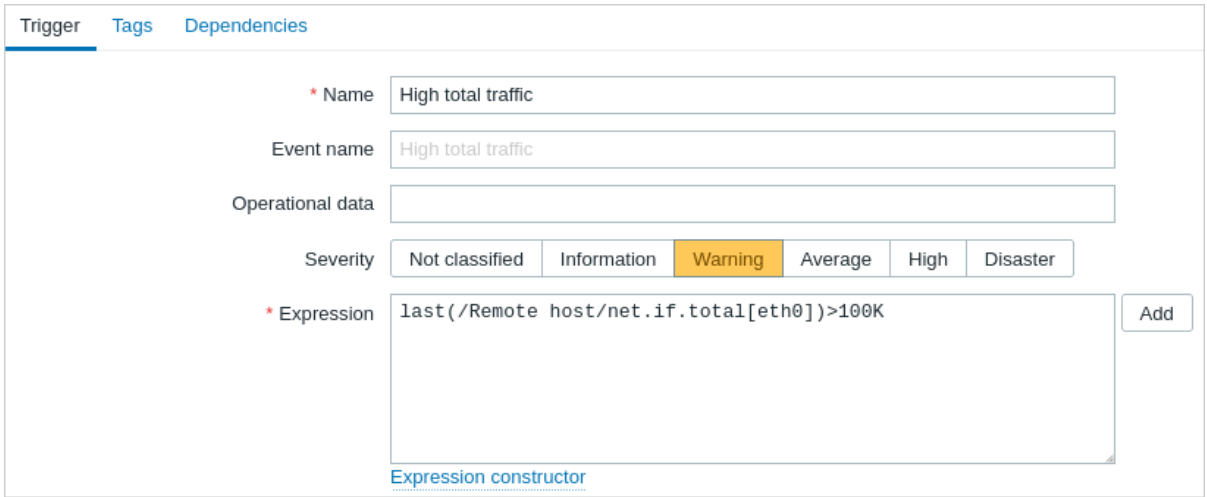
Diagramme anzeigen

Die gesammelten Daten können als Diagramme angezeigt werden. Um diese anzuzeigen, klicken Sie im Abschnitt *Neueste Daten* in der Zeile des Datenpunkts auf *Diagramm* oder wählen Sie die gewünschten Datenpunkte aus und klicken Sie unten auf *Diagramm anzeigen*.

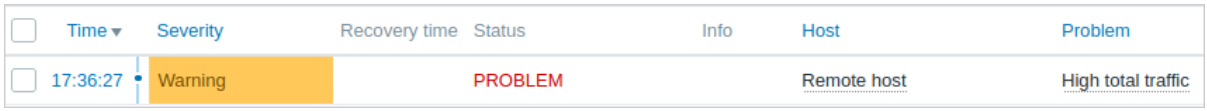


Auslöser konfigurieren

Sie können Auslöser festlegen, um ungewöhnlichen Netzwerkverkehr zu erkennen. Siehe die Anweisungen zum Konfigurieren eines Auslösers und fügen Sie den Auslöser hinzu, der signalisiert, dass der gesamte Verkehr beispielsweise zu hoch ist:



Lassen Sie nun den Verkehr den in der Auslöser-Ausdruck festgelegten Schwellenwert überschreiten und navigieren Sie zu *Monitoring* > *Probleme*, um zu prüfen, dass das Problem dort aufgeführt ist.



Problemalarme einrichten

Es gibt mehrere Möglichkeiten, Benachrichtigungen über das Problem zu erhalten. E-Mail ist die beliebteste davon. Folgen Sie den Anweisungen zum Einrichten einer Problembenachrichtigung per E-Mail. Sie können auch andere Medientypen auswählen, die für die Zustellung der Benachrichtigung verwendet werden sollen.

Siehe auch:

- Problem-Eskalationen - wie mehrstufige Alarmszenarien erstellt werden (z. B. zuerst eine Nachricht an den Systemadministrator senden und dann, wenn ein Problem nach 45 Minuten nicht behoben ist, eine Nachricht an den Leiter des Rechenzentrums senden).
- Problembestätigung - wie angezeigt wird, dass das Problem bekannt ist, wie Kommentare zur Behebung hinzugefügt werden und wie das Problem unterdrückt oder geschlossen wird.
- Linux mit Zabbix Agent überwachen - wie die grundlegende Überwachung der wichtigsten Datenpunkte durch Verknüpfen einer vorkonfigurierten Vorlage gestartet wird.

Netzwerkverkehr mit aktiven Prüfungen überwachen

Einführung

Diese Seite führt Sie durch die Schritte, die erforderlich sind, um mit der grundlegenden Überwachung Ihres Netzwerkverkehrs mit Zabbix mithilfe aktiver Prüfungen zu beginnen.

Für wen dieser Leitfaden gedacht ist

Dieser Leitfaden richtet sich an neue Zabbix-Benutzer und enthält die minimale Anzahl an Schritten, die erforderlich sind, um die grundlegende Überwachung Ihres Netzwerkverkehrs mithilfe aktiver Prüfungen zu aktivieren. Wenn Sie nach umfangreichen Anpassungsoptionen suchen oder eine fortgeschrittenere Konfiguration benötigen, lesen Sie den Abschnitt Konfiguration im Zabbix-Handbuch.

Voraussetzungen

Bevor Sie mit dieser Anleitung fortfahren, müssen Sie den Zabbix Server, das Zabbix Frontend und den Zabbix Agent gemäß den Anweisungen für Ihr Betriebssystem herunterladen und installieren. Beachten Sie, dass Sie den Zabbix Agent auf dem Rechner installieren sollten, auf dem die Überwachung des Netzwerkverkehrs erforderlich ist. Dies kann entweder derselbe Host sein, auf dem der Zabbix Server installiert ist, oder ein anderer Host.

Diese Anleitung enthält die Schritte zur Konfiguration der Überwachung des Netzwerkverkehrs der Schnittstelle *eth0* auf einem separaten Rechner mit dem Namen *Remote host*.

Zabbix für die Überwachung konfigurieren

Der Zabbix Agent kann Metriken im aktiven oder passiven Modus (gleichzeitig) erfassen. Weitere Informationen finden Sie unter Passive and active agent checks. In diesem Leitfaden wird die Überwachung mithilfe von **aktiven Prüfungen** beschrieben.

Konfigurieren des Zabbix Agent

1. Öffnen Sie die Konfigurationsdatei des Agent auf dem Rechner, auf dem der Agent installiert ist.

Wenn Sie den Zabbix Agent verwenden:

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

Wenn Sie Zabbix Agent 2 verwenden:

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

2. Fügen Sie die IP-Adresse oder den DNS-Namen (und optional den Port) Ihres Zabbix Servers zum Parameter ServerActive hinzu. Beispiel:

```
ServerActive=192.0.2.1:10051
```

Der Zabbix Agent verwendet diese Adresse, um eine Verbindung zum Trapper-Port des Zabbix Servers (Standard: 10051) herzustellen und die Konfigurationsdaten für aktive Prüfungen anzufordern.

3. Definieren Sie den Parameter Hostname. Dieser muss mit dem Hostnamen übereinstimmen, der im Zabbix Frontend definiert wird. In unserem Beispiel lautet er:

```
Hostname=Remote host
```

Der Wert von Hostname muss übereinstimmen, da der Zabbix Agent ihn bei aktiven Prüfungen verwendet, um die korrekte Host-Konfiguration vom Server abzurufen. Konkret initiiert der Agent eine Verbindung zum Server und identifiziert sich anhand des Werts von Hostname. Der Server stellt anschließend die Überwachungskonfiguration für diesen Host bereit. Wenn diese Werte voneinander abweichen, erhält der Agent nicht die passende Konfiguration, was zu fehlenden Metriken oder Problemen bei der Überwachung führt.

4. Starten Sie den Zabbix Agent neu.

Wenn Sie den Zabbix Agent verwenden:

```
systemctl restart zabbix-agent
```

Wenn Sie Zabbix Agent 2 verwenden:

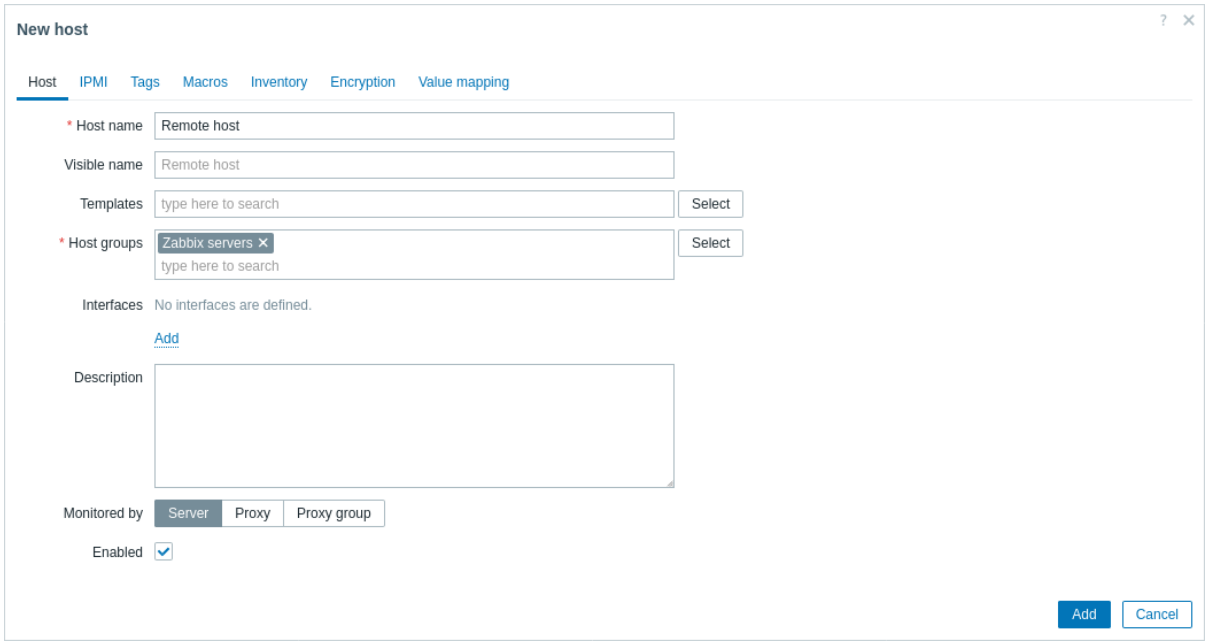
```
systemctl restart zabbix-agent2
```

Zabbix Frontend

1. Melden Sie sich im Zabbix Frontend an.

2. Erstellen Sie einen Host in der Zabbix-Weboberfläche.

- Geben Sie im Feld *Host name* einen Hostnamen ein (z. B. ”Remote host”), der mit dem zuvor in der Agent-Konfigurationsdatei definierten Wert des Parameters Hostname übereinstimmt.
- Geben Sie im Feld *Host groups* eine Hostgruppe ein oder wählen Sie sie aus (z. B. ”Zabbix servers”).
- Lassen Sie *Interfaces* undefiniert; für aktive Prüfungen ist keine Schnittstelle erforderlich, da der Agent die Verbindung zum Server initiiert, anstatt auf eine Verbindung vom Server zu warten.



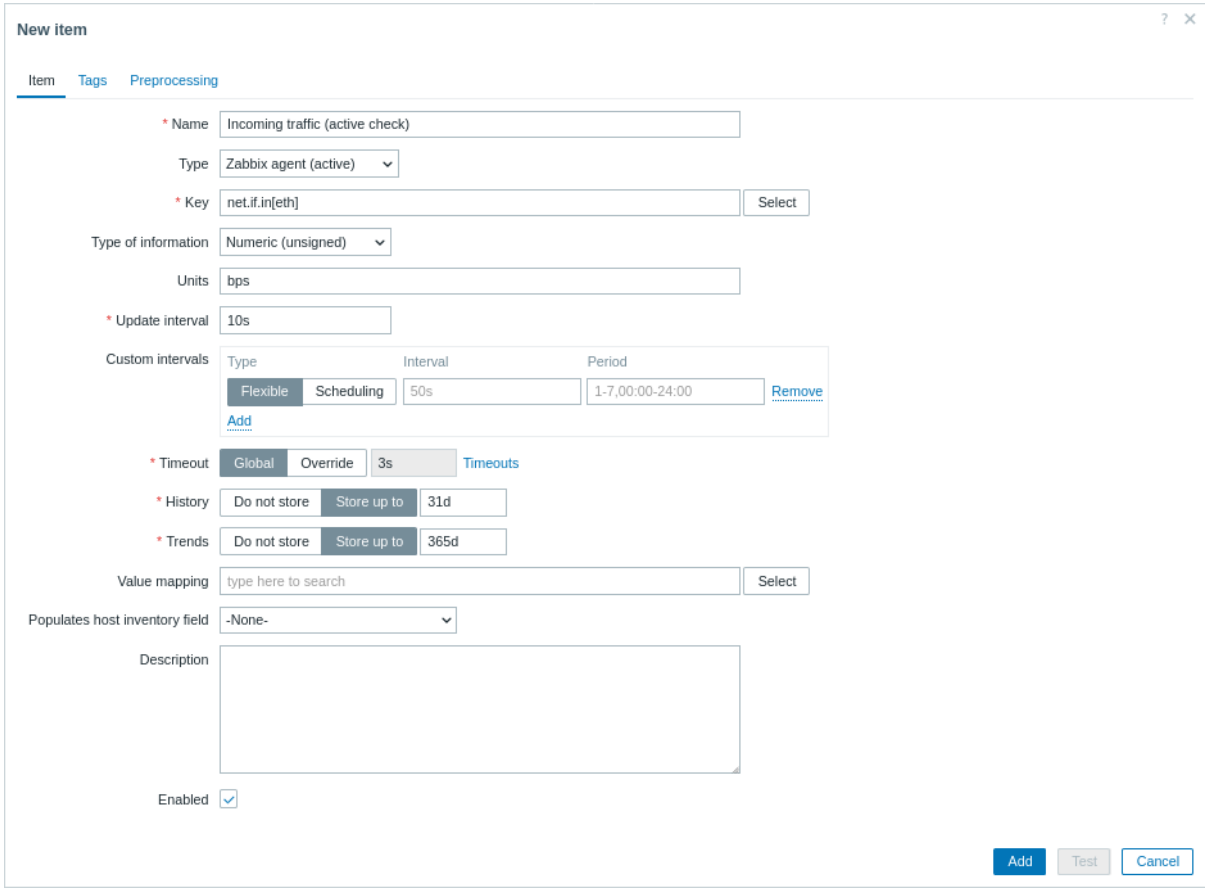
3. Klicken Sie auf *Add*, um den Host hinzuzufügen. Dieser Host repräsentiert den überwachten Linux-Rechner.

Elemente erstellen

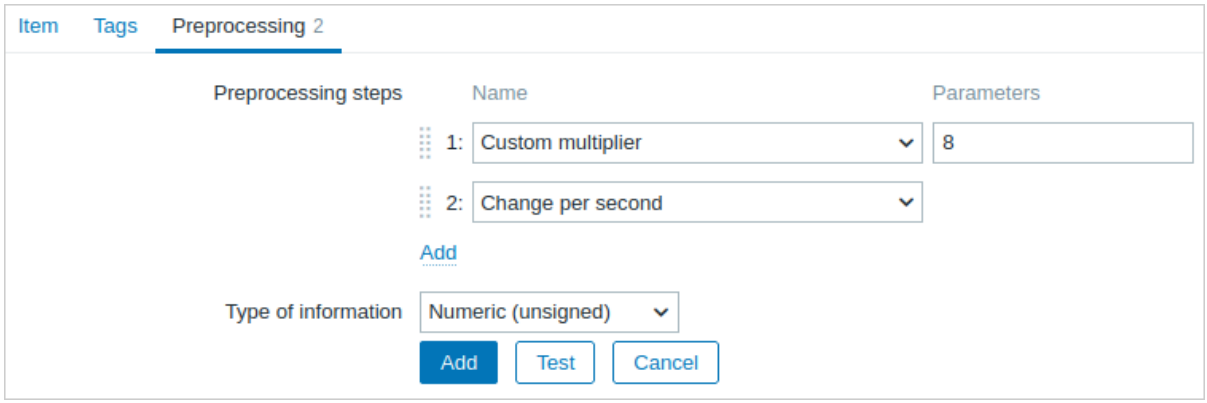
Folgen Sie den Anweisungen unter Ein Datenpunkt erstellen, um die Datenpunkte für die Verkehrsüberwachung hinzuzufügen, und zwar:

- Eingehender Verkehr
- Ausgehender Verkehr
- Gesamtverkehr

Eine einfache Konfiguration für den Datenpunkt zur Überwachung des eingehenden Verkehrs mit einer aktiven Prüfung würde wie folgt aussehen:



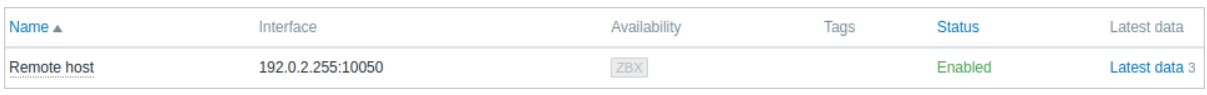
Damit die erfassten Daten für den praktischen Einsatz geeignet sind, können Sie beim Erstellen der Datenpunkte einige Vorverarbeitung-Schritte festlegen. Im vorliegenden Fall können dies eine Multiplikation mit 8 (um Bytes in Bits umzuwandeln) und die Darstellung als Änderung pro Sekunde sein.



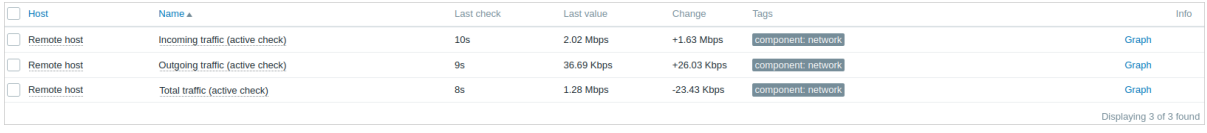
Gesammelte Daten anzeigen

Herzlichen Glückwunsch! Zu diesem Zeitpunkt überwacht Zabbix bereits Ihren Netzwerkverkehr.

Um die gesammelten Metriken anzuzeigen, öffnen Sie den Menübereich *Monitoring > Hosts* und klicken Sie in der Zeile des Hosts auf *Latest data*.



Sie sehen die Verkehrsdaten.



Siehe auch:

- Graphs anzeigen - wie die gesammelten Daten als Graphen angezeigt werden.
- Auslöser konfigurieren - wie Auslöser eingerichtet werden, um ungewöhnlichen Netzwerkverkehr zu erkennen.

- Problembenachrichtigungen einrichten - wie Benachrichtigungen über Problemsituationen eingerichtet werden.
- Problem-Eskalationen - wie mehrstufige Alarmszenarien erstellt werden (z. B. zuerst eine Nachricht an den Systemadministrator senden und dann, wenn ein Problem nach 45 Minuten nicht behoben ist, eine Nachricht an den Rechenzentrumsleiter senden).
- Problembestätigung - wie angegeben wird, dass das Problem bekannt ist, Kommentare zur Behebung hinzugefügt werden und das Problem unterdrückt oder geschlossen wird.
- Linux mit Zabbix Agent überwachen - wie mit der grundlegenden Überwachung der wichtigsten Datenpunkte durch Verknüpfung einer vorkonfigurierten Vorlage begonnen wird.
- Installation aus Paketen - wie Zabbix-Komponenten mithilfe offizieller RPM- und DEB-Pakete für verschiedene Linux-Distributionen installiert werden, um Zugriff auf die neuesten Funktionen und Fehlerbehebungen zu gewährleisten.

Websites mit Browser-Datenpunkten überwachen

Einführung

Diese Seite führt Sie durch die Schritte, die erforderlich sind, um mit Browser-Datenpunkten die grundlegende Überwachung von Websites zu starten.

Für wen dieser Leitfaden gedacht ist

Dieser Leitfaden richtet sich an neue Zabbix-Benutzer und enthält die minimale Anzahl an Schritten, die erforderlich sind, um die grundlegende Überwachung von Websites mit Browser-Datenpunkten zu aktivieren. Wenn Sie nach umfangreichen Anpassungsoptionen suchen oder eine fortgeschrittenere Konfiguration benötigen, lesen Sie die Seite Browser-Datenpunkte oder den Abschnitt Konfiguration im Zabbix-Handbuch.

Voraussetzungen

Bevor Sie mit dieser Anleitung fortfahren, müssen Sie den Zabbix Server und das Zabbix Frontend gemäß den Anweisungen für Ihr Betriebssystem herunterladen und installieren.

Je nach Ihrer Umgebung können einige Schritte in dieser Anleitung leicht abweichen. Diese Anleitung basiert auf einer Umgebung mit Ubuntu.

WebDriver konfigurieren

Browser-Items erfordern ein Automatisierungs-Framework (entweder Selenium Server oder ein reiner WebDriver, zum Beispiel ChromeDriver) als Web-Test-Endpunkt, der einen Browser steuert und mit ihm interagiert, indem er Testbefehle wie das Klicken auf Schaltflächen oder das Eingeben von Text ausführt.
Als Beispiel verwendet diese Anleitung Selenium Server mit Chrome in einem Docker-Container.

Es wird davon ausgegangen, dass Docker bereits konfiguriert ist. Diese Anleitung behandelt die Konfiguration von Docker nicht. Anweisungen zur Installation finden Sie unter Install Docker Engine on Ubuntu.

1. Starten Sie Selenium Server mit Chrome in einem Docker-Container mit den folgenden Optionen:

- **docker run --name browser** - startet einen neuen Docker-Container mit dem Namen "browser";
- **-p 4444:4444** - ordnet Port 4444 auf Ihrem Host dem Port 4444 im Container zu (dies ist der Port, den Selenium Server zum Annehmen von Befehlen verwendet);
- **-p 7900:7900** - ordnet Port 7900 auf Ihrem Host dem Port 7900 im Container zu (dies ist der Port, den der Virtual Network Computing (VNC)-Server verwendet, damit Sie die Browser-GUI aus der Ferne anzeigen können; dafür ist ein VNC-Client erforderlich);
- **--shm-size="2g"** - weist dem Container 2 GB gemeinsamen Speicher zu (dies ist wichtig, damit Chrome ordnungsgemäß ausgeführt werden kann, da es eine beträchtliche Menge an gemeinsamem Speicher benötigt, um Abstürze zu vermeiden);
- **-d** - führt den Container im Detached-Modus aus, d. h. er läuft im Hintergrund;
- **selenium/standalone-chrome:latest** - gibt das zu verwendende Docker-Image an; in diesem Fall die neueste Version von Selenium Server mit Chrome.

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

2. Stellen Sie sicher, dass der Docker-Container `browser` ausgeführt wird und erreichbar ist.

- Rufen Sie die IP-Adresse des Containers ab (in diesem Beispiel 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
#####     ...
```

- Testen Sie die Verbindung zum Container mit Ncat:

```
nc -zv 192.0.2.1 4444

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

- Rufen Sie den Webseiteninhalt von Selenium Server mit curl ab:

```
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:
Zur Fehlerbehebung lesen Sie die Docker-Dokumentation.

Zabbix Server konfigurieren

Browser-Datenpunkte werden von Zabbix-Prozessen des Typs *Browser-Poller* ausgeführt und verarbeitet. Diese müssen durch Anpassung des Server-Konfigurationsparameters StartBrowserPollers aktiviert werden. Zusätzlich sollte der Parameter WebDriverURL den zuvor konfigurierten Endpunkt für Webtests angeben.

Standardmäßig ist der Parameter StartBrowserPollers auf 1 gesetzt. Daher müssen Sie lediglich den Endpunkt für Webtests angeben.

1. Öffnen Sie die Server-Konfigurationsdatei von Zabbix:

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

2. Suchen Sie den Parameter WebDriverURL in der Zabbix-Server-Konfigurationsdatei und legen Sie ihn fest:

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

WebDriverURL=192.0.2.1:4444
```

3. Starten Sie den Zabbix Server neu.

```
systemctl restart zabbix-server
```

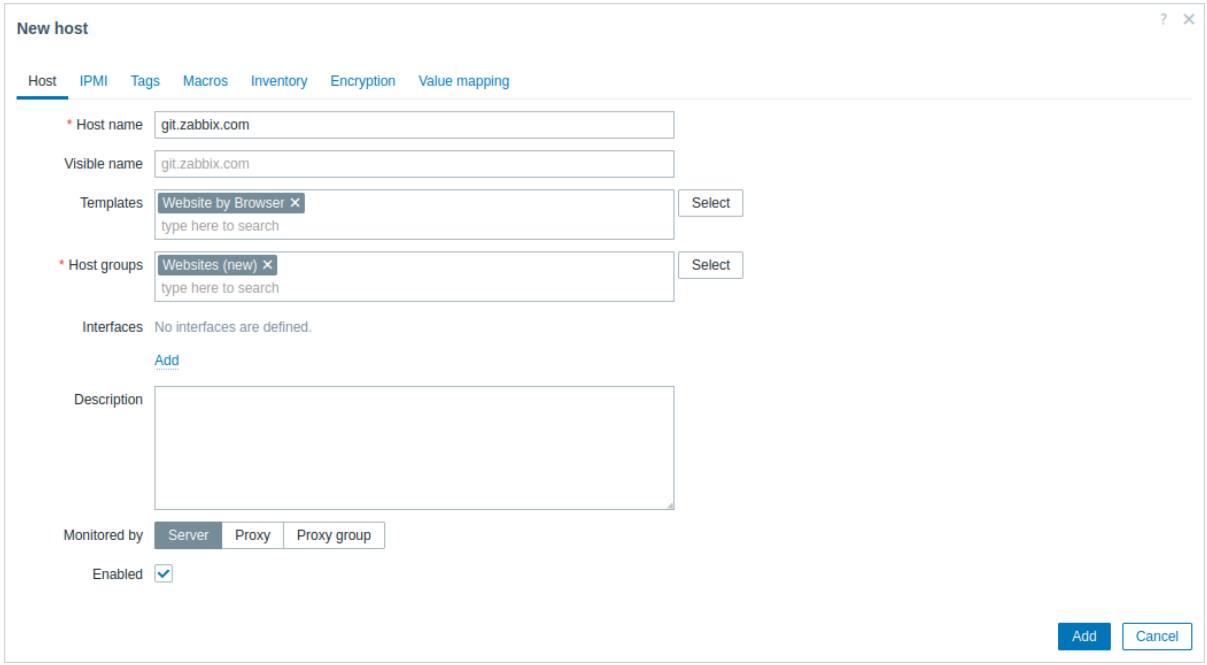
Zabbix Frontend konfigurieren

1. Melden Sie sich im Zabbix Frontend an.

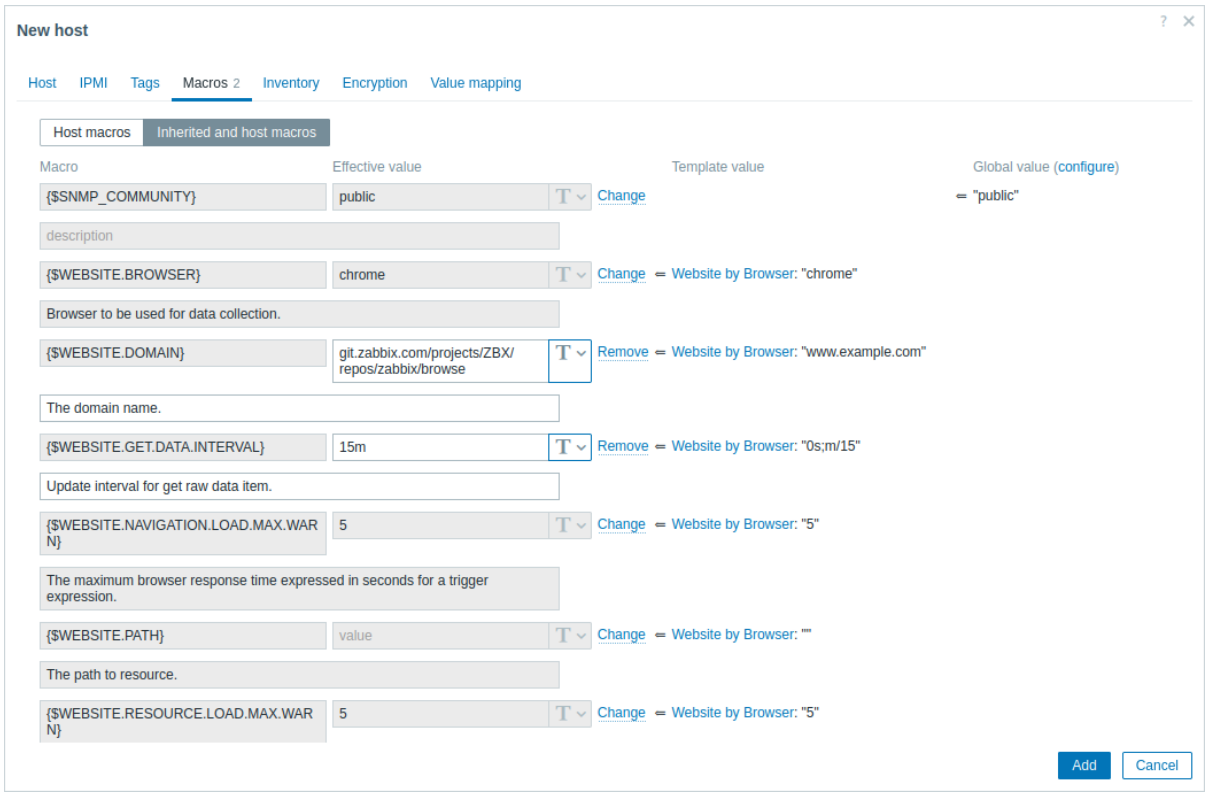
2. Erstellen Sie einen Host in der Zabbix-Weboberfläche:

- Geben Sie im Feld *Host name* einen Hostnamen ein (zum Beispiel "git.zabbix.com").
- Geben Sie im Feld *Templates* die Vorlage "Website by Browser" ein oder wählen Sie sie aus. Weitere Informationen zu dieser Vorlage finden Sie unter Website by Browser.

- Geben Sie im Feld *Host groups* eine Hostgruppe ein oder wählen Sie sie aus (zum Beispiel eine neue Hostgruppe ”Websites”).



- Wechseln Sie auf der Registerkarte *Macros* zu *Inherited and host macros*, suchen Sie nach den folgenden Makros und klicken Sie neben dem Makrowert auf *Change*, um ihn zu aktualisieren:
  - {$WEBSITE.DOMAIN} - Domainname (zum Beispiel git.zabbix.com/projects/ZBX/repos/zabbix/browse)
  - {$WEBSITE.GET.DATA.INTERVAL} - Aktualisierungsintervall für Datenpunkte (zum Beispiel 15m)

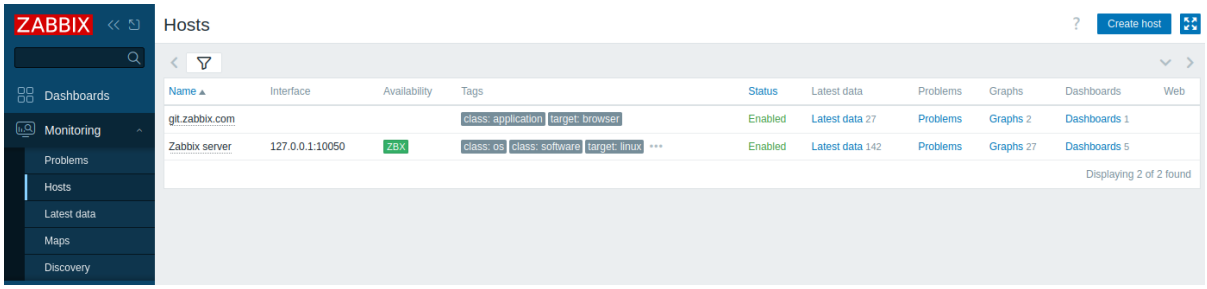


3. Klicken Sie auf die Schaltfläche *Add*, um den Host zu erstellen. Dieser Host repräsentiert die Website, die Sie überwachen möchten.

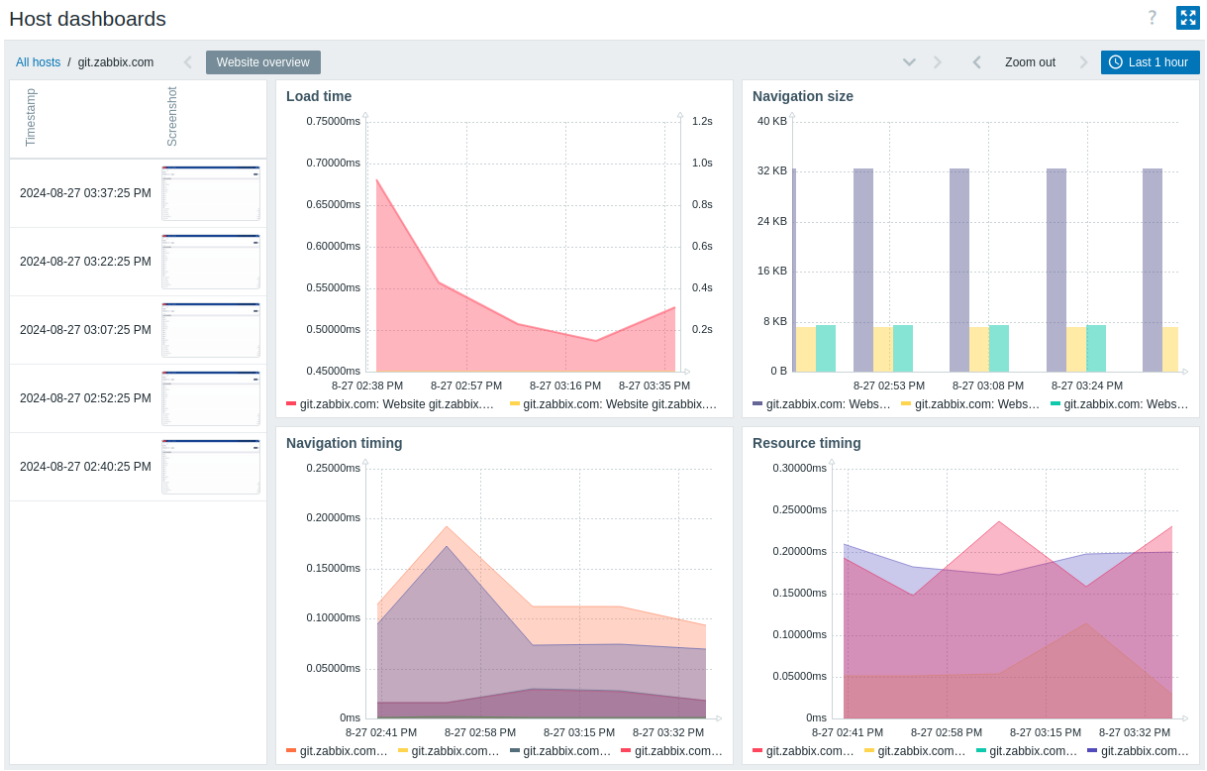
Erfasste Metriken anzeigen

Glückwunsch! Zu diesem Zeitpunkt überwacht Zabbix die von Ihnen angegebene Website bereits.

Um die erfassten Metriken anzuzeigen, navigieren Sie zum Menüabschnitt *Monitoring > Hosts* und klicken Sie neben dem Host auf *Dashboards*.



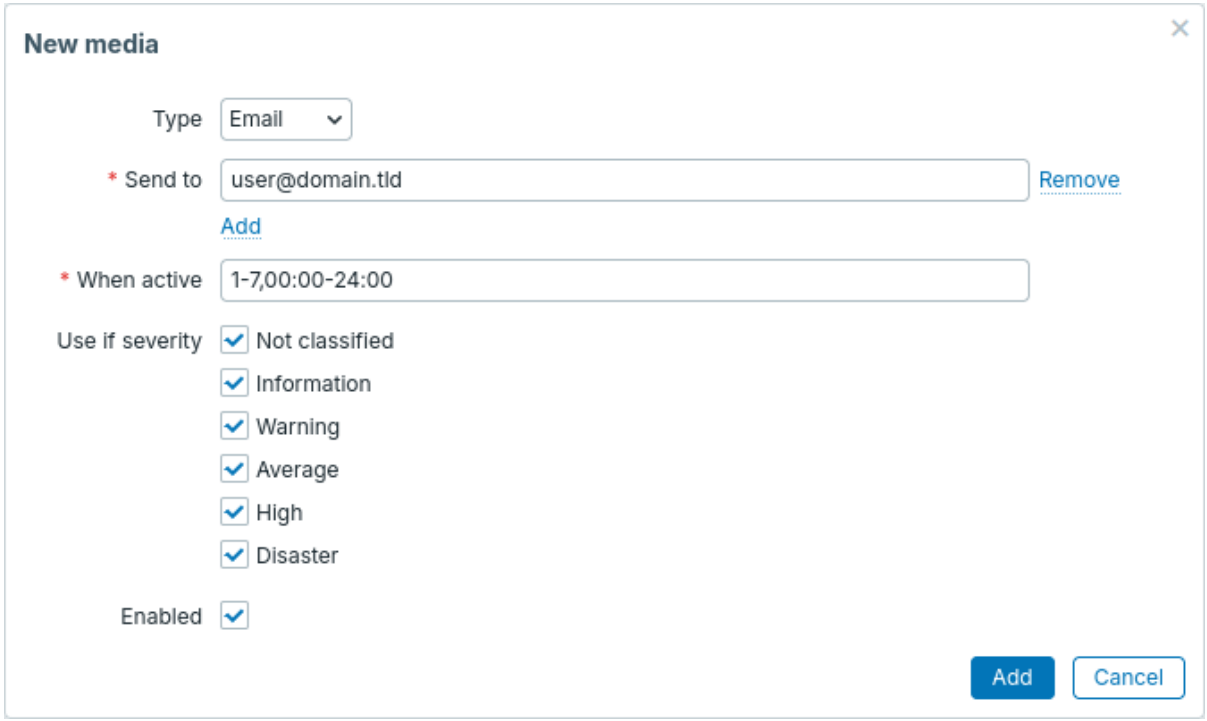
Diese Aktion führt Sie zum Host-Dashboard (auf Vorlagenebene konfiguriert) mit den wichtigsten von der Website erfassten Metriken.



Problemalarme einrichten

Zabbix kann Sie mit verschiedenen Methoden über ein Problem in Ihrer Infrastruktur benachrichtigen. Diese Anleitung enthält grundlegende Konfigurationsschritte zum Senden von E-Mail-Benachrichtigungen.

1. Navigieren Sie zu *Benutzereinstellungen > Profil*, wechseln Sie zur Registerkarte *Medien* und fügen Sie Ihre E-Mail-Adresse hinzu.



2. Folgen Sie der Anleitung zum Empfangen einer Problembenachrichtigung.

Wenn Zabbix das nächste Mal ein Problem erkennt, sollten Sie eine Benachrichtigung per E-Mail erhalten.

Testen Sie Ihre Konfiguration

Um Ihre Konfiguration zu testen, können wir ein echtes Problem simulieren, indem wir die Host-Konfiguration im Zabbix Frontend aktualisieren.

1. Öffnen Sie die Host-Konfiguration Ihrer Website in Zabbix.

2. Wechseln Sie zur Registerkarte *Macros* und wählen Sie *Inherited and host macros* aus.

3. Klicken Sie neben dem zuvor konfigurierten Wert der {$WEBSITE.DOMAIN}-Macro, beispielsweise previously configured, auf *Change* und setzen Sie einen falschen Domainnamen (zum Beispiel /git.zabbix.com/projects/ZBX/repos/zabbix/browse).

4. Klicken Sie auf *Update*, um die Host-Konfiguration zu aktualisieren.

5. Nach einigen Augenblicken erkennt Zabbix das Problem "Failed to get JSON of the requested website", da keine Verbindung zur angegebenen Website hergestellt werden kann. Das Problem wird unter *Monitoring > Problems* angezeigt.



Wenn Benachrichtigungen konfiguriert sind, erhalten Sie außerdem die Problembenachrichtigung.

6. Ändern Sie den Macro-Wert wieder auf den vorherigen Wert zurück, um das Problem zu beheben und die Überwachung der Website fortzusetzen.

Siehe auch

- Ein Datenpunkt erstellen - wie zusätzliche Metriken überwacht werden.
- Problem-Eskalationen - wie mehrstufige Alarmszenarien erstellt werden (z. B. zuerst eine Nachricht an den Systemadministrator senden und dann, wenn ein Problem nach 45 Minuten nicht behoben ist, eine Nachricht an den Rechenzentrumsleiter senden).
- Browser-Datenpunkte - wie Browser-Datenpunkte konfiguriert werden.
- Vorlage Website by Browser - zusätzliche Informationen zur Vorlage *Website by Browser*.

Website-Zertifikate mit Zabbix Agent 2 (passiv) überwachen

Einführung

Dieser Leitfaden bietet einen umfassenden Überblick darüber, wie SSL/TLS-Zertifikate mit dem Schlüssel `web.certificate.get` in Zabbix Agent 2 eingerichtet und überwacht werden. Er wurde entwickelt, um die Zertifikatsüberwachung für einzelne oder mehrere Websites zu vereinfachen, damit Administratoren potenzielle Probleme wie abgelaufene oder ungültige Zertifikate schnell erkennen können.

Für wen dieser Leitfaden gedacht ist

Dieser Leitfaden ist für neue Zabbix-Benutzer gedacht und enthält die minimale Anzahl an Schritten, die erforderlich sind, um die grundlegende Überwachung von Website-Zertifikaten zu aktivieren. Wenn Sie nach umfangreichen Anpassungsoptionen suchen oder eine erweiterte Konfiguration benötigen, lesen Sie den Abschnitt Konfiguration im Zabbix-Handbuch.

Voraussetzungen

Bevor Sie mit dieser Anleitung fortfahren, müssen Sie den Zabbix Server, das Zabbix Frontend und den Zabbix Agent 2 gemäß den Anweisungen für Ihr Betriebssystem herunterladen und installieren. Dieses Tutorial setzt voraus, dass sowohl der Zabbix Server als auch der Agent auf demselben Rechner installiert sind; daher wird in der Konfiguration `127.0.0.1` verwendet.

Zabbix Agent 2 konfigurieren

1. Öffnen Sie die Konfigurationsdatei des Zabbix Agent:

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

2. Setzen Sie den Parameter Server auf 127.0.0.1, da der Agent und der Server auf demselben Computer ausgeführt werden:

```
Server=127.0.0.1
```

3. Speichern Sie die Datei und starten Sie den Dienst des Zabbix Agent 2 neu:

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

4. Testen Sie nach der Einrichtung und Konfiguration des Zabbix Agent 2 dessen Verfügbarkeit mit:

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

Zabbix Agent 2 enthält standardmäßig das WebCertificate-Plugin, sodass keine separate Installation oder Konfiguration erforderlich ist.

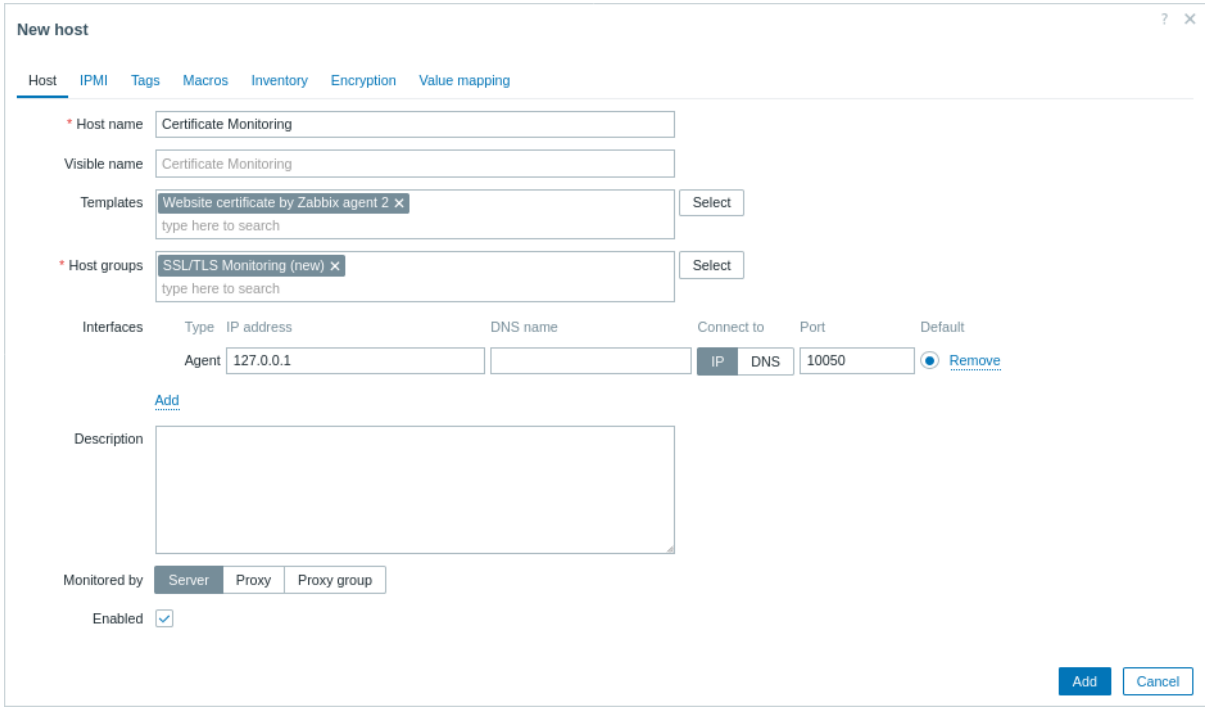
Zabbix Frontend konfigurieren

1. Melden Sie sich im Zabbix Frontend an.

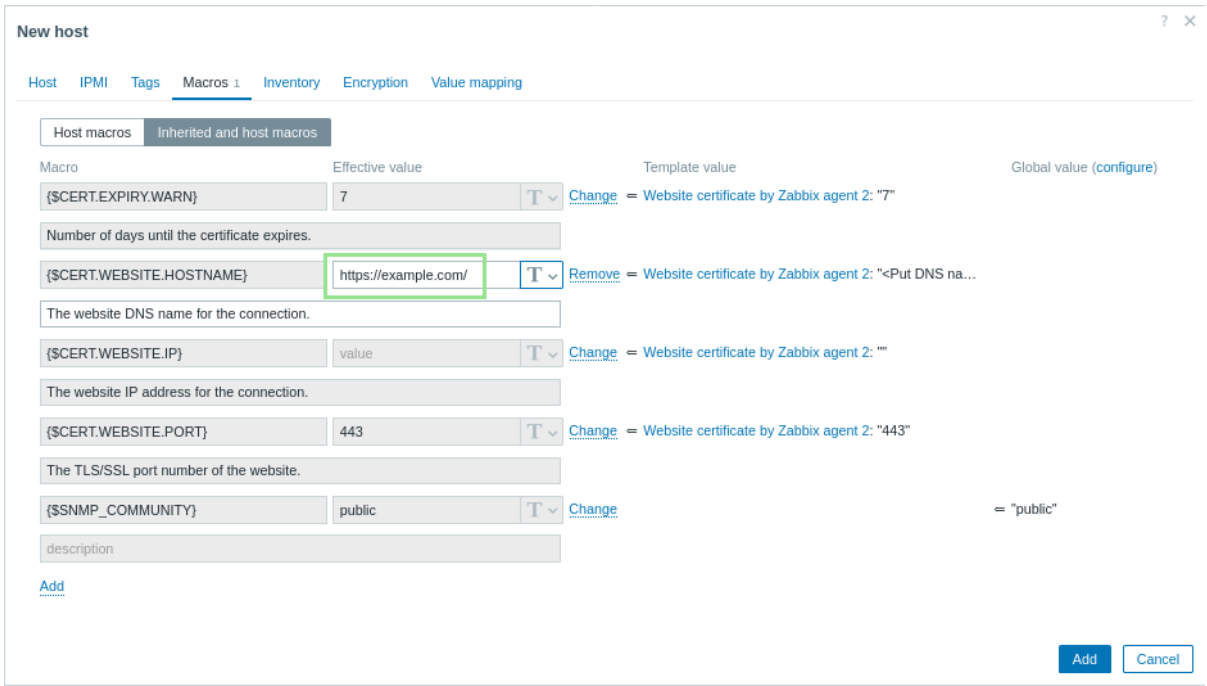
2. Navigieren Sie zu *Monitoring* > *Hosts*.

3. Klicken Sie auf einen vorhandenen Host, für den Sie Website-Zertifikate überwachen möchten, oder erstellen Sie einen Host, falls erforderlich:

- Geben Sie im Feld *Host name* einen Hostnamen ein (z. B. ”Certificate Monitoring”).
- Geben Sie im Feld *Templates* die Vorlage ”Website certificate by Zabbix agent 2” ein oder wählen Sie sie aus, die mit dem Host verknüpft werden soll.
- Geben Sie im Feld *Host groups* eine Hostgruppe ein oder wählen Sie sie aus (z. B. ”SSL/TLS Monitoring”).
- Fügen Sie im Feld *Interfaces* eine Schnittstelle vom Typ ”Agent” hinzu und geben Sie eine IP-Adresse an. In diesem Beispiel wird ”127.0.0.1” verwendet.



- Wechseln Sie im Tab *Macros* zu *Inherited and host macros*, suchen Sie nach den folgenden Makros und klicken Sie neben dem Makrowert auf *Change*, um ihn zu aktualisieren:
  - `{$CERT.WEBSITE.HOSTNAME}` - geben Sie den gewünschten DNS-Namen der Website als Wert ein.



4. Klicken Sie auf *Add*, um den Host hinzuzufügen.

Um mehrere Websites zu überwachen, können Sie im Makro `{$CERT.WEBSITE.HOSTNAME}` eine durch Kommas getrennte Liste ihrer Hostnamen angeben. Optional können Sie auch im Makro `{$CERT.WEBSITE.PORT}` eine durch Kommas getrennte Liste von Ports angeben, wobei jeder Port der jeweiligen Hostname in derselben Reihenfolge entspricht. Um beispielsweise example.com und example.org auf Port 8443 zu überwachen:

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

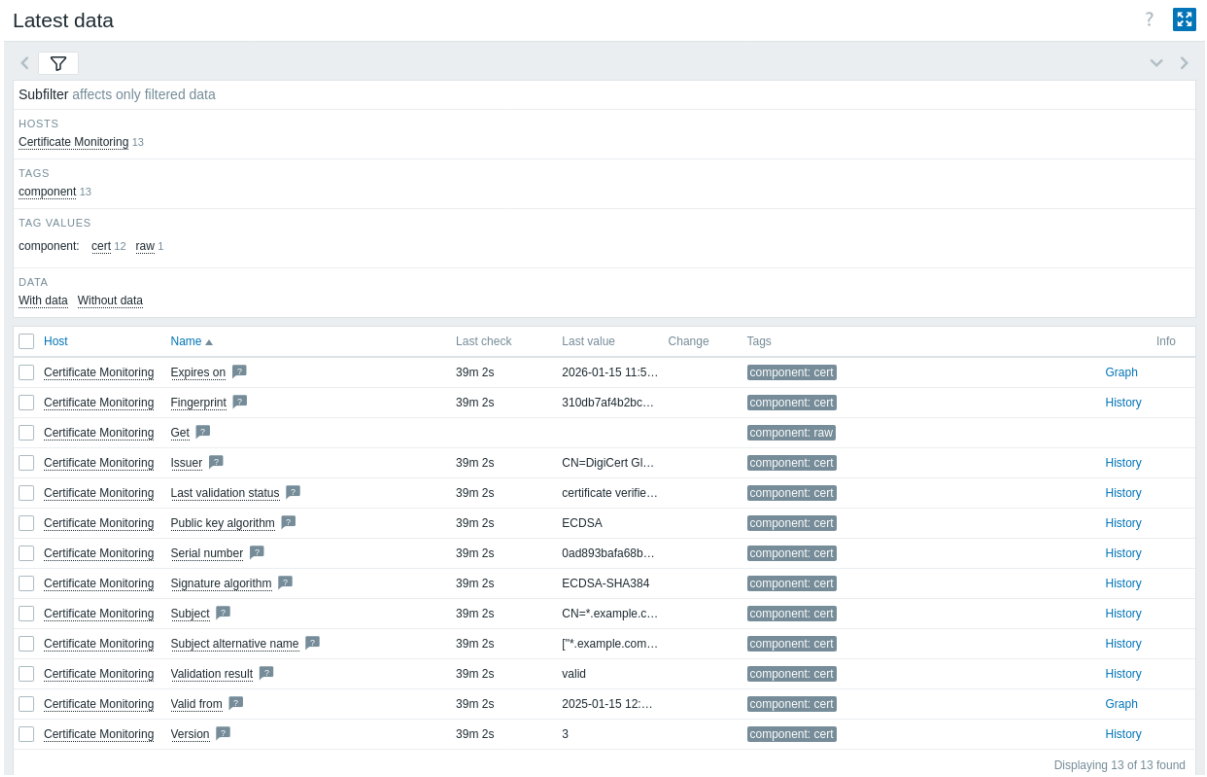
Für jede im Makro `{$CERT.WEBSITE.HOSTNAME}` angegebene Website erstellt Zabbix einen entsprechenden Satz von Datenpunkten und Auslösern. Dadurch ist eine individuelle Überwachung und Alarmierung für jedes SSL-Zertifikat der Website möglich.

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:…

Gesammelte Metriken anzeigen

Glückwunsch! Zu diesem Zeitpunkt überwacht Zabbix bereits das gewünschte Web-Zertifikat.

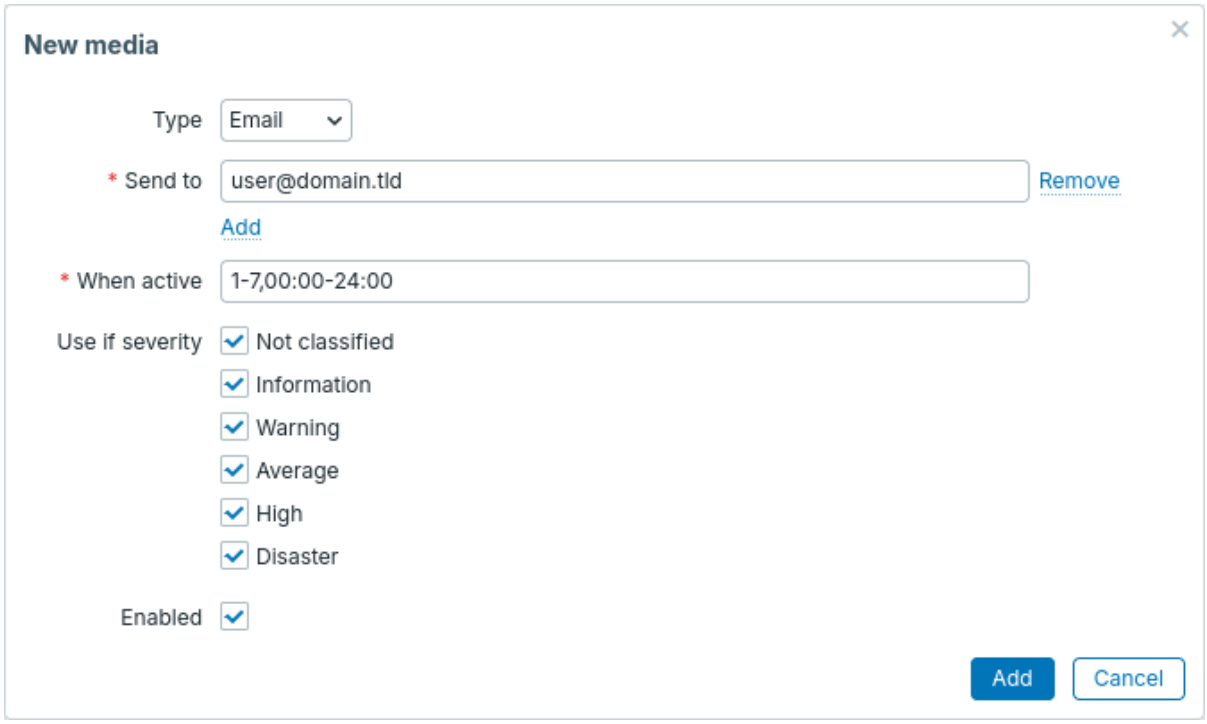
Um die gesammelten Metriken anzuzeigen, navigieren Sie zum Menübereich *Monitoring > Hosts* und klicken Sie neben dem Host auf *Latest data*, um alle zuletzt gesammelten Metriken in einer Liste anzuzeigen, z. B. Ablaufdatum, Aussteller und Betreff.



Problembenachrichtigungen einrichten

Zabbix kann Sie über Probleme in Ihrer Infrastruktur benachrichtigen. Diese Anleitung enthält grundlegende Konfigurationsschritte zum Senden von E-Mail-Benachrichtigungen.

1. Navigieren Sie zu *Benutzereinstellungen > Profil*, wechseln Sie zur Registerkarte *Media* und fügen Sie Ihre E-Mail-Adresse hinzu.



2. Folgen Sie der Anleitung zum Empfangen einer Problembenachrichtigung.

Wenn Zabbix das nächste Mal ein Problem erkennt, sollten Sie eine Benachrichtigung per E-Mail erhalten.

Testen Sie Ihre Konfiguration

Um Ihre Konfiguration zu testen, können wir ein echtes Problem simulieren, indem wir die Host-Konfiguration im Zabbix-Frontend aktualisieren.

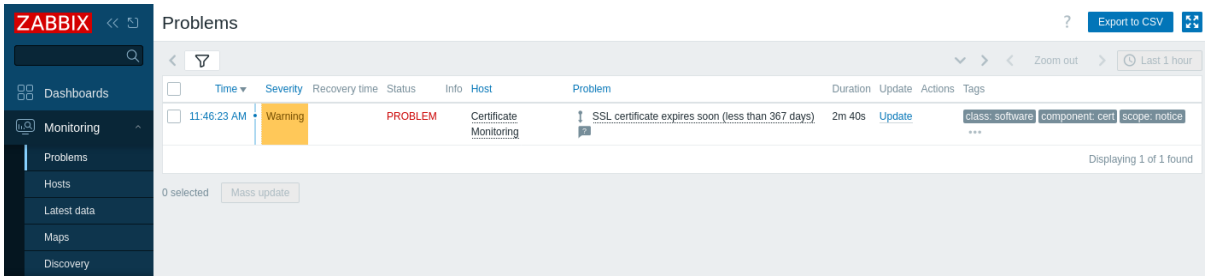
1. Öffnen Sie die Host-Konfiguration Ihres ”Certificate Monitoring”-Hosts in Zabbix.

2. Wechseln Sie zur Registerkarte *Macros* und wählen Sie *Inherited and host macros* aus.

3. Klicken Sie neben dem zuvor konfigurierten Wert des Makros `{$CERT.EXPIRY.WARN}` auf *Change* und setzen Sie eine sehr hohe Anzahl von Tagen (mehr als 365 Tage sollte ausreichen), um eine Warnung zu erhalten, bevor das Zertifikat abläuft.

4. Klicken Sie auf *Update*, um die Host-Konfiguration zu aktualisieren.

5. Nach einigen Augenblicken erkennt Zabbix das Problem "SSL certificate expires soon" mit der Anzahl der Tage bis zum Ablauf. Das Problem wird unter *Monitoring > Problems* angezeigt.



Wenn Benachrichtigungen konfiguriert sind, erhalten Sie außerdem die Problembenachrichtigung.

6. Ändern Sie den Makrowert wieder auf den vorherigen Wert zurück, um das Problem zu beheben und die Überwachung der Zertifikatswerte fortzusetzen.

Siehe auch

- Zabbix Agent 2 - listet Datenpunkt-Schlüssel auf.
- Vorlage *Website certificate by Zabbix agent 2* - zusätzliche Informationen zur Vorlage *Website certificate by Zabbix agent 2*.
- Vorlage *Website certificate by Zabbix agent 2 active* - zusätzliche Informationen zur Vorlage *Website certificate by Zabbix agent 2 active*.
- Websites mit Browser-Datenpunkten überwachen - wie die grundlegende Überwachung von Websites mit Browser-Datenpunkten gestartet wird.

Überwachen Sie einen Netzwerkswitch oder Router mit Zabbix

Einführung

Diese Anleitung führt Sie durch die erforderlichen Schritte, um mit Zabbix eine grundlegende Überwachung Ihres Netzwerk-Switches oder Routers zu starten. Ein Cisco-Router wird als Beispiel verwendet, das Verfahren gilt jedoch für jedes SNMP-fähige Netzwerkgerät.

Für wen diese Anleitung gedacht ist

Diese Anleitung richtet sich an neue Zabbix-Benutzer und Netzwerkadministratoren, die die grundlegende Überwachung von Netzwerkgeräten schnell aktivieren möchten. Wenn Sie eine tiefgehende Anpassung oder erweiterte Konfigurationsoptionen benötigen, lesen Sie bitte die Seite SNMP agent oder den Abschnitt Konfiguration des Zabbix-Handbuchs.

Voraussetzungen

Bevor Sie mit dieser Anleitung fortfahren, stellen Sie sicher, dass Sie Folgendes haben:

- Zabbix Server und Zabbix Frontend installiert: Installieren Sie diese gemäß den Anweisungen für Ihr Betriebssystem (siehe Installation from packages und Web interface installation).
- Zabbix Agent installiert, wenn lokale Netzwerkmetriken überwacht werden sollen.
- SNMP-fähiges Gerät: ein Netzwerk-Switch oder Router (zum Beispiel ein Cisco-Router) mit aktiviertem SNMP.
- MIB files installiert: Durch die Installation von MIB-Dateien kann Zabbix numerische OIDs in lesbare Namen und Beschreibungen übersetzen. Ohne geeignete MIB-Unterstützung sehen Sie möglicherweise nur numerische Werte, was die Konfiguration von Datenpunkten und die Fehlerbehebung erschwert.

So installieren Sie MIB-Dateien unter Ubuntu:

1. Installieren Sie das MIB-Downloader-Paket:

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

Wenn Sie herstellerspezifische MIBs hinzufügen müssen (z. B. von Cisco oder Juniper), legen Sie diese im entsprechenden MIB-Verzeichnis ab:

- Bei Linux-basierten Systemen gehören zu den üblichen Speicherorten `/usr/share/snmp/mibs/` oder `/usr/local/share/snmp/mi`
- Bei Zabbix-Installationen können MIB-Dateien in `/var/lib/zabbix/mibs/` gespeichert werden.

Stellen Sie sicher, dass die Umgebungsvariable `MIBDIRS` oder die Datei `snmp.conf` den richtigen Pfad enthält.

Um zu überprüfen, ob Ihr System die neuen MIBs erkennt, verwenden Sie:

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

Detaillierte Anweisungen finden Sie in der Dokumentation Ihrer SNMP-Bibliothek:

- Cisco MIBs
- Juniper MIBs

2. Bearbeiten Sie `/etc/snmp/snmp.conf` und kommentieren Sie die Zeile aus, die mit `mibs :` beginnt, damit das System alle verfügbaren MIBs laden kann.

3. Überprüfen Sie dies mit einem `snmpwalk` (zum Beispiel `snmpwalk -v 2c -c <your_community_string> <device_IP>`) und stellen Sie sicher, dass OIDs mit aussagekräftigen Namen angezeigt werden.

Je nach Ihrer Umgebung können sich einige Schritte in dieser Anleitung leicht unterscheiden. Diese Anleitung basiert auf einer Umgebung mit Ubuntu und einem überwachten Netzwerkgerät vom Typ Cisco Catalyst 3750V2-24FS.

Es wird davon ausgegangen, dass Ihr Netzwerkgerät bereits physisch installiert und verbunden ist.

Konfigurieren Sie das Netzwerkgerät (Cisco-Router-Beispiel)

Für die Überwachung über SNMP müssen Sie Ihr Netzwerkgerät so konfigurieren, dass SNMP-Abfragen zugelassen werden. Das folgende Beispiel gilt für SNMPv2 und berücksichtigt keine vorhandenen Einstellungen. Achtung: Das Anwenden dieser Befehle kann die aktuellen SNMP-Konfigurationen überschreiben.

Für einen Cisco-Router umfasst die Konfiguration in der Regel die unten beschriebenen Schritte.

SNMPv2-Beispiel

1. Aktivieren Sie SNMP und legen Sie die Community-String fest.

Melden Sie sich an an der Konsole Ihres Cisco-Routers an und wechseln Sie in den Konfigurationsmodus:

```
configure terminal
```

Aktivieren Sie dann SNMP, indem Sie einen Community-String mit Nur-Lesezugriff angeben. Zum Beispiel:

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

Ersetzen Sie `<your_community_string>` durch Ihren sicheren Community-String. Hinweis: Die Option RO (Read-Only) ermöglicht SNMP das Abrufen von Daten vom Gerät, verhindert jedoch jegliche Konfigurationsänderungen.

Aus Sicherheitsgründen wird empfohlen, den SNMP-Zugriff auf nur die erforderlichen Geräte zu beschränken. Weitere Hinweise zur Konfiguration von Zugriffskontrolllisten (ACLs) finden Sie in der offiziellen Cisco-Dokumentation.

2. Speichern Sie die Konfiguration.

Speichern Sie Ihre Änderungen, damit die SNMP-Einstellungen nach einem Neustart erhalten bleiben:

```
write memory
```

SNMPv3-Beispiel

SNMPv3 bietet erweiterte Sicherheit durch Authentifizierung und Verschlüsselung. Die Konfiguration ist sicherer als bei SNMPv2 und sollte anhand der gerätespezifischen Dokumentation überprüft werden.

1. Erstellen Sie eine SNMP-Gruppe.

Konfigurieren Sie eine SNMPv3-Gruppe mit aktivierter Privatsphäre (Verschlüsselung):

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

2. Erstellen Sie einen SNMP-Benutzer.

Fügen Sie einen SNMPv3-Benutzer mit Authentifizierung und Privatsphäre hinzu. Ersetzen Sie die Platzhalter durch die gewünschten Werte:

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

3. Speichern Sie die Konfiguration:

```
write memory
```

Weitere Details oder modellspezifische Anweisungen finden Sie in externen Cisco SNMP configuration tutorials. Diese Anleitung bietet jedoch die grundlegenden Schritte zum Aktivieren der SNMP-Überwachung.

Zabbix Frontend konfigurieren

Einen Host im Zabbix Frontend erstellen

1. Melden Sie sich im Zabbix Frontend an.

2. Fügen Sie einen neuen Host hinzu.

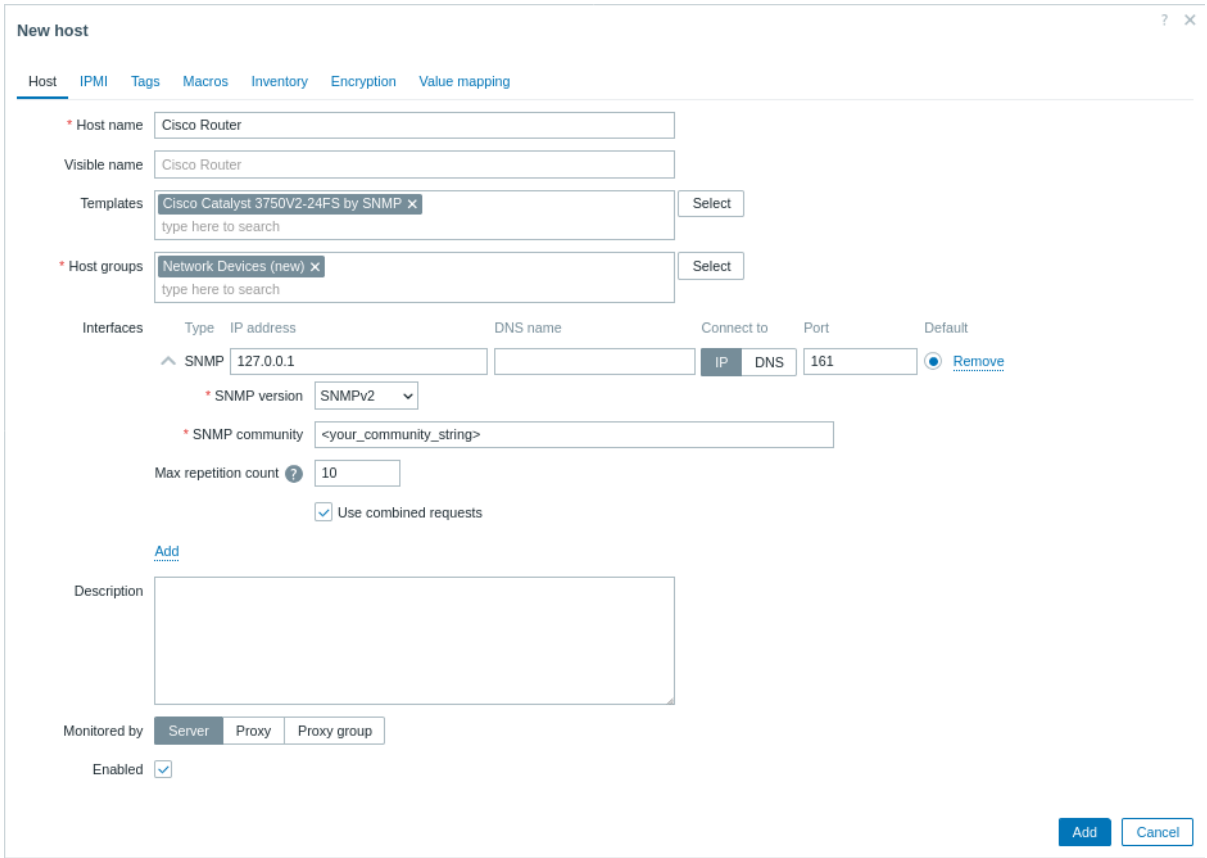
Navigieren Sie zu *Datenerfassung* > *Hosts* und klicken Sie auf *Host erstellen*.

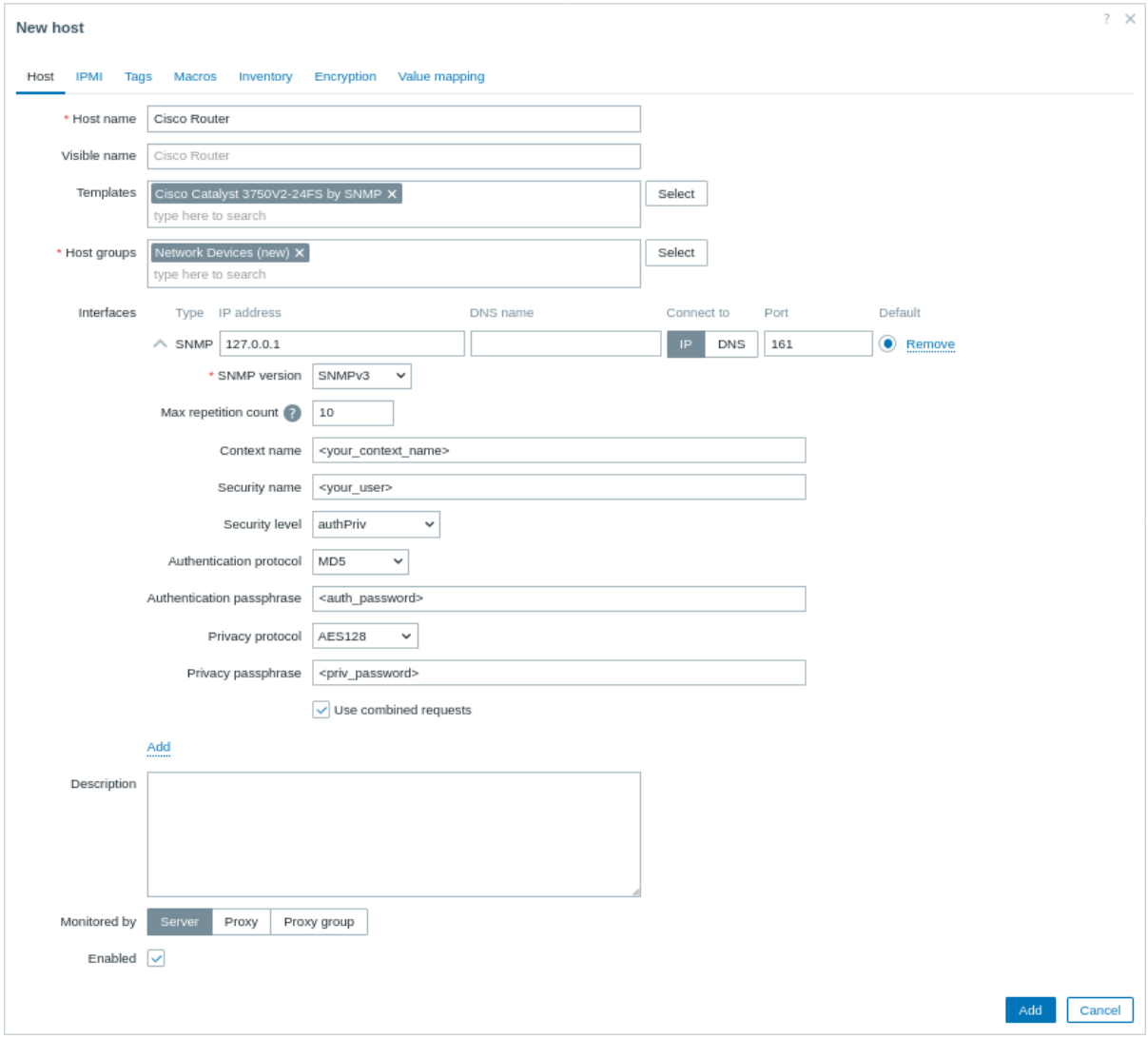
- *Host name*: Geben Sie einen Namen für Ihr Gerät ein (z. B. ”Cisco Router”).
- Host groups: Wählen Sie eine vorhandene Gruppe aus oder erstellen Sie eine neue Gruppe wie ”Network Devices”.
- Interfaces:
  - Klicken Sie unter Interfaces auf *Add*.
  - Wählen Sie *SNMP* als Schnittstellentyp aus.
  - Geben Sie die IP-Adresse oder den DNS-Namen Ihres Cisco-Routers ein.
  - Setzen Sie den standardmäßigen SNMP-Port (in der Regel 161).
  - Verwenden Sie das Dropdown-Menü, um die passende SNMP-Version auszuwählen (z. B. SNMPv2).
  - Für SNMPv1/v2 geben Sie die Community-String im Feld *SNMP community* ein. Für SNMPv3 werden zusätzliche Anmeldedaten (*Context name*, *Security name* und *Security level* usw.) abgefragt.

3. Vorlagen verknüpfen

Wählen Sie im Feld *Templates* die SNMP-Vorlage aus, die am besten zu Ihrem Gerät passt. Zabbix stellt eine Reihe vorgefertigter SNMP-Vorlagen für viele Gerätegruppen bereit. Wenn Sie beispielsweise ein Cisco-Gerät überwachen, wählen Sie die Vorlage, die dem Betriebssystem oder Modell Ihres Geräts entspricht (z. B. Cisco IOS SNMP oder Cisco Catalyst 3750<device model> SNMP).

4. Klicken Sie auf *Add*, um den Host zu speichern.



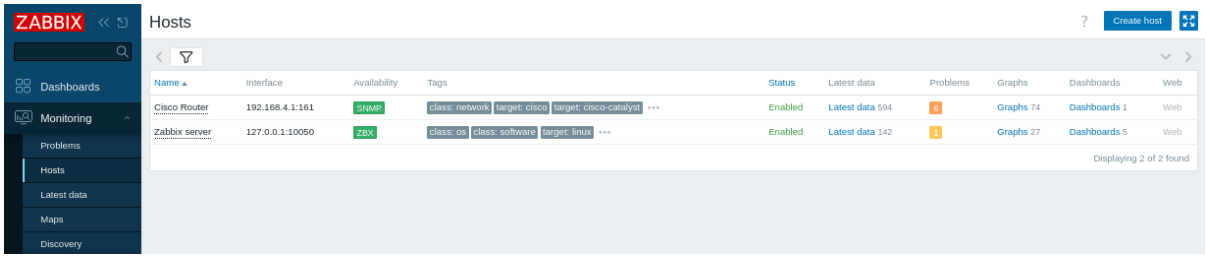


Erfasste Metriken anzeigen

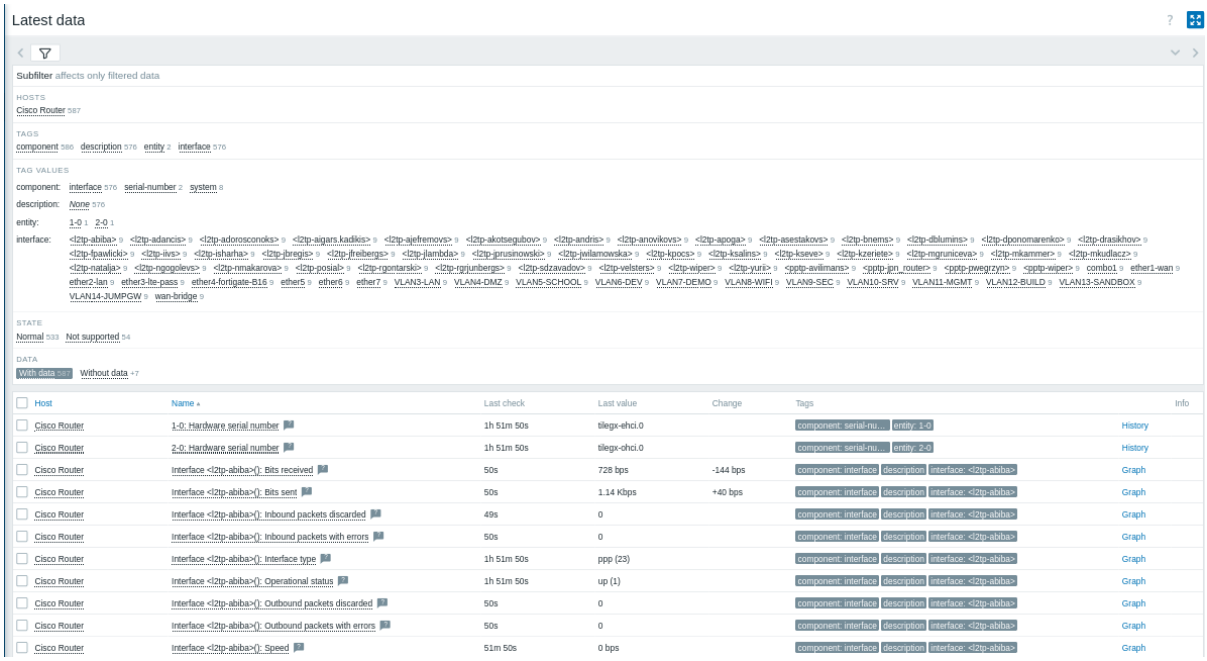
Glückwunsch! Zabbix ist jetzt so eingerichtet, dass es Ihr Netzwerkgerät überwacht.

Neueste Daten:

- Navigieren Sie im Zabbix Frontend zu Überwachung > Neueste Daten.



- Wählen Sie Ihren Host ”Cisco Router” (oder erkannte Hosts) aus, um Metriken wie Hardware- und Netzwerk-Uptime, ICMP-Verlust, Ping und Antwortzeit usw. anzuzeigen.



- Diagramme und Bildschirme:

Um die Leistungsdaten zu visualisieren, klicken Sie neben den SNMP-Datenpunkten auf *Diagramme*, um detaillierte Metriken anzuzeigen.

Als nächsten Schritt können Sie:

- Benutzerdefinierte SNMP-Datenpunkte hinzufügen, um zusätzliche Metriken zu überwachen.
- Problemalarme einrichten, um Benachrichtigungen über mögliche Probleme zu erhalten.

SNMP-Datenpunkte erstellen

Sobald der Host eingerichtet ist, können Sie Datenpunkte erstellen, um bestimmte Metriken zu überwachen. Hinweis: Dieser Schritt ist optional, wenn Sie eine Vorlage verwenden, da Vorlagen bereits Standard-Datenpunkt-Sätze enthalten.

1. Identifizieren Sie die SNMP OID:

Verwenden Sie den Befehl `snmpwalk`, um die verfügbaren OIDs auf Ihrem Gerät aufzulisten. Zum Beispiel:

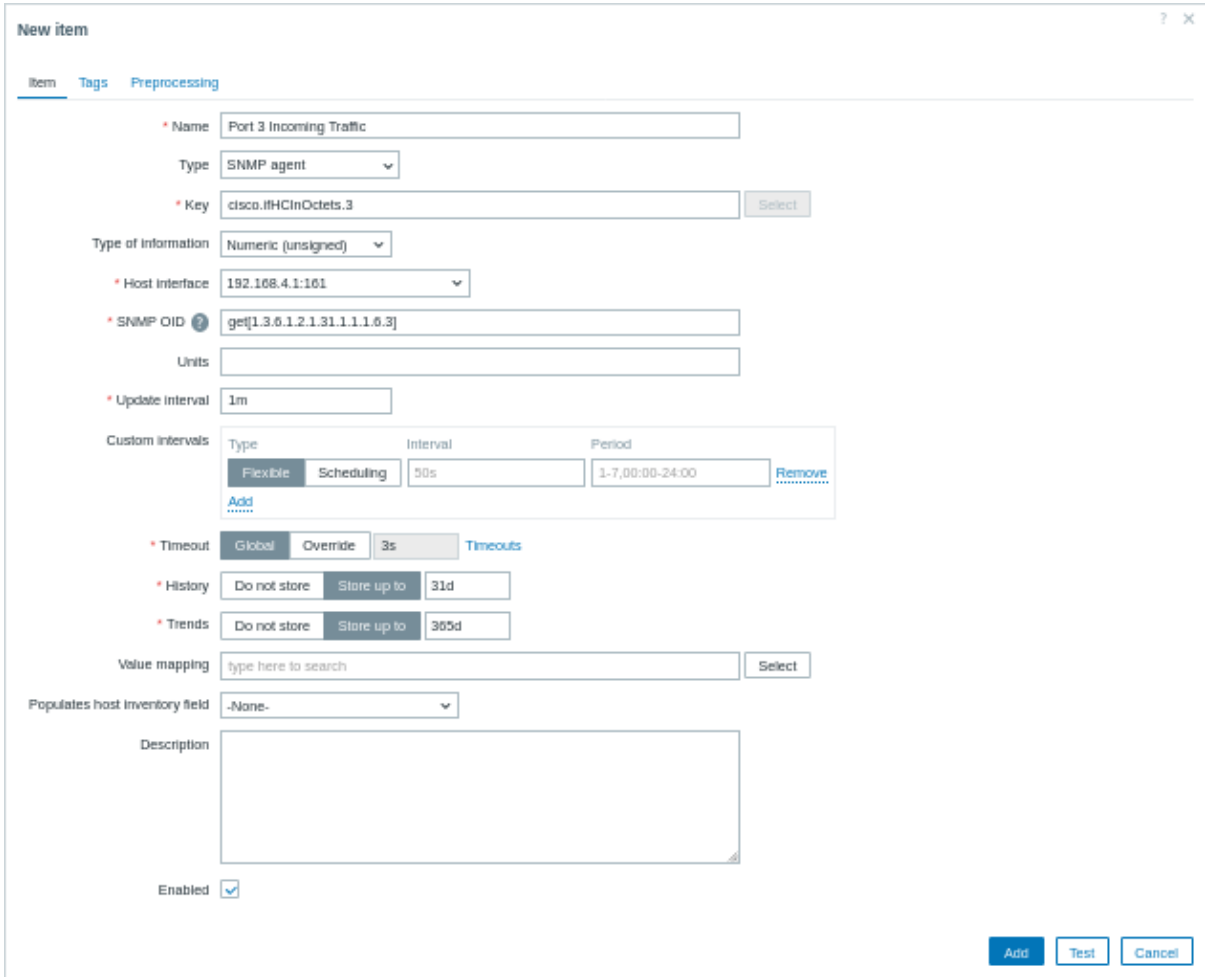
```
snmpwalk -v 2c -c <your_community_string> <device_IP> .
```

Suchen Sie die OID für die Metrik, die Sie überwachen möchten (z. B. IF-MIB::ifHCInOctets.3 für eingehenden Traffic auf Port 3). Um die numerische OID zu erhalten, können Sie Folgendes verwenden:

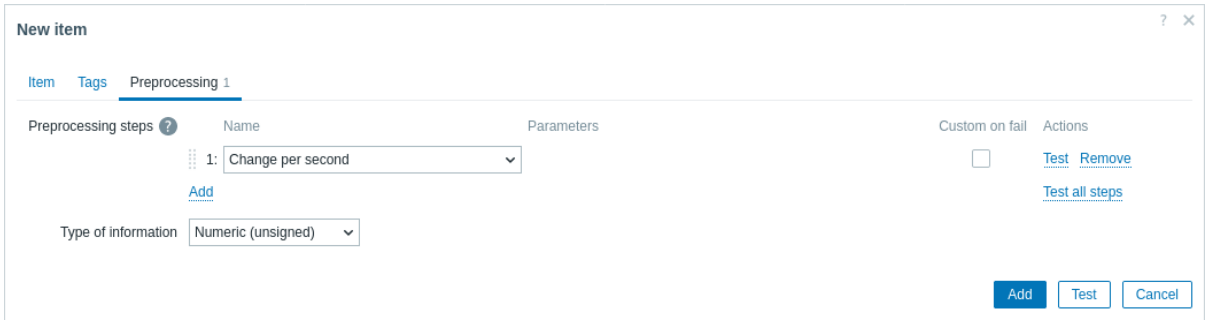
```
snmpget -v 2c -c <your_community_string> -On <device_IP> IF-MIB::ifHCInOctets.3
```

2. Erstellen Sie einen SNMP-Datenpunkt:

- Navigieren Sie zu *Datenerfassung* > *Hosts* und klicken Sie auf die Registerkarte *Datenpunkte* für Ihren SNMP-Host und dann auf *Datenpunkt erstellen*.
- *Name*: Geben Sie einen aussagekräftigen Namen ein (z. B. "Port 3 eingehender Traffic").
- *Typ*: Wählen Sie *SNMP agent*.
- *Key*: Geben Sie einen aussagekräftigen Key an (z. B. `cisco.ifHCInOctets.3`).
- *Host interface*: Stellen Sie sicher, dass die SNMP-Schnittstelle ausgewählt ist.
- *SNMP OID*: Geben Sie die OID in einem der unterstützten Formate ein, zum Beispiel:
  - `get[1.3.6.1.2.1.31.1.1.1.6.3]` für einen einzelnen Wert;
  - `walk[1.3.6.1.2.1.31.1.1.1.6.3]` um einen Teilbaum von Werten asynchron abzurufen.



- *Preprocessing* (falls erforderlich): Wenn der Datenpunkt einen kumulativen Zähler zurückgibt (z. B. Schnittstellen-Traffic), navigieren Sie zur Registerkarte *Preprocessing* und fügen Sie einen Preprocessing-Schritt wie "Change per second" hinzu, um die Rate zu berechnen.



Um mehrere Werte in einer SNMP-Transaktion abzurufen, können Sie mehrere OIDs mit der Syntax `walk[OID1,OID2,...]` angeben.

Übersetzen von OIDs zwischen numerischen und MIB-Namen

Wenn Sie mit SNMP arbeiten, müssen Sie möglicherweise zwischen numerischen OIDs und den entsprechenden MIB-Namen konvertieren. Diese Übersetzung hilft dabei, Metriken leichter zu identifizieren und Fehler zu beheben.

- Übersetzen eines MIB-Namens in eine numerische OID: Verwenden Sie den Befehl `snmptranslate` mit der Option `-On`. Um beispielsweise den MIB-Namen `IF-MIB::ifHCInOctets.3` in seine numerische OID zu übersetzen, führen Sie Folgendes aus:

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

Dieser Befehl könnte Folgendes ausgeben:

```
.1.3.6.1.2.1.31.1.1.1.6.3
```

- Übersetzen einer numerischen OID in ihren MIB-Namen: Verwenden Sie den Befehl `snmptranslate` mit der Option `-IR` (oder `-m ALL`), um die Übersetzung umzukehren. Um beispielsweise die numerische OID `.1.3.6.1.2.1.31.1.1.1.6.3` wieder in ihren MIB-Namen zu übersetzen, führen Sie Folgendes aus:

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

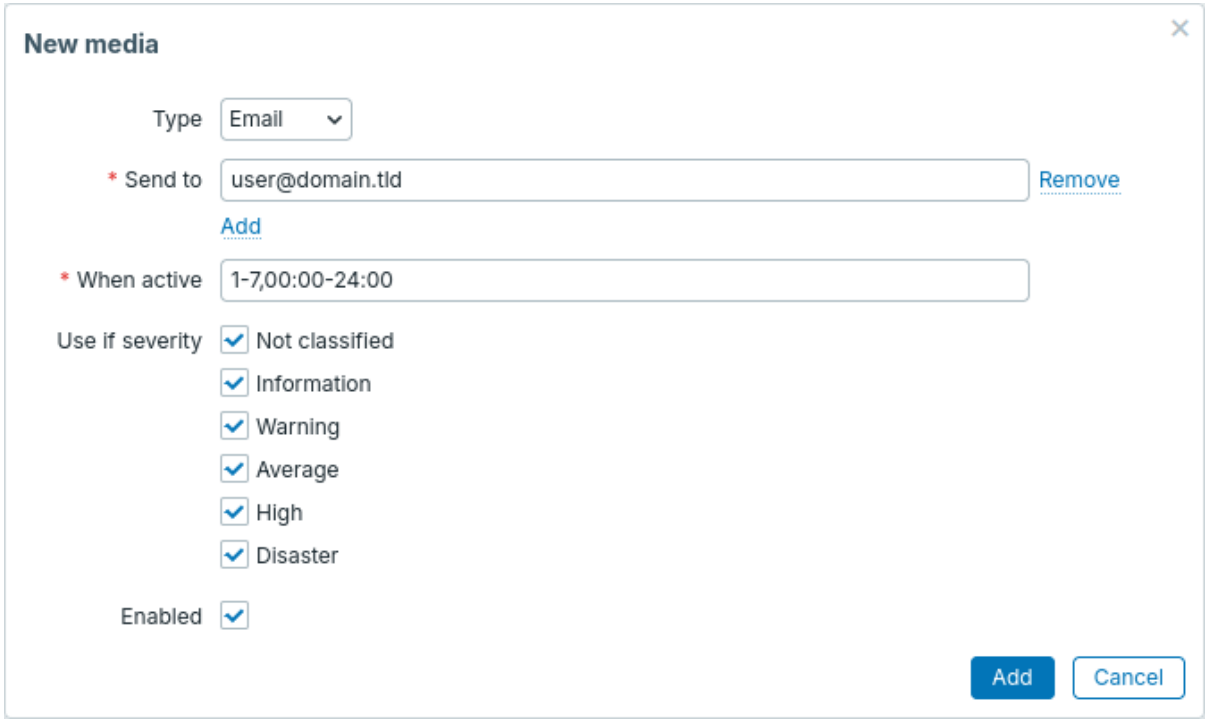
Dieser Befehl könnte Folgendes ausgeben:

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

Problemalarme einrichten

Diese Anleitung enthält grundlegende Konfigurationsschritte zum Senden von E-Mail-Benachrichtigungen.

1. Navigieren Sie zu *Benutzereinstellungen > Profil*, wechseln Sie zur Registerkarte *Medien* und fügen Sie Ihre E-Mail-Adresse hinzu.



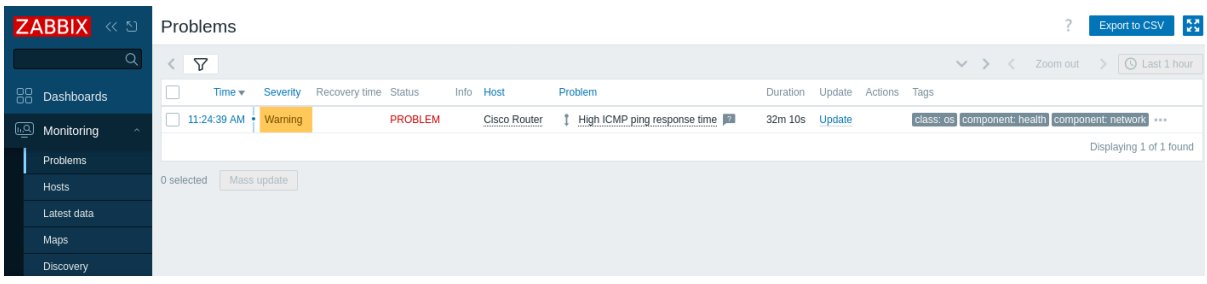
2. Folgen Sie der Anleitung zum Empfangen einer Problembenachrichtigung.

Wenn Zabbix das nächste Mal ein Problem erkennt, sollten Sie eine Benachrichtigung per E-Mail erhalten.

Testen Sie Ihre Konfiguration

Um sicherzustellen, dass Zabbix Netzwerkleistungsprobleme korrekt erkennt, simulieren Sie ein echtes Problem, indem Sie den Schwellenwert für die ICMP-Ping-Antwortzeit erhöhen.

1. Öffnen Sie in Zabbix die Host-Konfiguration Ihres ”Cisco Router”.
2. Navigieren Sie zum Tab *Macros* und wählen Sie *Inherited and host macros* aus.
3. Suchen Sie das Makro `{$ICMP_RESPONSE_TIME_WARN}` (oder ein ähnliches Makro für den Schwellenwert der Antwortzeit).
4. Setzen Sie einen sehr niedrigen Wert (z. B. 0.001), um einen Alarm auszulösen, wenn die Ping-Antwort diesen Wert überschreitet.
5. Klicken Sie auf *Update*, um die Änderungen zu übernehmen.
6. Warten Sie einen Moment, bis Zabbix das simulierte Problem erkennt.
7. Navigieren Sie zu *Monitoring > Problems*, um zu überprüfen, ob ein Alarm angezeigt wird (z. B. ”Hohe ICMP-Ping-Antwortzeit”).



Wenn Alarme konfiguriert sind, sollten Sie außerdem einen Problemalarm erhalten.

8. Setzen Sie den Makro-Wert wieder auf die ursprüngliche Einstellung zurück und klicken Sie auf *Update*, um die Änderungen zu speichern.

9. Bestätigen Sie, dass das Problem behoben ist und aus dem Abschnitt *Problems* verschwindet.

Fehlerbehebung bei SNMP-Monitoring

Wenn Sie feststellen, dass das SNMP-Symbol im Zabbix Frontend ROT angezeigt wird oder keine Daten erfasst werden, versuchen Sie die folgenden Schritte:

1. Prüfen Sie die SNMP-Konnektivität.

Für SNMPv2 führen Sie den folgenden Befehl von Ihrem Zabbix Server aus:

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

Dieser Befehl überprüft, ob das Gerät auf SNMP-Abfragen reagiert.

Für SNMPv3 fügen Sie die entsprechenden SNMPv3-Zugangsdaten hinzu:

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

Damit wird geprüft, ob die SNMPv3-Zugangsdaten korrekt sind und das Gerät sicher antwortet.

2. Stellen Sie sicher, dass MIB-Dateien wie in den Voraussetzungen beschrieben installiert und aktiviert sind. Dazu darf der folgende Befehl beim Abfragen eines Netzwerkgeräts keinen Fehler ausgeben:

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

Dies sollte übersetzte OIDs ohne Fehler zurückgeben.

3. Bestätigen Sie, dass die in Zabbix konfigurierte SNMP-Version und die Zugangsdaten mit denen auf Ihrem Gerät übereinstimmen. Überprüfen Sie beispielsweise die SNMP-Einstellungen in der Zabbix Host-Konfiguration und vergleichen Sie sie mit der Konfiguration Ihres Geräts. Auf einem Cisco-Gerät können Sie die SNMP-Einstellungen mit folgendem Befehl prüfen:

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

Dadurch wird sichergestellt, dass die Community-String-Angabe (für SNMPv2) oder die SNMPv3-Benutzerdetails korrekt sind.

4. Vergewissern Sie sich, dass SNMP auf Ihrem Netzwerkgerät korrekt aktiviert ist. Melden Sie sich auf einem Cisco-Router an der Konsole an und führen Sie Folgendes aus:

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

Dieser Befehl zeigt die aktive SNMP-Konfiguration an und hilft zu bestätigen, dass SNMP ordnungsgemäß konfiguriert ist.

5. Stellen Sie sicher, dass keine Firewalls oder Netzwerkprobleme den SNMP-Datenverkehr zwischen dem Zabbix Server und dem Gerät blockieren (typischerweise auf Port 161). Sie können die Konnektivität mit folgendem Befehl testen:

nc -zv <device_IP> 161

`nc -zv` prüft, ob Port 161 auf dem Gerät geöffnet ist und lauscht.

Wenn Sie zusätzlich UFW unter Ubuntu verwenden, prüfen Sie den Firewall-Status:

```
sudo ufw status
```

Oder bei iptables:

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

6. Prüfen Sie die Protokolldateien des Zabbix Servers auf SNMP-bezogene Fehler, um die Ursache einzugrenzen:

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

`tail -f` ermöglicht es Ihnen, Protokollaktualisierungen in Echtzeit zu überwachen.

Siehe auch:

- Erstellen eines Datenpunkts - erfahren Sie, wie Sie zusätzliche Metriken hinzufügen.
- SNMP Agent - zusätzliche Informationen zur SNMP-Überwachung mit Zabbix.
- Standardisierte Vorlagen für Netzwerkgeräte - Informationen zu verfügbaren SNMP-Vorlagen.
- Erkennung von SNMP OIDs - zusätzliche Informationen zur SNMP-Erkennung auf einem Switch.
- Konfigurieren einer Netzwerkerkennungsregel - zusätzliche Informationen dazu, wie eine von Zabbix verwendete Netzwerkerkennungsregel zum Erkennen von Hosts und Diensten konfiguriert wird.

Windows-Ereignisprotokoll mit aktiven Prüfungen überwachen

Einführung

Dieser Leitfaden erklärt, wie Sie Windows-Ereignisprotokolle mit Zabbix mithilfe aktiver Prüfungen überwachen. Mit den Windows-spezifischen Datenpunkt-Schlüsseln von Zabbix können Sie kritische Ereignisse (z. B. fehlgeschlagene Anmeldeversuche, Systemfehler usw.) in Echtzeit erfassen und analysieren.

Für wen dieser Leitfaden gedacht ist

Dieser Leitfaden richtet sich an neue Zabbix-Benutzer und Netzwerkadministratoren, die Windows-Ereignisprotokolle überwachen möchten. Informationen zu erweiterten Konfigurationsoptionen finden Sie in der Dokumentation zu den Windows-spezifischen Datenpunkt-Schlüsseln.

Voraussetzungen

Bevor Sie mit diesem Leitfaden fortfahren, müssen Sie den Zabbix Server und das Zabbix Frontend gemäß den Anweisungen für Ihr Betriebssystem herunterladen und installieren. Außerdem muss der Zabbix Agent auf dem Windows-Rechner, den Sie überwachen möchten, heruntergeladen und installiert sein.

Zabbix Agent für die Überwachung von Windows-Ereignisprotokollen konfigurieren

1. Öffnen Sie die Konfigurationsdatei des Zabbix Agent auf Ihrem Windows-Host und stellen Sie sicher, dass der Parameter `ServerActive` auf die IP-Adresse Ihres Zabbix-Servers gesetzt ist und der Parameter `Hostname` mit dem Hostnamen übereinstimmt, der im Zabbix Frontend definiert wird. Dadurch kann der Agent aktive Prüfungen für seinen Host und vom angegebenen Zabbix-Server anfordern. Beispiel:

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

2. Starten Sie den Zabbix-Agent-Dienst neu, um die Änderungen anzuwenden:

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

3. Überprüfen Sie, ob der Windows-Host ausgeführt wird:

- Stellen Sie sicher, dass der Zabbix-Agent-Dienst auf dem Windows-Host ausgeführt wird.
- Überprüfen Sie, ob der Windows-Host eine Verbindung zum Zabbix-Server über Port 10051 herstellen kann. Um die Verbindung vom Windows-Host zu testen, öffnen Sie PowerShell und führen Sie den folgenden Befehl aus:

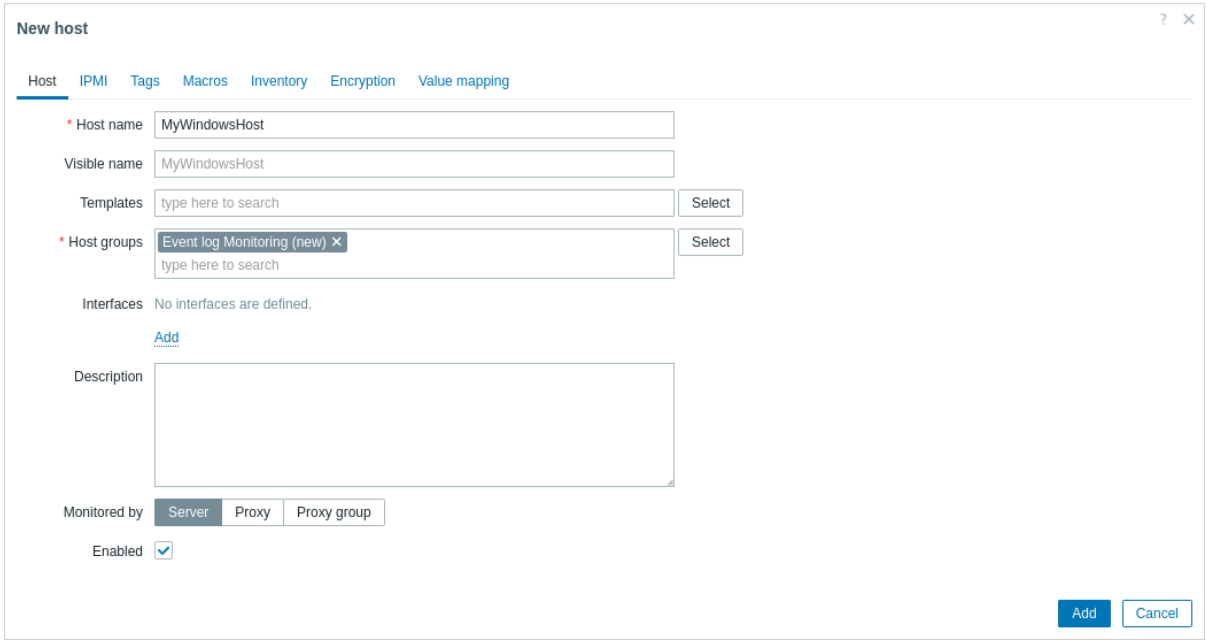
```
Test-NetConnection -ComputerName <Zabbix-server-IP> -Port 10051
```

Zabbix Frontend konfigurieren

1. Navigieren Sie zu *Data collection* > *Hosts* und erstellen Sie einen Host:

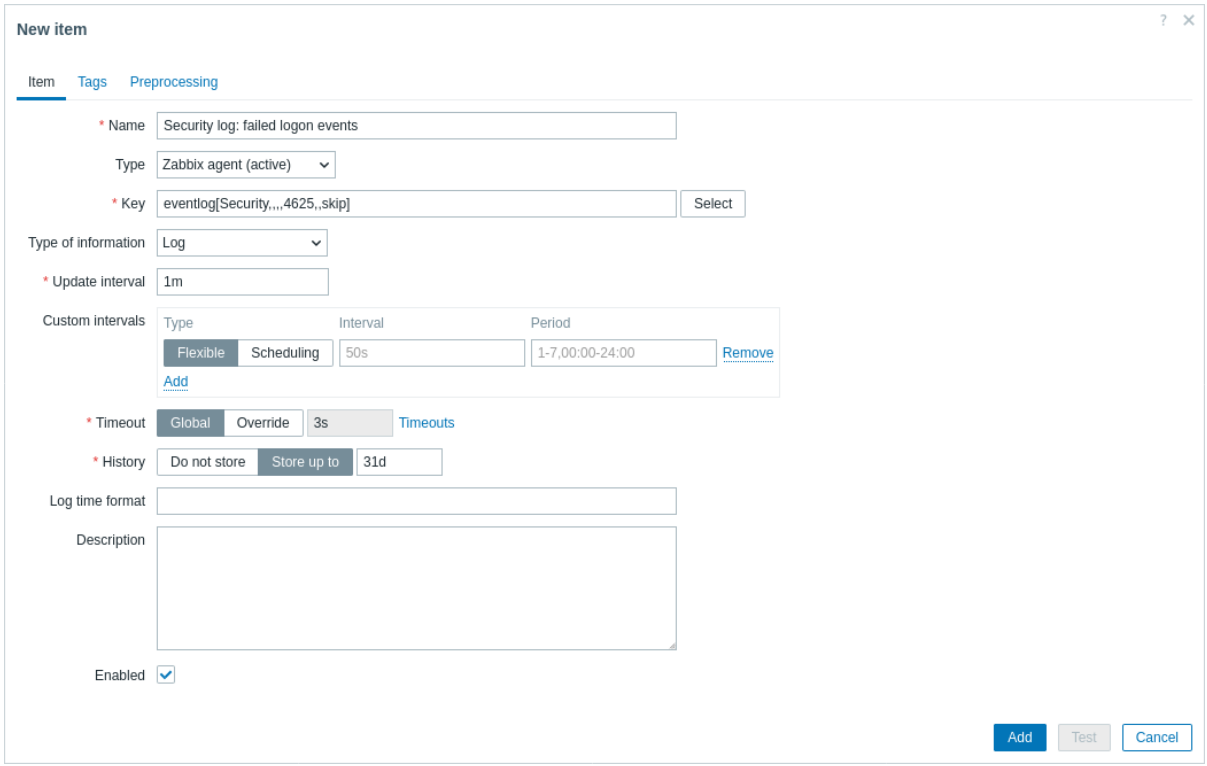
- Geben Sie im Feld *Host name* einen Hostnamen ein (z. B. ”MyWindowsHost”).
- Geben Sie im Feld *Host groups* eine Hostgruppe ein oder wählen Sie sie aus (z. B. ”Event log Monitoring”).
- Klicken Sie auf *Add*, um den konfigurierten Host zu speichern.

> **Note:**
> Im Feld *Templates* können Sie die Vorlage ”Windows by Zabbix agent active” hinzufügen, um die Fehlersuche zu erleichtern, indem Sie beobachten, ob andere aktive Datenpunkte auf demselben Host aktualisiert werden.



2. Erstellen Sie einen neuen Datenpunkt mit den folgenden Parametern:

- Geben Sie im Feld *Name* einen aussagekräftigen Namen für den Datenpunkt ein (z. B. "Security log: failed logon events").
- Wählen Sie in der Dropdown-Liste *Type* "Zabbix agent (active)" aus (erforderlich für die Überwachung von Ereignisprotokollen).
- Verwenden Sie im Feld *Key* den eventlog-Datenpunktschlüssel. Um beispielsweise fehlgeschlagene Anmeldeversuche (Event ID: 4625) im Security-Protokoll zu überwachen und Einträge zu ignorieren, die älter sind als die letzte Prüfung des Datenpunkts (unter Verwendung des Parameters `skip`), geben Sie den folgenden Datenpunktschlüssel ein: `eventlog[Security,,,,4625,,skip]`
- Wählen Sie in der Dropdown-Liste *Type of information* "Log" aus.



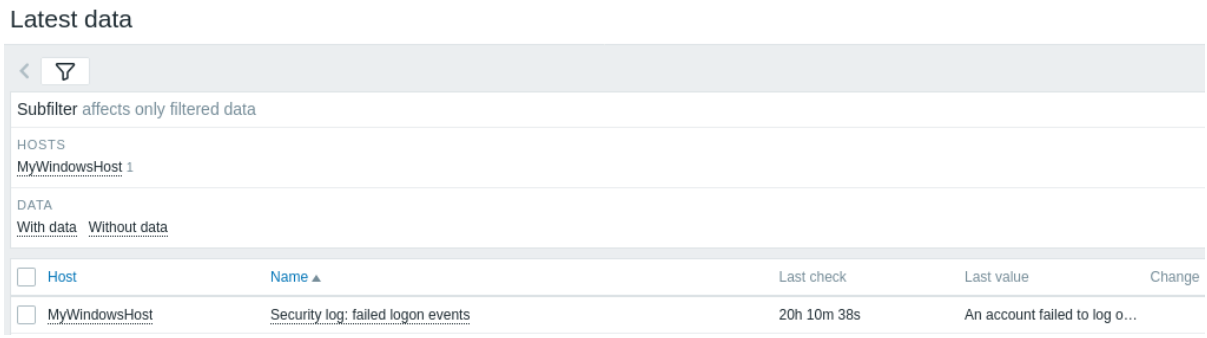
3. Klicken Sie auf *Add*, um den Datenpunkt zu speichern.

Testen und gesammelte Metriken anzeigen

Glückwunsch! Zabbix ist jetzt so eingerichtet, dass es Ihre Windows-Ereignisprotokolle sammelt. Um zu überprüfen, ob Ereignisprotokolle gesammelt werden, können Sie den Datenpunkt "Security log: failed logon events" testen, indem Sie sich von Ihrem Windows-Konto abmelden und versuchen, sich mit falschen Anmeldedaten anzumelden.

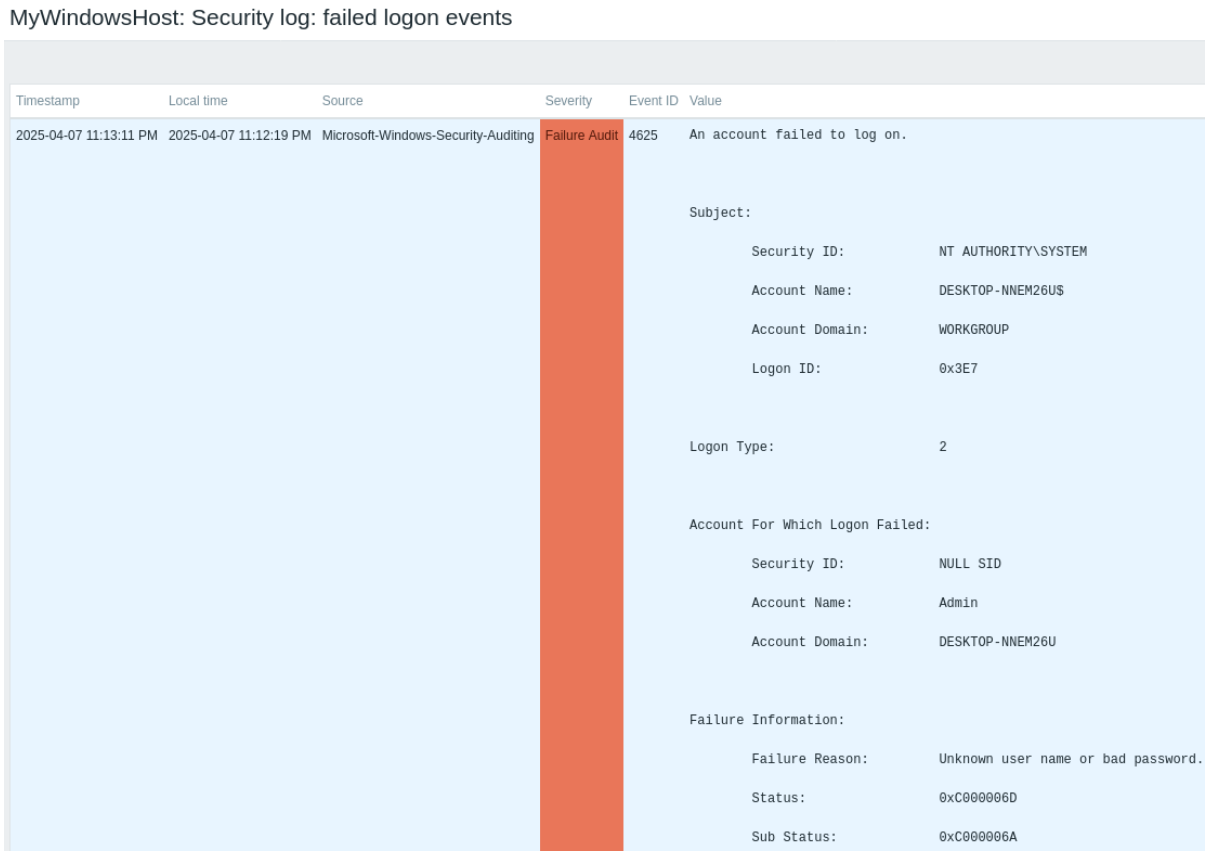
Sehen Sie sich anschließend die gesammelten Protokolle im Zabbix Frontend an:

1. Navigieren Sie im Zabbix Frontend zu *Monitoring* > *Latest data*.



2. Filtern Sie im Feld *Name* nach Ihrem Host ”MyWindowsHost”.

3. Klicken Sie auf *History*, um die aufgezeichneten Protokollwerte anzuzeigen.



4. Wenn keine Protokollwerte vorhanden sind, fahren Sie mit dem Abschnitt Troubleshooting in der Anleitung fort.

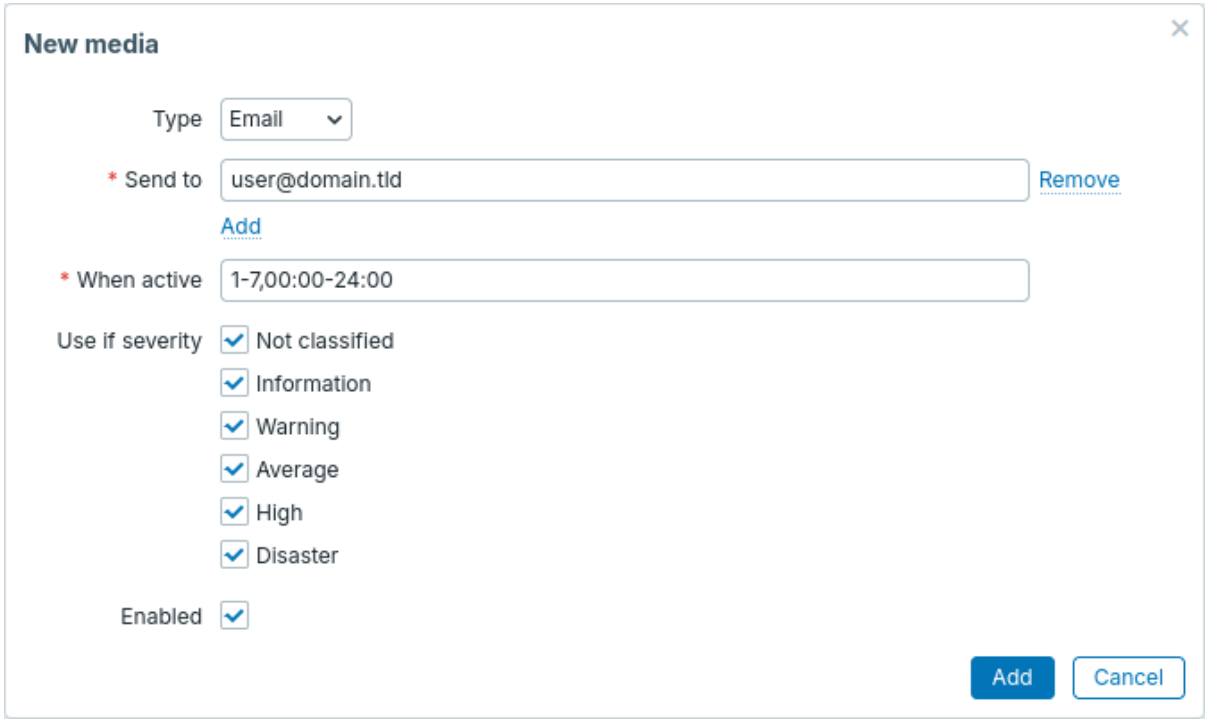
Problemalarme einrichten

Diese Anleitung enthält grundlegende Konfigurationsschritte zum Senden von E-Mail-Benachrichtigungen.

1. Navigieren Sie zu Datenerfassung > Hosts, um einen Auslöser zu definieren, der ausgelöst wird, wenn Ihr Ereignisprotokoll-Datenpunkt das gewünschte Muster erfasst. Um beispielsweise fehlgeschlagene Anmeldeversuche im Sicherheitsprotokoll zu erkennen, verwenden Sie die Funktion find():

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

2. Navigieren Sie zu *Benutzereinstellungen* > *Profil*, wechseln Sie zur Registerkarte *Medien* und fügen Sie Ihre E-Mail-Adresse hinzu.



3. Folgen Sie der Anleitung zum Empfangen einer Problembenachrichtigung.

Wenn Zabbix das nächste Mal ein Problem erkennt, sollten Sie eine Benachrichtigung per E-Mail erhalten.

Fehlerbehebung

Wenn beim Erfassen oder Anzeigen von Windows-Ereignisprotokollen Probleme auftreten, verwenden Sie die folgenden Tipps, um häufige Probleme zu identifizieren und zu beheben:

1. Listen Sie auf dem Zabbix Server (Linux) Ihre iptables-Regeln mit dem folgenden Befehl auf:

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

und überprüfen Sie, ob eine ACCEPT-Regel für TCP-Port 10051 vorhanden ist.

2. Stellen Sie sicher, dass Ihr `eventlog[...]`-Schlüssel den exakten Protokollnamen (Groß-/Kleinschreibung beachten), die Ereignis-ID, den Modus (z. B. skip) und andere Parameter genau so verwendet, wie in den Windows-spezifischen Item-Schlüsseln angegeben.

Siehe auch:

- Einen Datenpunkt erstellen - erfahren Sie, wie Sie zusätzliche Metriken hinzufügen.
- Zabbix Agent unter Microsoft Windows - detaillierte Installationsanweisungen.
- Windows mit Zabbix Agent überwachen - ein umfassender Leitfaden zur Einrichtung der grundlegenden Überwachung von Windows-Computern mit Zabbix Agent.
- Windows-spezifische Datenpunkt-Schlüssel - detaillierte Informationen zu Windows-spezifischen Datenpunkt-Schlüsseln, die von Zabbix Agents unterstützt werden, einschließlich derer für die Überwachung von Ereignisprotokollen.
- Überwachung von Protokolldateien - Anweisungen zur Konfiguration von Zabbix für die zentrale Überwachung und Analyse von Protokolldateien, anwendbar auf Windows-Ereignisprotokolle.

Installation: zusätzliche Anleitungen

Wenn Sie nach der einfachsten Möglichkeit suchen, Zabbix zu installieren, lesen Sie bitte Installation und erste Schritte.

Die folgenden Anleitungen behandeln **zusätzliche** oder systemspezifische Möglichkeiten zur Installation von Zabbix oder seinen Komponenten:

- Installation aus Containern
- Installation aus den Quellen
- Hinweise zur Installation aus Paketen
- Installation des Zabbix Agent aus einer Windows-MSI-Datei
- Erstellen des Zabbix Agent 2 unter Windows
- Erstellen des Zabbix Agent unter Windows

- Installation des macOS Agent aus einem PKG-Paket
- Zabbix Agent unter IBM AIX
- Erstellen des Zabbix Agent unter macOS
- Zabbix Appliance
- AWS-Bereitstellungsanleitung für den Zabbix Server
- AWS-Bereitstellungsanleitung für den Zabbix Proxy
- Azure-Bereitstellungsanleitung für den Zabbix Server
- Azure-Bereitstellungsanleitung für den Zabbix Proxy
- Bereitstellungsanleitung für Google Cloud

Installation aus Containern

Übersicht

Diese Seite beschreibt, wie Zabbix mit Docker oder Docker Compose bereitgestellt wird.

Docker Compose ist der schnellste Weg, Zabbix bereitzustellen. Es liest eine Konfigurationsdatei und startet automatisch alle Container, aus denen eine vollständige Zabbix-Installation besteht, in der richtigen Reihenfolge.

Docker (manuelle Bereitstellung) erreicht dasselbe Ergebnis, indem jede Komponente Schritt für Schritt bereitgestellt wird.

Voraussetzungen

Stellen Sie vor Beginn sicher, dass Docker (1.12.0 oder höher) auf Ihrem System installiert ist. Falls dies nicht der Fall ist, folgen Sie der Docker-Installationsanleitung. Für die Bereitstellung mit Docker Compose ist außerdem Docker Compose (2.24.0 oder höher) erforderlich.

Einige Zabbix-Komponenten erfordern, dass bestimmte Ports auf dem Host, auf dem Docker ausgeführt wird, geöffnet sind (z. B. 10051/TCP für den Zabbix-Server, 162/UDP für SNMP-Traps). Eine vollständige Liste der von Zabbix-Komponenten verwendeten Ports finden Sie unter Anforderungen. Für den Zabbix-Server und den Agent kann der Standardport geändert werden, indem die Umgebungsvariable `ZBX_LISTENPORT` im jeweiligen Container festgelegt wird.

Verfügbare Docker-Images

Zabbix stellt für jede Zabbix-Komponente ein Docker-Image bereit, die alle auf Docker Hub veröffentlicht werden. Jedes Image wird verwendet, um einen Container zu erstellen, in dem diese Komponente ausgeführt wird.

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

Note:
Um SNMP-Traps zu verwenden, muss der Container für SNMP-Traps ein Volume mit dem Zabbix-Server- oder Proxy-Container gemeinsam nutzen (siehe Verwendung dieses Images auf Docker Hub und das folgende Beispiel).

Alle Images der Zabbix-Komponenten auf Docker Hub basieren auf den neuesten Hauptversionen der unterstützten Betriebssysteme. Diese Images werden automatisch neu erstellt, wenn die zugrunde liegenden Betriebssystem-Images aktualisiert werden.

Image-Tags

Jedes Image unterstützt Tags zur Auswahl des Basisbetriebssystems und der Zabbix-Version:

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

Unterstützte <os>-Werte:

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

Unterstützte <version>-Werte:

- `latest` - Neueste stabile Zabbix-Version auf Alpine Linux
- `<os>-latest` - Neueste stabile Zabbix-Version auf dem ausgewählten Betriebssystem
- `<os>-trunk` - Neuester Entwicklungs-Build (nightly) auf dem ausgewählten Betriebssystem
- `<os>-X.X-latest` - Neueste Zabbix-Minor-Version einer bestimmten Zabbix-Major-Version auf dem ausgewählten Betriebssystem
- `<os>-X.X.*` - Bestimmte Zabbix-Minor-Version auf dem ausgewählten Betriebssystem

Beispiele:

```
#### Neuester stabiler Zabbix Proxy (SQLite3) auf Alpine Linux:
docker pull zabbix/zabbix-proxy-sqlite3:latest

#### Neuester stabiler Zabbix Proxy (SQLite3) auf Ubuntu:
docker pull zabbix/zabbix-proxy-sqlite3:ubuntu-latest

#### Neuester Entwicklungs-Build (nightly) von Zabbix Server (MySQL) auf Ubuntu:
docker pull zabbix/zabbix-server-mysql:ubuntu-trunk

#### Neueste 8.0-Minor-Version von Zabbix Server (MySQL) auf Alpine Linux:
docker pull zabbix/zabbix-server-mysql:alpine-8.0-latest

#### Version 8.0.1 von Zabbix Server (MySQL) auf Alpine Linux:
docker pull zabbix/zabbix-server-mysql:alpine-8.0.1
```

Docker Compose

Docker Compose ist der schnellste Weg, Zabbix bereitzustellen. Es liest eine Konfigurationsdatei (Compose-Datei) und übernimmt die gesamte Einrichtung – das Abrufen von Docker-Images, das Erstellen eines internen Netzwerks zwischen Containern, das Einrichten des Speichers, die Initialisierung der Datenbank und das Starten aller Komponenten in der richtigen Reihenfolge.

Das Repository offizieller Zabbix-Dockerfiles stellt sofort einsatzbereite Docker-Compose-Dateien und ein auf `.env` basierendes Konfigurationssystem bereit, das verschiedene Betriebssysteme, Datenbank-Backends und die Konfiguration der Zabbix-Komponenten unterstützt.

1. Klonen Sie das Repository, wechseln Sie in das entsprechende Verzeichnis und wechseln Sie zur Version 8.0:

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

2. (Optional) Passen Sie die Bereitstellung mit Umgebungsvariablen an. Dieser Schritt kann bei einer Standardkonfiguration übersprungen werden.

3. Stellen Sie die Standardkonfiguration bereit – Zabbix-Server, Weboberfläche (unter NGINX) und ein Datenbank-Backend (MySQL oder PostgreSQL), jeweils in einem eigenen Container unter Alpine Linux.

```
#### Mit MySQL als Datenbank:
docker compose -f ./compose.yaml up -d

#### Mit PostgreSQL als Datenbank:
docker compose -f ./compose_pgsql.yaml up -d
```

Nachdem die Container gestartet wurden (in der Regel innerhalb von 1–3 Minuten), startet der Zabbix-Server und die Weboberfläche steht zur Anmeldung bereit.

> **Note:**
> Verwenden Sie `docker compose ps`, um den Status der Container zu überprüfen. Alle Container (mit Ausnahme des Containers `zabbix-docker-server-db-init-1`) sollten den Status `Up` aufweisen. Wenn ein Container den Status `Exited` anzeigt, überprüfen Sie seine Protokolle mit `docker logs -f <container-name>` auf Fehler.

Umgebungsvariablen

Das Verhalten von Docker Compose und die Konfiguration der Zabbix-Komponenten können mithilfe von Umgebungsvariablen angepasst werden.

Variablen auf Compose-Ebene (definiert in der `.env`-Datei) steuern, welche Docker-Images, Ports und Netzwerk-IP-Bereiche verwendet werden. Sie können diese Variablen inline vor dem Befehl `docker compose` verwenden oder die `.env`-Datei bearbeiten.

Der folgende Befehl stellt beispielsweise eine vollständige Zabbix-Umgebung mit mehreren Containern mithilfe eines auf Ubuntu basierenden Images (anstelle des standardmäßigen Alpine Linux) bereit und stellt die Weboberfläche über NGINX an benutzerdefinierten HTTP- (8282) und HTTPS-Ports (8443) bereit:

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

Variablen auf Komponentenebene (definiert in den Dateien `env_vars/.env_<component>`) steuern die Konfiguration der Zabbix-Komponenten. Bearbeiten Sie die entsprechende `.env_<component>`-Datei, bevor Sie den Befehl `docker compose` ausführen.

Sie können beispielsweise die Anzahl der passiven Poller und die Cache-Einstellungen des Zabbix-Servers anpassen sowie die Zeitzone der Weboberfläche konfigurieren, indem Sie die folgenden Variablen bearbeiten:

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

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

> **Note:**
> Umgebungsvariablen auf Komponentenebene entsprechen den Konfigurationsparametern der Zabbix-Komponenten, verwenden jedoch eine andere Schreibweise (z. B. entspricht `ZBX_STARTPOLLERS` `StartPollers`). Einige Variablen sind Docker-spezifisch, und einige Konfigurationsparameter können nicht geändert werden (z. B. `PIDFile` und `LogType`). Wenn Sie Umgebungsvariablen verwenden, beachten Sie den Abschnitt *Environment variables* in der Beschreibung des jeweiligen Docker-Images auf Docker Hub.

Volumes

Docker Compose speichert persistente Daten im Verzeichnis `zbx_env/`, das neben der Compose-Datei erstellt wird. Dieses Verzeichnis bewahrt Komponentendaten über Container-Neustarts und Updates hinweg auf.

Der Inhalt von `zbx_env/` ist für das Image jeder Komponente vordefiniert. Zum Beispiel:

- `zbx_env/etc/` - wird vom Container der Zabbix-Weboberfläche verwendet
- `zbx_env/usr/` und `zbx_env/var/` - werden vom Zabbix-Server-Container verwendet

Weitere Informationen zu jedem Volume finden Sie im Abschnitt *Allowed volumes* in der Beschreibung des jeweiligen Docker-Images auf Docker Hub.

Makefile

Das Zabbix-Dockerfiles-Repository stellt außerdem eine Makefile als Abkürzung für häufig verwendete Docker-Compose-Aufgaben bereit. Anstatt den vollständigen Befehl `docker compose` zu verwenden, können Sie kürzere `make`-Befehle nutzen (führen Sie `make help` aus, um die verfügbaren Optionen anzuzeigen):

```
#### Standardbereitstellung (Zabbix-Server, Frontend, MySQL, alles unter Alpine Linux)
make up

#### Benutzerdefinierte Bereitstellung (Zabbix-Server, Frontend mit benutzerdefinierten NGINX-Ports, Postg
```

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

Attention:
Geben Sie beim Stoppen oder Entfernen von Containern immer denselben Datenbanktyp an, der bereitgestellt wurde (z. B. `make down DB=pgsql`).

Standardmäßig startet `make up` nur eine minimale Gruppe von Services (Zabbix-Server, Frontend und Datenbank). Dadurch bleibt die Einrichtung schlank, und es werden keine Komponenten gestartet, die möglicherweise nicht benötigt werden. Um zusätzliche Komponenten einzubinden, können Sie die folgenden Compose-Profile verwenden:

```
#### Standardbereitstellung + Zabbix Agent, Java gateway, Webservice und SNMP-Traps:
make up COMPOSE_PROFILES=full

#### COMPOSE_PROFILES=full + Zabbix Proxy (MySQL und SQLite3):
make up COMPOSE_PROFILES=all
```

Docker (manuelle Bereitstellung)

Verwenden Sie die manuelle Bereitstellung, wenn Sie Zabbix Schritt für Schritt bereitstellen, einzelne Komponenten ausführen, sie in eine vorhandene Umgebung integrieren oder eine alternative Container-Laufzeitumgebung wie Podman verwenden möchten.

Um beispielsweise einen Zabbix Proxy bereitzustellen, führen Sie den folgenden Befehl aus:

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

Dieser Befehl:

- Ruft das Docker-Image `zabbix/zabbix-proxy-sqlite3:alpine-8.0-latest` ab.
- Erstellt und startet einen Container `zabbix-proxy-sqlite3` auf Basis des Docker-Images (mit den Flags `--init` und `-d`).
- Konfiguriert den Zabbix Proxy-Parameter `Server` über die Umgebungsvariable `ZBX_SERVER_HOST`. Dieser Parameter gibt die IP-Adresse des Zabbix Servers an, von dem der Proxy Konfigurationsdaten abruft und an den er die gesammelten Daten sendet. Beachten Sie, dass andere für den aktiven Proxy-Betrieb erforderliche Variablen Standardwerte verwenden und weggelassen werden können.

Note:
Umgebungsvariablen entsprechen den Konfigurationsparametern von Zabbix-Komponenten mit einer anderen Namenskonvention (z. B. entspricht `ZBX_SERVER_HOST` `Server`). Einige Variablen sind Docker-spezifisch, und einige Konfigurationsparameter können nicht geändert werden (z. B. `PIDFile` und `LogType`). Wenn Sie Umgebungsvariablen verwenden, lesen Sie bitte den Abschnitt *Environment variables* in der Beschreibung des Docker-Images jeder Komponente auf Docker Hub.

- Bindet einen von Docker verwalteten Speicherort (z. B. `/var/lib/docker/volumes/zabbix-proxy-data/`) an das Verzeichnis `/var/lib/zabbix` des Containers, damit die Daten des Zabbix Proxy dauerhaft gespeichert werden, auch wenn der Container entfernt wird.

Note:
Weitere Informationen zu den einzelnen Volumes finden Sie im Abschnitt *Allowed volumes* in der Beschreibung des Docker-Images jeder Komponente auf Docker Hub.

Nachdem Sie den Zabbix Proxy-Container bereitgestellt haben, können Sie in Ihrer Zabbix Frontend mit dem Hinzufügen des Proxys fortfahren und Hosts konfigurieren, die von diesem Proxy überwacht werden sollen.

Die folgenden Beispiele decken drei weitere Bereitstellungsszenarien ab:

- Zabbix Server (MySQL) mit Java gateway
- Zabbix Server (PostgreSQL) mit SNMP-Traps
- Zabbix Server (MySQL) mit Java gateway auf RHEL 8–10

Weitere Beispiele finden Sie in der Beschreibung des Docker-Images jeder Komponente auf Docker Hub.

Zabbix Server (MySQL) mit Java gateway bereitstellen

Dieses Beispiel zeigt, wie Sie einen Zabbix Server mit einem MySQL-Backend, einem Java gateway für die JMX-Überwachung und einer webbasierten Oberfläche auf NGINX-Basis bereitstellen.

1. Erstellen Sie ein dediziertes Docker-Netzwerk, damit alle Container der Zabbix-Komponenten einander über den Containernamen erreichen können:

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

2. Starten Sie einen leeren MySQL-Server-Container (ersetzen Sie `zabbix_pwd` und `root_pwd` durch sichere Passwörter und verwenden Sie diese Werte in den folgenden Schritten):

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

3. Verfolgen Sie die MySQL-Protokolle und warten Sie, bis MySQL für Verbindungen bereit ist (verwenden Sie `Ctrl+C`, um die Protokollausgabe zu beenden):

```
docker logs -f mysql-server

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

4. Starten Sie einen Zabbix-Java-gateway-Container:

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

5. Aktivieren Sie `log_bin_trust_function_creators` im MySQL-Server-Container. Dadurch werden die Sicherheitseinschränkungen für gespeicherte Funktionen vorübergehend gelockert; die Einstellung wird in späteren Schritten deaktiviert. Dies ermöglicht es dem Zabbix-Datenbankbenutzer, gespeicherte Funktionen ohne das Privileg `SUPER` zu erstellen, das MySQL bei aktiviertem binären Logging benötigt (standardmäßig seit MySQL 8.0). Führen Sie den folgenden Befehl aus:

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

6. Starten Sie einen Zabbix-Server-Container und verbinden Sie ihn mit den Java-gateway- und MySQL-Server-Containern:

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

7. Verfolgen Sie die Protokolle des Zabbix Servers und warten Sie, bis der Zabbix Server die Initialisierung des Datenbankschemas abgeschlossen hat (verwenden Sie `Ctrl+C`, um die Protokollausgabe zu beenden):

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

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

8. Deaktivieren Sie `log_bin_trust_function_creators` im MySQL-Server-Container. Dadurch wird die Sicherheitseinschränkung wieder aktiviert, die Benutzer ohne SUPER-Privileg daran hindert, gespeicherte Funktionen zu erstellen. Führen Sie den folgenden Befehl aus:

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

9. Starten Sie einen Container für die Zabbix-Weboberfläche und verbinden Sie ihn mit den Zabbix-Server- und MySQL-Server-Containern:

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

Nachdem die Container gestartet wurden und ausgeführt werden, startet der Zabbix Server und die Weboberfläche steht zur Anmeldung zur Verfügung.

> **Note:**
> Verwenden Sie `docker ps`, um den Status der Container zu überprüfen. Alle Container sollten den Status Up haben. Wenn einer den Status Exited anzeigt, überprüfen Sie die Protokolle mit `docker logs -f <container-name>` auf Fehler.

Zabbix Server (PostgreSQL) mit SNMP-Traps bereitstellen

Dieses Beispiel zeigt, wie Sie einen Zabbix Server mit einem PostgreSQL-Backend, SNMP-Traps und einer webbasierten Oberfläche auf NGINX-Basis bereitstellen.

1. Erstellen Sie ein dediziertes Docker-Netzwerk, damit alle Container der Zabbix-Komponenten einander über den Containernamen erreichen können:

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

2. Starten Sie einen leeren PostgreSQL-Server-Container (ersetzen Sie `zabbix_pwd` durch ein sicheres Passwort und verwenden Sie diesen Wert in den folgenden Schritten):

```
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. Starten Sie einen Zabbix-SNMP-Traps-Container:

```
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. Starten Sie einen Zabbix-Server-Container und verbinden Sie ihn mit dem PostgreSQL-Server- und SNMP-Traps-Container:

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

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

5. Verfolgen Sie die Protokolle des Zabbix Servers und warten Sie, bis der Zabbix Server die Initialisierung des Datenbankschemas abgeschlossen hat (verwenden Sie `Ctrl+C`, um die Protokollausgabe zu beenden):

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

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

6. Starten Sie einen Container für die Zabbix-Weboberfläche und verbinden Sie ihn mit den Zabbix-Server- und PostgreSQL-Server-Containern:

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

Nachdem die Container gestartet wurden und ausgeführt werden, startet der Zabbix Server und die Weboberfläche steht zur Anmeldung zur Verfügung.

> **Note:**
> Verwenden Sie `docker ps`, um den Status der Container zu überprüfen. Alle Container sollten den Status `Up` haben. Wenn einer den Status `Exited` anzeigt, überprüfen Sie die Protokolle mit `docker logs -f <container-name>` auf Fehler.

Zabbix Server (MySQL) mit Java gateway unter RHEL 8–10 bereitstellen

Dieses Beispiel zeigt, wie Sie einen Zabbix Server mit einem MySQL-Backend, einem Java gateway für die JMX-Überwachung und einer webbasierten Oberfläche auf NGINX-Basis bereitstellen, die vollständig unter Red Hat Enterprise Linux 8, 9 oder 10 ausgeführt werden.

Unter Red Hat Enterprise Linux wird Podman anstelle von Docker als Container-Laufzeitumgebung empfohlen. Podman funktioniert ähnlich wie Docker, benötigt jedoch keinen als Root ausgeführten Hintergrunddienst und eignet sich daher besser für Red-Hat-Umgebungen.

1. Erstellen Sie einen neuen Pod mit dem Namen `zabbix` und geben Sie die Ports für die Zabbix-Weboberfläche und den Zabbix-Server-Trapper frei:

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

2. (Optional) Starten Sie einen Zabbix-Agent-Container im Pod `zabbix`:

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

3. Erstellen Sie ein Verzeichnis `./mysql/` auf dem RHEL-Host:

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

4. Starten Sie einen leeren MySQL-Server-Container (ersetzen Sie `zabbix_pwd` und `root_pwd` durch sichere Passwörter und verwenden Sie diese Werte in den folgenden Schritten):

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

5. Verfolgen Sie die MySQL-Protokolle und warten Sie, bis MySQL für Verbindungen bereit ist (verwenden Sie `Ctrl+C`, um die Protokollausgabe zu beenden):

```
podman logs -f mysql-server

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

6. Starten Sie einen Zabbix-Java-gateway-Container:

```
podman run --name zabbix-java-gateway -t \
  --restart=always \
  --pod=zabbix \
  -d registry.connect.redhat.com/zabbix/zabbix-java-gateway-80
```

7. Aktivieren Sie `log_bin_trust_function_creators` im MySQL-Server-Container. Dadurch werden die Sicherheitseinschränkungen für gespeicherte Funktionen vorübergehend gelockert; die Einstellung wird in späteren Schritten deaktiviert. Dies ermöglicht es dem Zabbix-Datenbankbenutzer, gespeicherte Funktionen ohne das Privileg `SUPER` zu erstellen, das MySQL bei aktiviertem binären Logging benötigt (standardmäßig seit MySQL 8.0). Führen Sie den folgenden Befehl aus:

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

8. Starten Sie einen Zabbix-Server-Container und verbinden Sie ihn mit den Java-gateway- und MySQL-Server-Containern:

```
podman run --name zabbix-server-mysql -t \
  -e DB_SERVER_HOST="127.0.0.1" \
  -e MYSQL_DATABASE="zabbix" \
  -e MYSQL_USER="zabbix" \
  -e MYSQL_PASSWORD="zabbix_pwd" \
  -e ZBX_JAVAGATEWAY="127.0.0.1" \
  --restart=always \
  --init \
  --pod=zabbix \
  -d registry.connect.redhat.com/zabbix/zabbix-server-mysql-80
```

9. Verfolgen Sie die Protokolle des Zabbix Servers und warten Sie, bis der Zabbix Server die Initialisierung des Datenbankschemas abgeschlossen hat (verwenden Sie `Ctrl+C`, um die Protokollausgabe zu beenden):

```
podman logs -f zabbix-server-mysql

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

10. Deaktivieren Sie `log_bin_trust_function_creators` im MySQL-Server-Container. Dadurch wird die Sicherheitseinschränkung wieder aktiviert, die Benutzer ohne `SUPER`-Privileg daran hindert, gespeicherte Funktionen zu erstellen. Führen Sie den folgenden Befehl aus:

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

11. Starten Sie einen Container für die Zabbix-Weboberfläche und verbinden Sie ihn mit den Zabbix-Server- und MySQL-Server-Containern:

```
podman run --name zabbix-web-mysql -t \
  -e ZBX_SERVER_HOST="127.0.0.1" \
  -e DB_SERVER_HOST="127.0.0.1" \
  -e MYSQL_DATABASE="zabbix" \
  -e MYSQL_USER="zabbix" \
  -e MYSQL_PASSWORD="zabbix_pwd" \
  -e PHP_TZ="Europe/Riga" \
  --restart=always \
  --pod=zabbix \
  -d registry.connect.redhat.com/zabbix/zabbix-web-mysql-80
```

Note:
Der Pod `zabbix` stellt den Port 80/TCP (HTTP) vom Port 8080/TCP des Containers `zabbix-web-mysql` für den Hostcomputer bereit.

Nachdem die Container gestartet wurden und ausgeführt werden, startet der Zabbix Server und die Weboberfläche steht zur Anmeldung zur Verfügung.

Note:
Verwenden Sie `docker ps`, um den Status der Container zu überprüfen. Alle Container sollten den Status `Up` haben. Wenn einer den Status `Exited` anzeigt, überprüfen Sie die Protokolle mit `docker logs -f <container-name>` auf Fehler.

Installation aus den Quellen

Sie können die allerneueste Version von Zabbix erhalten, indem Sie sie aus den Quellen kompilieren. Siehe auch Zabbix-Quellcode beziehen.

Eine Schritt-für-Schritt-Anleitung zur Installation von Zabbix aus den Quellen wird hier bereitgestellt.

Installation von Zabbix-Daemons

1 Quellarchiv herunterladen

Gehen Sie zur Zabbix-Download-Seite und laden Sie das Quellarchiv herunter. Nach dem Herunterladen entpacken Sie die Quellen, indem Sie Folgendes ausführen:

```
tar -zxvf zabbix-8.0.0.tar.gz
```

Note:
Geben Sie im Befehl die richtige Zabbix-Version an. Sie muss mit dem Namen des heruntergeladenen Archivs übereinstimmen.

2 Benutzerkonto erstellen

Alle Zabbix-Daemon-Prozesse werden unter nicht privilegierten Systembenutzern ausgeführt. Wenn ein Zabbix-Daemon von einem nicht privilegierten Benutzerkonto gestartet wird, läuft er weiterhin unter diesem Benutzer.

In der Standardkonfiguration wechselt ein als `root` gestarteter Daemon zum Benutzerkonto `zabbix`, das vorhanden sein muss. Um einen Systembenutzer und eine Gruppe `zabbix` zu erstellen, führen Sie die unten aufgeführten Befehle aus.

RedHat-basiertes System:

```
groupadd --system zabbix
useradd --system -g zabbix -d /usr/lib/zabbix -s /sbin/nologin -c "Zabbix Monitoring System" zabbix
```

Debian-basiertes System:

```
addgroup --system --quiet zabbix
adduser --quiet --system --disabled-login --ingroup zabbix --home /var/lib/zabbix --no-create-home zabbix
```

Es ist nicht erforderlich, ein separates Benutzerkonto für das Zabbix Frontend zu erstellen.

Sicherheitsempfehlung

Wenn der Zabbix Server und der Agent auf demselben Rechner ausgeführt werden, wird empfohlen, sie unter **separaten Benutzerkonten** auszuführen. Wenn beide unter demselben Benutzer ausgeführt werden, kann der Agent auf die Konfigurations-

datei des Servers zugreifen, wodurch vertrauliche Informationen – wie etwa das Datenbankpasswort – für jeden Benutzer mit Admin-Rechten in Zabbix offengelegt werden könnten.

Attention:
Die Ausführung von Zabbix als `root`, `bin` oder unter einem anderen Konto mit besonderen Rechten stellt ein Sicherheitsrisiko dar.

Home-Verzeichnis (optional)

Zabbix-Prozesse benötigen kein Home-Verzeichnis, daher wird dessen Erstellung im Allgemeinen nicht empfohlen. Wenn Sie jedoch Funktionen benötigen, die ein Home-Verzeichnis voraussetzen (zum Beispiel das Speichern von MySQL-Zugangsdaten in $HOME/.my.cnf), können Sie es mit den unten aufgeführten Befehlen erstellen.

Führen Sie auf RedHat-basierten Systemen Folgendes aus:

```
mkdir -m u=rwx,g=rwx,o= -p /usr/lib/zabbix
chown zabbix:zabbix /usr/lib/zabbix
```

Führen Sie auf Debian-basierten Systemen Folgendes aus:

```
mkdir -m u=rwx,g=rwx,o= -p /var/lib/zabbix
chown zabbix:zabbix /var/lib/zabbix
```

3 Zabbix-Datenbank erstellen

Für die Zabbix-Server- und Proxy-Daemons sowie das Zabbix-Frontend wird eine Datenbank benötigt. Für den Betrieb des Zabbix-Agent ist sie nicht erforderlich.

SQL-Skripte werden bereitgestellt, um das Datenbankschema zu erstellen und den Datensatz einzufügen. Die Zabbix-Proxy-Datenbank benötigt nur das Schema, während die Zabbix-Server-Datenbank zusätzlich zum Schema auch den Datensatz erfordert.

Nachdem Sie eine Zabbix-Datenbank erstellt haben, fahren Sie mit den folgenden Schritten zum Kompilieren von Zabbix fort.

4 Quellen konfigurieren

Für die Erstellung von Zabbix Server, Zabbix Proxy oder Zabbix Agent ist C99 mit GNU-Erweiterungen erforderlich. Diese Version kann explizit durch das Setzen von CFLAGS=”-std=gnu99” angegeben werden:

```
export CFLAGS="-std=gnu99"
```

Note:
Bei der Installation aus dem Zabbix Git-Repository muss zuerst Folgendes ausgeführt werden:
`./bootstrap.sh`

Beim Konfigurieren der Quellen für einen Zabbix Server oder Proxy müssen Sie den zu verwendenden Datenbanktyp angeben. Mit einem Server- oder Proxy-Prozess kann jeweils nur ein Datenbanktyp kompiliert werden.

Um alle unterstützten Konfigurationsoptionen anzuzeigen, führen Sie im entpackten Zabbix-Quellverzeichnis Folgendes aus:

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

Um die Quellen für einen Zabbix Server und Agent zu konfigurieren, können Sie beispielsweise Folgendes ausführen:

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

Um die Quellen für einen Zabbix Server (mit PostgreSQL usw.) zu konfigurieren, können Sie beispielsweise Folgendes ausführen:

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

Um die Quellen für einen Zabbix Proxy (mit SQLite usw.) zu konfigurieren, können Sie beispielsweise Folgendes ausführen:

```
./configure --prefix=/usr --enable-proxy --with-net-snmp --with-sqlite3 --with-ssh2
```

Um die Quellen für einen Zabbix Agent zu konfigurieren, können Sie Folgendes ausführen:

```
./configure --enable-agent
```

oder für Zabbix Agent 2:

```
./configure --enable-agent2
```

Note:
Zum Erstellen von Zabbix Agent 2 muss eine unterstützte Go-Version installiert sein.

Hinweise zu den Kompilierungsoptionen:

- `--enable-agent` - kompiliert den Zabbix Agent sowie die Befehlszeilenprogramme Zabbix get und Zabbix sender.
- `--with-libcurl` - erforderlich für die Überwachung virtueller Maschinen, die SMTP-Authentifizierung sowie `web.page.*`-Datenpunkte des Zabbix Agent und Datenpunkte für Telemetrieabfragen. Siehe auch: Anforderungen (libcurl).
- `--with-libxml2` - erforderlich für die Überwachung virtueller Maschinen.
- `--with-libpcre2[=DIR]` - Zabbix wird immer mit der PCRE2-Bibliothek kompiliert; diese Option ermöglicht lediglich die Angabe eines benutzerdefinierten Installationspfads für PCRE2.
- `--with-mysql=/path/to/mysql_config` - gibt den Pfad zur Konfiguration einer bestimmten MySQL-Clientbibliothek an. Dies ist nützlich, wenn mehrere Versionen von MySQL oder MariaDB installiert sind.
- `--enable-static` - verknüpft Bibliotheken statisch (nicht unterstützt auf Solaris). Verwenden Sie diese Option, wenn Sie kompilierte Binärdateien auf Systeme ohne die erforderlichen Bibliotheken verteilen möchten. Beim Erstellen des Zabbix Servers wird dies nicht empfohlen. Um den Server statisch zu erstellen, ist eine statische Version jeder externen Bibliothek erforderlich. Das Konfigurationsskript prüft dies nicht automatisch.
- `--with-stacksize=<value>` - legt die Stackgröße pro Thread in Kilobyte fest (z. B. `--with-stacksize=512`). Sie können diesen Wert erhöhen, wenn Zabbix aufgrund von Stacküberläufen abstürzt oder einfriert (z. B. während der Vorverarbeitung auf Systemen mit niedrigen Standardgrenzwerten für Thread-Stacks).
- `--with-apm` (nur mit `--enable-proxy` verwenden) - erforderlich für die Überwachung der Anwendungsleistung (APM) auf einem Zabbix Proxy. Siehe auch: Anforderungen (g++, gRPC, protobuf, Abseil).

Attention:
Wenn `./configure` aufgrund fehlender Bibliotheken oder anderer Probleme fehlschlägt, prüfen Sie die Datei `config.log` auf detaillierte Fehlerinformationen.
Wenn beispielsweise `libssl` fehlt, kann die unmittelbare Fehlermeldung irreführend sein:

```
checking for main in -lmysqlclient... no
configure: error: Not found mysqlclient library
```

In diesem Fall zeigt `config.log` die tatsächliche Ursache:

```
/usr/bin/ld: cannot find -lssl
/usr/bin/ld: cannot find -lcrypto
```

Siehe auch:

- Zabbix mit Verschlüsselungsunterstützung kompilieren
- Bekannte Kompilierungsprobleme

5 Alles erstellen und installieren

Note:
Bei der Installation aus dem Zabbix Git-Repository muss zuerst Folgendes ausgeführt werden:

```
$ make dbschema
```

`make install`

Dieser Schritt sollte von einem Benutzer mit ausreichenden Berechtigungen ausgeführt werden (üblicherweise `root` oder mithilfe von `sudo`).

Wenn `make install` ausgeführt wird, werden standardmäßig die Daemon-Binärdateien (`zabbix_server`, `zabbix_agentd`, `zabbix_proxy`) in `/usr/local/sbin` und die Client-Binärdateien (`zabbix_get`, `zabbix_sender`) in `/usr/local/bin` installiert.

Note:
Um einen anderen Speicherort als `/usr/local` anzugeben, verwenden Sie im vorherigen Schritt der Quellkonfiguration den Schlüssel `--prefix`, zum Beispiel `--prefix=/home/zabbix`. In diesem Fall werden die Daemon-Binärdateien unter `<prefix>/sbin` installiert, während die Dienstprogramme unter `<prefix>/bin` installiert werden. Die Handbuchseiten werden unter `<prefix>/share` installiert.

6 Konfigurationsdateien überprüfen und bearbeiten

- Bearbeiten Sie die Agent-Konfigurationsdatei von Zabbix (`zabbix_agentd.conf`).

Sie müssen diese Datei für jeden Host konfigurieren, auf dem `zabbix_agentd` installiert ist.

In der Datei müssen Sie die **IP-Adresse** des Zabbix-Servers angeben. Verbindungen von anderen Hosts werden abgelehnt.

- Bearbeiten Sie die Server-Konfigurationsdatei von Zabbix (`zabbix_server.conf`).

Sie müssen den Datenbanknamen, den Benutzer und das Passwort (falls verwendet) angeben.

Die übrigen Parameter können bei einer kleinen Installation (bis zu zehn überwachte Hosts) mit ihren Standardwerten beibehalten werden. Wenn Sie jedoch die Leistung des Zabbix-Servers (oder Proxys) maximieren möchten, sollten Sie die Standardparameter ändern.

- Wenn Sie einen Zabbix-Proxy installiert haben, bearbeiten Sie die Proxy-Konfigurationsdatei (`zabbix_proxy.conf`).

Sie müssen die IP-Adresse des Servers und den Hostnamen des Proxys (der dem Server bekannt sein muss) sowie den Datenbanknamen, den Benutzer und das Passwort (falls verwendet) angeben.

> **Note:**
> Bei SQLite muss der vollständige Pfad zur Datenbankdatei angegeben werden; ein Datenbankbenutzer und ein Passwort sind nicht erforderlich.

7 Daemons starten

Führen Sie zabbix_server auf der Server-Seite aus.

```
zabbix_server
```

> **Note:**
> Stellen Sie sicher, dass Ihr System die Zuweisung von 36 MB (oder etwas mehr) gemeinsam genutztem Speicher erlaubt, andernfalls startet der Server möglicherweise nicht und Sie sehen „Cannot allocate shared memory for <type of cache>." in der Server-Logdatei. Dies kann unter FreeBSD und Solaris 8 auftreten.

Führen Sie zabbix_agentd auf allen überwachten Rechnern aus.

```
zabbix_agentd
```

> **Note:**
> Stellen Sie sicher, dass Ihr System die Zuweisung von 2 MB gemeinsam genutztem Speicher erlaubt, andernfalls startet der Agent möglicherweise nicht und Sie sehen „Cannot allocate shared memory for collector." in der Agent-Logdatei. Dies kann unter Solaris 8 auftreten.

Wenn Sie den Zabbix Proxy installiert haben, führen Sie zabbix_proxy aus.

```
zabbix_proxy
```

Installation des Zabbix-Web-Interface

Kopieren von PHP-Dateien

Das Zabbix-Frontend ist in PHP geschrieben, daher wird zum Ausführen ein Webserver mit PHP-Unterstützung benötigt. Die Installation erfolgt einfach durch Kopieren der PHP-Dateien aus dem Verzeichnis ui in das HTML-Dokumentenverzeichnis des Webservers.

Zu den üblichen Speicherorten von HTML-Dokumentenverzeichnissen für Apache-Webserver gehören:

- /usr/local/apache2/htdocs (Standardverzeichnis bei der Installation von Apache aus den Quellen)
- /srv/www/htdocs (OpenSUSE, SLES)
- /var/www/html (Debian, Ubuntu, Fedora, RHEL)

Es wird empfohlen, ein Unterverzeichnis anstelle des HTML-Root-Verzeichnisses zu verwenden. Um ein Unterverzeichnis zu erstellen und die Zabbix-Frontend-Dateien dorthin zu kopieren, führen Sie die folgenden Befehle aus und ersetzen Sie <htdocs> durch das tatsächliche Verzeichnis:

```
mkdir <htdocs>/zabbix
cd ui
cp -a . <htdocs>/zabbix
```

Wenn Sie eine andere Sprache als Englisch verwenden möchten, finden Sie in Installation zusätzlicher Frontend-Sprachen die entsprechenden Anweisungen.

Installation des Frontends

Informationen zum Installationsassistenten des Zabbix-Frontends finden Sie auf der Seite Installation der Weboberfläche.

Installieren von ladbaren Zabbix agent 2-Plugins

Die Installation von ladbaren Zabbix agent 2-Plugins ist nur erforderlich, wenn Sie Ziele überwachen möchten, die nicht von integrierten Plugins abgedeckt werden (z. B. MongoDB-Server oder -Cluster, PostgreSQL und seine Forks usw.). Siehe die vollständige Liste der ladbaren Plugins und integrierten Plugins.

Attention:
Bitte prüfen Sie vor der Installation eines Plugins die README-Datei. Sie kann spezifische Anforderungen und Installationsanweisungen enthalten.

Um aus den Quellen zu installieren, laden Sie zunächst das Quellarchiv des ladbaren Plugins von hier herunter und entpacken Sie es.

Um das Plugin zu kompilieren, wechseln Sie in das entpackte Plugin-Verzeichnis und führen Sie `make` aus:

```
make
```

Note:
Zum Erstellen ladbarer Zabbix agent 2-Plugins muss eine unterstützte Go-Version installiert sein.

Die ausführbare Plugin-Datei kann an beliebiger Stelle abgelegt werden, solange sie von Zabbix agent 2 geladen werden kann. Geben Sie den Pfad zur Plugin-Binärdatei in der Plugin-Konfigurationsdatei an, z. B. in `postgresql.conf` für das PostgreSQL-Plugin:

```
Plugins.PostgreSQL.System.Path=/path/to/executable/zabbix-agent2-plugin-postgresql
```

Der Pfad zur Plugin-Konfigurationsdatei muss im Parameter `Include` der Zabbix agent 2-Konfigurationsdatei angegeben werden:

```
Include=/path/to/plugin/configuration/file/postgresql.conf
```

Weitere Details zur Konfiguration von Plugins finden Sie unter Einrichtung.

Von Zabbix bereitgestellte ladbare Plugins verwenden einfache Makefiles mit den folgenden Build-Zielen:

- `make` - das Plugin erstellen
- `make clean` - alle Dateien löschen, die beim Erstellen des Plugins erzeugt wurden
- `make check` - Selbsttests ausführen (erfordert ein echtes Überwachungsziel, z. B. eine PostgreSQL-Datenbank)
- `make style` - den Go-Code-Stil mit `golangci-lint` prüfen
- `make format` - den Go-Code mit `go fmt` formatieren
- `make dist` - ein Quellarchiv einschließlich aller Abhängigkeiten erstellen

Installation von Java gateway

Es ist nur dann erforderlich, Java gateway zu installieren, wenn Sie JMX-Anwendungen überwachen möchten. Java gateway ist leichtgewichtig und benötigt keine Datenbank.

Um aus den Quellen zu installieren, laden Sie zunächst das Quellarchiv herunter und entpacken Sie es.

Um Java gateway zu kompilieren, führen Sie das Skript `./configure` mit der Option `--enable-java` aus. Es wird empfohlen, die Option `--prefix` anzugeben, um einen Installationspfad außerhalb des Standardpfads `/usr/local` festzulegen, da die Installation von Java gateway nicht nur eine einzelne ausführbare Datei, sondern einen kompletten Verzeichnisbaum erstellt.

```
./configure --enable-java --prefix=$PREFIX
```

Um Java gateway zu kompilieren und als JAR-Datei zu paketieren, führen Sie `make` aus. Beachten Sie, dass Sie für diesen Schritt die ausführbaren Dateien `javac` und `jar` in Ihrem Pfad benötigen.

```
make
```

Nun haben Sie die Datei `zabbix-java-gateway-$VERSION.jar` in `src/zabbix_java/bin`. Wenn Sie Java gateway problemlos aus `src/zabbix_java` im Distributionsverzeichnis ausführen können, können Sie mit den Anweisungen zum Konfigurieren und Ausführen von Java gateway fortfahren. Andernfalls stellen Sie sicher, dass Sie über ausreichende Berechtigungen verfügen, und führen Sie `make install` aus.

```
make install
```

Weitere Details zur Konfiguration und zum Betrieb von Java gateway finden Sie unter Einrichtung.

Installieren des Zabbix web service

Die Installation des Zabbix web service ist nur erforderlich, wenn Sie geplante Berichte verwenden möchten.

Um aus den Quellen zu installieren, laden Sie zunächst das Quellarchiv herunter und entpacken Sie es.

Um den Zabbix web service zu kompilieren, führen Sie das Skript `./configure` mit der Option `--enable-webservice` aus.

Note:
Eine unterstützte Go-Version muss installiert sein, um den Zabbix web service zu erstellen.

Starten Sie zabbix_web_service auf dem Rechner, auf dem der web service installiert ist:

```
zabbix_web_service
```

Weitere Details zur Konfiguration der Erstellung geplanter Berichte finden Sie unter Einrichtung.

Zabbix MCP-Server installieren

Um die Installation aus den Quellen durchzuführen, laden Sie zunächst das Quellarchiv herunter und extrahieren Sie es.

Um den Zabbix MCP-Server zu kompilieren, führen Sie im Quellverzeichnis des MCP-Servers den folgenden Befehl aus:

```
make build
```

> **Note:**
> Zum Erstellen des Zabbix MCP-Servers muss eine unterstützte Go-Version installiert sein.

Um eine bestimmte Version in der Ausgabe von `zabbix_mcp_server -V` und im `User-Agent` der Zabbix-API-Anfragen anzugeben, setzen Sie die Variable `VERSION`, zum Beispiel mit `make build VERSION=8.0.0`.

Um die Binärdatei nach `/usr/local/sbin` und die Beispielkonfigurationsdatei nach `/etc/zabbix` zu installieren, führen Sie Folgendes aus:

```
sudo make install
```

Bearbeiten Sie die Konfigurationsdatei `/etc/zabbix/zabbix_mcp_server.conf` und setzen Sie mindestens die folgenden Parameter:

- `FrontendURL` - die URL der Zabbix-API, die mit `api_jsonrpc.php` endet;
- `AllowedIP` - die Adressen der MCP-Clients, denen die Verbindung erlaubt ist.

Um Verbindungen von MCP-Clients zu verschlüsseln, setzen Sie `TLSAccept=cert` und geben Sie `TLSCertFile` sowie `TLSKeyFile` an, oder führen Sie den MCP-Server hinter einem vertrauenswürdigen Reverse-Proxy aus, der TLS verarbeitet.

Falls der Benutzer `zabbix` nicht existiert (zum Beispiel, wenn nur der MCP-Server auf dem Rechner installiert ist), erstellen Sie ihn wie unter Benutzerkonto erstellen beschrieben.

Überprüfen Sie die Konfigurationsdatei und starten Sie zabbix_mcp_server:

```
zabbix_mcp_server -c /etc/zabbix/zabbix_mcp_server.conf -T
zabbix_mcp_server -c /etc/zabbix/zabbix_mcp_server.conf
```

Weitere Informationen zur Konfiguration des Servers und zur Verbindung von MCP-Clients finden Sie unter MCP-Server.

Zabbix-Quellcode beziehen

Es gibt mehrere Möglichkeiten, den Zabbix-Quellcode zu beziehen:

- Sie können die veröffentlichten stabilen Versionen von der offiziellen Zabbix-Website herunterladen
- Sie können Nightly Builds von der Entwicklerseite der offiziellen Zabbix-Website herunterladen
- Sie können die neueste Entwicklungsversion aus dem Git-Quellcode-Repository beziehen:
  - Der primäre Speicherort des vollständigen Repositorys ist https://git.zabbix.com/scm/zbx/zabbix.git
  - Master und unterstützte Releases werden außerdem auf Github gespiegelt unter https://github.com/zabbix/zabbix

Zum Klonen des Repositorys muss ein Git-Client installiert sein. Das offizielle Git-Client-Paket für die Befehlszeile heißt in Distributionen üblicherweise **git**. Um es beispielsweise unter Debian/Ubuntu zu installieren, führen Sie Folgendes aus:

```
sudo apt-get update
sudo apt-get install git
```

Um den gesamten Zabbix-Quellcode zu holen, wechseln Sie in das Verzeichnis, in dem Sie den Code ablegen möchten, und führen Sie aus:

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
```

Kompilierungsprobleme

Dies sind die bekannten Probleme bei der Kompilierung von Zabbix aus den Quellen. Für alle anderen Fälle siehe die Seite Bekannte Probleme.

Bibliothek an einem nicht standardmäßigen Speicherort

Zabbix ermöglicht es Ihnen, eine Bibliothek an einem nicht standardmäßigen Speicherort anzugeben. Im folgenden Beispiel führt Zabbix `curl-config` vom angegebenen nicht standardmäßigen Speicherort aus und verwendet dessen Ausgabe, um die richtige zu verwendende libcurl zu bestimmen.

```
$ ./configure --enable-server --with-mysql --with-libcurl=/usr/local/bin/curl-config
```

Dies funktioniert, wenn dies die einzige im System installierte libcurl ist, möglicherweise jedoch nicht, wenn eine weitere libcurl an einem standardmäßigen Speicherort installiert ist (zum Beispiel durch den Paketmanager). Dies ist der Fall, wenn Sie für Zabbix eine neuere Version der Bibliothek benötigen und die ältere für andere Anwendungen.

Daher funktioniert die Angabe einer Komponente an einem nicht standardmäßigen Speicherort nicht immer, wenn dieselbe Komponente auch an einem standardmäßigen Speicherort vorhanden ist.

Wenn Sie beispielsweise eine neuere in `/usr/local` installierte libcurl verwenden, während das libcurl-Paket weiterhin installiert ist, könnte Zabbix die falsche Bibliothek auswählen und die Kompilierung schlägt fehl:

```
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
```

Hier ist die Funktion `curl_easy_header()` in der älteren `/usr/lib/x86_64-linux-gnu/libcurl.so` nicht verfügbar, jedoch in der neueren `/usr/local/lib/libcurl.so`.

Das Problem liegt in der Reihenfolge der Linker-Flags, und eine Lösung besteht darin, den vollständigen Pfad zur Bibliothek in einer LDFLAGS-Variablen anzugeben:

```
$ LDFLAGS="-Wl,--no-as-needed /usr/local/lib/libcurl.so" ./configure --enable-server --with-mysql --with-l
```

Beachten Sie die Option `-Wl,--no-as-needed`, die auf einigen Systemen erforderlich sein kann (siehe auch: Standard-Linking-Optionen auf Debian-basierten Systemen).

Stackgröße auf einigen Systemen zu klein

Wenn Zabbix aufgrund von Stack-Überläufen abstürzt oder einfriert, können Sie die Stackgröße pro Thread mit der Option `--with-stacksize` erhöhen, wenn Sie die Quellen konfigurieren. Dieses Problem kann auf Systemen mit niedrigen Standardlimits für den Thread-Stack auftreten, insbesondere während der Vorverarbeitung, wenn mehrere Threads erstellt werden.

Das folgende Beispiel setzt die Stackgröße auf 512 KB pro Thread:

```
./configure --enable-server --with-mysql --with-stacksize=512
```

Sie können die Stack-Limits für System-Threads zur Laufzeit mit dem Befehl `ulimit -s` auf Linux-basierten Systemen überprüfen.

Hinweise zur Installation aus Paketen

Übersicht

Offizielle **Vorabversionen** von Zabbix 8.0 sind auf der Zabbix-Website verfügbar. Dort können Sie Ihr Betriebssystem und die Zabbix-Komponente auswählen, um Installationsanweisungen und Befehle zu generieren.

Zusätzlich zu den unten aufgeführten Hinweisen zur Paketinstallation konsultieren Sie bitte auch die Seite Installation und Einrichtung.

Pakete sind für die folgenden Linux-Distributionen verfügbar:

- Red Hat Enterprise Linux und seine Derivate: 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:**
> Einige Betriebssystemdistributionen, insbesondere Debian-basierte Distributionen, stellen eigene Zabbix-Pakete bereit. Diese Pakete werden von Zabbix **nicht** unterstützt und können veraltet sein oder die neuesten Funktionen und Fehlerbehebungen nicht enthalten. Es wird empfohlen, ausschließlich offizielle Pakete aus dem offiziellen Zabbix-Repository zu verwenden. Wenn Sie Zabbix zuvor aus dem Repository Ihres Betriebssystems installiert haben, finden Sie unter Aktualisieren von Zabbix-Paketen aus Betriebssystem-Repositorys die entsprechenden Schritte.

Die Pakete unterstützen MySQL/PostgreSQL-Datenbanken sowie Apache/Nginx-Webserver. Beachten Sie, dass Zabbix Server und Proxy nicht dieselbe Datenbank verwenden können. Verwenden Sie unterschiedliche Datenbanknamen, wenn beide auf demselben Host installiert sind.

Falls erforderlich, sind separate Pakete für Zabbix Agent/Agent 2, Zabbix get und Zabbix sender im offiziellen Zabbix-Repository verfügbar.

Auf Debian- oder Ubuntu-basierten Systemen kann die Installation eines zusätzlichen Zabbix-Pakets mit `apt` auch Aktualisierungen für bereits installierte Zabbix-Pakete aus demselben Repository auswählen. Um ein Paket zu installieren, ohne ein installiertes Zabbix-Server- oder Proxy-Paket zu aktualisieren, halten Sie dieses Paket vor der Installation vorübergehend zurück und heben Sie die Zurückhaltung anschließend wieder auf:

```
apt-mark hold zabbix-server-mysql
apt install zabbix-sender
apt-mark unhold zabbix-server-mysql
```

Zabbix stellt außerdem vorkompilierte Binärdateien des Zabbix Agent für Nicht-Linux-Betriebssysteme bereit. Siehe:

- Installation des Windows Agent aus einer MSI-Datei
- Installation des macOS Agent aus einer PKG-Datei
- Zabbix Agent unter IBM AIX
- Legacy-Binärdateien (für ältere oder weniger verbreitete Systeme wie HP-UX, NetBSD, Tru64 und ältere Versionen von SLES)

Hinweise zur Paketinstallation

Die folgenden Hinweise gelten für alle Systeme:

- Bei Verwendung von PostgreSQL bewirkt `DBHost=localhost` (oder eine IP-Adresse) in der Zabbix-Konfiguration von Server/Proxy, dass PostgreSQL einen Netzwerk-Socket anstelle eines lokalen UNIX-Sockets verwendet. Informationen zur entsprechenden Einrichtung finden Sie unter SELinux-Konfiguration.
- Bei Verwendung von TimescaleDB finden Sie weitere Informationen unter TimescaleDB-Einrichtung.
- Wenn Sie das Zabbix-Java gateway (zur Überwachung von JMX-Anwendungen) installieren, finden Sie weitere Einrichtungsanweisungen für RHEL-basierte Systeme und Debian-basierte Systeme.
- Wenn Sie den Zabbix-MCP server installieren, installieren Sie das Paket `zabbix-mcp-server` und aktualisieren Sie die Konfigurationsdatei (mindestens `FrontendURL` und `AllowedIP`), bevor Sie den Dienst starten.
- Informationen zum Ausführen des Zabbix Agent als Root finden Sie unter Agent als Root ausführen.
- Das Application Performance Monitoring (APM) auf einem Zabbix Proxy wird von folgenden Paketen unterstützt:
  - RHEL 10 und dessen Derivate - installieren Sie das Zabbix-Proxy-APM-Paket (z. B. `zabbix-proxy-mysql-apm` anstelle von `zabbix-proxy-mysql`). Außerdem müssen Sie das EPEL-Repository installieren und die Paket-Repositories für CRB-Abhängigkeiten aktivieren.
  - Debian 12 oder höher, Raspberry Pi OS (Raspbian) 12 oder höher, Ubuntu 24.04 oder höher - APM ist in den Zabbix-Proxy-Paketen enthalten.
  - Auf anderen Systemen können Sie den Zabbix Proxy aus den Quellen mit der Option `--with-apm` installieren.

Die folgenden Hinweise gelten für RHEL und dessen Derivate:

- Wenn Sie das EPEL-Repository aktiviert haben (z. B. für das Zabbix-Proxy-APM-Paket), das ebenfalls Zabbix-Pakete bereitstellt, muss es vor der Installation offizieller Zabbix-Pakete von der Paketauflösung ausgeschlossen werden. Siehe Unbeabsichtigte Installation von EPEL-Zabbix-Paketen.
- Informationen zur Installation von Zabbix-Paketen in Red Hat-UBI-Umgebungen finden Sie unter Zabbix-Pakete für RHEL in Red Hat-UBI-Umgebungen.
- Für die Verwendung von ICMP-Ping-Datenpunkten sind Pakete für `fping` ebenfalls im offiziellen Zabbix-Repository verfügbar.

SELinux-Konfiguration

Zabbix verwendet socketbasierte Interprozesskommunikation. Auf Systemen, auf denen Security-Enhanced Linux (SELinux) aktiviert ist, müssen Sie möglicherweise SELinux-Regeln hinzufügen, damit Zabbix UNIX-Domain-Sockets im Verzeichnis SocketDir erstellen und verwenden kann. Socket-Dateien werden von Zabbix Server (alerter, preprocessing, IPMI) und Zabbix Proxy (IPMI) verwendet und sind vorhanden, solange der Prozess ausgeführt wird.

Wenn SELinux im Enforcing-Modus aktiviert ist, führen Sie die folgenden Befehle aus, um die Kommunikation zwischen Zabbix Frontend und Server zu ermöglichen:

Für RHEL 7 (und neuer), AlmaLinux, CentOS Stream, Oracle Linux, Rocky Linux 8 (und neuer):

```
setsebool -P httpd_can_connect_zabbix on
```

Wenn auf die Datenbank über das Netzwerk zugegriffen wird (einschließlich `localhost` für PostgreSQL), erlauben Sie dem Zabbix Frontend außerdem die Verbindung zur Datenbank:

```
setsebool -P httpd_can_network_connect_db on
```

Für RHEL vor Version 7:

```
setsebool -P httpd_can_network_connect on
setsebool -P zabbix_can_network on
```

Starten Sie nach dem Anwenden der SELinux-Einstellungen Apache neu:

```
systemctl restart httpd
```

Optional können Sie ein vordefiniertes Paket `zabbix-selinux-policy` aus dem Zabbix Official Repository installieren. Dieses Paket wird für alle unterstützten Betriebssystemversionen bereitgestellt, um die Bereitstellung von Zabbix zu vereinfachen und zu verhindern, dass Benutzer SELinux aufgrund der Komplexität der Konfiguration deaktivieren.
 Für Version 10 von AlmaLinux, CentOS Stream, RHEL, Rocky Linux und Oracle Linux wird die SELinux-Richtlinie durch das Paket `zabbix-selinux-policy-main` bereitgestellt; das Paket `zabbix-selinux-policy` hat eine Abhängigkeit entweder von `zabbix-selinux-policy-main` oder `selinux-policy-targeted-extra`. Das Paket `zabbix-selinux-policy-main` steht in Konflikt mit `selinux-policy-targeted-extra`; daher können beide Pakete nicht gleichzeitig installiert werden.

Attention:
Für maximale Sicherheit wird empfohlen, benutzerdefinierte SELinux-Einstellungen zu verwenden.

Das Paket `zabbix-selinux-policy` (`zabbix-selinux-policy-main` für Version 10 von AlmaLinux, CentOS Stream, RHEL, Rocky Linux und Oracle Linux) enthält eine grundlegende SELinux-Richtlinie, die es Zabbix erlaubt, Sockets zu erstellen und zu verwenden, und die die HTTPd-Verbindung zu PostgreSQL (von Frontend verwendet) aktiviert.

Die Quelldatei `zabbix_policy.te` enthält die folgenden Regeln:

```
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;
```

Debuginfo-Pakete

Debuginfo-Pakete enthalten Debugging-Symbole für Zabbix-Binärdateien. Sie sind für die normale Installation oder den normalen Betrieb nicht erforderlich, aber für erweiterte Fehlerbehebung nützlich.

Bitte beachten Sie, dass Debuginfo-Pakete für die folgenden Versionen verfügbar sind: Red Hat Enterprise Linux 7 und älter (einschließlich Derivate) sowie SUSE Linux Enterprise Server 15 und älter (einschließlich Derivate).

Um das Repository `zabbix-debuginfo` zu aktivieren:

- Unter RHEL 7 bearbeiten Sie `/etc/yum.repos.d/zabbix.repo` und setzen `enabled=1` für den Abschnitt `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
```

- Unter SUSE bearbeiten Sie `/etc/zypp/repos.d/zabbix.repo` und setzen `enabled=1` für den Abschnitt `zabbix-debuginfo`:

```
[zabbix-debuginfo]
name=Zabbix Official Repository debuginfo
type=rpm-md
baseurl=https://repo.zabbix.com/zabbix/8.0/unstable/sles/15/$basearch/debuginfo/
```

```
gpgcheck=1
gpgkey=https://repo.zabbix.com/zabbix/8.0/unstable/sles/15/$basearch/debuginfo/repodata/repomd.xml.key
enabled=0
update=1
```

Nach der Aktivierung installieren Sie die Pakete:

- Unter RHEL installieren Sie ein einzelnes Paket mit Debug-Informationen für alle Zabbix-Komponenten:

```
dnf install zabbix-debuginfo
```

- Unter SUSE installieren Sie komponentenspezifische debuginfo-Pakete:

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

Aktivieren instabiler Release-Repositories

Die folgenden Anweisungen dienen zum Aktivieren der instabilen Zabbix-Release-Repositories (standardmäßig deaktiviert), die für Release Candidates kleinerer Zabbix-Versionen verwendet werden.

Installieren oder aktualisieren Sie zunächst das neueste zabbix-release-Paket. Um rc-Pakete auf Ihrem System zu aktivieren, gehen Sie wie folgt vor:

Red Hat Enterprise Linux

Öffnen Sie die Datei `/etc/yum.repos.d/zabbix.repo` und setzen Sie `enabled=1` für das Repository `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

Öffnen Sie `/etc/apt/sources.list.d/zabbix.list` und entfernen Sie die Auskommentierung von „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

Öffnen Sie die Datei `/etc/zypp/repos.d/zabbix.repo` und setzen Sie `enabled=1` für das Repo `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 des Windows-Agenten aus MSI

Überblick

Der Zabbix Agent kann unter Windows mit 32-Bit- oder 64-Bit-MSI-Installer-Paketen installiert werden, die zum Download verfügbar sind.

Die Mindestanforderungen an das Betriebssystem für die MSI-Installation sind:

- **Für Zabbix Agent:** Windows XP (64-Bit) oder Windows Server 2003
- **Für Zabbix Agent 2:** Windows 10 (32-Bit) oder Windows Server 2016

32-Bit-Pakete können nicht auf 64-Bit-Systemen installiert werden.

Die Pakete enthalten:

- TLS-Unterstützung (die TLS-Konfiguration ist optional)
- Die Dienstprogramme Zabbix get und Zabbix sender (können zusammen mit Zabbix Agent/Agent 2 oder separat installiert werden)

Attention:
Zabbix Agent 2-Pakete enthalten keine ladbaren Plugins (MongoDB, PostgreSQL, MSSQL); diese müssen separat heruntergeladen und installiert werden.

Die Installation kann mit dem Setup-Assistenten oder über die Befehlszeile durchgeführt werden.

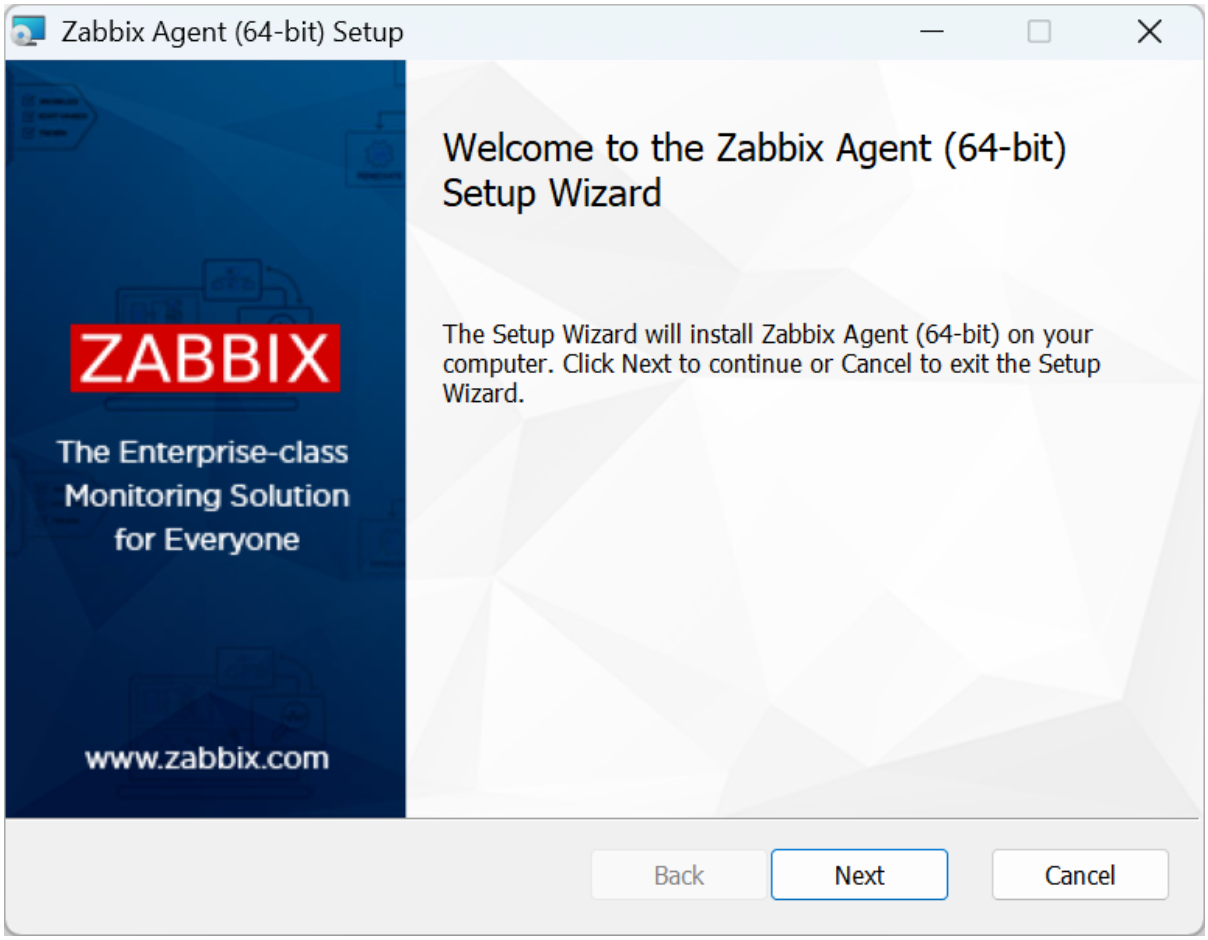
Obwohl die Installation mit MSI-Paketen vollständig unterstützt wird, wird für eine ordnungsgemäße Fehlerbehandlung empfohlen, mindestens Microsoft .NET Framework 2 zu installieren.

Attention:
Es wird empfohlen, den vom Installer vorgegebenen Standard-Installationspfad zu verwenden. Die Verwendung eines benutzerdefinierten Pfads ohne die erforderlichen Berechtigungen kann die Sicherheit der Installation beeinträchtigen.

Installation über den Setup-Assistenten

Die folgenden Installationsschritte gelten sowohl für den Zabbix Agent als auch für Zabbix Agent 2.

1. Doppelklicken Sie auf die heruntergeladene MSI-Datei, um die Installation zu starten:

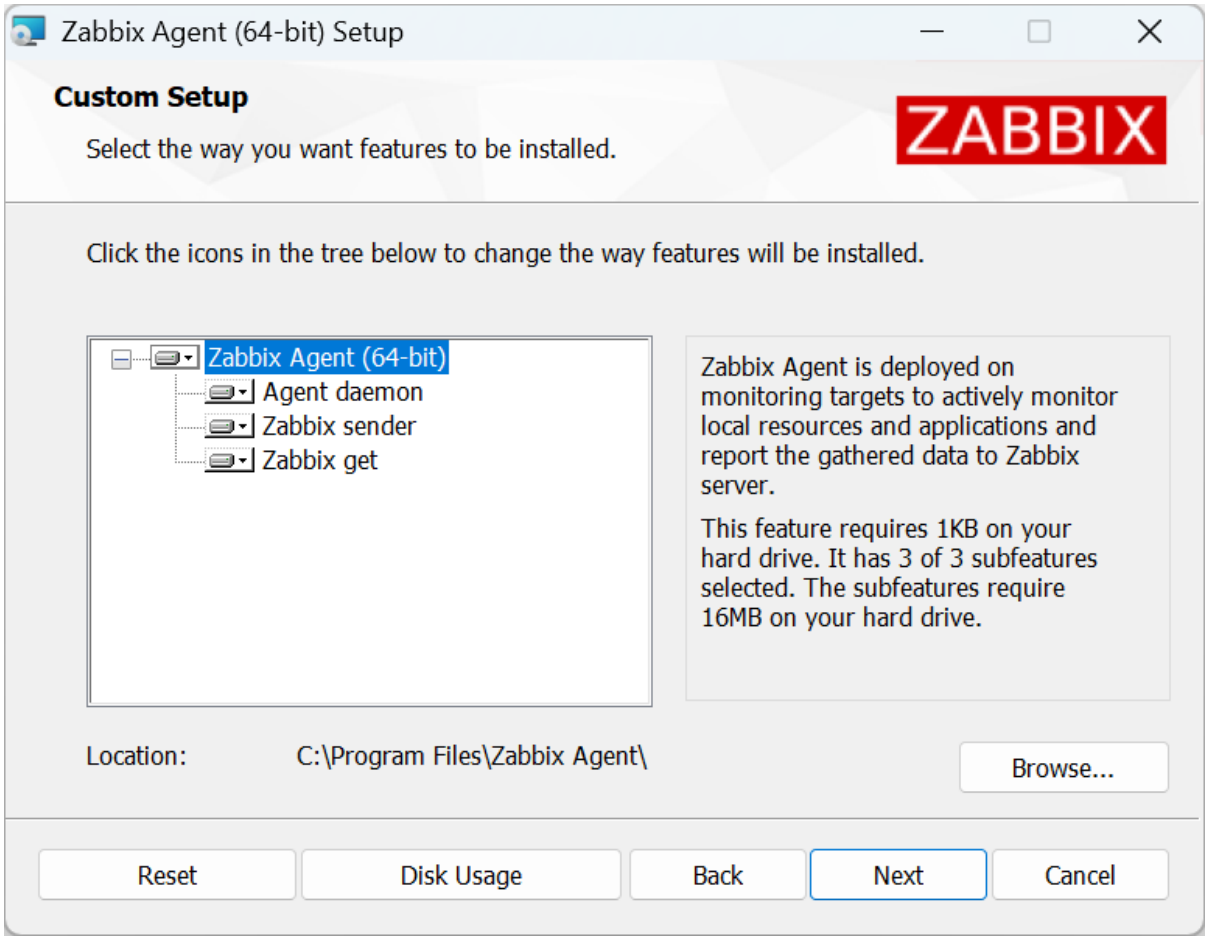


2. Akzeptieren Sie die Endbenutzer-Lizenzvereinbarung:



3. Wählen Sie die zu installierenden Zabbix-Komponenten (Agent daemon, Zabbix sender, Zabbix get) aus:

Attention:
Es wird empfohlen, den vom Installationsprogramm vorgegebenen Standardspeicherort zu verwenden. Die Verwendung eines benutzerdefinierten Speicherorts ohne die erforderlichen Berechtigungen kann die Sicherheit der Installation beeinträchtigen.

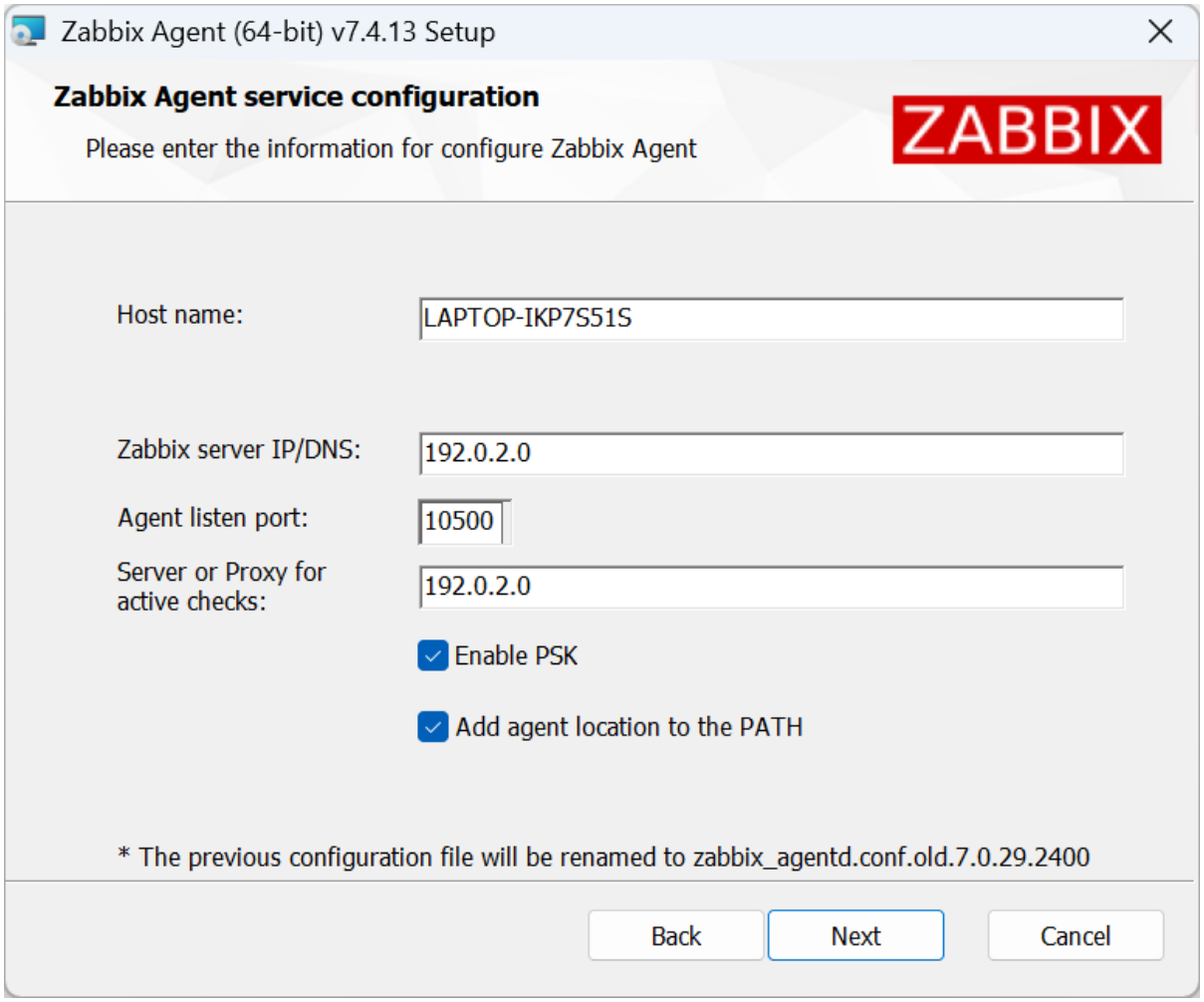


Note:
Wenn ein benutzerdefinierter Speicherort außerhalb des Verzeichnisses `Program Files` verwendet wird und der Zielordner bereits vorhanden ist, ändert das Installationsprogramm dessen Berechtigungen nicht. Bei einer Neuinstallation werden Sie aufgefordert, entweder die entsprechenden Berechtigungen manuell festzulegen oder ohne deren Änderung fortzufahren.

4. Konfigurieren Sie die folgenden Parameter. Ihre Werte werden in der Konfigurationsdatei des Zabbix Agent festgelegt:

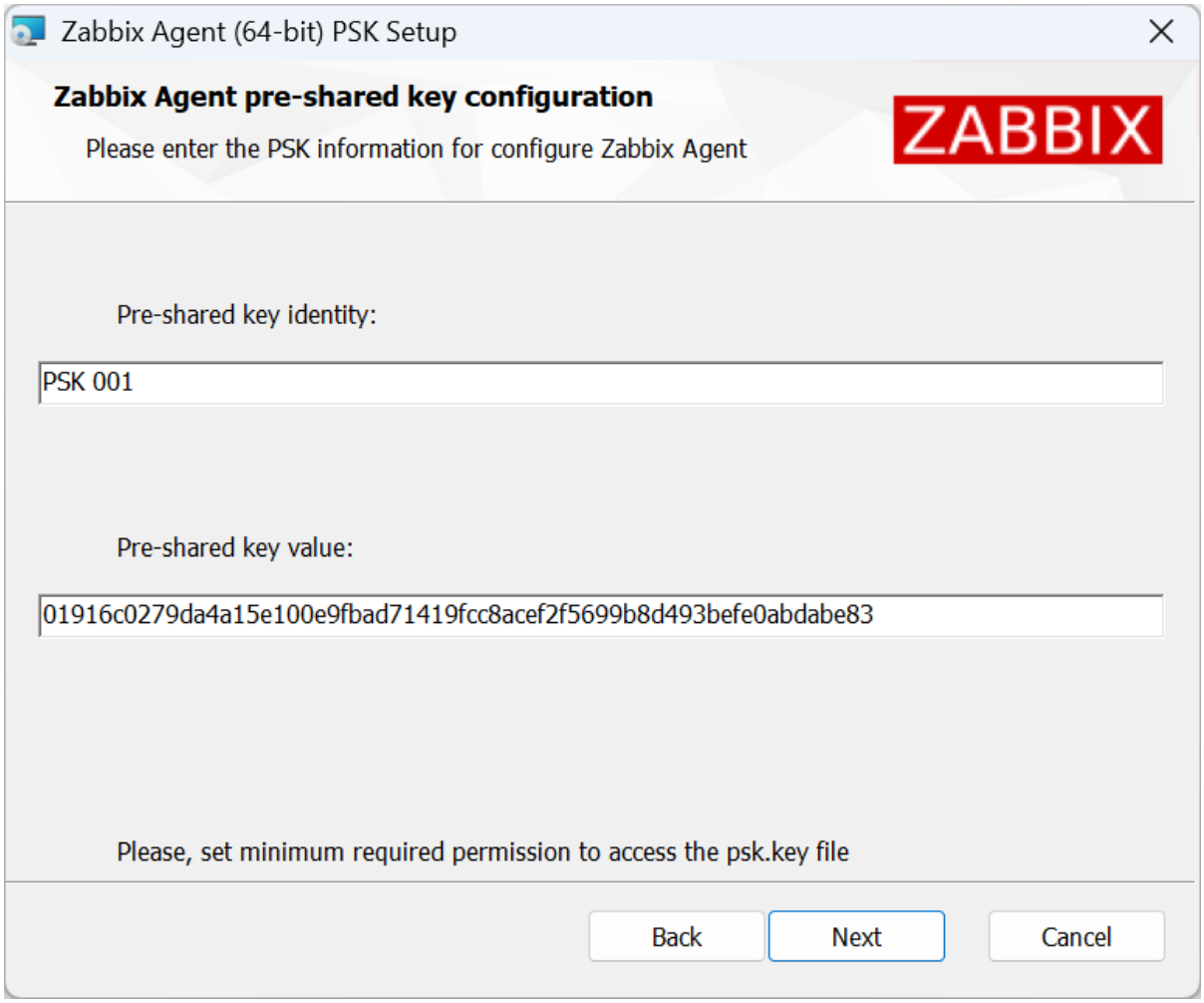
Parameter	Beschreibung
Host name	Der Hostname des Computers, auf dem der Zabbix Agent installiert wird. Legt den Parameter Hostname fest.
Zabbix server IP/DNS	Eine durch Kommas getrennte Liste von IP-Adressen, optional in CIDR-Notation, oder DNS-Namen von Zabbix-Servern oder Zabbix-Proxys. Dieser Parameter ist **obligatorisch**. Legt den Parameter Server fest.
Agent listen port	Der Agent wartet an diesem Port auf Verbindungen vom Server. Legt den Parameter ListenPort fest.
Server or Proxy for active checks	Die Adresse des Zabbix-Servers/Proxys oder die Clusterkonfiguration, von der aktive Prüfungen abgerufen werden. Die Adresse des Servers/Proxys ist eine IP-Adresse oder ein DNS-Name mit einem optionalen, durch einen Doppelpunkt getrennten Port. Legt den Parameter ServerActive fest.
Enable PSK	Aktivieren Sie das Kontrollkästchen, um TLS-Unterstützung mithilfe vorab geteilter Schlüssel zu aktivieren. Legt die Parameter TLSConnect und TLSAccept auf `psk` fest.
Add agent location to the PATH	Aktivieren Sie das Kontrollkästchen, um den Speicherort des Zabbix Agent zur Systemvariablen PATH hinzuzufügen.

Note:
Wenn ein vorhandener Zabbix Agent erkannt wird, werden die Parameterwerte aus dessen Konfigurationsdatei angezeigt. Zusätzlich wird die vorhandene Konfigurationsdatei während der Installation umbenannt und eine neue Konfigurationsdatei erstellt.



5. Konfigurieren Sie die PSK-Parameter, wenn Sie im vorherigen Schritt das Kontrollkästchen *Enable PSK* aktiviert haben. Diese Parameter werden ebenfalls in der Konfigurationsdatei des Zabbix Agent festgelegt:

Parameter	Beschreibung
Pre-shared key identity	Die Identitätszeichenfolge des vorab geteilten Schlüssels. Legt den Parameter TLSPSKIdentity fest.
Pre-shared key value	Der Zeichenfolgenwert des vorab geteilten Schlüssels. Erstellt die Datei `psk.key`, die den Schlüssel enthält, und legt den Parameter TLSPSKFile auf den Speicherort des Schlüssels fest (Standard: `C:\Program Files\Zabbix Agent\psk.key`). Es wird **empfohlen**, den Zugriff auf die Datei mit dem vorab geteilten Schlüssel einzuschränken, indem Sie deren Sicherheitseinstellungen so anpassen, dass nur der Zabbix Agent (oder der Benutzer, unter dem der Agent ausgeführt wird) sie lesen kann.

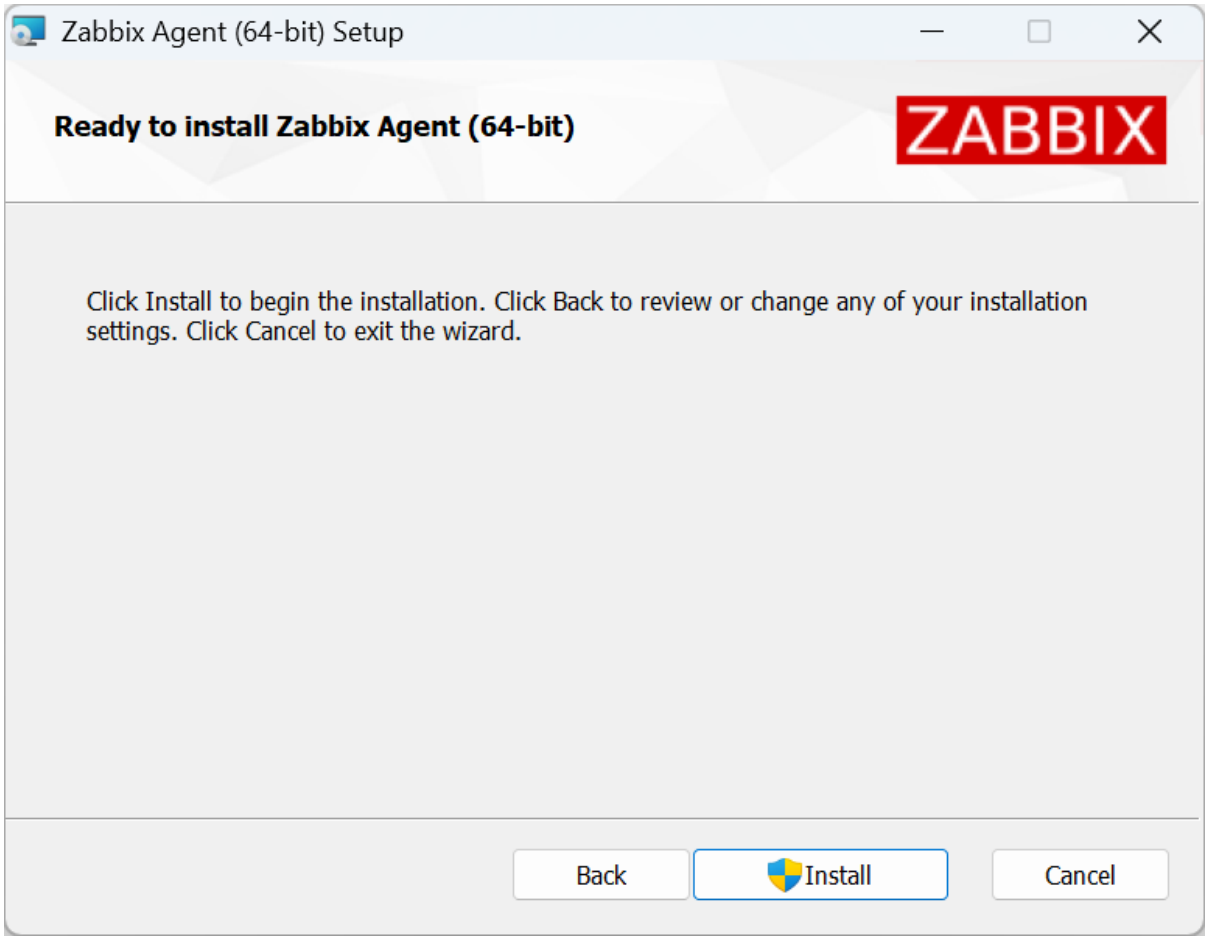


6. Klicken Sie auf *Install*, um die Installation zu starten.

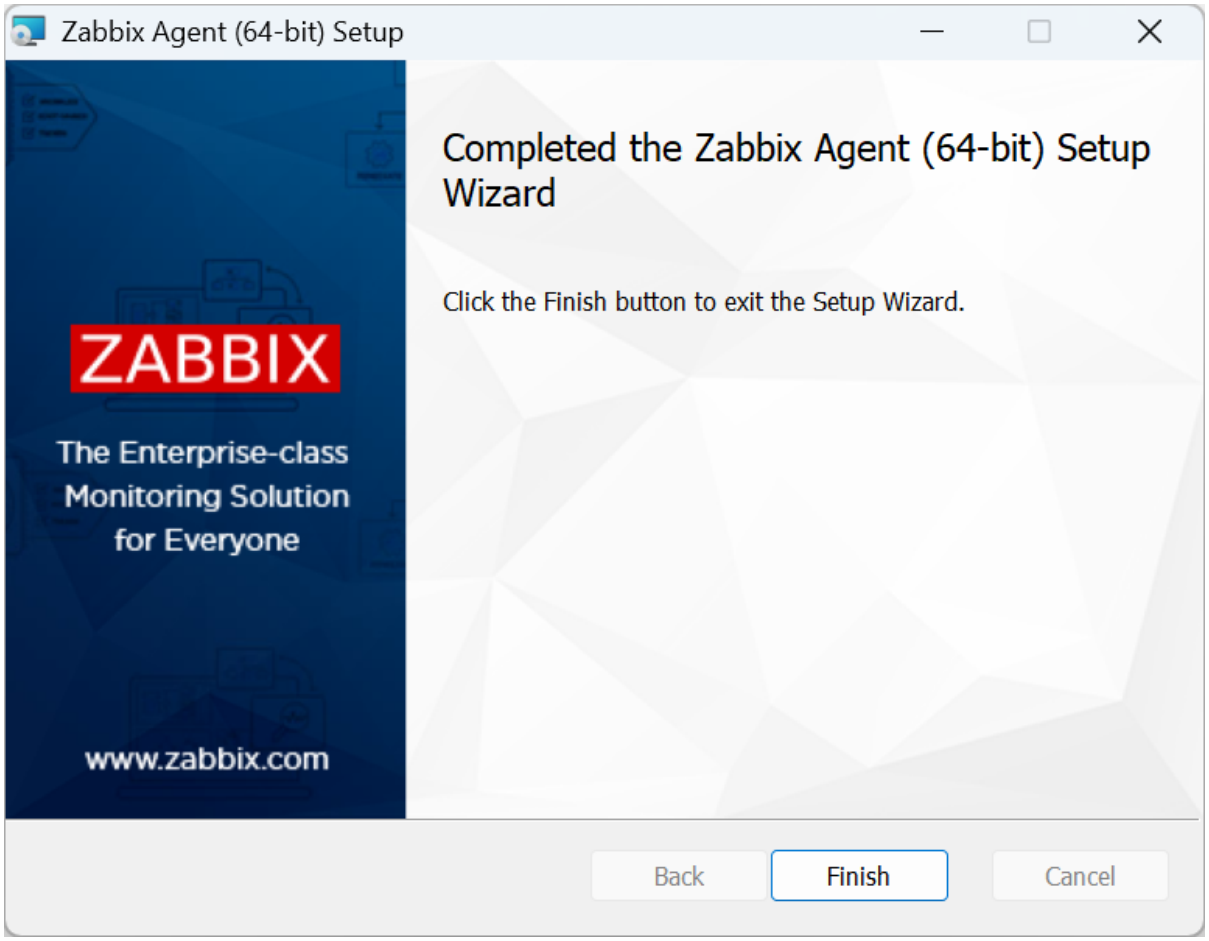
Alle ausgewählten Zabbix-Komponenten und die Konfigurationsdatei des Zabbix Agent werden am angegebenen Speicherort installiert (Standard: `C:\Program Files\Zabbix Agent`). Dasselbe gilt für Zabbix Agent 2, mit der Ausnahme, dass zusätzliche Konfigurationsdateien für dessen integrierte Plugins im Unterordner `zabbix_agent2.d\plugins.d` installiert werden.

Zusätzlich wird `zabbix_agentd.exe` (oder `zabbix_agent2.exe`) als Windows-Dienst mit verzögertem automatischem Start eingerichtet (oder mit automatischem Start bei Windows-Versionen vor Windows Vista/Server 2008).

Wenn während der Installation eine andere Version des Zabbix Agent ausgeführt wird, werden Sie aufgefordert, entweder die Anwendung zu schließen und einen Neustart zu versuchen oder sie geöffnet zu lassen. In diesem Fall ist ein Neustart erforderlich.



7. Klicken Sie auf die Schaltfläche *Finish*, um den Setup-Assistenten zu beenden.



Installation über die Befehlszeile

Der Zabbix Agent kann über die Befehlszeile installiert werden, indem das MSI-Installationsprogramm mit `msiexec` ausgeführt wird. Zum Beispiel:

```
msiexec.exe /l*v "C:\package.log" /i "C:\zabbix_agent-8.0.0-windows-amd64-openssl.msi" /qn+ SERVER=192.0.2
```

Diese Methode ermöglicht unbeaufsichtigte Installationen und benutzerdefinierte Konfigurationen mithilfe von Parametern.

Unterstützte Parameter

Zabbix agent MSI-Installationspakete unterstützen die folgenden Parameter sowohl für Zabbix agent als auch für Zabbix agent 2.

Note:
Die Parameter von Zabbix agent/agent2 werden während der Installation in der Konfigurationsdatei festgelegt. Klicken Sie auf einen Parameternamen, um seine detaillierte Beschreibung und Konfigurationsbeispiele auf der Seite Zabbix agent (Windows) anzuzeigen. Für Zabbix agent 2 siehe die Seite Zabbix agent 2 (Windows).

Parameter	Beschreibung
ADDDEFAULT	Eine durch Kommas getrennte Liste von Komponenten, die in ihrer Standardkonfiguration installiert werden sollen. Weitere Informationen finden Sie unter ADDDEFAULT property. Mögliche Werte: `AgentProgram`, `GetProgram`, `SenderProgram`, `ALL` Beispiel: `ADDDEFAULT=AgentProgram,GetProgram`
ADDLOCAL	Eine durch Kommas getrennte Liste von Komponenten, die lokal installiert werden sollen. Weitere Informationen finden Sie unter ADDLOCAL property. Mögliche Werte: `AgentProgram`, `GetProgram`, `SenderProgram`, `ALL` Beispiel: `ADDLOCAL=AgentProgram,SenderProgram`
ALLOWDENYKEY	Eine durch Semikolons getrennte Liste von AllowKey- oder DenyKey-Parametern zum Einschränken von Zabbix agent-Prüfungen. Falls erforderlich, verwenden Sie einen Backslash, um das Trennzeichen zu maskieren (`\;`). Setzt die Parameter AllowKey und DenyKey in der Konfigurationsdatei des Agenten. Beispiel: `ALLOWDENYKEY="AllowKey=system.run[more C:\Windows\System32\drivers\etc\hosts\\; echo 'File read complete'];DenyKey=system.run[*]"`
CONF	Der vollständige Pfad zu einer Vorlage für die Konfigurationsdatei für Zabbix agent. Während der Installation wird diese Datei zur Konfigurationsdatei des Agenten. Die Datei muss mindestens die Parameter Server und LogFile enthalten. Beispiel: `CONF="C:\full\path\to\example.conf"`
DONOTSTART	Verwenden Sie `DONOTSTART=1`, um zu verhindern, dass das MSI-Installationsprogramm den Zabbix agent-Dienst startet.
ENABLEPATH	Verwenden Sie `ENABLEPATH=1`, um den Speicherort von Zabbix agent zur System-PATH-Variable hinzuzufügen.
ENABLEPERSISTENTBUFFER	Nur Zabbix agent 2. Aktiviert die Verwendung des lokalen persistenten Speichers für aktive Datenpunkte.
HOSTINTERFACE	Ein optionaler Parameter, der die Host-Schnittstelle definiert.
HOSTMETADATA	Ein optionaler Parameter, der die Host-Metadaten definiert.
HOSTMETADATAITEM	Ein optionaler Parameter, der einen Datenpunkt definiert, der zum Abrufen der Host-Metadaten verwendet wird.
HOSTNAME	Ein optionaler Parameter, der den Hostnamen definiert.
HOSTNAMEITEM	Ein optionaler Parameter, der einen Datenpunkt definiert, der zum Abrufen des Hostnamens verwendet wird.
INCLUDE	Eine durch Semikolons getrennte Liste einzelner Dateien oder aller Dateien in einem Verzeichnis, die in die Konfigurationsdatei von Zabbix agent aufgenommen werden sollen.
IGNOREALREADYEXISTSFOLDER	Verwenden Sie bei Neuinstallationen `IGNOREALREADYEXISTSFOLDER=1`, um die Installation in einen bereits vorhandenen Zielordner außerhalb des Verzeichnisses `Program Files` zu erlauben. Wenn dieser Parameter nicht angegeben wird, schlägt die Installation fehl, wenn ein solcher Zielordner bereits vorhanden ist.
INSTALLFOLDER	Der vollständige Pfad zu einem Ordner, in dem Zabbix-Komponenten und die Konfigurationsdatei von Zabbix agent installiert werden. Für Zabbix agent 2 werden zusätzliche Konfigurationsdateien für integrierte Plugins im Unterordner `zabbix_agent2.d\plugins.d` installiert. Beispiel: `INSTALLFOLDER="C:\Program Files\Zabbix Agent"`
LISTENIP	Eine durch Kommas getrennte Liste von IP-Adressen, auf denen der Agent lauschen soll.
LISTENPORT	Der Agent lauscht auf diesem Port auf Verbindungen vom Server.
LOGFILE	Der Name der Protokolldatei von Zabbix agent.
LOGTYPE	Der Typ der Protokollausgabe.

Parameter	Beschreibung
NONMSICONFNAME	Der vollständige Pfad zu einer benutzerdefinierten Konfigurationsdatei für Zabbix agent. Während der Installation werden alle gültigen Agent-Konfigurationsparameter, die in dieser Datei vorhanden sind (beschränkt auf die in dieser Tabelle aufgeführten), in die neu erstellte Agent-Konfigurationsdatei geschrieben. Die Datei muss mindestens den Parameter Server enthalten. Beispiel: `NONMSICONFNAME="C:\full\path\to\example.conf"`
PERSISTENTBUFFERFILE	Nur Zabbix agent 2. Die Datei, in der Zabbix agent 2 die SQLite-Datenbank speichern soll.
PERSISTENTBUFFERPERIOD	Nur Zabbix agent 2. Der Zeitraum, für den Daten gespeichert werden sollen, wenn keine Verbindung zum Server oder Proxy besteht.
SERVER	Eine durch Kommas getrennte Liste von IP-Adressen, optional in CIDR-Notation, oder DNS-Namen von Zabbix-Servern oder Zabbix-Proxys. Dieser Parameter ist **verpflichtend**, außer wenn STARTAGENTS auf 0 gesetzt ist.
SERVERACTIVE	Die Adresse des Zabbix-Servers/Proxys oder die Clusterkonfiguration, von der aktive Prüfungen abgerufen werden.
SKIP	Verwenden Sie `SKIP=fw`, um zu verhindern, dass das MSI-Installationsprogramm eine Windows-Firewall-Ausnahmeregel für Zabbix agent hinzufügt.
SOURCEIP	Die Quell-IP-Adresse für ausgehende Verbindungen zum Zabbix-Server oder Zabbix-Proxy oder für Verbindungen beim Ausführen bestimmter Datenpunkte (web.page.get, net.tcp.port usw.).
STARTAGENTS	Die Anzahl der vorab erzeugten Instanzen von zabbix_agentd, die passive Prüfungen verarbeiten. Wenn auf 0 gesetzt, sind passive Prüfungen deaktiviert und der Agent lauscht auf keinem TCP-Port.
STARTUPTYPE	Starttyp des Zabbix agent-Dienstes. Mögliche Werte: **automatic** - startet den Dienst automatisch beim Windows-Start; **delayed** - *(Standard)* verzögert den Start des Dienstes, nachdem die automatisch gestarteten Dienste ihren Start abgeschlossen haben (verfügbar unter Windows Vista/Server 2008 und neueren Versionen); **manual** - startet den Dienst manuell (durch einen Benutzer oder eine Anwendung); **disabled** - deaktiviert den Dienst, sodass er nicht durch einen Benutzer oder eine Anwendung gestartet werden kann. Beispiel: `STARTUPTYPE=disabled`
STATUSPORT	Nur Zabbix agent 2. Wenn gesetzt, lauscht der Agent auf diesem Port auf HTTP-Statusanfragen (http://localhost:<port>/status).
TIMEOUT	Gibt an, wie lange beim Aufbau der Verbindung und beim Datenaustausch mit dem Zabbix-Proxy oder Server gewartet werden soll (in Sekunden).
TLSACCEPT	Die eingehenden Verbindungen, die akzeptiert werden sollen (für passive Prüfungen verwendet). Wenn auf `psk` gesetzt, wird auch TLSCONNECT auf `psk` gesetzt (sofern nicht anders angegeben).
TLSCAFILE	Der vollständige Pfad zu einer Datei mit den Zertifikaten der obersten CA(s) zur Überprüfung des Zertifikats des Gegenübers.
TLSCERTFILE	Der vollständige Pfad zu einer Datei mit dem Zertifikat oder der Zertifikatskette des Agenten.
TLSCONNECT	Wie der Agent eine Verbindung zum Zabbix-Server oder Proxy herstellen soll (für aktive Prüfungen verwendet). Wenn auf `psk` gesetzt, wird auch TLSACCEPT auf `psk` gesetzt (sofern nicht anders angegeben).
TLSCRLFILE	Der vollständige Pfad zu einer Datei mit widerrufenen Zertifikaten.
TLSKEYFILE	Der vollständige Pfad zu einer Datei mit dem privaten Schlüssel von Zabbix agent.
TLSPSKFILE	Der vollständige Pfad zu einer Datei mit dem vorab gemeinsam genutzten Schlüssel von Zabbix agent. Wenn sowohl TLSPSKFILE als auch TLSPSKVALUE gesetzt sind, wird der Wert von TLSPSKVALUE in die in TLSPSKFILE angegebene Datei geschrieben. Es wird **empfohlen**, den Zugriff auf die Datei mit dem vorab gemeinsam genutzten Schlüssel durch Anpassung der Dateisicherheit so einzuschränken, dass nur Zabbix agent (oder der Benutzer, unter dem der Agent ausgeführt wird) sie lesen kann.
TLSPSKIDENTITY	Die Identitätszeichenfolge des vorab gemeinsam genutzten Schlüssels.
TLSPSKVALUE	Der Zeichenfolgenwert des vorab gemeinsam genutzten Schlüssels. Wenn sowohl TLSPSKFILE als auch TLSPSKVALUE gesetzt sind, wird der Wert von TLSPSKVALUE in die in TLSPSKFILE angegebene Datei geschrieben. Beispiel: `TLSPSKVALUE=1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952`
TLSSERVERCERTISSUER	Der zulässige Aussteller des Server-(Proxy-)Zertifikats.
TLSSERVERCERTSUBJECT	Der zulässige Betreff des Server-(Proxy-)Zertifikats.
UNSAFEUSERPARAMETERS	Erlaubt, dass alle Zeichen in Argumenten an benutzerdefinierte Parameter übergeben werden.

Beispiele

Das folgende Beispiel installiert den Zabbix Agent mit benutzerdefinierter Konfiguration. Außerdem wird die TLS-Unterstützung mithilfe vorab gemeinsam genutzter Schlüssel aktiviert.

```
mkdir "C:\Program Files\Zabbix Agent" 2>nul
msiexec.exe /l*v "C:\package.log" /i "C:\zabbix_agent-8.0.0-windows-amd64-openssl.msi" /qn+^
 SERVER=192.0.2.1^
 INSTALLFOLDER="C:\Program Files\Zabbix Agent"^
 HOSTNAME=LAPTOP-IKP7S51S^
 TLSACCEPT=psk^
 TLSCONNECT=psk^
 TLSPSKIDENTITY="PSK 001"^
 TLSPSKFILE="C:\Program Files\Zabbix Agent\psk.key"^
 TLSPSKVALUE=1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952^
 ENABLEPATH=1^
 ALLOWDENYKEY="AllowKey=system.run[type C:\Windows\System32\drivers\etc\hosts];DenyKey=system.run[*]"
```

Das nächste Beispiel installiert eine neuere Version des Zabbix Agent und verwendet eine **Vorlage für die Konfigurationsdatei** (`CONF="C:\agent-template.conf"`). Während der Installation wird diese Datei zur Konfigurationsdatei des Agent. Um Parameter aus der alten Konfigurationsdatei zu übernehmen, verwenden Sie Parameterplatzhalter (z. B. `[AllowDenyKey]`).

```
msiexec.exe /l*v "C:\package.log" /i "C:\zabbix_agent-8.0.1-windows-amd64-openssl.msi" /qn+ NONMSICONFNAME

#### agent-template.conf example:
LogFile=[LogFile]
[AllowDenyKey]
Server=192.0.2.8
Hostname=DESKTOP-X9F4A2B
[Include]
[TLSConnect]
[TLSAccept]
[TLSPSKIdentity]
[TLSPSKFile]
```

Alternativ können Sie eine **benutzerdefinierte Konfigurationsdatei** (`NONMSICONFNAME="C:\agent-custom.conf"`) verwenden. Während der Installation werden alle gültigen Konfigurationsparameter des Agent, die in dieser Datei vorhanden sind (beschränkt auf die in der obigen Tabelle aufgeführten), in die neu erstellte Konfigurationsdatei des Agent geschrieben. Um die vorhandene Konfiguration des Agent beizubehalten, definieren Sie die zu erhaltenden Parameter.

```
msiexec.exe /l*v "C:\package.log" /i "C:\zabbix_agent-8.0.1-windows-amd64-openssl.msi" /qn+ NONMSICONFNAME

#### agent-custom.conf example:
Server=192.0.2.8
Hostname=DESKTOP-X9F4A2B
```

Zabbix Agent 2 ladbare Plugins

Ladbare Plugins für Zabbix Agent 2 können unter Windows mit 64-Bit-MSI-Installer-Paketen installiert werden, die hier heruntergeladen werden können.

Die Mindestanforderungen an das Betriebssystem für die MSI-Installation sind Windows 10 (64-Bit) oder Windows Server 2016.

> **Attention:**
> Bevor Sie ein Plugin installieren, prüfen Sie bitte dessen README-Datei. Sie kann spezifische Anforderungen und Installationsanweisungen enthalten.

Ähnlich wie bei Zabbix Agent/agent2 können ladbare Plugins mit dem Setup-Assistenten oder über die Befehlszeile installiert werden.

Installation über den Setup-Assistenten

1. Doppelklicken Sie auf die heruntergeladene MSI-Datei, um die Installation zu starten.
2. Akzeptieren Sie die Endbenutzer-Lizenzvereinbarung.
3. Wählen Sie die zu installierenden ladbaren Plugins für Zabbix Agent 2 aus.

Attention:
Es wird empfohlen, den vom Installationsprogramm vorgegebenen Standard-Installationspfad zu verwenden. Die Verwendung eines benutzerdefinierten Pfads ohne die erforderlichen Berechtigungen kann die Sicherheit der Installation beeinträchtigen.

4. Klicken Sie auf *Install*, um die Installation zu beginnen.

Alle ausgewählten ladbaren Plugins für Zabbix Agent 2 werden an dem von Ihnen angegebenen Speicherort installiert (Standard: `C:\Program Files\Zabbix Agent 2`), wobei ihre Konfigurationsdateien im Unterordner `zabbix_agent2.d` installiert werden.

5. Klicken Sie auf die Schaltfläche *Finish*, um den Setup-Assistenten zu beenden.

Installation über die Befehlszeile

Zabbix Agent 2 ladbare Plugins können über die Befehlszeile installiert werden, indem der MSI-Installer mit `msiexec` ausgeführt wird. Zum Beispiel:

```
msiexec.exe /l*v "C:\package.log" /i "C:\zabbix_agent2_plugins-8.0.0-windows-amd64.msi" /qn+
```

Die MSI-Installerpakete für ladbare Zabbix Agent 2-Plugins unterstützen die folgenden Parameter.

Parameter	Beschreibung
ADDDEFAULT	Eine durch Kommas getrennte Liste von Komponenten, die in ihrer Standardkonfiguration installiert werden. Weitere Informationen finden Sie unter ADDDEFAULT property. Mögliche Werte: `ALL`, `CephPlugin`, `EmberplusPlugin`, `MongodbPlugin`, `MssqlPlugin`, `NvidiagpuPlugin`, `PostgresqlPlugin` Beispiel: `ADDDEFAULT=MongodbPlugin,PostgresqlPlugin`
ADDLOCAL	Eine durch Kommas getrennte Liste von Komponenten, die lokal installiert werden. Weitere Informationen finden Sie unter ADDLOCAL property. Mögliche Werte: `ALL`, `CephPlugin`, `EmberplusPlugin`, `MongodbPlugin`, `MssqlPlugin`, `NvidiagpuPlugin`, `PostgresqlPlugin` Beispiel: `ADDLOCAL=MongodbPlugin,MssqlPlugin`
INSTALLFOLDER	Der vollständige Pfad zu einem Ordner, in dem Zabbix-Komponenten installiert werden; die Konfigurationsdateien werden im Unterordner `zabbix_agent2.d` installiert. Beispiel: `INSTALLFOLDER="C:\Program Files\Zabbix Agent 2"`

Erstellen von Zabbix Agent 2 unter Windows

Übersicht

Diese Seite zeigt, wie Sie Zabbix Agent 2 unter Windows 10 (64-Bit oder 32-Bit) aus den Quellen erstellen.

Sowohl 32-Bit- als auch 64-Bit-Versionen können auf einer 64-Bit-Plattform erstellt werden. Auf einer 32-Bit-Plattform kann jedoch nur die 32-Bit-Version erstellt werden.

Für die Erstellung von Zabbix Agent 2 benötigen Sie:

- MinGW-Buildtools
- Die Programmiersprache Go
- OpenSSL (für Verschlüsselungsfunktionen in Zabbix)
- PCRE2 (Perl-kompatible reguläre Ausdrücke; für Funktionen zum Abgleichen von Mustern mit regulären Ausdrücken in Zabbix)

Sie können Zabbix Agent 2 mit einer der folgenden Methoden erstellen:

- Verwendung von vcpkg – ein automatisierter Ansatz, der die Abhängigkeitsverwaltung mithilfe eines C++-Paketmanagers vereinfacht.
- Manuelle Erstellung – ein manueller Ansatz, bei dem alle Abhängigkeiten vor dem Kompilieren des Agent installiert werden müssen.

Attention:
Bevor Sie mit dem Erstellungsprozess beginnen, beachten Sie bitte Folgendes:

- Verwenden Sie zum Ausführen von Befehlen die Eingabeaufforderung, die von einem Benutzer mit ausreichenden Berechtigungen zum Schreiben in geschützte Ordner gestartet wurde. Verwenden Sie jedoch bei der Installation von OpenSSL und PCRE2 das MSYS2-Terminal.
- Es wird empfohlen, für alle Quelldateien und Build-Ordner ein Arbeitsverzeichnis unter `C:\Zabbix` zu erstellen. Kompilierte Komponenten sollten jedoch unter `C:\Zabbix\x64` (oder bei 32-Bit-Builds unter `C:\Zabbix\x86`) installiert werden.

Zabbix Agent 2 mit vcpkg erstellen

Dieser Abschnitt enthält Anweisungen zum Erstellen des Zabbix Agent mit vcpkg, einem Paketmanager, der die Verwaltung von Abhängigkeiten und die Integration in C++-Projekte vereinfacht.

1. Laden Sie die Build Tools for Visual Studio 2022 herunter und installieren Sie sie. Wählen Sie während der Installation unbedingt die Workload *Desktop development with C++* aus, die den vcpkg-Paketmanager enthält.

2. Laden Sie Go herunter und installieren Sie es (als MSI-Installer verfügbar). Geben Sie während der Installation `C:\Zabbix\Go` als Installationsordner an.

3. Laden Sie die MinGW-Distribution herunter, die die Microsoft Visual C-Laufzeitbibliothek verwendet, zum Beispiel:

   - Für 64-Bit-Builds: `x86_64-15.1.0-release-win32-seh-msvcrt-rt_v12-rev0.7z`
   - Für 32-Bit-Builds: `i686-15.1.0-release-win32-dwarf-msvcrt-rt_v12-rev0.7z`

Attention:
Die MinGW-Version muss die Mindestanforderungen erfüllen.

Entpacken Sie sie anschließend nach `C:\Zabbix\mingw64` (oder für 32-Bit-Builds nach `C:\Zabbix\mingw32`).

4. Initialisieren Sie vcpkg und installieren Sie die zum Erstellen des Zabbix Agent 2 erforderlichen Abhängigkeiten (beachten Sie, dass dies einige Zeit dauern kann):

```
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

#### Für 64-Bit-Builds:
set PATH=C:\Zabbix\mingw64\bin;%PATH%
vcpkg install --triplet x64-mingw-static --x-install-root=x64

#### Für 32-Bit-Builds:
set PATH=C:\Zabbix\mingw32\bin;%PATH%
vcpkg install --triplet x86-mingw-static --x-install-root=x86
```

5. Laden Sie das Zabbix-Quellarchiv herunter und entpacken Sie es nach `C:\Zabbix\zabbix-8.0.0`.

6. Wechseln Sie in das Zabbix-Build-Verzeichnis (`C:\Zabbix\zabbix-8.0.0\build\mingw`) und erstellen Sie das folgende `build.bat`-Skript:

   - Für 64-Bit-Builds:

```
:: MinGW und Go für die aktuelle Sitzung zur Systemvariablen `PATH` hinzufügen:
set PATH=C:\Zabbix\mingw64\bin;%PATH%
set PATH=C:\Zabbix\Go\bin;%PATH%

:: vcpkg-Installationspfad festlegen:
set vcpkg="C:\Zabbix\x64\x64-mingw-static"

:: Linker-Flags für die Crypt32-Bibliothek festlegen:
SET CGO_LDFLAGS="-lCrypt32"

:: Build-Prozess ausführen:
```

```
mingw32-make GOFLAGS="-buildvcs=false" ARCH=AMD64 ^
    PCRE2="%vcpkg%" ^
    OPENSSL="%vcpkg%" ^
    all
```

- Für 32-Bit-Builds:

```
:: MinGW und Go für die aktuelle Sitzung zur Systemvariablen `PATH` hinzufügen:
set PATH=C:\Zabbix\mingw32\bin;%PATH%
set PATH=C:\Zabbix\Go\bin;%PATH%

:: vcpkg-Installationspfad festlegen:
set vcpkg="C:\Zabbix\x86\x86-mingw-static"

:: Linker-Flags für die Crypt32-Bibliothek festlegen:
SET CGO_LDFLAGS="-lCrypt32"

:: Build-Prozess ausführen:
mingw32-make GOFLAGS="-buildvcs=false" ARCH=x86 ^
    PCRE2="%vcpkg%" ^
    OPENSSL="%vcpkg%" ^
    all
```

7. Kompilieren Sie den Zabbix Agent 2, indem Sie das Skript ausführen:

```
build.bat
```

Nach der Kompilierung befindet sich die Binärdatei des Zabbix Agent 2 unter `C:\Zabbix\zabbix-8.0.0\bin\win64` (für 64-Bit-Builds) oder unter `C:\Zabbix\zabbix-8.0.0\bin\win32` (für 32-Bit-Builds). Die Konfigurationsdateien des Zabbix Agent 2 befinden sich unter `C:\Zabbix\zabbix-8.0.0\src\go\conf`.

Um den Agent auszuführen, kopieren Sie `zabbix_agent2.exe` und die zugehörigen Konfigurationsdateien in einen dedizierten Ordner (z. B. `C:\Zabbix\agent2`) und starten Sie anschließend den Agent:

```
mkdir C:\Zabbix\agent2

#### Für 64-Bit-Builds:
copy C:\Zabbix\zabbix-8.0.0\bin\win64\zabbix_agent2.exe C:\Zabbix\agent2\

#### Für 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.conf C:\Zabbix\agent2\
xcopy /E /I C:\Zabbix\zabbix-8.0.0\src\go\conf\zabbix_agent2.d C:\Zabbix\agent2\zabbix_agent2.d\

C:\Zabbix\agent2\zabbix_agent2.exe -c C:\Zabbix\agent2\zabbix_agent2.conf
```

Falls erforderlich, fahren Sie mit der Kompilierung der ladbaren Plugins für den Zabbix Agent 2 fort.

Kompilieren von ladbaren Zabbix Agent 2-Plugins

1. Laden Sie den Zabbix-Plugin-Quellcode herunter, der zu Ihrer Zabbix Agent 2-Version passt (z. B. `zabbix-agent2-plugin-postgresql-8` und entpacken Sie ihn nach `C:\Zabbix`.

> **Attention:**
> Bevor Sie ein Plugin installieren, prüfen Sie bitte die README-Datei. Sie kann spezifische Anforderungen und Installationsanweisungen enthalten.

2. Wechseln Sie in das entpackte Plugin-Verzeichnis und kompilieren Sie das Plugin:

```
cd C:\Zabbix\zabbix-agent2-plugin-ember-plus-8.0.0

#### Für 64-Bit-Builds:
mingw32-make ARCH=AMD64

#### Für 32-Bit-Builds:
mingw32-make ARCH=x86
```

Nach der Kompilierung befinden sich die Binärdatei des Zabbix Agent 2-Plugins und die zugehörige Konfigurationsdatei im selben Plugin-Verzeichnis.

Die ausführbare Plugin-Datei kann sich an beliebiger Stelle befinden, solange sie von Zabbix Agent 2 geladen werden kann. Geben Sie den Pfad zur Plugin-Binärdatei in der Plugin-Konfigurationsdatei an, z. B. in `postgresql.conf` für das PostgreSQL-Plugin:

```
Plugins.PostgreSQL.System.Path=/path/to/executable/zabbix-agent2-plugin-postgresql
```

Der Pfad zur Plugin-Konfigurationsdatei muss im Parameter Include der Zabbix Agent 2-Konfigurationsdatei angegeben werden:

```
Include=/path/to/plugin/configuration/file/postgresql.conf
```

Weitere Informationen zur Konfiguration von Plugins finden Sie unter Einrichtung.

Zabbix Agent 2 manuell erstellen

Attention:
Diese Methode zum Erstellen von Zabbix Agent 2 eignet sich für Benutzer, die die vollständige Kontrolle über die Build-Umgebung benötigen oder sich in einer eingeschränkten Umgebung befinden, in der die Verwendung von vcpkg nicht möglich ist.

Dieser Abschnitt enthält Anweisungen zum manuellen Erstellen von Zabbix Agent 2. Dazu gehören die Installation der erforderlichen Build-Tools und Abhängigkeiten sowie die anschließende Kompilierung des Agenten.

Einrichtung der Build-Tools

1. Laden Sie MSYS2 herunter und installieren Sie es (als MSI-Installer verfügbar). Stellen Sie während der Installation sicher, dass Sie `C:\Zabbix\msys64` als Installationsordner angeben.

2. Laden Sie Go herunter und installieren Sie es (als MSI-Installer verfügbar; siehe die derzeit unterstützten Go-Versionen). Stellen Sie während der Installation sicher, dass Sie `C:\Zabbix\Go` als Installationsordner angeben.

3. Laden Sie die MinGW-Distribution herunter, die die Microsoft Visual C-Laufzeitbibliothek verwendet; zum Beispiel:

- Für 64-Bit-Builds: `x86_64-15.1.0-release-win32-seh-msvcrt-rt_v12-rev0.7z`
- Für 32-Bit-Builds: `i686-15.1.0-release-win32-dwarf-msvcrt-rt_v12-rev0.7z`

Attention:
Die MinGW-Version muss die Mindestanforderungen erfüllen.

Entpacken Sie sie anschließend nach `C:\Zabbix\mingw64` (oder für 32-Bit-Builds nach `C:\Zabbix\mingw32`).

OpenSSL installieren

Note:
Um den Zabbix Agent ohne TLS-Unterstützung zu kompilieren, fahren Sie mit dem Abschnitt PCRE2 installieren fort.

1. Öffnen Sie das MSYS2-MSYS-Terminal mit Administratorrechten und führen Sie die folgenden Befehle aus:

```
pacman -S perl-Locale-Maketext-Simple
pacman -S nasm
pacman -S make
pacman -S cmake
```

2. Laden Sie das OpenSSL-Quellarchiv herunter und extrahieren Sie es nach `C:\Zabbix\openssl-3.5.0`.

3. Wechseln Sie in das extrahierte OpenSSL-Verzeichnis und erstellen Sie das folgende `build.sh`-Skript:

- Für 64-Bit-Builds:

```
####!/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
```

- Für 32-Bit-Builds:

```
####!/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:
Stellen Sie sicher, dass Nicht-Administratoren der Schreibzugriff auf das Verzeichnis `C:\Zabbix\x64\OpenSSL-Win64-350-static` entzogen wird. Andernfalls lädt der Agent SSL-Einstellungen aus einem Pfad, der von nicht privilegierten Benutzern geändert werden kann, was eine potenzielle Sicherheitslücke darstellt.

- Die Option `no-shared` sorgt dafür, dass libcrypto.lib und libssl.lib eigenständige statische OpenSSL-Bibliotheken sind. Dadurch enthalten Zabbix-Binärdateien OpenSSL, ohne dass externe DLLs erforderlich sind. Das bedeutet, dass Zabbix-Binärdateien auf andere Windows-Computer kopiert werden können, ohne OpenSSL-Bibliotheken mitzukopieren. Wenn jedoch eine neue OpenSSL-Fehlerbehebungsversion veröffentlicht wird, muss der Zabbix Agent neu kompiliert werden.
- Ohne die Option `no-shared` ist Zabbix zur Laufzeit auf OpenSSL-DLLs angewiesen. Das bedeutet, dass OpenSSL-Updates möglicherweise keine Neukompilierung des Zabbix Agent erfordern. Beim Kopieren auf andere Computer müssen jedoch auch die OpenSSL-DLLs kopiert werden.

Weitere Informationen zu anderen OpenSSL-Konfigurationsoptionen finden Sie in der OpenSSL-Dokumentation.

4. Konfigurieren und installieren Sie OpenSSL, indem Sie das Skript ausführen (beachten Sie, dass dies einige Zeit dauern kann):

```
cd /c/Zabbix/openssl-3.5.0
./build.sh
```

Installation von PCRE2

1. Laden Sie das PCRE2-Quellarchiv herunter und entpacken Sie es nach `C:\Zabbix\pcre2-10.45`.

2. Öffnen Sie das MSYS2-MSYS-Terminal mit Administratorrechten. Erstellen Sie dann im entpackten PCRE2-Verzeichnis ein Verzeichnis `build` und wechseln Sie dorthin:

```
mkdir /c/Zabbix/pcre2-10.45/build
cd /c/Zabbix/pcre2-10.45/build
```

3. Konfigurieren Sie PCRE2:

```
#### Für 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" ..

#### Für 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:
Falls Fehler auftreten, wird empfohlen, den CMake-Cache zu löschen, bevor Sie versuchen, den CMake-Build-Prozess zu wiederholen. Der Cache (`CMakeCachecache.txt`) befindet sich möglicherweise im Build-Verzeichnis des entpackten PCRE2-Verzeichnisses.

4. Installieren Sie PCRE2:

```
make install
```

Kompilieren von Zabbix Agent 2

1. Laden Sie das Zabbix-Quellarchiv herunter und extrahieren Sie es nach `C:\Zabbix\zabbix-8.0.0`.

Wenn Sie ein Quellarchiv aus dem unveränderten Quellcode-Repository erstellen müssen (z. B. um benutzerdefinierte Patches anzuwenden oder aus dem neuesten Quellcode zu erstellen), führen Sie die folgenden Befehle auf einem **Linux**-Computer mit installiertem Go aus (erforderlich für die Konfiguration von Zabbix Agent 2):

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
cd zabbix
./bootstrap.sh
./configure
make dist
```

Dadurch wird ein Quellarchiv erstellt, das anschließend auf einen Windows-Computer kopiert werden kann.

2. Öffnen Sie die Eingabeaufforderung mit Administratorrechten. Navigieren Sie anschließend zum Zabbix-Build-Verzeichnis und kompilieren Sie Zabbix Agent. Stellen Sie sicher, dass Sie die Verzeichnisse, in denen OpenSSL und PCRE2 installiert sind, korrekt angeben:

- Für 64-Bit-Builds:

```
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

#### Mit TLS-Unterstützung:
mingw32-make ARCH=AMD64 PCRE2="C:\Zabbix\x64\PCRE2" OPENSSL="C:\Zabbix\x64\OpenSSL-Win64-350-static"

#### Ohne TLS-Unterstützung:
mingw32-make ARCH=AMD64 PCRE2="C:\Zabbix\x64\PCRE2"
```

- Für 32-Bit-Builds:

```
cd C:\Zabbix\zabbix-8.0.0\build\mingw
set PATH=C:\Zabbix\mingw32\bin;%PATH%

#### Mit TLS-Unterstützung:
mingw32-make ARCH=x86 PCRE2="C:\Zabbix\x86\PCRE2" OPENSSL="C:\Zabbix\x86\OpenSSL-Win32-350-static"

#### Ohne TLS-Unterstützung:
mingw32-make ARCH=x86 PCRE2="C:\Zabbix\x86\PCRE2"
```

Nach der Kompilierung befindet sich die Binärdatei von Zabbix Agent 2 in `C:\Zabbix\zabbix-8.0.0\bin\win64` (oder bei 32-Bit-Builds in `C:\Zabbix\zabbix-8.0.0\bin\win32`). Die Konfigurationsdateien von Zabbix Agent 2 befinden sich in `C:\Zabbix\zabbix-8.0.0\src\go\conf`.

Um den Agent auszuführen, kopieren Sie die Binärdatei `zabbix_agent2.exe` und die zugehörigen Konfigurationsdateien in einen dedizierten Ordner (z. B. `C:\Zabbix\agent2`) und starten Sie anschließend den Agent:

```
mkdir C:\Zabbix\agent2
copy C:\Zabbix\zabbix-8.0.0\bin\win64\zabbix_agent2.exe C:\Zabbix\agent2\
copy C:\Zabbix\zabbix-8.0.0\src\go\conf\zabbix_agent2.conf C:\Zabbix\agent2\
xcopy /E /I C:\Zabbix\zabbix-8.0.0\src\go\conf\zabbix_agent2.d C:\Zabbix\agent2\zabbix_agent2.d\

C:\Zabbix\agent2\zabbix_agent2.exe -c C:\Zabbix\agent2\zabbix_agent2.conf
```

Fahren Sie bei Bedarf mit dem Kompilieren ladbarer Plugins für Zabbix Agent 2 fort.

Erstellen des Zabbix Agent unter Windows

Übersicht

Diese Seite zeigt, wie der Zabbix Agent unter Windows 10 (64-Bit) aus dem Quellcode erstellt wird.

Diese Anweisungen gelten für Windows-Versionen, die Visual Studio 2022 unterstützen.

Für die Erstellung des Zabbix Agent werden folgende Komponenten benötigt:

- C-Compiler (in den Build Tools für Visual Studio 2022 enthalten)
- OpenSSL (für Verschlüsselungsfunktionen in Zabbix)
- PCRE2 (Perl-kompatible reguläre Ausdrücke; für Funktionen zum Abgleichen von Mustern mit regulären Ausdrücken in Zabbix)

Sie können den Zabbix Agent mit einer der folgenden Methoden erstellen:

- Verwendung von vcpkg – ein automatisierter Ansatz, der die Verwaltung von Abhängigkeiten mithilfe eines C++-Paketmanagers vereinfacht.

- Manuelle Erstellung – ein manueller Ansatz, bei dem alle Abhängigkeiten vor dem Kompilieren des Agent installiert werden müssen.

Je nach Ihren Überwachungsanforderungen können zusätzliche Bibliotheken erforderlich sein. Weitere Informationen finden Sie unter Anforderungen.

Attention:
Beachten Sie vor Beginn des Erstellungsprozesses Folgendes:

- Verwenden Sie zum Ausführen von Befehlen die „x64 Native Tools Command Prompt“ (in den Build Tools für Visual Studio 2022 enthalten), die von einem Benutzer mit ausreichenden Berechtigungen zum Schreiben in geschützte Ordner gestartet wurde.
- Es wird empfohlen, ein Arbeitsverzeichnis unter `C:\Zabbix` für alle Quelldateien und Build-Ordner zu erstellen. Die kompilierten Komponenten sollten jedoch in `C:\Program Files\Zabbix\x64` installiert werden.

Erstellen des Zabbix Agent mit vcpkg

Dieser Abschnitt enthält Anweisungen zum Erstellen des Zabbix Agent mit vcpkg, einem Paketmanager, der die Verwaltung von Abhängigkeiten und die Integration in C++-Projekte vereinfacht.

1. Laden Sie die Build Tools for Visual Studio 2022 herunter und installieren Sie sie. Stellen Sie während der Installation sicher, dass Sie die Workload *Desktop development with C++* auswählen. Sie enthält die für das Erstellen des Agent mit vcpkg erforderlichen Tools:

- C-Compiler (Microsoft Visual C++)
- Befehlszeilentool NMake
- Paketmanager vcpkg
- ”x64 Native Tools Command Prompt”

2. Öffnen Sie `cmd.exe` mit Administratorrechten, initialisieren Sie vcpkg und installieren Sie die für das Erstellen des Zabbix Agent erforderlichen Abhängigkeiten (beachten Sie, dass dies einige Zeit dauern kann):

```
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. Laden Sie das Zabbix-Quellarchiv herunter und extrahieren Sie es nach `C:\Zabbix\zabbix-8.0.0`.

4. Wechseln Sie in das Zabbix-Build-Verzeichnis (`C:\Zabbix\zabbix-8.0.0\build\win32\project`) und erstellen Sie das folgende `build.bat`-Skript. Stellen Sie sicher, dass Sie die Verzeichnisse, in denen OpenSSL und PCRE2 installiert sind, korrekt angeben:

```
:: 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. Öffnen Sie die ”x64 Native Tools Command Prompt” und kompilieren Sie den Zabbix Agent, indem Sie das Skript ausführen:

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

Wenn Sie eine andere Eingabeaufforderung geöffnet haben, initialisieren Sie zuerst die Visual-Studio-Buildumgebung:

```
"%ProgramFiles(x86)%\Microsoft Visual Studio\2022\BuildTools\VC\Auxiliary\Build\vcvarsall.bat" x64
cd C:\Zabbix\zabbix-8.0.0\build\win32\project
build.bat
```

Pfade, die Leerzeichen enthalten, müssen in Anführungszeichen eingeschlossen werden.

Nach der Kompilierung befinden sich die Binärdateien der Zabbix-Komponenten in `C:\Zabbix\zabbix-8.0.0\bin\win64`. Die Konfigurationsdatei des Zabbix Agent befindet sich in `C:\Zabbix\zabbix-8.0.0\conf`.

Um den Agent auszuführen, kopieren Sie `zabbix_agentd.exe` und die zugehörige Konfigurationsdatei in einen dedizierten Ordner (z. B. `C:\Zabbix\agent`) und starten Sie anschließend den Agent:

```
mkdir C:\Zabbix\agent
copy C:\Zabbix\zabbix-8.0.0\bin\win64\zabbix_agentd.exe C:\Zabbix\agent\
copy C:\Zabbix\zabbix-8.0.0\conf\zabbix_agentd.conf C:\Zabbix\agent\

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

Zabbix Agent manuell erstellen

Attention:
Diese Methode zum Erstellen des Zabbix Agent ist für Benutzer geeignet, die die vollständige Kontrolle über die Build-Umgebung benötigen oder sich in einer eingeschränkten Umgebung befinden, in der die Verwendung von vcpkg nicht möglich ist.

Dieser Abschnitt enthält Anweisungen zum manuellen Erstellen des Zabbix Agent. Dazu gehören die Installation der erforderlichen Build-Tools und Abhängigkeiten (Perl, OpenSSL, PCRE2) sowie die anschließende Kompilierung des Agent.

Buildtools installieren

1. Laden Sie Build Tools for Visual Studio 2022 herunter und installieren Sie sie. Wählen Sie während der Installation unbedingt die Workload *Desktopentwicklung mit C++* aus, die die für die manuelle Erstellung des Agent erforderlichen Tools enthält:

- C-Compiler (Microsoft Visual C++)
- NMake-Befehlszeilentool
- „x64 Native Tools Command Prompt“

OpenSSL installieren

Note:
Um den Zabbix Agent ohne TLS-Unterstützung zu kompilieren, fahren Sie mit dem Abschnitt PCRE2 installieren fort.

1. Laden Sie Strawberry Perl herunter und installieren Sie es (verfügbar als MSI-Installer). Geben Sie während der Installation `C:\Zabbix\Strawberry` als Installationsordner an.

2. Installieren Sie das Perl-Modul `Text::Template`:

```
cpanm Text::Template
```

3. Überprüfen Sie, ob der Netwide Assembler (NASM; für die Kompilierung von OpenSSL erforderlich) während der Installation von Strawberry Perl kompiliert wurde:

```
nasm -v
#### NASM version 2.16.01 compiled on May  3 2024
```

Wenn NASM nicht kompiliert wurde, installieren Sie es manuell. Weitere Informationen finden Sie in der NASM-Dokumentation.

4. Laden Sie das OpenSSL-Quellarchiv herunter und extrahieren Sie es nach `C:\Zabbix\openssl-3.5.0`.

5. Wechseln Sie in das extrahierte Verzeichnis und konfigurieren Sie OpenSSL, zum Beispiel:

```
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:
Wenn Sie beim Kompilieren des Zabbix Agent unter Windows ein benutzerdefiniertes Verzeichnis für OpenSSL auswählen (z. B. `C:\zabbix` oder `C:\openssl-64bit`), stellen Sie sicher, dass der Schreibzugriff für Benutzer ohne Administratorrechte auf dieses Verzeichnis entzogen wird. Andernfalls lädt der Agent SSL-Einstellungen aus einem Pfad, der von nicht privilegierten Benutzern geändert werden kann, was zu einer potenziellen Sicherheitslücke führen kann.

- Die Option `no-shared` sorgt dafür, dass libcrypto.lib und libssl.lib eigenständige statische OpenSSL-Bibliotheken sind. Dadurch enthalten Zabbix-Binärdateien OpenSSL, ohne dass externe DLLs erforderlich sind. Das bedeutet, dass Zabbix-Binärdateien auf andere Windows-Computer kopiert werden können, ohne OpenSSL-Bibliotheken zu benötigen. Wenn jedoch eine neue OpenSSL-Fehlerbehebungsversion veröffentlicht wird, muss der Zabbix Agent neu kompiliert werden.

- Ohne die Option `no-shared` ist Zabbix zur Laufzeit auf OpenSSL-DLLs angewiesen. Das bedeutet, dass OpenSSL-Updates möglicherweise keine Neukompilierung des Zabbix Agent erfordern. Beim Kopieren auf andere Computer müssen die OpenSSL-DLLs jedoch ebenfalls kopiert werden.

Weitere Informationen zu anderen OpenSSL-Konfigurationsoptionen finden Sie in der OpenSSL-Dokumentation.

6. Kompilieren Sie OpenSSL und führen Sie die Tests aus (beachten Sie, dass dies einige Zeit dauern kann):

> **Attention:**
> Führen Sie die Tests ohne Administratorrechte aus. Andernfalls kann dies zu unerwarteten Ergebnissen oder Sicherheitsrisiken führen. Wenn einige Tests fehlschlagen, finden Sie Informationen zur Fehlerbehebung in der OpenSSL-Dokumentation.

```
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. Installieren Sie OpenSSL:

```
nmake install
```

Um nur Softwarekomponenten (Bibliotheken und Header-Dateien, jedoch keine Dokumentation) zu installieren, können Sie `nmake install_sw` verwenden.

Installation von PCRE2

1. Laden Sie CMake herunter und installieren Sie es (als MSI-Installer verfügbar). Achten Sie während der Installation darauf, `C:\Zabbix\CMake` als Installationsordner anzugeben und die Option *Add CMake to the PATH environment variable* auszuwählen.

2. Laden Sie das PCRE2-Quellarchiv herunter und entpacken Sie es nach `C:\Zabbix\pcre2-10.45`.

3. Erstellen Sie im entpackten PCRE2-Verzeichnis ein `build`-Verzeichnis und wechseln Sie dorthin:

```
mkdir C:\Zabbix\pcre2-10.45\build
cd C:\Zabbix\pcre2-10.45\build
```

4. Konfigurieren Sie PCRE2:

```
cmake -G "NMake Makefiles" -DPCRE_SUPPORT_UNICODE_PROPERTIES=ON -DCMAKE_BUILD_TYPE=Release -DCMAKE_INSTALL
```

> **Note:**
> Falls Fehler auftreten, wird empfohlen, den CMake-Cache zu löschen, bevor Sie versuchen, den CMake-Build-Prozess zu wiederholen. Der Cache (`CMakeCachecache.txt`) befindet sich im `build`-Verzeichnis des entpackten PCRE2-Verzeichnisses.

5. Erstellen Sie PCRE2 mit NMake:

```
nmake
```

6. Installieren Sie PCRE2:

```
cmake --install .
```

Kompilieren des Zabbix Agent

1. Laden Sie das Zabbix-Quellarchiv herunter und extrahieren Sie es nach `C:\Zabbix\zabbix-8.0.0`.

Wenn Sie ein Quellarchiv aus dem unveränderten Quellcode-Repository erstellen müssen (z. B. um benutzerdefinierte Patches anzuwenden oder den neuesten Quellcode zu kompilieren), führen Sie die folgenden Befehle auf einem **Linux**-Computer aus:

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
cd zabbix
./bootstrap.sh
./configure
make dist
```

Dadurch wird ein Quellarchiv erstellt, das anschließend auf einen Windows-Computer kopiert werden kann.

2. Navigieren Sie zum Zabbix-Build-Verzeichnis, öffnen Sie die „x64 Native Tools Command Prompt“ und kompilieren Sie den Zabbix Agent (oder andere Komponenten). Stellen Sie sicher, dass Sie die Verzeichnisse, in denen OpenSSL und PCRE2 installiert sind, korrekt angeben:

```
"%ProgramFiles(x86)%\Microsoft Visual Studio\2022\BuildTools\VC\Auxiliary\Build\vcvarsall.bat" x64

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

#### Mit TLS-Unterstützung:
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

#### Ohne TLS-Unterstützung:
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
```

Schließen Sie Pfade, die Leerzeichen enthalten, in Anführungszeichen ein.

Nach der Kompilierung befinden sich die Binärdateien der Zabbix-Komponenten in `C:\Zabbix\zabbix-8.0.0\bin\win64`. Die Konfigurationsdatei des Zabbix Agent befindet sich in `C:\Zabbix\zabbix-8.0.0\conf`.

Um den Agent auszuführen, kopieren Sie `zabbix_agentd.exe` und dessen Konfigurationsdatei in einen dedizierten Ordner (z. B. `C:\Zabbix\agent`) und starten Sie anschließend den Agent:

```
mkdir C:\Zabbix\agent
copy C:\Zabbix\zabbix-8.0.0\bin\win64\zabbix_agentd.exe C:\Zabbix\agent\
copy C:\Zabbix\zabbix-8.0.0\conf\zabbix_agentd.conf C:\Zabbix\agent\

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

macOS-Agent-Installation aus PKG

Übersicht

Der Zabbix Agent kann unter macOS mithilfe von PKG-Installationspaketen installiert werden, die zum Download verfügbar sind.

Zabbix-Agent-Pakete sind mit oder ohne Verschlüsselung verfügbar.

Agent installieren

Der Agent kann über die grafische Benutzeroberfläche oder über die Befehlszeile installiert werden, zum Beispiel:

```
sudo installer -pkg zabbix_agent-8.0.0-macos-arm64-openssl.pkg -target /
```

Stellen Sie sicher, dass Sie in dem Befehl die richtige Zabbix-Paketversion verwenden. Sie muss mit dem Namen des heruntergeladenen Pakets übereinstimmen.

Agent starten

Der Agent wird nach der Installation oder einem Neustart automatisch gestartet.

Bei Bedarf können Sie die Konfigurationsdatei unter `/usr/local/etc/zabbix/zabbix_agentd.conf` bearbeiten.

Um den Agent manuell zu starten, können Sie Folgendes ausführen:

```
sudo launchctl start com.zabbix.zabbix_agentd
```

Um den Agent manuell zu stoppen:

```
sudo launchctl stop com.zabbix.zabbix_agentd
```

Während eines Upgrades wird die vorhandene Konfigurationsdatei nicht überschrieben. Stattdessen wird eine neue Datei `zabbix_agentd.conf.NEW` erstellt, die bei Bedarf zur Überprüfung und Aktualisierung der vorhandenen Konfigurationsdatei verwendet werden kann. Denken Sie daran, den Agent nach manuellen Änderungen an der Konfigurationsdatei neu zu starten.

Fehlerbehebung und Entfernen des Agent

In diesem Abschnitt sind einige nützliche Befehle aufgeführt, die zur Fehlerbehebung und zum Entfernen einer Zabbix-Agent-Installation verwendet werden können.

Prüfen, ob der Zabbix-Agent ausgeführt wird:

```
ps aux | grep zabbix_agentd
```

Prüfen, ob der Zabbix-Agent aus Paketen installiert wurde:

```
pkgutil --pkgs | grep zabbix
com.zabbix.pkg.ZabbixAgent
```

Anzeigen der Dateien, die aus dem Installer-Paket installiert wurden (beachten Sie, dass das führende / in dieser Ansicht nicht angezeigt wird):

```
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
```

Zabbix-Agent stoppen, wenn er mit `launchctl` gestartet wurde:

```
sudo launchctl unload /Library/LaunchDaemons/com.zabbix.zabbix_agentd.plist
```

Dateien entfernen (einschließlich Konfiguration und Protokollen), die mit dem Installer-Paket installiert wurden:

```
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
```

Vergessen, dass der Zabbix-Agent installiert wurde:

```
sudo pkgutil --forget com.zabbix.pkg.ZabbixAgent
```

Erstellen des Zabbix Agent unter macOS

Übersicht

Dieser Abschnitt zeigt, wie Zabbix macOS-Agent-Binärdateien aus dem Quellcode mit oder ohne TLS erstellt werden.

Voraussetzungen

Sie benötigen Entwicklerwerkzeuge für die Befehlszeile (Xcode ist nicht erforderlich), Automake, pkg-config und PCRE (v8.x) oder PCRE2 (v10.x). Wenn Sie Agent- Binärdateien mit TLS erstellen möchten, benötigen Sie außerdem OpenSSL oder GnuTLS.

Um Automake und pkg-config zu installieren, benötigen Sie den Paketmanager Homebrew von https://brew.sh/. Um ihn zu installieren, öffnen Sie das Terminal und führen Sie den folgenden Befehl aus:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Installieren Sie dann Automake und pkg-config:

```
brew install automake
brew install pkg-config
```

Die Vorbereitung der Bibliotheken PCRE, OpenSSL und GnuTLS hängt davon ab, wie sie mit dem Agent verknüpft werden sollen.

Wenn Sie beabsichtigen, Agent-Binärdateien auf einem macOS-Rechner auszuführen, auf dem diese Bibliotheken bereits vorhanden sind, können Sie vorkompilierte Bibliotheken verwenden, die von Homebrew bereitgestellt werden. Dies sind typischerweise macOS-Rechner, die Homebrew zum Erstellen von Zabbix-Agent-Binärdateien oder für andere Zwecke verwenden.

Wenn Agent-Binärdateien auf macOS-Rechnern verwendet werden sollen, auf denen keine Shared-Version dieser Bibliotheken vorhanden ist, sollten Sie statische Bibliotheken aus den Quellen kompilieren und den Zabbix-Agent damit verknüpfen.

Erstellen von Agent-Binärdateien mit gemeinsam genutzten Bibliotheken

Installieren Sie PCRE2 (ersetzen Sie *pcre2* bei Bedarf in den folgenden Befehlen durch *pcre*):

```
brew install pcre2
```

Wenn Sie mit TLS erstellen, installieren Sie OpenSSL und/oder GnuTLS:

```
brew install openssl
brew install gnutls
```

Laden Sie den Zabbix-Quellcode herunter:

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
```

Erstellen Sie den Agent ohne TLS:

```
cd zabbix
./bootstrap.sh
./configure --sysconfdir=/usr/local/etc/zabbix --enable-agent --enable-ipv6
make
make install
```

Erstellen Sie den Agent mit OpenSSL:

```
cd zabbix
./bootstrap.sh
./configure --sysconfdir=/usr/local/etc/zabbix --enable-agent --enable-ipv6 --with-openssl=/usr/local/opt/
make
make install
```

Erstellen Sie den Agent mit 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
```

Erstellen von Agent-Binärdateien mit statischen Bibliotheken ohne TLS

Nehmen wir an, dass die statischen PCRE-Bibliotheken in `$HOME/static-libs` installiert werden. Wir verwenden PCRE2 10.39.

```
PCRE_PREFIX="$HOME/static-libs/pcre2-10.39"
```

Laden Sie PCRE herunter und erstellen Sie es mit Unterstützung für Unicode-Eigenschaften:

```
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
```

Laden Sie den Zabbix-Quellcode herunter und erstellen Sie den 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
```

Erstellen von Agent-Binärdateien mit statischen Bibliotheken mit OpenSSL

Beim Erstellen von OpenSSL wird empfohlen, nach einem erfolgreichen Build `make test` auszuführen. Selbst wenn der Build erfolgreich war, schlagen Tests manchmal fehl. Ist dies der Fall, sollten die Probleme untersucht und behoben werden, bevor Sie fortfahren.

Nehmen wir an, dass die statischen Bibliotheken von PCRE und OpenSSL in `$HOME/static-libs` installiert werden. Wir verwenden PCRE2 10.39 und OpenSSL 1.1.1a.

```
PCRE_PREFIX="$HOME/static-libs/pcre2-10.39"
OPENSSL_PREFIX="$HOME/static-libs/openssl-1.1.1a"
```

Erstellen wir die statischen Bibliotheken in `static-libs-source`:

```
mkdir static-libs-source
cd static-libs-source
```

Laden Sie PCRE mit Unterstützung für Unicode-Eigenschaften herunter und erstellen Sie es:

```
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 ..
```

Laden Sie OpenSSL herunter und erstellen Sie es:

```
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 ..
```

Laden Sie den Zabbix-Quellcode herunter und erstellen Sie den 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
```

Erstellen von Agent-Binärdateien mit statischen Bibliotheken mit GnuTLS

GnuTLS hängt vom Nettle-Krypto-Backend und der GMP-Arithmetikbibliothek ab. Anstelle der vollständigen GMP-Bibliothek wird in dieser Anleitung mini-gmp verwendet, das in Nettle enthalten ist.

Beim Erstellen von GnuTLS und Nettle wird empfohlen, nach einem erfolgreichen Build `make check` auszuführen. Selbst wenn der Build erfolgreich war, schlagen Tests manchmal fehl. In diesem Fall sollten die Probleme untersucht und behoben werden, bevor Sie fortfahren.

Nehmen wir an, dass die statischen Bibliotheken von PCRE, Nettle und GnuTLS in `$HOME/static-libs` installiert werden. Wir verwenden PCRE2 10.39, Nettle 3.4.1 und 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"
```

Erstellen wir die statischen Bibliotheken in `static-libs-source`:

```
mkdir static-libs-source
cd static-libs-source
```

Nettle herunterladen und erstellen:

```
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 ..
```

GnuTLS herunterladen und erstellen:

```
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 ..
```

Zabbix-Quellcode herunterladen und Agent erstellen:

```
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
```

AWS-Bereitstellungshandbuch für den Zabbix Server

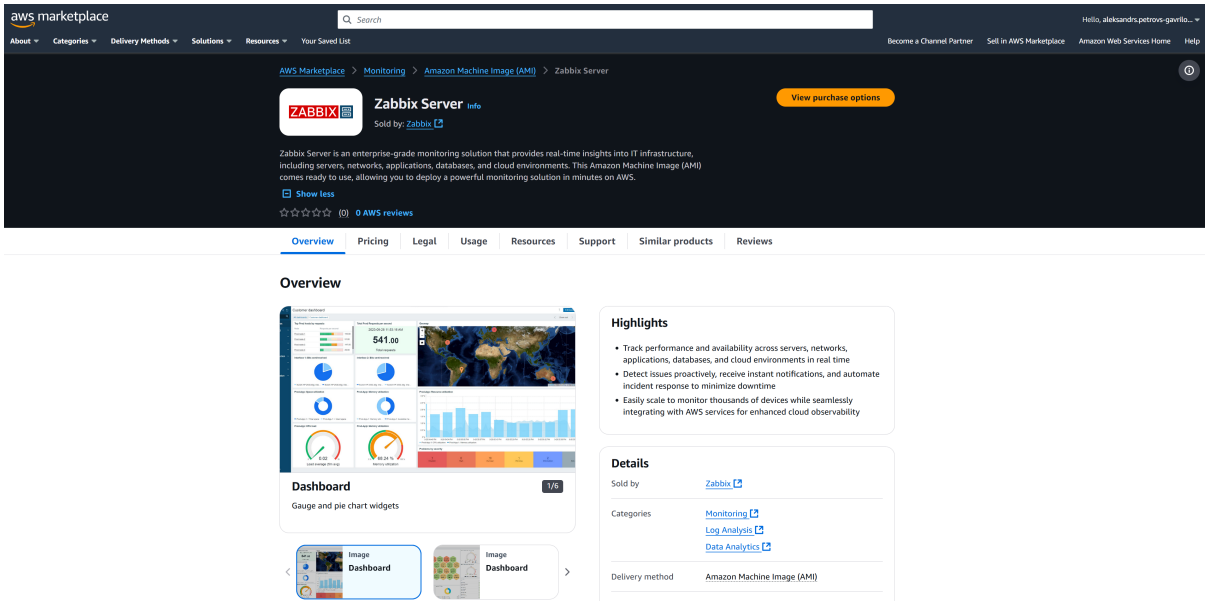
Übersicht

Diese Seite beschreibt, wie der Zabbix Server mithilfe von AWS bereitgestellt wird.

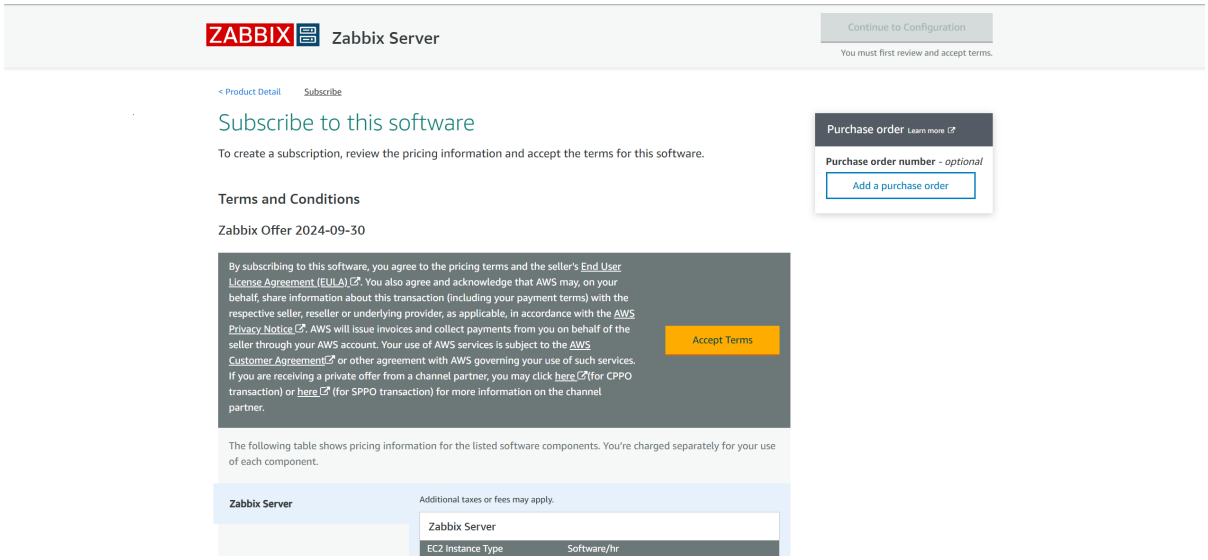
> **Attention:**
> Für die Bereitstellung auf der Plattform sind mehrere Zabbix-Versionen verfügbar. Lesen Sie unbedingt die Dokumentation für dieselbe Zabbix-Version, die Sie bereitgestellt haben. Die Dokumentation für verschiedene Zabbix-Versionen kann über das Dropdown-Menü für Versionen ausgewählt werden.

Bereitstellen der Appliance

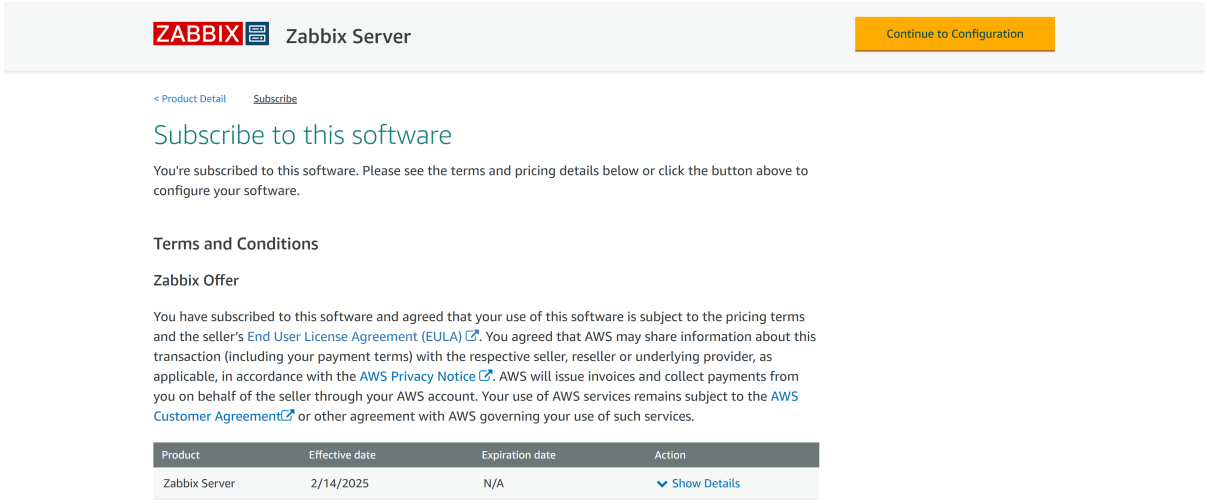
Um die Zabbix Appliance bereitzustellen, rufen Sie zunächst die Zabbix AWS Marketplace-Seite auf. Klicken Sie anschließend auf *View purchase options*, um das Image bereitzustellen und zur Abonnementseite zu gelangen.



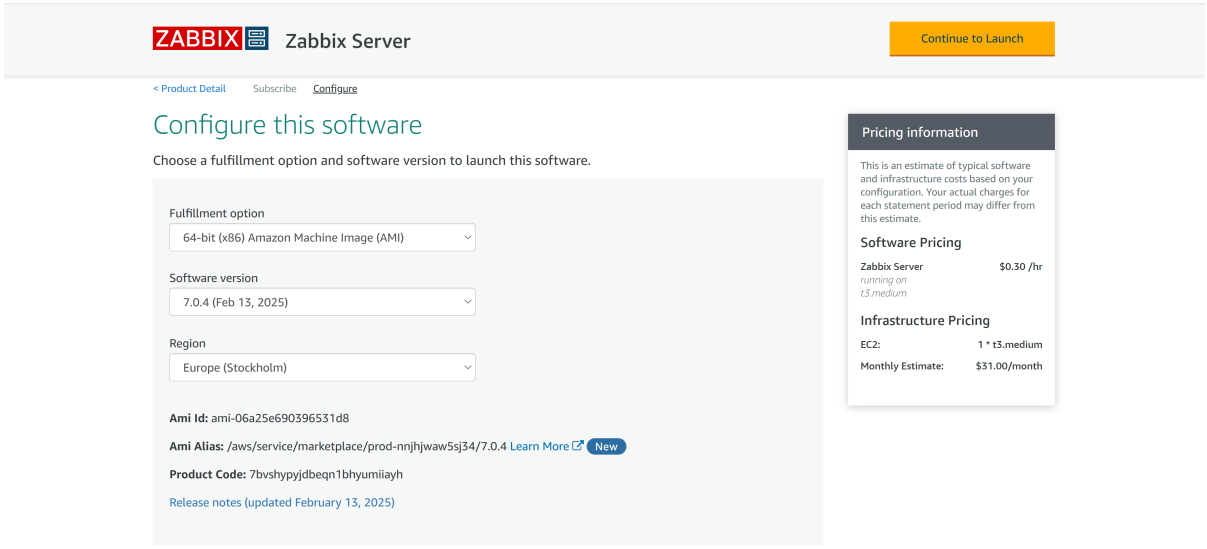
Klicken Sie auf die Schaltfläche *Accept terms*, um die *Terms and Conditions* zu akzeptieren.



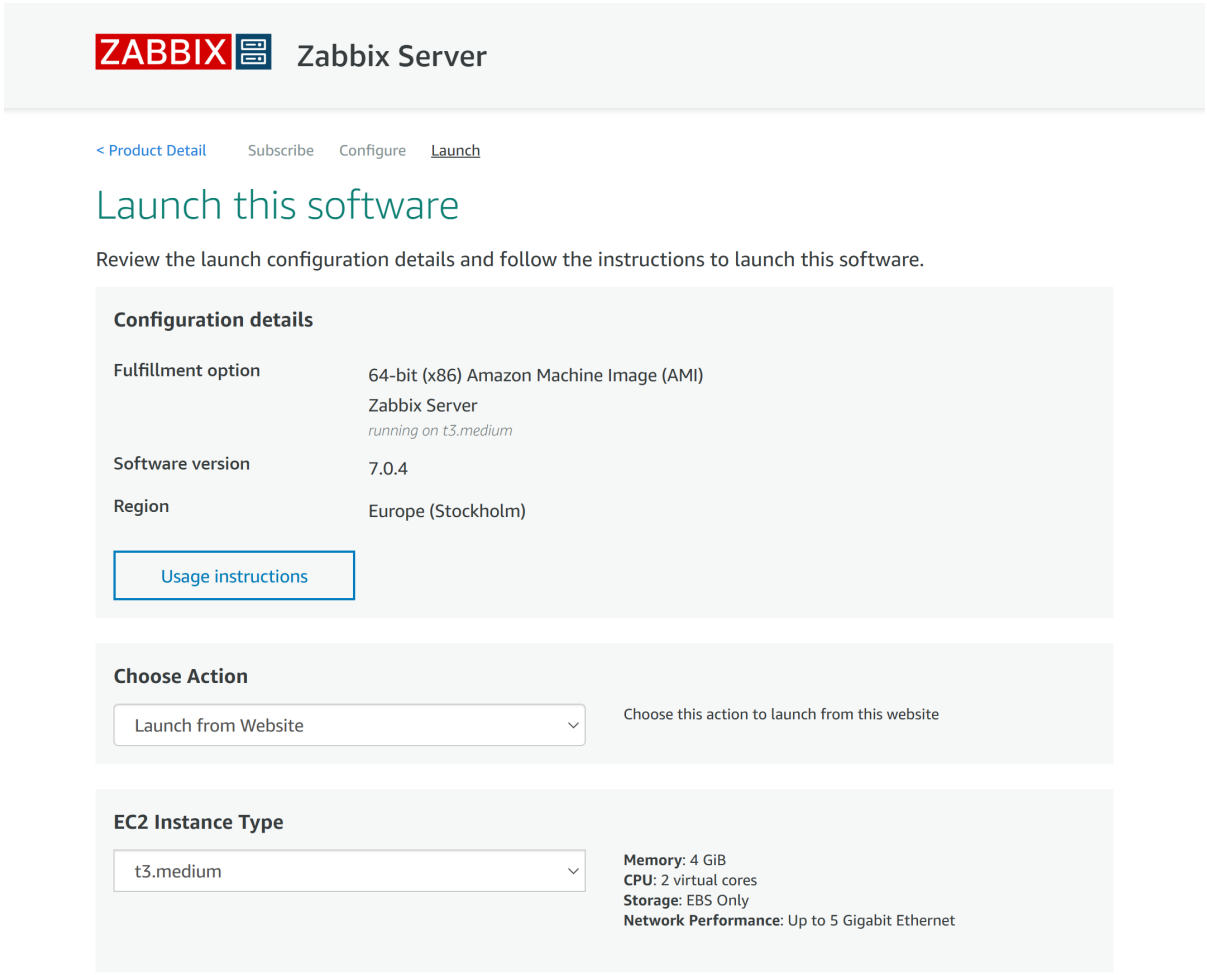
Klicken Sie auf die Schaltfläche *Continue to Configuration*, um zum Abschnitt „Configuration“ für die Zabbix Appliance zu gelangen.



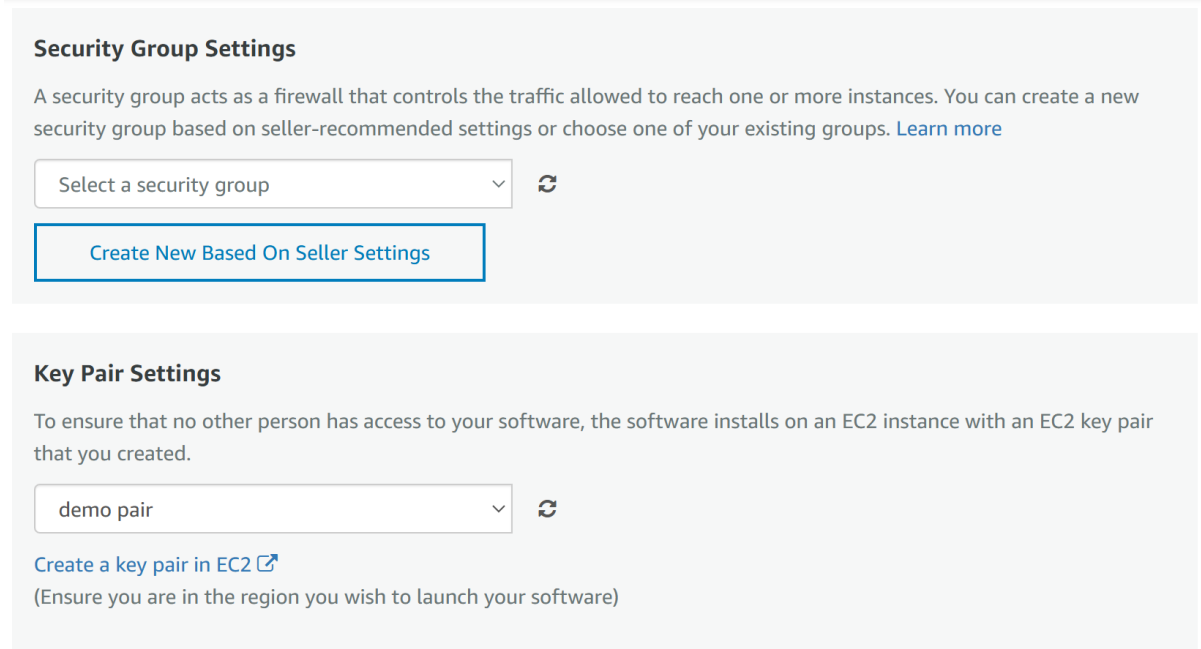
Wählen Sie die Fulfilment-Option, die Zabbix-Version und Ihre bevorzugte Region aus.



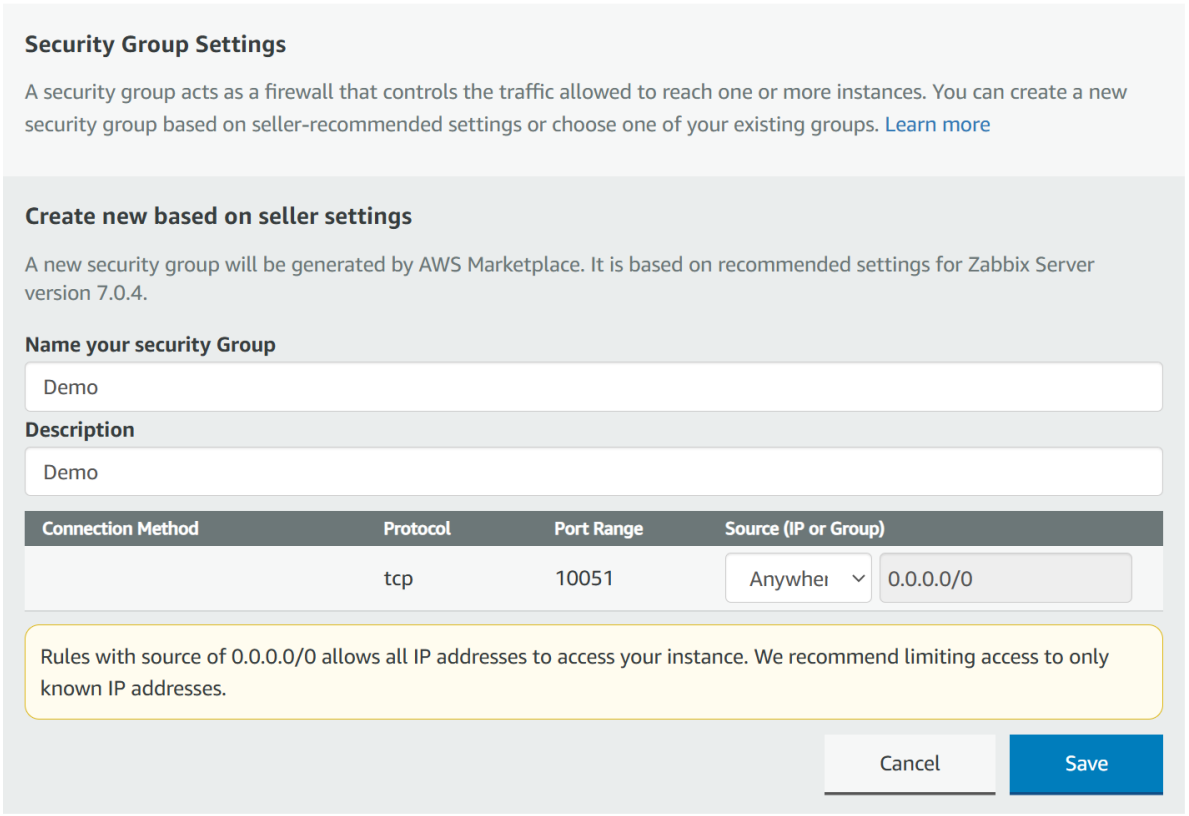
Klicken Sie auf die Schaltfläche *Continue to Launch*. Wählen Sie auf der Launch-Seite die bevorzugte Launch-Option, den EC2 Instance Type sowie die VPC- und Subnet-Einstellungen aus.



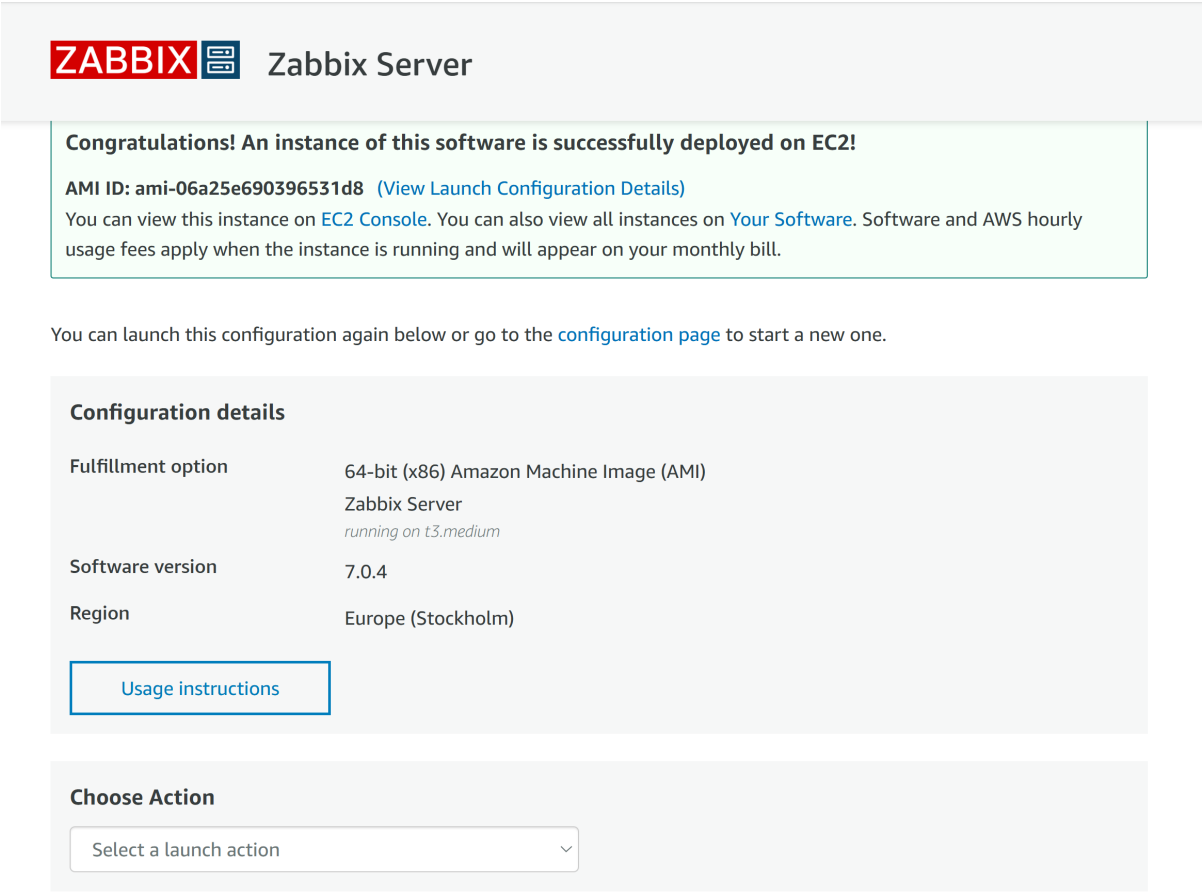
Geben Sie unten auf der Launch-Seite die Einstellungen für die Security Group an und wählen Sie ein EC2-Schlüsselpaar aus oder erstellen Sie eines. Stellen Sie sicher, dass Sie Ihren privaten Schlüssel an einem sicheren Ort speichern!



Wenn Sie *Create a new Security Group based on seller settings* auswählen, wird TCP-Port 10051 (Zabbix Server) zugelassen. Hinweis: Wenn Sie über SSH und HTTP/HTTPS (Frontend) auf die Instanz zugreifen möchten, müssen Sie in den Einstellungen Ihrer Security Group ebenfalls SSH- und HTTP/HTTPS-Verbindungen zulassen.

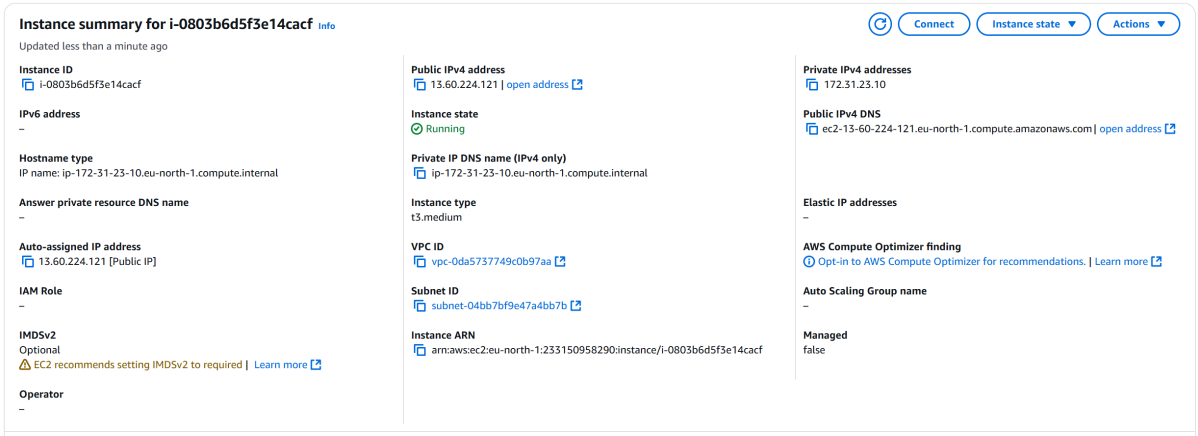


Herzlichen Glückwunsch, Ihre Zabbix Appliance wurde erfolgreich bereitgestellt.

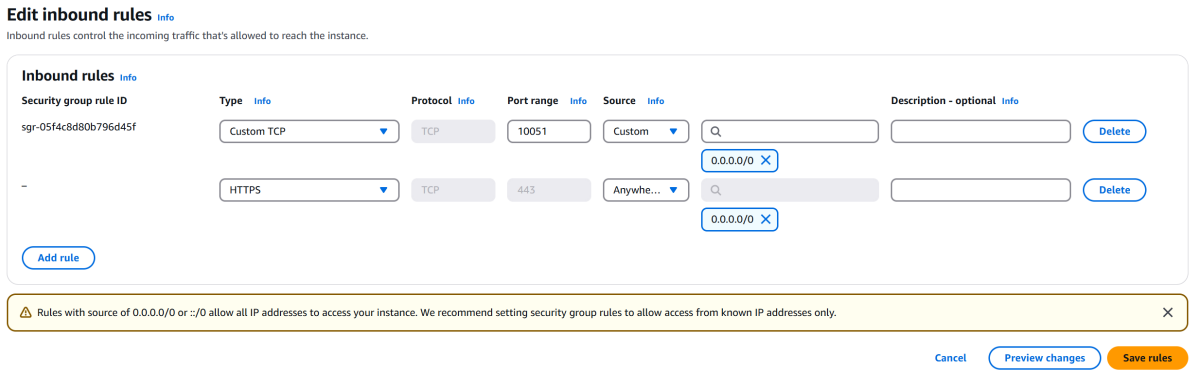


Zugriff auf die Appliance

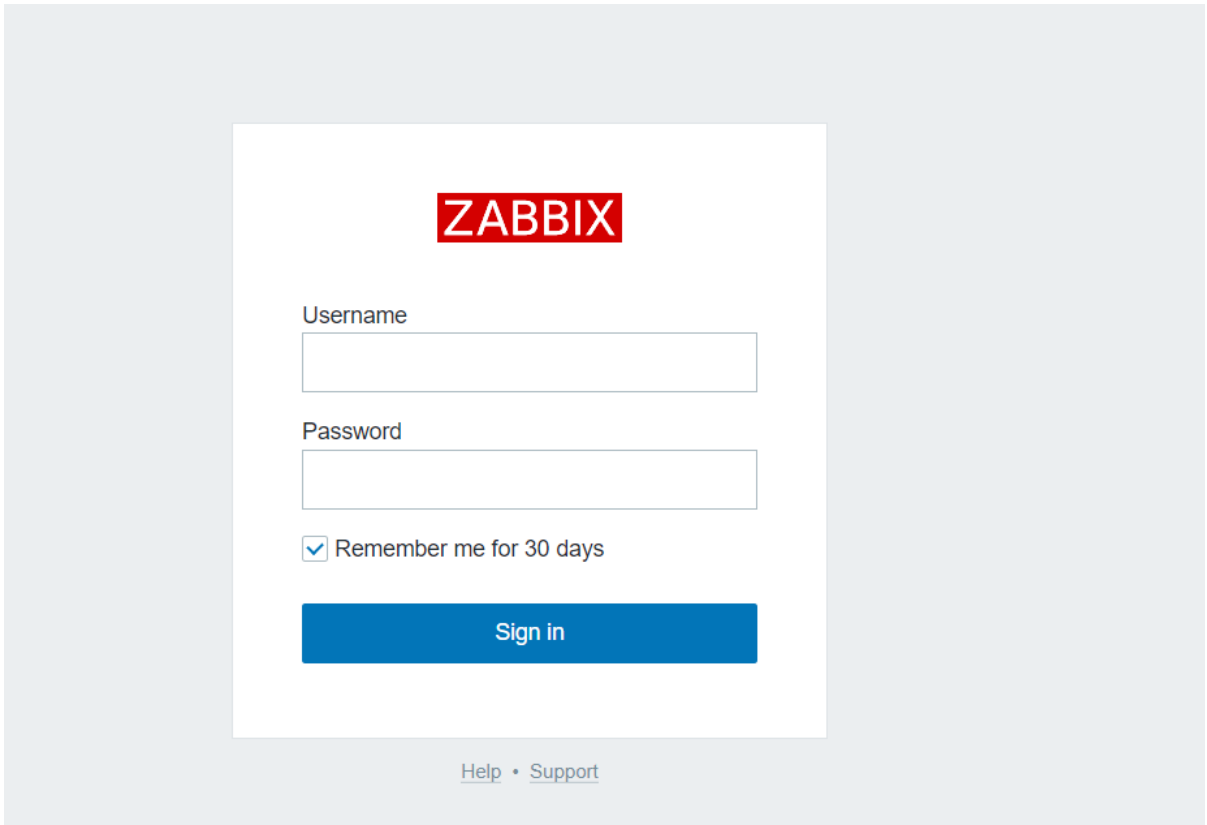
Öffnen Sie den Abschnitt *Instances* und wählen Sie die neu bereitgestellte Zabbix-Appliance-Instanz aus.



Um über die öffentliche IPv4-Adresse oder den öffentlichen IPv4-DNS-Namen der Instanz auf das Zabbix-Frontend zuzugreifen, erlauben Sie eingehenden HTTPS-Datenverkehr über Port 443 in der Sicherheitsgruppe.



Nun können Sie auf das Zabbix-Frontend zugreifen, indem Sie die öffentliche IPv4-Adresse oder den öffentlichen IPv4-DNS-Namen der Zabbix-Appliance öffnen.



Um das Passwort zu erhalten, müssen Sie entweder über die serielle Konsole oder per SSH eine Verbindung zur Maschine herstellen. Der Standardbenutzername für SSH lautet rocky. Vergessen Sie nicht, das zuvor ausgewählte oder generierte Schlüsselpaar zu verwenden. Beachten Sie, dass die Generierung der Anmeldedaten einige Zeit dauern kann. Die Generierung der Anmeldedaten kann einige Minuten dauern. Warten Sie einige Minuten, bevor Sie versuchen, eine Verbindung herzustellen.

```
Last login: Tue Nov 16 15:16:07 2021 from ec2-13-48-4-200.eu-north-1.compute.amazonaws.com
********************************************************************************

Zabbix frontend credentials:

Username: Admin

Password: gJ8f7XpgJbsnc2BU48g

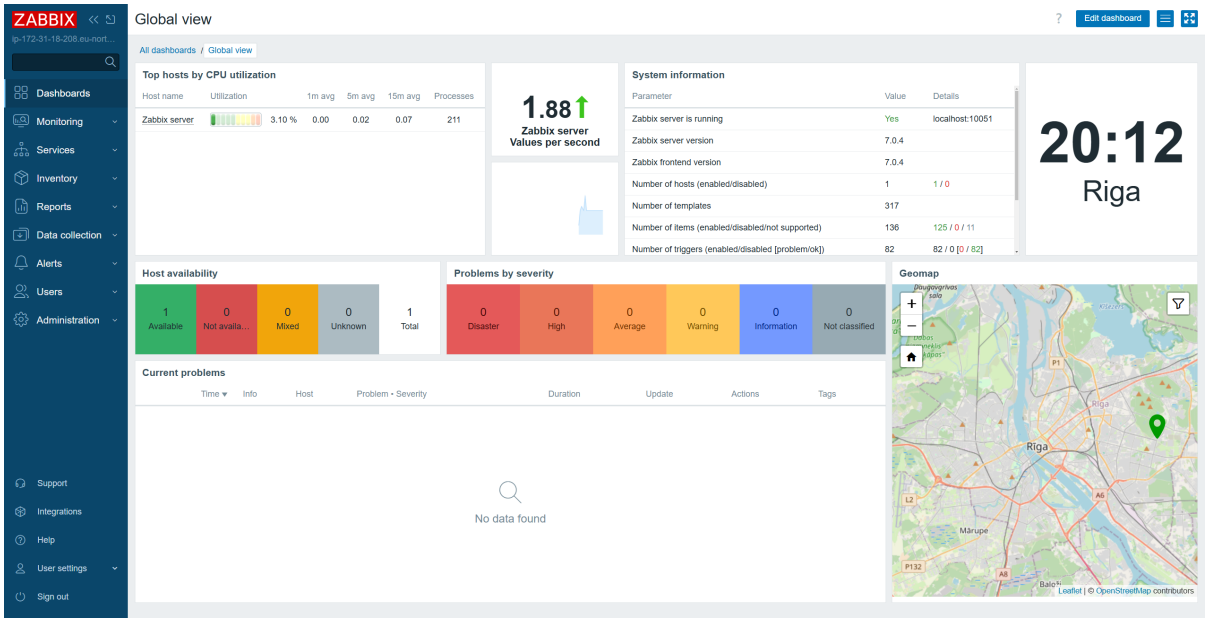
To learn about available professional services, including technical suppport and training, please visit https://www.zabbix.com/services

Official Zabbix documentation available at https://www.zabbix.com/documentation/current/

Note! Do not forget to change timezone PHP variable in /etc/php.d/99-zabbix.ini file.

********************************************************************************
```

Nun können Sie diese Informationen verwenden, um sich bei Ihrem Zabbix-Frontend anzumelden.



Zugriff auf die Datenbank

Sie können auf die Zabbix-Datenbank mit dem Benutzer `root` zugreifen, ohne ein Passwort einzugeben.

```
[root@ip-172-31-29-138 ~]# mysql -uroot
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 124
Server version: 8.0.26 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Die Anmeldedaten sind in der Konfigurationsdatei `/root/.my.cnf` gespeichert.

Die Standardzeitzone des Zabbix Frontends kann unter *Administration* > *Allgemein* > *GUI* geändert werden.

AWS-Bereitstellungshandbuch für den Zabbix Proxy

Übersicht

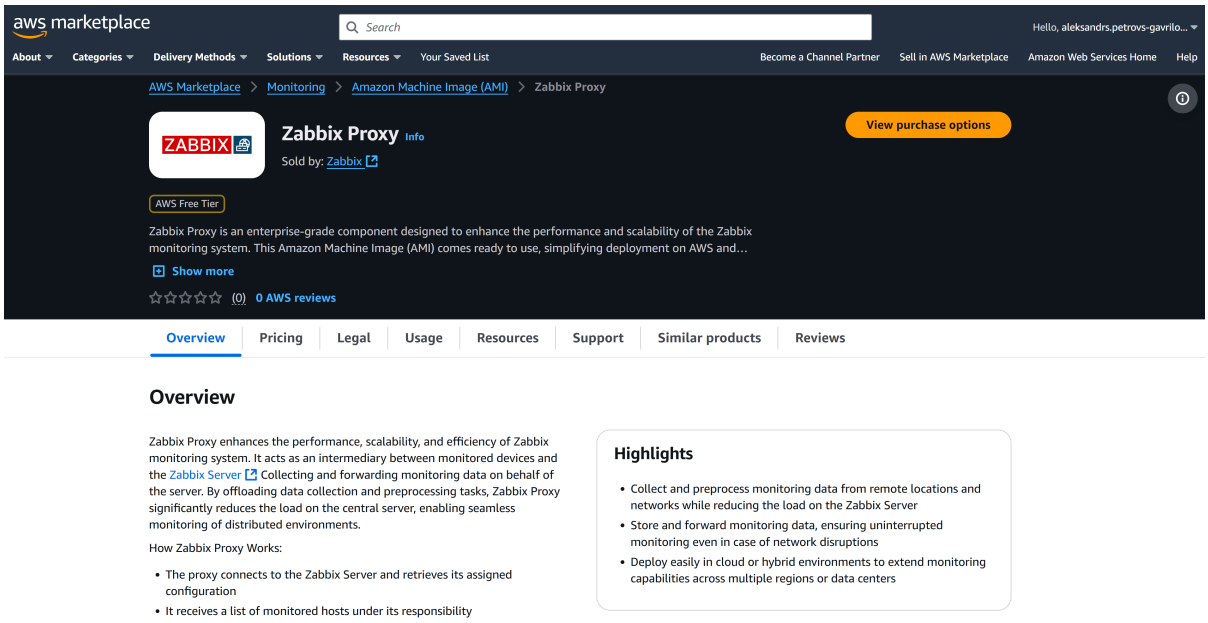
Diese Seite beschreibt, wie Sie einen Zabbix Proxy mithilfe von AWS bereitstellen.

Attention:
Für die Bereitstellung auf der Plattform sind mehrere Zabbix-Versionen verfügbar. Lesen Sie unbedingt die Dokumentation für dieselbe Zabbix-Version, die Sie bereitgestellt haben. Die Dokumentation für verschiedene Zabbix-Versionen kann über das Dropdown-Menü für die Version ausgewählt werden.

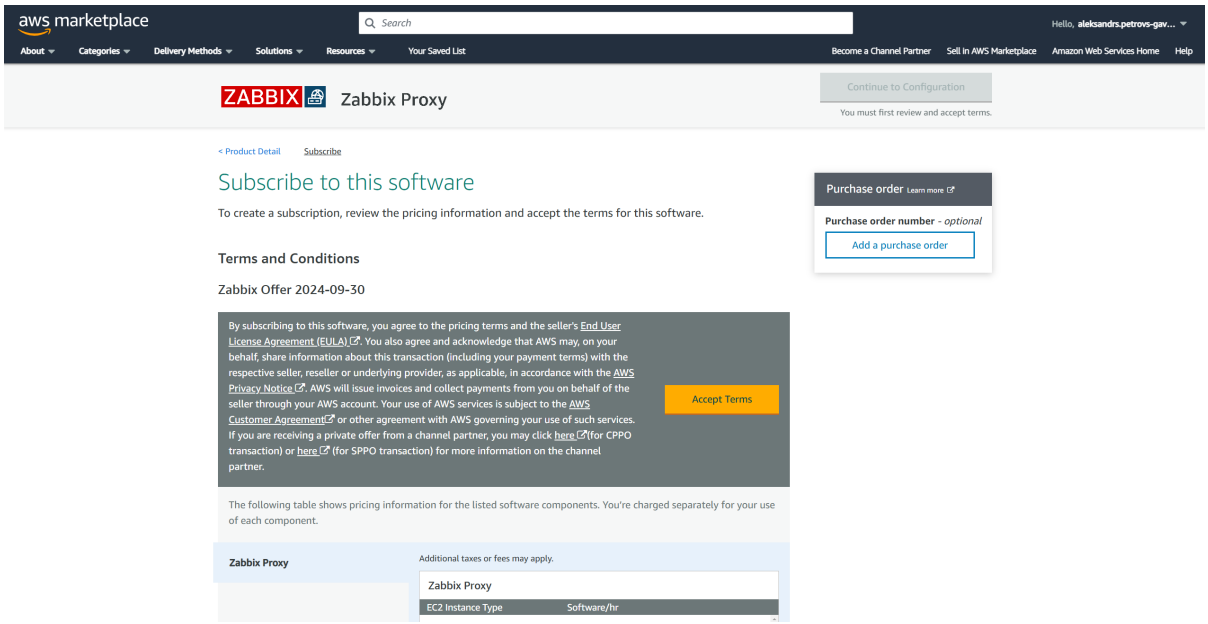
Bereitstellung des Zabbix Proxy

Attention:
Für die Verwendung der Zabbix Proxy-Komponente ist eine funktionierende Zabbix-Server-Instanz erforderlich.

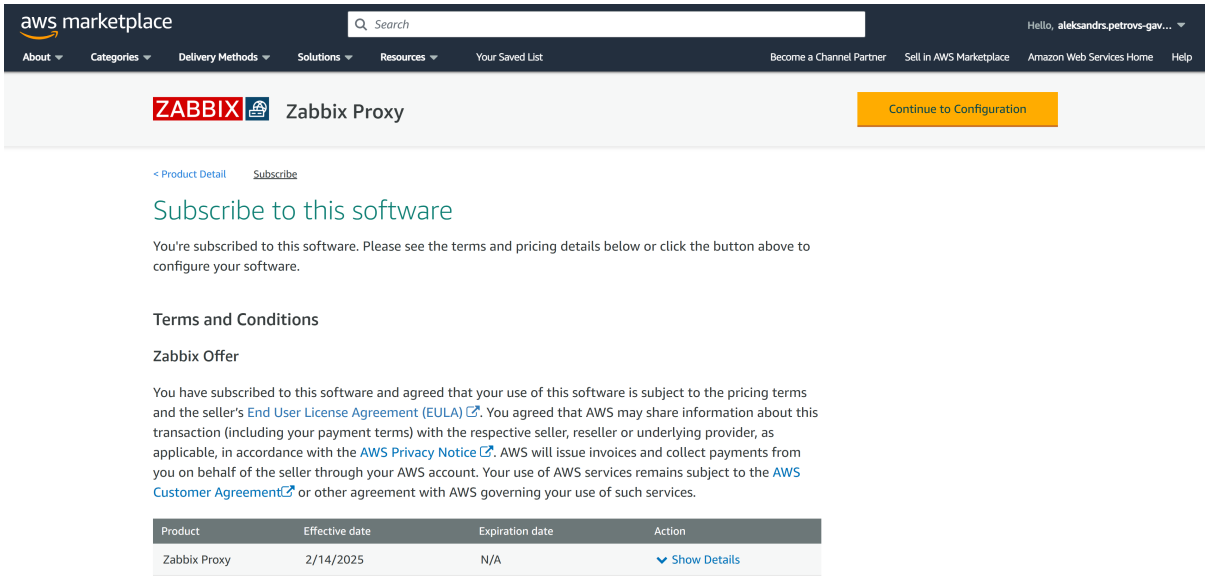
Um den Zabbix Proxy bereitzustellen, rufen Sie zunächst die Zabbix AWS Marketplace-Seite auf. Klicken Sie anschließend auf *View purchase options*, um das Image bereitzustellen, und fahren Sie mit der Abonnementseite fort.



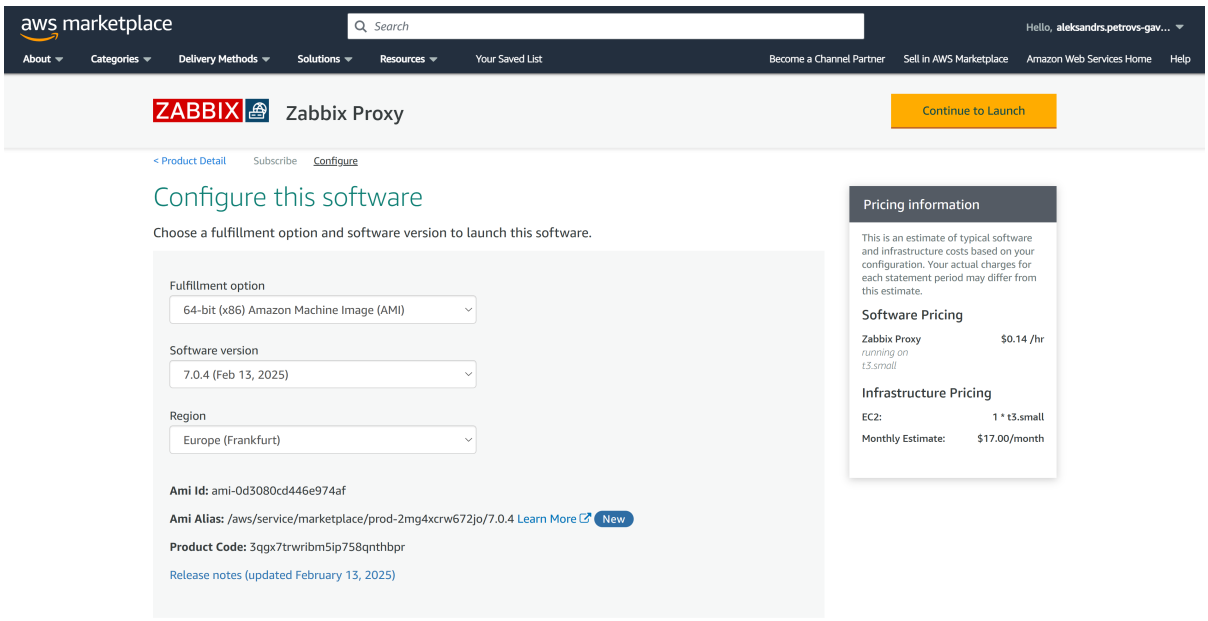
Klicken Sie auf die Schaltfläche *Accept terms*, um die *Terms and Conditions* zu akzeptieren.



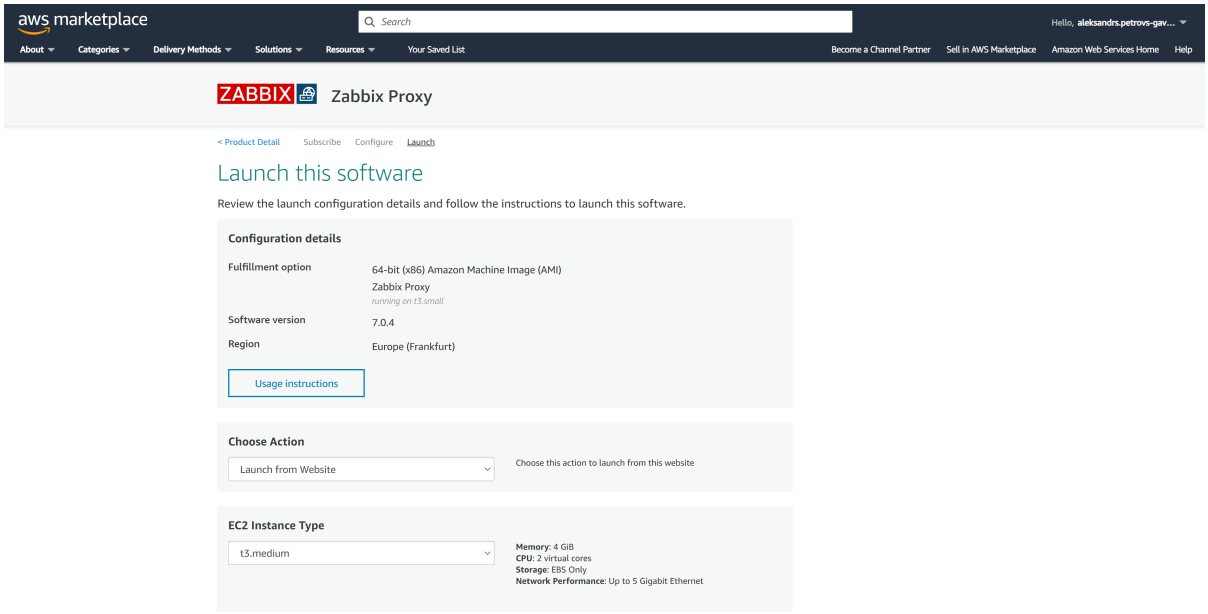
Klicken Sie auf die Schaltfläche *Continue to Configuration*, um zum Abschnitt „Configuration“ für den Zabbix Proxy zu wechseln.



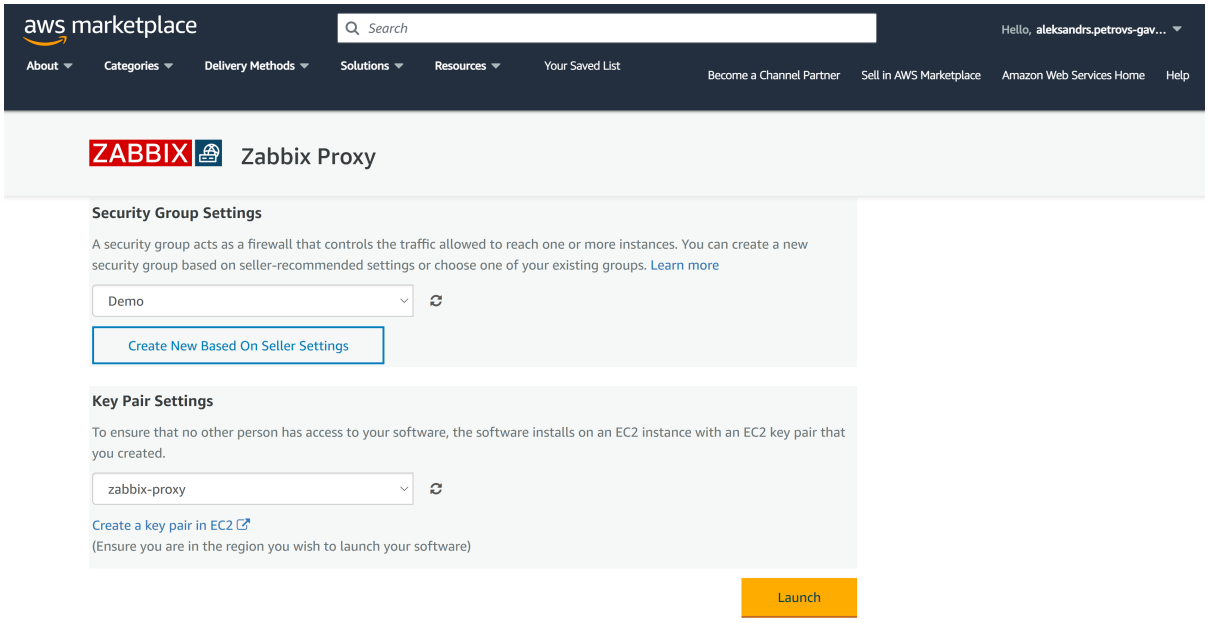
Wählen Sie die Fulfilment-Option, die Zabbix-Version und Ihre bevorzugte Region aus.



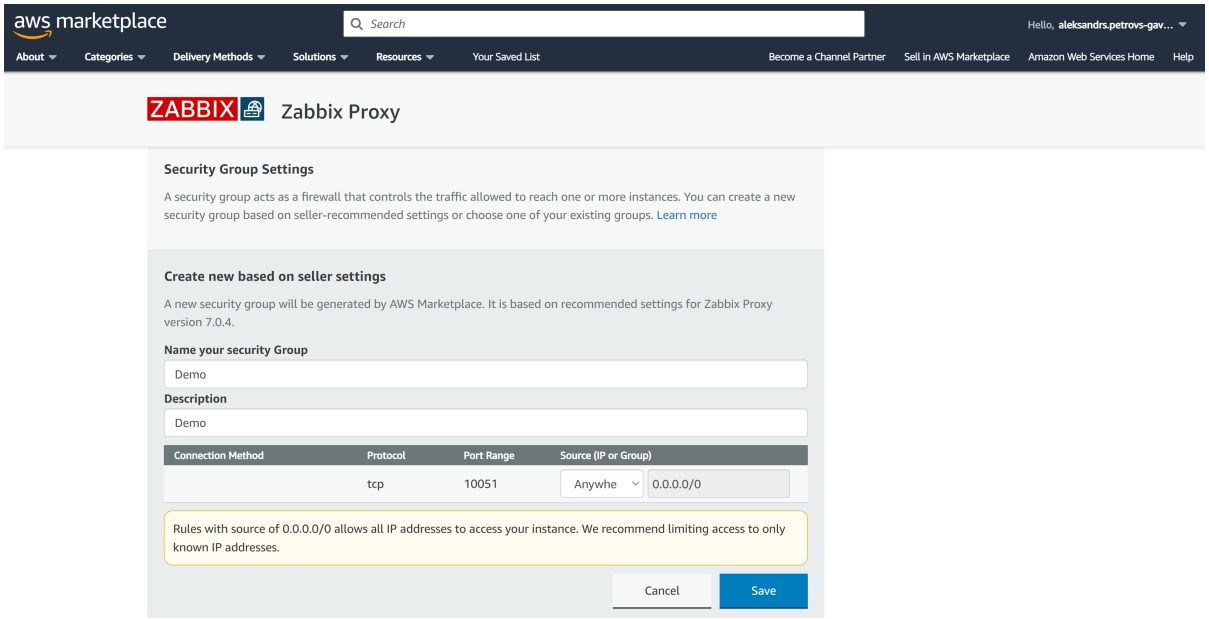
Klicken Sie auf die Schaltfläche *Continue to Launch*. Wählen Sie auf der Launch-Seite die bevorzugte Launch-Option, den EC2-Instanztyp sowie die VPC- und Subnet-Einstellungen aus.



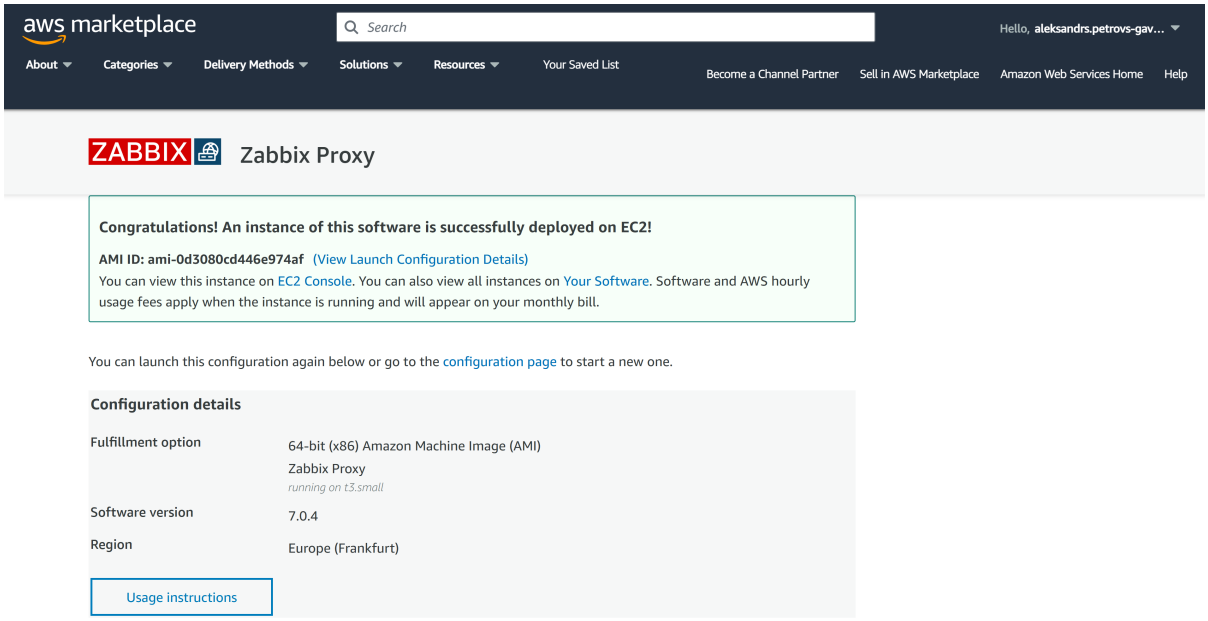
Geben Sie unten auf der Launch-Seite die Einstellungen für die Sicherheitsgruppe an und wählen Sie ein EC2-Schlüsselpaar aus oder erstellen Sie eines. Bewahren Sie Ihren privaten Schlüssel an einem sicheren Ort auf.



Wenn Sie *Create a new Security Group based on the Seller settings* auswählen, wird der TCP-Port 10051 (Zabbix Server) freigegeben. Hinweis: Wenn Sie über SSH auf die Instanz zugreifen möchten, müssen Sie in den Einstellungen der Sicherheitsgruppe zusätzlich SSH-Verbindungen zulassen.

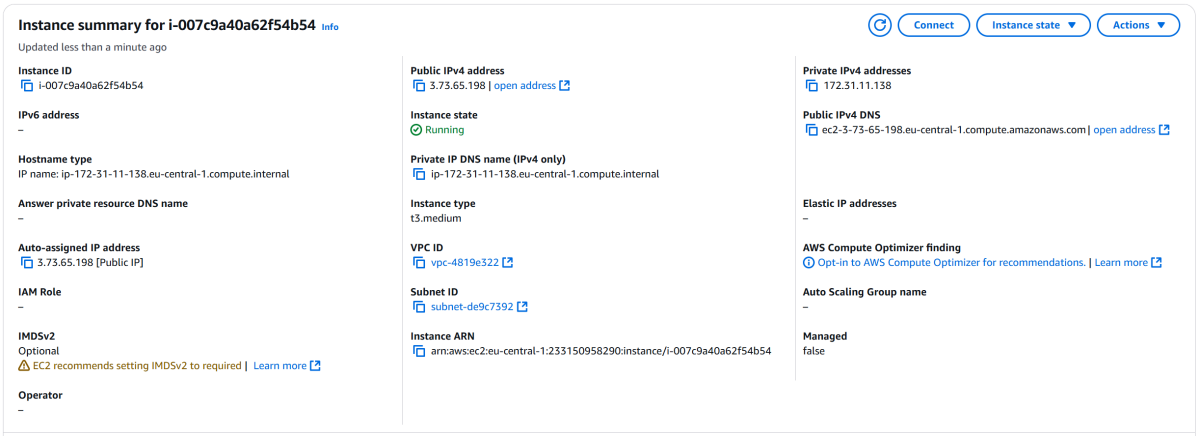


Klicken Sie abschließend auf die Schaltfläche *Launch*. Herzlichen Glückwunsch, Ihr Zabbix Proxy wird nun bereitgestellt.



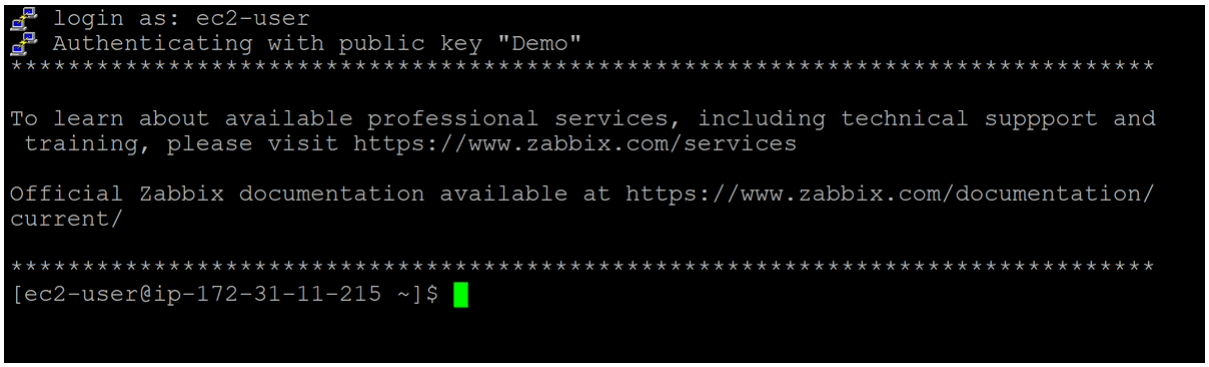
Konfigurieren und Zugreifen auf den Zabbix Proxy

Öffnen Sie den Abschnitt „Instances“ in der EC2-Konsole und wählen Sie die neu bereitgestellte Zabbix-Proxy-Instanz aus.



Stellen Sie über SSH eine Verbindung zur öffentlichen IPv4-Adresse oder zum öffentlichen IPv4-DNS des Zabbix Proxy her. Der Standard-SSH-Benutzername lautet `rocky`. Vergessen Sie nicht, das zuvor generierte Schlüsselpaar zu verwenden.

Die Generierung der Anmeldedaten kann einige Minuten dauern. Warten Sie einige Minuten, bevor Sie versuchen, eine Verbindung herzustellen.



Die Konfigurationsdatei des Zabbix Proxy befindet sich unter `/etc/zabbix/zabbix_proxy.conf`.

Die Konfigurationsdatei kann mit einem Texteditor Ihrer Wahl geöffnet werden, zum Beispiel mit vi:

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

Die anfängliche Proxy-Konfiguration besteht darin, die Parameter `ProxyMode`, `Server` und `Hostname` in der Konfigurationsdatei des Zabbix Proxy anzupassen.

Der Proxy-Modus kann durch Bearbeiten des Parameters `ProxyMode` geändert werden. Setzen Sie `ProxyMode=0` für den aktiven Modus oder `ProxyMode=1` für den passiven Modus.

Der Parameter `Server` sollte die IP-Adresse des Zabbix Server enthalten. Zum Beispiel:

```
Server=192.168.50.141
```

Der Parameter `Hostname` sollte den Hostnamen des Zabbix Proxy enthalten. Der hier definierte Hostname muss mit dem Proxy-Namen übereinstimmen, der während der Proxy-Konfiguration im Zabbix Frontend angegeben wurde. Zum Beispiel:

```
Hostname=EC2-Frankfurt
```

Sobald die Konfiguration abgeschlossen ist, starten Sie den Zabbix Proxy mit dem folgenden Befehl neu:

```
sudo systemctl restart zabbix-proxy
```

Hinweis: Verwenden Sie sudo, um Konfigurationsdateien zu bearbeiten und Befehle über die Befehlszeile auszuführen.

```
### Option: ProxyMode
#       Proxy operating mode.
#       0 - proxy in the active mode
#       1 - proxy in the passive mode
#
# Mandatory: no
# Default:
# ProxyMode=0

### Option: Server
#       If ProxyMode is set to active mode:
#               IP address or DNS name of Zabbix server to get configuration data from and send data to.
#       If ProxyMode is set to passive mode:
#               List of comma delimited IP addresses, optionally in CIDR notation, or DNS names of Zabbix
server.
#               Incoming connections will be accepted only from the addresses listed here.
#               If IPv6 support is enabled then '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' are treated
 equally
#               and '::/0' will allow any IPv4 or IPv6 address.
#               '0.0.0.0/0' can be used to allow any IPv4 address.
#               Example: Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
#
# Mandatory: yes
# Default:
# Server=

Server=127.0.0.1
```

```
### Option: Hostname
#       Unique, case sensitive Proxy name. Make sure the Proxy name is known to the server!
#       Value is acquired from HostnameItem if undefined.
#
# Mandatory: no
# Default:
# Hostname=

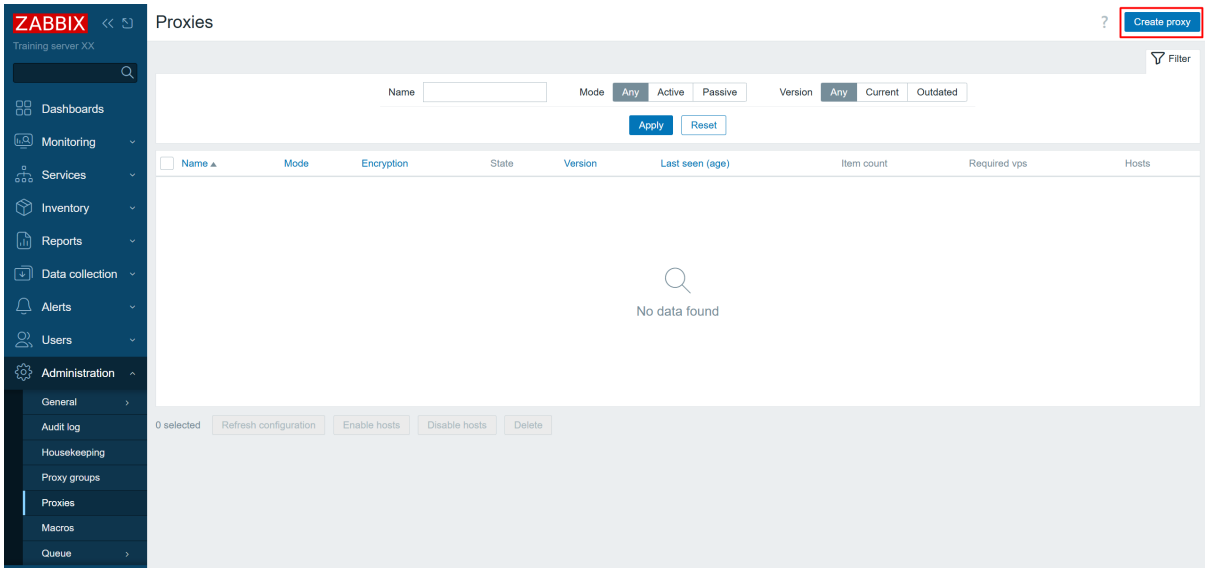
Hostname=EC2-Frankfurt
```

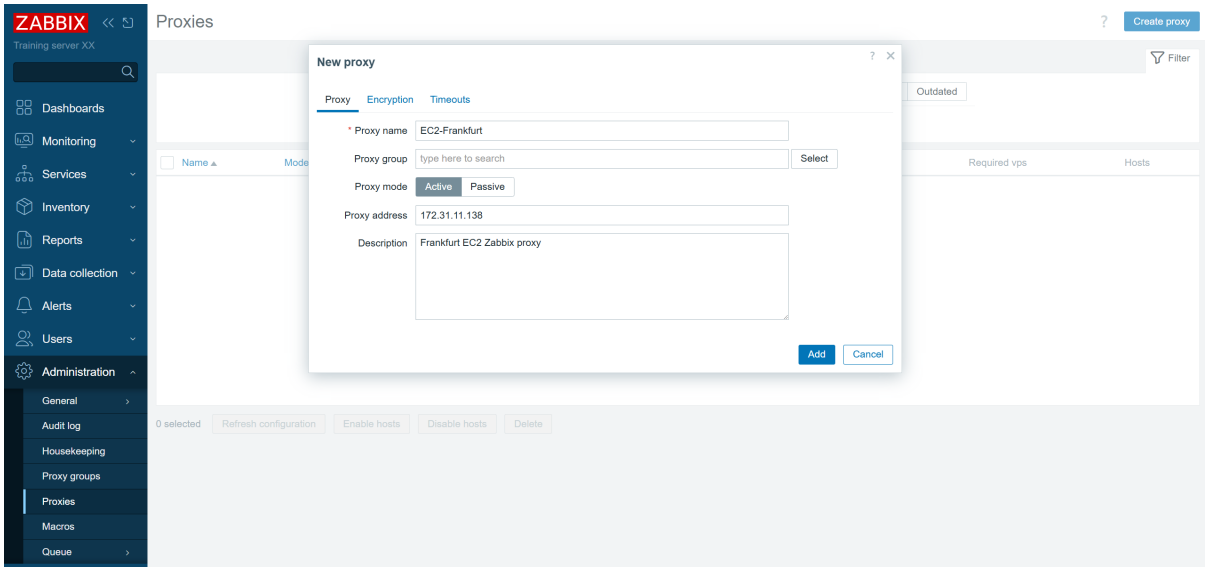
Öffnen Sie das Zabbix Frontend und navigieren Sie zu *Administration* > *Proxies*. Klicken Sie auf die Schaltfläche *Create proxy*.

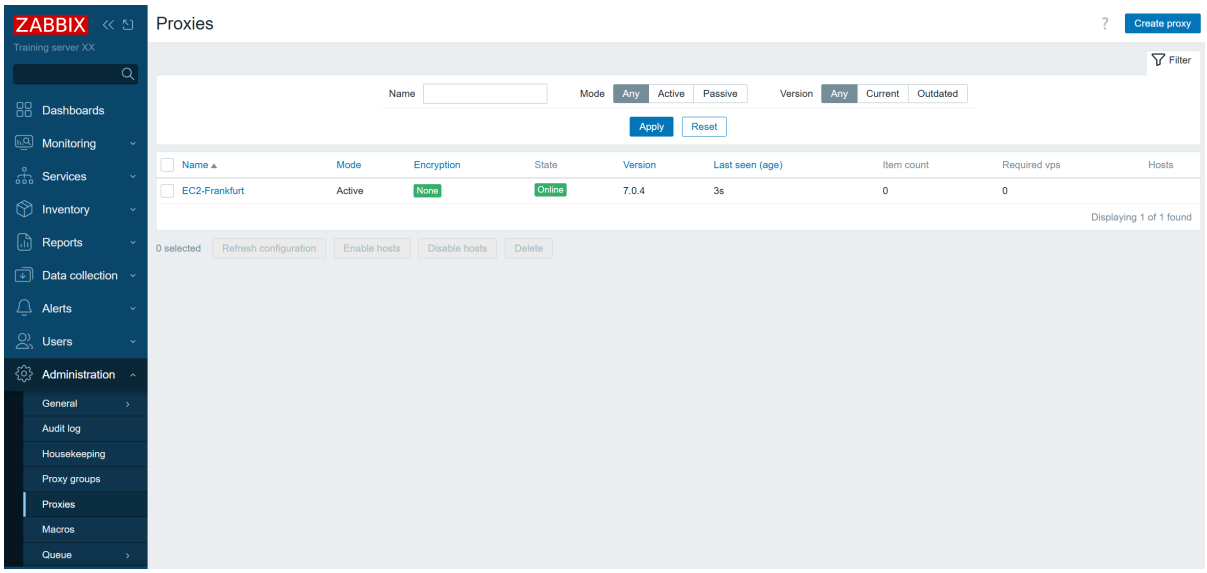
Geben Sie denselben Proxy-Namen wie den Wert des Parameters `Hostname` in der Proxy-Konfigurationsdatei ein.

Wählen Sie den Proxy-Modus aus und geben Sie die Proxy-IP-Adresse an (optional für aktive Proxies).

Klicken Sie auf *Create proxy*.







Speicherort der Proxy-Datenbank

Die SQLite-Datenbank des Proxy wird in `/tmp/zabbix_proxy.sqlite3` gespeichert und durch den Parameter `DBName` in der Konfigurationsdatei des Zabbix Proxy angegeben.

Azure-Bereitstellungshandbuch für den Zabbix Server

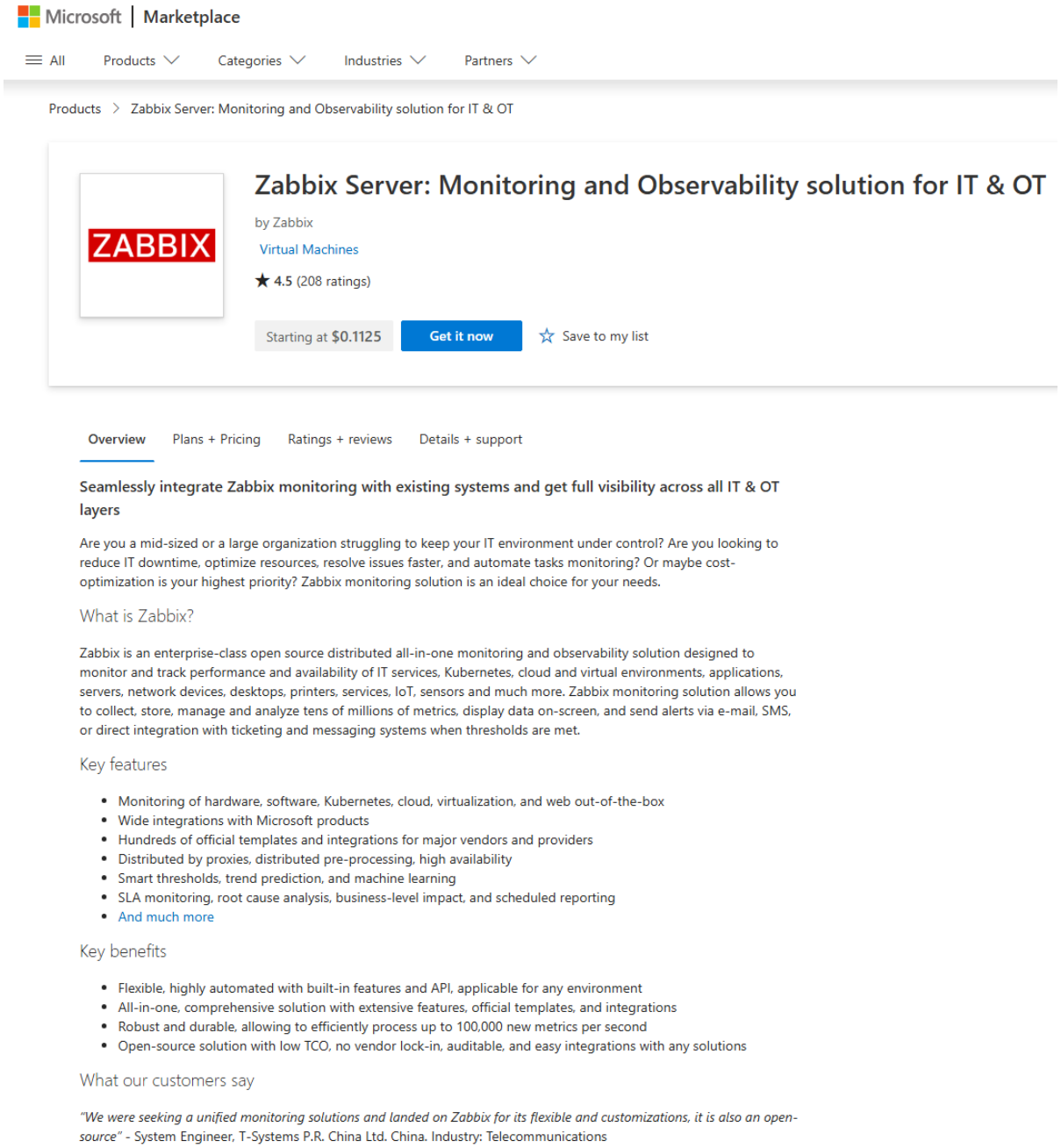
Übersicht

Diese Seite beschreibt, wie der Zabbix Server mithilfe von Azure bereitgestellt wird.

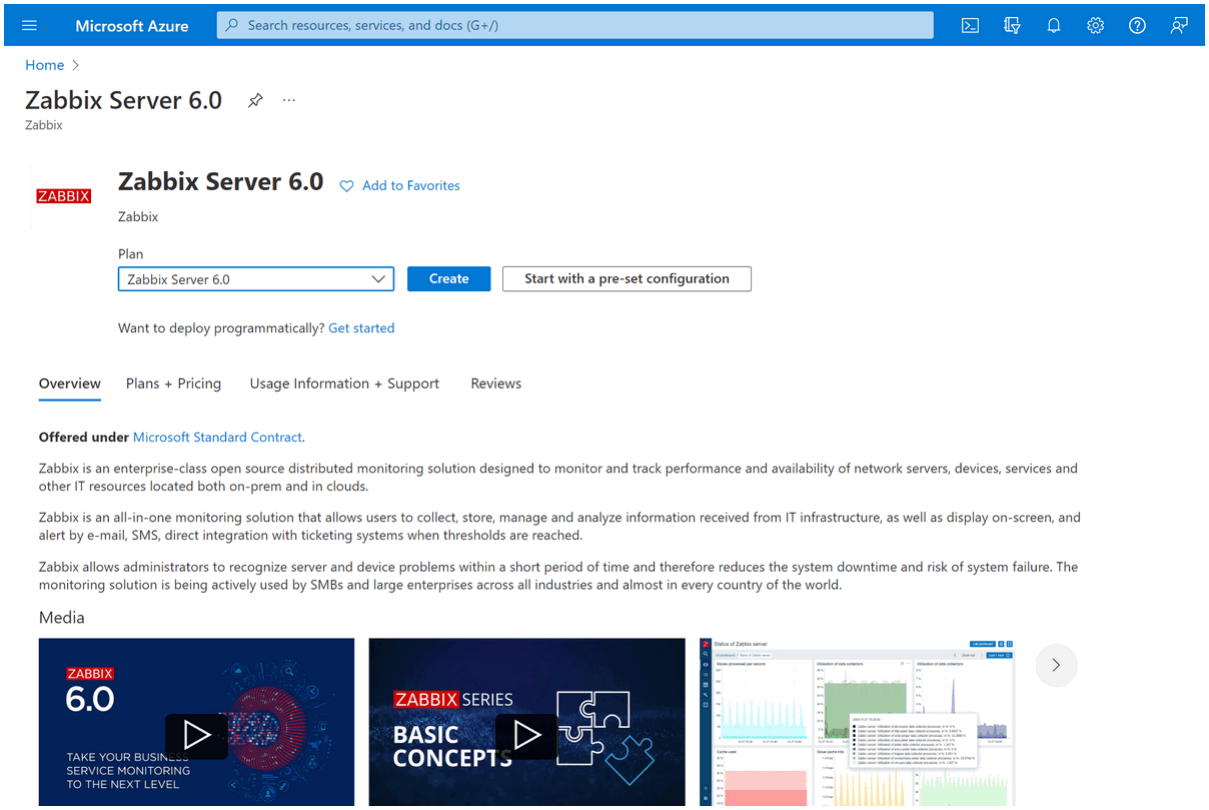
> **Attention:**
> Für die Bereitstellung auf der Plattform sind mehrere Zabbix-Versionen verfügbar. Lesen Sie unbedingt die Dokumentation für dieselbe Zabbix-Version, die Sie bereitgestellt haben. Die Dokumentation für verschiedene Zabbix-Versionen kann über das Dropdown-Menü für Versionen ausgewählt werden.

Bereitstellen der Appliance

Um die Zabbix Appliance bereitzustellen, rufen Sie zunächst die Zabbix Azure Marketplace-Seite auf. Lesen Sie die Übersicht und klicken Sie anschließend auf *Get it now*.



Lesen Sie die Bereitstellungsübersicht, die Preispläne und weitere Details. Klicken Sie anschließend entweder auf *Create* oder auf *Start with a pre-set configuration*, um Ihre Zabbix Appliance bereitzustellen.

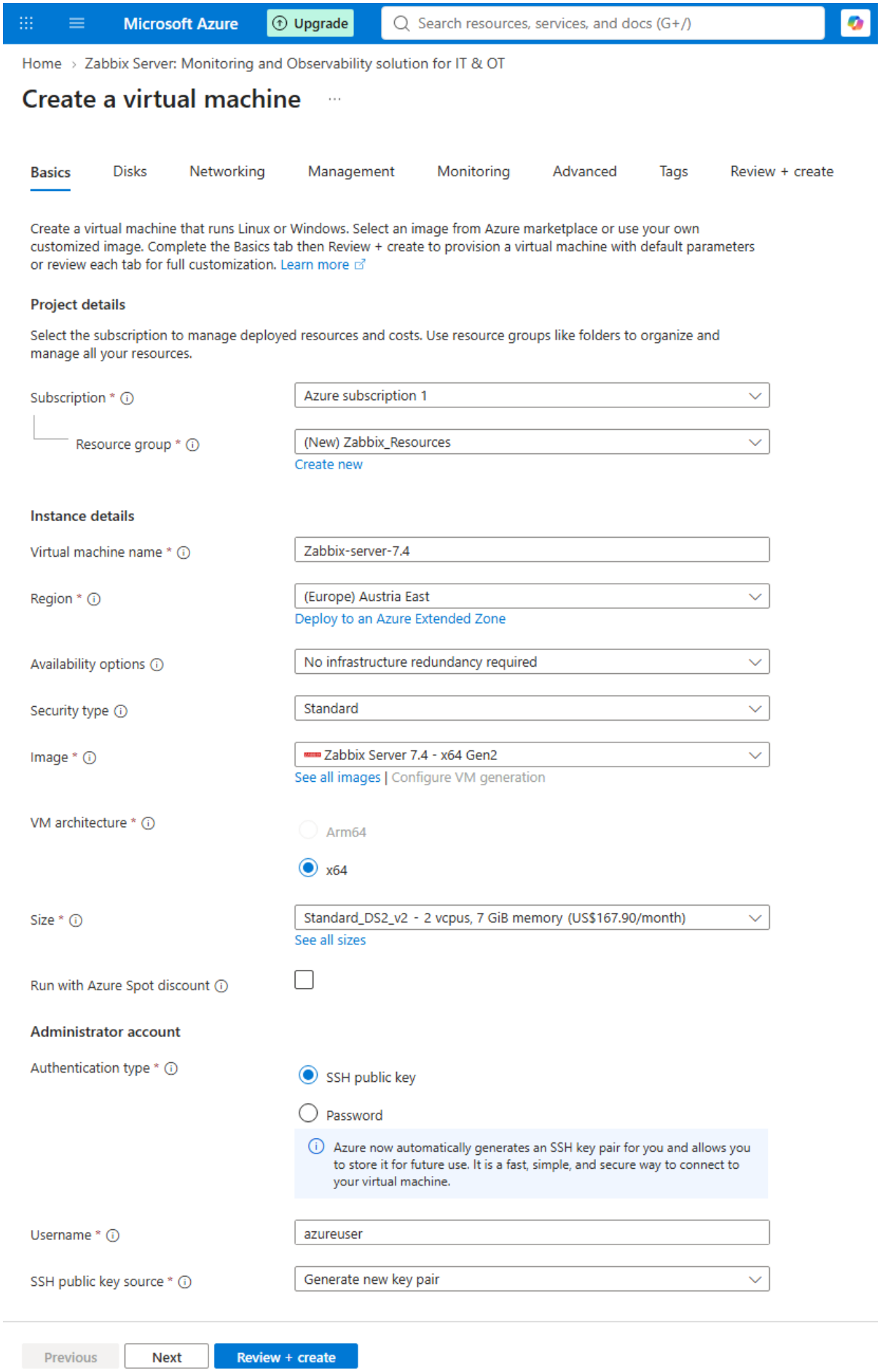


Konfigurieren Sie die VM, indem Sie die VM-Größe auswählen, ein Administratorkonto erstellen und die Einstellungen für Datenträger, Netzwerk, Verwaltung und weitere Bereiche festlegen. Diese Einstellungen können je nach Anforderungen Ihrer Infrastruktur variieren.

Sie können die folgende Konfiguration als Beispiel verwenden:

Registerkarte *Basics*:

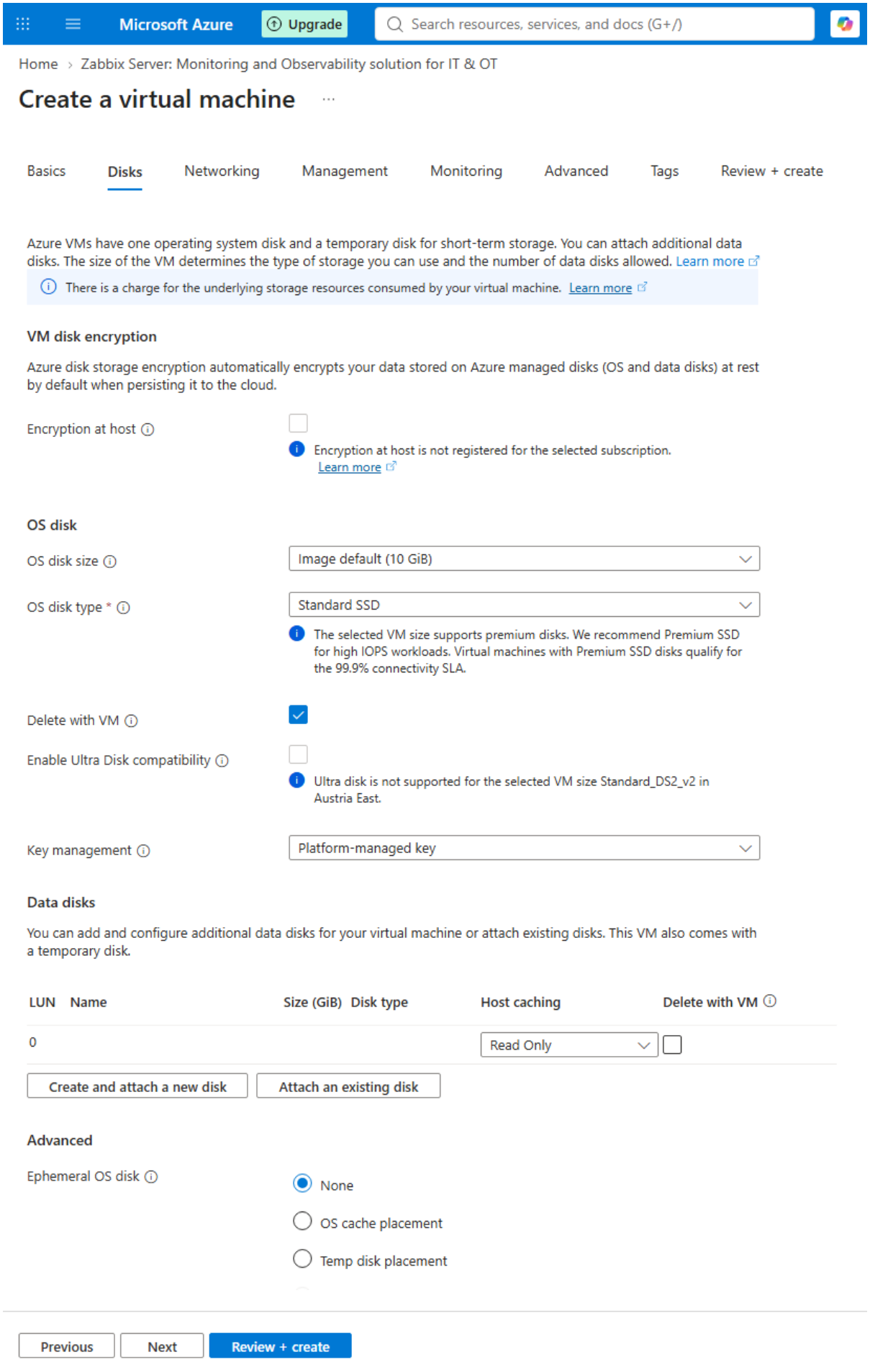
- *Subscription*: Azure subscription 1
- *Resource group*: Zabbix_Resources
- *Virtual machine name*: Zabbix-server--7.4
- *Availability options*: No infrastructure redundancy required
- *Security type*: Standard
- *Image*: Zabbix Server 7.4 - x64 Gen2
- *Size*: Standard_DS2_V2 - 2 vcpus, 7 GiB memory
- *Authentication type*: SSH public key
- *Username*: azureuser
- *SSH public key source*: Generate new key pair



Registerkarte *Disks*:

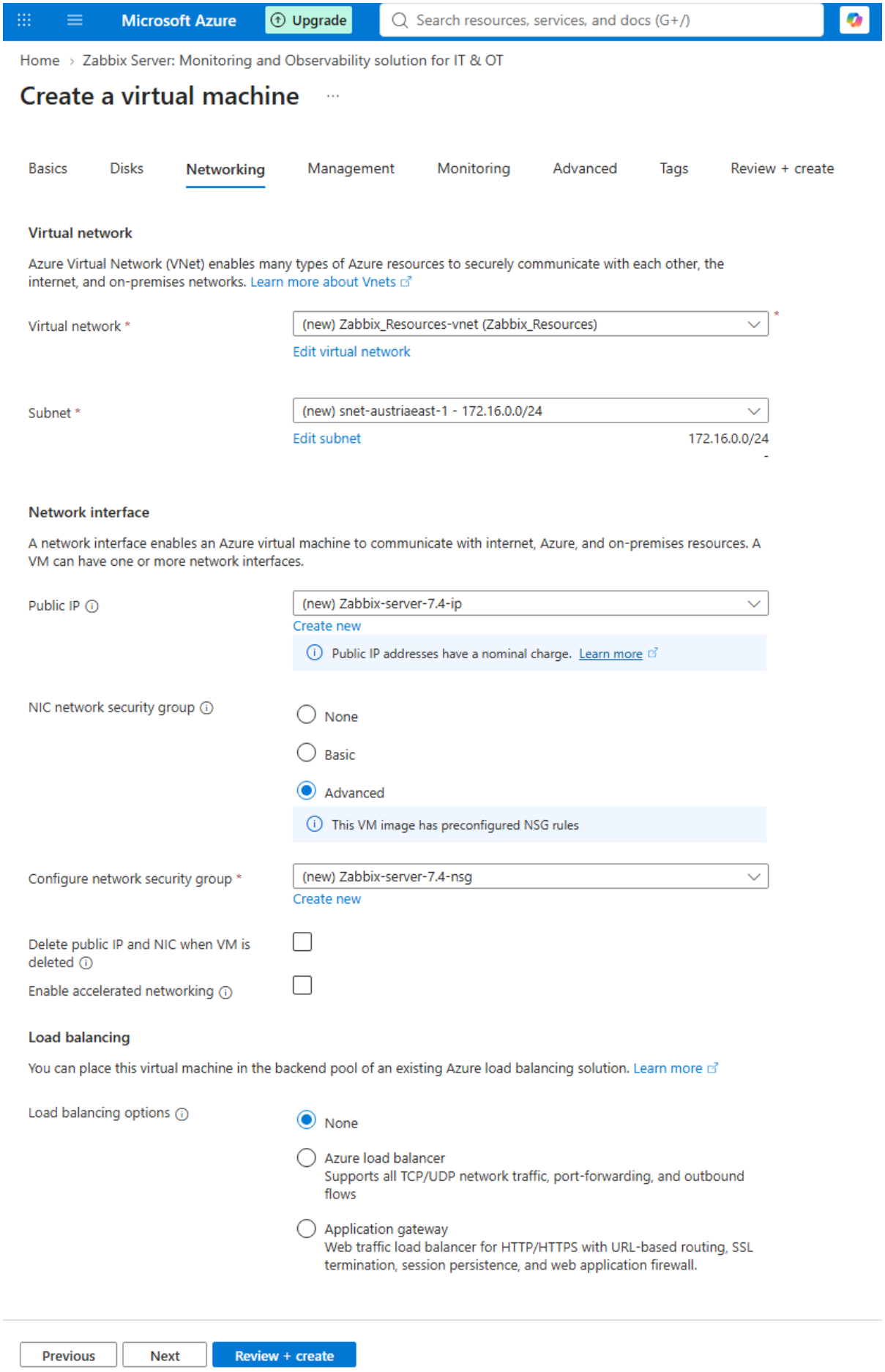
- *Encryption at host*: unchecked
- *OS disk type*: Standard SSD

- *Delete with VM*: checked
- *Enable Ultra Disk compatibility*: unchecked

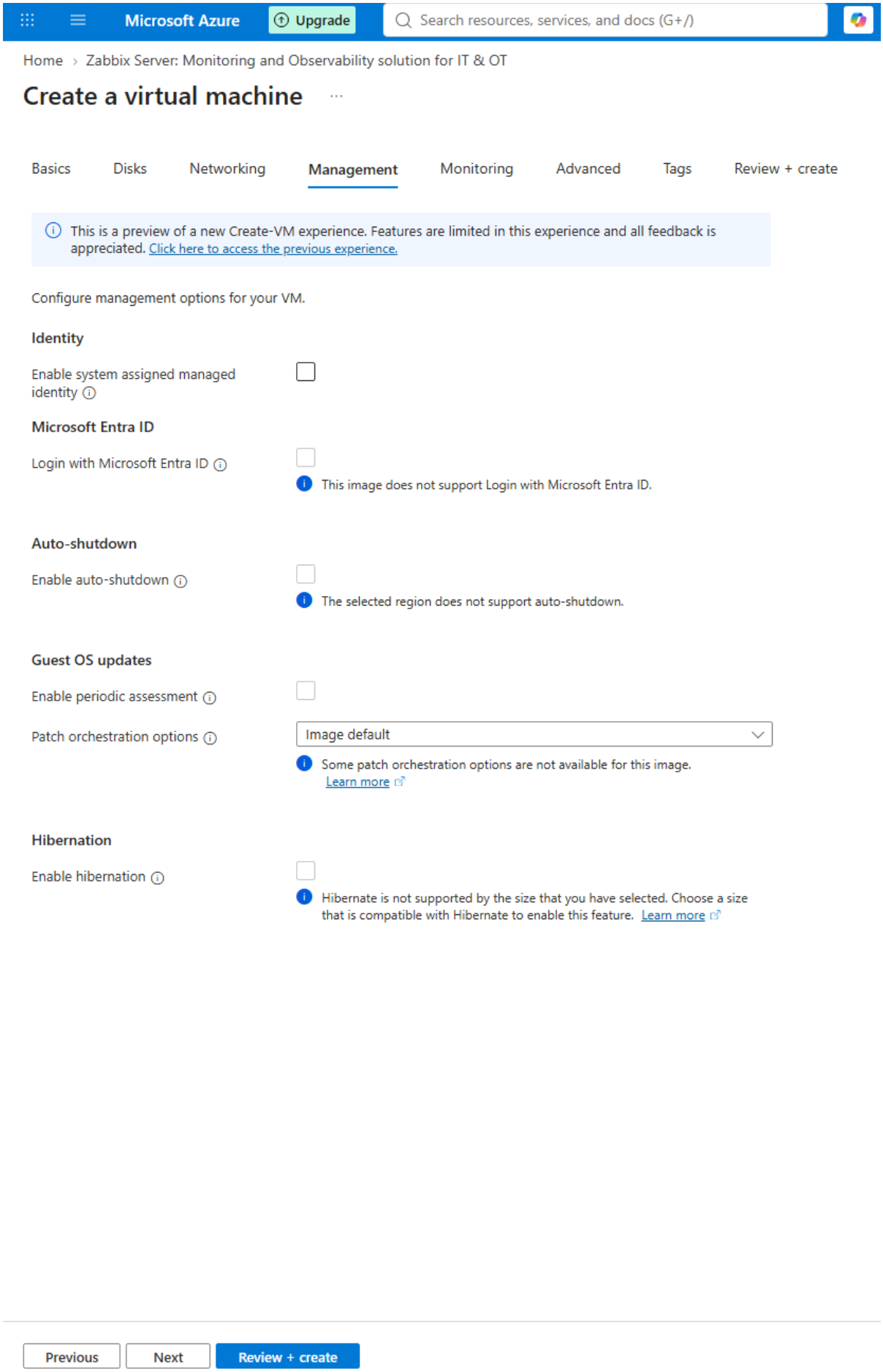


Registerkarte *Networking*:

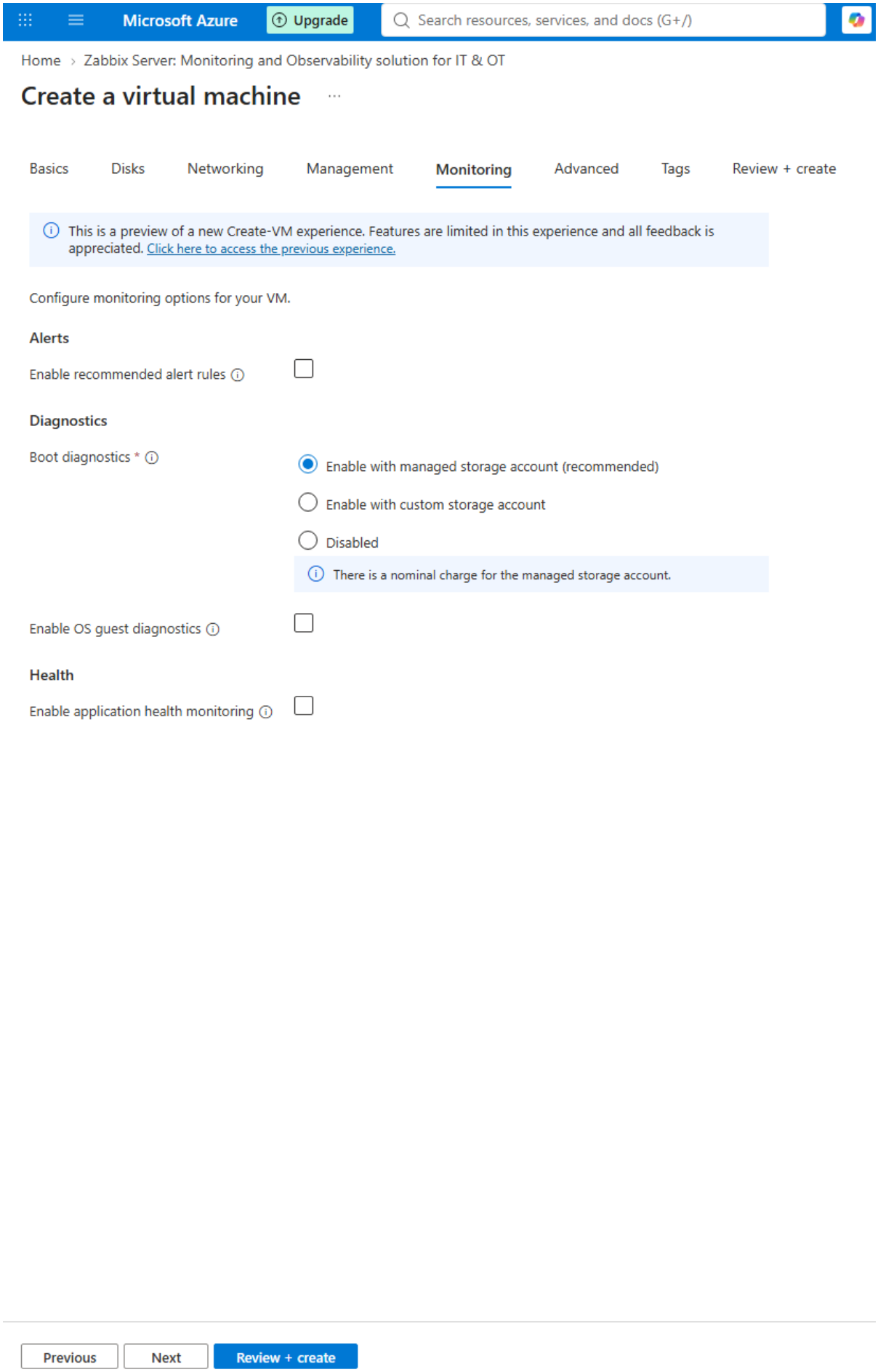
- *Virtual network*: Zabbix_Resources-vnet
- *Public IP*: Zabbix-server-7.4-ip
- *NIC network security group*: Advanced
- *Configure network security group*: Zabbix-server-7.4-nsg



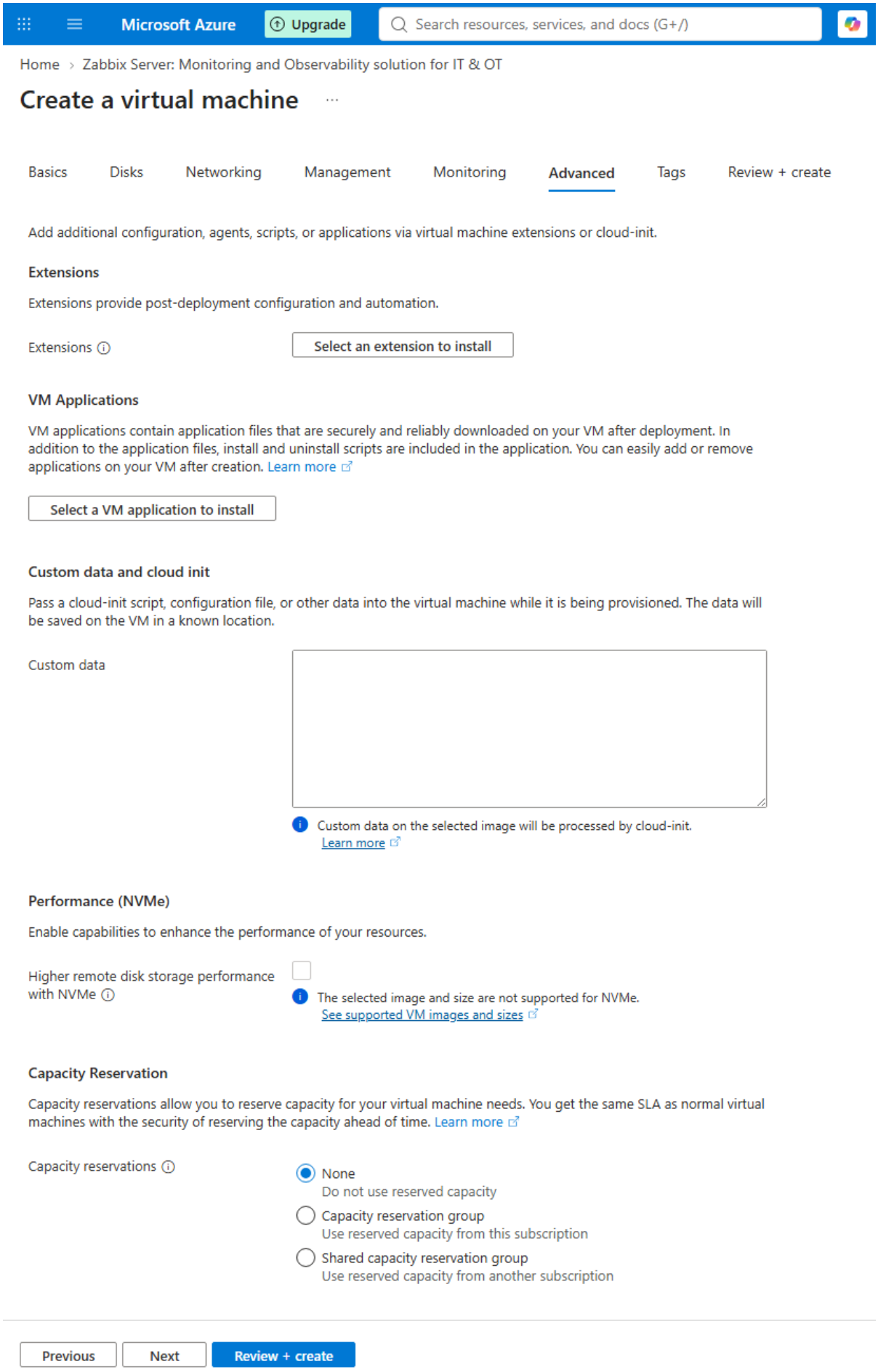
Registerkarte *Management*: Die Parameter können auf den Standardwerten belassen werden.



Registerkarte *Monitoring*: Die Parameter können auf den Standardwerten belassen werden.



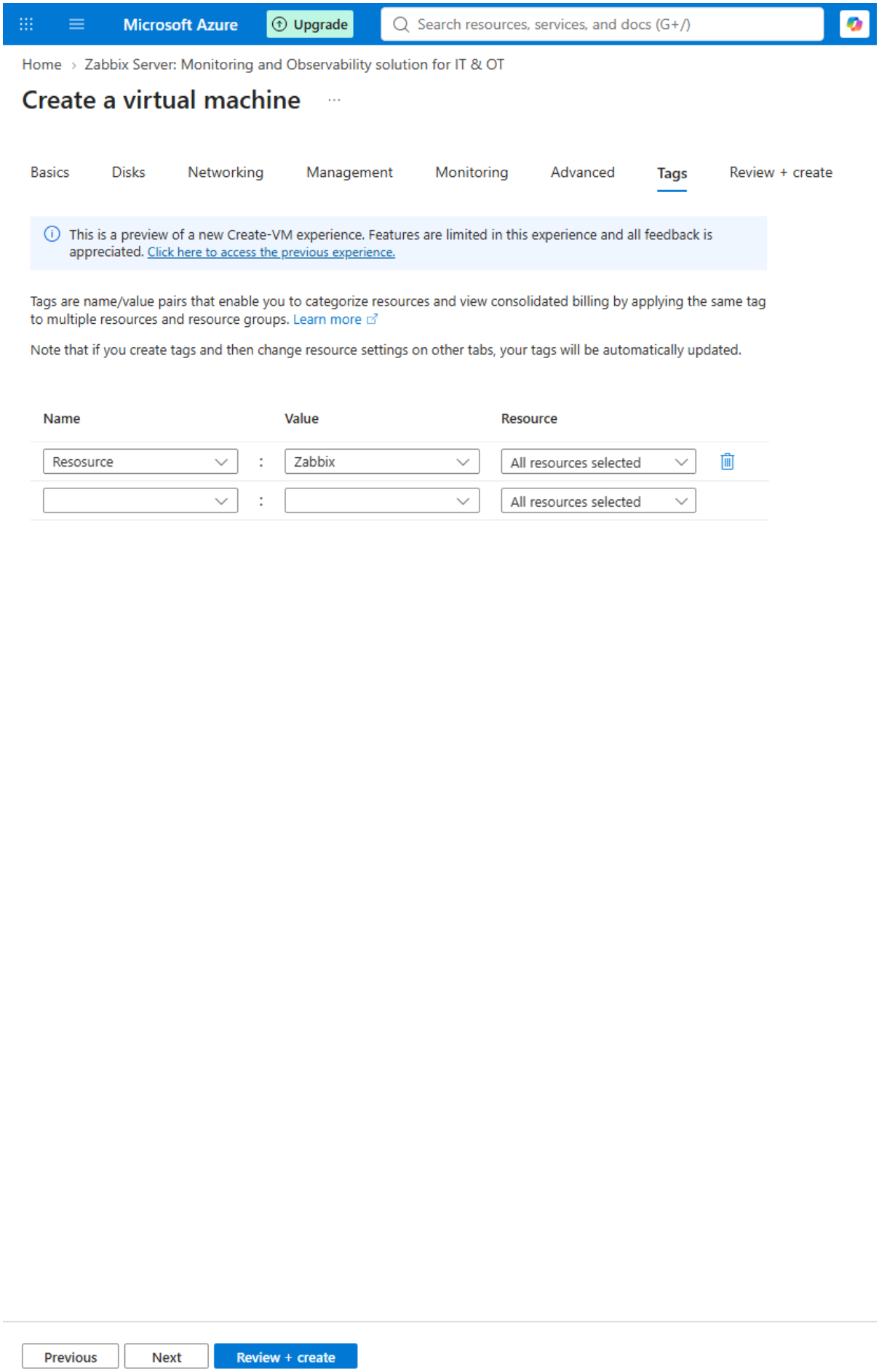
Registerkarte *Advanced*: Die Parameter können auf den Standardwerten belassen werden.



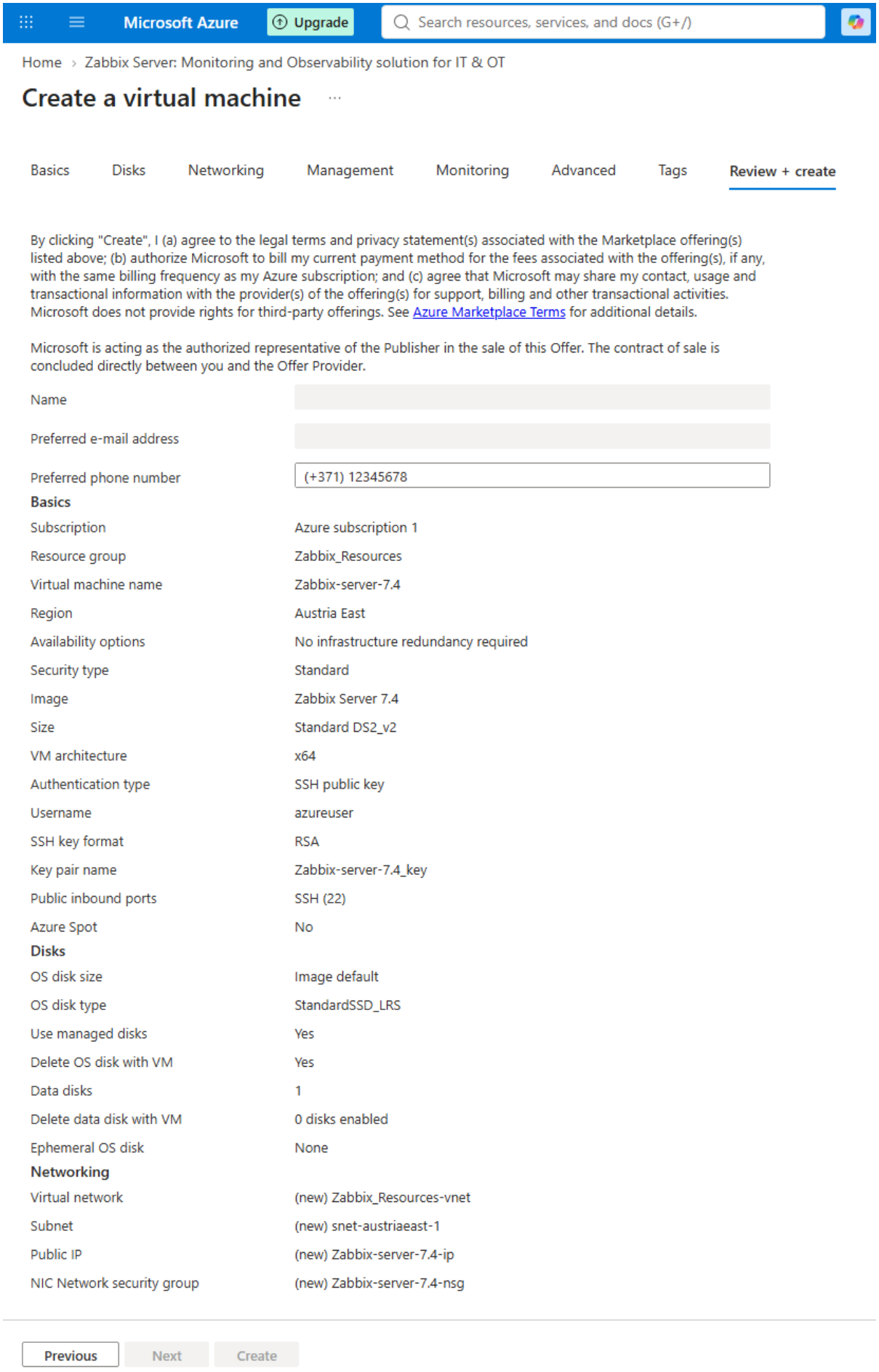
Registerkarte *Tags*:

- *Name*: Resource
- *Value*: Zabbix

- *Resource*: All resource selected



Bestätigen Sie die VM-Details und erstellen Sie die VM.



Sie werden zur Statusseite der VM-Bereitstellung weitergeleitet. Warten Sie, bis die Bereitstellung abgeschlossen ist.

Sobald die Bereitstellung abgeschlossen ist, öffnen Sie die Ressource und notieren Sie sich ihre öffentliche IP-Adresse. Nun müssen Sie per SSH (oder über eine andere verfügbare Methode) eine Verbindung zum Zabbix Server herstellen, um die Anmeldedaten für

das Zabbix Frontend abzurufen.

```
********************************************************************************

Zabbix frontend credentials:

Username: Admin

Password: zfpvagcs0s+srUUK+EM

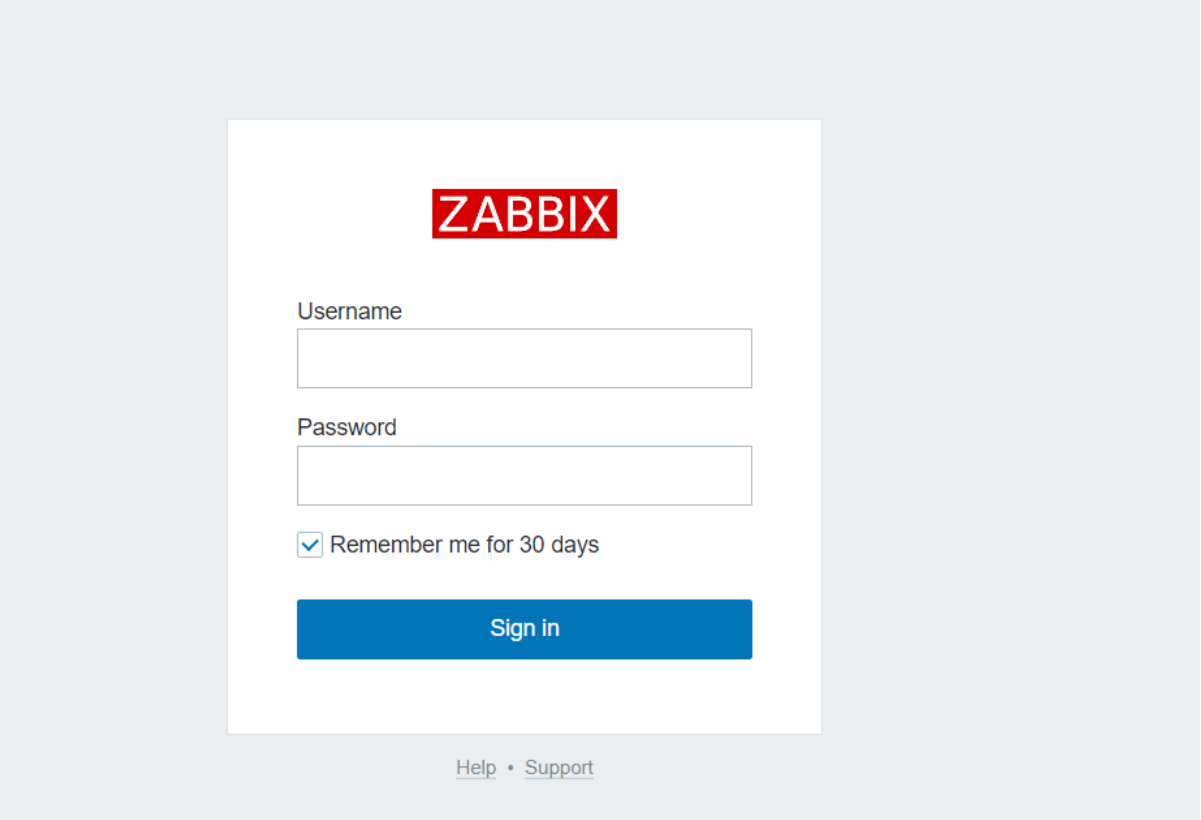

To learn about available professional services, including technical suppport and
 training, please visit https://www.zabbix.com/services

Official Zabbix documentation available at https://www.zabbix.com/documentation/
current/

Note! Do not forget to change timezone PHP variable in /etc/php-fpm.d/zabbix.con
f file.


********************************************************************************
[azureuser@Zabbix-server-6 ~]$
```

Greifen Sie auf das Zabbix Frontend zu, indem Sie die öffentliche IP-Adresse der Zabbix Appliance in einem Webbrowser öffnen. Verwenden Sie zur Anmeldung den im vorherigen Schritt erhaltenen Benutzernamen und das Passwort.



Zugriff auf die Datenbank

Sie können auf die Zabbix-Datenbank mit dem Benutzer `root` zugreifen, ohne ein Passwort einzugeben. Die Anmeldedaten sind in der Konfigurationsdatei `/root/.my.cnf` gespeichert.

Azure-Bereitstellungshandbuch für den Zabbix Proxy

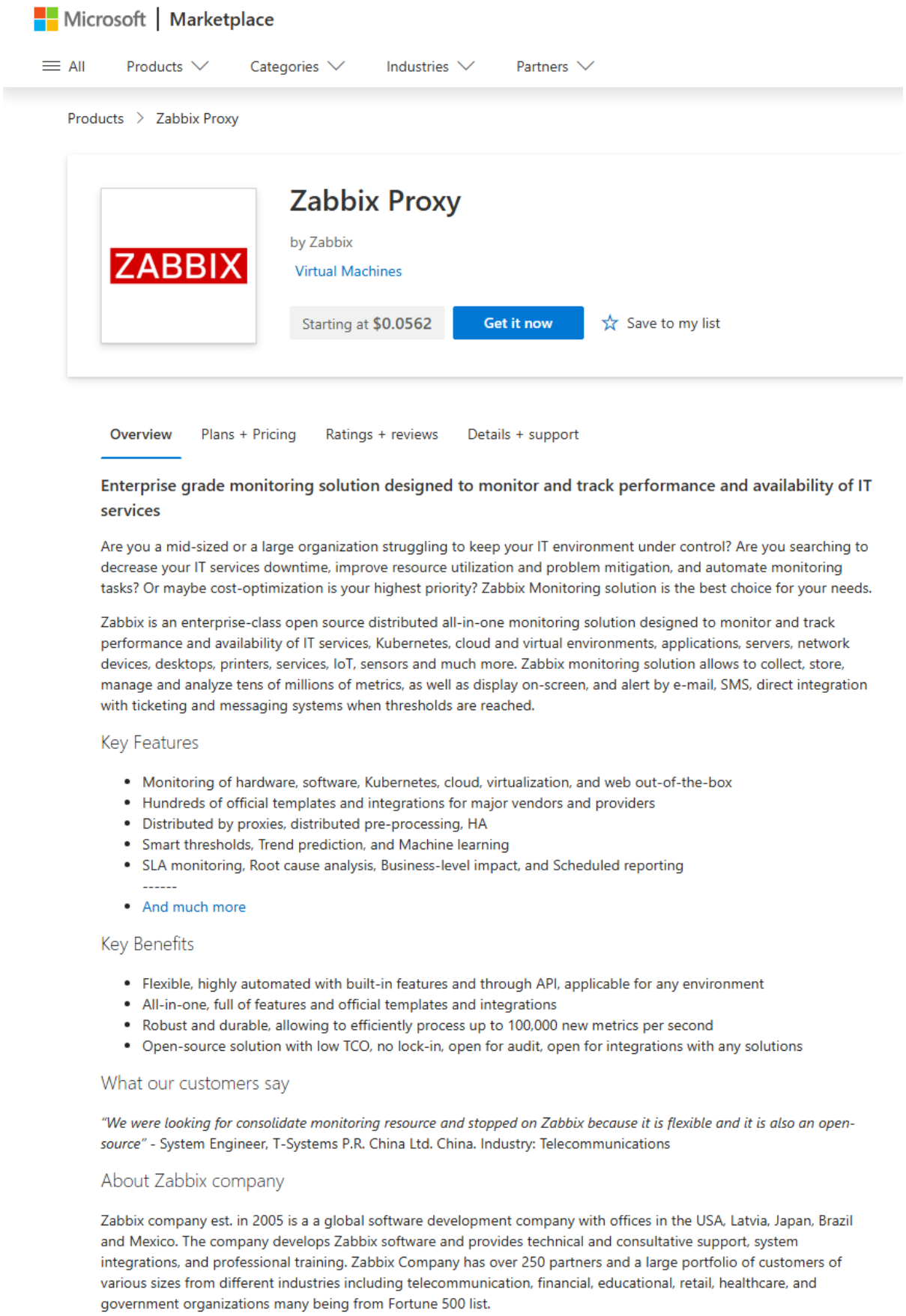
Übersicht

Diese Seite beschreibt, wie Sie einen Zabbix Proxy mithilfe von Azure bereitstellen.

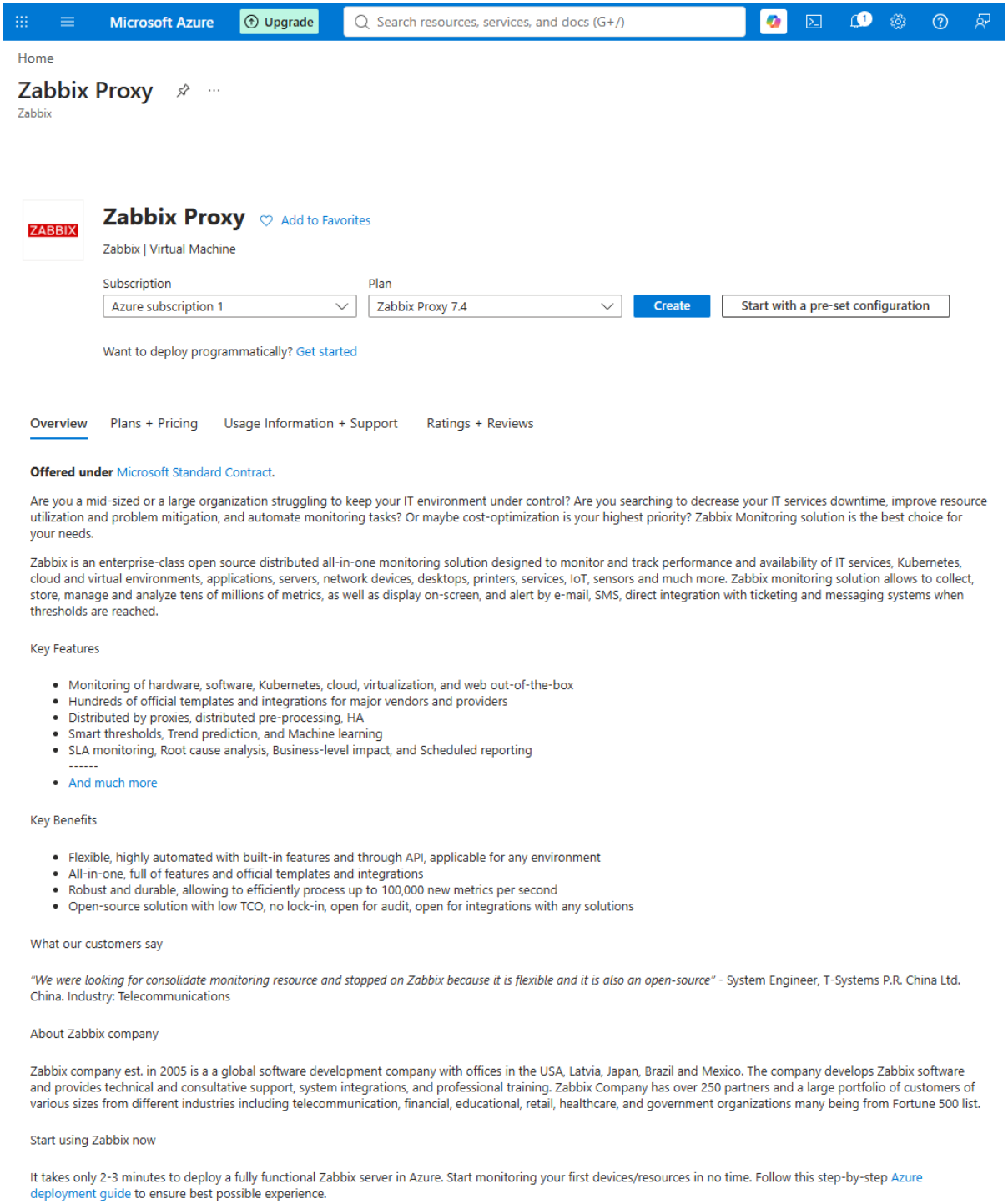
Attention:
Für die Bereitstellung auf der Plattform sind mehrere Zabbix-Versionen verfügbar. Lesen Sie unbedingt die Dokumentation für dieselbe Zabbix-Version, die Sie bereitgestellt haben. Die Dokumentation für verschiedene Zabbix-Versionen kann über das Dropdown-Menü für die Version ausgewählt werden.

Bereitstellung des Zabbix Proxy

Um den Zabbix Proxy bereitzustellen, rufen Sie zunächst die Zabbix Azure Marketplace-Seite auf. Klicken Sie auf *Get it now*.



Hier können Sie zusätzliche Informationen zur aktuellen Bereitstellung lesen, z. B. die allgemeine Übersicht, Preispläne und mehr. Klicken Sie entweder auf *Create* oder auf *Start with a pre-set configuration*.

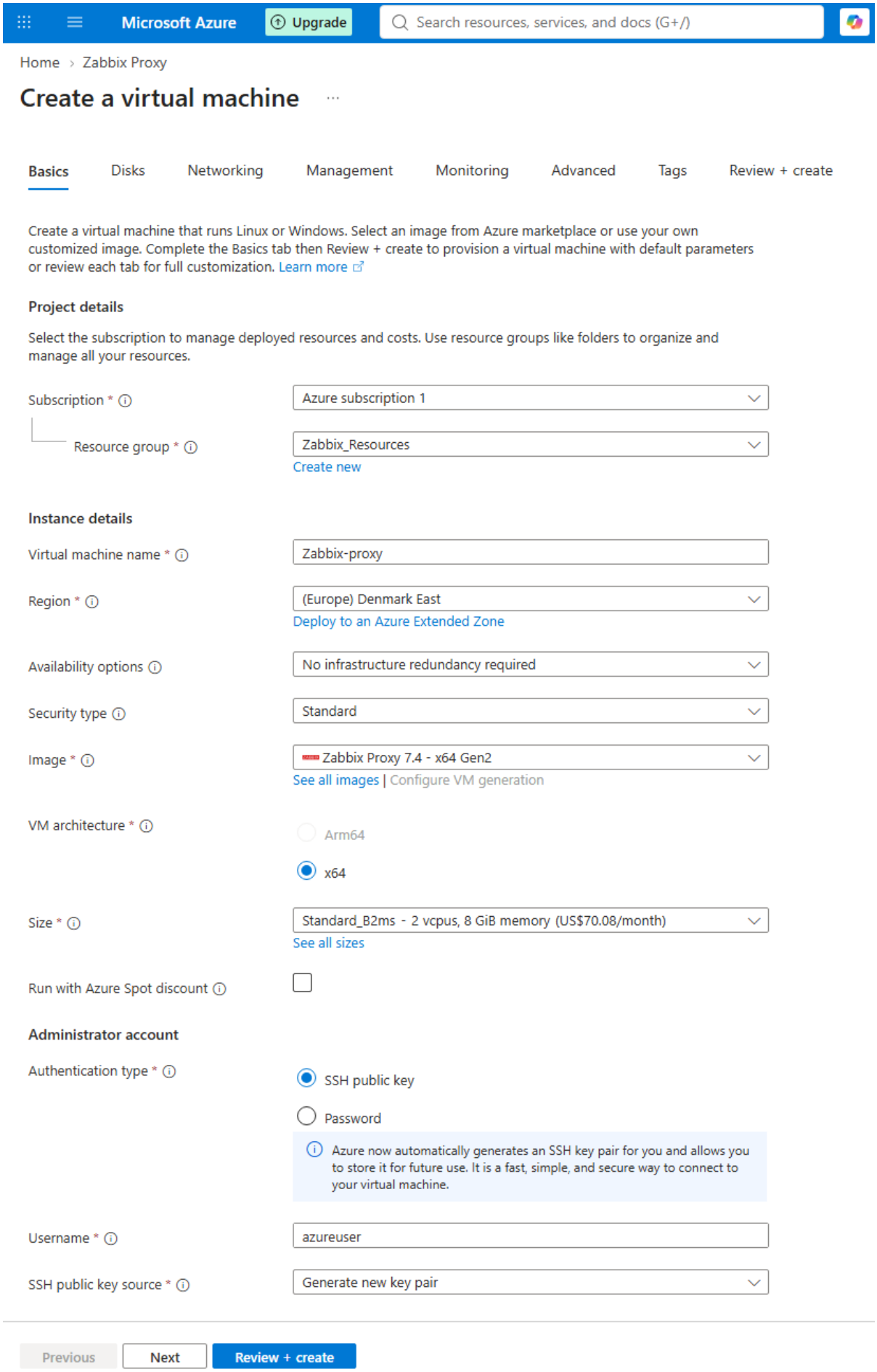


Fahren Sie mit der detaillierten VM-Konfiguration fort: Wählen Sie die VM-Größe, das Administratorkonto, den Datenträger, das Netzwerk, die Verwaltung und weitere Einstellungen aus. Diese Einstellungen können je nach Ihren Infrastrukturanforderungen variieren.

Sie können die folgende Konfiguration als Beispiel verwenden:

Registerkarte „Basics“:

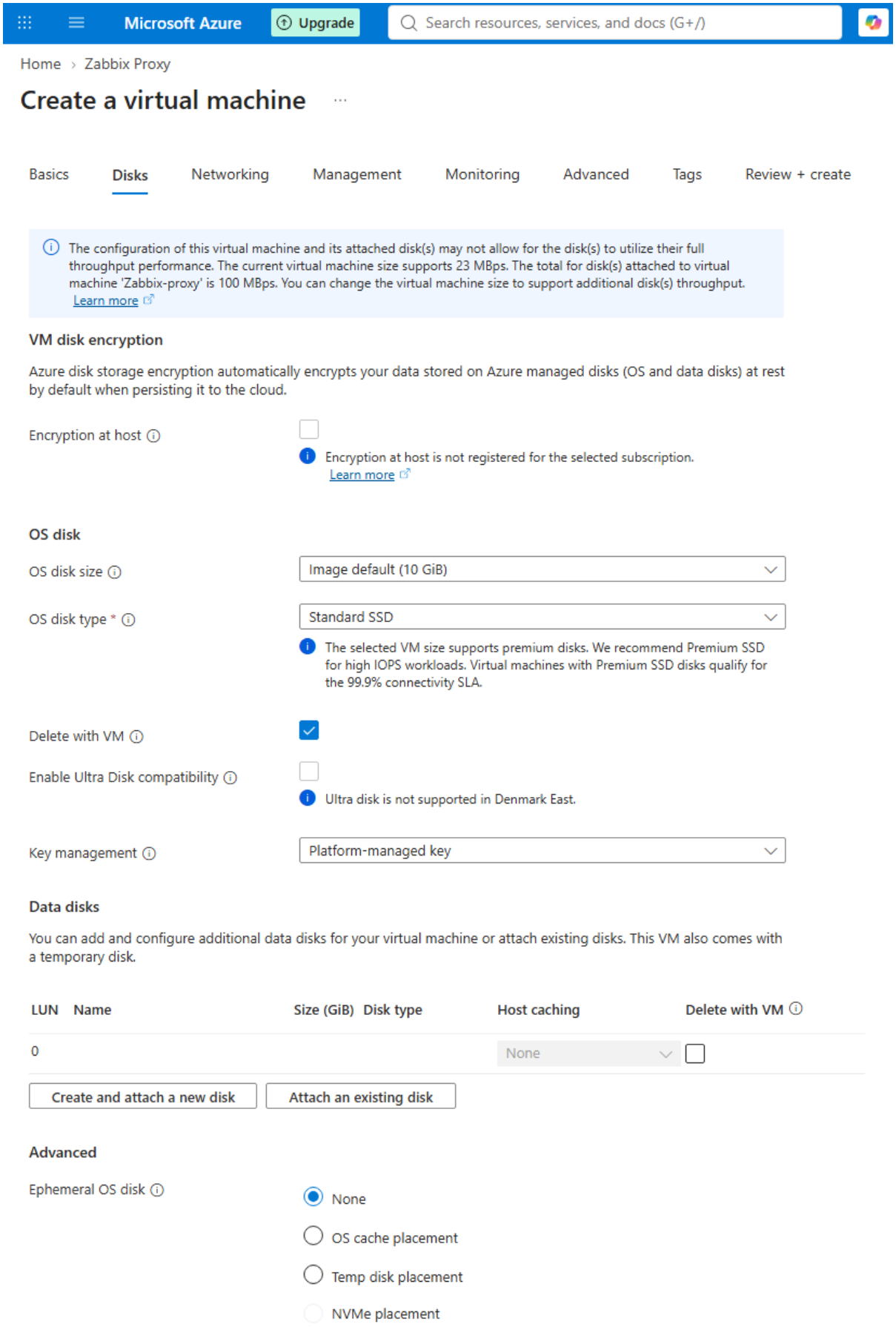
- *Subscription*: Azure subscription 1
- *Resource group*: Zabbix_Resources
- *Virtual machine name*: Zabbix-proxy
- *Availability options*: No infrastructure redundancy required
- *Security type*: Standard
- *Image*: Zabbix Proxy 7.4 - x64 Gen2
- *Size*: Standard_B2ms - 2 vcpus, 8 GiB memory
- *Authentication type*: SSH public key
- *Username*: azureuser
- *SSH public key source*: Generate new key pair



Registerkarte „Disks“:

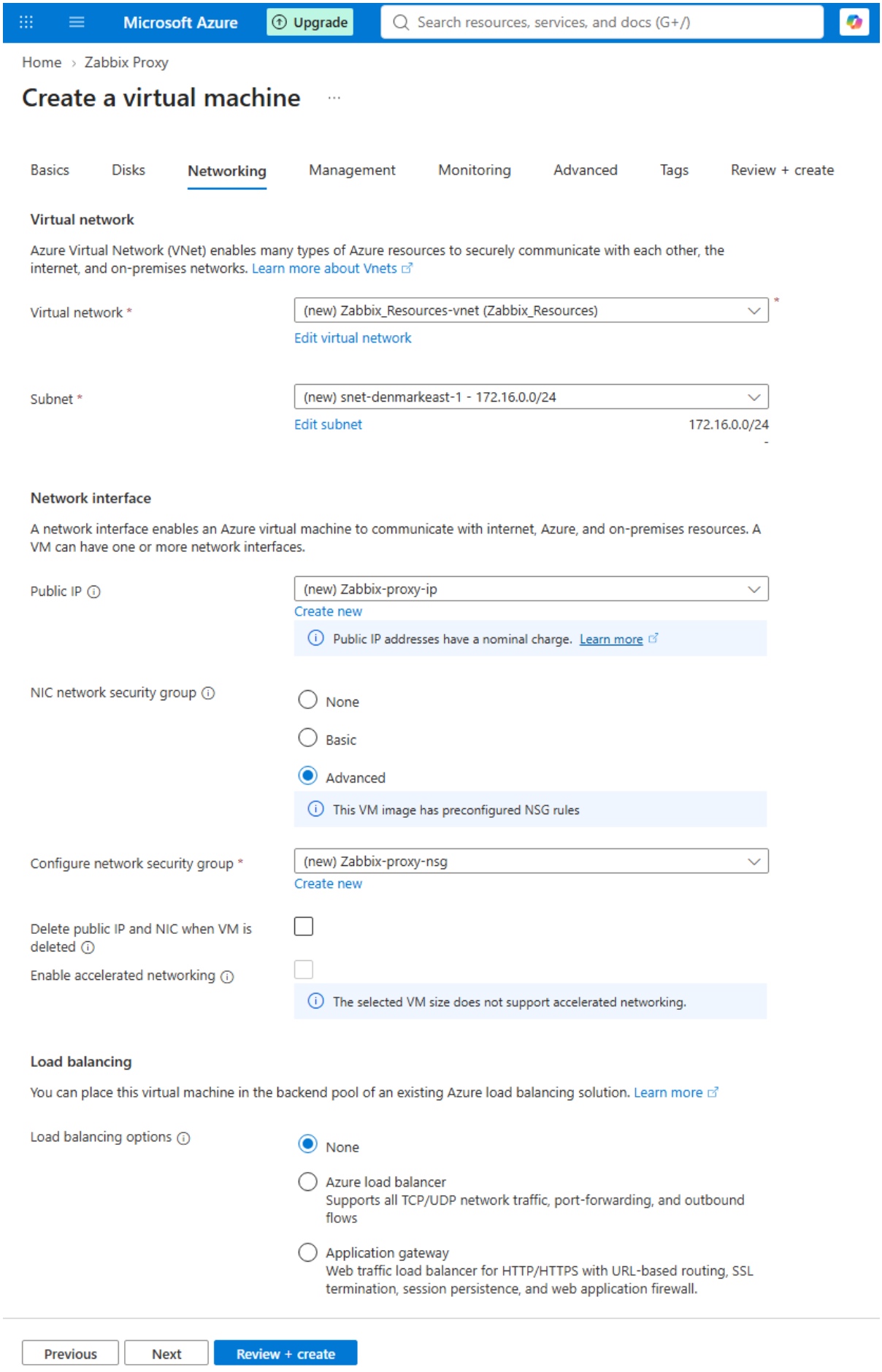
- *Encryption at host*: unchecked

- *OS disk type*: Standard SSD
- *Delete with VM*: checked
- *Enable Ultra Disk compatibility*: unchecked

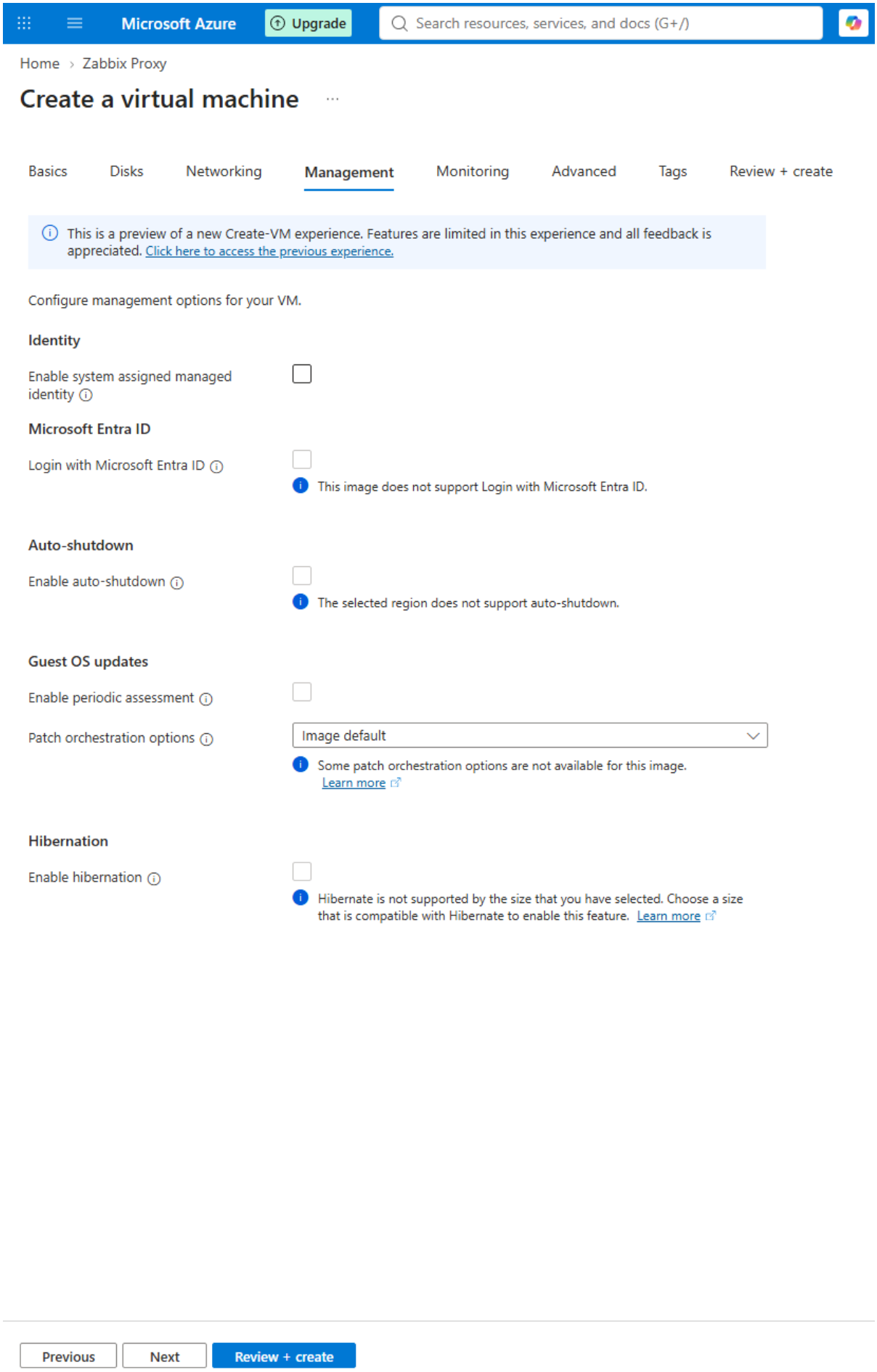


Registerkarte „Networking“:

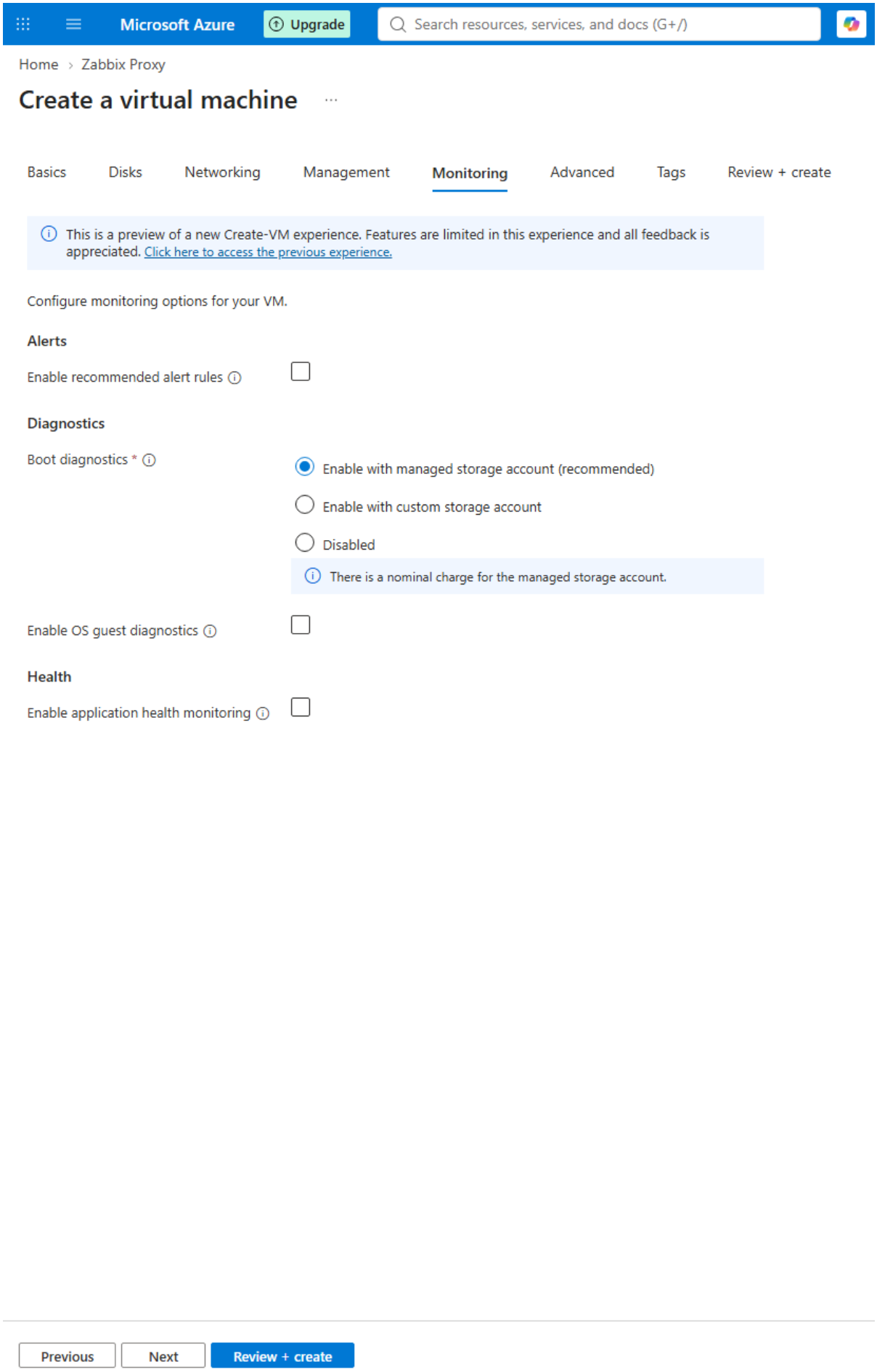
- *Virtual network*: Zabbix_Resources-vnet
- *Public IP*: Zabbix-proxy-ip
- *NIC network security group*: Advanced
- *Configure network security group*: Zabbix-proxy-nsg



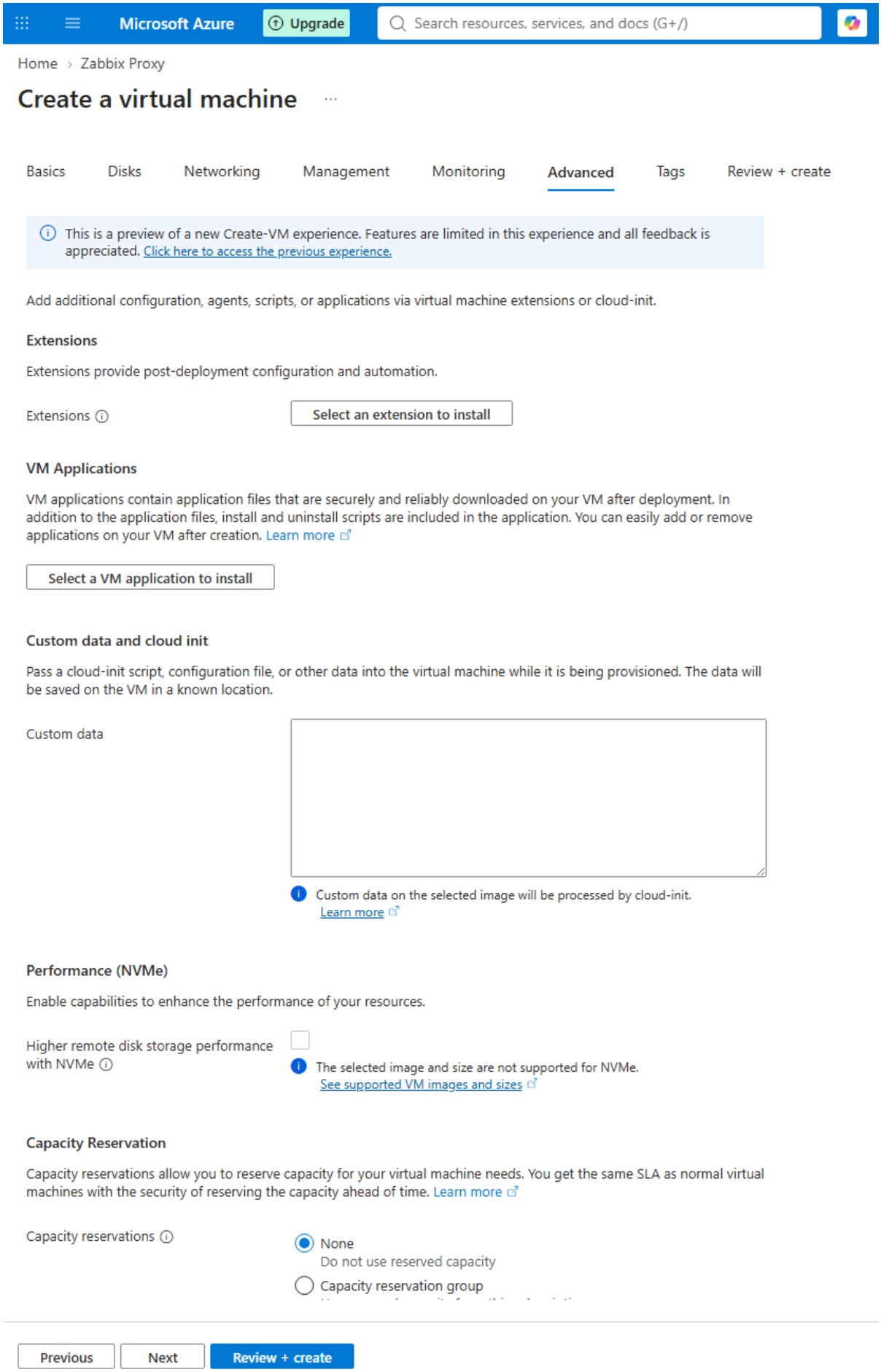
Registerkarte „Management“: Die Parameter können auf den Standardwerten belassen werden.



Registerkarte „Monitoring“: Die Parameter können auf den Standardwerten belassen werden.



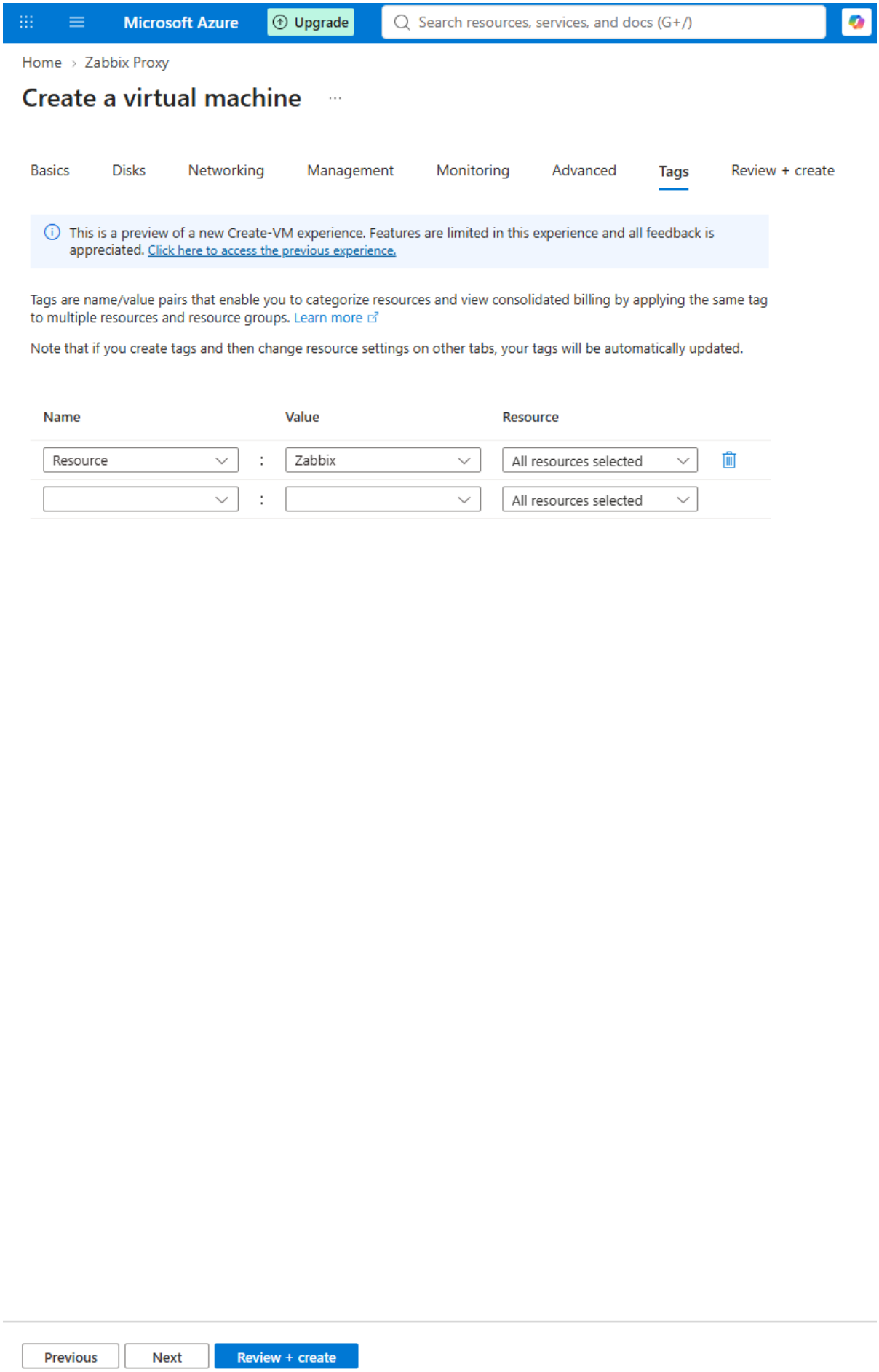
Registerkarte „Advanced“: Die Parameter können auf den Standardwerten belassen werden.



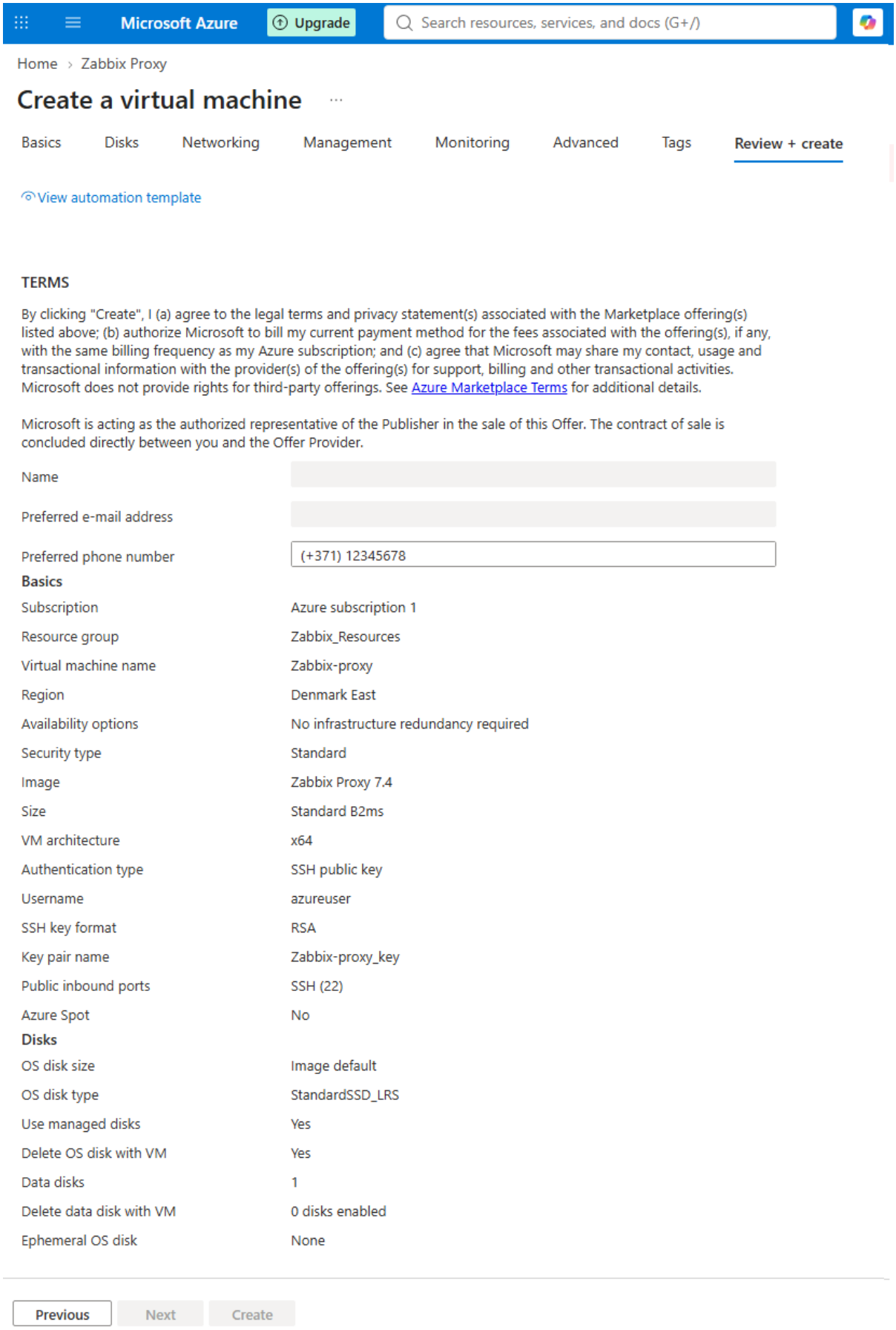
Registerkarte „Tags“:

- *Name*: Resource
- *Value*: Zabbix

- *Resource*: All resource selected



Bestätigen Sie die VM-Details und erstellen Sie die VM.



Sie werden zur Statusseite der VM-Bereitstellung weitergeleitet. Warten Sie, bis die Bereitstellung abgeschlossen ist.

Öffnen Sie nach Abschluss der Bereitstellung die Ressource und notieren Sie sich ihre öffentliche IP-Adresse. Stellen Sie über SSH und mit den während der VM-Bereitstellung erstellten Administratoranmeldedaten eine Verbindung zum Zabbix Proxy her, um den Proxy zu konfigurieren.

```
*******************************************************************************

To learn about available professional services, including technical suppport and
 training, please visit https://www.zabbix.com/services

Official Zabbix documentation available at https://www.zabbix.com/documentation/
current/

*******************************************************************************
[azureuser@Zabbix-proxy ~]$
```

Die Konfigurationsdatei des Zabbix Proxy befindet sich in `/etc/zabbix/zabbix_proxy.conf`.

Die anfängliche Proxy-Konfiguration besteht darin, die Parameter `ProxyMode`, `Server` und `Hostname` in der Konfigurationsdatei des Zabbix Proxy zu ändern.

Der Proxy-Modus kann durch Bearbeiten des Parameters `ProxyMode` geändert werden. Setzen Sie `ProxyMode=0` für den aktiven Modus oder `ProxyMode=1` für den passiven Modus.

Der Parameter `Server` sollte die IP-Adresse des Zabbix Server enthalten.

Der Parameter `Hostname` sollte den Hostnamen des Zabbix Proxy enthalten. Der hier definierte Hostname muss mit dem während der Proxy-Konfiguration im Zabbix Frontend angegebenen Proxy-Namen übereinstimmen.

```
############ GENERAL PARAMETERS #################

### Option: ProxyMode
#       Proxy operating mode.
#       0 - proxy in the active mode
#       1 - proxy in the passive mode
#
# Mandatory: no
# Default:
# ProxyMode=0

### Option: Server
#       If ProxyMode is set to active mode:
#               IP address or DNS name (address:port) or cluster (address:port;address2:port) of Zabbix server to get
configuration data from and send data to.
#               If port is not specified, default port is used.
#               Cluster nodes need to be separated by semicolon.
#       If ProxyMode is set to passive mode:
#               List of comma delimited IP addresses, optionally in CIDR notation, or DNS names of Zabbix server.
#               Incoming connections will be accepted only from the addresses listed here.
#               If IPv6 support is enabled then '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' are treated equally
#               and '::/0' will allow any IPv4 or IPv6 address.
#               '0.0.0.0/0' can be used to allow any IPv4 address.
#               Example: Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
#
# Mandatory: yes
# Default:
# Server=

Server=127.0.0.1
```

```
### Option: Hostname
#       Unique, case sensitive Proxy name. Make sure the Proxy name is known to the server!
#       Value is acquired from HostnameItem if undefined.
#
# Mandatory: no
# Default:
# Hostname=

Hostname=Zabbix proxy
```

Öffnen Sie das Zabbix Frontend und navigieren Sie zu *Administration* > *Proxies*. Klicken Sie auf die Schaltfläche *Create proxy*.

Geben Sie denselben Proxy-Namen wie den Wert des Parameters *Hostname* in der Proxy-Konfigurationsdatei ein.

Wählen Sie den Proxy-Modus aus und geben Sie die IP-Adresse des Proxy an (optional für aktive Proxys).

Klicken Sie auf die Schaltfläche *Create proxy*, um einen Proxy zu erstellen.

Speicherort der Proxy-Datenbank

Die SQLite-Datenbank des Proxy befindet sich unter `/tmp/zabbix_proxy.sqlite3` und wird durch den Parameter `DBName` in der Konfigurationsdatei des Zabbix Proxy angegeben.

Leitfaden zur Bereitstellung in Google Cloud

Übersicht

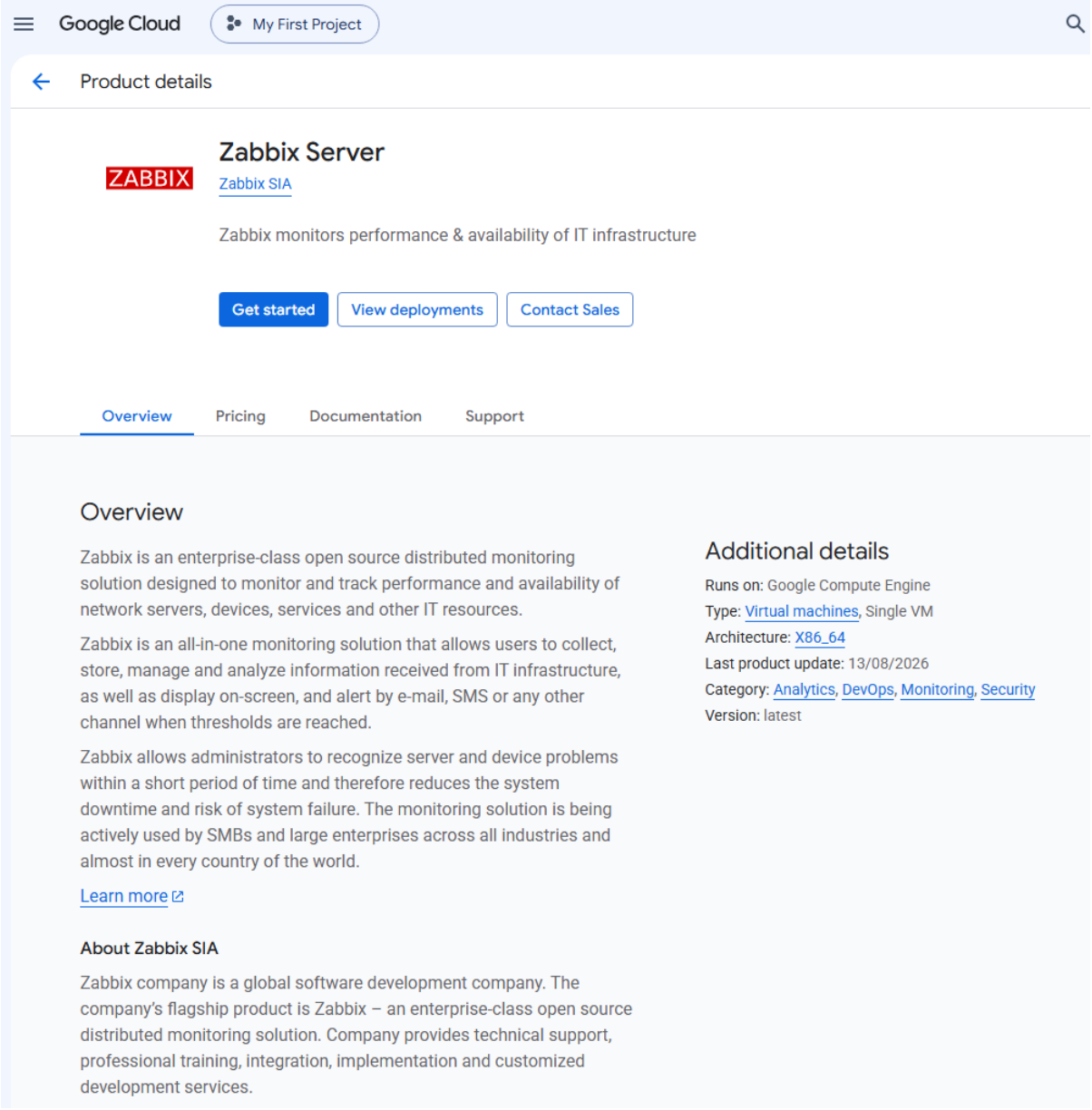
Auf dieser Seite wird beschrieben, wie Zabbix über den Google Cloud Marketplace bereitgestellt wird.

Attention:
Für die Bereitstellung auf der Plattform sind mehrere Zabbix-Versionen verfügbar. Lesen Sie unbedingt die Dokumentation für dieselbe Zabbix-Version, die Sie bereitgestellt haben. Die Dokumentation für verschiedene Zabbix-Versionen kann über das Dropdown-Menü für Versionen ausgewählt werden.

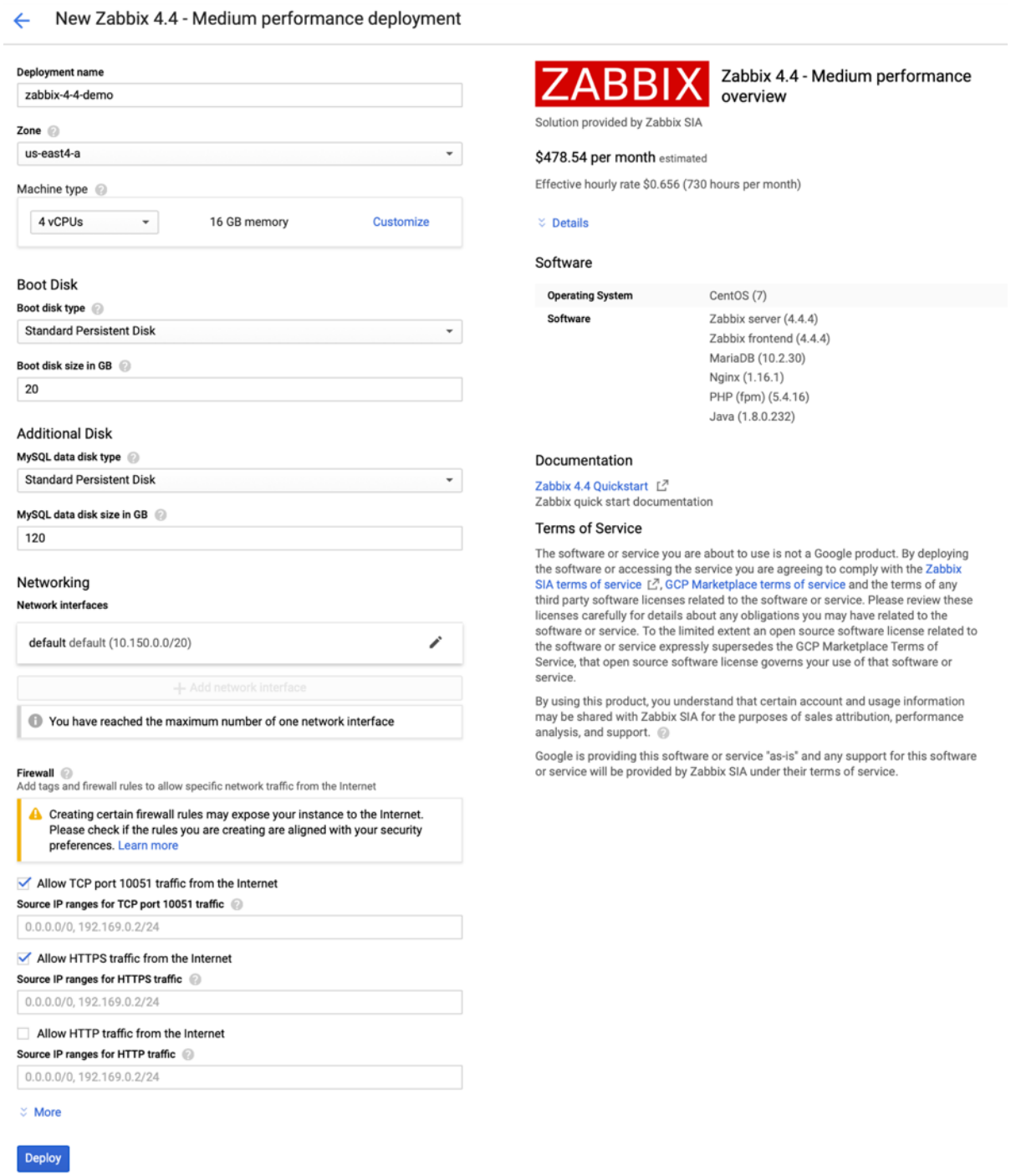
Appliance bereitstellen

Stellen Sie zunächst die Zabbix Appliance bereit.
Mit dem Google Cloud Marketplace ist das ganz einfach.
Rufen Sie im Marketplace einfach das Produkt Zabbix Server auf.

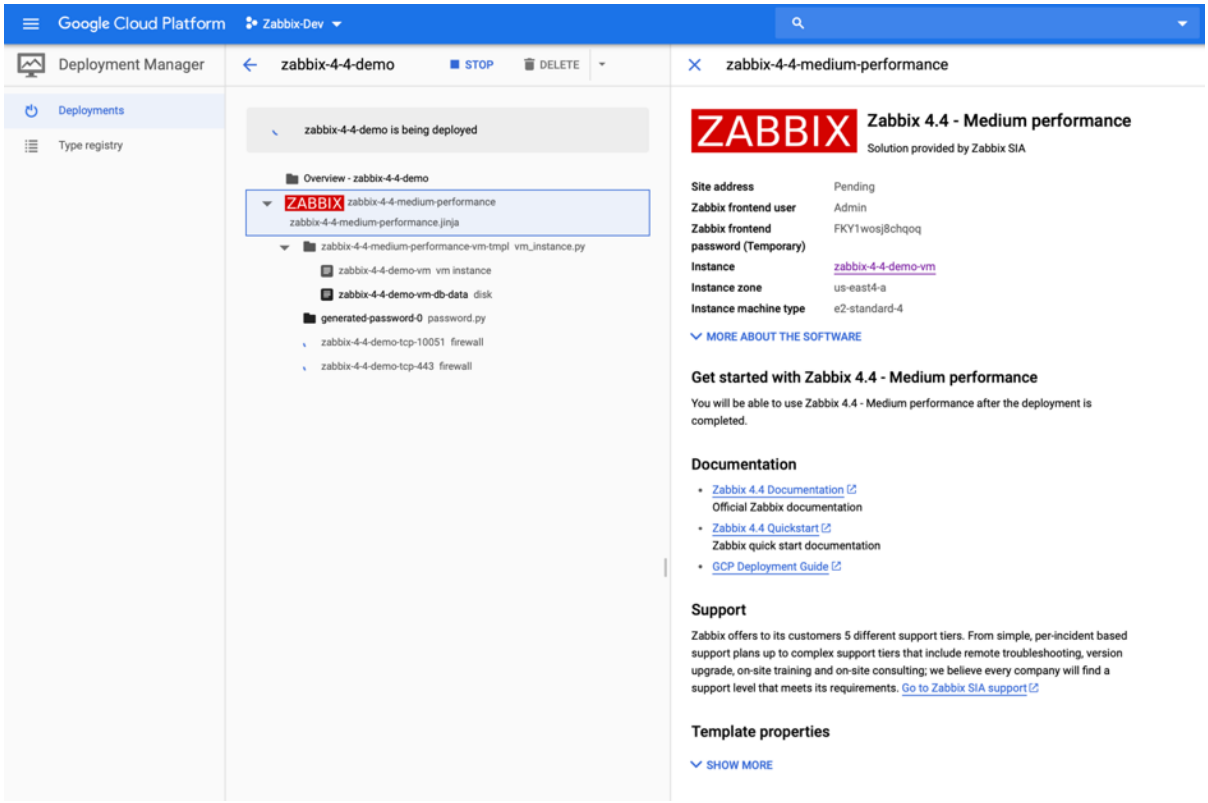
Klicken Sie auf *Get started*.



Sie können die Standardeinstellungen übernehmen oder sie anpassen.
Klicken Sie anschließend auf *Deploy*.



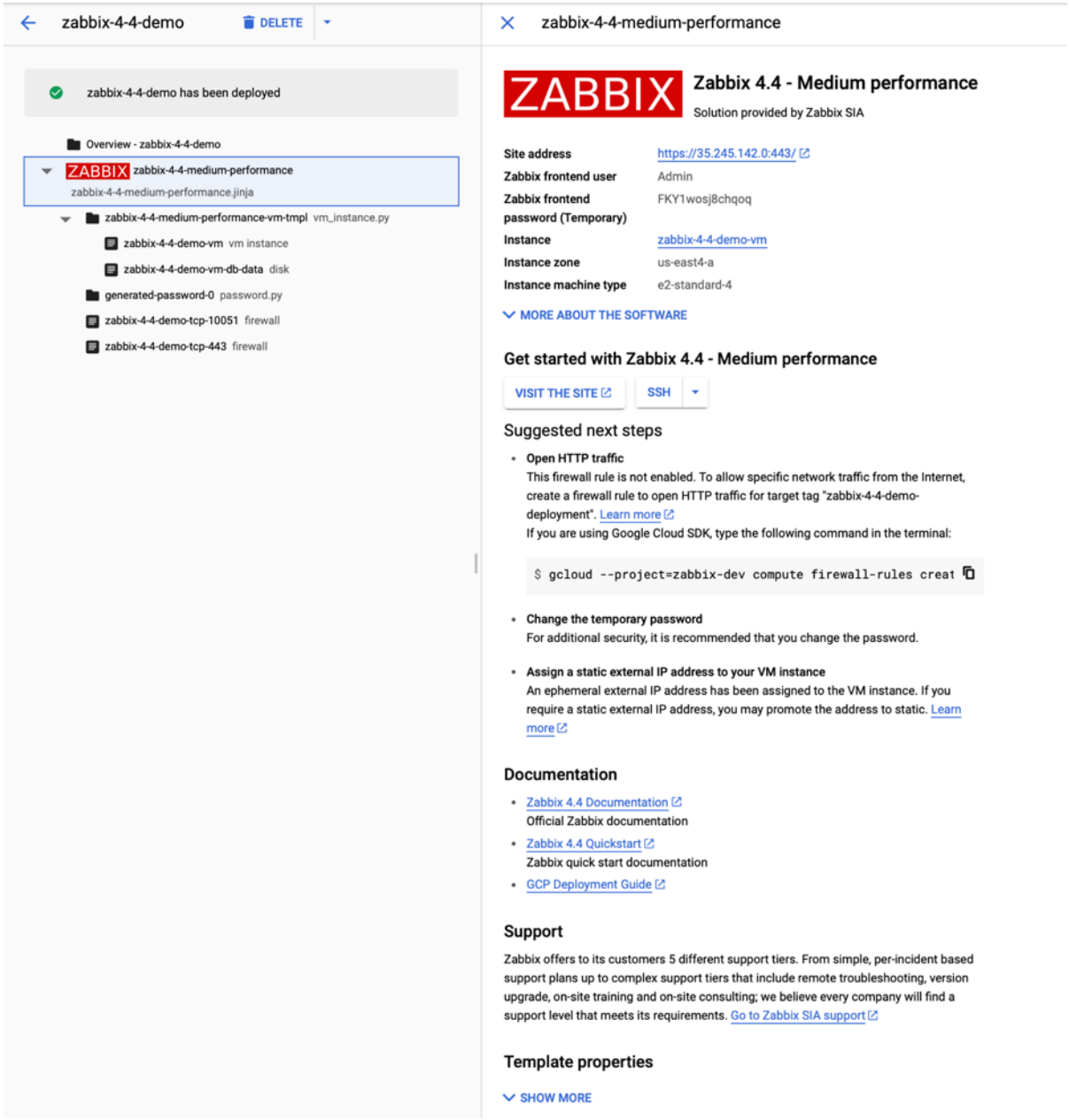
Die Bereitstellung läuft nun.
Warten Sie, bis sie abgeschlossen ist.



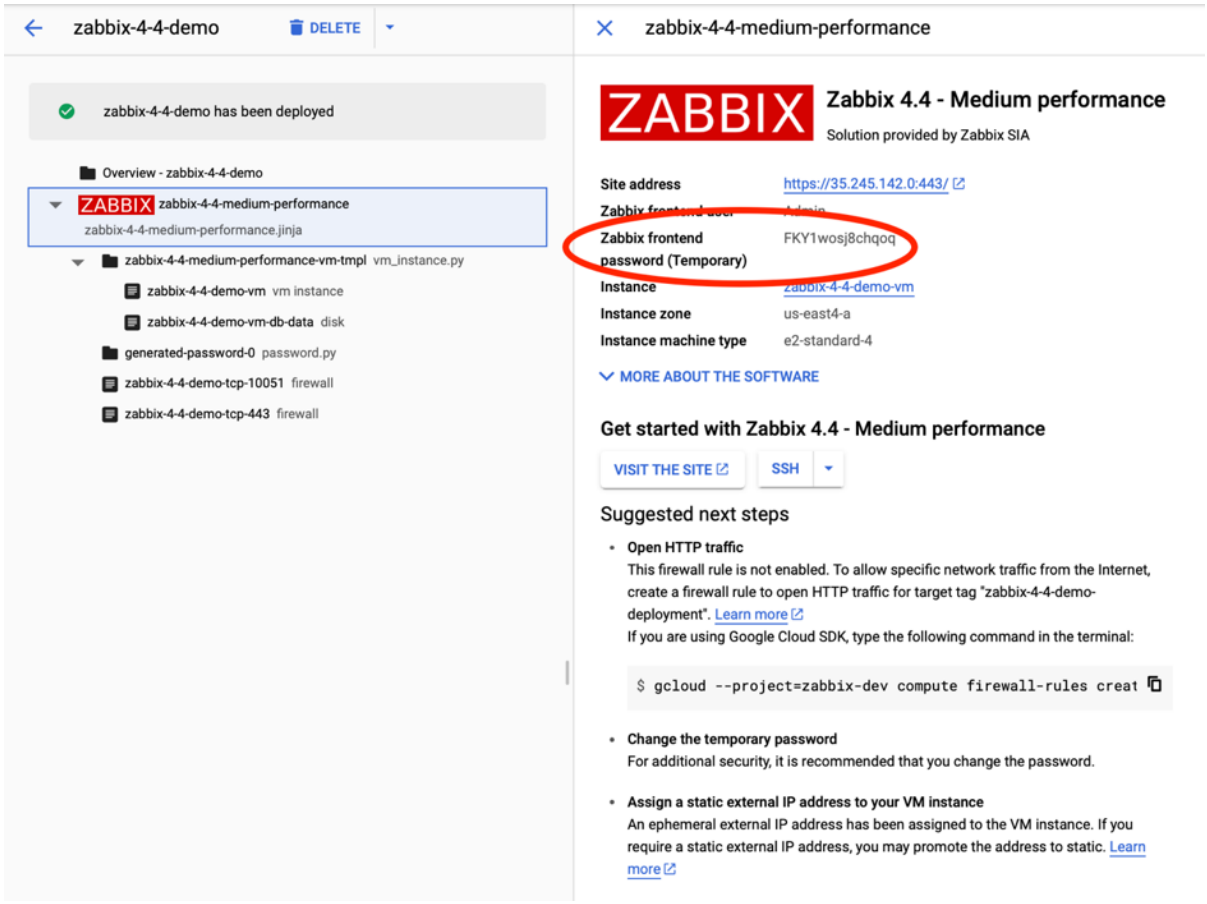
Das war’s! Die Bereitstellung wurde nun gestartet.

Zugriff auf die Appliance

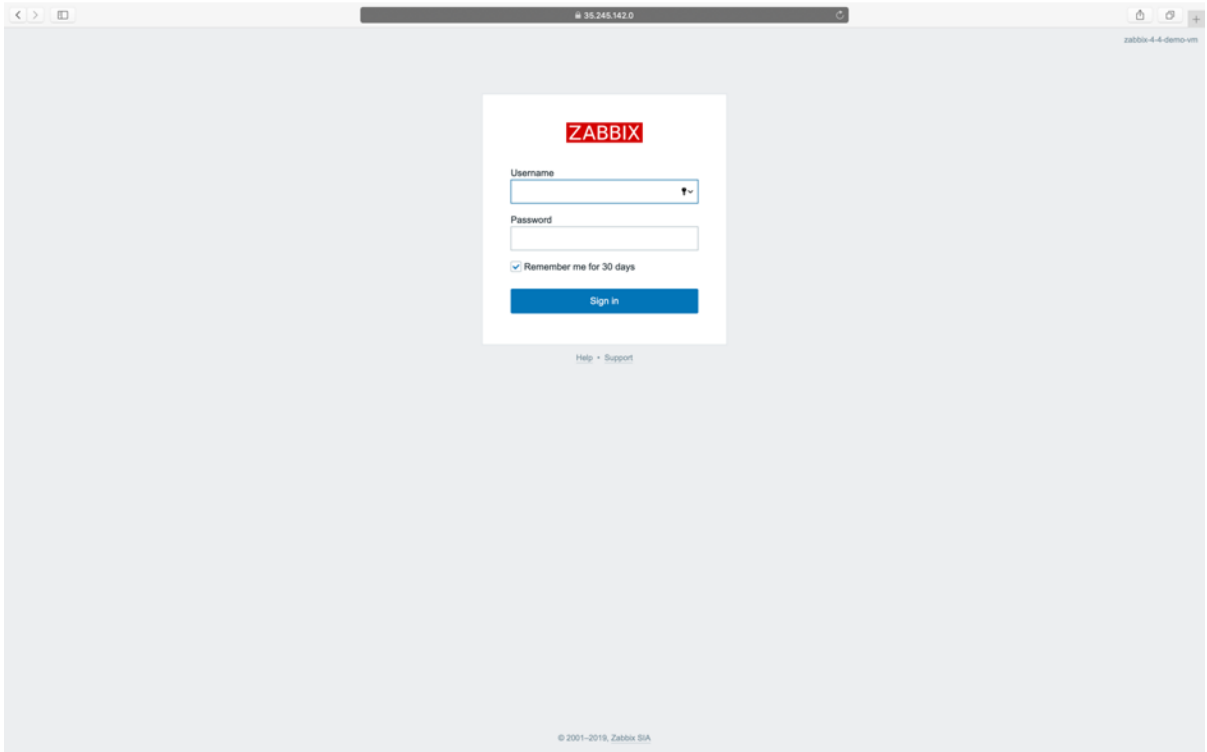
Nach Abschluss der Bereitstellung sollte eine Meldung angezeigt werden, die angibt, dass die Appliance bereitgestellt wird.

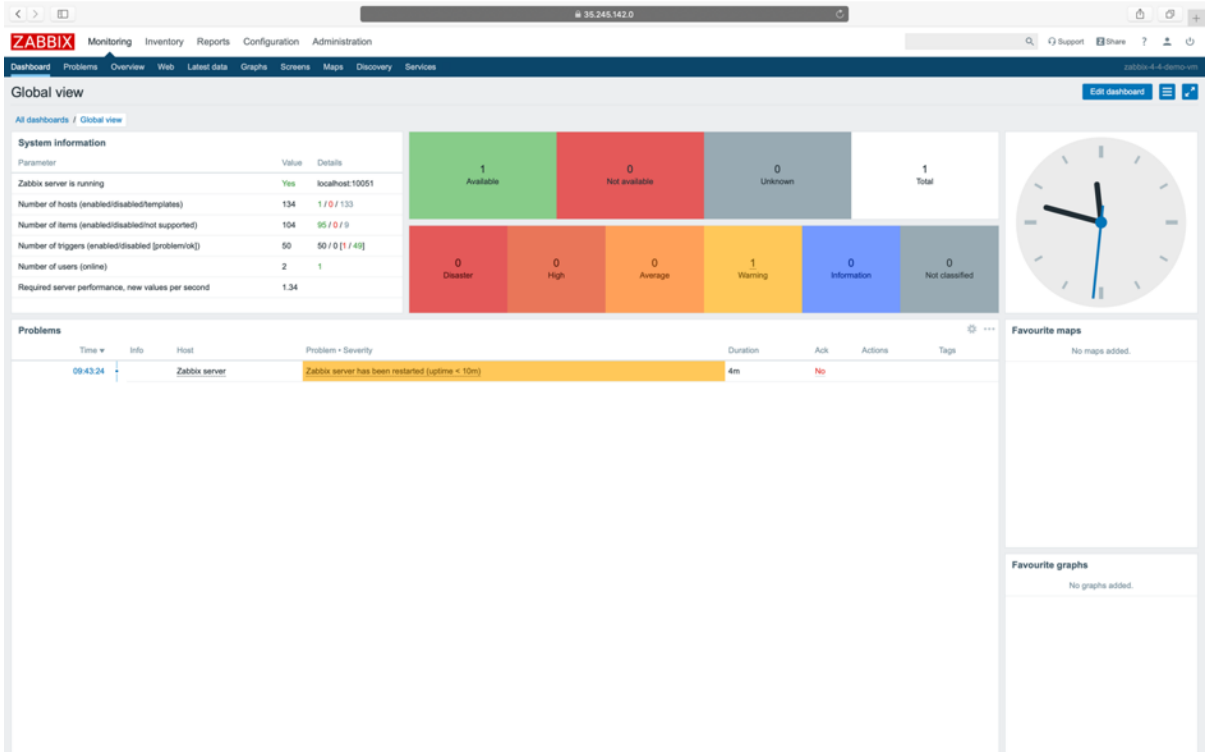


Öffnen Sie einen Webbrowser und rufen Sie die *Site Address* auf, um das Zabbix-Frontend anzuzeigen. Rufen Sie zuvor das Passwort des Zabbix-*Admin*-Benutzers ab, um sich bei Ihrer Zabbix-Appliance anzumelden.

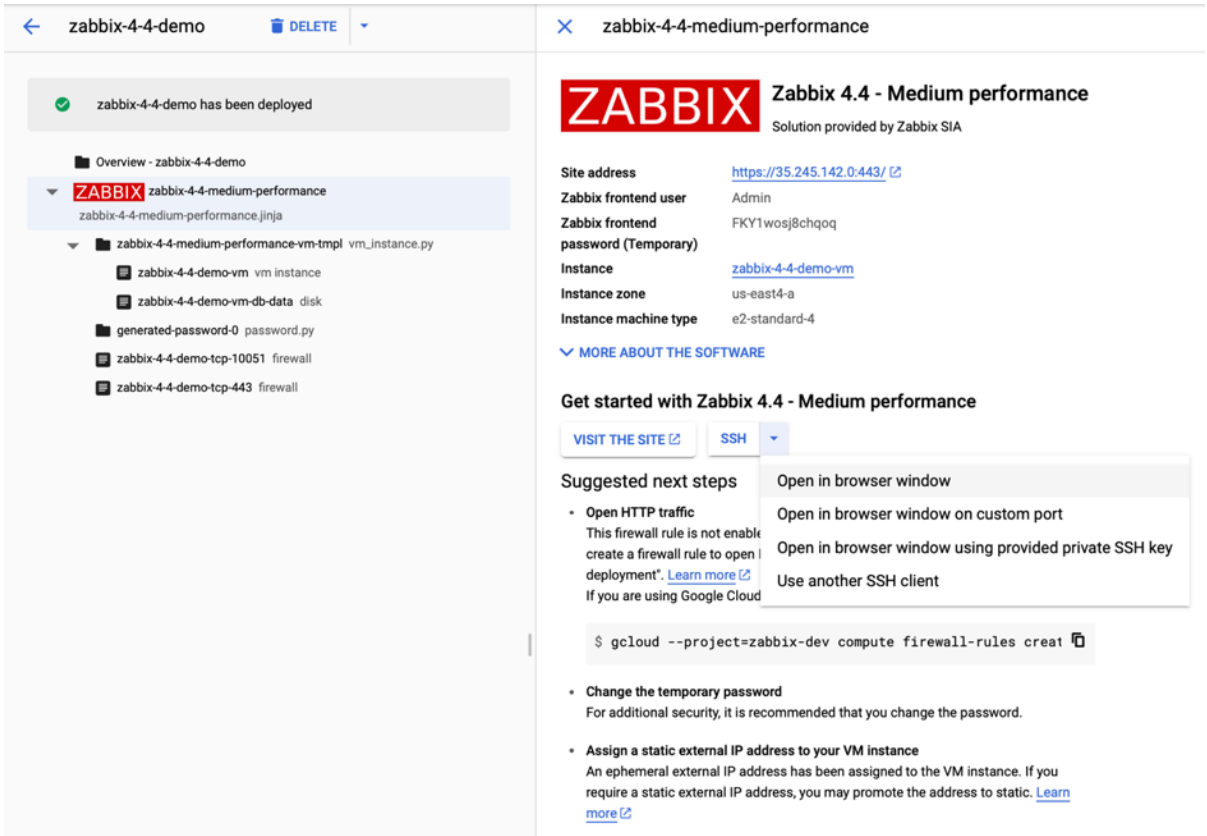


Fertig! Sie verfügen nun über eine Zabbix-Appliance.





Sie können sich auch bei der Instanz anmelden, um mit der Datenbank zu interagieren. Kehren Sie dazu zur Google-Konsole zurück und folgen Sie dem roten Pfeil wie unten gezeigt, um über die Option *Open in browser window* eine SSH-Sitzung zu starten.



MySQL ist ohne ein `root`-Passwort verfügbar. Das Passwort ist in der Konfigurationsdatei `/root/.my.cnf` gespeichert. Standardmäßig verwendet die Zabbix-Appliance die Zeitzone UTC. Wenn Sie diese ändern müssen, bearbeiten Sie die PHP-Konfigurationsdirektive `date.timezone` in `/etc/php.d/99-zabbix.ini` und starten Sie den php-fpm-Prozess neu: `systemctl restart php-fpm`.

Upgrade

Vor dem Upgrade wird dringend empfohlen, alle relevanten Upgrade-Hinweise zu konsultieren.

Es ist außerdem hilfreich zu wissen, welche Komponenten Sie aktualisieren müssen.

Übersicht

Schritt-für-Schritt-Anleitungen für das Upgrade sind verfügbar für:

- Red Hat Enterprise Linux (unter Verwendung von Paketen)
- Debian/Ubuntu (unter Verwendung von Paketen)
- Upgrade mithilfe von Containern
- Upgrade mithilfe von Quellen

Verwandte Anleitungen:

- Für Server in einem Hochverfügbarkeitscluster (HA) siehe Upgrade eines HA-Clusters
- Für die TimescaleDB-Datenbank siehe Upgrade des TimescaleDB-Schemas

Zu aktualisierende Komponenten

Die Aktualisierung von Zabbix Agents wird empfohlen, ist jedoch nicht zwingend erforderlich.

Die Aktualisierung von Zabbix Proxys wird dringend empfohlen.
Der Zabbix Server unterstützt Proxys derselben Hauptversion wie der Server vollständig.
Der Zabbix Server unterstützt außerdem Proxys, die **nicht älter** als die vorherige LTS-Version des Zabbix Servers sind, jedoch mit eingeschränkter Funktionalität (Datensammlung, Ausführung von Remote-Befehlen, sofortige Prüfungen von Datenpunktwerten).
Die Aktualisierung der Konfiguration ist ebenfalls deaktiviert, und veraltete Proxys funktionieren nur mit der alten Konfiguration.

> **Attention:**
> Proxies, die älter als die vorherige LTS-Version des Zabbix-Servers oder neuer als die Hauptversion des Zabbix-Servers sind, werden nicht unterstützt. Der Zabbix-Server ignoriert Daten von nicht unterstützten Proxies, und die gesamte Kommunikation mit dem Zabbix-Server schlägt mit einer Warnung fehl. Weitere Informationen finden Sie unter Versionskompatibilität.

Um Ausfallzeiten und Datenverluste während des Upgrades zu minimieren, wird empfohlen, den Zabbix-Server zu stoppen, zu aktualisieren und zu starten und anschließend die Zabbix-Proxies nacheinander zu stoppen, zu aktualisieren und zu starten. Während der Ausfallzeit des Servers setzen die laufenden Proxies die Datensammlung fort. Sobald der Server wieder läuft, senden veraltete Proxies die Daten an den neueren Server (die Proxy-Konfiguration wird jedoch nicht aktualisiert) und bleiben teilweise funktionsfähig. Alle Benachrichtigungen zu Problemen, die während der Ausfallzeit des Zabbix-Servers aufgetreten sind, werden erst nach dem Start des aktualisierten Servers generiert.

Wenn der Zabbix Proxy zum ersten Mal gestartet wird und die SQLite-Datenbankdatei fehlt, erstellt der Proxy sie automatisch.

Hinweis: Wenn der Zabbix Proxy SQLite3 verwendet und beim Start feststellt, dass die Version der vorhandenen Datenbankdatei älter als erforderlich ist, wird er die **Datenbankdatei automatisch löschen** und eine neue erstellen. Daher gehen die in der SQLite-Datenbankdatei gespeicherten Verlaufsdaten verloren. Wenn die Version des Zabbix Proxy älter als die Version der Datenbankdatei ist, protokolliert Zabbix einen Fehler und wird beendet.

Je nach Größe der Datenbank kann die Aktualisierung der Datenbank auf Version 8.0 lange dauern.

Hinweise zum Upgrade

Ein direktes Upgrade auf Zabbix 8.0.x wird ab den Versionen **2.0**.x unterstützt. Für das Upgrade von früheren Versionen konsultieren Sie die Zabbix-Dokumentation für Version 2.0 und frühere Versionen.

> **Note:**
> Beachten Sie, dass nach dem Upgrade einige Integrationen von Drittanbietersoftware in Zabbix beeinträchtigt sein können, wenn die externe Software nicht mit der aktualisierten Zabbix-Version kompatibel ist.

Die folgenden Upgrade-Hinweise sind verfügbar:

Upgrade von	Vollständige Upgrade-Hinweise lesen	Wichtigste Änderungen zwischen den Versionen
7.4.x	Für: Zabbix 8.0	Zabbix Agents vor Version 1.6 werden nicht mehr unterstützt. Die erforderlichen Mindestversionen der Datenbanken wurden angehoben. Die erforderliche Mindestversion von PHP wurde von 8.0.0 auf 8.2.0 angehoben. Das Zeichen % wurde der Liste `UnsafeUserParameters` für den Zabbix Agent und Zabbix Agent 2 hinzugefügt.
7.2.x	Für: Zabbix 7.4 Zabbix 8.0	Die Unterstützung der PCRE-Bibliothek wurde eingestellt.
7.0.x	Für: Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Die Unterstützung von Oracle-Datenbanken wurde eingestellt.
6.4.x	Für: Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Die erforderliche Mindestversion von PHP wurde von 7.4.0 auf 8.0.0 angehoben. Asynchrone Poller für Agent-, HTTP-Agent-, SNMP-Walk[OID]- und SNMP-Get[OID]-Prüfungen. Separate Datenbanktabelle für Proxys. Der Standardspeicherort der Konfigurationsdatei des Windows Agents wurde geändert. Oracle-Datenbanken sind veraltet. Der alte numerische (Gleitkomma-)Werttyp wurde entfernt.
6.2.x	Für: Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Die erforderliche Mindestversion von MySQL wurde von 8.0.0 auf 8.0.30 angehoben. Die Bibliothek 'libevent_pthreads' ist für den Zabbix Server/Proxy erforderlich. Beim ersten Start nach einem Upgrade löscht ein Zabbix Proxy mit SQLite3 automatisch die alte Datenbankversion (einschließlich aller Verlaufsdaten) und erstellt eine neue.
6.0.x LTS	Für: Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Die erforderliche Mindestversion von PHP wurde von 7.2.5 auf 7.4.0 angehoben. Die Überwachung von Services wurde grundlegend überarbeitet. Deterministische Auslöser müssen während des Upgrades erstellt werden. Wenn die binäre Protokollierung für MySQL/MariaDB aktiviert ist, sind dafür Superuser-Berechtigungen oder das Setzen der Variablen/des Konfigurationsparameters *log_bin_trust_function_creators = 1* erforderlich. Anweisungen zum Setzen der Variablen finden Sie unter Skripte zur Datenbankerstellung.
5.4.x	Für: Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Die erforderlichen Mindestversionen der Datenbanken wurden angehoben. Der Server/Proxy startet nicht, wenn die Datenbank veraltet ist. Aufgrund einer Änderung der Datenbankstruktur gingen Audit-Log-Einträge verloren.
5.2.x	Für: Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Die erforderlichen Mindestversionen der Datenbanken wurden angehoben. Aggregierte Datenpunkte wurden als separater Typ entfernt.

Upgrade von	Vollständige Upgrade-Hinweise lesen	Wichtigste Änderungen zwischen den Versionen
5.0.x LTS	Für: 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	Die erforderliche Mindestversion von PHP wurde von 7.2.0 auf 7.2.5 angehoben. Der Algorithmus zum Hashen von Passwörtern wurde von MD5 auf bcrypt geändert.
4.4.x	Für: 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	Die Unterstützung von IBM DB2 wurde eingestellt. Die erforderliche Mindestversion von PHP wurde von 5.4.0 auf 7.2.0 angehoben. Die erforderlichen Mindestversionen der Datenbanken wurden angehoben. Das Verzeichnis der Zabbix-PHP-Dateien wurde geändert.
4.2.x	Für: 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	Die Medientypen Jabber und Ez Texting wurden entfernt.
4.0.x LTS	Für: 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	Ältere Proxys können keine Daten mehr an einen aktualisierten Server melden. Neuere Agents können nicht mehr mit einem älteren Zabbix Server arbeiten.
3.4.x	Für: 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	Die Bibliotheken ’libpthread’ und ’zlib’ sind jetzt obligatorisch. Die Unterstützung für das Klartextprotokoll wurde eingestellt; der Header ist obligatorisch. Zabbix Agents vor Version 1.4 werden nicht mehr unterstützt. Der Parameter Server ist in der Konfiguration passiver Proxys jetzt obligatorisch.

Upgrade von	Vollständige Upgrade-Hinweise lesen	Wichtigste Änderungen zwischen den Versionen
3.2.x	Für: 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	SQLite-Unterstützung als Backend-Datenbank wurde für den Zabbix Server/das Frontend eingestellt. Perl-kompatible reguläre Ausdrücke (PCRE) werden anstelle von POSIX Extended unterstützt. Die Bibliotheken 'libpcre' und 'libevent' sind für den Zabbix Server obligatorisch. Prüfungen des Exit-Codes wurden für Benutzerparameter, Remote-Befehle und system.run[]-Datenpunkte ohne das Flag 'nowait' sowie für vom Zabbix Server ausgeführte Skripte hinzugefügt. Das Zabbix Java gateway muss aktualisiert werden, um neue Funktionen zu unterstützen.
3.0.x LTS	Für: 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	Das Datenbank-Upgrade kann abhängig von der Größe der Verlaufstabelle langsam sein.
2.4.x	Für: 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	Die erforderliche Mindestversion von PHP wurde von 5.3.0 auf 5.4.0 angehoben. Der Agent-Parameter LogFile muss angegeben werden.

Upgrade von	Vollständige Upgrade-Hinweise lesen	Wichtigste Änderungen zwischen den Versionen
2.2.x LTS	Für: 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	Die knotenbasierte verteilte Überwachung wurde entfernt.
2.0.x	Für: 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	Die erforderliche Mindestversion von PHP wurde von 5.1.6 auf 5.3.0 angehoben. Für einen ordnungsgemäßen Betrieb des Servers ist eine MySQL-Datenbank mit Groß-/Kleinschreibung erforderlich; für den ordnungsgemäßen Betrieb des Zabbix Servers mit einer MySQL-Datenbank sind der Zeichensatz utf8 und die Sortierung utf8_bin erforderlich. Siehe Skripte zur Datenbankerstellung. Die PHP-Erweiterung 'mysqli' ist anstelle von 'mysql' erforderlich.

Upgrade von Paketen

Übersicht

Dieser Abschnitt beschreibt die Schritte, die für ein erfolgreiches Upgrade mit offiziellen RPM- und DEB-Paketen erforderlich sind, die von Zabbix bereitgestellt werden für:

- Red Hat Enterprise Linux
- Debian/Ubuntu

Zabbix-Pakete aus OS-Repositories

Einige OS-Distributionen (insbesondere Debian-basierte Distributionen) stellen ihre eigenen Zabbix-Pakete bereit. Diese Pakete **werden von Zabbix nicht unterstützt** und können veraltet sein oder die neuesten Funktionen und Fehlerbehebungen nicht enthalten. Es wird empfohlen, nur offizielle Pakete aus dem Zabbix Official Repository zu verwenden.

Wenn Sie ein Upgrade von Paketen durchführen, die von OS-Distributionen bereitgestellt wurden (oder diese irgendwann installiert hatten), befolgen Sie dieses Verfahren, um auf offizielle Zabbix-Pakete umzusteigen:

1. Deinstallieren Sie die alten Pakete.
2. Prüfen Sie, ob Dateien aus der alten Installation übrig geblieben sind, und entfernen Sie diese.
3. Installieren Sie die offiziellen Zabbix-Pakete gemäß den von Zabbix bereitgestellten Installationsanweisungen.

Führen Sie kein direktes Upgrade durch, da dies zu einer fehlerhaften Installation führen kann.

Debian/Ubuntu

Übersicht

Dieser Abschnitt enthält Anweisungen für das Upgrade von Zabbix **7.4.x** auf die neueste Version von Zabbix **8.0.x** unter Verwendung offizieller Zabbix-Pakete für Debian/Ubuntu.

Warning:
Bitte lesen Sie vor dem Upgrade die entsprechenden Hinweise zum Upgrade und stellen Sie sicher, dass Ihr System die Anforderungen für Zabbix 8.0 erfüllt.

Note:
Ziehen Sie in Betracht, während des Upgrades zwei parallele SSH-Sitzungen zu verwenden: eine zum Ausführen der Upgrade-Schritte und eine weitere zur Überwachung der Server-/Proxy-Protokolle. Führen Sie beispielsweise in der zweiten Sitzung `tail -f zabbix_server.log` oder `tail -f zabbix_proxy.log` aus, um die neuesten Protokolleinträge und mögliche Fehler in Echtzeit anzuzeigen. Dies kann für Produktivumgebungen entscheidend sein.

Anweisungen für das Upgrade zwischen Nebenversionen von Zabbix 8.0.x (zum Beispiel von 8.0.1 auf 8.0.3) finden Sie unter Upgrade zwischen Nebenversionen.

Upgrade-Verfahren

1 Zabbix-Prozesse stoppen

Stoppen Sie den Zabbix-Server, um sicherzustellen, dass keine neuen Daten in die Datenbank eingefügt werden:

```
systemctl stop zabbix-server
```

Wenn Sie Zabbix Proxy, Agent oder Agent 2 aktualisieren, stoppen Sie auch diese Komponenten:

```
systemctl stop zabbix-proxy
systemctl stop zabbix-agent
systemctl stop zabbix-agent2
```

2 Zabbix-Datenbank sichern

Sichern Sie Ihre vorhandene Zabbix-Datenbank, um sich gegen Upgrade-Fehler abzusichern (zum Beispiel Probleme mit dem Speicherplatz, Stromausfall oder unerwartete Probleme).

3 Zabbix-Konfigurationsdateien, PHP-Dateien und Zabbix-Binärdateien sichern

Sichern Sie vorhandene Zabbix-Konfigurationsdateien, PHP-Dateien und Zabbix-Binärdateien.

Für Konfigurationsdateien führen Sie Folgendes aus:

```
mkdir /opt/zabbix-backup/
cp /etc/zabbix/zabbix_server.conf /opt/zabbix-backup/
##### Für Apache:
cp /etc/apache2/conf-enabled/zabbix.conf /opt/zabbix-backup/
##### Für Nginx:
cp /etc/nginx/conf.d/zabbix.conf /opt/zabbix-backup/
cp /etc/zabbix/nginx.conf /opt/zabbix-backup/
```

Für PHP-Dateien und Zabbix-Binärdateien führen Sie Folgendes aus:

```
cp -R /usr/share/zabbix/ /opt/zabbix-backup/
cp -R /usr/share/zabbix-* /opt/zabbix-backup/
```

4 Repository-Konfigurationspaket aktualisieren

Bevor Sie mit dem Upgrade fortfahren, deinstallieren Sie Ihr aktuelles Zabbix-Repository-Paket:

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

Möglicherweise müssen Sie auch alte Zabbix-Pakete manuell aus Ihrem Arbeitsverzeichnis entfernen (z. B. `rm zabbix-release_latest+de` bevor Sie das neue Paket herunterladen, damit der Paketmanager während des Upgrade-Vorgangs keine veraltete Version erneut verwendet.

Installieren Sie anschließend das neueste Repository-Konfigurationspaket, um die Kompatibilität mit den neuesten Paketen sicherzustellen und aktuelle Sicherheits-Patches oder Fehlerbehebungen einzuschließen.

Unter **Debian 13** führen Sie aus:

```
wget https://repo.zabbix.com/zabbix/8.0/release/debian/pool/main/z/zabbix-release/zabbix-release_latest+de
dpkg -i zabbix-release_latest+debian13_all.deb
```

Unter **Debian 12** führen Sie aus:

```
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:
Für ältere Debian-Versionen ersetzen Sie den obigen Link durch den korrekten Link aus dem Zabbix-Repository. Beachten Sie jedoch, dass Pakete für diese Versionen möglicherweise nicht alle Zabbix-Komponenten enthalten. Um diese Komponenten über Pakete zu aktualisieren, sollten Sie ein Upgrade Ihres Betriebssystems in Betracht ziehen. Eine Liste der enthaltenen Komponenten finden Sie unter Zabbix packages.

Unter **Ubuntu 26.04** führen Sie aus:

```
wget https://repo.zabbix.com/zabbix/8.0/release/ubuntu/pool/main/z/zabbix-release/zabbix-release_latest+ub
dpkg -i zabbix-release_latest+ubuntu26.04_all.deb
```

Unter **Ubuntu 24.04** führen Sie aus:

```
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
```

Unter **Ubuntu 22.04** führen Sie aus:

```
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:
Für ältere Ubuntu-Versionen ersetzen Sie den obigen Link durch den korrekten Link aus dem Zabbix-Repository. Beachten Sie jedoch, dass Pakete für diese Versionen möglicherweise nicht alle Zabbix-Komponenten enthalten. Um diese Komponenten über Pakete zu aktualisieren, sollten Sie ein Upgrade Ihres Betriebssystems in Betracht ziehen. Eine Liste der enthaltenen Komponenten finden Sie unter Zabbix packages.

Möglicherweise wird eine Eingabeaufforderung zur Zabbix-Repository-Konfiguration angezeigt:

```
Configuration file '/etc/apt/sources.list.d/zabbix.list'
==> Deleted (by you or by a script) since installation.
==> Package distributor has shipped an updated version.
What would you like to do about it ?  Your options are:
Y or I  : install the package maintainer's version
N or O  : keep your currently-installed version
D       : show the differences between the versions
Z       : start a shell to examine the situation
The default action is to keep your current version.
*** zabbix.list (Y/I/N/O/D/Z) [default=N] ?
```

Geben Sie `Y` (oder `I`) ein, um die Version des Paketbetreuers der Zabbix-Repository-Konfiguration zu installieren.

Aktualisieren Sie dann die Repository-Informationen:

```
apt update
```

5 Zabbix-Komponenten aktualisieren

Um Zabbix-Komponenten zu aktualisieren, führen Sie folgenden Befehl aus:

```
apt install --only-upgrade zabbix-server-mysql zabbix-frontend-php zabbix-agent
```

- Bei Verwendung von PostgreSQL ersetzen Sie `mysql` im Befehl durch `pgsql`.
- Wenn Sie einen Proxy aktualisieren, ersetzen Sie `server` im Befehl durch `proxy`.
- Wenn Sie Zabbix agent 2 aktualisieren, ersetzen Sie `zabbix-agent` im Befehl durch `zabbix-agent2 zabbix-agent2-plugin-*`.

Attention:
Die Aktualisierung von Zabbix agent 2 mit dem Befehl `apt install zabbix-agent2` kann zu einem Fehler führen. Weitere Informationen finden Sie unter *Bekannte Probleme*.

Möglicherweise wird eine Eingabeaufforderung zur Konfiguration des Zabbix-Servers (oder Proxys) angezeigt:

```
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] ?
```

Geben Sie die Option ein, die am besten zu Ihrer Situation passt. Geben Sie beispielsweise `D` ein, um die aktuelle und die neue Konfiguration zu vergleichen. Entscheiden Sie anschließend, ob Sie die Version des Paketbetreuers installieren möchten (`Y` oder `I`).

Aktualisieren Sie anschließend das Zabbix-Frontend und starten Sie den Webserver neu.

Für Apache führen Sie Folgendes aus:

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

Für Nginx führen Sie Folgendes aus:

```
apt install zabbix-nginx-conf
systemctl restart nginx
```

Für Nginx wird möglicherweise eine Eingabeaufforderung zur Konfigurationsdatei `nginx.conf` angezeigt. Geben Sie die Option ein, die am besten zu Ihrer Situation passt:

```
Configuration file '/etc/zabbix/nginx.conf'
 ==> Modified (by you or by a script) since installation.
 ==> Package distributor has shipped an updated version.
   What would you like to do about it ?  Your options are:
    Y or I  : install the package maintainer's version
    N or O  : keep your currently-installed version
      D     : show the differences between the versions
      Z     : start a shell to examine the situation
 The default action is to keep your current version.
*** nginx.conf (Y/I/N/O/D/Z) [default=N] ?
```

Wenn Sie nach der Aktualisierung ein zusätzliches Zabbix-Paket aus demselben Repository installieren müssen, kann `apt` während der Abhängigkeitsauflösung Aktualisierungen für bereits installierte Zabbix-Pakete auswählen. Um die Aktualisierung eines installierten Zabbix-Servers oder Proxy-Pakets zu vermeiden, halten Sie das Paket vorübergehend zurück und heben Sie die Zurückhaltung anschließend wieder auf:

```
apt-mark hold zabbix-server-mysql
apt install zabbix-sender
apt-mark unhold zabbix-server-mysql
```

6 Überprüfen der Konfigurationsparameter der Komponente

Prüfen Sie die relevanten Upgrade-Hinweise, um festzustellen, ob Änderungen an den Konfigurationsparametern erforderlich sind.

Informationen zu neuen optionalen Parametern finden Sie auf der Seite Was ist neu.

Wenn Nginx als Webserver verwendet wird und die Konfigurationsdatei `/etc/zabbix/nginx.conf` während des Upgrades neu installiert wurde, überprüfen Sie die Datei und passen Sie die Direktiven `listen` und `server_name` an Ihre Zabbix-Frontend-Konfiguration an.

7 Zabbix-Prozesse starten

Starten Sie die aktualisierten Zabbix-Komponenten:

```
systemctl start zabbix-server
systemctl start zabbix-proxy
systemctl start zabbix-agent
systemctl start zabbix-agent2
```

8 Cookies und Cache des Webbrowsers löschen

Nach dem Upgrade müssen Sie möglicherweise Cookies und den Cache des Webbrowsers löschen, damit die Zabbix-Weboberfläche ordnungsgemäß funktioniert.

Upgrade zwischen Minor-Versionen

Es ist möglich, zwischen den Zabbix-8.0.x-Minor-Versionen zu aktualisieren (zum Beispiel von 8.0.1 auf 8.0.3).

Aktualisieren Sie zunächst die Repository-Informationen:

```
apt update
```

Führen Sie dann zum Upgrade aller Zabbix-Komponenten Folgendes aus:

```
apt install --only-upgrade 'zabbix*'
```

- Um nur den Zabbix Server zu aktualisieren, ersetzen Sie `'zabbix*'` im Befehl durch `'zabbix-server*'`.
- Um nur den Zabbix Proxy zu aktualisieren, ersetzen Sie `'zabbix*'` im Befehl durch `'zabbix-proxy*'`.
- Um nur den Zabbix Agent zu aktualisieren, ersetzen Sie `'zabbix*'` im Befehl durch `'zabbix-agent*'`.
- Um nur den Zabbix Agent 2 zu aktualisieren, ersetzen Sie `'zabbix*'` im Befehl durch `'zabbix-agent2*'`.

Red Hat Enterprise Linux

Überblick

Dieser Abschnitt enthält Anweisungen für das Upgrade von Zabbix **7.4.x** auf die neueste Version von Zabbix **8.0.x** unter Verwendung offizieller Zabbix-Pakete für Red Hat Enterprise Linux oder dessen Derivate – AlmaLinux, CentOS Stream, Oracle Linux und Rocky Linux.

Warning:
Bitte lesen Sie vor dem Upgrade die entsprechenden Hinweise zum Upgrade und stellen Sie sicher, dass Ihr System die Anforderungen für Zabbix 8.0 erfüllt.

Note:
Ziehen Sie in Betracht, während des Upgrades zwei parallele SSH-Sitzungen zu verwenden: eine zum Ausführen der Upgrade-Schritte und eine weitere zur Überwachung der Server-/Proxy-Protokolle. Führen Sie beispielsweise in der zweiten Sitzung `tail -f zabbix_server.log` oder `tail -f zabbix_proxy.log` aus, um die neuesten Protokolleinträge und mögliche Fehler in Echtzeit anzuzeigen. Dies kann für Produktivumgebungen entscheidend sein.

Anweisungen zum Upgrade zwischen Zabbix 8.0.x-Nebenversionen (zum Beispiel von 8.0.1 auf 8.0.3) finden Sie unter Upgrade zwischen Nebenversionen.

Upgrade-Verfahren

1 Zabbix-Prozesse stoppen

Stoppen Sie den Zabbix Server, um sicherzustellen, dass keine neuen Daten in die Datenbank eingefügt werden:

```
systemctl stop zabbix-server
```

Wenn Sie Zabbix Proxy, Agent oder Agent 2 aktualisieren, stoppen Sie auch diese Komponenten:

```
systemctl stop zabbix-proxy
systemctl stop zabbix-agent
systemctl stop zabbix-agent2
```

2 Zabbix-Datenbank sichern

Sichern Sie Ihre bestehende Zabbix-Datenbank, um sich gegen Fehler bei der Aktualisierung abzusichern (zum Beispiel Probleme mit dem Festplattenspeicher, Stromausfall oder unerwartete Probleme).

3 Sichern Sie Zabbix-Konfigurationsdateien, PHP-Dateien und Zabbix-Binärdateien

Sichern Sie vorhandene Zabbix-Konfigurationsdateien, PHP-Dateien und Zabbix-Binärdateien.

Führen Sie für Konfigurationsdateien Folgendes aus:

```
mkdir /opt/zabbix-backup/
cp /etc/zabbix/zabbix_server.conf /opt/zabbix-backup/
cp /etc/httpd/conf.d/zabbix.conf  /opt/zabbix-backup/
```

Führen Sie für PHP-Dateien und Zabbix-Binärdateien Folgendes aus:

```
cp -R /usr/share/zabbix/ /opt/zabbix-backup/
cp -R /usr/share/zabbix-* /opt/zabbix-backup/
```

4 Repository-Konfigurationspaket aktualisieren

Aktualisieren Sie vor dem Upgrade Ihr aktuelles Repository-Paket auf die neueste Version, um die Kompatibilität mit den neuesten Paketen sicherzustellen und aktuelle Sicherheits-Patches oder Fehlerbehebungen einzubeziehen.

Führen Sie unter **RHEL 10** Folgendes aus:

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

Wenn Sie den Zabbix Proxy unter RHEL 10 aktualisieren und APM verwenden möchten, installieren Sie außerdem das EPEL-Repository und aktivieren Sie das CRB-Repository. Beide stellen Abhängigkeiten für das APM-Paket des Zabbix Proxy bereit:

```
dnf -y install epel-release || dnf -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-$(
crb enable
```

Führen Sie unter **RHEL 9** Folgendes aus:

```
rpm -Uvh https://repo.zabbix.com/zabbix/8.0/release/rhel/9/noarch/zabbix-release-latest.el9.noarch.rpm
```

> **Note:**
> Ersetzen Sie bei älteren RHEL-Versionen oder deren Derivaten den obigen Link durch den entsprechenden Link aus dem Zabbix-Repository. Beachten Sie jedoch, dass Pakete für diese Versionen möglicherweise nicht alle Zabbix-Komponenten enthalten. Um diese Komponenten aus Paketen zu aktualisieren, sollten Sie ein Upgrade Ihres Betriebssystems in Betracht ziehen. Eine Liste der enthaltenen Komponenten finden Sie unter Zabbix-Pakete.

Bereinigen Sie anschließend den Cache des Paketmanagers `dnf` (einschließlich Headern, Metadaten und Paketdateien, die während vorheriger Installationen oder Aktualisierungen heruntergeladen wurden):

```
dnf clean all
```

Beim nächsten `dnf`-Vorgang lädt `dnf` neue Metadaten aus den Repositorys herunter, da die alten Metadaten gelöscht wurden.

Siehe auch: Bekannte Probleme zum Aktualisieren des Repository-Konfigurationspakets unter RHEL.

5 Zabbix-Komponenten aktualisieren

Um die Zabbix-Komponenten zu aktualisieren, führen Sie folgenden Befehl aus:

```
dnf install zabbix-server-mysql zabbix-web-mysql zabbix-selinux-policy zabbix-agent
```

- Wenn Sie PostgreSQL verwenden, ersetzen Sie `mysql` im Befehl durch `pgsql`.
- Wenn Sie den Proxy aktualisieren, ersetzen Sie `server` im Befehl durch `proxy`.
- Wenn Sie RHEL 10 verwenden und APM nutzen möchten, fügen Sie dem Namen des Proxy-Pakets `-apm` hinzu (z. B. `zabbix-proxy-mysql-apm`).
- Wenn Sie Agent 2 aktualisieren, ersetzen Sie `zabbix-agent` im Befehl durch `zabbix-agent2 zabbix-agent2-plugin-*`.
- Wenn andere Zabbix-Pakete installiert sind, sollten Sie auch diese aktualisieren, indem Sie sie in den Befehl aufnehmen.

> **Attention:**
> Das Aktualisieren von Zabbix Agent 2 mit dem Befehl `dnf install zabbix-agent2` kann zu einem Fehler führen. Weitere Informationen finden Sie unter *Bekannte Probleme*.

Um anschließend das Zabbix Frontend mit Apache zu aktualisieren und Apache neu zu starten, führen Sie folgende Befehle aus:

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

6 Überprüfen der Konfigurationsparameter der Komponente

Prüfen Sie die relevanten Upgrade-Hinweise, um festzustellen, ob Änderungen an den Konfigurationsparametern erforderlich sind.

Informationen zu neuen optionalen Parametern finden Sie auf der Seite Was ist neu.

7 Zabbix-Prozesse starten

Starten Sie die aktualisierten Zabbix-Komponenten:

```
systemctl start zabbix-server
systemctl start zabbix-proxy
systemctl start zabbix-agent
systemctl start zabbix-agent2
```

8 Cookies und Cache des Webbrowsers löschen

Nach dem Upgrade müssen Sie möglicherweise Cookies und den Cache des Webbrowsers löschen, damit die Zabbix-Weboberfläche ordnungsgemäß funktioniert.

Upgrade zwischen Minor-Versionen

Es ist möglich, zwischen Minor-Versionen von Zabbix 8.0.x upzugraden (zum Beispiel von 8.0.1 auf 8.0.3).

Um alle Zabbix-Komponenten upzugraden, führen Sie Folgendes aus:

```
dnf upgrade 'zabbix-*'
```

- Um nur den Zabbix Server upzugraden, ersetzen Sie `'zabbix-*'` im Befehl durch `'zabbix-server-*'`.
- Um nur den Zabbix Proxy upzugraden, ersetzen Sie `'zabbix-*'` im Befehl durch `'zabbix-proxy-*'`.
- Um nur den Zabbix Agent upzugraden, ersetzen Sie `'zabbix-*'` im Befehl durch `'zabbix-agent-*'`.
- Um nur den Zabbix Agent 2 upzugraden, ersetzen Sie `'zabbix-*'` im Befehl durch `'zabbix-agent2-*'`.

Probleme beim Upgrade

Escape-bezogen Zeichenbegrenzung für Funktionsparameter nach dem Upgrade überschritten

In Zabbix 7.0 wurde korrektes Escaping von Backslashes in String-Parametern von Verlaufsfunktionen hinzugefügt.

Da beim Upgrade von Zabbix-Versionen vor 7.0 zusätzliche Backslashes eingefügt werden, führt dies zu längeren Parametern. Dadurch können Auslöser-Funktionen beschädigt werden, wenn die Parameterlänge die maximale Datengröße von 255 Zeichen überschreitet.

Um zu veranschaulichen, welche Probleme während des Datenbank-Upgrades auftreten können, welche Auswirkungen sie haben und wie sie erkannt und behoben werden können, betrachten wir ein Beispiel-Setup mit 3 Datenpunkten und 2 Auslösern.

Einrichtung

1. Zabbix-Version 6.0.
2. Datei `/tmp/ONE`, die 114 Backslashes mit einem a am Ende enthält:

```
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
```

3. Ein Datenpunkt `vfs.file.contents[/tmp/ONE]`, um /tmp/ONE zu lesen.
4. Zwei berechnete Datenpunkte:

Calc_228 - sein Parameter enthält 228 Backslashes, die maskiert sind, sodass sie mit den 114 Backslashes aus der Datei /tmp/ONE **übereinstimmen**:

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

Calc_232 - sein Parameter enthält 232 Backslashes, die maskiert sind, sodass sie **nicht** mit den 114 Backslashes aus der Datei /tmp/ONE übereinstimmen, da sie mindestens 116 echte Backslashes benötigen:

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

5. Zwei Auslöser:

Trig_228 - ähnlich wie bei Calc_228 enthält sein Parameter 228 Backslashes, um mit 114 echten Backslashes aus der Datei /tmp/ONE **übereinzustimmen**:

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

Trig_232 - ähnlich wie bei Calc_232 enthält sein Parameter 232 Backslashes, um mit 116 echten Backslashes übereinzustimmen, daher stimmt er **nicht** mit den 114 Backslashes aus der Datei /tmp/ONE überein:

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

6. Calc_228 gibt 1 zurück und Trig_228 **löst aus**.
7. Calc_232 gibt 0 zurück und Trig_232 **löst nicht** aus.

Upgrade

1. Führen Sie das Upgrade auf 8.0.0 aus und prüfen Sie die Logs auf die Warnung:

```
2485502:20250228:115442.236 DBpatch_6050165(): cannot save in DB function parameter: resulting size 477 is
functionid:33792 function:'find'
used on host: 'Zabbix server'
  in trigger: 'TRIG_228'.
Current parameter value:
'$,1m:now,"regexp","\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\a"'
Resulting escaped value would be:
'$,1m:now,"regexp","\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
DB upgrade and Zabbix server can continue to run, but to make sure this function works correctly
MANUAL INTERVENTION IS REQUIRED !
Need to manually reduce the size of this parameter with macro as a workaround.

2485502:20250228:115442.237 DBpatch_6050165(): cannot save in DB function parameter: resulting size 485 is
functionid:33795 function:'find'
used on host: 'Zabbix server'
  in trigger: 'TRIG_232'.
Current parameter value:
'$,1m:now,"regexp","\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\a"'
Resulting escaped value would be:
'$,1m:now,"regexp","\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
DB upgrade and Zabbix server can continue to run, but to make sure this function works correctly
MANUAL INTERVENTION IS REQUIRED !
Need to manually reduce the size of this parameter with macro as a workaround.
```

2. Das Datenbank-Upgrade wird **abgeschlossen** und der Zabbix Server läuft weiter.
3. Calc_228 gibt weiterhin 1 zurück, während Calc_232 0 zurückgibt.
4. Die Auslöser wurden wie folgt aktualisiert:

Trig_228:

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

Trig_232:

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

Beide Auslöser - Trig_228 und Trig_232 - **lösen jetzt aus**! Dies ist unerwartet, daher ist ein manueller Eingriff erforderlich.

Manuelles Eingreifen

Auslöser-Ausdrücke müssen aktualisiert werden, sodass ihre Backslashes maskiert werden. Dies kann jedoch nicht direkt erfolgen, da die resultierenden Parameter zu lang wären.

Deshalb müssen Sie Makros hinzufügen:

{$228_BACKSLASHES_A}:

```
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
```

{$232_BACKSLASHES_A}:

```
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
```

und die Auslöser-Ausdrücke so **aktualisieren**, dass sie diese verwenden:

Trig_228:

```
find(/Zabbix server/vfs.file.contents[/tmp/ONE],1m:now,"regexp",{$228_BACKSLASHES_A})
```

Trig_232:

```
find(/Zabbix server/vfs.file.contents[/tmp/ONE],1m:now,"regexp",{$232_BACKSLASHES_A})
```

Jetzt funktionieren beide Auslöser auf dieselbe Weise wie zuvor – nur Trig_228 wird **ausgelöst**.

Es gibt keine Probleme mit den berechneten Datenpunkten, sie könnten jedoch zur Konsistenz ebenfalls aktualisiert werden:

Calc_228:

```
find(/Zabbix server/vfs.file.contents[/tmp/ONE],1m:now,"regexp",{$228_BACKSLASHES_A})
```

Calc_232:

```
find(/Zabbix server/vfs.file.contents[/tmp/ONE],1m:now,"regexp",{$232_BACKSLASHES_A})
```

Upgrade aus den Quellen

Übersicht

Dieser Abschnitt beschreibt die erforderlichen Schritte für ein erfolgreiches Upgrade von Zabbix **7.4**.x auf Zabbix **8.0**.x unter Verwendung offizieller Zabbix-Quellen.

> **Warning:**
> Lesen Sie vor dem Upgrade unbedingt die relevanten **Upgrade-Hinweise**!

Möglicherweise möchten Sie auch die Anforderungen für 8.0 prüfen.

> **Note:**
> Es kann hilfreich sein, während des Upgrades zwei parallele SSH-Sitzungen auszuführen: In einer Sitzung führen Sie die Upgrade-Schritte aus, während Sie in der anderen die Protokolle des Servers/Proxys überwachen. Führen Sie beispielsweise in der zweiten SSH-Sitzung `tail -f zabbix_server.log` oder `tail -f zabbix_proxy.log` aus, um die neuesten Protokolleinträge und mögliche Fehler in Echtzeit anzuzeigen. Dies kann für Produktivinstanzen entscheidend sein.

Server-Aktualisierungsprozess

1 Server stoppen

Stoppen Sie den Zabbix Server, um sicherzustellen, dass keine neuen Daten in die Datenbank eingefügt werden.

2 Sichern Sie die vorhandene Zabbix-Datenbank

Dies ist ein sehr wichtiger Schritt.
Stellen Sie sicher, dass Sie über eine Sicherungskopie Ihrer Datenbank verfügen.
Sie hilft Ihnen, wenn das Upgrade-Verfahren fehlschlägt (fehlender Speicherplatz, Stromausfall oder ein anderes unerwartetes Problem).

3 Konfigurationsdateien, PHP-Dateien und Zabbix-Binärdateien sichern

Erstellen Sie eine Sicherungskopie der Zabbix-Binärdateien, Konfigurationsdateien und des PHP-Dateiverzeichnisses.

4 Neue Server-Binärdateien installieren

Verwenden Sie diese Anweisungen, um den Zabbix Server aus den Quellen zu kompilieren.

5 Server-Konfigurationsparameter überprüfen

Stellen Sie sicher, dass Sie die Upgrade-Hinweise prüfen, um festzustellen, ob Änderungen an den Konfigurationsparametern erforderlich sind.

Informationen zu neuen optionalen Parametern finden Sie auf der Seite Was ist neu.

6 Neue Zabbix-Binärdateien starten

Starten Sie die neuen Binärdateien. Überprüfen Sie die Protokolldateien, um festzustellen, ob die Binärdateien erfolgreich gestartet wurden.

Der Zabbix Server aktualisiert die Datenbank automatisch. Beim Start meldet der Zabbix Server die aktuellen (obligatorischen und optionalen) sowie die erforderlichen Datenbankversionen. Wenn die aktuelle obligatorische Version älter als die erforderliche Version ist, führt der Zabbix Server automatisch die erforderlichen Patches für die Datenbankaktualisierung aus. Der Start und der

Fortschritt (in Prozent) der Datenbankaktualisierung werden in die Protokolldatei des Zabbix Servers geschrieben. Nach Abschluss der Aktualisierung wird die Meldung „database upgrade fully completed“ in die Protokolldatei geschrieben. Wenn einer der Aktualisierungspatches fehlschlägt, wird der Zabbix Server nicht gestartet. Der Zabbix Server wird ebenfalls nicht gestartet, wenn die aktuelle obligatorische Datenbankversion neuer als die erforderliche Version ist. Der Zabbix Server wird nur gestartet, wenn die aktuelle obligatorische Datenbankversion der erforderlichen obligatorischen Version entspricht.

```
8673:20161117:104750.259 current database version (mandatory/optional): 03040000/03040000
8673:20161117:104750.259 required mandatory version: 03040000
```

Bevor Sie den Server starten:

- Stellen Sie sicher, dass der Datenbankbenutzer über ausreichende Berechtigungen verfügt (Tabelle erstellen, Tabelle löschen, Index erstellen, Index löschen).
- Stellen Sie sicher, dass ausreichend freier Speicherplatz vorhanden ist.

7 Neue Zabbix-Frontend installieren

Die mindestens erforderliche PHP-Version ist 8.2.0. Aktualisieren Sie PHP bei Bedarf und befolgen Sie die Installationsanweisungen.

8 Cookies und Cache des Webbrowsers löschen

Nach dem Upgrade müssen Sie möglicherweise die Cookies und den Cache des Webbrowsers löschen, damit das Zabbix-Webinterface ordnungsgemäß funktioniert.

Proxy-Aktualisierungsprozess

1 Proxy stoppen

Zabbix Proxy stoppen.

2 Konfigurationsdateien und Zabbix-Proxy-Binärdateien sichern

Erstellen Sie eine Sicherungskopie der Zabbix-Proxy-Binärdatei und der Konfigurationsdatei.

3 Neue Proxy-Binärdateien installieren

Verwenden Sie diese Anweisungen, um den Zabbix Proxy aus den Quellen zu kompilieren.

4 Proxy-Konfigurationsparameter überprüfen

Stellen Sie sicher, dass Sie die Upgrade-Hinweise prüfen, um festzustellen, ob Änderungen an den Konfigurationsparametern erforderlich sind.

Informationen zu neuen optionalen Parametern finden Sie auf der Seite Was ist neu.

5 Neuen Zabbix Proxy starten

Starten Sie den neuen Zabbix Proxy. Prüfen Sie die Protokolldateien, um festzustellen, ob der Proxy erfolgreich gestartet wurde.

Der Zabbix Proxy aktualisiert die Datenbank automatisch. Die Datenbankaktualisierung erfolgt ähnlich wie beim Starten des Zabbix Servers.

Agent-Upgradeprozess

> **Attention:**
> Das Upgrade von Agents ist nicht obligatorisch. Sie müssen Agents nur aktualisieren, wenn dies für den Zugriff auf die neue Funktionalität erforderlich ist.

Das in diesem Abschnitt beschriebene Upgrade-Verfahren kann sowohl für das Upgrade des Zabbix Agent als auch des Zabbix Agent 2 verwendet werden.

1 Agent stoppen

Zabbix Agent stoppen.

2 Sichern Sie Konfigurationsdateien und Zabbix-Agent-Binärdateien

Erstellen Sie eine Sicherungskopie der Zabbix-Agent-Binärdatei und der Konfigurationsdatei.

3 Neue Agent-Binärdateien installieren

Verwenden Sie diese Anweisungen, um den Zabbix Agent aus den Quellen zu kompilieren.

Alternativ können Sie vorkompilierte Zabbix Agents von der Zabbix-Downloadseite herunterladen.

4 Überprüfen der Agent-Konfigurationsparameter

Stellen Sie sicher, dass Sie die Upgrade-Hinweise prüfen, um festzustellen, ob Änderungen an den Konfigurationsparametern erforderlich sind.

Informationen zu neuen optionalen Parametern finden Sie auf der Seite Was ist neu.

5 Neuen Zabbix Agent starten

Starten Sie den neuen Zabbix Agent. Prüfen Sie die Protokolldateien, um festzustellen, ob der Agent erfolgreich gestartet wurde.

Upgrade zwischen Minor-Versionen

Beim Upgrade zwischen Minor-Versionen von 8.0.x (zum Beispiel von 8.0.1 auf 8.0.3) müssen für Server/Proxy/Agent dieselben Aktionen ausgeführt werden wie beim Upgrade zwischen Major-Versionen. Der einzige Unterschied besteht darin, dass beim Upgrade zwischen Minor-Versionen keine Änderungen an der Datenbank vorgenommen werden.

Upgrade von Containern

Übersicht

Dieser Abschnitt beschreibt die Schritte, die für ein erfolgreiches Upgrade auf Zabbix-Container der Version **8.0**.x erforderlich sind.

Separate Anweisungssätze sind verfügbar für:

- Upgrade des Zabbix-Komponenten-Images
- Upgrade der Docker-Compose-Datei

Warning:
Lesen Sie vor dem Upgrade unbedingt die entsprechenden **Upgrade-Hinweise**!

Attention:
Bevor Sie mit dem Upgrade beginnen, vergewissern Sie sich, dass Benutzer über die erforderlichen Berechtigungen für die Datenbank zu Upgrade-Zwecken verfügen.

 Bei Upgrades von Zabbix 6.0 oder älter müssen während des Upgrades deterministische Auslöser erstellt werden. Wenn die binäre Protokollierung für MySQL/MariaDB aktiviert ist, sind dafür Superuser-Rechte oder das Setzen der Variablen/des Konfigurationsparameters `log_bin_trust_function_creators = 1` erforderlich. Anweisungen zum Setzen der Variablen finden Sie unter Skripte zur Datenbankerstellung.

 Beachten Sie, dass die Variable bei Ausführung über eine Konsole nur temporär gesetzt wird und beim Neustart eines Docker verworfen wird. Lassen Sie in diesem Fall Ihren SQL-Dienst weiterlaufen und stoppen Sie nur den Dienst `zabbix-server`, indem Sie `docker compose down zabbix-server` und anschließend `docker compose up -d zabbix-server` ausführen.

 Alternativ können Sie diese Variable in der Konfigurationsdatei setzen.

Abhängig von der Größe der Datenbank kann das Upgrade der Datenbank auf Version 8.0 lange dauern.

Zabbix-Komponenten-Image aktualisieren

Diese Anweisungen aktualisieren Ihre Docker-Images, die als Basis für Ihre Zabbix-Komponenten-Container verwendet werden.

Die folgenden Schritte verwenden einen Zabbix-Proxy als Beispiel. Ersetzen Sie `zabbix-proxy-sqlite3` durch den Containernamen Ihrer Komponente.

1. Prüfen Sie alle Container, um ihre Namen und aktuellen Image-Versionen zu ermitteln:

```
docker ps -a

#### CONTAINER ID   IMAGE                                          (...) NAMES
#### 5a0c2e3d02b4   zabbix/zabbix-proxy-sqlite3:alpine-7.4-latest   (...) zabbix-proxy-sqlite3
```

2. Laden Sie die neue Image-Version herunter:

```
docker pull zabbix/zabbix-proxy-sqlite3:alpine-8.0-latest
```

3. Stoppen Sie den Zabbix-Komponenten-Container:

```
docker stop zabbix-proxy-sqlite3
```

4. Entfernen Sie den Zabbix-Komponenten-Container:

Attention:
Stellen Sie sicher, dass Sie alle Konfigurationsflags und Volume-Mounts notiert haben, die im ursprünglichen `docker run`-Befehl verwendet wurden, bevor Sie fortfahren, da sie zum erneuten Erstellen des Containers benötigt werden.

```
docker rm zabbix-proxy-sqlite3
```

5. Starten Sie einen neuen Container mit dem aktualisierten Image:

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

Note:
Weitere Informationen zum Ausführen von Zabbix-Containern finden Sie unter Docker (manuelle Bereitstellung).

6. Überprüfen Sie den Erfolg des Upgrades, indem Sie die Container-Logs prüfen:

```
docker logs -f zabbix-proxy-sqlite3
```

Docker Compose-Datei aktualisieren

Mit diesen Anweisungen werden die Docker Compose-Dateien aktualisiert, die für die Bereitstellung von Zabbix-Komponenten verwendet werden. Befolgen Sie diese Anweisungen, wenn Sie Zabbix mit Docker Compose installiert haben.

1. Prüfen Sie alle Container, um ihre Namen und aktuellen Image-Versionen zu ermitteln:

```
docker ps -a

#### CONTAINER ID   IMAGE                                          (...) NAMES
#### d095f486deef   zabbix/zabbix-server-mysql:alpine-7.4-latest   (...) zabbix-docker-zabbix-server-1
#### 70627d833da0   zabbix/zabbix-web-nginx-mysql:alpine-7.4-latest (...) zabbix-docker-zabbix-web-ngina
#### aaaa664cc185   zabbix/zabbix-server-mysql:alpine-7.4-latest   (...) zabbix-docker-server-db-init-1
#### 9ab3e77492ff   mysql:8.0-oracle                               (...) zabbix-docker-mysql-server-1
```

2. Wechseln Sie in das zuvor geklonte Zabbix-Docker-Repository, laden Sie die neuesten Änderungen herunter und wechseln Sie zum 8.0-Branch:

Attention:
Wenn Sie `.env`, `compose.yaml` oder andere Dateien in Ihrem lokalen Repository geändert haben, kann ein `git pull` diese überschreiben. Sichern Sie alle lokalen Änderungen, bevor Sie diesen Schritt ausführen.

```
cd zabbix-docker
git pull
git checkout 8.0
```

3. Stellen Sie die Zabbix-Komponenten mit der aktualisierten Docker Compose-Datei bereit:

```
#### Mit MySQL als Datenbank:
docker compose -f ./compose.yaml up -d

#### Mit PostgreSQL als Datenbank:
docker compose -f ./compose_pgsql.yaml up -d
```

Note:
Nachdem alle Container neu erstellt und gestartet wurden, kann es 1 bis 3 Minuten dauern, bis die Datenerfassung wieder aufgenommen wird. Weitere Informationen zum Ausführen von Zabbix-Containern finden Sie unter Docker Compose.

4. Überprüfen Sie anhand der Container-Protokolle, ob das Upgrade erfolgreich war (ersetzen Sie `zabbix-docker-zabbix-server-1` durch den Namen des Containers Ihrer Komponente):

```
docker logs -f zabbix-docker-zabbix-server-1
```

Bekannte Probleme

Siehe auch: Kompilierungsprobleme.

Upgrade

SQL-Modus-Einstellung für ein erfolgreiches Upgrade

Die Einstellung `sql_mode` in MySQL/MariaDB muss den Modus "STRICT_TRANS_TABLES" gesetzt haben. Falls dieser fehlt, schlägt das Upgrade der Zabbix-Datenbank fehl (siehe auch ZBX-19435).

Upgrade mit MariaDB 10.2.1 und früher

Das Upgrade von Zabbix kann fehlschlagen, wenn Datenbanktabellen mit MariaDB 10.2.1 oder früher erstellt wurden, da in diesen Versionen das Standard-Zeilenformat compact ist. Dies kann behoben werden, indem das Zeilenformat auf dynamic geändert wird (siehe auch ZBX-17690).

MySQL-Datenbankzeichensatz und -Kollation

Zabbix-Installationen, die eine MySQL-Datenbank verwenden, die mit dem Zeichensatz utf8mb3 und der Kollation utf8mb3_bin erstellt wurde, können auf Probleme stoßen. Beispielsweise kann ein bekanntes MySQL-Problem während der Aktualisierung des Datenbankschemas oder bei anderen DDL-Operationen zu Fehlern vom Typ `Row size too large (> 8126)` führen.
 Es wird dringend empfohlen, Datenbanken mit dem Zeichensatz utf8mb3 und der Kollation utf8mb3_bin in den Zeichensatz utf8mb4 und die Kollation utf8mb4_bin zu konvertieren, wie unter Reparieren des Zeichensatzes und der Kollation der Zabbix-Datenbank beschrieben.

Speicher- und Dateideskriptor-Leck mit libcurl 8.21.0

Ein bekanntes Problem in libcurl 8.21.0 kann bei Zabbix-Funktionen, die libcurl verwenden, zu Speicher- und Dateideskriptor-Lecks führen, wenn TLS-gesicherte Verbindungen verarbeitet werden.
 Eine Umgehungslösung für dieses Problem besteht darin, libcurl vorübergehend auf Version 8.20.0 herunterzustufen, bis eine Version mit der Fehlerbehebung verfügbar ist.

Vorlagen

Vorlagenkompatibilität in Dual-Stack-Umgebungen (IPv4/IPv6)

In Dual-Stack-Umgebungen (Systeme, die sowohl IPv4 als auch IPv6 unterstützen) wird der Hostname `localhost` in der Regel sowohl in IPv4- als auch in IPv6-Adressen aufgelöst. Da viele Betriebssysteme und DNS-Resolver IPv6 häufig gegenüber IPv4 priorisieren, funktionieren Zabbix-Vorlagen möglicherweise nicht korrekt, wenn der zu überwachende Dienst so konfiguriert ist, dass er nur auf IPv4 lauscht.

Dienste, die nicht für das Lauschen auf IPv6-Adressen konfiguriert sind, sind möglicherweise nicht erreichbar, was zu Überwachungsfehlern führt. Benutzer können den Zugriff für IPv4 korrekt konfigurieren und dennoch aufgrund des Standardverhaltens, bei dem IPv6 priorisiert wird, auf Verbindungsprobleme stoßen.

Eine mögliche Abhilfe besteht darin, sicherzustellen, dass die Dienste (Nginx, Apache, PostgreSQL usw.) so konfiguriert sind, dass sie sowohl auf IPv4- als auch auf IPv6-Adressen lauschen, und dass dem Zabbix Server/Agent der Zugriff über IPv6 erlaubt ist. Verwenden Sie außerdem in Zabbix-Vorlagen und -Konfigurationen ausdrücklich `localhost` anstelle von `127.0.0.1`, um die Kompatibilität mit IPv4 und IPv6 sicherzustellen.

Zum Beispiel müssen Sie bei der Überwachung von PostgreSQL mit der Vorlage PostgreSQL by Zabbix agent 2 möglicherweise die Datei `pg_hba.conf` bearbeiten, um Verbindungen für den Benutzer `zbx_monitor` zuzulassen. Wenn in der Dual-Stack-Umgebung IPv6 priorisiert wird (das System löst localhost zu `::1` auf) und Sie `localhost` konfigurieren, aber nur einen IPv4-Eintrag (`127.0.0.1/32`) hinzufügen, schlägt die Verbindung fehl, da kein passender IPv6-Eintrag vorhanden ist.

Das folgende Beispiel einer `pg_hba.conf`-Datei stellt sicher, dass der Benutzer `zbx_monitor` von der lokalen Maschine aus mit sowohl IPv4- als auch IPv6-Adressen und unterschiedlichen Authentifizierungsmethoden eine Verbindung zu jeder Datenbank herstellen kann:

```
### 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
```

Falls erforderlich, können Sie bei der Konfiguration des Makros der Vorlage PostgreSQL by Zabbix agent 2 für die Verbindungszeichenfolge auch direkt die IPv4-Adresse (`127.0.0.1`) verwenden.

Versehentliche Installation von EPEL-Zabbix-Paketen

Wenn das EPEL-Repository installiert und aktiviert ist, kann die Installation von Zabbix-Paketen dazu führen, dass EPEL-Versionen anstelle der offiziellen Zabbix-Pakete installiert werden. So beheben Sie das Problem:

1. Entfernen Sie alle Zabbix-Pakete, die aus EPEL installiert wurden:

```
dnf remove zabbix-server-mysql
```

2. Schließen Sie Zabbix-Pakete aus EPEL aus, indem Sie die folgende Zeile zur Datei `/etc/yum.repos.d/epel.repo` hinzufügen:

```
[epel]
...
excludepkgs=zabbix*
```

3. Installieren Sie das offizielle Zabbix-Server-Paket erneut:

```
dnf install zabbix-server-mysql
```

Während der Installation enthalten offizielle Zabbix-Pakete das Wort `release` in ihrer Versionszeichenfolge (z. B. `7.0.0-release1.el8`), wodurch sie sich von EPEL-Paketen unterscheiden.

Zabbix-Pakete für RHEL in Red Hat UBI-Umgebungen

Wenn Sie Zabbix aus Red Hat Enterprise Linux-Paketen in Umgebungen mit Red Hat Universal Base Image installieren, stellen Sie sicher, dass der Zugriff auf die erforderlichen Repositories und Abhängigkeiten vorhanden ist. Zabbix-Pakete hängen von den Bibliotheken `libOpenIPMI.so` und `libOpenIPMIposix.so` ab, die von keinem Paket in den standardmäßig aktivierten Paketmanager-Repositories auf UBI-Systemen bereitgestellt werden und zu Installationsfehlern führen.

Die Bibliotheken `libOpenIPMI.so` und `libOpenIPMIposix.so` sind im Paket `OpenIPMI-libs` verfügbar, das vom Repository `redhat-#-for-<arch>-appstream-rpms` bereitgestellt wird. Der Zugriff auf dieses Repository wird über Subskriptionen gesteuert, die im Fall von UBI-Umgebungen durch das Einhängen der Repository-Konfiguration und der Verzeichnisse mit Geheimnissen des RHEL-Hosts in den Dateisystem-Namespace des Containers weitergegeben werden.

Weitere Informationen finden Sie unter ZBX-24291.

Abgelaufener Signaturschlüssel für RHEL-Pakete

Beim Upgrade von Zabbix auf Red Hat Enterprise Linux oder dessen Derivaten kann ein Problem mit einem abgelaufenen Signaturschlüssel für Pakete im Zabbix repository auftreten. Wenn ein Signaturschlüssel abläuft, führen Versuche, Paketsignaturen zu verifizieren, zu einem Fehler, der darauf hinweist, dass das Zertifikat oder der Schlüssel nicht mehr gültig ist. Zum Beispiel:

```
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
```

Um solche Probleme zu beheben, installieren Sie das neueste Paket `zabbix-release` für Ihre spezifische RHEL-Variante manuell neu (ersetzen Sie den untenstehenden Link durch den korrekten Link aus dem Zabbix repository).

Zum Beispiel führen Sie unter **RHEL 10** Folgendes aus:

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

Aktualisieren Sie anschließend die Repository-Informationen:

```
dnf update
```

Weitere Informationen finden Sie unter ZBX-24761.

Datenbank-TLS-Verbindung mit MariaDB

Die Datenbank-TLS-Verbindung wird mit der Option `verify_ca` für den Parameter DBTLSConnect parameter nicht unterstützt, wenn MariaDB verwendet wird.

Mögliche Deadlocks mit MySQL/MariaDB

Bei Betrieb unter hoher Last und wenn mehr als ein LLD-Worker beteiligt ist, kann es zu einem Deadlock kommen, der durch einen InnoDB-Fehler im Zusammenhang mit der Strategie zur Sperrung von Zeilen verursacht wird (siehe upstream bug). Der Fehler wurde in MySQL ab Version 8.0.29 behoben, jedoch nicht in MariaDB. Weitere Details finden Sie unter ZBX-21506.

Globale Ereigniskorrelation

Ereignisse werden möglicherweise nicht korrekt korreliert, wenn das Zeitintervall zwischen dem ersten und dem zweiten Ereignis sehr klein ist, d. h. eine halbe Sekunde oder weniger.

NetBSD 8.0 und neuer

Verschiedene Zabbix-Prozesse können auf NetBSD-Versionen 8.X und 9.X beim Start zufällig abstürzen. Dies ist auf die zu kleine Standard-Stackgröße (4 MB) zurückzuführen, die durch Ausführen von Folgendem erhöht werden muss:

```
ulimit -s 10240
```

Weitere Informationen finden Sie im zugehörigen Problembericht: ZBX-18275.

Einschränkungen regulärer Ausdrücke in Zabbix Agent 2

Zabbix Agent 2 unterstützt aufgrund der Einschränkungen der standardmäßigen Go-regexp-Bibliothek keine Lookaheads und Lookbehinds in regulären Ausdrücken.

IPMI-Prüfungen

IPMI-Prüfungen funktionieren nicht mit dem standardmäßigen OpenIPMI-Bibliothekspaket unter Debian vor 9 (stretch) und Ubuntu vor 16.04 (xenial). Um dies zu beheben, kompilieren Sie die OpenIPMI-Bibliothek mit aktivierter OpenSSL-Unterstützung neu, wie in ZBX-6139 beschrieben.

IPMI – Nicht vertrauenswürdige Hosts können OpenIPMI zum Absturz bringen

In der von Zabbix zum Abrufen von IPMI-Daten verwendeten OpenIPMI-Bibliothek gibt es einen Fehler, der durch speziell präparierte Antworten eines nicht vertrauenswürdigen Geräts ausgelöst werden kann. Ein nicht vertrauenswürdiges IPMI-Gerät kann präparierte Daten senden, die einen Absturz der OpenIPMI-Bibliothek verursachen. Dies kann wiederum dazu führen, dass der Zabbix-Serverprozess, der das IPMI-Polling durchführt, beendet wird.

SSH-Prüfungen

- Einige Linux-Distributionen wie Debian und Ubuntu unterstützen keine verschlüsselten privaten Schlüssel (mit Passphrase), wenn die Bibliothek libssh2 aus Paketen installiert wurde. Weitere Details finden Sie unter ZBX-4850.
- Bei Verwendung von libssh 0.9.x auf einigen Linux-Distributionen mit OpenSSH 8 können SSH-Prüfungen gelegentlich „Cannot read data from SSH server“ melden. Dies wird durch ein libssh-Problem verursacht (ausführlicherer Bericht). Es wird erwartet, dass der Fehler mit der stabilen Veröffentlichung von libssh 0.9.5 behoben wurde. Siehe auch ZBX-17756 für Details.
- Die Verwendung der Pipe „|“ im SSH-Skript kann zu einem Fehler „Cannot read data from SSH server“ führen. In diesem Fall wird empfohlen, die Version der libssh-Bibliothek zu aktualisieren. Siehe auch ZBX-21337 für Details.

ODBC-Prüfungen

- Der MySQL-unixODBC-Treiber sollte nicht mit einem Zabbix Server oder Zabbix Proxy verwendet werden, der gegen die MariaDB-Connector-Bibliothek kompiliert wurde, und umgekehrt. Wenn möglich, ist es aufgrund eines upstream bug außerdem besser, nicht denselben Connector wie den Treiber zu verwenden. Empfohlene Konfiguration:

  PostgreSQL-, SQLite- oder Oracle-Connector → MariaDB- oder MySQL-unixODBC-Treiber
MariaDB-Connector → MariaDB-unixODBC-Treiber
MySQL-Connector → MySQL-unixODBC-Treiber

  Weitere Informationen und verfügbare Workarounds finden Sie unter ZBX-7665.

- Von Microsoft SQL Server abgefragte XML-Daten können auf Linux- und UNIX-Systemen auf verschiedene Weise abgeschnitten werden.

- Es wurde beobachtet, dass die Verwendung von ODBC-Prüfungen zur Überwachung von Oracle-Datenbanken mit verschiedenen Versionen von Oracle Instant Client für Linux zum Absturz des Zabbix Servers führt.
 Siehe auch: ZBX-18402, ZBX-20803.

- Wenn Sie den FreeTDS-UnixODBC-Treiber verwenden, müssen Sie einer SQL-Abfrage eine Anweisung 'SET NOCOUNT ON' voranstellen (zum Beispiel `SET NOCOUNT ON DECLARE @strsql NVARCHAR(max) SET @strsql = ....`). Andernfalls kann der Datenpunkt zur Datenbanküberwachung in Zabbix die Informationen nicht abrufen und meldet den Fehler "SQL query returned empty result".
 Weitere Informationen finden Sie unter ZBX-19917.

Falscher Parameter für die Anfragemethode in Datenpunkten

Der Parameter für die Anfragemethode, der nur in HTTP-Prüfungen verwendet wird, kann infolge eines Upgrades von einer Zabbix-Version vor 4.0 fälschlicherweise auf „1“ gesetzt sein, einen Nicht-Standardwert für alle Datenpunkte. Einzelheiten zur Behebung dieser Situation finden Sie unter ZBX-19308.

Web-Überwachung und HTTP-Agent

Der Zabbix Server verliert auf einigen Linux-Distributionen aufgrund eines Upstream-Fehlers Speicher, wenn „SSL verify peer“ in Webszenarien oder im HTTP-Agent aktiviert ist. Weitere Informationen und verfügbare Behelfslösungen finden Sie unter ZBX-10486.

Einfache Prüfungen

In **fping**-Versionen vor v3.10 gibt es einen Fehler, durch den doppelte Echo-Reply-Pakete falsch verarbeitet werden. Dies kann zu unerwarteten Ergebnissen bei den Datenpunkten `icmpping`, `icmppingloss`, `icmppingsec` führen. Es wird empfohlen, die neueste Version von **fping** zu verwenden. Weitere Informationen finden Sie unter ZBX-11726.

Fehler bei der Ausführung von fping in rootless-Containern

Wenn Container im rootless-Modus oder in einer Umgebung mit spezifischen Einschränkungen ausgeführt werden, können bei der Durchführung von ICMP-Prüfungen Fehler im Zusammenhang mit der Ausführung von fping auftreten, z. B. `fping: Operation not permitted` oder der Verlust aller Pakete an alle Ressourcen.

Um dieses Problem zu beheben, fügen Sie `--cap-add=net_raw` zu den Befehlen „docker run“ oder „podman run“ hinzu.

Zusätzlich kann die Ausführung von fping in Nicht-Root-Umgebungen eine sysctl-Änderung erfordern, z. B.:

```
sudo sysctl -w "net.ipv4.ping_group_range=0 1995"
```

wobei „1995“ die zabbix-GID ist. Weitere Details finden Sie unter ZBX-22833.

SNMP-Prüfungen

Wenn das Betriebssystem OpenBSD verwendet wird, kann ein Use-after-free-Fehler in der Net-SNMP-Bibliothek bis einschließlich Version 5.7.3 einen Absturz des Zabbix Server verursachen, wenn der Parameter SourceIP in der Konfigurationsdatei des Zabbix Server gesetzt ist. Als Behelfslösung setzen Sie bitte den Parameter SourceIP nicht. Dasselbe Problem betrifft auch Linux, führt dort jedoch nicht dazu, dass der Zabbix Server nicht mehr funktioniert. Ein lokaler Patch für das net-snmp-Paket unter OpenBSD wurde angewendet und wird mit OpenBSD 6.3 veröffentlicht.

SNMP-Datenspitzen

Es wurden Spitzen in SNMP-Daten beobachtet, die mit bestimmten physischen Faktoren wie Spannungsspitzen im Stromnetz zusammenhängen können. Siehe ZBX-14318 für weitere Details.

SNMP-Traps

Das Paket ”net-snmp-perl”, das für SNMP-Traps benötigt wird, wurde in RHEL 8.0-8.2 entfernt und in RHEL 8.3 wieder hinzugefügt.

Wenn Sie also RHEL 8.0-8.2 verwenden, ist die beste Lösung ein Upgrade auf RHEL 8.3.

Weitere Informationen finden Sie auch unter ZBX-17192.

Absturz des Alerter-Prozesses in RHEL 7

In RHEL 7 wurden Fälle eines Absturzes des Alerter-Prozesses eines Zabbix-Servers festgestellt. Weitere Informationen finden Sie unter ZBX-10461.

Upgrade von Zabbix Agent 2 (6.0.5 oder älter)

Beim Upgrade von Zabbix Agent 2 (Version 6.0.5 oder älter) aus Paketen kann ein pluginbezogener Dateikonfliktfehler auftreten. Um den Fehler zu beheben, sichern Sie Ihre Agent-2-Konfiguration (falls erforderlich), deinstallieren Sie Agent 2 und installieren Sie ihn erneut.

Führen Sie auf RHEL-basierten Systemen Folgendes aus:

```
dnf remove zabbix-agent2
dnf install zabbix-agent2
```

Führen Sie auf Debian-basierten Systemen Folgendes aus:

```
apt remove zabbix-agent2
apt install zabbix-agent2
```

Weitere Informationen finden Sie unter ZBX-23250.

Wechselnde Frontend-Gebietsschemata

Es wurde beobachtet, dass sich die Gebietsschemata des Frontends scheinbar ohne ersichtlichen Grund ändern können, d. h., einige Seiten (oder Teile von Seiten) werden in einer Sprache angezeigt, während andere Seiten (oder Teile von Seiten) in einer anderen Sprache dargestellt werden. Typischerweise kann das Problem auftreten, wenn mehrere Benutzer vorhanden sind, von denen einige ein Gebietsschema verwenden, während andere ein anderes verwenden.

Eine bekannte Abhilfe besteht darin, Multithreading in PHP und Apache zu deaktivieren.

Das Problem hängt damit zusammen, wie das Festlegen des Gebietsschemas in PHP funktioniert: Die Informationen zum Gebietsschema werden pro Prozess und nicht pro Thread verwaltet. Wenn mehrere Projekte in einer Multithread-Umgebung vom selben Apache-Prozess ausgeführt werden, ist es daher möglich, dass das Gebietsschema in einem anderen Thread geändert wird und sich dadurch die Verarbeitung der Daten im Zabbix-Thread ändert.

Weitere Informationen finden Sie in den zugehörigen Problemberichten:

- ZBX-10911 (Problem mit wechselnden Frontend-Gebietsschemata)
- ZBX-16297 (Problem bei der Zahlenverarbeitung in Diagrammen unter Verwendung der Funktion `bcdiv` aus den BC-Math-Funktionen)

Diagramme

Probleme mit Diagrammen (klassisch)

Wenn Probleme mit klassischen Diagrammen auftreten, wird empfohlen, die GD-Bibliothek (libgd) auf Version 2.3.3-13 oder höher und PHP auf Version 8.0.19, 8.1.33, 8.2.29, 8.3.25, 8.4.12 oder höher zu aktualisieren.

Sommerzeit

Änderungen durch die Sommerzeit (DST) führen zu Unregelmäßigkeiten bei der Anzeige der Beschriftungen der X-Achse (doppelte Datumsangaben, fehlende Datumsangaben usw.).

Summenaggregation

Bei Verwendung der Summenaggregation in einem Diagramm für einen Zeitraum von weniger als einer Stunde zeigen Diagramme falsche (multiplizierte) Werte an, wenn die Daten aus Trends stammen.

Textüberlappung

Bei einigen Frontend-Sprachen (z. B. Japanisch) können lokale Schriftarten zu Textüberlappungen in der Diagrammlegende führen. Um dies zu vermeiden, verwenden Sie Version 2.3.0 (oder höher) der PHP-GD-Erweiterung.

Überwachung von Protokolldateien

`log[]`- und `logrt[]`-Datenpunkte lesen die Protokolldatei wiederholt vom Anfang an erneut, wenn das Dateisystem zu 100 % voll ist und die Protokolldatei erweitert wird (weitere Informationen finden Sie unter ZBX-10884).

Langsame MySQL-Abfragen

Der Zabbix Server erzeugt langsame `SELECT`-Abfragen, wenn für Datenpunkte keine Werte vorhanden sind. Dieses Problem tritt bekanntermaßen in den MySQL-Versionen 5.6/5.7 auf (eine ausführlichere Diskussion finden Sie unter ZBX-10652) und kann in bestimmten Fällen auch in späteren MySQL-Versionen auftreten. Eine mögliche Umgehung besteht darin, den Optimizer `index_condition_pushdown` oder `prefer_ordering_index` in MySQL zu deaktivieren. Beachten Sie jedoch, dass diese Umgehung nicht alle Probleme im Zusammenhang mit langsamen Abfragen beheben kann.

Persistente Filtereinstellungen aus Links

Beim Öffnen eines Links zu einer Zabbix-Frontend-Seite, die Filtereinstellungen einschließlich der Zeitauswahl enthält, wird der Filter automatisch für den Benutzer in der Datenbank gespeichert und ersetzt die zuvor gespeicherten Filter- und/oder Zeitauswahleinstellungen für diese Seite. Diese Einstellungen bleiben aktiv, bis der Benutzer sie manuell aktualisiert oder zurücksetzt.

Problem mit IPv6-Adressen in SNMPv3-Traps

Aufgrund eines Fehlers in net-snmp wird die IPv6-Adresse bei der Verwendung von SNMPv3 in SNMP-Traps möglicherweise nicht korrekt angezeigt. Weitere Informationen sowie eine mögliche Behelfslösung finden Sie unter ZBX-14541.

Gekürzte lange IPv6-IP-Adresse in Informationen zu fehlgeschlagenen Anmeldungen

Eine Meldung über einen fehlgeschlagenen Anmeldeversuch zeigt nur die ersten 39 Zeichen einer gespeicherten IP-Adresse an, da dies die Zeichenbegrenzung des Datenbankfelds ist. Das bedeutet, dass IPv6-IP-Adressen mit mehr als 39 Zeichen unvollständig angezeigt werden.

Zabbix Agent-Prüfungen unter Windows

Nicht vorhandene DNS-Einträge in einem `Server`-Parameter der Zabbix-Agent-Konfigurationsdatei (`zabbix_agentd.conf`) können die Antwortzeit des Zabbix Agent unter Windows erhöhen. Dies liegt daran, dass der Windows-DNS-Caching-Dienst negative Antworten für IPv4-Adressen nicht zwischenspeichert. Für IPv6-Adressen werden negative Antworten jedoch zwischengespeichert, daher besteht eine mögliche Umgehung darin, IPv4 auf dem Host zu deaktivieren.

YAML-Export/Import

Es gibt einige bekannte Probleme mit YAML Export/Import:

- Fehlermeldungen sind nicht übersetzbar;
- Gültiges JSON mit einer .yaml-Dateierweiterung kann manchmal nicht importiert werden;
- Nicht in Anführungszeichen gesetzte, menschenlesbare Datumsangaben werden automatisch in Unix-Zeitstempel konvertiert.

Einrichtungsassistent auf SUSE mit NGINX und php-fpm

Der Frontend-Einrichtungsassistent kann die Konfigurationsdatei auf SUSE mit NGINX + php-fpm nicht speichern. Dies wird durch eine Einstellung in der Unit `/usr/lib/systemd/system/php-fpm.service` verursacht, die Zabbix daran hindert, in `/etc` zu schreiben. (eingeführt in PHP 7.4).

Es gibt zwei mögliche Workarounds:

- Setzen Sie die Option ProtectSystem in der systemd-Unit von php-fpm auf 'true' statt auf 'full'.
- Speichern Sie die Datei `/etc/zabbix/web/zabbix.conf.php` manuell.

Weiterleitung des Authorization-Headers

In einigen Fällen können Apache oder NGINX verhindern, dass der Authorization-Header in API-Anfragen Zabbix erreicht. Dies kann zu Authentifizierungsproblemen bei der Verwendung der Zabbix-API oder von Single-Sign-On-(SSO-)Diensten wie SAML mit Okta führen.

Um dies zu beheben, aktualisieren Sie die Konfiguration Ihres Webservers.

Für **Apache** fügen Sie, wenn Sie ihn als Reverse-Proxy verwenden (Nicht-CGI-Setup), die folgende Direktive zu `/etc/httpd/conf/httpd.co` (auf RHEL-basierten Systemen) oder `/etc/apache2/apache2.conf` (auf Debian/Ubuntu) hinzu:

```
SetEnvIfNoCase ^Authorization$ "(.+)" HTTP_AUTHORIZATION=$1
```

Wenn Apache Skripte direkt zur Verarbeitung von Anfragen ausführt (z. B. mit mod_cgi), fügen Sie stattdessen die folgende Direktive hinzu:

```
CGIPassAuth On
```

Im Gegensatz dazu verarbeitet **NGINX** den Authorization-Header automatisch. Wenn NGINX jedoch als Reverse-Proxy fungiert, können Sie den Authorization-Header explizit weiterleiten, indem Sie die folgenden Direktiven zu `/etc/nginx/nginx.conf` hinzufügen (für den Speicherort Ihres Zabbix-Frontend):

```
...
location / {
...
    proxy_set_header Authorization $http_authorization;
    proxy_pass http://backend_server;
...
}
```

Starten Sie nach der Aktualisierung der Konfiguration Ihren Webserver neu.

Weitere Informationen finden Sie unter:

- ZBX-22952
- Apache 2.4 + PHP-FPM and Authorization headers
- SetEnvIfNoCase und CGIPassAuth-Direktiven
- NGINX Reverse Proxy

Chromium für den Zabbix-Web-Service unter Ubuntu 20

Obwohl der Zabbix-Web-Service in den meisten Fällen mit Chromium ausgeführt werden kann, verursacht die Verwendung von Chromium unter Ubuntu 20.04 den folgenden Fehler:

```
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
```

Dieser Fehler tritt auf, weil `/var/lib/zabbix` als Home-Verzeichnis des Benutzers „zabbix“ verwendet wird.

Benutzerdefinierte MySQL-Fehlercodes

Wenn Zabbix erkennt, dass die Backend-Datenbank nicht erreichbar ist, sendet es eine Benachrichtigung und versucht weiterhin, eine Verbindung herzustellen. Für bestimmte Datenbank-Engines werden spezifische Fehlercodes erkannt. In MySQL umfassen diese erkannten Fehlercodes:

- 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

Zusätzlich kann bei der Verwendung von Zabbix mit einer MySQL-Installation auf Azure die allgemeine Fehlermeldung *[9002] Some errors occurred* in den Zabbix-Protokollen erscheinen. Diese Meldung wird von der Datenbank an den Zabbix Server oder Proxy gesendet. Um die Ursache des Fehlers zu ermitteln, prüfen Sie bitte die Azure-Protokolle.

Ungültige reguläre Ausdrücke nach dem Wechsel zu PCRE2

In Zabbix 6.0 wurde die Unterstützung für PCRE2 hinzugefügt. Obwohl PCRE weiterhin unterstützt wird, wurden die Zabbix-Installationspakete für RHEL 7 und neuer, SLES (alle Versionen), Debian 9 und neuer sowie Ubuntu 16.04 und neuer auf die Verwendung von PCRE2 umgestellt. Der Wechsel zu PCRE2 bringt zwar viele Vorteile mit sich, kann jedoch dazu führen, dass bestimmte vorhandene PCRE-Regexp-Muster ungültig werden oder sich anders verhalten. Insbesondere betrifft dies das Muster *[\w-\.]*. Um diesen Regexp wieder gültig zu machen, ohne die Semantik zu beeinträchtigen, ändern Sie den Ausdruck in *[-\w\.]* . Dies liegt daran, dass PCRE2 das Minuszeichen als Trennzeichen behandelt und dadurch innerhalb einer Zeichenklasse einen Bereich erstellt.

Geomap-Widget-Fehler

Die Karten im Geomap-Widget werden möglicherweise nicht korrekt geladen, wenn Sie von einer älteren Zabbix-Version mit NGINX aktualisiert haben und während des Upgrades nicht zur neuen NGINX-Konfigurationsdatei gewechselt sind.

Um das Problem zu beheben, können Sie die alte Konfigurationsdatei verwerfen, die Konfigurationsdatei aus dem Paket der aktuellen Version verwenden und sie wie in den Download-Anweisungen im Abschnitt *e. PHP für das Zabbix-Frontend konfigurieren* beschrieben neu konfigurieren.

Alternativ können Sie eine vorhandene NGINX-Konfigurationsdatei manuell bearbeiten (typischerweise `/etc/zabbix/nginx.conf`). Öffnen Sie dazu die Datei und suchen Sie den folgenden Block:

```
location ~ /(api\/|conf[^\.]|include|locale|vendor) {
        deny            all;
        return          404;
}
```

Ersetzen Sie diesen Block dann durch:

```
location ~ /(api\/|conf[^\.]|include|locale) {
        deny            all;
        return          404;
}

location /vendor {
        deny            all;
        return          404;
}
```

Vorverarbeitung — globale Variablen sind unsicher

JavaScript in der Vorverarbeitung wird pro Anfrage ausgeführt, aber Zuweisungen an nicht deklarierte Bezeichner (zum Beispiel `secret = value`) erzeugen implizite globale Variablen, die über die aktuelle Ausführung hinaus bestehen bleiben können. Das Speichern sensibler Daten (Token, Passwörter usw.) in impliziten globalen Variablen erhöht das Risiko einer versehentlichen Offenlegung oder Wiederverwendung durch nachfolgende Vorverarbeitungsläufe oder andere Integrationen, die in derselben Umgebung ausgeführt werden.

Verlassen Sie sich nicht auf implizite globale Variablen. Deklarieren Sie Variablen immer mit `var` oder `const`, und vermeiden Sie es, Geheimnisse an globale Objekte anzuhängen (zum Beispiel `globalThis` oder `window`). Es gibt keine unterstützte Möglichkeit, integrierte globale Objekte innerhalb der Vorverarbeitung zu überschreiben.

Sicheres Beispiel:

```
var apiToken = payload.token;
var count = 1;
return JSON.stringify({ token: apiToken, calls: count });
```

Prozessorgruppen unter Windows

Laut der Microsoft-Dokumentation haben Systeme mit weniger als 64 logischen Prozessoren immer nur eine einzige Prozessorgruppe, Gruppe 0. Zabbix-Benutzer haben jedoch einen seltenen Fehler ZBX-20260 gemeldet, bei dem auf Systemen mit 64 oder

weniger logischen Prozessoren zwei Prozessorgruppen vorhanden sind. Dies führte dazu, dass die Leistungsindikatoren "\Processor(n)" nur für eine von zwei Prozessorgruppen verfügbar waren. Die eigentliche Ursache dieses Fehlers ist nicht bekannt. Ein ähnlicher Fall wurde jedoch auf stackoverflow.com beschrieben; dort lag die Ursache in der Interaktion zwischen BIOS und Windows.

Einschränkungen der Filterung mit utf8mb4-Sortierungen

Filter (z. B. unter *Datenerfassung* > *Wartung*) funktionieren möglicherweise nicht korrekt, wenn sie auf Entitäten angewendet werden, die bestimmte Unicode-Zeichen enthalten (z. B. ℓ, ∉). Dieses Problem entsteht dadurch, wie die standardmäßige Sortierung utf8mb4_bin für MySQL- oder MariaDB-Datenbanken die Sortierung und den Vergleich von Unicode-Zeichen verarbeitet.

Um diese Einschränkung zu beheben, können Benutzer die Sortierung von Datenbankspalten auf Alternativen wie utf8mb4_0900_bin, utf8mb4_0900_ai_ci oder utf8mb4_unicode_520_ci ändern. Beachten Sie jedoch, dass das Ändern der Sortierung zu unerwartetem Verhalten bei der Verarbeitung von Leerzeichen sowie bei der Sortierung und Filterung anderer Zeichen führen kann.

Weitere Informationen zum Ändern von Sortierungen finden Sie in der MySQL-Dokumentation oder der MariaDB-Dokumentation. Einzelheiten zu den Unterschieden zwischen Sortierungen finden Sie unter Unicode Character Sets in der MySQL-Dokumentation.

Profiling übermäßiger Speichernutzung mit tcmalloc

Wenn Sie vermuten, dass Ihre Zabbix-Installation zu viel Speicher verwendet, können Sie die Speicherprofilierungsfunktion von tcmalloc nutzen, um den Speicherverbrauch von Zabbix Server/Proxy zu untersuchen.

1. Wenn Sie Zabbix aus den Quellen installieren, konfigurieren Sie zusätzliche Flags:

```
export CFLAGS="-std=gnu99 -g -O0"
```

Das Flag `-std=gnu99` ist für das Erstellen von Zabbix Server, Zabbix Proxy oder Zabbix Agent erforderlich. Das Flag `-g` fügt zusätzliche Debugging-Informationen hinzu, während `-O0` Optimierungen deaktiviert, die die Profilierung von tcmalloc beeinträchtigen können.

2. Setzen Sie vor dem Starten des Zabbix Server die folgenden Umgebungsvariablen. Diese Variablen teilen tcmalloc mit, wie die Speichernutzung verfolgt und gemeldet werden soll:

```
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. Lösen Sie einen Profil-Dump aus, indem Sie Signal 5 an den Zielprozess senden. Ersetzen Sie 1234 durch die tatsächliche Prozess-ID (PID):

```
kill -5 1234
```

4. Geben Sie das erzeugte Profil aus:

```
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
```

In diesem Beispiel ist zbx_malloc2 für fast alle Speicherzuweisungen verantwortlich.

Siehe auch:

- ZBX-25050 und ZBX-25584 für die zugehörigen Problemberichte.
- GCC Option Summary zu Kompilierungsoptionen (`-std=gnu99`, `-g`, `-O0` usw.).

- Gperftools Heap Profiler Dokumentation zu Umgebungsvariablen für die tcmalloc-Profilierung.

MySQL 8.0 Group Replication im Multi-Primary-Modus

Bei der Verwendung von MySQL 8.0 Group Replication im Multi-Primary-Modus kann während des Commit-Vorgangs von Transaktionen ein Fehler ähnlich dem folgenden auftreten:

```
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
```

Dieser Fehler scheint durch Probleme bei Rollback-Operationen mit Fremdschlüssel-Constraints ausgelöst zu werden.

Siehe auch:

- ZBX-26060 für den zugehörigen Problembericht.
- MySQL Bug #96758 "Rollbacks with Foreign Keys on single node" für das zugrunde liegende Problem.

Anmeldung und systemd

Wir empfehlen, einen *zabbix*-Benutzer als Systembenutzer anzulegen, also ohne Möglichkeit zur Anmeldung. Einige Benutzer ignorieren diese Empfehlung und verwenden dasselbe Konto zur Anmeldung (z. B. über SSH) am Host, auf dem Zabbix ausgeführt wird. Dies kann beim Abmelden zum Absturz des Zabbix-Daemons führen. In diesem Fall wird im Zabbix-Serverprotokoll etwa Folgendes angezeigt:

```
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
```

und im Zabbix-Agent-Protokoll:

```
zabbix_agentd [27796]: [file:'log.c',line:238] lock failed: [22] Invalid argument
```

Dies geschieht aufgrund der standardmäßigen systemd-Einstellung `RemoveIPC=yes`, die in `/etc/systemd/logind.conf` konfiguriert ist. Wenn Sie sich vom System abmelden, werden die zuvor von Zabbix erstellten Semaphore entfernt, was den Absturz verursacht.

Ein Zitat aus der systemd-Dokumentation:

```
RemoveIPC=

Controls whether System V and POSIX IPC objects belonging to the user shall be removed when the
user fully logs out. Takes a boolean argument. If enabled, the user may not consume IPC resources
after the last of the user's sessions terminated. This covers System V semaphores, shared memory
and message queues, as well as POSIX shared memory and message queues. Note that IPC objects of the
root user and other system users are excluded from the effect of this setting. Defaults to "yes".
```

Für dieses Problem gibt es zwei Lösungen:

1. (empfohlen) Verwenden Sie das *zabbix*-Konto ausschließlich für Zabbix-Prozesse und erstellen Sie für andere Zwecke ein eigenes Konto.
2. (nicht empfohlen) Setzen Sie `RemoveIPC=no` in `/etc/systemd/logind.conf` und starten Sie das System neu. Beachten Sie, dass `RemoveIPC` ein systemweiter Parameter ist. Eine Änderung wirkt sich auf das gesamte System aus.

Anforderungen

Hardware

Arbeitsspeicher

Zabbix benötigt sowohl physischen Speicher als auch Festplattenspeicher. Die Menge des benötigten Festplattenspeichers hängt natürlich von der Anzahl der Hosts und Parameter ab, die überwacht werden. Wenn Sie planen, eine lange Historie der überwachten Parameter aufzubewahren, sollten Sie mindestens einige Gigabyte einplanen, damit genügend Speicherplatz vorhanden ist, um die Historie in der Datenbank zu speichern. Jeder Zabbix-Daemon-Prozess benötigt mehrere Verbindungen zu einem Datenbank-Server. Die für die Verbindung zugewiesene Speichermenge hängt von der Konfiguration der Datenbank-Engine ab.

> **Note:**
> Je mehr physischer Speicher vorhanden ist, desto schneller arbeitet die Datenbank (und damit auch Zabbix).

CPU

Zabbix und insbesondere die Zabbix-Datenbank können je nach Anzahl der überwachten Parameter und der gewählten Datenbank-Engine erhebliche CPU-Ressourcen erfordern.

Andere Hardware

Ein serieller Kommunikationsanschluss und ein serielles GSM-Modem sind erforderlich, um die SMS-Benachrichtigungsunterstützung in Zabbix zu verwenden. Ein USB-zu-Seriell-Konverter funktioniert ebenfalls.

Beispiele für die Hardwarekonfiguration

Die Tabelle enthält Beispiele für Hardwarekonfigurationen unter der Annahme einer **Linux/BSD/Unix**-Plattform.

Dies sind Beispiele für Größe und Hardwarekonfiguration als Ausgangspunkt. Jede Zabbix-Installation ist einzigartig. Stellen Sie sicher, dass Sie die Leistung Ihres Zabbix-Systems in einer Staging- oder Entwicklungsumgebung benchmarken, damit Sie Ihre Anforderungen vollständig verstehen, bevor Sie die Zabbix-Installation in der Produktionsumgebung bereitstellen.

Größe der Installation	Überwachte Metriken[1]	CPU/vCPU-Kerne	Arbeitsspeicher (GiB)	Datenbank	Amazon EC2[2]
Klein	1 000	2	8	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.large/m6g.large
Mittel	10 000	4	16	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.xlarge/m6g.xlarge
Groß	100 000	16	64	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.4xlarge/m6g.4xlarge
Sehr groß	1 000 000	32	96	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.8xlarge/m6g.8xlarge

[1] Für die Dimensionierung wird 1 überwachte Metrik ungefähr durch die Formel 1 Metrik = 1 Datenpunkt + 1 Auslöser + 1 Graph dargestellt. Dies ist nur eine Schätzregel und keine Formel zur Berechnung der überwachten Metriken.
 [2] Beispiel mit Amazon-EC2-Instanzen für allgemeine Zwecke unter Verwendung der ARM64- oder x86_64-Architektur. Bei der Evaluierung und beim Testen der Zabbix-Installation sollte vor der Installation in der Produktionsumgebung ein geeigneter Instanztyp wie Compute-, Memory- oder Storage-optimiert ausgewählt werden.

> **Note:**
> Die tatsächliche Konfiguration hängt stark von der Anzahl der aktiven Datenpunkte und den Aktualisierungsintervallen ab (Details finden Sie im Abschnitt Datenbankgröße dieser Seite). Es wird dringend empfohlen, die Datenbank bei großen Installationen auf einem separaten Server zu betreiben.

Unterstützte Plattformen

Aufgrund von Sicherheitsanforderungen und der geschäftskritischen Natur des Monitoring-Servers ist UNIX das einzige Betriebssystem, das die erforderliche Leistung, Fehlertoleranz und Ausfallsicherheit zuverlässig bereitstellen kann. Zabbix läuft auf marktführenden Versionen.

Zabbix-Komponenten sind für die folgenden Plattformen verfügbar und getestet:

Plattform	Server	Agent	Agent 2	Kommentare
Linux	x	x	x	

Plattform	Server	Agent	Agent 2	Kommentare
Windows	-	x	x	Der Zabbix-Agent wird auf allen Desktop- und Serverversionen ab Windows XP (64-Bit)/Server 2003 unterstützt. Zabbix Agent 2 wird auf allen Desktop- und Serverversionen ab Windows 10 (32-Bit)/Server 2016 unterstützt, da er nur mit einer unterstützten Go-Version kompiliert wird, um kritische Sicherheitslücken zu vermeiden. Seit Go 1.21 wurden die minimal erforderlichen Windows-Versionen angehoben, wodurch Windows 10/Server 2016 zur Mindestversion für Zabbix Agent 2 wird.
macOS	x	x	-	
IBM AIX	x	x	-	Der Zabbix-Agent funktioniert nicht auf AIX-Plattformen unterhalb der Versionen 6.1 TL07 / 7.1 TL01.
FreeBSD	x	x	-	
OpenBSD	x	x	-	
Solaris	x	x	-	
NetBSD	x	x	-	
HP-UX	x	x	-	

Note:
Zabbix Server/Agent kann auch auf anderen Unix-ähnlichen Betriebssystemen funktionieren.

Attention:
Zabbix deaktiviert Core Dumps, wenn es mit Verschlüsselung kompiliert wurde, und startet nicht, wenn das System das Deaktivieren von Core Dumps nicht zulässt.

Erforderliche Software

Zabbix basiert auf modernen Webservern, führenden Datenbank-Engines und der Skriptsprache PHP.

Externe Software von Drittanbietern

Wenn die erforderliche Software/Bibliothek als obligatorisch angegeben ist, ist sie zwingend erforderlich.
Optionale Komponenten werden für die Unterstützung bestimmter Funktionen benötigt.

Software	Obligatorischer Status	Unterstützte Versionen	Kommentare
*MySQL/Percona	Eine von beiden	8.4.0-9.7.X	Erforderlich, wenn MySQL (oder Percona) als Backend-Datenbank von Zabbix verwendet wird. Die InnoDB-Engine ist erforderlich. Wir empfehlen, zum Erstellen von Server/Proxy die Bibliothek C API (libmysqlclient) zu verwenden. Wir empfehlen dringend, in Produktionsumgebungen ausschließlich LTS-Releases zu verwenden. Releases mit der Bezeichnung „rolling release“, „early access“ oder ähnlich werden nicht empfohlen.
MariaDB		10.11.00-12.3.X	Die InnoDB-Engine ist erforderlich. Wir empfehlen, zum Erstellen von Server/Proxy die Bibliothek MariaDB Connector/C zu verwenden. Siehe auch: Mögliche Deadlocks mit MariaDB und Zugriff auf UI-Elemente mit MariaDB 10.5.1-10.5.9. Wir empfehlen dringend, in Produktionsumgebungen ausschließlich LTS-Releases zu verwenden. Releases mit der Bezeichnung „rolling release“, „early access“ oder ähnlich werden nicht empfohlen.

Software	Obligatorischer Status	Unterstützte Versionen	Kommentare
PostgreSQL		15.0-18.X	Erforderlich, wenn PostgreSQL als Backend-Datenbank von Zabbix verwendet wird. Abhängig von der Installationsgröße kann es erforderlich sein, die PostgreSQL-Konfigurationseigenschaft *work_mem* zu erhöhen (der Standardwert beträgt 4 MB), damit die von der Datenbank für einen bestimmten Vorgang verwendete Speichermenge ausreicht und die Abfrageausführung nicht zu viel Zeit in Anspruch nimmt.
TimescaleDB für PostgreSQL		2.20.X-2.29.X	Erforderlich, wenn TimescaleDB als PostgreSQL-Datenbankerweiterung verwendet wird. Stellen Sie sicher, dass Sie die TimescaleDB Community Edition installieren, die Komprimierung unterstützt. Details zur Kompatibilität der PostgreSQL- und TimescaleDB-Versionen finden Sie in der TimescaleDB-Dokumentation.
Elasticsearch		7.X	Erforderlich, wenn Elasticsearch als Backend-Datenbank von Zabbix verwendet wird. Elasticsearch kann nur zum Speichern von Verlaufsdaten verwendet werden und wird nur mit Zabbix-Servern unterstützt. Die Unterstützung von Elasticsearch ist derzeit experimentell. Siehe auch die erforderliche Software für Server/Proxy.
ClickHouse		26.3.0.0-26.9.X.X	Erforderlich, wenn ClickHouse als Backend-Datenbank von Zabbix zum Speichern von Verlaufsdaten (nur mit Zabbix-Servern unterstützt) oder als Telemetriedatenquelle verwendet wird. Um ClickHouse als Telemetriedatenquelle zu verwenden, ist ClickHouse 26.8.17.4 oder höher erforderlich. Die Systemzeit auf dem Zabbix-Server und dem ClickHouse-Server muss ebenfalls synchronisiert sein. Andernfalls können bei Telemetrieabfragen Daten fehlen oder dieselben Daten zweimal zurückgegeben werden.
SQLite	Optional	3.3.5-3.53.X	SQLite wird nur mit Zabbix-Proxys unterstützt. Erforderlich, wenn SQLite als Zabbix-Proxy-Datenbank verwendet wird.
smartmontools		7.1 oder höher	Erforderlich für Zabbix Agent 2.
who			Erforderlich für das Plugin zur Ermittlung der Benutzeranzahl.
dpkg			Erforderlich für das Plugin system.sw.packages.
pkgtool			Erforderlich für das Plugin system.sw.packages.
rpm			Erforderlich für das Plugin system.sw.packages.
pacman			Erforderlich für das Plugin system.sw.packages.
q applets			`qlist` und `qsize` als Bestandteil der q applets sind für das Plugin system.sw.packages unter Gentoo Linux erforderlich.

Note:
Obwohl Zabbix mit Datenbanken arbeiten kann, die in den Betriebssystemen verfügbar sind, empfehlen wir für eine optimale Nutzung, Datenbanken zu verwenden, die aus den offiziellen Repositories der Datenbankentwickler installiert wurden.

Frontend

Die minimal unterstützte Bildschirmbreite für das Zabbix-Frontend beträgt 1200px.

Wenn die erforderliche Software/Bibliothek als obligatorisch angegeben ist, wird sie zwingend benötigt. Optionale Komponenten werden für die Unterstützung bestimmter Funktionen benötigt.

Software	Obligatorisch	Unterstützte Versionen	Kommentare
PHP	Ja	8.2.0 - 8.5.X	
Apache	Eine von	2.4 oder höher	

Software	Obligatorisch	Unterstützte Versionen	Kommentare
Nginx		1.20 oder höher	
MySQL	Eine von	Siehe Drittanbieter- und externe Umgebungssoftware	
PostgreSQL			
PHP-Erweiterungen			
mysqli	Ja		Erforderlich, wenn MySQL als Zabbix-Backend-Datenbank verwendet wird.
pgsql			Erforderlich, wenn PostgreSQL als Zabbix-Backend-Datenbank verwendet wird.
bcmath			php-bcmath (*--enable-bcmath*)
mbstring			php-mbstring (*--enable-mbstring*)
sockets			php-net-socket (*--enable-sockets*); erforderlich für die Unterstützung von Benutzerskripten.
gd		2.0.28 oder höher	php-gd (falls vom Distributor als separates Paket bereitgestellt); die PHP-GD-Erweiterung muss PNG-Bilder (*--with-png-dir*), JPEG-Bilder (*--with-jpeg-dir*) und FreeType 2 (*--with-freetype-dir*) unterstützen. Version 2.3.0 oder höher kann erforderlich sein, um eine mögliche Überlappung von Text in Diagrammen für einige Frontend-Sprachen zu vermeiden.
libxml		2.6.15 oder höher	php-xml (falls vom Distributor als separates Paket bereitgestellt)
xmlwriter			php-xmlwriter (falls vom Distributor als separates Paket bereitgestellt)
xmlreader			php-xmlreader (falls vom Distributor als separates Paket bereitgestellt)
ctype			php-ctype (*--enable-ctype*)
session			php-session (falls vom Distributor als separates Paket bereitgestellt)
ldap	Nein		php-ldap; erforderlich für die LDAP-Authentifizierung.
openssl			php-openssl; erforderlich für die SAML-Authentifizierung.
gettext			php-gettext (*--with-gettext*); erforderlich für Übersetzungen.
cURL		7.19.4 oder höher	php-curl; erforderlich für Duo Universal Prompt MFA, SMTP-Authentifizierung und zum Testen der Telemetriedatenquelle.

Drittanbieter-Frontend-Bibliotheken, die mit Zabbix bereitgestellt werden:

Bibliothek	Obligatorisch	Bereitgestellte Version	Kommentare
jQuery JavaScript Library	Ja	3.6.0	JavaScript-Bibliothek, die den Prozess der browserübergreifenden Entwicklung vereinfacht.
jQuery UI		1.12.1	Eine Sammlung von Benutzeroberflächeninteraktionen, Effekten, Widgets und Themes, die auf jQuery aufbaut.
SAML PHP Toolkit		4.3.1	Ein PHP-Toolkit, das Unterstützung für die SAML-2.0-Authentifizierung hinzufügt, um die Anmeldung bei Zabbix zu ermöglichen.
Duo Universal PHP library		1.1.2	Eine PHP-Bibliothek, die Unterstützung für die Multi-Faktor-Authentifizierung mit Duo Universal Prompt für Zabbix hinzufügt.
Symfony Yaml Component		5.1.0	Fügt Unterstützung für den Export und Import von Zabbix-Konfigurationselementen im YAML-Format hinzu.

Note:
Zabbix funktioniert möglicherweise auch mit früheren Versionen von Apache, MySQL und PostgreSQL.

Attention:
Für andere Schriftarten als die standardmäßige DejaVu-Schriftart ist möglicherweise die PHP-Funktion imagerotate erforderlich. Wenn sie fehlt, werden diese Schriftarten möglicherweise bei der Anzeige eines Diagramms falsch dargestellt. Diese Funktion ist nur verfügbar, wenn PHP mit der gebündelten GD-Bibliothek kompiliert wurde, was bei Debian und anderen Distributionen nicht der Fall ist.

Drittanbieter-Bibliotheken, die zum Schreiben und Debuggen von Zabbix-Frontend-Code verwendet werden:

Bibliothek	Obligatorisch	Mindestversion	Beschreibung
Composer	Nein	2.4.1	Ein Paketmanager auf Anwendungsebene für PHP, der ein Standardformat für die Verwaltung von Abhängigkeiten von PHP-Software und erforderlichen Bibliotheken bereitstellt.
PHPUnit		8.5.29	Ein PHP-Unit-Testing-Framework zum Testen des Zabbix-Frontends.
SASS		3.4.22	Eine Präprozessor-Skriptsprache, die interpretiert und in Cascading Style Sheets (CSS) kompiliert wird.

Webbrowser auf Client-Seite

Cookies und JavaScript müssen aktiviert sein.

Die neuesten stabilen Versionen von Google Chrome, Mozilla Firefox, Microsoft Edge, Apple Safari und Opera werden unterstützt.

Warning:
Die Same-Origin-Policy für IFrames ist implementiert, was bedeutet, dass Zabbix nicht in Frames auf einer anderen Domain eingebettet werden kann.

 Seiten, die in einen Zabbix-Frame eingebettet sind, haben jedoch Zugriff auf das Zabbix Frontend (über JavaScript), wenn sich die in den Frame eingebettete Seite und das Zabbix Frontend auf derselben Domain befinden. Eine Seite wie `http://secure-zabbix.com/cms/page.html`, die in Dashboards auf `http://secure-zabbix.com/zabbix/` eingebettet ist, hat vollen JS-Zugriff auf Zabbix.

Server/Proxy

Wenn als obligatorisch angegeben, ist die erforderliche Software/Bibliothek zwingend notwendig. Optionale Komponenten werden zur Unterstützung bestimmter Funktionen benötigt.

Anforderung	Obligatorisch	Beschreibung
libpcre2	Ja	Die PCRE2-Bibliothek wird für die Unterstützung von Perl-kompatiblen regulären Ausdrücken (PCRE) benötigt. PCRE2 v10.x wird unterstützt.
libevent		Für die Interprozesskommunikation erforderlich. Version 2.0.10 oder höher.
libevent-pthreads		Für die Interprozesskommunikation erforderlich.
libpthread		Für die Unterstützung von Mutexen und Lese-/Schreibsperren erforderlich (kann Bestandteil von libc sein).
libresolv		Für die DNS-Auflösung erforderlich (kann Bestandteil von libc sein).
libiconv		Für die Konvertierung von Textkodierungen und -formaten erforderlich (kann Bestandteil von libc sein). Für den Zabbix Server unter Linux obligatorisch.
libz		Für die Unterstützung der Komprimierung erforderlich.
libm		Mathematikbibliothek. Nur vom Zabbix Server benötigt.
libmysqlclient	Eine von	Erforderlich, wenn MySQL verwendet wird.
libmariadb		Erforderlich, wenn MariaDB verwendet wird.
libpq5		Erforderlich, wenn PostgreSQL verwendet wird; die Version von *libpq5* muss der Version der verwendeten PostgreSQL-Datenbank entsprechen oder höher sein.
libsqlite3		Erforderlich, wenn SQLite verwendet wird. Nur für den Zabbix Proxy erforderlich.
libOpenIPMI	Nein	Für die IPMI-Unterstützung erforderlich. Nur vom Zabbix Server benötigt.
libssh2 oder *libssh*		Erforderlich für SSH-Prüfungen. Version 1.8.0 oder höher (libssh2); 0.9.0 oder höher (libssh).

Anforderung	Obligatorisch	Beschreibung
libcurl		Erforderlich für die folgenden Funktionen: - Webüberwachung, VMware-Überwachung und HTTP-Agent-Datenpunkte(/manual/config/items/itemtypes/http) (für alle: Version 7.19.1 oder höher); - [web.page.*]-Datenpunkte des Zabbix Agent(/manual/config/items/itemtypes/zabbix_agent#web.page.get) (Version 7.19.1 oder höher; siehe auch die Anforderungen für den Agent); - SMTP-Authentifizierung (Basic: Version 7.20.0 oder höher; OAuth: Version 7.33 oder höher; siehe auch die Anforderungen für das Frontend); - Elasticsearch (Version 7.28.0 oder höher); - [Telemetry-Abfrage]-Datenpunkte(/manual/config/items/itemtypes/telemetry_query) (Version 7.28.0 oder höher). Für alle Funktionen wird Version 7.28.0 oder höher empfohlen. Um aktualisierte cURL-Funktionen für `web.page.*`-Datenpunkte zu verwenden, starten Sie den Zabbix Server/Proxy neu. Verwenden Sie für die SMTP-Authentifizierung zur Laufzeit das Paket `libcurl-full`.
libxml2		Erforderlich für die VMware-Überwachung und die XML-XPath-Vorverarbeitung.
net-snmp		Für die SNMP-Unterstützung erforderlich. Version 5.3.0 oder höher. Die Unterstützung starker Verschlüsselungsprotokolle (AES192/AES192C, AES256/AES256C) ist ab der net-snmp-Bibliothek 5.8 verfügbar; auf RHEL-8+-basierten Systemen wird empfohlen, net-snmp 5.8.15 oder höher zu verwenden.
libunixodbc		Erforderlich für die Datenbanküberwachung.
libgnutls oder *libopenssl*		Bei Verwendung von Verschlüsselung erforderlich. Mindestversionen: *libgnutls* - 3.1.18, *libopenssl* - 1.0.1
libldap		Für die LDAP-Unterstützung erforderlich.
fping		Erforderlich für ICMP-Ping-Datenpunkte.
c-ares		Für die asynchrone DNS-Auflösung erforderlich, wenn Zabbix mit der Option `--with-ares` konfiguriert wurde. Andernfalls wird *libevent* verwendet. Mindestversion: 1.16.0. Die Funktion zur Zwischenspeicherung und zum Failover von DNS-Abfragen (Abfrage-Cache standardmäßig aktiviert) erfordert c-ares 1.26.0 oder höher. Auf Systemen, auf denen c-ares über das Betriebssystempaket bereitgestellt wird, muss sichergestellt werden, dass die installierte Laufzeitbibliothek die für DNS-Caching und Failover erforderlichen Resolver-Funktionen bereitstellt.
g++		C++-Compiler. Erforderlich, um den Zabbix Proxy mit der Option `--with-apm` zu kompilieren.
gRPC		Erforderlich für die Unterstützung von APM auf dem Zabbix Proxy. Erforderliche Bibliotheken: *grpc++*, *grpc*, *gpr*. Das Dienstprogramm `grpc_cpp_plugin` wird ebenfalls zum Kompilieren des Zabbix Proxy benötigt. Die erforderliche Version ist in den Standardpaketquellen der folgenden Betriebssysteme verfügbar: - Ubuntu 23.04 oder höher; - Debian 12 (Bookworm) oder höher; - Alpine Linux 3.15 oder höher; - RHEL, Rocky Linux, CentOS Stream 9 oder höher (nur aus dem EPEL-Repository); - Fedora X oder höher. Auf älteren Betriebssystemversionen muss *gRPC* aus dem Quellcode erstellt werden.
protobuf		Erforderlich für die Unterstützung von APM auf dem Zabbix Proxy. Der Compiler `protoc` wird ebenfalls zum Kompilieren des Zabbix Proxy benötigt.
Abseil		Erforderlich für die Unterstützung von APM auf dem Zabbix Proxy. Erforderliche Bibliotheken: *absl_synchronization*, *absl_base*. Wird normalerweise zusammen mit *gRPC* installiert.

Agent

Requirement	Mandatory status	Description
libpcre2	Yes	Die PCRE2-Bibliothek wird für die Unterstützung von Perl Compatible Regular Expression (PCRE) benötigt. PCRE2 v10.x wird unterstützt. Erforderlich für die Protokollüberwachung. Auch unter Windows erforderlich.
libpthread		Erforderlich für die Unterstützung von Mutex und Read-Write-Lock (kann Teil von libc sein). Unter Windows nicht erforderlich.
libresolv		Erforderlich für die DNS-Auflösung (kann Teil von libc sein). Unter Windows nicht erforderlich.
libiconv		Erforderlich für die Textkodierung bzw. Formatkonvertierung nach UTF-8 in Protokolleinträgen, Dateiinhalten, Datei-Regex und regmatch-Datenpunkten (kann Teil von libc sein). Unter Windows nicht erforderlich.
libgnutls or *libopenssl*	No	Erforderlich bei Verwendung von Verschlüsselung. Mindestversionen: *libgnutls* - 3.1.18, *libopenssl* - 1.0.1 Unter Microsoft Windows ist OpenSSL 1.1.1 oder höher erforderlich.
libldap		Erforderlich, wenn LDAP verwendet wird. Unter Windows nicht unterstützt.
libcurl		Erforderlich für die erweiterte Unterstützung von Zabbix Agent web.page.*-Datenpunkten. Ohne libcurl ist die grundlegende Funktionalität verfügbar (z. B. web.page.get[http://example.com]). Mit libcurl unterstützt der Agent zusätzliche Funktionen, z. B. HTTP-URLs mit Anmeldedaten (z. B. http://user:password@example.com) und HTTPS-URLs. Version 7.19.1 oder höher ist erforderlich (7.28.0 oder höher wird empfohlen). Um aktualisierte cURL-Funktionen zu verwenden, starten Sie den Zabbix Agent neu.
libmodbus		Nur erforderlich, wenn Modbus-Überwachung verwendet wird. Version 3.0 oder höher.

Agent 2

Requirement	Mandatory status	Description
Go	Yes	Erforderlich, um Zabbix Agent 2 und seine Plugins aus dem Quellcode zu erstellen. Go 1.25 oder höher wird unterstützt. Installationsanweisungen finden Sie unter go.dev. Die von Zabbix Agent 2 und seinen Plugins verwendeten Go-Bibliotheken sind im Zabbix-Git-Repository aufgeführt (im Repository als `indirect` markierte Bibliotheken sind Abhängigkeiten anderer erforderlicher Bibliotheken): - Zabbix Agent 2 - Plugin support - PostgreSQL - MongoDB - MSSQL - Ember+ - NVIDIA GPU - Example plugin
libpcre2	Yes	Die PCRE2-Bibliothek ist für die Unterstützung von Perl Compatible Regular Expression (PCRE) erforderlich. PCRE2 v10.x wird unterstützt. Erforderlich für die Protokollüberwachung. Außerdem unter Windows erforderlich.
libopenssl	No	Erforderlich bei Verwendung von Verschlüsselung. Auf UNIX-Plattformen ist OpenSSL 1.0.1 oder höher erforderlich. Die OpenSSL-Bibliothek muss mit aktivierter PSK-Unterstützung kompiliert sein. LibreSSL wird nicht unterstützt. Auf Microsoft-Windows-Systemen ist OpenSSL 1.1.1 oder höher erforderlich.

Web-Service

Die neueste stabile Version von Google Chrome wird für die Erstellung geplanter Berichte mit dem Zabbix-Web-Service unterstützt.

Die erforderliche Go-Version zum Erstellen des Web-Service entspricht derjenigen, die für Zabbix agent 2 verwendet wird.

Java gateway

Wenn Sie Zabbix aus dem Quellcode-Repository oder einem Archiv bezogen haben, sind die erforderlichen Abhängigkeiten bereits im Quellcodebaum enthalten.

Wenn Sie Zabbix aus dem Paket Ihrer Distribution bezogen haben, werden die erforderlichen Abhängigkeiten bereits vom Paketsystem bereitgestellt.

In beiden oben genannten Fällen ist die Software einsatzbereit und es sind keine zusätzlichen Downloads erforderlich.

Wenn Sie jedoch Ihre eigenen Versionen dieser Abhängigkeiten bereitstellen möchten (beispielsweise wenn Sie ein Paket für eine Linux-Distribution vorbereiten), finden Sie unten eine Liste der Bibliotheksversionen, mit denen Java gateway bekanntermaßen funktioniert. Zabbix funktioniert möglicherweise auch mit anderen Versionen dieser Bibliotheken.

Die folgende Tabelle enthält die JAR-Dateien, die derzeit im Originalcode zusammen mit Java gateway gebündelt sind:

Bibliothek	Erforderlicher Status	Gebündelte Version	Kommentare
android-json	Ja	4.3r1	JSON (JavaScript Object Notation) ist ein leichtgewichtiges Datenaustauschformat. Dies ist die mit `org.json` kompatible Android-Implementierung, die aus dem Android SDK extrahiert wurde.
logback-classic		1.5.25	
logback-core		1.5.25	
slf4j-api		2.0.16	

Java gateway kann entweder mit Oracle Java oder dem quelloffenen OpenJDK (Version 8 oder neuer) erstellt werden. Die von Zabbix bereitgestellten Pakete werden mit OpenJDK kompiliert. Die folgende Tabelle enthält die OpenJDK-Pakete, die von den jeweiligen Distributionen zum Erstellen der Zabbix-Pakete verwendet werden:

Distribution	OpenJDK-Paket
AlmaLinux 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.12.0.8-1.2)
AlmaLinux 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2)
AlmaLinux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.332.b09-2; arm64: 1.8.0.382.b05-2)
Amazon Linux 2023	`java-22-amazon-corretto-devel` (amd64, arm64: 22.0.2+9-1)
CentOS Stream 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.11.0.10-2)
CentOS Stream 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2)
CentOS Stream 8	`java-1.8.0-openjdk-devel` (amd64, arm64: 1.8.0.362.b08-3)
CentOS 7	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.282.b08-1)
Debian 13	`default-jdk-headless` (amd64, arm64: 2:1.21-76)
Debian 12	`default-jdk-headless` (amd64, arm64: 2:1.17-74)
Debian 11	`default-jdk-headless` (amd64: 2:1.11-72)
OpenSUSE Leap 16	`java-21-openjdk-devel` (amd64, arm64: 21.0.11.0-160000.2.1)
OpenSUSE Leap 15	`java-17-openjdk-devel` (amd64: 17.0.5.0-150400.3.9.3; arm64: 17.0.8.0-150400.3.27.1)
Oracle Linux 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.11.0.10-2.0.1)
Oracle Linux 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2.0.1)
Oracle Linux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.372.b07-4.0.1); `java-11-openjdk-devel` (arm64: 11.0.20.0.8-3.0.1)
Oracle Linux 7	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.282.b08-1)
Raspberry Pi OS 13	`default-jdk-headless` (arm64, armhf: 2:1.21-76)
Raspberry Pi OS 12	`default-jdk-headless` (arm64, armhf: 2:1.17-74)
Raspberry Pi OS 11	`default-jdk-headless` (arm64: 2:1.11-72; armhf: 2:1.11-72+b4)
RHEL 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.12.0.8-1.2)
RHEL 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2)
RHEL 8	`java-1.8.0-openjdk-devel` (amd64: 17.0.15.0.6-2)
RHEL 7	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.282.b08-1)
Rocky Linux 10	`java-21-openjdk-devel` (amd64, arm64: 21.0.12.0.8-1.2)
Rocky Linux 9	`java-17-openjdk-devel` (amd64, arm64: 17.0.15.0.6-2)
Rocky Linux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.372.b07-4; arm64: 1.8.0.382.b05-2)
SLES 16	`java-21-openjdk-devel` (amd64, arm64: 21.0.11.0-160000.1.1)
SLES 15	`java-17-openjdk-devel` (amd64: 17.0.5.0-150400.3.9.3; arm64: 17.0.8.0-150400.3.27.1)
Ubuntu 26.04	`default-jdk-headless` (amd64, arm64: 2:1.25-77)

Distribution	OpenJDK-Paket
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)

Bridge-Adapter

Der Zabbix-Bridge-Adapter hat keine besonderen Anforderungen.

Standardports

Die folgende Liste enthält Ports, die Zabbix-Komponenten in einer Standardkonfiguration verwenden können:

Zabbix-Komponente	Protokoll und Port	Verbindung initiiert von
Zabbix Agent	TCP 10050	Zabbix Server oder Zabbix Proxy (nur passive Prüfungen)
Zabbix Agent 2	TCP 10050	Zabbix Server oder Zabbix Proxy (nur passive Prüfungen)
Zabbix Server	TCP 10051	Aktive Zabbix Agents, aktiver Zabbix Proxy, Zabbix sender, Zabbix Frontend
Zabbix Proxy	TCP 10051	Aktive Zabbix Agents, Zabbix sender, Zabbix Server (nur passiver Proxy)
Zabbix Java gateway	TCP 10052	Zabbix Server oder Proxy
Zabbix web service	TCP 10053	Zabbix Server
Zabbix MCP server	TCP 8443	MCP-Clients (KI-Agents)
Zabbix Frontend	TCP 80	Webbrowser, Zabbix web service (geplante Berichte), API-Clients
	TCP 443	
SNMP-Traps (snmptrapd)	UDP 162	SNMP-Geräte
Zabbix bridge adapter	TCP 10054	Zabbix Server

Warning:
Öffnen Sie nur die Ports, die von den tatsächlich ausgeführten Komponenten benötigt werden, und beschränken Sie jede Firewall-Regel auf die spezifischen Hosts, die eine Verbindung herstellen müssen (erlauben Sie beispielsweise Port 10050 nur vom Zabbix Server oder Proxy und nicht von beliebigen Adressen). Setzen Sie Zabbix-Ports nicht direkt dem Internet aus.
 Härten Sie die Installation, bevor Sie Ports öffnen: Aktivieren Sie die Verschlüsselung zwischen den Komponenten, beschränken Sie die zulässigen Hosts in den Konfigurationsdateien der Komponenten und lesen Sie die bewährten Sicherheitsverfahren.

Datenbankgröße

Zabbix-Konfigurationsdaten benötigen eine feste Menge an Speicherplatz auf der Festplatte und wachsen kaum.

Die Größe der Zabbix-Datenbank hängt hauptsächlich von diesen Variablen ab, die die Menge der gespeicherten historischen Daten bestimmen:

- Anzahl der verarbeiteten Werte pro Sekunde

Dies ist die durchschnittliche Anzahl neuer Werte, die der Zabbix Server jede Sekunde empfängt. Wenn wir zum Beispiel 3000 Datenpunkte zur Überwachung mit einer Aktualisierungsrate von 60 Sekunden haben, wird die Anzahl der Werte pro Sekunde wie folgt berechnet: 3000/60 = **50**.

Das bedeutet, dass jede Sekunde 50 neue Werte zur Zabbix-Datenbank hinzugefügt werden.

- Housekeeper-Einstellungen für History

Zabbix speichert Werte für einen festen Zeitraum, normalerweise mehrere Wochen oder Monate. Jeder neue Wert benötigt eine bestimmte Menge an Speicherplatz für Daten und Index.

Wenn wir also 30 Tage History behalten möchten und 50 Werte pro Sekunde empfangen, beträgt die Gesamtzahl der Werte ungefähr (**30***24*3600)* **50** = 129.600.000, also etwa 130M Werte.

Je nach verwendeter Datenbank-Engine und Art der empfangenen Werte (Floats, Integer, Zeichenketten, Protokolldateien usw.) kann der Speicherplatz für einen einzelnen Wert zwischen 40 Bytes und mehreren hundert Bytes variieren. Normalerweise sind es etwa 90 Bytes pro Wert für numerische Datenpunkte[2]. In unserem Fall bedeutet das, dass 130M Werte 130M * 90 Bytes = **10.9GB** Speicherplatz benötigen.

Note:
Die Größe von Text-/Log-Datenpunktwerten lässt sich nicht exakt vorhersagen, aber Sie können mit etwa 500 Bytes pro Wert rechnen.

- Housekeeper-Einstellung für Trends

Zabbix speichert für jeden Datenpunkt einen 1-Stunden-Satz aus Max/Min/Avg/Count-Werten in der Tabelle **trends**. Die Daten werden für Trendanalysen und Langzeitdiagramme verwendet. Der Zeitraum von einer Stunde kann nicht angepasst werden.

Die Zabbix-Datenbank benötigt je nach Datenbanktyp etwa 90 Bytes pro Eintrag. Nehmen wir an, wir möchten Trenddaten für 5 Jahre aufbewahren. Werte für 3000 Datenpunkte benötigen 3000*24*365* **90** = **2.2GB** pro Jahr oder **11GB** für 5 Jahre.

- Housekeeper-Einstellungen für Events

Jedes Zabbix-Event benötigt ungefähr 250 Bytes Speicherplatz[1]. Es ist schwer, die Anzahl der von Zabbix täglich erzeugten Events zu schätzen. Im schlimmsten Fall können wir annehmen, dass Zabbix ein Event pro Sekunde erzeugt.

Für jedes wiederhergestellte Event wird ein event_recovery-Datensatz erstellt. Normalerweise werden die meisten Events wiederhergestellt, daher können wir einen event_recovery-Datensatz pro Event annehmen. Das bedeutet zusätzliche 80 Bytes pro Event.

Optional können Events Tags haben, wobei jeder Tag-Datensatz ungefähr 100 Bytes Speicherplatz benötigt[1]. Die Anzahl der Tags pro Event (#tags) hängt von der Konfiguration ab. Daher benötigt jeder zusätzliche #tags * 100 Bytes Speicherplatz.

Das bedeutet: Wenn wir Events für 3 Jahre aufbewahren möchten, wären dafür 3*365*24*3600* (250+80+#tags*100) = **~30GB**+#tags*100B Speicherplatz erforderlich[2].

Note:
[1] Mehr bei nicht-ASCII-Eventnamen, Tags und Werten.
 [2] Die Größenangaben basieren auf MySQL und können bei anderen Datenbanken abweichen.

Die Tabelle enthält Formeln, mit denen der für das Zabbix-System erforderliche Speicherplatz berechnet werden kann:

Parameter	Formel für den erforderlichen Speicherplatz (in Bytes)
Zabbix-Konfiguration	Feste Größe. Normalerweise 10MB oder weniger.
History	days*(items/refresh rate)*24*3600*bytes items : Anzahl der Datenpunkte days : Anzahl der Tage, für die History aufbewahrt werden soll refresh rate : durchschnittliche Aktualisierungsrate der Datenpunkte bytes : Anzahl der Bytes, die zum Speichern eines einzelnen Werts erforderlich sind, abhängig von der Datenbank-Engine, normalerweise ~90 Bytes.
Trends	days*(items/3600)*24*3600*bytes items : Anzahl der Datenpunkte days : Anzahl der Tage, für die History aufbewahrt werden soll bytes : Anzahl der Bytes, die zum Speichern eines einzelnen Trends erforderlich sind, abhängig von der Datenbank-Engine, normalerweise ~90 Bytes.
Events	days*events*24*3600*bytes events : Anzahl der Events pro Sekunde. Im schlimmsten Fall ein (1) Event pro Sekunde. days : Anzahl der Tage, für die History aufbewahrt werden soll bytes : Anzahl der Bytes, die zum Speichern eines einzelnen Events erforderlich sind, abhängig von der Datenbank-Engine, normalerweise ~330 + durchschnittliche Anzahl der Tags pro Event * 100 Bytes.

Der insgesamt erforderliche Speicherplatz kann also wie folgt berechnet werden:

Konfiguration + History + Trends + Events

Der Speicherplatz wird nach der Installation von Zabbix NICHT sofort verwendet. Die Datenbankgröße wächst zunächst und hört dann zu einem bestimmten Zeitpunkt auf zu wachsen, abhängig von den Housekeeper-Einstellungen.

Zeitsynchronisierung

Es ist sehr wichtig, auf dem Server, auf dem Zabbix läuft, eine präzise Systemzeit zu haben. ntpd ist der am weitesten verbreitete Daemon, der die Zeit des Hosts mit der Zeit anderer Rechner synchronisiert. Es wird dringend empfohlen, auf allen Systemen, auf denen Zabbix-Komponenten ausgeführt werden, eine synchronisierte Systemzeit beizubehalten.

Was ist neu und Hinweise zum Upgrade: 8.0

Zabbix 8.0.0 baut auf Zabbix 7.4.0 auf und bietet neue Funktionen und Verbesserungen.

Auf dieser Seite außerdem:

- Was ist neu in den Vorlagen, einschließlich neuer Vorlagen und Änderungen an bestehenden Vorlagen.
- Hinweise zum Upgrade, einschließlich inkompatibler Änderungen.

Siehe auch: API-Änderungen.

Upgrade-Hinweise

Diese Hinweise gelten für das Upgrade von Zabbix 7.4.x auf Zabbix 8.0.0.

Alle Hinweise sind in folgende Kategorien unterteilt:

- **Breaking changes** - Änderungen, die bestehende Installationen beeinträchtigen können, sowie andere wichtige Informationen zum Upgrade-Prozess
- **Other** - alle übrigen Informationen, die die Änderungen an der Zabbix-Funktionalität beschreiben

Siehe auch:

- Upgrade procedure für alle relevanten Informationen zum Upgrade von Versionen vor Zabbix 8.0.0;
- Upgrading HA cluster für Anweisungen zum Upgrade von Servern in einem **High-Availability**-(HA)-Cluster.

Breaking changes

Andere

Neuerungen und Hinweise zum Upgrade: 8.0.x

Diese Seite stellt alle neuen Funktionen vor, die in den Nebenversionen der Zabbix-Hauptversion enthalten sind. Änderungen an Vorlagen und Hinweise zum Upgrade sind ebenfalls enthalten.

Siehe auch Neuerungen der Hauptversion.

Was ist neu in Zabbix 8.0.1 Diese Version wurde noch nicht veröffentlicht.

Siehe Upgrade-Hinweise für diese Version.

Upgrade-Hinweise

Zabbix-Appliance

Übersicht Die Zabbix Appliance ermöglicht die sofortige Bereitstellung von Zabbix Server und Frontend, anstatt diese manuell einzurichten oder einen vorhandenen Server für Zabbix wiederzuverwenden.

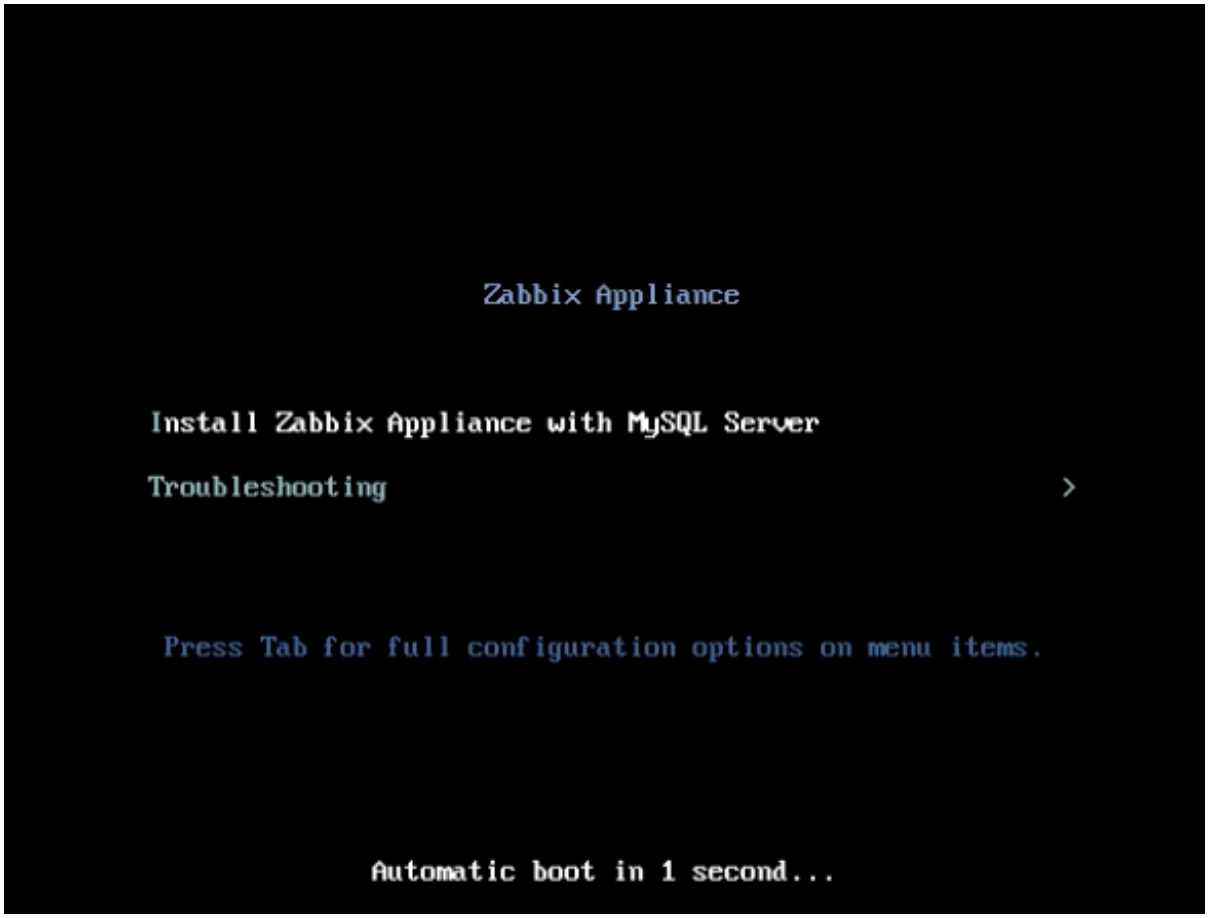
Die Appliance für Zabbix 8.0 basiert auf Derivaten von Red Hat Enterprise Linux 10, wie AlmaLinux oder Rocky Linux, und enthält einen vorkonfigurierten Zabbix Server, der auf MySQL ausgeführt wird, sowie das Frontend, das auf einem NGINX-Webserver ausgeführt wird.

> **Attention:**
> Diese Appliance ist für die Evaluierung von Zabbix vorgesehen. Von ihrer Verwendung in ernsthaften Produktionsumgebungen wird abgeraten.

Appliance-Images stehen in den folgenden Formaten zum Download bereit:

- Installations-CD/DVD (.iso)
- VMware (.vmx) – siehe Hinweise
- Open Virtualization Format (.ovf)
- Microsoft Hyper-V (.vhd/.vhdx) – siehe Hinweise
- KVM, Parallels, QEMU, USB-Stick, VirtualBox, Xen (.raw) – siehe Hinweise
- KVM, QEMU (.qcow2)

Bootmenü der Zabbix-Installations-CD/DVD:



Schnellstart Voraussetzungen

1. Stellen Sie sicher, dass der Host-Rechner über ausreichende Ressourcen verfügt, um die Systemanforderungen der virtuellen Maschine zu erfüllen:

- *RAM*: 4 GB
- *Festplattenspeicher*: Für die virtuelle Maschine sollten mindestens 8 GB zugewiesen werden
- *CPU*: mindestens 2 Kerne

2. Falls noch nicht installiert, installieren Sie die Virtualisierungssoftware zum Booten des Appliance-Images (zum Beispiel VirtualBox).
3. Laden Sie die Appliance in dem von Ihrer Virtualisierungssoftware unterstützten Format herunter.
4. Überprüfen Sie die Netzwerkeinstellungen, um sicherzustellen, dass DHCP auf dem Host-Rechner aktiviert ist.

Installation

1. Starten Sie die Appliance-VM von dem heruntergeladenen Image. Anweisungen finden Sie in der Dokumentation Ihrer Virtualisierungssoftware, zum Beispiel in der VirtualBox-Dokumentation.
2. Konfigurieren Sie die Netzwerkeinstellungen der virtuellen Maschine so, dass der Zugriff von einem Browser auf dem Host möglich ist. Dies kann durch Aktivieren des *Bridged mode* erreicht werden.
3. Melden Sie sich mit den standardmäßigen System-Anmeldedaten an der virtuellen Maschine an.
4. Um die IP-Adresse abzurufen, führen Sie auf der virtuellen Maschine den folgenden Befehl aus:

```
ip addr show
```

5. Öffnen Sie einen Browser auf dem Host und rufen Sie die IP-Adresse auf, die die Appliance per DHCP erhalten hat.
6. Melden Sie sich mit den standardmäßigen Anmeldedaten bei Zabbix an und beginnen Sie mit dem Monitoring.

Konfiguration Dieser Abschnitt beschreibt häufig benötigte Standardkonfigurationseinstellungen sowie verfügbare Anpassungsoptionen.

Zugangsdaten

System

- Benutzername: `root`
- Passwort: `zabbix`

Zabbix Frontend

- Benutzername: `Admin`
- Passwort: `zabbix`

Nach der Anmeldung können Sie das Standardpasswort in den Einstellungen des Benutzerprofils ändern oder neue Benutzer erstellen und den Standardbenutzer löschen.

Datenbank

Die Passwörter für alle Datenbankbenutzer werden während des Installationsvorgangs zufällig generiert. Für die Datenbank sind die folgenden Benutzer definiert:

Root:

- Benutzername: `root`
- Passwort: Das Passwort wird in der Datei `/root/.my.cnf` gespeichert. Unter dem Root-Konto muss kein Passwort eingegeben werden.

Zabbix Server:

- Benutzername: `zabbix_srv`
- Passwort: Das Passwort wird in der Server-Konfigurationsdatei gespeichert.

Zabbix Frontend:

- Benutzername: `zabbix_web`
- Passwort: Das Passwort wird in der Frontend-Konfigurationsdatei gespeichert.

Um das Passwort eines Datenbankbenutzers zu ändern, ändern Sie es in MySQL und in der entsprechenden Konfigurationsdatei.

Zugriff auf das Frontend

Auf das Zabbix-Frontend kann unter *http://<IP-Adresse der virtuellen Maschine>* zugegriffen werden.

Standardmäßig ist der Zugriff von überall aus erlaubt. Um den Zugriff einzuschränken, ändern Sie `/etc/nginx/conf.d/zabbix.conf`. Starten Sie NGINX nach dem Speichern der bearbeiteten Datei neu, indem Sie sich per SSH als root user anmelden und folgenden Befehl ausführen:

```
systemctl restart nginx
```

Statische IP-Adresse

Standardmäßig verwendet das Appliance DHCP, um die IP-Adresse zu beziehen. So legen Sie eine statische IP-Adresse fest:

1. Melden Sie sich als root user an.
2. Führen Sie die folgenden Befehle aus und ersetzen Sie die Werte durch Ihre eigene IP-Adresse:

```
nmcli con mod eth0 ipv4.addresses 192.168.1.10/24    # Appliance IP-Adresse/CIDR-Präfix
nmcli con mod eth0 ipv4.gateway 192.168.1.1     # Gateway-IP-Adresse
nmcli con mod eth0 ipv4.dns 8.8.8.8    # IP-Adresse des DNS-Servers
nmcli con mod eth0 ipv4.method manual
nmcli con up eth0
```

Alternativ können Sie die Datei `/etc/NetworkManager/system-connections/eth0.nmconnection` öffnen und die folgenden Zeilen hinzufügen:

```
[ipv4]
address1=192.168.1.10/24
dns=8.8.8.8;
gateway=192.168.1.1
```

`method=manual`

Nachdem Sie die bearbeitete Datei gespeichert haben, führen Sie die Befehle aus:

```
nmcli con reload
nmcli con up eth0
```

Um zu überprüfen, ob die Änderungen an der Netzwerkkonfiguration übernommen wurden, führen Sie den folgenden Befehl aus:

```
nmcli dev show eth0
```

Firewall-Konfiguration

Zur Verwaltung der Firewall-Einstellungen verwendet die Appliance iptables mit vordefinierten Regeln:

- SSH-Port (22 TCP) öffnen
- Ports für Zabbix Agent (10050 TCP) und Zabbix trapper (10051 TCP) öffnen
- HTTP- (80 TCP) und HTTPS-Ports (443 TCP) öffnen
- SNMP-Trap-Port (162 UDP) öffnen
- Ausgehende Verbindungen zum NTP-Port (123 UDP) öffnen
- ICMP-Pakete auf 5 Pakete pro Sekunde begrenzen
- Alle anderen eingehenden Verbindungen verwerfen

Um zusätzliche Ports zu öffnen, ändern Sie die Datei `/etc/sysconfig/iptables` und laden Sie die Firewall-Regeln neu:

`systemctl reload iptables`

Repositories

Die Zabbix Appliance verwendet das Paket *zabbix-release* aus dem Zabbix-Repository. Repositories werden im Verzeichnis `/etc/yum.repos.d/*` konfiguriert.

Zeitzone

Standardmäßig verwendet die Appliance UTC für die Systemuhr. Um die Zeitzone zu ändern, kopieren Sie die entsprechende Datei aus `/usr/share/zoneinfo` nach `/etc/localtime`, zum Beispiel:

`cp /usr/share/zoneinfo/Europe/Riga /etc/localtime`

> **Note:**
> Die Zeitzone des Frontends von Zabbix wird separat festgelegt und kann in den Frontend-Einstellungen geändert werden. Die Standardzeitzone für das Zabbix Frontend ist Europe/Riga.

Dateispeicherorte

- Die Konfigurationsdateien des Zabbix Server und des Agent befinden sich in `/etc/zabbix`.
- Die Protokolldateien des Zabbix Server und des Agent befinden sich in `/var/log/zabbix`.
- Die Konfigurationsdatei des Zabbix Frontend befindet sich in `/etc/zabbix/web`.
- Das Zabbix Frontend befindet sich in `/usr/share/zabbix`.
- Das Home-Verzeichnis des Benutzers *zabbix* befindet sich in `/var/lib/zabbix`.

Systemdienste

Systemd-Dienste sind verfügbar. Um die Liste der Zabbix-Dienste anzuzeigen, führen Sie den folgenden Befehl auf der virtuellen Maschine aus:

`systemctl list-units zabbix*`

Formatspezifische Hinweise zu Bildern VMware

Die Images im *vmdk*-Format können direkt in den Produkten VMware Player, Server und Workstation verwendet werden. Für die Verwendung in ESX, ESXi und vSphere müssen sie mit VMware vCenter Converter konvertiert werden (für den Download ist eine Authentifizierung erforderlich). Bei Verwendung von VMware vCenter Converter können Probleme mit dem hybriden Netzwerkadapter auftreten. In diesem Fall können Sie versuchen, während des Konvertierungsvorgangs den E1000-Adapter anzugeben. Alternativ können Sie nach Abschluss der Konvertierung den vorhandenen Adapter löschen und einen E1000-Adapter hinzufügen.

HDD-/Flash-Image (raw)

Um das Image zu booten, führen Sie Folgendes aus:

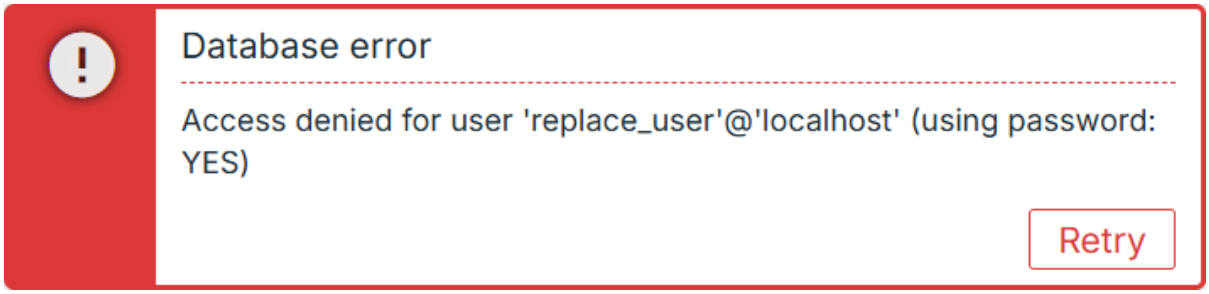
`dd if=./zabbix_appliance_8.0.0.raw of=/dev/sdc bs=4k conv=fdatasync`

Ersetzen Sie `/dev/sdc` durch den Gerätepfad Ihrer Flash-/HDD-Festplatte.

Hyper-V

Wenn die Appliance in Hyper-V nicht startet, versuchen Sie, **Ctrl+Alt+F2** zu drücken, um zu einer TTY-Sitzung zu wechseln.

Fehlerbehebung Wenn beim Versuch, sich im Frontend anzumelden, die Fehlermeldung *Access denied for user 'replace_user'@'localhost' (using password: YES)* angezeigt wird, kann dies darauf hindeuten, dass die Installation noch läuft.



Wenn der Fehler nach einigen Minuten Wartezeit weiterhin auftritt oder Sie ein anderes unerwartetes Verhalten feststellen, wurde der Installationsprozess wahrscheinlich nicht erfolgreich abgeschlossen. In diesem Fall empfehlen wir, das aktuelle Appliance zu löschen und es anhand derselben Installationsanweisungen erneut bereitzustellen. Dieser Schritt behebt das Problem in der Regel.

Beachten Sie, dass ein Versuch, eine fehlerhafte Installation manuell zu reparieren, nicht empfohlen wird, da dies zu weiteren Komplikationen führen kann.

Prozesse

Übersicht Zabbix besteht aus mehreren Prozessen, die zusammenarbeiten, um Systeme zu überwachen und Benachrichtigungen zu senden. Jeder Prozess hat eine spezifische Rolle in der Überwachungsinfrastruktur.

Kernprozesse

- Server - der zentrale Prozess, der das Polling und Trapping von Daten durchführt, Auslöser berechnet und Benachrichtigungen an Benutzer sendet.
- Agent - sammelt Daten von überwachten Hosts.
- Agent 2 - ein Agent der neuen Generation mit Plugin-Architektur und erweiterten Funktionen.
- Proxy - sammelt im Auftrag des Zabbix-Servers Überwachungsdaten von überwachten Geräten und sendet die Informationen in einer verteilten Umgebung an einen zentralen Zabbix-Server.
- Java gateway - ermöglicht die Überwachung von Java-Anwendungen.
- MCP server - ermöglicht es KI-Agenten, mithilfe des Model Context Protocol mit Zabbix zu arbeiten.

Für einen einfachen Zugriff auf Zabbix von überall und von jeder Plattform aus steht eine webbasierte Schnittstelle zur Verfügung. Die Schnittstelle ist Bestandteil des Zabbix-Servers und läuft normalerweise (aber nicht zwingend) auf demselben physischen Rechner wie der Server.

Alle Konfigurationsinformationen sowie die von Zabbix erfassten Daten werden in einer Datenbank gespeichert.

Überschrift Der Header ist in allen Anfrage- und Antwortnachrichten zwischen Zabbix-Komponenten vorhanden.
Er wird benötigt, um die Nachrichtenlänge zu bestimmen, ob die Nachricht komprimiert ist oder nicht und ob es sich um ein großes Paket handelt oder nicht.

Das Zabbix-Kommunikationsprotokoll hat pro Verbindung ein Paketgrößenlimit von 1 GB. Das Limit von 1 GB
gilt sowohl für die Länge der empfangenen Paketdaten als auch für die Länge der unkomprimierten Daten.

Beim Senden der Konfiguration an den Zabbix-Proxy wird das Paketgrößenlimit auf 4 GB erhöht, um
die Synchronisierung großer Konfigurationen zu ermöglichen. Wenn die Datenlänge vor der Komprimierung 4 GB überschreitet, verwendet der Zabbix-Server automatisch
das große Paketformat (0x04-Flag), wodurch das Paketgrößenlimit auf 16 GB erhöht wird.

Beachten Sie, dass das große Paketformat zwar zum Senden beliebiger Daten verwendet werden kann, derzeit jedoch nur der Zabbix-Proxy
configuration syncer Pakete verarbeiten kann, die größer als 1 GB sind.

Headerstruktur Der Header besteht aus vier Feldern. Alle Zahlen im Header sind im Little-Endian-Format dargestellt.

Field	Size	Size (large packet)	Description
<PROTOCOL>	4	4	"ZBXD" oder 5A 42 58 44
<FLAGS>	1	1	Protokoll-Flags: 0x01 - Zabbix-Kommunikationsprotokoll 0x02 - Komprimierung 0x04 - großes Paket
<DATALEN>	4	8	Datenlänge.
<RESERVED>	4	8	Bei Verwendung von Komprimierung (Flag 0x02) - die Länge der unkomprimierten Daten Wenn keine Komprimierung verwendet wird - 00 00 00 00

Header-Beispiele Hier sind einige Codeausschnitte, die zeigen, wie Sie dem zu sendenden Datenpaket den Zabbix-Protokoll-Header hinzufügen, um das Paket zu erhalten, das Sie an Zabbix senden sollten, damit es korrekt interpretiert wird. Diese Codeausschnitte gehen davon aus, dass die Daten nicht größer als 1 GB sind, sodass das Format für große Pakete nicht verwendet wird.

Python

```
packet = b"ZBXD\1" + struct.pack("<II", len(data), 0) + data
```

oder

```
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;
```

oder

```
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;
```

oder

```
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"
```

Server

Übersicht Der Zabbix Server ist der zentrale Prozess der Zabbix-Software.

Der Server führt das Polling und Trapping von Daten durch, berechnet Auslöser und sendet Benachrichtigungen an Benutzer. Er ist die zentrale Komponente, an die Zabbix Agenten und Proxys Daten zur Verfügbarkeit und Integrität von Systemen melden. Der Server kann Netzwerkdienste selbst aus der Ferne prüfen (z. B. Webserver und Mailserver) mithilfe einfacher Dienstprüfungen.

Der Server ist das zentrale Repository, in dem alle Konfigurations-, Statistik- und Betriebsdaten gespeichert werden, und er ist die Instanz in Zabbix, die Administratoren aktiv benachrichtigt, wenn Probleme in einem der überwachten Systeme auftreten.

Die Funktionsweise eines grundlegenden Zabbix Servers ist in drei separate Komponenten unterteilt; diese sind: Zabbix Server, Frontend und Datenbankspeicher.

Alle Konfigurationsinformationen für Zabbix werden in der Datenbank gespeichert, mit der sowohl der Server als auch das Frontend interagieren. Wenn Sie beispielsweise über das Frontend (oder die API) einen neuen Datenpunkt erstellen, wird er der Tabelle `items` in der Datenbank hinzugefügt. Etwa einmal pro Minute fragt der Zabbix Server dann die Tabelle `items` nach einer Liste der aktiven Datenpunkte ab, die anschließend in einem Cache innerhalb des Zabbix Servers gespeichert wird. Deshalb kann es bis zu zwei Minuten dauern, bis Änderungen, die im Zabbix Frontend vorgenommen wurden, im Abschnitt mit den neuesten Daten angezeigt werden.

Siehe auch:

- High availability setup for Zabbix servers
- Configuration parameters: Zabbix server

Laufender Server Wenn als Paket installiert

Der Zabbix Server läuft als Daemon-Prozess. Der Server kann durch Ausführen von Folgendem gestartet werden:

```
systemctl start zabbix-server
```

Dies funktioniert auf den meisten GNU/Linux-Systemen. Auf anderen Systemen müssen Sie möglicherweise Folgendes ausführen:

```
/etc/init.d/zabbix-server start
```

Entsprechend verwenden Sie zum Stoppen/Neustarten/Anzeigen des Status die folgenden Befehle:

```
systemctl stop zabbix-server
systemctl restart zabbix-server
systemctl status zabbix-server
```

Manuell starten

Wenn das oben nicht funktioniert, müssen Sie es manuell starten. Suchen Sie den Pfad zur Binärdatei `zabbix_server` und führen Sie Folgendes aus:

```
zabbix_server
```

Sie können die folgenden Befehlszeilenparameter mit dem Zabbix Server verwenden:

Parameter	Beschreibung
`-c --config <config-file>`	Pfad zur Konfigurationsdatei.
`-f --foreground`	Zabbix Server im Vordergrund ausführen.
`-R --runtime-control <option>`	Administrative Funktionen ausführen.
`-T --test-config`	Konfigurationsdatei validieren und beenden.
`-h --help`	Hilfsinformationen anzeigen und beenden.
`-V --version`	Versionsnummer anzeigen.

Beispiele für das Ausführen von Zabbix Server mit Befehlszeilenparametern:

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

Laufzeitsteuerung

Optionen der Laufzeitsteuerung:

Option	Beschreibung	Ziel
`config_cache_reload`	Konfigurations-Cache neu laden. Wird ignoriert, wenn der Cache gerade geladen wird.	
`history_cache_clear=<target>`	Verlaufs-Cache für den durch seine ID angegebenen Datenpunkt leeren. Wirkt sich auf alle Werte des Datenpunkts aus, außer auf den ersten und letzten Wert.	**target** - ID des Datenpunkts
`diaginfo[=<section>]`	Diagnoseinformationen in der Server-Protokolldatei sammeln.	`historycache` - Statistiken des Verlaufs-Caches; `valuecache` - Statistiken des Werte-Caches; `preprocessing` - Statistiken des Preprocessing-Managers; `alerting` - Statistiken des Alert-Managers; `lld` - Statistiken des LLD-Managers; `locks` - Liste der Mutexes (auf *BSD*-Systemen leer); `connector` - Statistiken für Connectoren mit der längsten Warteschlange.
`ha_status`	Status des Hochverfügbarkeitsclusters (HA) protokollieren.	
`ha_remove_node=<target>`	Den durch seinen Namen oder seine ID angegebenen Hochverfügbarkeitsknoten (HA) entfernen. Beachten Sie, dass aktive/Standby-Knoten nicht entfernt werden können.	**target** - Name oder ID des Knotens (kann durch Ausführen von `ha_status` ermittelt werden).

Option	Beschreibung	Ziel
`ha_set_failover_delay=<delay>`	Failover-Verzögerung der Hochverfügbarkeit (HA) festlegen. Zeitsuffixe werden unterstützt, z. B. 10s, 1m.	
`proxy_config_cache[=<target>]`	Proxy-Konfigurationscache neu laden.	**target** - durch Kommas getrennte Liste von Proxy-Namen. Wenn kein Ziel angegeben ist, wird die Konfiguration für alle Proxys neu geladen.
`secrets_reload`	Secrets aus Vault neu laden.	
`service_cache_reload`	Cache des Servicemanagers neu laden.	
`snmp_cache_reload`	SNMP-Cache neu laden – SNMP-Engine-Eigenschaften (Engine-Zeit, Engine-Boots, Engine-ID, Zugangsdaten) für alle Hosts löschen. Verwenden Sie dies, um bei der Fehlerbehebung von SNMP-Problemen ein globales Leeren des Caches zu erzwingen.	
`housekeeper_execute`	Die Housekeeping-Prozedur starten. Wird ignoriert, wenn die Housekeeping-Prozedur gerade ausgeführt wird.	
`trigger_housekeeper_execute`	Die Housekeeping-Prozedur für Auslöser starten. Wird ignoriert, wenn die Housekeeping-Prozedur für Auslöser gerade ausgeführt wird.	
`log_level_increase[=<target>]`	Protokollierungsstufe erhöhen; wirkt sich auf alle Prozesse aus, wenn kein Ziel angegeben ist. Auf *BSD*-Systemen nicht unterstützt.	**process type** - alle Prozesse des angegebenen Typs (z. B. `poller`). Siehe alle Server-Prozesstypen. **process type,N** - Prozesstyp und Nummer (z. B. `poller,3`). **pid** - Prozesskennung (1 bis 65535). Für größere Werte geben Sie das Ziel als „process type,N“ an.
`log_level_decrease[=<target>]`	Protokollierungsstufe verringern; wirkt sich auf alle Prozesse aus, wenn kein Ziel angegeben ist. Auf *BSD*-Systemen nicht unterstützt.	
`prof_enable[=<target>]`	Profiling aktivieren. Wirkt sich auf alle Prozesse aus, wenn kein Ziel angegeben ist. Aktiviertes Profiling liefert Details zu allen RW-Locks/Mutexes nach Funktionsnamen.	**process type** - alle Prozesse des angegebenen Typs (z. B. `history syncer`) Unterstützte Prozesstypen als Profiling-Ziele: `alerter`, `alert manager`, `availability manager`, `configuration syncer`, `discovery manager`, `escalator`, `history poller`, `history syncer`, `housekeeper`, `http poller`, `icmp pinger`, `ipmi manager`, `ipmi poller`, `java poller`, `lld manager`, `lld worker`, `odbc poller`, `poller`, `preprocessing manager`, `proxy poller`, `self-monitoring`, `service manager`, `snmp trapper`, `task manager`, `timer`, `trapper`, `unreachable poller`, `vmware collector`. **process type,N** - Prozesstyp und Nummer (z. B. `history syncer,1`). **pid** - Prozesskennung (1 bis 65535). Für größere Werte geben Sie das Ziel als „process type,N“ an. **scope** - `rwlock`, `mutex`, `processing` können mit dem Prozesstyp und der Nummer (z. B. `history syncer,1,processing`) oder mit allen Prozessen dieses Typs (z. B. `history syncer,rwlock`) verwendet werden.

Option	Beschreibung	Ziel
`prof_disable[=<target>]`	Profiling deaktivieren. Wirkt sich auf alle Prozesse aus, wenn kein Ziel angegeben ist.	**process type** - alle Prozesse des angegebenen Typs (z. B. `history syncer`). Unterstützte Prozesstypen als Profiling-Ziele: siehe `prof_enable`. **process type,N** - Prozesstyp und Nummer (z. B. `history syncer,1`). **pid** - Prozesskennung (1 bis 65535). Für größere Werte geben Sie das Ziel als „process type,N“ an.

Wenn ein Prozesstyp als Ziel für die Laufzeitsteuerung angegeben wird, gilt der Befehl nur für laufende Prozesse dieses Typs. Wenn derzeit kein Prozess des angegebenen Typs ausgeführt wird, gibt der Befehl einen Fehler zurück:

```
zabbix_server -R log_level_increase='discovery manager'
Cannot redirect signal: "discovery manager" process does not exist
```

Beispiel für die Verwendung der Laufzeitsteuerung zum Neuladen des Konfigurationscaches des Servers:

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

Beispiele für die Verwendung der Laufzeitsteuerung zum Neuladen der Proxy-Konfiguration:

```
### Konfiguration aller Proxy neu laden:
zabbix_server -R proxy_config_cache_reload

### Konfiguration von Proxy1 und Proxy2 neu laden:
zabbix_server -R proxy_config_cache_reload=Proxy1,Proxy2
```

Beispiel für die Verwendung der Laufzeitsteuerung zum Leeren des Verlaufscaches für einen Datenpunkt:

```
zabbix_server -c /usr/local/etc/zabbix_server.conf -R history_cache_clear=42243
```

Beispiele für die Verwendung der Laufzeitsteuerung zum Sammeln von Diagnoseinformationen:

```
### Alle verfügbaren Diagnoseinformationen in der Server-Logdatei sammeln:
zabbix_server -R diaginfo

### Statistiken des Verlaufscaches in der Server-Logdatei sammeln:
zabbix_server -R diaginfo=historycache
```

Beispiel für die Verwendung der Laufzeitsteuerung zum Neuladen des SNMP-Caches:

```
zabbix_server -R snmp_cache_reload
```

Attention:
Wenn eine SNMPv3-Schnittstelle über die Zabbix UI aktualisiert wird, lädt Zabbix in den meisten Fällen die neuen SNMPv3-Anmeldedaten für diese Schnittstelle automatisch neu; verwenden Sie `-R snmp_cache_reload` nur, wenn das Abfragen nach einer Änderung der Anmeldedaten weiterhin fehlschlägt (z. B. aufgrund von Inkonsistenzen bei engineBoots/engineID oder bei Geräten, die nicht RFC-konform sind), oder wenn Sie zum Troubleshooting ein globales Leeren des SNMP-Caches erzwingen müssen.

Beispiel für die Verwendung der Laufzeitsteuerung zum Auslösen der Ausführung des Housekeepers:

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

Beispiele für die Verwendung der Laufzeitsteuerung zum Ändern des Log-Levels:

```
### Log-Level aller Prozesse erhöhen:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase

### Log-Level des zweiten Poller-Prozesses erhöhen:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase=poller,2

### Log-Level des Prozesses mit PID 1234 erhöhen:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase=1234
```

```
### Log-Level aller http poller-Prozesse verringern:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_decrease="http poller"
```

Beispiel zum Setzen der HA-Failover-Verzögerung auf das Minimum von 10 Sekunden:

```
zabbix_server -R ha_set_failover_delay=10s
```

Prozessbenutzer

Der Zabbix Server ist dafür ausgelegt, als Nicht-Root-Benutzer ausgeführt zu werden. Er wird unter dem jeweiligen Nicht-Root-Benutzer ausgeführt, mit dem er gestartet wurde. Sie können den Server daher problemlos unter jedem Nicht-Root-Benutzer ausführen.

Wenn Sie versuchen, ihn als `root` auszuführen, wechselt er zu einem fest kodierten Benutzer `zabbix`, der auf Ihrem System vorhanden sein muss. Sie können den Server nur dann als `root` ausführen, wenn Sie den Parameter `AllowRoot` in der Server-Konfigurationsdatei entsprechend ändern.

Wenn Zabbix Server und Agent auf demselben Rechner ausgeführt werden, wird empfohlen, für den Server einen anderen Benutzer zu verwenden als für den Agent. Andernfalls kann der Agent, wenn beide unter demselben Benutzer ausgeführt werden, auf die Konfigurationsdatei des Servers zugreifen, und jeder Benutzer mit Admin-Rechten in Zabbix kann dann beispielsweise das Datenbankpasswort relativ leicht auslesen.

Konfigurationsdatei

Der Standardpfad zur Server-Konfigurationsdatei hängt vom Betriebssystem und der Installationsmethode von Zabbix ab:

- Bei der Installation aus den Quellen: `/usr/local/etc/zabbix_server.conf` oder der Pfad, der durch die Kompilierungsvariablen `--sysconfdir` oder `--prefix` festgelegt wurde.
- Bei der Installation aus Paketen und bei der Appliance: `/etc/zabbix/zabbix_server.conf`.

Weitere Informationen zum Konfigurieren von `zabbix_server` finden Sie in den Optionen der Konfigurationsdatei.

Startskripte

Die Skripte werden verwendet, um Zabbix-Prozesse während des Systemstarts/-herunterfahrens automatisch zu starten/stoppen. Die Skripte befinden sich im Verzeichnis misc/init.d.

Server-Prozesstypen und Threads

- `agent poller` – asynchroner Poller-Prozess für passive Prüfungen mit einem Worker-Thread;
- `alert manager` – Manager der Alarmwarteschlange;
- `alert syncer` – Prozess zum Schreiben von Alarmen in die Datenbank;
- `alerter` – Prozess zum Senden von Benachrichtigungen;
- `availability manager` – Prozess zum Aktualisieren der Host-Verfügbarkeit;
- `browser poller` – Poller für Browser-Datenpunktprüfungen;
- `configuration syncer` – Prozess zur Verwaltung des In-Memory-Caches der Konfigurationsdaten;
- `configuration syncer worker` – Prozess zum Auflösen und Synchronisieren von Benutzer-Makrowerten in Datenpunktnamen;
- `connector manager` – Manager-Prozess für Konnektoren;
- `connector worker` – Prozess zur Verarbeitung von Anforderungen des Connector-Managers;
- `discovery manager` – Manager-Prozess zur Erkennung von Geräten;
- `discovery worker` – Prozess zur Verarbeitung von Erkennungsaufgaben des Discovery-Managers;
- `escalator` – Prozess zur Eskalation von Aktionen;
- `ha manager` – Prozess zur Verwaltung der Hochverfügbarkeit;
- `history poller` – Prozess zur Verarbeitung berechneter Prüfungen, die eine Datenbankverbindung erfordern;
- `history syncer` – Prozess zum Schreiben von Verlaufsdaten in die Datenbank;
- `housekeeper` – Prozess zum Entfernen veralteter Daten (Datenpunktverlauf und Trends, Benutzersitzungen, Ereignisse usw.) sowie von Daten, die von gelöschten Objekten zurückgelassen wurden;
- `http agent poller` – asynchroner Poller-Prozess für HTTP-Prüfungen mit einem Worker-Thread;
- `http poller` – Poller für das Web-Monitoring;
- `icmp pinger` – Poller für icmpping-Prüfungen;
- `internal poller` – Poller für interne Prüfungen;
- `ipmi manager` – Manager für IPMI-Poller;
- `ipmi poller` – Poller für IPMI-Prüfungen;
- `java poller` – Poller für Java-Prüfungen;
- `lld manager` – Manager-Prozess für Low-Level-Discovery-Aufgaben;
- `lld worker` – Worker-Prozess für Low-Level-Discovery-Aufgaben;
- `odbc poller` – Poller für ODBC-Prüfungen;

- poller – normaler Poller für passive Prüfungen;
- preprocessing manager – Manager für Vorverarbeitungsaufgaben mit Vorverarbeitungs-Worker-Threads;
- preprocessing worker – Thread für die Datenvorverarbeitung;
- proxy poller – Poller für passive Proxys;
- proxy group manager – Manager für Load-Balancing und Hochverfügbarkeit von Proxys;
- report manager – Manager für geplante Aufgaben zur Berichtserstellung;
- report writer – Prozess zur Erstellung geplanter Berichte;
- self-monitoring – Prozess zum Erfassen interner Serverstatistiken;
- service manager – Prozess zur Verwaltung von Services durch den Empfang von Informationen über Probleme, Problem-Tags und Problembehebungen vom History-Syncer, Task-Manager und Alert-Manager;
- snmp poller – asynchroner Poller-Prozess für SNMP-Prüfungen mit einem Worker-Thread (nur walk[OID]- und get[OID]-Datenpunkte);
- snmp trapper – Trapper für SNMP-Traps;
- supervisor – koordiniert den Start und das Herunterfahren von Server-Prozessen und Threads und überwacht deren Bereitschaft;
- task manager – Prozess zur Remote-Ausführung von Aufgaben, die von anderen Komponenten angefordert wurden (z. B. Problem schließen, Problem bestätigen, Datenpunktwert jetzt prüfen, Funktion für Remote-Befehle);
- telemetry query poller – Poller für Prüfungen von Telemetrieabfrage-Datenpunkten;
- timer – Timer zur Verarbeitung von Wartungszeiträumen;
- trapper – Trapper für aktive Prüfungen, Traps und die Proxy-Kommunikation;
- trigger housekeeper – Prozess zum Entfernen von Problemen und Ereignissen, die von inzwischen gelöschten Auslösern generiert wurden;
- unreachable poller – Poller für nicht erreichbare Geräte;
- vmware collector – VMware-Datensammler, der für das Erfassen von Daten aus VMware-Services zuständig ist.

Die Server-Protokolldatei kann verwendet werden, um diese Prozesstypen zu beobachten.

Die Server-Protokolldatei wird mit Lese- und Schreibberechtigungen ausschließlich für den Dateieigentümer erstellt. Zusätzlich ist die Datei für die Eigentümergruppe lesbar. Alle anderen Berechtigungen sind verweigert.

Verschiedene Typen von Zabbix-Server-Prozessen können mithilfe des internen Datenpunkts zabbix[process,<type>,<mode>,<state>] überwacht werden.

Statistik der Transaktionen des History-Syncers

Der Prozessname des History-Syncers zeigt detaillierte Statistiken zu den Transaktionen des History-Syncers an:

```
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
```

Bei "A+B triggers":

- A - Auslöser, die aufgrund von History-Werten verarbeitet wurden;
- B - Auslöser, die aufgrund von Timern verarbeitet wurden.

Die Zeitangaben in processed...in N (<timings>) sec sind:

- Zeit, die zum Schreiben von Datenpunkt-Werten in die Datenbank aufgewendet wurde;
- Zeit, die zum Aktualisieren von Datenpunkt-Daten (Status, Fehler, Host-Inventar usw.) aufgewendet wurde;
- Zeit, die zum Schreiben von Trends in die Datenbank aufgewendet wurde;
- Zeit, die zur Berechnung von Auslösern aufgewendet wurde;
- Zeit, die für die Verarbeitung von Ereignissen und Aktionen aufgewendet wurde.

Housekeeping-Prozedur

Der Prozess housekeeper entfernt in regelmäßigen Abständen veraltete Daten (Datenpunkt-Historie und Trends, Benutzersitzungen, Ereignisse usw.) sowie Daten, die von gelöschten Objekten zurückgelassen wurden. Er läuft in Zyklen, wobei die Häufigkeit und das Löschlimit pro Zyklus durch HousekeepingFrequency und MaxHousekeeperDelete bestimmt werden. Alle Daten, die in einem Zyklus nicht entfernt werden, werden in den nächsten übernommen. Die automatische Housekeeping-Funktion kann unter *Administration* > *Housekeeping* aktiviert und konfiguriert werden.

Wenn der Prozess housekeeper Daten entfernt, die von gelöschten Objekten zurückgelassen wurden, verwendet er Aufgaben aus der Tabelle housekeeper, die immer dann gefüllt wird, wenn ein Objekt gelöscht wird. Wenn Sie beispielsweise einen Host löschen, entfernt Zabbix dessen Datenpunkte, aber nicht deren Historie, Trends oder Probleme. Stattdessen füllen Datenbank-Trigger die Tabelle housekeeper mit Aufgaben, die aus den folgenden Feldern bestehen:

- housekeeperid - Aufgaben-ID
- object - Objekttyp (0 - Datenpunkt; 1 - Auslöser; 2 - Dienst; 3 - erkannter Host; 4 - erkannter Dienst)
- objectid - Objekt-ID (hilft dem housekeeper, objektspezifische Daten zu finden)

Wenn Sie beispielsweise einen Host mit zwei Datenpunkten und einem Auslöser löschen, wird die Tabelle `housekeeper` wie folgt gefüllt:

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

Datenbank-Trigger füllen die Tabelle `housekeeper` mit Aufgaben, ohne objektspezifische Daten zu prüfen; diese Prüfung wird vom Prozess `housekeeper` beim Entfernen der Daten durchgeführt:

- Für gelöschte Datenpunkte (einschließlich LLD-Regeln) - entfernt deren Daten aus allen History- und Trends-Tabellen (`history`, `history_str`, `history_log`, `history_uint`, `history_text`, `history_bin`, `history_json`, `trends`, `trends_uint`). Entfernt außerdem deren interne Ereignisse aus der Tabelle `problem`.
- Für gelöschte Auslöser - entfernt deren Ereignisse aus ereignisbezogenen Tabellen (`problem`, `event_symptom`, `events`; die Datenbank selbst entfernt außerdem Ereignisse aus der Tabelle `event_recovery`). Benachrichtigt außerdem den Prozess `service manager` über die entfernten Ereignisse.

Note:
Ein separater Prozess `trigger housekeeper` übernimmt eine engere Aufgabe - das Entfernen von Problemen und Ereignissen, die keine bekannte Quell-Auslöser haben. Seine Ausführungsfrequenz wird durch `ProblemHousekeepingFrequency` gesteuert.
Bis die Trigger-Housekeeping-Prozedur gestartet wird, können Probleme, die von inzwischen gelöschten Auslösern erstellt wurden, weiterhin Service-Probleme erzeugen und diesen Diensten zuweisen. Wenn Ihre Umgebung viele Statusberechnungsregeln für Dienste auf Basis häufig erkannter/nicht erkannter Auslöser umfasst, sollten Sie die Häufigkeit der Housekeeping-Prozedur erhöhen, indem Sie den Server-Konfigurationsparameter `ProblemHousekeepingFrequency` anpassen.

- Für gelöschte Dienste - entfernt deren Probleme aus der Tabelle `problem`; deren Ereignisse aus der Tabelle `events` werden erst nach dem konfigurierten Datenspeicherzeitraum entfernt.
- Für gelöschte erkannte Hosts/Dienste (Netzwerkerkennung) - entfernt deren Ereignisse aus der Tabelle `events`.

Der Prozess `housekeeper` entfernt nur solche Ereignisse, die nicht mit Problemen verknüpft sind. Ein veraltetes Problem-/Wiederherstellungsereignis wird beispielsweise nicht entfernt, wenn es mit einem offenen Problem verknüpft ist. Wenn `housekeeper` veraltete Entitäten entfernt, werden zuerst Probleme und dann Ereignisse entfernt.

Note:
Tabellen, die den Modus `partition` verwenden (TimescaleDB-partitionierte Tabellen), werden übersprungen; verarbeitet werden nur Tabellen im Modus `regular`.

Unterstützte Plattformen Aufgrund der Sicherheitsanforderungen und der geschäftskritischen Natur des Serverbetriebs ist UNIX das einzige Betriebssystem, das die erforderliche Leistung, Fehlertoleranz und Ausfallsicherheit zuverlässig bereitstellen kann. Zabbix läuft auf marktführenden Versionen.

Der Zabbix Server wird auf den folgenden Plattformen getestet:

- Linux
- Solaris
- AIX
- HP-UX
- Mac OS X
- FreeBSD
- OpenBSD
- NetBSD
- SCO Open Server

Note:
Zabbix kann möglicherweise auch auf anderen Unix-ähnlichen Betriebssystemen funktionieren.

Locale Beachten Sie, dass der Server eine UTF-8-Locale benötigt, damit einige Textobjekte korrekt interpretiert werden können. Die meisten modernen Unix-ähnlichen Systeme verwenden standardmäßig eine UTF-8-Locale; es gibt jedoch einige Systeme, auf denen dies ausdrücklich festgelegt werden muss.

Hochverfügbarkeits-Setup für Zabbix-Server

Übersicht

Hochverfügbarkeit (HA) wird typischerweise in kritischen Infrastrukturen benötigt, die sich praktisch keine Ausfallzeit leisten können. Für jeden Dienst, der ausfallen kann, muss daher eine Failover-Option vorhanden sein, um im Falle eines Ausfalls des aktuellen Dienstes die Aufgabe zu übernehmen.

Zabbix bietet eine **native** Hochverfügbarkeitslösung, die einfach einzurichten ist und keine vorherige HA-Expertise erfordert. Die native Zabbix-HA kann als zusätzliche Schutzebene gegen Software-/Hardwareausfälle des Zabbix Server oder zur Verringerung von Ausfallzeiten aufgrund von Wartungsarbeiten nützlich sein.

Im Hochverfügbarkeitsmodus von Zabbix werden mehrere Zabbix Server als Knoten in einem Cluster betrieben. Während ein Zabbix Server im Cluster aktiv ist, befinden sich die anderen im Standby-Modus und sind bereit, bei Bedarf die Aufgabe zu übernehmen.



Der Wechsel zu Zabbix HA ist unverbindlich. Sie können jederzeit wieder zum Einzelbetrieb zurückkehren.

Siehe auch: Implementierungsdetails

Aktivieren der Hochverfügbarkeit

Starten des Zabbix Servers als Clusterknoten

In der Serverkonfiguration sind zwei Parameter erforderlich, um einen Zabbix Server als Clusterknoten zu starten:

- Der Parameter `HANodeName` muss für jeden Zabbix Server angegeben werden, der als HA-Clusterknoten fungieren soll.

Dies ist eine eindeutige Knotenkennung (z. B. `zabbix-node-01`), unter der der Server in Agent- und Proxy-Konfigurationen referenziert wird. Wenn Sie `HANodeName` nicht angeben, wird der Server im Standalone-Modus gestartet.

- Der Parameter `NodeAddress` muss für jeden Knoten angegeben werden.

Der Parameter `NodeAddress` (`address:port`) wird vom Zabbix Frontend verwendet, um eine Verbindung zum aktiven Serverknoten herzustellen. `NodeAddress` muss mit der IP-Adresse oder dem FQDN-Namen des jeweiligen Zabbix Servers übereinstimmen.

Starten Sie alle Zabbix Server neu, nachdem Sie Änderungen an den Konfigurationsdateien vorgenommen haben. Sie werden nun als Clusterknoten gestartet. Der neue Status der Server kann unter *Reports* > *System information* sowie durch Ausführen des folgenden Befehls angezeigt werden:

```
zabbix_server -R ha_status
```

Dieser Laufzeitbefehl protokolliert den aktuellen Status des HA-Clusters im Zabbix-Serverprotokoll (und in der Standardausgabe):

```
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
```

Vorbereitung des Frontends

Stellen Sie sicher, dass `address:port` des Zabbix Servers nicht in der Frontend-Konfigurationsdatei von Zabbix (`zabbix.conf.php`) definiert ist.

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

Das Zabbix Frontend erkennt den aktiven Knoten automatisch, indem es die Einstellungen aus der Knotentabelle in der Zabbix-Datenbank liest.
Die Knotenadresse des aktiven Knotens wird als Adresse des Zabbix Servers verwendet.

Proxy-Konfiguration

HA-Cluster-Knoten (Server) müssen in der Konfiguration entweder eines passiven oder eines aktiven Zabbix Proxy aufgeführt werden.

Für einen passiven Proxy müssen die Knotennamen im Parameter Server des Proxys aufgeführt und durch ein **Komma** getrennt werden.

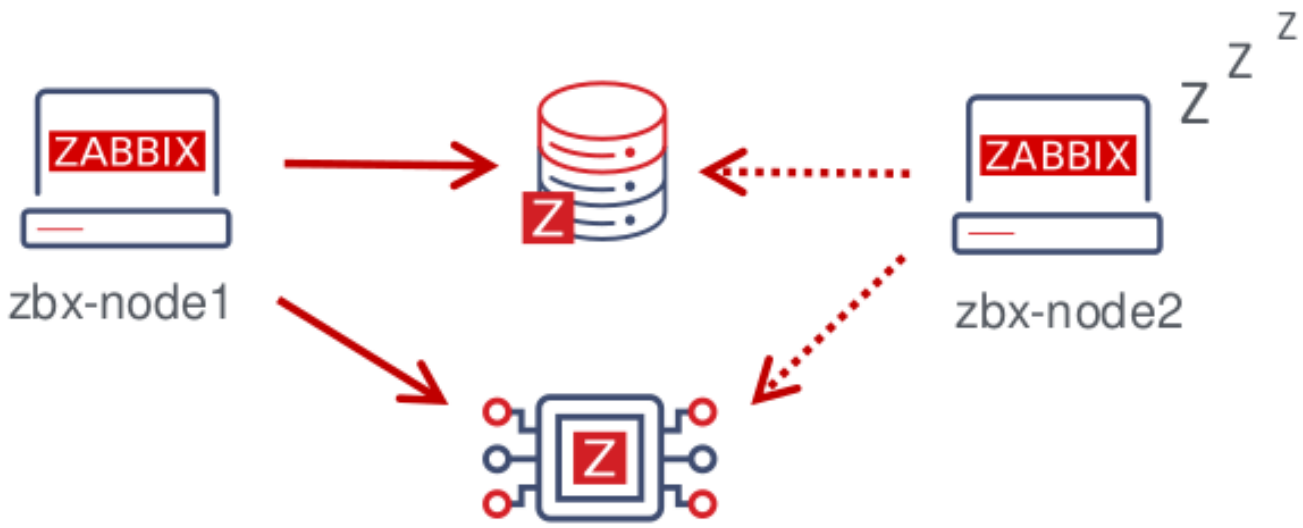
```
Server=zabbix-node-01,zabbix-node-02
```

Für einen aktiven Proxy müssen die Knotennamen im Parameter Server des Proxys aufgeführt und durch ein **Semikolon** getrennt werden.

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

Agent-Konfiguration

HA-Cluster-Knoten (Server) müssen in der Konfiguration von Zabbix Agent oder Zabbix Agent 2 aufgeführt sein.



Um passive Prüfungen zu aktivieren, müssen die Knotennamen im Server-Parameter aufgeführt werden, getrennt durch ein **Komma**.

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

Um aktive Prüfungen zu aktivieren, müssen die Knotennamen im ServerActive-Parameter aufgeführt werden. Beachten Sie, dass bei aktiven Prüfungen die Knoten von anderen Servern durch ein Komma getrennt sein müssen, während die Knoten selbst durch ein **Semikolon** getrennt werden müssen, z. B.:

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

Failover auf Standby-Knoten

Zabbix führt automatisch ein Failover auf einen anderen Knoten durch, wenn der aktive Knoten stoppt. Damit das Failover stattfinden kann, muss mindestens ein Knoten den Status „standby“ haben.

Wie schnell erfolgt das Failover? Alle Knoten aktualisieren alle 5 Sekunden ihre letzte Zugriffszeit (und ihren Status, falls er sich geändert hat). Daher:

- Wenn der aktive Knoten herunterfährt und seinen Status erfolgreich als „stopped“ melden kann, übernimmt ein anderer Knoten innerhalb von **5 Sekunden**.

- Wenn der aktive Knoten herunterfährt bzw. nicht mehr verfügbar ist, ohne seinen Status aktualisieren zu können, warten die Standby-Knoten **Failover-Verzögerung** + 5 Sekunden, bevor sie übernehmen.

Die Failover-Verzögerung ist konfigurierbar; der unterstützte Bereich liegt zwischen 10 Sekunden und 15 Minuten (standardmäßig eine Minute). Um die Failover-Verzögerung zu ändern, können Sie Folgendes ausführen:

```
zabbix_server -R ha_set_failover_delay=5m
```

HA-Cluster verwalten

Der aktuelle Status des HA-Clusters kann mithilfe der dedizierten Optionen zur Laufzeitsteuerung verwaltet werden:

- `ha_status` - den Status des HA-Clusters im Zabbix-Serverprotokoll (und in stdout) protokollieren;
- `ha_remove_node=target` - einen HA-Knoten entfernen, der durch `<target>` identifiziert wird - den Namen oder die ID des Knotens (Name/ID können aus der Ausgabe der Ausführung von `ha_status` abgerufen werden), z. B.:

```
zabbix_server -R ha_remove_node=zabbix-node-02
```

Beachten Sie, dass aktive Knoten/Standby-Knoten nicht entfernt werden können.

- `ha_set_failover_delay=delay` - die HA-Failover-Verzögerung festlegen (zwischen 10 Sekunden und 15 Minuten; Zeitsuffixe werden unterstützt (z. B. `30s`, `1m`)).

Der Knotenstatus kann überwacht werden:

- Unter *Reports* > *System information*.
- Im Dashboard-Widget *System information*.
- Mithilfe der Laufzeitsteuerungsoption `ha_status` des Servers (siehe oben).

Der interne Datenpunkt `zabbix[cluster,discovery,nodes]` kann zur Knotenerkennung verwendet werden, da er ein JSON mit Informationen zu den Hochverfügbarkeitsknoten zurückgibt.

Deaktivieren eines HA-Clusters

Um einen High-Availability-Cluster zu deaktivieren:

1. Erstellen Sie Sicherungskopien der Konfigurationsdateien.
2. Stoppen Sie die Standby-Knoten.
3. Entfernen Sie den Parameter HANodeName vom aktiven primären Server.
4. Starten Sie den primären Server neu (er wird im Standalone-Modus gestartet).

Upgrade des HA-Clusters

Um ein Major-Version-Upgrade für die HA-Knoten durchzuführen:

1. Stoppen Sie alle Knoten.
2. Erstellen Sie ein vollständiges Datenbank-Backup.
3. Wenn die Datenbank Replikation verwendet, stellen Sie sicher, dass alle Knoten synchron sind und keine Probleme aufweisen. Führen Sie kein Upgrade durch, wenn die Replikation fehlerhaft ist.
4. Wählen Sie einen einzelnen Knoten aus, der das Datenbank-Upgrade ausführt, ändern Sie seine Konfiguration in den Standalone-Modus, indem Sie `HANodeName` auskommentieren, und führen Sie das Upgrade darauf aus.
5. Stellen Sie sicher, dass das Datenbank-Upgrade vollständig abgeschlossen ist (*System information* sollte anzeigen, dass der Zabbix-Server ausgeführt wird).
6. Starten Sie den Knoten im HA-Modus neu.
7. Führen Sie das Upgrade für die restlichen Knoten durch und starten Sie sie (es ist nicht erforderlich, sie in den Standalone-Modus zu versetzen, da die Datenbank zu diesem Zeitpunkt bereits aktualisiert ist).

Bei einem Minor-Version-Upgrade reicht es aus, den ersten Knoten zu aktualisieren, sicherzustellen, dass er aktualisiert wurde und läuft, und dann das Upgrade auf dem nächsten Knoten zu starten.

Implementierungsdetails

Der High-Availability-(HA)-Cluster ist eine optionale Lösung und wird für den Zabbix Server unterstützt. Die native HA-Lösung ist auf eine einfache Nutzung ausgelegt, funktioniert standortübergreifend und stellt keine besonderen Anforderungen an die von Zabbix erkannten Datenbanken. Benutzer können entweder die native Zabbix-HA-Lösung oder eine HA-Lösung eines Drittanbieters verwenden, je nachdem, was den Anforderungen an hohe Verfügbarkeit in ihrer Umgebung am besten entspricht.

Die Lösung besteht aus mehreren `zabbix_server`-Instanzen oder Knoten. Jeder Knoten:

- Wird separat konfiguriert.
- Verwendet dieselbe Datenbank.
- Kann mehrere Modi haben: aktiv, Standby, nicht verfügbar, gestoppt.

Zu einem Zeitpunkt kann nur ein Knoten aktiv (arbeitend) sein. Ein Standby-Knoten führt nur einen Prozess aus - den HA-Manager. Ein Standby-Knoten führt keine Datenerfassung, Verarbeitung oder andere reguläre Serveraktivitäten aus; er lauscht nicht auf Ports; er hat nur minimale Datenbankverbindungen.

Sowohl aktive als auch Standby-Knoten aktualisieren ihre letzte Zugriffszeit alle 5 Sekunden. Jeder Standby-Knoten überwacht die letzte Zugriffszeit des aktiven Knotens. Wenn die letzte Zugriffszeit des aktiven Knotens länger als 'failover delay' Sekunden zurückliegt, schaltet sich der Standby-Knoten selbst zum aktiven Knoten um und weist dem zuvor aktiven Knoten den Status 'unavailable' zu.

Der aktive Knoten überwacht seine eigene Datenbankverbindung - wenn sie länger als `failover delay`−5 Sekunden unterbrochen ist, muss er die gesamte Verarbeitung stoppen und in den Standby-Modus wechseln. Der aktive Knoten überwacht außerdem den Status der Standby-Knoten - wenn die letzte Zugriffszeit eines Standby-Knotens länger als 'failover delay' Sekunden zurückliegt, wird dem Standby-Knoten der Status `unavailable` zugewiesen.

Die Knoten sind so ausgelegt, dass sie über kleinere Zabbix-Versionen hinweg kompatibel sind.

Konfigurationsparameter: Zabbix Server

Übersicht

Konfigurationsparameter ermöglichen die Anpassung des Verhaltens des Zabbix Servers.

Geben Sie Parameterwerte in der Server-Konfigurationsdatei von Zabbix (`zabbix_server.conf`) an. Alle Parameter sind **optional**, sofern nicht ausdrücklich angegeben ist, dass der Parameter obligatorisch ist.

Diese Seite ist wie folgt strukturiert:

- Parameterübersicht (um vollständige Details zu einem Parameter anzuzeigen, klicken Sie auf seinen Namen)
- Parameterdetails

Parameter	Beschreibung
AlertScriptsPath	Geben Sie den Speicherort der Alarmskripte an.
AllowRoot	Erlauben Sie dem Server, als `root` ausgeführt zu werden.
AllowSoftwareUpdateCheck	Erlauben Sie den Empfang von Informationen über Softwareaktualisierungen von `zabbix.com`.
AllowUnsupportedDBVersions	Erlauben Sie dem Server, mit nicht unterstützten Datenbankversionen zu arbeiten.
BridgeAdapterURL	Legen Sie die HTTP(S)-URL des Zabbix-Bridge-Adapters fest.
CacheSize	Legen Sie die Größe des Konfigurations-Caches fest (in Byte).
CacheUpdateFrequency	Legen Sie die Aktualisierungsfrequenz des Konfigurations-Caches fest (in Sekunden).
DBHost	Geben Sie den Datenbank-Host (oder das Socket-Verzeichnis) an.
DBName	Geben Sie den Datenbanknamen an. **Obligatorisch**.
DBPassword	Geben Sie das Datenbankpasswort an.
DBPort	Geben Sie den Port des Datenbank-Hosts an.
DBSchema	Geben Sie den Namen des Datenbankschemas an. Wird für PostgreSQL verwendet.
DBSocket	Geben Sie den Pfad zur MySQL-Socketdatei an.
DBUser	Geben Sie den Datenbankbenutzer an.
DBTLSConnect	Geben Sie einen Wert an, um eine TLS-Verbindung zur Datenbank zu erzwingen.
DBTLSCAFile	Geben Sie den vollständigen Pfadnamen der Datei an, die die Zertifikate der übergeordneten Zertifizierungsstelle(n) zur Überprüfung des Datenbankzertifikats enthält.
DBTLSCertFile	Geben Sie den vollständigen Pfadnamen der Datei an, die das Zabbix-Serverzertifikat zur Authentifizierung bei der Datenbank enthält.
DBTLSKeyFile	Geben Sie den vollständigen Pfadnamen der Datei an, die den privaten Schlüssel zur Authentifizierung bei der Datenbank enthält.
DBTLSCipher	Geben Sie die Liste der Verschlüsselungschiffren an, die der Zabbix Server für TLS-Protokolle bis einschließlich TLS v1.2 zulässt. Wird nur für MySQL unterstützt.
DBTLSCipher13	Geben Sie die Liste der Verschlüsselungssammlungen an, die der Zabbix Server für das TLS-v1.3-Protokoll zulässt. Wird nur für MySQL ab Version 8.0.16 unterstützt.
DebugLevel	Legen Sie die Debug-Stufe fest.
EnableGlobalScripts	Aktivieren Sie globale Skripte auf dem Zabbix Server.
EnableMobileDevices	Aktivieren Sie die Unterstützung mobiler Geräte auf dem Zabbix Server.
ExportDir	Geben Sie ein Verzeichnis an, um den Echtzeit-Export von Ereignissen, Verlaufsdaten und Trends zu aktivieren.
ExportFileSize	Legen Sie die maximale Größe der Exportdatei fest (in Byte).
ExportType	Geben Sie Entitätstypen für den Echtzeit-Export an (Ereignisse, Verlaufsdaten, Trends).
ExternalScripts	Geben Sie den Speicherort externer Skripte an.
Fping6Location	Geben Sie den Speicherort von fping6 an.

Parameter	Beschreibung
FpingLocation	Geben Sie den Speicherort von fping an.
FrontendAllowedIP	Geben Sie IP-Adressen oder CIDR-Bereiche an, von denen Verbindungen vom Frontend zulässig sind.
HANodeName	Geben Sie den Knotennamen in einem Hochverfügbarkeitscluster an.
HistoryCacheSize	Legen Sie die Größe des Verlaufs-Caches fest.
HistoryIndexCacheSize	Legen Sie die Größe des Verlaufsindex-Caches fest.
HistoryProvider	Legen Sie den Speicheranbieter für Verlaufsdaten fest (Elasticsearch, ClickHouse oder SQL).
HistoryStorageDateIndex	Dieser Parameter ist **veraltet** (verwenden Sie stattdessen HistoryProvider). Aktivieren Sie das Speichern von Verlaufswerten in verschiedenen Indizes auf Grundlage des Datums.
HistoryStorageURL	Dieser Parameter ist **veraltet** (verwenden Sie stattdessen HistoryProvider). Geben Sie die URL des Verlaufsdatenspeichers an.
HistoryStorageTypes	Dieser Parameter ist **veraltet** (verwenden Sie stattdessen HistoryProvider). Geben Sie die Werttypen an, die an den Verlaufsdatenspeicher gesendet werden sollen.
HousekeepingFrequency	Legen Sie die Frequenz des Bereinigungsvorgangs fest (in Stunden).
Include	Geben Sie ein Verzeichnis oder einzelne Dateien an, die in die Konfigurationsdatei aufgenommen werden sollen.
JavaGateway	Geben Sie die IP-Adresse (oder den Hostnamen) des Zabbix Java gateway an.
JavaGatewayPort	Geben Sie den Port des Zabbix Java gateway an.
ListenBacklog	Legen Sie die maximale Anzahl ausstehender Verbindungen in der TCP-Warteschlange fest.
ListenIP	Geben Sie die IP-Adresse an, an der der Trapper lauscht.
ListenPort	Geben Sie den Port an, an dem der Trapper lauscht.
LoadModule	Geben Sie das Modul an, das beim Start des Servers geladen werden soll.
LoadModulePath	Geben Sie den vollständigen Pfad zum Speicherort der Servermodule an.
LogFile	Geben Sie die Server-Logdatei an.
LogFileSize	Legen Sie die maximale Größe der Logdatei fest.
LogSlowQueries	Legen Sie die maximale Dauer fest, die eine Datenbankabfrage dauern darf, bevor sie protokolliert wird (in Millisekunden).
LogType	Geben Sie den Typ der Logausgabe an.
MaxConcurrentChecksPerPoller	Legen Sie die maximale Anzahl asynchroner Prüfungen fest, die gleichzeitig von jedem HTTP-Agent-Poller, Agent-Poller oder SNMP-Poller ausgeführt werden können.
MaxHousekeeperDelete	Legen Sie die maximale Anzahl der Zeilen fest, die pro Aufgabe in einem Bereinigungszyklus gelöscht werden dürfen.
NodeAddress	Geben Sie eine IP-Adresse oder einen Hostnamen mit optionalem Port an, um die Verbindung des Frontends zum Server zu überschreiben.
PidFile	Geben Sie die PID-Datei an.
ProblemHousekeepingFrequency	Legen Sie die Frequenz zum Löschen von Problemen für gelöschte Auslöser fest.
ProxyConfigFrequency	Legen Sie die Frequenz fest, mit der der Server Konfigurationsdaten an den Zabbix Proxy sendet.
ProxyDataFrequency	Legen Sie die Frequenz fest, mit der der Server Verlaufsdaten vom Zabbix Proxy anfordert.
ServiceManagerSyncFrequency	Legen Sie die Frequenz für die Synchronisierung der Konfiguration eines Servicemanagers fest.
SMSDevices	Geben Sie Modemdateien an, die den Zabbix Server verwenden dürfen.
SNMPTrapperFile	Geben Sie die temporäre Datei an, die zur Übergabe von Daten vom SNMP-Trap-Daemon an den Server verwendet wird.
SocketDir	Geben Sie das Verzeichnis zum Speichern der von internen Zabbix-Diensten verwendeten IPC-Sockets an.
SourceIP	Geben Sie die Quell-IP-Adresse an.
SSHKeyLocation	Geben Sie den Speicherort öffentlicher und privater Schlüssel für SSH-Prüfungen und -Aktionen an.
SSLCertLocation	Geben Sie den Speicherort der SSL-Clientzertifikatdateien für die Clientauthentifizierung an.
SSLKeyLocation	Geben Sie den Speicherort der privaten SSL-Schlüsseldateien für die Clientauthentifizierung an.
SSLCALocation	Geben Sie den Speicherort der Zertifizierungsstellen-(CA-)Dateien für die Überprüfung von SSL-Serverzertifikaten an.
StartAgentPollers	Legen Sie die Anzahl vorab gestarteter Instanzen asynchroner Zabbix-Agent-Poller fest.
StartAlerters	Legen Sie die Anzahl vorab gestarteter Instanzen von Alarmierern fest.
StartBrowserPollers	Legen Sie die Anzahl vorab gestarteter Instanzen von Browser-Datenpunkt-Pollern fest.
StartConnectors	Legen Sie die Anzahl vorab gestarteter Instanzen von Connector-Workern fest.
StartDBSyncers	Legen Sie die Anzahl vorab gestarteter Instanzen von Verlaufsdaten-Synchronisierern fest.
StartDiscoverers	Legen Sie die Anzahl vorab gestarteter Instanzen von Discovery-Workern fest.
StartEscalators	Legen Sie die Anzahl vorab gestarteter Instanzen von Eskalierern fest.
StartHistoryPollers	Legen Sie die Anzahl vorab gestarteter Instanzen von Verlaufsdaten-Pollern fest.
StartHTTPAgentPollers	Legen Sie die Anzahl vorab gestarteter Instanzen asynchroner HTTP-Agent-Poller fest.

Parameter	Beschreibung
StartHTTPPollers	Legen Sie die Anzahl vorab gestarteter Instanzen von HTTP-Pollern fest.
StartIPMIPollers	Legen Sie die Anzahl vorab gestarteter Instanzen von IPMI-Pollern fest.
StartJavaPollers	Legen Sie die Anzahl vorab gestarteter Instanzen von Java-Pollern fest.
StartLLDProcessors	Legen Sie die Anzahl vorab gestarteter Instanzen von Low-Level-Discovery-(LLD-)Workern fest.
StartODBCPollers	Legen Sie die Anzahl vorab gestarteter Instanzen von ODBC-Pollern fest.
StartPingers	Legen Sie die Anzahl vorab gestarteter Instanzen von ICMP-Pingern fest.
StartPollersUnreachable	Legen Sie die Anzahl vorab gestarteter Instanzen von Pollern für nicht erreichbare Hosts fest (einschließlich IPMI und Java).
StartPollers	Legen Sie die Anzahl vorab gestarteter Instanzen von Pollern fest.
StartPreprocessors	Legen Sie die Anzahl vorab gestarteter Instanzen von Vorverarbeitungs-Workern fest.
StartProxyPollers	Legen Sie die Anzahl vorab gestarteter Instanzen von Pollern für passive Proxys fest.
StartReportWriters	Legen Sie die Anzahl vorab gestarteter Instanzen von Berichtsschreibern fest.
StartSNMPPollers	Legen Sie die Anzahl vorab gestarteter Instanzen asynchroner SNMP-Poller fest.
StartSNMPTrapper	Setzen Sie den Wert auf „1“, um einen SNMP-Trapper-Prozess zu starten.
StartTelemetryQueryPollers	Legen Sie die Anzahl vorab gestarteter Instanzen asynchroner Telemetrieabfrage-Poller fest.
StartTimers	Legen Sie die Anzahl vorab gestarteter Instanzen von Timern fest.
StartTrappers	Legen Sie die Anzahl vorab gestarteter Instanzen von Trappern fest.
StartVMwareCollectors	Legen Sie die Anzahl vorab gestarteter VMware-Collector-Instanzen fest.
StatsAllowedIP	Geben Sie IP-Adressen oder DNS-Namen externer Zabbix-Instanzen an, die Statistiken anfordern dürfen.
TelemetryProvider	Legen Sie die Quelle der Telemetriedaten fest.
Timeout	Geben Sie an, wie lange (in Sekunden) auf den Verbindungsaufbau und den Datenaustausch mit dem Zabbix Proxy, Agent, Webservice und bei SNMP-Prüfungen gewartet werden soll (ausgenommen SNMP-`walk[OID]`- und `get[OID]`-Datenpunkte).
TLSCAFile	Geben Sie den vollständigen Pfadnamen der Datei an, die die Zertifikate der übergeordneten Zertifizierungsstelle(n) zur Überprüfung von Peer-Zertifikaten enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSCertFile	Geben Sie den vollständigen Pfadnamen der Datei an, die das Serverzertifikat oder die Zertifikatskette enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSCipherAll	Geben Sie die GnuTLS-Prioritätszeichenfolge oder die OpenSSL-(TLS-1.2-)Chiffrenzeichenfolge an. Überschreiben Sie damit die Standardkriterien für die Auswahl der Verschlüsselungssammlung bei zertifikats- und PSK-basierter Verschlüsselung.
TLSCipherAll13	Geben Sie die Chiffrenzeichenfolge für OpenSSL 1.1.1 oder neuer unter TLS 1.3 an. Überschreiben Sie damit die Standardkriterien für die Auswahl der Verschlüsselungssammlung bei zertifikats- und PSK-basierter Verschlüsselung.
TLSCipherCert	Geben Sie die GnuTLS-Prioritätszeichenfolge oder die OpenSSL-(TLS-1.2-)Chiffrenzeichenfolge an. Überschreiben Sie damit die Standardkriterien für die Auswahl der Verschlüsselungssammlung bei zertifikatsbasierter Verschlüsselung.
TLSCipherCert13	Geben Sie die Chiffrenzeichenfolge für OpenSSL 1.1.1 oder neuer unter TLS 1.3 an. Überschreiben Sie damit die Standardkriterien für die Auswahl der Verschlüsselungssammlung bei zertifikatsbasierter Verschlüsselung.
TLSCipherPSK	Geben Sie die GnuTLS-Prioritätszeichenfolge oder die OpenSSL-(TLS-1.2-)Chiffrenzeichenfolge an. Überschreiben Sie damit die Standardkriterien für die Auswahl der Verschlüsselungssammlung bei PSK-basierter Verschlüsselung.
TLSCipherPSK13	Geben Sie die Chiffrenzeichenfolge für OpenSSL 1.1.1 oder neuer unter TLS 1.3 an. Überschreiben Sie damit die Standardkriterien für die Auswahl der Verschlüsselungssammlung bei PSK-basierter Verschlüsselung.
TLSCRLFile	Geben Sie den vollständigen Pfadnamen der Datei an, die widerrufene Zertifikate enthält. Dieser Parameter wird für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet.
TLSFrontendAccept	Geben Sie die Verschlüsselungsstufe für eingehende Verbindungen vom Frontend an.
TLSFrontendCertIssuer	Geben Sie den zulässigen Aussteller des Frontend-Zertifikats an.
TLSFrontendCertSubject	Geben Sie den zulässigen Betreff des Frontend-Zertifikats an.
TLSKeyFile	Geben Sie den vollständigen Pfadnamen der Datei an, die den privaten Serverschlüssel enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSListen	Erfordern Sie ausschließlich TLS-Verbindungen am Trapper-Socket.
TmpDir	Geben Sie das temporäre Verzeichnis an.
TrapperTimeout	Geben Sie das Trapper-Timeout in Sekunden an für: - das Abrufen von Verlaufsdaten vom Zabbix Proxy; - das Senden von Konfigurationsdaten an den Zabbix Proxy; - die Ausführung globaler Skripte oder entfernter Befehle auf dem Zabbix Server.

Parameter	Beschreibung
TrendCacheSize	Legen Sie die Größe des Trend-Caches fest.
TrendFunctionCacheSize	Legen Sie die Größe des Trendfunktions-Caches fest.
UnavailableDelay	Legen Sie die Frequenz fest, mit der die Verfügbarkeit eines Hosts während des Zeitraums der Nichtverfügbarkeit geprüft wird.
UnreachableDelay	Legen Sie die Frequenz fest, mit der die Verfügbarkeit eines Hosts während des Zeitraums der Nichterreichbarkeit geprüft wird.
UnreachablePeriod	Legen Sie fest, nach wie vielen Sekunden der Nichterreichbarkeit ein Host als nicht verfügbar behandelt wird.
User	Geben Sie die Berechtigungen auf einen bestimmten, vorhandenen Benutzer im System ab.
ValueCacheSize	Legen Sie die Größe des Verlaufswert-Caches fest.
Vault	Geben Sie den Vault-Anbieter an.
VaultAppRoleID	Geben Sie die AppRole-ID des HashiCorp Vault an.
VaultAppSecretID	Geben Sie die AppRole-Secret-ID des HashiCorp Vault an.
VaultDBPath	Geben Sie den Speicherort an, aus dem Datenbankanmeldedaten anhand von Schlüsseln abgerufen werden.
VaultPrefix	Geben Sie ein benutzerdefiniertes Präfix für den Vault-Pfad oder die Abfrage an.
VaultTLSCertFile	Geben Sie den Namen der SSL-Zertifikatdatei an, die für die Clientauthentifizierung verwendet wird.
VaultTLSKeyFile	Geben Sie den Namen der privaten SSL-Schlüsseldatei an, die für die Clientauthentifizierung verwendet wird.
VaultToken	Geben Sie das Authentifizierungstoken des HashiCorp Vault an.
VaultURL	Geben Sie die URL des Vault Servers an.
VMwareCacheSize	Legen Sie die Größe des gemeinsam genutzten Speichers zum Speichern von VMware-Daten fest.
VMwareFrequency	Legen Sie die Frequenz der Datenerfassung von einem einzelnen VMware-Dienst fest (in Sekunden).
VMwarePerfFrequency	Legen Sie die Frequenz des Abrufs von Leistungszählerstatistiken von einem einzelnen VMware-Dienst fest (in Sekunden).
VMwareTimeout	Legen Sie die maximale Anzahl von Sekunden fest, die ein `vmware collector` auf eine Antwort vom VMware-Dienst wartet.
WebDriverURL	Geben Sie die URL der WebDriver-Schnittstelle an.
WebServiceURL	Geben Sie die URL des Webservice im Format `http[s]://host:port/report` an.

Beachten Sie:

- Die Standardwerte entsprechen den Daemon-Standards und nicht den Werten in den mitgelieferten Konfigurationsdateien.
- Werte unterstützen Umgebungsvariablen.
- Zabbix unterstützt Konfigurationsdateien nur in UTF-8-Kodierung ohne BOM.
- Kommentare, die mit # beginnen, werden nur am Anfang einer Zeile unterstützt.

Warning:
Zu viele Prozesse können das Limit der Dateideskriptoren pro Prozess für den Vorverarbeitungsmanager ausschöpfen.

Zu solchen Prozessen gehören die Datenerfassung durch Poller, Poller für nicht erreichbare Hosts, ODBC-Poller, HTTP-Poller, Java-Poller, Pinger, Trapper und Proxy-Poller sowie der IPMI-Manager, SNMP-Trapper, Vorverarbeitungs-Worker und Discovery-Worker. Das Ausschöpfen des Dateideskriptorlimits führt dazu, dass der Zabbix Server **angehalten** wird, typischerweise kurz nach dem Start, manchmal jedoch erst später.

Um solche Probleme zu vermeiden, überprüfen Sie die Zabbix-Server-Konfigurationsdatei, um die Anzahl gleichzeitiger Prüfungen und Prozesse zu optimieren. Stellen Sie außerdem bei Bedarf sicher, dass das Dateideskriptorlimit ausreichend hoch festgelegt ist, indem Sie die Systemlimits überprüfen und anpassen.

Parameterdetails

AlertScriptsPath

Geben Sie den Speicherort des alert script an. Abhängig von der Installationsvariable `datadir` zur Kompilierzeit.

Standard: `/usr/local/share/zabbix/alertscripts`

AllowRoot

Erlaubt dem Server, als `root` zu laufen. Wenn diese Option deaktiviert ist und der Server von `root` gestartet wird, versucht der Server stattdessen zum Benutzer `zabbix` zu wechseln. Hat keine Wirkung, wenn der Start unter einem normalen Benutzer erfolgt.

Standard: 0
 Werte: 0 - nicht erlauben; 1 - erlauben

AllowSoftwareUpdateCheck

Erlaubt der Zabbix-UI, Informationen über Software-Updates von `zabbix.com` zu empfangen.

Standard: 1
 Werte: 0 - nicht erlauben; 1 - erlauben

AllowUnsupportedDBVersions

Erlaubt dem Server, mit nicht unterstützten Datenbankversionen zu arbeiten.

Standard: 0
 Werte: 0 - nicht erlauben; 1 - erlauben

BridgeAdapterURL

Legen Sie die HTTP(S)-URL des Zabbix-Bridge-Adapters fest.

Beispiel:

```
BridgeAdapterURL=https://bridge-adapter.example.com:10054/rpc
```

CacheSize

Legt die Größe des Konfigurations-Caches fest (in Bytes). Die Größe des gemeinsam genutzten Speichers zum Speichern von Host-, Datenpunkt- und Auslöserdaten.

Standard: 32M
 Bereich: 128K-64G

CacheUpdateFrequency

Legt die Aktualisierungshäufigkeit des Konfigurationscaches fest (in Sekunden). Siehe auch die Optionen zur Laufzeitsteuerung.

Standard: 10
 Bereich: 1-3600

DBHost

Geben Sie den Datenbank-Host (oder das Socket-Verzeichnis) an.

Für MySQL:

- `localhost` oder eine leere Zeichenfolge verwendet den standardmäßigen UNIX-Domain-Socket.

Für PostgreSQL:

- `localhost` wird über DNS aufgelöst (typischerweise zu 127.0.0.1).
- Eine leere Zeichenfolge verwendet den standardmäßigen UNIX-Domain-Socket.
- Ein Pfad (zum Beispiel `/var/run/pgbouncer`) führt dazu, dass der UNIX-Socket an diesem Pfad verwendet wird.
- Eine durch Kommas getrennte Liste kann mehrere `host:port`-Werte enthalten. Jeder Host wird der Reihe nach versucht, bis eine Lese-/Schreibverbindung hergestellt ist.

Beispiel: `DBHost=localhost:5431,127.0.0.1:20051,zabbix.domain,[::1]:30051,[12fc::1]`

Standard: `localhost`

DBName

Geben Sie den Datenbanknamen an.

Erforderlich: **Ja**

DBPassword

Geben Sie das Datenbankpasswort an. Kommentieren Sie diese Zeile aus, wenn kein Passwort verwendet wird.

DBPort

Geben Sie den Port des Zabbix-Datenbank-Hosts an (siehe `DBHost`).
 `DBSocket` und `DBPort` schließen sich in der Serverkonfiguration gegenseitig aus. Geben Sie nur einen von beiden an oder lassen Sie beide undefiniert.

Standard für MySQL: 3306

Standard für PostgreSQL: 5432

Bereich: 1024-65535

DBSchema

Geben Sie den Namen des Datenbankschemas an. Wird für PostgreSQL verwendet.

DBSocket

Geben Sie den Pfad zur MySQL-Socket-Datei an.
 DBSocket und DBPort schließen sich in der Serverkonfiguration gegenseitig aus. Geben Sie nur einen von beiden an oder lassen Sie beide undefiniert.

DBUser

Geben Sie den Datenbankbenutzer an.

DBTLSConnect

Geben Sie einen Wert an, um eine TLS-Verbindung zur Datenbank zu erzwingen: - *required* - Verbindung über TLS herstellen; - *verify_ca* - Verbindung über TLS herstellen und das Zertifikat überprüfen; - *verify_full* - Verbindung über TLS herstellen, das Zertifikat überprüfen und verifizieren, dass die durch DBHost angegebene Datenbankidentität mit ihrem Zertifikat übereinstimmt.

Bei MySQL werden ab Version 5.7.11 sowie bei PostgreSQL die folgenden Werte unterstützt: required, verify_ca, verify_full.
 Bei MariaDB werden ab Version 10.2.6 die Werte required und verify_full unterstützt.
Standardmäßig ist keine Option gesetzt, und das Verhalten hängt von der Datenbankkonfiguration ab.

DBTLSCAFile

Geben Sie den vollständigen Pfadnamen der Datei an, die die Zertifikate der obersten CA(s) für die Datenbankzertifikatsprüfung enthält.

Pflicht: nein (ja, wenn DBTLSConnect auf verify_ca oder verify_full gesetzt ist)

DBTLSCertFile

Geben Sie den vollständigen Pfadnamen der Datei an, die das Zabbix-Serverzertifikat für die Authentifizierung an der Datenbank enthält.

DBTLSKeyFile

Geben Sie den vollständigen Pfadnamen der Datei an, die den privaten Schlüssel für die Authentifizierung an der Datenbank enthält.

DBTLSCipher

Geben Sie die Liste der Verschlüsselungs-Cipher an, die der Zabbix Server für TLS-Protokolle bis einschließlich TLS v1.2 zulässt. Nur für MySQL unterstützt.

DBTLSCipher13

Geben Sie die Liste der Verschlüsselungs-Cipher Suites an, die der Zabbix Server für das TLS-v1.3-Protokoll zulässt. Nur für MySQL unterstützt, beginnend mit Version 8.0.16.

DebugLevel

Legt den Debug-Level fest: - *0* - grundlegende Informationen über das Starten und Stoppen von Zabbix-Prozessen; - *1* - kritische Informationen; - *2* - Fehlerinformationen; - *3* - Warnungen; - *4* - für das Debugging (liefert viele Informationen); - *5* - erweitertes Debugging (liefert noch mehr Informationen).
 Siehe auch die Optionen zur Laufzeitsteuerung.

Standard: 3
 Bereich: 0-5

EnableGlobalScripts

Globale Skripte auf dem Zabbix-Server aktivieren.
 Hinweis: Die Ausführung globaler Skripte ist standardmäßig aktiviert. Bei Neuinstallationen ab Zabbix 7.0 wird EnableGlobalScripts jedoch explizit auf 0 (deaktiviert) gesetzt.

Standard: 1
 Werte: 0 - deaktivieren; 1 - aktivieren

EnableMobileDevices

Aktiviert die Unterstützung für mobile Geräte auf dem Zabbix Server.

Wenn diese Option aktiviert ist, verarbeitet der Zabbix Server Anfragen zum Hinzufügen oder Entfernen mobiler Geräte und sendet Push-Benachrichtigungen an diese Geräte.

Standard: 1
 Werte: 0 - deaktivieren; 1 - aktivieren

ExportDir

Geben Sie ein Verzeichnis an, um den Echtzeit-Export von Ereignissen, Verlauf und Trends zu aktivieren.

ExportFileSize

Legt die maximale Größe der Exportdatei fest (in Bytes). Wird für die Rotation verwendet, wenn ExportDir gesetzt ist.

Standard: 1G
 Bereich: 1M-1G

ExportType

Geben Sie Entitätstypen für den Echtzeit-Export an (Ereignisse, Verlauf, Trends). Eine durch Kommas getrennte Liste. Standardmäßig werden alle Typen exportiert. Nur gültig, wenn `ExportDir` gesetzt ist.
 Hinweis: Wenn `ExportType` angegeben ist, `ExportDir` jedoch nicht, handelt es sich um einen Konfigurationsfehler und der Server wird nicht gestartet.

Beispiel für den Export von Verlauf und Trends:

```
ExportType=history,trends
```

Beispiel nur für den Export von Ereignissen:

```
ExportType=events
```

ExternalScripts

Geben Sie den Speicherort des externen Skripts an. Abhängig von der Compile-Time-Installationsvariable `datadir`.

Standard: `/usr/local/share/zabbix/externalscripts`

Fping6Location

Geben Sie den Speicherort von fping6 an. Stellen Sie sicher, dass das fping6-Binary dem Benutzer root gehört und das SUID-Flag gesetzt ist. Lassen Sie das Feld leer (`Fping6Location=`), wenn Ihr fping-Dienstprogramm IPv6-Adressen verarbeiten kann.

Standard: `/usr/sbin/fping6`

FpingLocation

Geben Sie den Speicherort von fping an. Stellen Sie sicher, dass das fping-Binary dem Benutzer root gehört und das SUID-Flag gesetzt ist.

Standard: `/usr/sbin/fping`

FrontendAllowedIP

Geben Sie IP-Adressen oder CIDR-Bereiche an, die Verbindungen vom Frontend aus zulassen. Eine durch Kommas getrennte Liste. Standardmäßig werden für Frontend-Anfragen alle Verbindungen akzeptiert. Wenn die IPv6-Unterstützung aktiviert ist, werden `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` gleich behandelt und `::/0` erlaubt jede IPv4- oder IPv6-Adresse. `0.0.0.0/0` kann verwendet werden, um jede IPv4-Adresse zuzulassen.

Beispiel:

```
FrontendAllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

HANodeName

Geben Sie den Knotennamen in einem Hochverfügbarkeitscluster an. Wenn das Feld leer ist, arbeitet der Server im Standalone-Modus, und es wird ein Knoten mit leerem Namen erstellt.

HistoryCacheSize

Legt die Größe des History-Cache fest (in Bytes). Die Größe des gemeinsam genutzten Speichers zum Speichern von Verlaufsdaten.

Standard: `16M`
 Bereich: 128K-16G

HistoryIndexCacheSize

Legt die Größe des History-Index-Caches fest (in Bytes). Die Größe des Shared Memory für die Indizierung der im History-Cache gespeicherten History-Daten. Die Größe des Index-Caches benötigt ungefähr 100 Bytes, um einen Datenpunkt zwischenzuspeichern.

Standard: `4M`
 Bereich: 128K-16G

HistoryProvider

Legen Sie den History-Speicheranbieter mit dem Format `<provider>[;<options>]` fest, wobei:

- `<provider>` - einer der unterstützten History-Speicheranbieter: `sql` (Standard), `elasticsearch` oder `clickhouse`.
- `<options>` - durch Kommas getrennte `key=value`-Paare. Werte können in Anführungszeichen gesetzt werden.

Optionen für alle History-Speicheranbieter:

- `value_types` - legt die zu speichernden Wertetypen fest (`uint`, `dbl`, `str`, `log`, `text`, `json`). Jeder kann nur einmal und nur für einen einzelnen Anbieter verwendet werden.
- `precache` - setzt `1` (Standard), um Daten aus dem History-Speicheranbieter gesammelt in den Value-Cache vorzuladen. Dies kann das Aufwärmen des Value-Cache nach einem Neustart erheblich verbessern.

Optionen für Elasticsearch und ClickHouse:

- `url` - legt die HTTP- oder HTTPS-URL fest, die für die Verbindung mit dem History-Speicheranbieter verwendet wird.
- `log_slow_queries` - legt die maximale Dauer (in Millisekunden) fest, die eine Abfrage an den History-Speicheranbieter dauern darf, bevor sie protokolliert wird. Diese Option wird ab `DebugLevel=3` aktiviert.
- `source_ip` - legt die Quell-IP-Adresse fest, die für ausgehende Verbindungen zum History-Speicheranbieter verwendet wird.

Optionen für Elasticsearch:

- `date_index` - setzen Sie 1, um das Speichern von History-Werten in unterschiedlichen Indizes basierend auf dem Datum zu aktivieren.

Optionen für ClickHouse:

- `db` - legt den Namen der Datenbank fest.
- `username` - legt den Benutzernamen fest, der für die Anmeldung an der Datenbank verwendet wird.
- `password` - legt das Passwort fest, das für die Anmeldung an der Datenbank verwendet wird.
- `ssl_cert_file`, `ssl_key_file`, `ssl_key_password`, `ssl_verify_peer`, `ssl_verify_host`, `ssl_ca_location`, `ssl_cert_location`, `ssl_key_location` - setzen Sie diese Optionen, um eine sichere (TLS/SSL) Verbindung zur Datenbank zu konfigurieren.

Beispiel für ClickHouse:

```
HistoryProvider=clickhouse;value_types="uint,dbl,str,log,text,json",url=http://localhost:8123,db=zabbix,us
```

Beispiel für Elasticsearch:

```
HistoryProvider=elasticsearch;value_types="uint,dbl,str,log,text,json",url=http://localhost:9200,date_inde
```

Beispiel für ClickHouse und Elasticsearch:

```
HistoryProvider=clickhouse;value_types="uint,dbl,str",url=http://localhost:8123,db=zabbix,username=zabbix,
HistoryProvider=elasticsearch;value_types="log,text,json",url=http://localhost:9200,date_index=0
```

HistoryStorageDateIndex

Dieser Parameter ist veraltet (verwenden Sie stattdessen HistoryProvider).
 Aktiviert die Speicherung von Verlaufswerten in unterschiedlichen Indizes basierend auf dem Datum. Dieser Parameter wird für die Elasticsearch-Einrichtung verwendet.

Standard: 0
 Werte: 0 - deaktiviert; 1 - aktiviert

HistoryStorageURL

Dieser Parameter ist veraltet (verwenden Sie stattdessen HistoryProvider).
 Geben Sie die URL des History-Speichers an. Dieser Parameter wird für das Elasticsearch-Setup verwendet.

HistoryStorageTypes

Dieser Parameter ist veraltet (verwenden Sie stattdessen HistoryProvider).
 Geben Sie die Werttypen an, die an den History-Speicher gesendet werden sollen. Eine durch Kommas getrennte Liste. Dieser Parameter wird für das Elasticsearch-Setup verwendet.

Standard: `uint,dbl,str,log,text,json`

HousekeepingFrequency

Legen Sie die Häufigkeit des Housekeeping-Vorgangs fest (in Stunden). Housekeeping entfernt veraltete Informationen aus der Datenbank.
Hinweis: Um zu verhindern, dass der Housekeeper überlastet wird (zum Beispiel, wenn die Aufbewahrungszeiträume für History und Trends stark reduziert werden), werden pro Housekeeping-Zyklus und für jeden Datenpunkt nicht mehr als 4-mal `HousekeepingFrequency` Stunden veralteter Informationen gelöscht. Wenn `HousekeepingFrequency` also 1 ist, werden pro Zyklus nicht mehr als 4 Stunden veralteter Informationen (beginnend mit dem ältesten Eintrag) gelöscht.
Hinweis: Um die Last beim Start des Servers zu verringern, wird Housekeeping für 30 Minuten nach dem Serverstart verschoben. Wenn `HousekeepingFrequency` also 1 ist, wird der allererste Housekeeping-Vorgang nach dem Serverstart nach 30 Minuten ausgeführt und danach stündlich wiederholt.
Sie können das automatische Housekeeping deaktivieren, indem Sie `HousekeepingFrequency` auf 0 setzen. In diesem Fall kann der Housekeeping-Vorgang nur über die Laufzeitsteuerungsoption `housekeeper_execute` gestartet werden, und der Umfang der in einem Housekeeping-Zyklus gelöschten veralteten Informationen beträgt das 4-Fache des Zeitraums seit dem letzten Housekeeping-Zyklus, jedoch nicht weniger als 4 Stunden und nicht mehr als 4 Tage.
Siehe auch die Optionen und Details zur Laufzeitsteuerung sowie zum Housekeeping-Vorgang.

Standard: 1
 Bereich: 0-24

Include

Geben Sie ein Verzeichnis oder einzelne Dateien an, die in die Konfigurationsdatei aufgenommen werden sollen.
 Während der Zabbix-Installation wird dieser Parameter auf `/usr/local/etc` gesetzt, sofern Sie dies nicht zur Kompilierzeit ändern.

Allgemeine Regeln für den Parameter `Include`:

- Der Pfad kann relativ zum Speicherort der Datei `zabbix_server.conf` angegeben werden.
- Sie können mehrere `Include`-Parameter in einer Konfigurationsdatei angeben.
- Zabbix verarbeitet den Parameter `Include` sofort, sobald er ihn liest. Platzieren Sie ihn daher am besten am Ende der Konfigurationsdatei.

Regeln für eingebundene Dateien:

- Jede Datei muss lesbar sein.
- Jede Datei muss eine korrekte Syntax haben. Andernfalls wird die Komponente nicht gestartet.

Regeln für das Einbinden von Verzeichnissen oder Dateimustern:

- Gehen Sie nicht davon aus, dass die Einbindung in einer bestimmten Reihenfolge erfolgt (z. B. werden Dateien nicht alphabetisch eingebunden).
- Definieren Sie denselben Parameter nicht in mehr als einer eingebundenen Datei (z. B. um eine allgemeine Einstellung durch eine spezifische zu überschreiben), da Sie nicht vorhersagen können, welcher Wert verwendet wird.
- Achten Sie auf Sicherungskopien, die einige Texteditoren automatisch erstellen (z. B. kann beim Bearbeiten von `include/my_specific.conf` die Datei `include/my_specific.conf.BAK` erstellt werden), da beide Dateien eingebunden werden.

Um nur relevante Dateien aus einem Verzeichnis einzubinden, verwenden Sie den Platzhalter (`*`) für die Mustererkennung.

Beispiel:

```
Include=/absolute/path/to/config/files/*.conf
```

JavaGateway

Geben Sie die IP-Adresse oder den Hostnamen des Zabbix Java gateway an. Nur erforderlich, wenn Java-Poller gestartet sind.

JavaGatewayPort

Geben Sie den Port des Zabbix Java gateway an.

Standard: 10052
 Bereich: 1024-32767

ListenBacklog

Legt die maximale Anzahl ausstehender Verbindungen in der TCP-Warteschlange fest.
Der Standardwert ist eine fest codierte Konstante, die vom System abhängt.
Der maximal unterstützte Wert hängt ebenfalls vom System ab; zu hohe Werte können stillschweigend auf das „implementierungsabhängige Maximum“ gekürzt werden.

Standard: `SOMAXCONN`
 Bereich: 0 - INT_MAX

ListenIP

Geben Sie die IP-Adresse an, auf der der Trapper lauschen soll. Eine durch Kommas getrennte Liste.
 Der Trapper lauscht auf allen Netzwerkschnittstellen, wenn dieser Parameter nicht angegeben ist.

Standard: `0.0.0.0`

ListenPort

Geben Sie den Trapper-Listenport an.

Standard: 10051
 Bereich: 1024-32767

LoadModule

Geben Sie das beim Start des Servers zu ladende Modul an.
Module werden verwendet, um die Funktionalität des Servers zu erweitern.
Das Modul muss sich in dem durch `LoadModulePath` angegebenen Verzeichnis befinden, oder der Pfad muss dem Modulnamen vorangestellt werden.
Wenn der vorangestellte Pfad absolut ist (mit / beginnt), wird `LoadModulePath` ignoriert.
 Formate:

- `LoadModule=<module.so>`
- `LoadModule=<path/module.so>`
- `LoadModule=</abs_path/module.so>`

Es können mehrere `LoadModule`-Parameter angegeben werden.

LoadModulePath

Geben Sie den vollständigen Pfad zum Speicherort der Server-Module an. Die Standardeinstellung hängt von den Kompilierungsoptionen ab.

LogFile

Geben Sie die Server-Logdatei an.

Erforderlich: Ja, wenn LogType auf `file` gesetzt ist; andernfalls nein

LogFileSize

Legt die maximale Größe der Protokolldatei fest (in MB).
0 - automatische Protokollrotation deaktivieren.
Hinweis: Wenn das Größenlimit der Protokolldatei erreicht wird und die Dateirotation aus irgendeinem Grund fehlschlägt, wird die vorhandene Protokolldatei gekürzt und neu begonnen.

Standard: 1
 Bereich: 0-1024
 Erforderlich: Ja, wenn LogType auf `file` gesetzt ist; andernfalls nein

LogSlowQueries

Legt die maximale Dauer fest, die eine Datenbankabfrage vor dem Protokollieren benötigen darf (in Millisekunden).
0 - langsame Abfragen nicht protokollieren.
Diese Option wird ab DebugLevel=3 aktiviert.

Standard: 0
 Bereich: 0-3600000

LogType

Geben Sie den Typ der Protokollausgabe an: - *file* - schreibt das Protokoll in die durch den Parameter LogFile angegebene Datei; - *system* - schreibt das Protokoll in syslog; - *console* - schreibt das Protokoll in die Standardausgabe.

Standardwert: `file`

MaxConcurrentChecksPerPoller

Legt die maximale Anzahl asynchroner Prüfungen fest, die von jedem HTTP-Agent-Poller, Agent-Poller oder SNMP-Poller gleichzeitig ausgeführt werden können. Siehe StartHTTPAgentPollers, StartAgentPollers und StartSNMPPollers.

Standard: 1000
 Bereich: 1-1000

MaxHousekeeperDelete

Legt die maximale Anzahl von Zeilen fest, die pro Aufgabe in einem Housekeeping-Zyklus aus den Tabellen `history`, `history_str`, `history_log`, `history_uint`, `history_text`, `history_bin`, `history_json`, `trends`, `trends_uint` und `problem` gelöscht werden.
Housekeeping-Aufgaben sind in der Tabelle `housekeeper` im Format `[housekeeperid]`, `[object]`, `[objectid]` vorhanden.
 Dieser Parameter gilt nur für das Löschen von Daten, die von bereits gelöschten Datenpunkten zurückgelassen wurden.
 Wenn er auf 0 gesetzt ist, wird überhaupt kein Limit verwendet.
 In diesem Fall müssen Sie genau wissen, was Sie tun, damit die Datenbank nicht überlastet wird.
 Wenn viele Datenpunkte gelöscht werden, erhöht sich die Last für die Datenbank, da der Housekeeper alle Historiedaten entfernen muss, die diese Datenpunkte hatten.
 Wenn wir beispielsweise nur 1 Vorlagenprototyp aus der Vorlage entfernen müssen, diese Vorlage jedoch mit 50 Hosts verknüpft ist und für jeden Host der Prototyp in 100 echte Datenpunkte erweitert wird, müssen insgesamt 5000 Datenpunkte entfernt werden (1*50*100).
 Wenn für `MaxHousekeeperDelete` (`MaxHousekeeperDelete=500`) der Wert 500 festgelegt ist, muss der Housekeeper-Prozess in einem Zyklus bis zu 2500000 Werte (5000*500) für die gelöschten Datenpunkte aus den Tabellen history und trends entfernen.

Siehe auch Details zum Housekeeping-Verfahren.

Standard: 5000
 Bereich: 0-1000000

NodeAddress

Geben Sie IP oder Hostname mit optionalem Port an, um zu überschreiben, wie das Frontend eine Verbindung zum Server herstellt.
Format: <address>[:<port>]

Wenn IP oder Hostname nicht festgelegt ist, wird der Wert von ListenIP verwendet. Wenn ListenIP nicht festgelegt ist, wird der Wert `localhost` verwendet.
Wenn kein Port festgelegt ist, wird der Wert von ListenPort verwendet. Wenn ListenPort nicht festgelegt ist, wird der Wert 10051 verwendet.

Diese Option kann durch die in der Frontend-Konfiguration angegebene Adresse überschrieben werden.

Siehe auch: Parameter HANodeName; High Availability aktivieren.

Standard: `localhost:10051`

PidFile

Geben Sie die PID-Datei an.

Standard: `/tmp/zabbix_server.pid`

ProblemHousekeepingFrequency

Legt die Häufigkeit des Löschens von Problemen für gelöschte Auslöser fest (in Sekunden).
 Siehe auch die Details zum Housekeeping-Verfahren.

Standard: 60
 Bereich: 1-3600

ProxyConfigFrequency

Legt die Häufigkeit fest, mit der der Server Konfigurationsdaten an den Zabbix-Proxy sendet. Wird nur für Proxys im passiven Modus verwendet.

Standard: 10
 Bereich: 1-604800

ProxyDataFrequency

Legt die Häufigkeit fest, mit der der Server Verlaufsdaten vom Zabbix-Proxy anfordert. Wird nur für Proxys im passiven Modus verwendet.

Standard: 1
 Bereich: 1-3600

ServiceManagerSyncFrequency

Legt die Häufigkeit der Synchronisierung der Konfiguration eines Service Managers fest (in Sekunden).

Standard: 60
 Bereich: 1-3600

SMSDevices

Geben Sie Modemdateien an, die vom Zabbix Server verwendet werden dürfen. Eine durch Kommas getrennte Liste.
 Das Senden von SMS ist nicht möglich, wenn dieser Parameter nicht gesetzt ist.

Beispiel:

```
SMSDevices=/dev/ttyUSB0,/dev/ttyUSB1
```

SNMPTrapperFile

Geben Sie die temporäre Datei an, die zum Übergeben von Daten vom SNMP-Trap-Daemon an den Server verwendet wird.
Sie muss mit der in `zabbix_trap_receiver.pl` oder der SNMPTT-Konfigurationsdatei angegebenen Datei übereinstimmen.

Standard: `/tmp/zabbix_traps.tmp`

SocketDir

Geben Sie das Verzeichnis an, in dem IPC-Sockets gespeichert werden, die von internen Zabbix-Diensten verwendet werden.

Standard: `/tmp`

SourceIP

Geben Sie die Quell-IP-Adresse an für:

- ausgehende Verbindungen zu Zabbix Proxy und Zabbix Agent
- agentenlose Verbindungen (VMware, SSH, JMX, SNMP, Telnet und einfache Prüfungen)
- HTTP-Agent-Verbindungen
- JavaScript-HTTP-Anfragen für Skript-Datenpunkte
- JavaScript-HTTP-Anfragen für die Vorverarbeitung
- das Senden von Benachrichtigungs-E-Mails (Verbindungen zum SMTP-Server)
- webhook-Benachrichtigungen (JavaScript-HTTP-Verbindungen)
- Verbindungen zum Vault

SSHKeyLocation

Geben Sie den Speicherort der öffentlichen und privaten Schlüssel für SSH-Prüfungen und -Aktionen an.

SSLCertLocation

Geben Sie den Speicherort der SSL-Clientzertifikatdateien für die Clientauthentifizierung an.
 Dieser Parameter wird beim Web-Monitoring, für die Kommunikation mit Vault, für den Verlaufsspeicheranbieter ClickHouse und für die ClickHouse-Telemetriedatenquelle verwendet.

SSLKeyLocation

Geben Sie den Speicherort der privaten SSL-Schlüsseldateien für die Clientauthentifizierung an.
 Dieser Parameter wird beim Web-Monitoring, für die Kommunikation mit Vault, für den Verlaufsdatenspeicher ClickHouse und für die Telemetriedatenquelle ClickHouse verwendet.

SSLCALocation

Geben Sie den Speicherort der Dateien der Zertifizierungsstelle (CA) für die Überprüfung von SSL-Serverzertifikaten an. Wenn dieser Parameter nicht festgelegt ist, wird das systemweite Verzeichnis verwendet.
 Beachten Sie, dass der Wert dieses Parameters als libcurl-Option `CURLOPT_CAPATH` festgelegt wird. Bei libcurl-Versionen vor 7.42.0 hat dies nur Auswirkungen, wenn

libcurl für die Verwendung von OpenSSL kompiliert wurde. Weitere Informationen finden Sie auf der cURL-Webseite.
 Dieser Parameter wird beim Web-Monitoring, bei der SMTP-Authentifizierung, für HTTP-Agent-Datenpunkte, für die Kommunikation mit Vault, für den Verlaufsspeicheranbieter ClickHouse und für die Telemetriedatenquelle ClickHouse verwendet.

StartAgentPollers

Legt die Anzahl der vorab erzeugten Instanzen von Zabbix Agent Pollern fest. Siehe MaxConcurrentChecksPerPoller.

Standard: 1
 Bereich: 0-1000

StartAlerters

Legt die Anzahl der vorab per Fork erzeugten Instanzen von alerters fest.

Standard: 3
 Bereich: 1-100

StartBrowserPollers

Legt die Anzahl der vorab erzeugten Instanzen von Browser-Datenpunkt-Pollern fest.

Standard: 1
 Bereich: 0-1000

StartConnectors

Legt die Anzahl der vorab erzeugten Instanzen von Connector-Workern fest. Der Prozess des Connector-Managers wird automatisch gestartet, wenn ein Connector-Worker gestartet wird.

Standard: 0
 Bereich: 0-1000

StartDBSyncers

Legt die Anzahl der vorab erzeugten Instanzen von history syncers fest.
Hinweis: Seien Sie vorsichtig, wenn Sie diesen Wert ändern; eine Erhöhung kann mehr schaden als nützen. Grob gesagt sollte der Standardwert ausreichen, um bis zu 4000 NVPS zu verarbeiten.

Standard: 4
 Bereich: 1-100

StartDiscoverers

Legt die Anzahl der vorab erzeugten Instanzen von discovery workers[1] fest.

Standard: 5
 Bereich: 0-1000

StartEscalators

Legt die Anzahl der vorab erzeugten Instanzen von Escalators fest.

Standard: 1
 Bereich: 1-100

StartHistoryPollers

Legt die Anzahl der vorab geforkten Instanzen von history pollers fest.
Nur für berechnete Prüfungen erforderlich.

Standard: 5
 Bereich: 0-1000

StartHTTPAgentPollers

Legt die Anzahl der vorab erzeugten Instanzen von HTTP-Agent-Pollern fest. Siehe MaxConcurrentChecksPerPoller.

Standard: 1
 Bereich: 0-1000

StartHTTPPollers

Legt die Anzahl der vorab erzeugten Instanzen von HTTP pollers[1] fest.

Standard: 1
 Bereich: 0-1000

StartIPMIPollers

Legt die Anzahl der vorab erzeugten Instanzen von IPMI pollers fest.

Standard: 0
 Bereich: 0-1000

StartJavaPollers

Legt die Anzahl der vorab erzeugten Instanzen von Java pollers[1] fest.

Standard: 0
 Bereich: 0-1000

StartLLDProcessors

Legt die Anzahl der vorab erzeugten Instanzen von Low-Level-Discovery-(LLD)-Workern[1] fest.
 Der LLD-Manager-Prozess wird automatisch gestartet, wenn ein LLD-Worker gestartet wird.

Standard: 2
 Bereich: 1-100

StartODBCPollers

Legt die Anzahl der vorab erzeugten Instanzen von ODBC pollers[1] fest.

Standard: 1
 Bereich: 0-1000

StartPingers

Legt die Anzahl der vorab per Fork erzeugten Instanzen von ICMP pingers[1] fest.

Standard: 1
 Bereich: 0-1000

StartPollersUnreachable

Legt die Anzahl der vorab erzeugten Instanzen von Pollern für nicht erreichbare Hosts fest (einschließlich IPMI und Java)[1].
 Mindestens ein Poller für nicht erreichbare Hosts muss ausgeführt werden, wenn reguläre, IPMI- oder Java-Poller gestartet werden.

Standard: 1
 Bereich: 0-1000

StartPollers

Legt die Anzahl der vorab erzeugten Instanzen von pollers[1] fest.

Standard: 5
 Bereich: 0-1000

StartPreprocessors

Legt die Anzahl der vorab gestarteten Instanzen von Preprocessing-worker[1]-Threads fest und sollte nicht kleiner als die Anzahl der verfügbaren CPU-Kerne sein. Es sollten mehr Worker festgelegt werden, wenn das Preprocessing nicht CPU-gebunden ist und viele Netzwerkanfragen hat.

Standard: 16
 Bereich: 1-1000

StartProxyPollers

Legt die Anzahl der vorab erzeugten Instanzen von Pollern für passive Proxies[1] fest.

Standard: 1
 Bereich: 0-250

StartReportWriters

Legt die Anzahl der vorab per Fork erzeugten Instanzen von report writers fest.
Wenn auf 0 gesetzt, ist die geplante Berichtserstellung deaktiviert.
Der Prozess des Berichtsmanagers wird automatisch gestartet, wenn ein report writer gestartet wird.

Standard: 0
 Bereich: 0-100

StartSNMPPollers

Legt die Anzahl der vorab erzeugten Instanzen von SNMP-Pollern fest. Siehe `MaxConcurrentChecksPerPoller`.

Standard: 1
 Bereich: 0-1000

StartSNMPTrapper

Auf 1 setzen, um einen SNMP trapper-Prozess zu starten.

Standard: 0
 Bereich: 0-1

StartTelemetryQueryPollers

Legt die Anzahl der vorab gestarteten Instanzen asynchroner Telemetrieabfrage-Poller fest.
Siehe MaxConcurrentChecksPerPoller.

Standard: 1
 Bereich: 0-1000

StartTimers

Legt die Anzahl der vorab geforkten Instanzen von timers fest.
Der Timers-Prozess verarbeitet Wartungszeiträume.

Standard: 1
 Bereich: 1-1000

StartTrappers

Legt die Anzahl der vorab erzeugten Instanzen von trappers[1] fest.
 Trappers akzeptieren eingehende Verbindungen von Zabbix sender, aktiven Agenten und aktiven Proxies.

Standard: 5
 Bereich: 0-1000

StartVMwareCollectors

Legt die Anzahl der vorab erzeugten VMware-Sammler-Instanzen fest.

Standard: 0
 Bereich: 0-250

StatsAllowedIP

Geben Sie IP-Adressen (optional in CIDR-Notation) oder DNS-Namen externer Zabbix-Instanzen an, die berechtigt sind, Statistiken anzufordern. Eine durch Kommas getrennte Liste.
 Wenn dieser Parameter nicht gesetzt ist, werden keine Statistikanfragen akzeptiert.
 Wenn die IPv6-Unterstützung aktiviert ist, werden `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` gleich behandelt, und `::/0` erlaubt jede IPv4- oder IPv6-Adresse. `0.0.0.0/0` kann verwendet werden, um jede IPv4-Adresse zuzulassen.

Beispiel:

```
StatsAllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

TelemetryProvider

Legen Sie die Telemetriedatenquelle im Format `<provider>[;<options>]` fest, wobei:

- `<provider>` - eine der unterstützten Telemetriedatenquellen: `clickhouse`
- `<options>` - durch Kommas getrennte `key=value`-Paare. Werte können in Anführungszeichen gesetzt werden.

Optionen für ClickHouse:

- `url` - legt die HTTP- oder HTTPS-URL fest, die für die Verbindung zur Telemetriedatenquelle verwendet wird.
- `db` - legt den Namen der Datenbank fest.
- `username` - legt den Benutzernamen für die Anmeldung an der Datenbank fest.
- `password` - legt das Passwort für die Anmeldung an der Datenbank fest.
- `ssl_cert_file`, `ssl_key_file`, `ssl_key_password`, `ssl_verify_peer`, `ssl_verify_host`, `ssl_ca_location`, `ssl_cert_location`, `ssl_key_location` - legen Sie diese Optionen fest, um eine sichere (TLS/SSL-)Verbindung zur Datenbank zu konfigurieren.
- `source_ip` - legt die Quell-IP-Adresse fest, die für ausgehende Verbindungen zur Telemetriedatenquelle verwendet wird.
- `vault_path` - legt den Pfad zu einem Vault-Secret fest, das die Werte `username` und `password` enthält. Verwenden Sie diese Option nicht zusammen mit den Optionen `username` und `password`.

Beispiel für ClickHouse:

```
TelemetryProvider=clickhouse;url=http://localhost:8123,db=zabbix,username=zabbix,password=zabbix
```

Beispiel für ClickHouse mit Vault:

```
TelemetryProvider=clickhouse;url=http://localhost:8123,db=zabbix,vault_path=clickhouse
```

Timeout

Geben Sie an, wie lange beim Aufbau der Verbindung und beim Austausch von Daten mit dem Zabbix Proxy, Agent, Webdienst und bei Legacy-SNMP-Prüfungen gewartet werden soll (eine einzelne OID-Nummer oder ein String).

Dieser Parameter definiert die Dauer für verschiedene Kommunikationsvorgänge:

- Ausführung von Remote-Befehlen auf dem Zabbix Agent
- Ausführung von SSH/Telnet-Befehlen
- Anfragen an den Zabbix-Webdienst
- Kommunikations-Timeout für die Mediatype-Testanfrage und die Methode `history.push`
- Neuplanung von Datenpunkten, wenn eine IPMI-Schnittstelle nicht verfügbar wird
- Senden einer Antwort an den Zabbix Proxy, wenn der Datenaustausch aufgrund von Berechtigungs- oder Verschlüsselungsproblemen fehlschlägt
- Frist für IPC-Async-Sockets und Laufzeitsteuerungsoptionen
- JMX-Verbindungen
- Abrufen von Statistiken von einem entfernten Zabbix Proxy oder Server
- Senden von Antworten an das Zabbix Frontend
- DNS-Anfragen des asynchronen Pollers
- Antwort für das Heartbeat von aktiven Prüfungen
- Abrufen von Zabbix Agent-Daten (Werte) von aktiven Agents
- Abrufen von Daten vom Zabbix Sender
- Antwort, wenn der Trapper eine Anfrage nicht verarbeiten kann
- Senden der Liste aktiver Prüfungen an den Zabbix Agent

Dieses Timeout wird **nicht** für jene Prüfungen verwendet, für die im Frontend flexible Timeout-Einstellungen konfiguriert sind (auf globaler, Proxy- oder pro-Datenpunkt-Ebene). Beispielsweise verwenden SNMP-`walk[OID]`- und `get[OID]`-Datenpunkte das im Frontend konfigurierte Timeout; Legacy-SNMP-Prüfungen verwenden weiterhin den Server-Timeout-Wert.

Standard: 3
 Bereich: 1-30

TLSCAFile

Geben Sie den vollständigen Pfadnamen der Datei an, die die Zertifikate der obersten CA(s) für die Überprüfung von Peer-Zertifikaten enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

TLSCertFile

Geben Sie den vollständigen Pfadnamen der Datei an, die das Serverzertifikat oder die Zertifikatskette enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

TLSCipherAll

Geben Sie den GnuTLS-Prioritätsstring oder den OpenSSL-Cipher-String (TLS 1.2) an. Überschreiben Sie die standardmäßigen Auswahlkriterien für die Ciphersuite bei zertifikats- und PSK-basierter Verschlüsselung.

Beispiel:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

Geben Sie die Cipher-String für OpenSSL 1.1.1 oder neuer in TLS 1.3 an. Überschreiben Sie die Standardkriterien für die Auswahl der Cipher Suite für zertifikats- und PSK-basierte Verschlüsselung.

Beispiel für GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL
```

Beispiel für OpenSSL:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherCert

Geben Sie die GnuTLS-Prioritätszeichenfolge oder die OpenSSL-Cipher-Zeichenfolge (TLS 1.2) an. Überschreiben Sie die standardmäßigen Auswahlkriterien für die Cipher Suite bei zertifikatsbasierter Verschlüsselung.

Beispiel für GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Beispiel für OpenSSL:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

TLSCipherCert13

Geben Sie die Cipher-String für OpenSSL 1.1.1 oder neuer in TLS 1.3 an. Überschreiben Sie die standardmäßigen Auswahlkriterien für die Cipher Suite für zertifikatsbasierte Verschlüsselung.

TLSCipherPSK

Geben Sie die GnuTLS-Prioritätszeichenfolge oder die Cipher-String von OpenSSL (TLS 1.2) an. Überschreiben Sie die standardmäßigen Auswahlkriterien für die Ciphersuite bei PSK-basierter Verschlüsselung.

Beispiel für GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Beispiel für OpenSSL:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

Geben Sie die Cipher-String für OpenSSL 1.1.1 oder neuer in TLS 1.3 an. Überschreiben Sie die standardmäßigen Auswahlkriterien für die Ciphersuite bei PSK-basierter Verschlüsselung.

Beispiel:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCRLFile

Geben Sie den vollständigen Pfadnamen der Datei an, die widerrufene Zertifikate enthält. Dieser Parameter wird für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet.

TLSFrontendAccept

Geben Sie die Verschlüsselungsstufe für eingehende Verbindungen vom Frontend an.

Es können mehrere Werte angegeben werden, getrennt durch Kommas:

- unencrypted - Verbindungen ohne Verschlüsselung akzeptieren.
- cert - Verbindungen akzeptieren, die mit TLS und einem Zertifikat gesichert sind.

Standard: `unencrypted`

TLSFrontendCertIssuer

Geben Sie den zulässigen Aussteller des Frontend-Zertifikats an.

TLSFrontendCertSubject

Geben Sie den zulässigen Betreff des Frontend-Zertifikats an.

TLSKeyFile

Geben Sie den vollständigen Pfadnamen der Datei an, die den privaten Schlüssel des Servers enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

TLSListen

TLS-only-Verbindungen auf dem Trapper-Socket erzwingen.

Unterstützte Werte:

- required - nur TLS-Verbindungen akzeptieren

TmpDir

Geben Sie das temporäre Verzeichnis an.

Standard: `/tmp`

TrapperTimeout

Geben Sie das Trapper-Timeout in Sekunden an für:

- das Abrufen historischer Daten von Zabbix Proxy
- das Senden von Konfigurationsdaten an Zabbix Proxy
- die Ausführung globaler Skripte oder die Ausführung von Remote-Befehlen auf Zabbix Server

Standard: 300
 Bereich: 1-300

TrendCacheSize

Legt die Größe des Trend-Cache in Bytes fest.
Die Größe des gemeinsam genutzten Speichers zum Speichern von Trends.

Standard: `4M`
 Bereich: 128K-16G

TrendFunctionCacheSize

Legt die Größe des Cache für Trendfunktionen in Bytes fest.
Die Größe des gemeinsam genutzten Speichers zum Zwischenspeichern berechneter Trendfunktionsdaten.

Standard: `4M`
 Bereich: 128K-2G

UnavailableDelay

Legt die Häufigkeit der Überprüfung der Host-Verfügbarkeit während des Zeitraums unavailability in Sekunden fest.

Standard: 60
 Bereich: 1-3600

UnreachableDelay

Legt die Häufigkeit fest, mit der die Verfügbarkeit des Hosts während des [Unreachable]-Zeitraums (/manual/appendix/items/unreachability#ur
interface) in Sekunden überprüft wird.

Standard: 15
 Bereich: 1-3600

UnreachablePeriod

Legt fest, nach wie vielen Sekunden der Unerreichbarkeit der Host als nicht verfügbar behandelt wird.

Standard: 45
 Bereich: 1-3600

Benutzer

Rechte auf einen bestimmten, vorhandenen Benutzer auf dem System herabstufen.
Wirkt nur, wenn es als 'root' ausgeführt wird und AllowRoot deaktiviert ist.

Standard: zabbix

ValueCacheSize

Legt die Größe des Verlaufswert-Cache (in Byte) fest.
 Die Größe des gemeinsam genutzten Speichers zum Zwischenspeichern von Verlaufsdatenanforderungen für Datenpunkte.
 Die Einstellung 0 deaktiviert den Wert-Cache (nicht empfohlen).
 Wenn der Wert-Cache keinen gemeinsam genutzten Speicher mehr zur Verfügung hat, wird alle 5 Minuten eine Warnmeldung in das Server-Log geschrieben.

Standard: 8M
 Bereich: 0, 128K-64G

Vault

Geben Sie den Vault-Anbieter an: - *HashiCorp* - HashiCorp KV Secrets Engine Version 2; - *CyberArk* - CyberArk Central Credential Provider
muss mit dem im Frontend festgelegten Vault-Anbieter übereinstimmen.

Standard: HashiCorp

VaultAppRoleID

Geben Sie die HashiCorp Vault AppRole-ID an.
 Wenn festgelegt, wird die AppRole-Authentifizierung aktiviert. Kann nicht zusammen mit VaultToken verwendet werden.
 Nur für HashiCorp Vault unterstützt.

VaultAppSecretID

Geben Sie die Secret ID der HashiCorp-Vault-AppRole an.
 Nur für HashiCorp Vault unterstützt.

VaultDBPath

Geben Sie den Speicherort an, von dem Datenbankanmeldedaten anhand von Schlüsseln abgerufen werden. Dabei handelt es sich je nach Vault um einen Vault-Pfad oder eine Abfrage.

Die für HashiCorp verwendeten Schlüssel sind `password` und `username`.

Beispielpfad mit `VaultPrefix=/v1/secret/data/zabbix/`:

```
database
```

Beispielpfad ohne VaultPrefix:

```
secret/zabbix/database
```

Die für CyberArk verwendeten Schlüssel sind `Content` und `UserName`.

Beispiel:

```
AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix_server_database
```

Diese Option kann nur verwendet werden, wenn `DBUser` und `DBPassword` nicht angegeben sind.

VaultPrefix

Geben Sie ein benutzerdefiniertes Präfix für den Vault-Pfad oder die Abfrage an, abhängig vom Vault. Wenn nichts angegeben wird, werden die am besten geeigneten Standardwerte verwendet.
 Beachten Sie, dass `data` bei HashiCorp automatisch nach dem Mountpoint angehängt wird, wenn `VaultPrefix` nicht angegeben ist.

Beispiel für ein Präfix für HashiCorp:

```
/v1/secret/data/zabbix/
```

Beispiel für ein Präfix für Cyberark:

```
/AIMWebService/api/Accounts?
```

VaultTLSCertFile

Geben Sie den Namen der SSL-Zertifikatsdatei an, die für die Client-Authentifizierung verwendet wird. Die Zertifikatsdatei muss im PEM1-Format vorliegen.
 Wenn die Zertifikatsdatei auch den privaten Schlüssel enthält, lassen Sie das Feld für die SSL-Schlüsseldatei leer.
 Das Verzeichnis, das diese Datei enthält, wird durch den Konfigurationsparameter SSLCertLocation angegeben.
 Diese Option kann weggelassen werden, wird jedoch für den CyberArkCCP-Vault empfohlen.

VaultTLSKeyFile

Geben Sie den Namen der SSL-Privatschlüsseldatei an, die für die Client-Authentifizierung verwendet wird. Die Privatschlüsseldatei muss im PEM1-Format vorliegen.
Das Verzeichnis, das diese Datei enthält, wird durch den Konfigurationsparameter SSLKeyLocation angegeben.
Diese Option kann weggelassen werden, wird jedoch für den CyberArkCCP-Vault empfohlen.

VaultToken

Geben Sie das Authentifizierungstoken für den HashiCorp Vault an. Das HashiCorp-Vault-Authentifizierungstoken sollte ausschließlich für den Server von Zabbix mit Nur-Lese-Berechtigung für die in Vault macros angegebenen Pfade und mit Nur-Lese-Berechtigung für den im optionalen Konfigurationsparameter VaultDBPath angegebenen Pfad generiert worden sein.
 Es ist ein Fehler, wenn `VaultToken` und die Umgebungsvariable `VAULT_TOKEN` gleichzeitig definiert sind.
 Kann nicht mit VaultAppRoleID verwendet werden.
 Wird nur für HashiCorp Vault unterstützt.

VaultURL

Geben Sie die URL des Vault-Servers an. Das systemweite Verzeichnis für CA-Zertifikate wird verwendet, wenn SSLCALocation nicht angegeben ist.

Standard: `https://127.0.0.1:8200`

VMwareCacheSize

Legt die Größe des gemeinsam genutzten Speichers zum Speichern von VMware-Daten fest. Mit der internen Prüfung `zabbix[vmware,buffer,...]` kann die Nutzung des VMware-Cache überwacht werden (siehe Interne Prüfungen).
Beachten Sie, dass kein gemeinsam genutzter Speicher zugewiesen wird, wenn keine `vmware collector`-Instanzen so konfiguriert sind, dass sie gestartet werden.

Standard: 8M
 Bereich: 256K-2G

VMwareFrequency

Legt die Häufigkeit der Datenerfassung von einem einzelnen VMware-Dienst fest (in Sekunden). Diese Häufigkeit sollte auf das kleinste Aktualisierungsintervall eines beliebigen VMware-Überwachungs-Datenpunkts gesetzt werden.

Standard: 60
 Bereich: 10-86400

VMwarePerfFrequency

Legt die Häufigkeit der Abfrage von Leistungsindikator-Statistiken von einem einzelnen VMware-Dienst fest (in Sekunden).
Diese Häufigkeit sollte auf das kleinste Aktualisierungsintervall eines beliebigen VMware-Monitoring-Datenpunkts gesetzt werden, der VMware-Leistungsindikatoren verwendet.

Standard: 60
 Bereich: 10-86400

VMwareTimeout

Legt die maximale Anzahl von Sekunden fest, die ein `vmware collector` auf eine Antwort vom VMware-Dienst (vCenter oder ESX-Hypervisor) wartet.

Standard: 10
 Bereich: 1-300

WebDriverURL

Geben Sie die URL der WebDriver-Schnittstelle an.

Beispiel (verwendet mit dem eigenständigen Selenium WebDriver-Server):

```
WebDriverURL=http://localhost:4444
```

WebServiceURL

Geben Sie die Webservice-URL im Format `http[s]://host:port/report` an.

Beispiel:

```
WebServiceURL=http://localhost:10053/report
```

Das Schema `http://` kann nur bei Verbindungen ohne TLS (HTTP) weggelassen werden; bei konfiguriertem TLS muss `https://` verwendet werden.

Fußnoten

[1] Beachten Sie, dass zu viele Datenerfassungsprozesse (Poller, unerreichbare Poller, ODBC-Poller, HTTP-Poller, Java-Poller, Pinger, Trapper, Proxypoller) zusammen mit dem IPMI-Manager, SNMP-Trapper, Vorverarbeitungs-Workern und Discovery-Workern das pro Prozess geltende Limit für Dateideskriptoren des Vorverarbeitungs-Managers erschöpfen können.

Warning:
Wenn das Limit für Dateideskriptoren erschöpft ist, wird der Zabbix Server angehalten, typischerweise kurz nach dem Start, manchmal jedoch erst nach längerer Zeit. Um solche Probleme zu vermeiden, prüfen Sie die Konfigurationsdatei des Zabbix Server, um die Anzahl gleichzeitiger Prüfungen und Prozesse zu optimieren. Stellen Sie außerdem bei Bedarf sicher, dass das Limit für Dateideskriptoren ausreichend hoch gesetzt ist, indem Sie die Systemlimits überprüfen und anpassen.

Proxy

Übersicht Der Zabbix Proxy ist ein Prozess, der Überwachungsdaten von einem oder mehreren überwachten Geräten erfassen und die Informationen an den Zabbix Server senden kann. Dabei arbeitet er im Wesentlichen im Auftrag des Servers. Alle erfassten Daten werden lokal zwischengespeichert und anschließend an den Zabbix Server übertragen, zu dem der Proxy gehört.

Die Bereitstellung eines Proxys ist optional, kann jedoch sehr vorteilhaft sein, um die Last eines einzelnen Zabbix Servers zu verteilen. Wenn nur Proxys Daten erfassen, benötigt die Verarbeitung auf dem Server weniger CPU- und Festplatten-I/O-Ressourcen.

Ein Zabbix Proxy ist die ideale Lösung für die zentrale Überwachung entfernter Standorte, Zweigstellen und Netzwerke ohne lokale Administratoren.

Für den Zabbix Proxy ist eine separate Datenbank erforderlich.

Attention:
Beachten Sie, dass SQLite, MySQL und PostgreSQL von Zabbix Proxy unterstützt werden.

Siehe auch:

- Verteilte Überwachung mit Proxys
- Lastverteilung und Hochverfügbarkeit von Proxys
- Synchronisierung der Überwachungskonfiguration
- Protokoll für den Datenaustausch zwischen Server und Proxy
- Konfigurationsparameter: Zabbix Proxy
- Einrichten eines Zabbix Cloud Proxys

Ausführender Proxy Wenn als Paket installiert

Der Zabbix Proxy läuft als Daemon-Prozess. Der Proxy kann mit folgendem Befehl gestartet werden:

```
systemctl start zabbix-proxy
```

Dies funktioniert auf den meisten GNU/Linux-Systemen. Auf anderen Systemen müssen Sie möglicherweise Folgendes ausführen:

```
/etc/init.d/zabbix-proxy start
```

Verwenden Sie entsprechend die folgenden Befehle, um den Zabbix Proxy zu stoppen, neu zu starten oder seinen Status anzuzeigen:

```
systemctl stop zabbix-proxy
systemctl restart zabbix-proxy
systemctl status zabbix-proxy
```

Manuell starten

Wenn das oben nicht funktioniert, müssen Sie es manuell starten. Suchen Sie den Pfad zur `zabbix_proxy`-Binärdatei und führen Sie Folgendes aus:

```
zabbix_proxy
```

Sie können die folgenden Befehlszeilenparameter mit Zabbix Proxy verwenden:

Parameter	Description
`-c --config <config-file>`	Pfad zur Konfigurationsdatei.
`-f --foreground`	Zabbix Proxy im Vordergrund ausführen.
`-R --runtime-control <option>`	Administrative Funktionen ausführen.
`-T --test-config`	Konfigurationsdatei validieren und beenden.
`-h --help`	Hilfsinformationen anzeigen und beenden.

Parameter	Description
`-V --version`	Versionsnummer anzeigen.

Beispiele für das Ausführen von Zabbix Proxy mit Befehlszeilenparametern:

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

Laufzeitsteuerung

Optionen der Laufzeitsteuerung:

Option	Beschreibung	Ziel
`config_cache_reload`	Konfigurationscache neu laden. Wird ignoriert, wenn der Cache derzeit geladen wird. Ein aktiver Zabbix Proxy verbindet sich mit dem Zabbix Server und fordert Konfigurationsdaten an. Ein passiver Zabbix Proxy fordert beim nächsten Verbindungsaufbau des Servers zum Proxy Konfigurationsdaten vom Zabbix Server an.	
`history_cache_clear[<target>]`	Den Verlaufscache für den durch seine ID angegebenen Datenpunkt leeren. Wirkt sich auf alle Werte des Datenpunkts aus, außer auf den ersten und den letzten Wert.	**target** - ID des Datenpunkts.
`diaginfo[=<section>]`	Diagnoseinformationen in der Protokolldatei des Proxys sammeln.	`historycache` - Statistiken des Verlaufscaches; `preprocessing` - Statistiken des Preprocessing-Managers; `locks` - Liste der Mutexes (ist auf *BSD*-Systemen leer).
`snmp_cache_reload`	SNMP-Cache neu laden - SNMP-Engine-Eigenschaften (Engine-Zeit, Engine-Boots, Engine-ID, Anmeldedaten) für alle Hosts löschen. Verwenden Sie dies, um beim Troubleshooting von SNMP-Problemen einen globalen Cache-Löschvorgang zu erzwingen.	
`housekeeper_execute`	Die Housekeeping-Prozedur starten. Wird ignoriert, wenn die Housekeeping-Prozedur derzeit ausgeführt wird.	
`log_level_increase[=<target>]`	Protokollierungsstufe erhöhen, betrifft alle Prozesse, wenn kein Ziel angegeben ist. Auf *BSD*-Systemen nicht unterstützt.	**process type** - alle Prozesse des angegebenen Typs (z. B. `poller`). Siehe alle Proxy-Prozesstypen. **process type,N** - Prozesstyp und Nummer (z. B. `poller,3`) **pid** - Prozesskennung (1 bis 65535). Für größere Werte geben Sie das Ziel als 'process type,N' an.
`log_level_decrease[=<target>]`	Protokollierungsstufe verringern, betrifft alle Prozesse, wenn kein Ziel angegeben ist. Auf *BSD*-Systemen nicht unterstützt.	
`prof_enable[=<target>]`	Profiling aktivieren. Betrifft alle Prozesse, wenn kein Ziel angegeben ist. Aktiviertes Profiling liefert Details zu allen rwlocks/Mutexes nach Funktionsnamen.	**process type** - alle Prozesse des angegebenen Typs (z. B. `history syncer`). Siehe alle Proxy-Prozesstypen. **process type,N** - Prozesstyp und Nummer (z. B. `history syncer,1`). **pid** - Prozesskennung (1 bis 65535). Für größere Werte geben Sie das Ziel als 'process type,N' an. **scope** - `rwlock`, `mutex`, `processing` kann mit dem Prozesstyp und der Nummer verwendet werden (z. B. `history syncer,1,processing`) oder mit allen Prozessen eines Typs (z. B. `history syncer,rwlock`).

Option	Beschreibung	Ziel
`prof_disable[=<target>]`	Profiling deaktivieren. Betrifft alle Prozesse, wenn kein Ziel angegeben ist.	**process type** - alle Prozesse des angegebenen Typs (z. B. `history syncer`). Siehe alle Proxy-Prozesstypen. **process type,N** - Prozesstyp und Nummer (z. B. `history syncer,1`). **pid** - Prozesskennung (1 bis 65535). Für größere Werte geben Sie das Ziel als 'process type,N' an.

Wenn ein Prozesstyp als Ziel für die Laufzeitsteuerung angegeben wird, gilt der Befehl nur für laufende Prozesse dieses Typs. Wenn derzeit kein Prozess des angegebenen Typs ausgeführt wird, gibt der Befehl einen Fehler zurück:

```
zabbix_proxy -R log_level_increase='data sender'
Cannot redirect signal: "data sender" process does not exist
```

Beispiel für die Verwendung der Laufzeitsteuerung zum Neuladen des Konfigurationscaches des Proxys:

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

Beispiel für die Verwendung der Laufzeitsteuerung zum Leeren des Verlaufscaches für einen Datenpunkt:

```
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R history_cache_clear=42243
```

Beispiele für die Verwendung der Laufzeitsteuerung zum Sammeln von Diagnoseinformationen:

```
### Alle verfügbaren Diagnoseinformationen in der Protokolldatei des Proxys sammeln:
zabbix_proxy -R diaginfo

### Statistiken des Verlaufscaches in der Protokolldatei des Proxys sammeln:
zabbix_proxy -R diaginfo=historycache
```

Beispiel für die Verwendung der Laufzeitsteuerung zum Neuladen des SNMP-Caches:

```
zabbix_proxy -R snmp_cache_reload
```

Attention:
Wenn eine SNMPv3-Schnittstelle über die Zabbix UI aktualisiert wird, lädt Zabbix in den meisten Fällen die neuen SNMPv3-Anmeldedaten für diese Schnittstelle automatisch neu; verwenden Sie `-R snmp_cache_reload` nur, wenn das Abfragen nach einer Änderung der Anmeldedaten weiterhin fehlschlägt (zum Beispiel aufgrund von Inkonsistenzen bei engineBoots/engineID oder bei Geräten, die nicht RFC-konform sind), oder wenn Sie zum Troubleshooting einen globalen SNMP-Cache-Löschvorgang erzwingen müssen.

Beispiel für die Verwendung der Laufzeitsteuerung zum Auslösen der Ausführung von Housekeeper:

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

Beispiele für die Verwendung der Laufzeitsteuerung zum Ändern der Protokollierungsstufe:

```
### Protokollierungsstufe aller Prozesse erhöhen:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase

### Protokollierungsstufe des zweiten Poller-Prozesses erhöhen:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=poller,2

### Protokollierungsstufe des Prozesses mit PID 1234 erhöhen:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=1234

### Protokollierungsstufe aller http poller-Prozesse verringern:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_decrease="http poller"
```

Benutzerverarbeitung

Der Zabbix Proxy ist dafür ausgelegt, als Benutzer ohne Root-Rechte ausgeführt zu werden. Er wird als derjenige Benutzer ohne Root-Rechte ausgeführt, unter dem er gestartet wurde. Daher können Sie den Proxy problemlos als beliebigen Benutzer ohne Root-Rechte ausführen.

Wenn Sie versuchen, ihn als „root“ auszuführen, wechselt er zu einem fest codierten Benutzer „zabbix“, der auf Ihrem System vorhanden sein muss. Sie können den Proxy nur als „root“ ausführen, wenn Sie den Parameter `AllowRoot` in der Proxy-Konfigurationsdatei entsprechend ändern.

Konfigurationsdatei

Der Standardpfad zur Konfigurationsdatei hängt vom Betriebssystem und von der Installationsmethode von Zabbix ab:

- Bei einer Installation aus den Quellen: `/usr/local/etc/zabbix_proxy.conf` oder, wie durch Variablen zur Kompilierzeit festgelegt, `--sysconfdir` oder `--prefix`.
- Bei einer Installation aus Paketen: `/etc/zabbix/zabbix_proxy.conf`.

Weitere Informationen zur Konfiguration von `zabbix_proxy` finden Sie in den Optionen der Konfigurationsdatei.

Proxy-Prozesstypen und Threads

- `agent poller` – asynchroner Poller-Prozess für passive Prüfungen mit einem Worker-Thread;
- `apm manager` – Manager-Prozess für Application Performance Monitoring (APM);
- `apm worker` – Prozess zur Verarbeitung von Application-Performance-Monitoring-Aufgaben (APM) des `apm manager`;
- `availability manager` – Prozess zur Aktualisierung der Host-Verfügbarkeit;
- `browser poller` – Poller für Browser-Datenpunktprüfungen;
- `configuration syncer` – Prozess zur Verwaltung des In-Memory-Caches von Konfigurationsdaten;
- `data sender` – Proxy-Datensender;
- `discovery manager` – Manager-Prozess zur Erkennung von Geräten;
- `discovery worker` – Prozess zur Verarbeitung von Erkennungsaufgaben des `discovery manager`;
- `history syncer` – Schreiber für die Verlaufsdatenbank;
- `housekeeper` – Prozess zum Entfernen veralteter Datenpunktverläufe;
- `http agent poller` – asynchroner Poller-Prozess für HTTP-Prüfungen mit einem Worker-Thread;
- `http poller` – Poller für das Web-Monitoring;
- `icmp pinger` – Poller für `icmpping`-Prüfungen;
- `internal poller` – Poller für interne Prüfungen;
- `ipmi manager` – Manager für IPMI-Poller;
- `ipmi poller` – Poller für IPMI-Prüfungen;
- `java poller` – Poller für Java-Prüfungen;
- `odbc poller` – Poller für ODBC-Prüfungen;
- `poller` – normaler Poller für passive Prüfungen;
- `preprocessing manager` – Manager für Vorverarbeitungsaufgaben mit Vorverarbeitungs-Worker-Threads;
- `preprocessing worker` – Thread für die Datenvorverarbeitung;
- `self-monitoring` – Prozess zum Erfassen interner Serverstatistiken;
- `snmp poller` – asynchroner Poller-Prozess für SNMP-Prüfungen mit einem Worker-Thread (nur `walk[OID]`- und `get[OID]`-Datenpunkte);
- `snmp trapper` – Trapper für SNMP-Traps;
- `supervisor` – koordiniert den Start und das Herunterfahren von Server-Prozessen und -Threads und überwacht deren Bereitschaft;
- `task manager` – Prozess zur Remote-Ausführung von Aufgaben, die von anderen Komponenten angefordert werden (z. B. Problem schließen, Problem bestätigen, Datenpunktwert jetzt prüfen, Funktion für Remote-Befehle);
- `telemetry query poller` – Poller für Prüfungen von Telemetrieabfrage-Datenpunkten;
- `trapper` – Trapper für aktive Prüfungen, Traps und die Proxy-Kommunikation;
- `unreachable poller` – Poller für nicht erreichbare Geräte;
- `vmware collector` – VMware-Datensammler, der für das Erfassen von Daten aus VMware-Diensten zuständig ist.

Die Proxy-Protokolldatei kann verwendet werden, um diese Prozesstypen zu beobachten.

Die Proxy-Protokolldatei wird mit Lese- und Schreibberechtigungen ausschließlich für den Dateieigentümer erstellt. Zusätzlich ist die Datei für die Eigentümergruppe lesbar. Alle anderen Berechtigungen sind verweigert.

Verschiedene Typen von Zabbix-Proxy-Prozessen können mithilfe des internen `zabbix[process,<type>,<mode>,<state>]`-Datenpunkts überwacht werden.

Statistik der Transaktionen des History-Syncers

Der Prozessname des History-Syncers zeigt detaillierte Statistiken zu den Transaktionen des History-Syncers an.

```
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
```

Die Zeitangaben in `processed...in N (<timings>) sec` sind:

- Zeit, die für das Schreiben von Datenpunktwerten in die Datenbank aufgewendet wurde.

- Zeit, die für die Aktualisierung der Datenpunktdaten (Status, Fehler) aufgewendet wurde.

Housekeeping-Verfahren

Der Zabbix Proxy verfügt über den Prozess `housekeeper`, der veraltete Datenpunkt-Historie und Trends entfernt. Er läuft in Zyklen, wobei die Häufigkeit durch `HousekeepingFrequency` bestimmt wird und das Löschlimit pro Zyklus durch `ProxyLocalBuffer` und `ProxyOfflineBuffer` festgelegt wird. Im Gegensatz zum Housekeeping-Verfahren des Zabbix Servers verwendet der Prozess `housekeeper` des Proxys nicht die Tabelle `housekeeper` – er löscht alle veralteten Daten einmal pro Housekeeping-Zyklus.

Unterstützte Plattformen Der Zabbix Proxy läuft auf derselben Liste von unterstützten Plattformen wie der Zabbix Server.

Speicherpuffer Mit dem Speicherpuffer können Sie neue Daten (Datenpunktwerte, Netzwerkerkennung, automatische Host-Registrierung) im Puffer speichern und ohne Zugriff auf die Datenbank auf den Zabbix-Server hochladen. Der Speicherpuffer wurde in Zabbix 7.0 für den Proxy eingeführt.

In Installationen vor Zabbix 7.0 wurden die erfassten Daten vor dem Hochladen auf den Zabbix-Server in der Datenbank gespeichert. Für diese Installationen bleibt dieses Verhalten auch nach dem Upgrade auf Zabbix 7.0 standardmäßig bestehen.

Für eine optimierte Leistung wird empfohlen, die Verwendung des Speicherpuffers auf dem Proxy zu konfigurieren. Dies ist möglich, indem Sie den Wert von `ProxyBufferMode` von `disk` (fest codierter Standardwert für bestehende Installationen) auf `hybrid` (empfohlen) oder `memory` ändern. Außerdem muss die Größe des Speicherpuffers festgelegt werden (Parameter `ProxyMemoryBufferSize`).

Im Hybridmodus ist der Puffer vor Datenverlust geschützt, da nicht gesendete Daten in die Datenbank geschrieben werden, wenn der Proxy beendet wird, der Puffer voll ist oder die Daten zu alt sind. Wenn alle Werte in die Datenbank geschrieben wurden, verwendet der Proxy wieder den Speicherpuffer.

Im Speichermodus wird der Speicherpuffer verwendet, es besteht jedoch kein Schutz vor Datenverlust. Wenn der Proxy beendet wird oder der Speicher überläuft, werden die nicht gesendeten Daten verworfen.

Der Hybridmodus (`ProxyBufferMode=hybrid`) wird seit Zabbix 7.0 auf alle neuen Installationen angewendet.

Zusätzliche Parameter wie `ProxyMemoryBufferSize` und `ProxyMemoryBufferAge` definieren jeweils die Größe des Speicherpuffers und das maximale Alter der Daten im Puffer.

Beachten Sie, dass der Proxy bei einer widersprüchlichen Konfiguration einen Fehler ausgibt und nicht gestartet werden kann, zum Beispiel wenn:

- `ProxyBufferMode` auf `hybrid` oder `memory` gesetzt ist und `ProxyMemoryBufferSize` den Wert 0 hat.
- `ProxyBufferMode` auf `hybrid` oder `memory` gesetzt ist und `ProxyLocalBuffer` nicht den Wert 0 hat.

Locale Beachten Sie, dass der Proxy ein UTF-8-Locale benötigt, damit einige textuelle Datenpunkte korrekt interpretiert werden können. Die meisten modernen Unix-ähnlichen Systeme verwenden standardmäßig ein UTF-8-Locale; es gibt jedoch einige Systeme, auf denen dies ausdrücklich festgelegt werden muss.

Berechnung von Warteschlangen während der Wartung

> **Attention:**
> Der Zabbix Proxy ist sich von Wartungszeiträumen nicht bewusst; weitere
> Warteschlangen während der Wartung.

Verwendung des Zabbix-Frontends hinter einem Proxy Wenn das Zabbix-Frontend hinter einem Proxyserver ausgeführt wird, muss der Cookie-Pfad in der Proxy-Konfigurationsdatei umgeschrieben werden, damit er dem über den Reverse-Proxy bereitgestellten Pfad entspricht.
Siehe die folgenden Beispiele.
Wenn der Cookie-Pfad nicht umgeschrieben wird, können bei der Anmeldung am Zabbix-Frontend Autorisierungsprobleme auftreten.

Beispielkonfiguration für NGINX

```
### ..
location / {
### ..
proxy_cookie_path /zabbix /;
proxy_pass http://192.168.0.94/zabbix/;
### ..
```

Beispielkonfiguration für Apache

```
### ..
ProxyPass "/" http://host/zabbix/
ProxyPassReverse "/" http://host/zabbix/
ProxyPassReverseCookiePath /zabbix /
ProxyPassReverseCookieDomain host zabbix.example.com
### ..
```

Verteilte Überwachung mit Proxys

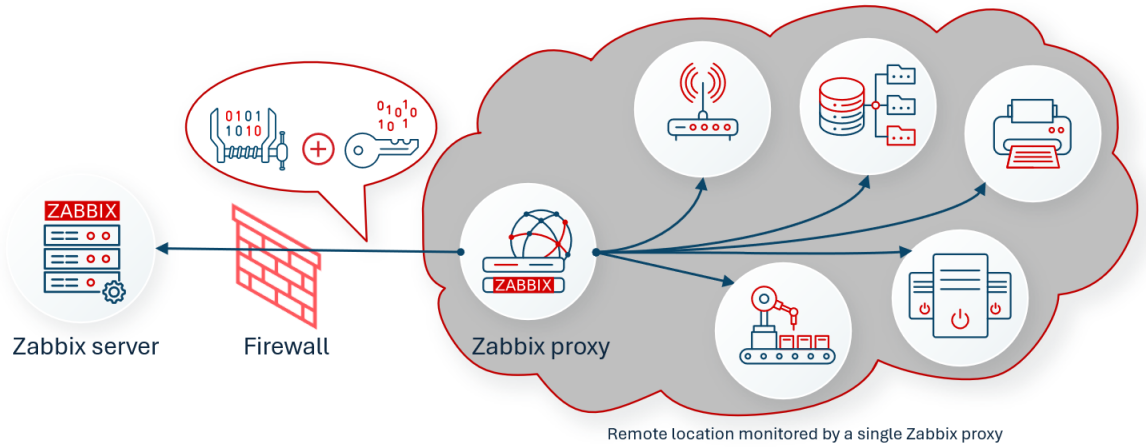
Übersicht

Zabbix-Proxys können verwendet werden, um eine effektive und zuverlässige verteilte Überwachung einzurichten.

Proxys werden lokal bereitgestellt, um Leistungs- und Verfügbarkeitsdaten zu erfassen. Anschließend werden die Daten an einen zentralen Server gemeldet.

Ein Zabbix-Proxy kann verwendet werden, um:

- Remote-Standorte zu überwachen
- Standorte mit unzuverlässiger Kommunikation zu überwachen
- den Zabbix-Server bei der Überwachung von Tausenden Geräten zu entlasten
- Telemetriedaten für das Application Performance Monitoring (APM) zu erfassen



Remote location monitored by a single Zabbix proxy

Der Proxy benötigt nur eine TCP-Verbindung zum Zabbix-Server. Dadurch lässt sich eine Firewall leichter umgehen, da Sie nur eine Firewall-Regel konfigurieren müssen.

Attention:
Der Zabbix-Proxy muss eine separate Datenbank verwenden. Wenn Sie ihn auf die Datenbank des Zabbix-Servers verweisen lassen, wird die Konfiguration beschädigt.

Alle vom Proxy erfassten Daten werden lokal gespeichert, bevor sie an den Server übertragen werden. Auf diese Weise gehen aufgrund vorübergehender Kommunikationsprobleme mit dem Server keine Daten verloren. Die Parameter `ProxyLocalBuffer` und `ProxyOfflineBuffer` in der Proxy-Konfigurationsdatei steuern, wie lange die Daten lokal gespeichert werden.

Attention:
Es kann vorkommen, dass ein Proxy, der die neuesten Konfigurationsänderungen direkt aus der Datenbank des Zabbix-Servers erhält, über eine aktuellere Konfiguration verfügt als der Zabbix-Server, dessen Konfiguration aufgrund des Werts von CacheUpdateFrequency möglicherweise nicht so schnell aktualisiert wird. Infolgedessen kann der Proxy mit der Datenerfassung beginnen und diese Daten an den Zabbix-Server senden, der die Daten ignoriert.

Der Zabbix-Proxy ist ein Datensammler. Er berechnet keine Auslöser, verarbeitet keine Ereignisse und sendet keine Warnungen. Eine Übersicht über die Funktionen des Proxys finden Sie in der folgenden Tabelle:

Funktion		Vom Proxy unterstützt
Datenpunkte		
	Zabbix-Agent-Prüfungen	**Ja**
	Zabbix-Agent-Prüfungen (aktiv)	**Ja** [1]

Funktion		Vom Proxy unterstützt
	Einfache Prüfungen	**Ja**
	Trapper-Datenpunkte	**Ja**
	SNMP-Prüfungen	**Ja**
	SNMP-Traps	**Ja**
	IPMI-Prüfungen	**Ja**
	JMX-Prüfungen	**Ja**
	Überwachung von Protokolldateien	**Ja**
	Interne Prüfungen	**Ja**
	SSH-Prüfungen	**Ja**
	Telnet-Prüfungen	**Ja**
	Externe Prüfungen	**Ja**
	Abhängige Datenpunkte	**Ja**
	Skript-Datenpunkte	**Ja**
	Browser-Datenpunkte	**Ja**
	Telemetrieabfrage-Datenpunkte	**Ja**
Integrierte Webüberwachung		**Ja**
Vorverarbeitung von Datenpunktwerten		**Ja**
Netzwerkerkennung		**Ja**
Automatische Registrierung aktiver Agents		**Ja**
Low-Level-Discovery		**Ja** [2]
Remote-Befehle		**Ja**
Erfassung von Telemetriedaten (APM)		**Ja** [3]
Berechnung von Auslösern		*Nein*
Verarbeitung von Ereignissen		*Nein*
Ereigniskorrelation		*Nein*
Senden von Warnungen		*Nein*

Note:
[1] Damit ein Agent den Proxy und nicht den Server nach aktiven Prüfungen fragt, muss der Proxy im Parameter **ServerActive** in der Agent-Konfigurationsdatei aufgeführt sein.
 [2] Bei LLD erfasst und verarbeitet der Zabbix-Proxy die Daten nur vor und sendet sie anschließend zur weiteren Verarbeitung an den Zabbix-Server.
 [3] Telemetriedaten werden nur erfasst, wenn APM auf dem Proxy aktiviert ist (siehe die Registerkarte *APM* in der Proxy-Konfiguration); der Zabbix-Server kann keine Telemetriedaten empfangen.

Proxy-Funktionen

Bei der Entscheidung, ob ein Proxy verwendet werden soll oder nicht, müssen mehrere Aspekte berücksichtigt werden.

	Proxy
Leichtgewichtig	**Ja**
GUI	Nein
Arbeitet unabhängig	**Ja**
Einfache Wartung	**Ja**
Automatische DB-Erstellung	**Ja**[1]
Lokale Administration	Nein
Für Embedded-Hardware geeignet	**Ja**
Einweg-TCP-Verbindungen	**Ja**
Zentralisierte Konfiguration	**Ja**
Erzeugt Benachrichtigungen	Nein

[1] Die Funktion zur automatischen DB-Erstellung funktioniert nur mit SQLite. Andere unterstützte Datenbanken erfordern ein manuelles Setup.

Attention:
Der Zabbix Proxy ist sich Wartungszeiträumen nicht bewusst; siehe Berechnung der Warteschlangen während der Wartung für Details.

Schutz vor Überlastung

Wenn der Zabbix Server für einige Zeit ausgefallen war und Proxys viele Daten gesammelt haben und der Server dann startet, kann es zu einer Überlastung kommen (die Auslastung des History-Cache bleibt für einige Zeit bei 95-100 %). Diese Überlastung kann zu Leistungseinbußen führen, bei denen Prüfungen langsamer verarbeitet werden, als sie sollten. Zum Schutz vor diesem Szenario wurde eine Funktion implementiert, um Probleme zu vermeiden, die durch die Überlastung des History-Cache entstehen.

Wenn der History-Cache des Zabbix Server voll ist, wird der Schreibzugriff auf den History-Cache gedrosselt, wodurch die Prozesse zur Datenerfassung des Servers blockiert werden. Der häufigste Fall einer Überlastung des History-Cache tritt nach einer Server-Ausfallzeit auf, wenn Proxys gesammelte Daten hochladen. Um dies zu vermeiden, wurde eine Proxy-Drosselung hinzugefügt (derzeit kann sie nicht deaktiviert werden).

Wenn die Auslastung des History-Cache 80 % erreicht, wechselt der Zabbix Server in den Drosselungsmodus. Im Drosselungsmodus akzeptiert der Server Proxy-Daten nur dann, wenn die Auslastung des History-Cache unter 60 % liegt, und rotiert die akzeptierten Proxys. Sobald die Auslastung des History-Cache unter 20 % fällt, wechselt der Server wieder in den Normalmodus.

Zusätzlich drosselt der Zabbix Server im Normalmodus einzelne Proxys, die sehr große Pakete (10.000+ Datensätze) senden, wenn die Auslastung des History-Cache 60 % überschreitet. Diese Entscheidung wird in dem Moment getroffen, in dem der Server einen Proxy-Upload auswertet, und kann daher in Diagrammen zur Auslastung des History-Cache nicht immer sofort sichtbar sein (der interne Datenpunkt `zabbix[wcache,history,pused]` und sein Aktualisierungsintervall können kurze Spitzen möglicherweise verpassen).

Dieser Drosselungsmodus bleibt aktiv, bis entweder die Cache-Auslastung erneut 80 % erreicht oder auf 20 % fällt oder die Drosselliste leer ist. Im ersten Fall akzeptiert der Server wieder keine Proxy-Daten. In den beiden anderen Fällen arbeitet der Server wieder normal und akzeptiert Daten von allen Proxys.

Die obigen Informationen lassen sich in der folgenden Tabelle veranschaulichen:

History write cache usage	Zabbix server mode	Zabbix server action
Erreicht 80 %	Warten	Akzeptiert keine Proxy-Daten mehr, behält aber eine *Drosselliste* bei (priorisierte Liste von Proxys, die später kontaktiert werden).
Erreicht 60 %	Normal, aber auf Drosselung vorbereitet	Kann sehr große Proxy-Uploads (mehr als 10.000 Datensätze) bei der Entscheidung über die Datenannahme ablehnen; akzeptiert weiterhin andere Proxy-Daten.
Fällt auf 20 %	Normal	Verwirft die Drosselliste und beginnt wieder, Proxy-Daten normal zu akzeptieren.

Sie können den internen Datenpunkt zabbix[wcache,history,pused] verwenden, um dieses Verhalten des Zabbix Server mit einer Metrik zu korrelieren.

Konfiguration

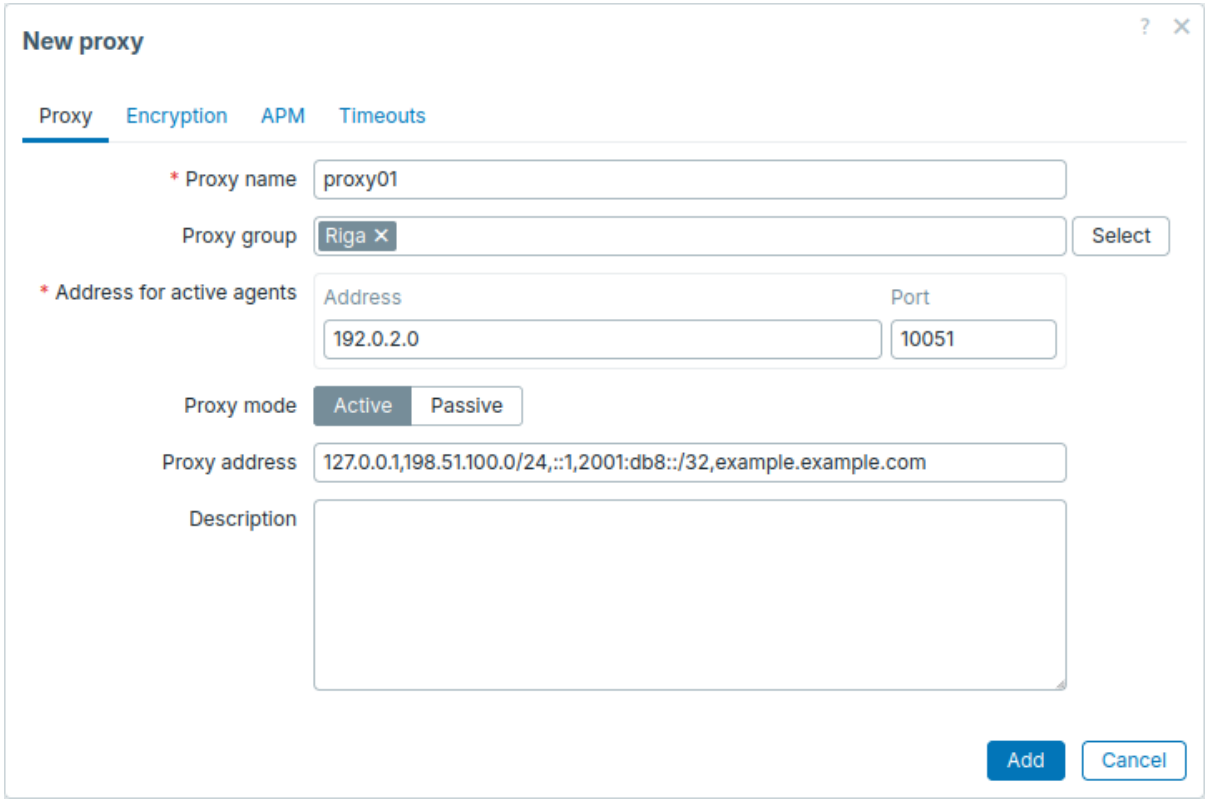
Nachdem Sie einen Proxy installiert und konfiguriert haben, ist es an der Zeit, ihn im Zabbix-Frontend zu konfigurieren.

Proxys hinzufügen

Note:
Anstatt einen Proxy in Ihrer eigenen Infrastruktur zu installieren, können Sie einen in Zabbix Cloud bereitstellen. Klicken Sie dazu unter *Administration* > *Proxies* auf *Create proxy in Cloud*; Zabbix Cloud wird in einem neuen Browser-Tab geöffnet. Nachdem Sie dort einen Proxy-Knoten erstellt haben, konfigurieren Sie den Proxy im Zabbix Frontend wie unten beschrieben. Weitere Informationen finden Sie unter Einen Zabbix Cloud-Proxy einrichten.

So konfigurieren Sie einen Proxy im Zabbix Frontend:

1. Gehen Sie zu *Administration* > *Proxies*.
2. Klicken Sie auf *Create proxy*.
3. Geben Sie die Proxy-Parameter im Formular ein.

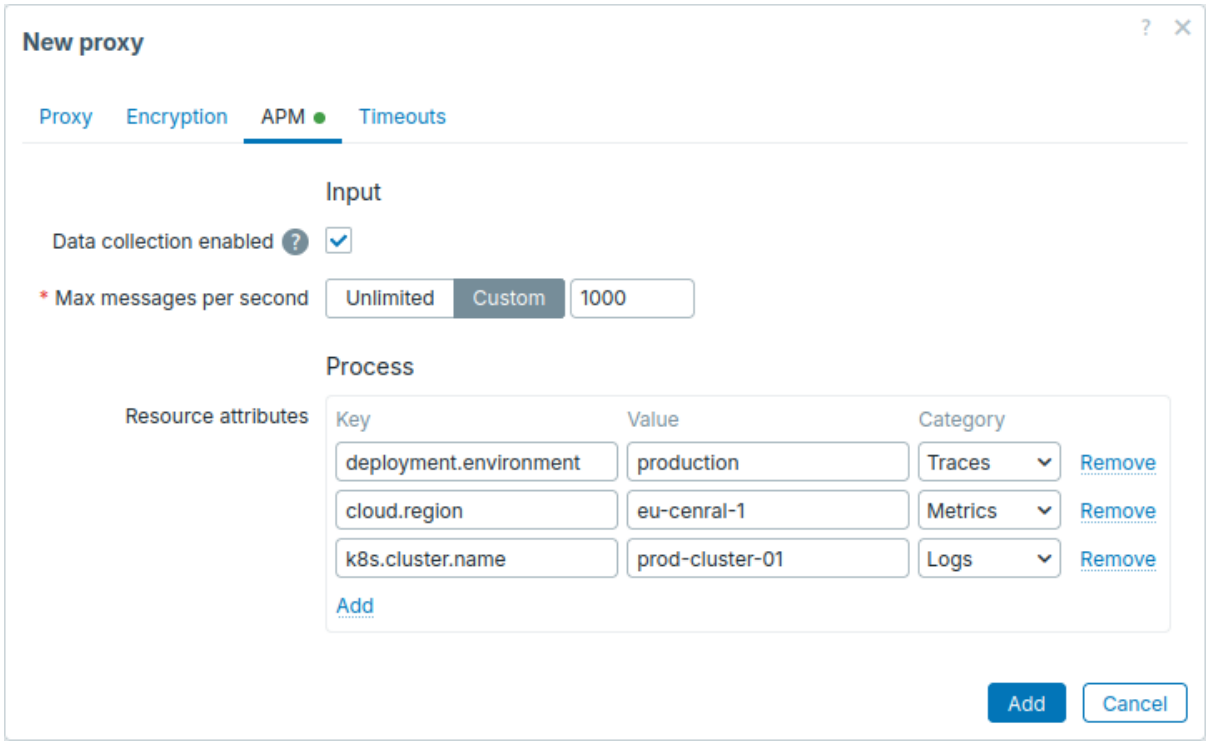


Parameter	Beschreibung
Proxy name	Geben Sie den Namen des Proxys ein. Er muss mit dem Namen im Parameter *Hostname* der Proxy-Konfigurationsdatei übereinstimmen.
Proxy group	Wählen Sie eine Proxy-Gruppe für den Proxy Lastenausgleich/hohe Verfügbarkeit aus.
Address for active agents	Geben Sie die Adresse ein, mit der sich die überwachten aktiven Agents oder Sender verbinden müssen. Unterstützt **nur** Zabbix 7.0 Agents oder höher. Diese Adresse wird verwendet, um sowohl aktive als auch passive Proxys zu verbinden. Dieses Feld ist nur verfügbar, wenn im Feld *Proxy group* eine Proxy-Gruppe ausgewählt wurde.
Address	IP-Adresse/DNS-Name, mit dem eine Verbindung hergestellt werden soll.
Port	TCP-Portnummer (standardmäßig 10051), mit dem eine Verbindung hergestellt werden soll. Benutzermakros werden unterstützt.
Proxy mode	Wählen Sie den Proxy-Modus aus. **Active** – Der Proxy verbindet sich mit dem Zabbix Server und fordert Konfigurationsdaten an **Passive** – Der Zabbix Server verbindet sich mit dem Proxy *Beachten Sie*, dass bei unverschlüsselter Kommunikation (sensibel) Proxy-Konfigurationsdaten für Parteien zugänglich werden können, die Zugriff auf den Trapper-Port des Zabbix Servers haben, wenn ein aktiver Proxy verwendet wird. Dies ist möglich, weil sich jeder als aktiver Proxy ausgeben und Konfigurationsdaten anfordern kann, wenn keine Authentifizierung erfolgt oder Proxy-Adressen im Feld *Proxy address* nicht eingeschränkt sind.
Proxy address	Wenn angegeben, werden Anfragen aktiver Proxys nur von dieser Liste kommaseparierter IP-Adressen, optional in CIDR-Notation, oder DNS-Namen aktiver Zabbix Proxys akzeptiert. Dieses Feld ist nur verfügbar, wenn im Feld *Proxy mode* ein aktiver Proxy ausgewählt wurde. Makros werden nicht unterstützt.
Interface	Geben Sie die Schnittstellendetails für einen passiven Proxy ein. Dieses Feld ist nur verfügbar, wenn im Feld *Proxy mode* ein passiver Proxy ausgewählt wurde.
Address	IP-Adresse/DNS-Name des passiven Proxys.
Port	TCP-Portnummer des passiven Proxys (standardmäßig 10051). Benutzermakros werden unterstützt.
Description	Geben Sie die Beschreibung des Proxys ein.

Auf der Registerkarte **Encryption** können Sie verschlüsselte Verbindungen mit dem Proxy voraussetzen.

Parameter	Beschreibung
Connections to proxy	Wie der Server eine Verbindung zum passiven Proxy herstellt: keine Verschlüsselung (Standard), über PSK (Pre-Shared Key) oder über ein Zertifikat.
Connections from proxy	Wählen Sie aus, welche Verbindungstypen vom aktiven Proxy zugelassen werden. Mehrere Verbindungstypen können gleichzeitig ausgewählt werden (nützlich zum Testen und zum Wechseln zu einem anderen Verbindungstyp). Standardmäßig ist „No encryption“ ausgewählt.
Issuer	Zulässiger Aussteller des Zertifikats. Das Zertifikat wird zunächst mit der CA (Zertifizierungsstelle) validiert. Wenn es gültig und von der CA signiert ist, kann das Feld *Issuer* verwendet werden, um die zulässige CA weiter einzuschränken. Dieses Feld ist optional und für den Fall vorgesehen, dass Ihre Zabbix-Installation Zertifikate von mehreren CAs verwendet.
Subject	Zulässiger Betreff des Zertifikats. Das Zertifikat wird zunächst mit der CA validiert. Wenn es gültig und von der CA signiert ist, kann das Feld *Subject* verwendet werden, um nur einen Wert der Zeichenfolge *Subject* zuzulassen. Wenn dieses Feld leer ist, wird jedes gültige, von der konfigurierten CA signierte Zertifikat akzeptiert.
PSK identity	Identitätszeichenfolge des Pre-Shared Keys. Geben Sie keine vertraulichen Informationen in die PSK-Identität ein, da sie unverschlüsselt über das Netzwerk übertragen wird, um einem Empfänger mitzuteilen, welcher PSK verwendet werden soll.
PSK	Pre-Shared Key (Hexadezimalzeichenfolge). Maximale Länge: 512 Hexadezimalstellen (256-Byte-PSK), wenn Zabbix die GnuTLS- oder OpenSSL-Bibliothek verwendet, bzw. 64 Hexadezimalstellen (32-Byte-PSK), wenn Zabbix die mbed-TLS-(PolarSSL-)Bibliothek verwendet. Beispiel: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952

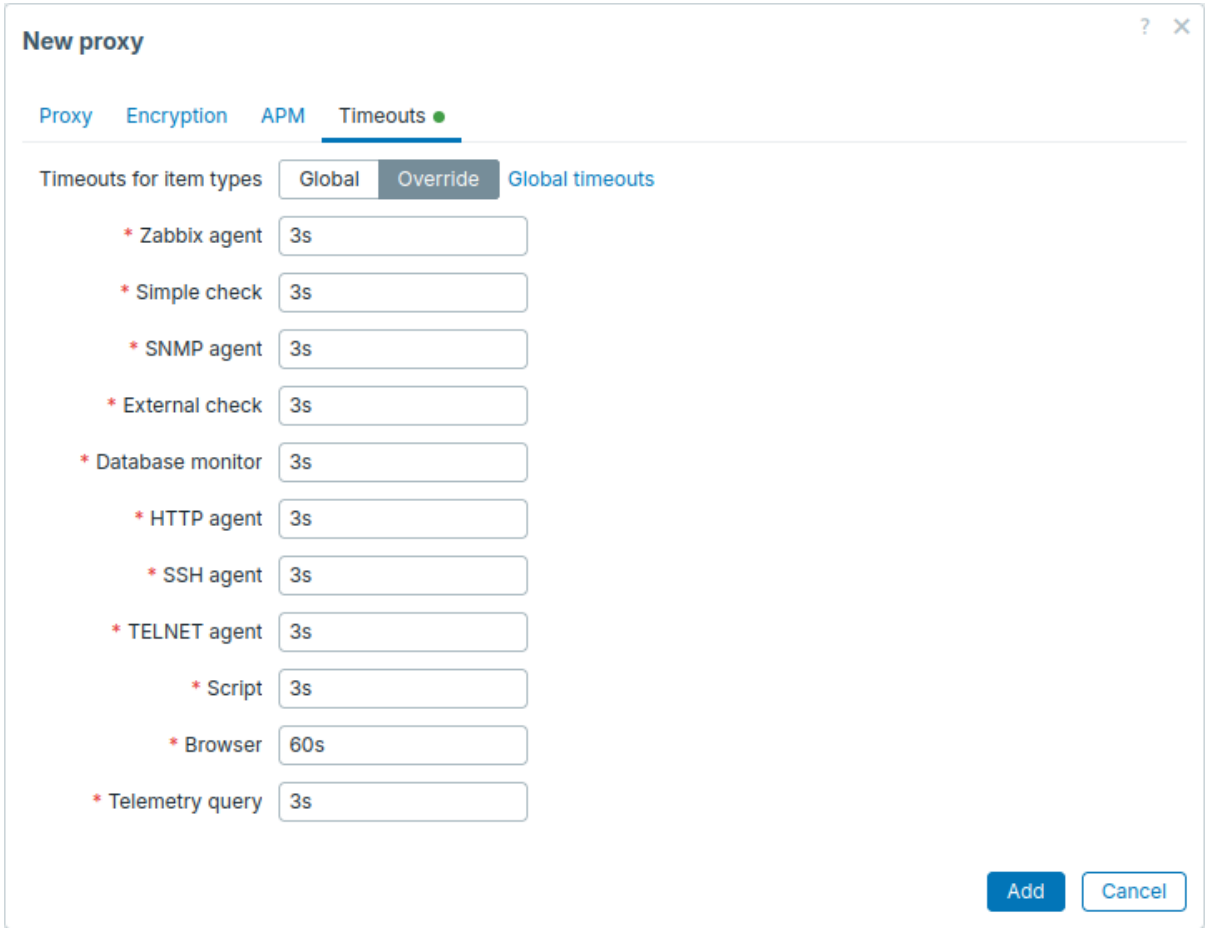
Auf der Registerkarte **APM** können Sie das Application Performance Monitoring (APM) auf dem Proxy konfigurieren. Der Proxy kann Telemetriedaten (Traces, Metriken und Logs) erfassen, die von Anwendungen oder OpenTelemetry Collectors über das OpenTelemetry-Protokoll (OTLP/gRPC) gesendet werden, und die Daten in die Telemetriedatenquelle schreiben.



Parameter	Beschreibung
Data collection enabled	Aktivieren Sie das Kontrollkästchen, um die Erfassung von Telemetriedaten auf dem Proxy zu aktivieren. Setzen Sie vor der Aktivierung der Telemetriedatenerfassung den Parameter `StartAPMCollector` in der Proxy-Konfigurationsdatei auf 1 (erfordert einen Zabbix Proxy mit APM-Unterstützung).

Parameter	Beschreibung
Max messages per second	Legen Sie die maximale Anzahl von Nachrichten pro Sekunde fest, die der Proxy akzeptiert: **Unlimited** – Alle eingehenden Nachrichten akzeptieren. **Custom** – Bis zur angegebenen Anzahl von Nachrichten pro Sekunde akzeptieren. Eine Nachricht ist eine von einer Anwendung, einem OpenTelemetry Collector oder einer anderen Telemetriequelle gesendete OTLP-Anfrage. Ihre Größe hängt vom Sender ab. Beispielsweise kann ein OpenTelemetry Collector Tausende von Spans in einer Nachricht bündeln, während eine einfache Anwendung kleinere Nachrichten senden kann (z. B. eine pro Trace). Wenn das Limit überschritten wird, antwortet der Proxy mit dem gRPC-Statuscode `UNAVAILABLE` (temporärer Fehler). Abhängig von seiner Konfiguration kann der Sender daher versuchen, die Nachricht später erneut zu senden.
Add resource attributes	Geben Sie die Ressourcenattribute an, die den eingehenden Telemetriedaten des ausgewählten Typs hinzugefügt werden sollen (z. B. um alle von diesem Proxy empfangenen Daten mit ihrer Umgebung oder Region zu kennzeichnen). Sie können diese Attribute anschließend zum Filtern von Telemetriedaten verwenden, beispielsweise in Bedingungen von Telemetrieabfragen. Wenn eingehende Daten bereits ein Ressourcenattribut mit demselben Namen enthalten, wird das eingehende Attribut anstelle des hier angegebenen beibehalten.
Name	Geben Sie den Attributnamen ein (z. B. `deployment.environment`, `cloud.region`). Er muss für den ausgewählten *Type* eindeutig sein.
Value	Geben Sie den Attributwert ein (z. B. `production`, `eu-central-1`). Kann leer sein.
Type	Wählen Sie den Typ der eingehenden Daten aus, für den das Attribut gilt: *Traces*, *Metrics* oder *Logs*.

Auf der Registerkarte **Timeouts** können Sie globale Timeouts für Datenpunkttypen überschreiben, die dies unterstützen.



Parameter	Beschreibung
Timeouts for item types	Legen Sie das Datenpunkt-Timeout (abhängig von seinem Typ) fest: **Global** – Globales Timeout verwenden (wird im ausgegrauten Feld *Timeout* für jeden Datenpunkttyp angezeigt); **Override** – Benutzerdefiniertes Timeout festlegen (im Feld *Timeout* für jeden Datenpunkttyp). Zulässiger Bereich: 1–600 s (Standard: von den globalen Timeouts übernommen). Zeit-Suffixe, z. B. 30s und 1m, sowie Benutzermakros werden unterstützt. Über den Link *Global timeouts* können Sie globale Timeouts konfigurieren. Beachten Sie, dass der Link *Global timeouts* nur Benutzern vom Typ *Super admin* angezeigt wird, die über Berechtigungen für den Frontend-Bereich *Administration* > *General* verfügen. Beachten Sie, dass Timeouts auf Proxy-Ebene zwar die globalen Timeouts überschreiben, jedoch durch individuelle Datenpunkt-Timeouts überschrieben werden, sofern diese konfiguriert sind.

Note:

Wenn die Hauptversion des Proxys nicht mit der Hauptversion des Servers übereinstimmt, wird das -Symbol neben *Timeouts for item types* angezeigt. Beim Darüberfahren wird die Meldung „Timeouts disabled because the proxy and server versions do not match“ eingeblendet. In solchen Fällen verwendet der Proxy den Parameter `Timeout` aus der Proxy-Konfigurationsdatei.

Das Bearbeitungsformular eines vorhandenen Proxys enthält die folgenden zusätzlichen Schaltflächen:

- *Refresh configuration* – Konfiguration des Proxys aktualisieren;
- *Clone* – Einen neuen Proxy auf Grundlage der Eigenschaften des vorhandenen Proxys erstellen;
- *Delete* – Den Proxy löschen.

Host-Konfiguration

Sie können im Formular Host-Konfiguration über das Feld *Überwacht von* festlegen, dass ein einzelner Host von einem Proxy oder einer Proxy-Gruppe überwacht werden soll.



Die Massenaktualisierung von Hosts ist eine weitere Möglichkeit, festzulegen, dass Hosts von einem Proxy oder einer Proxy-Gruppe überwacht werden sollen.

Proxy-Lastverteilung und Hochverfügbarkeit

Übersicht

Zabbix-Proxys können in Proxy-Gruppen organisiert werden, um Proxy-Lastverteilung und Hochverfügbarkeit zu ermöglichen.

Proxy-Lastverteilung und Hochverfügbarkeit bezeichnen die automatische Umverteilung von Hosts zwischen Proxys innerhalb einer Proxy-Gruppe:

- Wenn ein Proxy offline geht, werden seine Hosts auf andere Proxys verschoben, wodurch eine hohe Proxy-Verfügbarkeit aufrechterhalten wird.
- Wenn ein Proxy deutlich mehr oder weniger Hosts als andere Proxys hat, werden seine Hosts auf andere Proxys verschoben, um die Proxy-Last auszugleichen.

Die Umverteilung von Hosts funktioniert nur zwischen Proxys in einer Gruppe, die die folgenden Bedingungen erfüllen:

- Die Proxys verwenden Zabbix 7.0 oder höher.
- Die Proxy-Version stimmt mit der Zabbix-Server-Version überein. Bei Verwendung von Zabbix Agent (passiv) muss die Proxy-Version mit der Agent-Version übereinstimmen. Aktive Agents benötigen lediglich Zabbix 7.0 oder höher.
- Die Proxy-Gruppe hat einen Online-Status.
- Die Hosts sind so konfiguriert, dass sie von einer Proxy-Gruppe und nicht von einzelnen Proxys überwacht werden.

Note:
Der Zustand einer Proxy-Gruppe kann mit internen Prüfungen von jedem Host überwacht werden, der einer Proxy-Gruppe zugewiesen ist. Um jedoch den Zustand eines einzelnen Proxys in einer Gruppe zu überwachen, weisen Sie den Host diesem Proxy zu. Andernfalls können die Ergebnisse inkonsistent sein.

Host-Umverteilung

Das Load Balancing und die Hochverfügbarkeit von Proxys werden vom Zabbix Server über den Proxy-Gruppen-Manager verwaltet, der kontinuierlich den Status aller Proxys in jeder Proxy-Gruppe sowie deren Host-Verteilung überwacht.

Die Hochverfügbarkeit von Proxys innerhalb einer Gruppe wird durch Proxy-Failover sichergestellt: Wenn ein Proxy offline geht, werden seine Hosts sofort auf andere Proxys umverteilt. Auch das Load Balancing der Proxys findet statt, da Hosts den Proxys mit der geringsten Anzahl zugewiesener Hosts neu zugewiesen werden.

Zusätzlich wird das Load Balancing der Proxys ausgelöst, wenn sich die Anzahl der Hosts eines Proxys vom Gruppendurchschnitt um mindestens 10 Hosts und um den Faktor 2 unterscheidet (Host-Überschuss oder Host-Defizit). Bleibt das Ungleichgewicht nach einer Schonfrist bestehen (10 x Failover-Verzögerung), wird die Proxy-Gruppe zur Host-Umverteilung in die Warteschlange eingereiht.

Der Proxy-Gruppen-Manager verteilt Hosts nach der folgenden Logik um:

1. Berechnen Sie die durchschnittliche Anzahl von Hosts pro Proxy.
2. Bei Proxys mit Host-Überschuss: Verschieben Sie die überschüssigen Hosts in den Pool der nicht zugewiesenen Proxys.
3. Bei Proxys mit Host-Defizit: Berechnen Sie, wie viele Hosts erforderlich sind, um den Ausgleich zu erreichen.
4. Entfernen Sie die erforderliche Anzahl von Hosts von den Proxys mit den meisten Hosts.
5. Verschieben Sie nicht zugewiesene Hosts auf die Proxys mit den wenigsten Hosts.

Beispiele für die Host-Umverteilung:

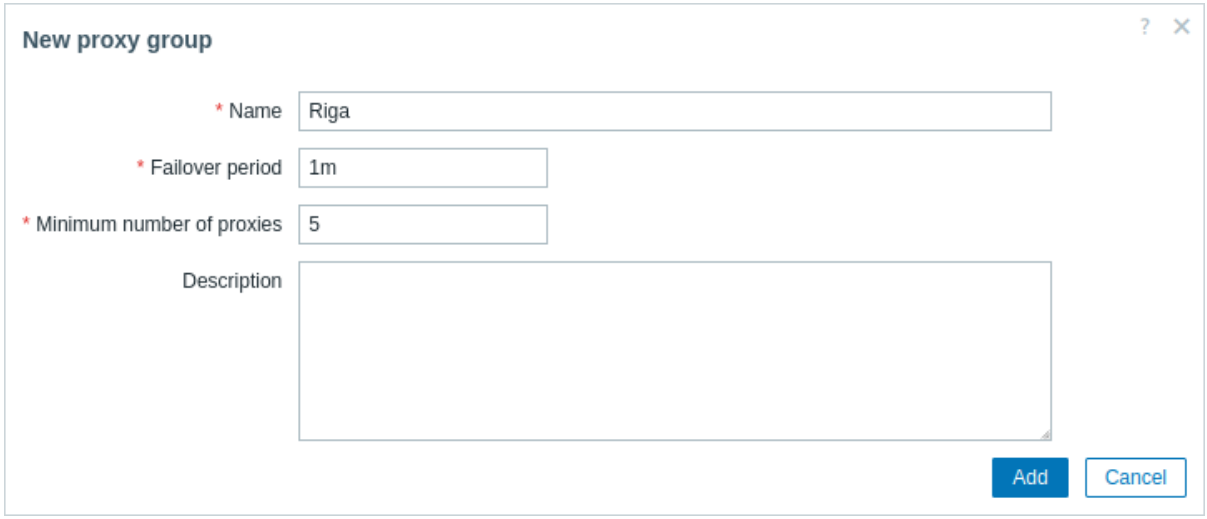
Hosts auf Proxy	Gruppendurchschnitt	Host-Neuzuweisung
100	50	Ja
60	50	Nein
40	50	Nein
25	50	Ja
15	5	Ja
10	5	Nein

Attention:
Wenn eine Proxy-Gruppe von weniger als 10 überwachten Hosts umfasst, kann dies zu einer ungleichmäßigen Host-Verteilung zwischen den Proxys in der Gruppe führen.

Konfigurieren einer Proxy-Gruppe

So konfigurieren Sie eine Proxy-Gruppe im Zabbix-Frontend:

1. Gehen Sie zu *Administration > Proxy groups*.
2. Klicken Sie auf *Create proxy group*.
3. Geben Sie die Parameter der Proxy-Gruppe im Formular ein.



Parameter	Beschreibung
Name	Name der Proxy-Gruppe.
Failover period	Zeitraum in Sekunden, während dessen ein Proxy in der Proxy-Gruppe mit dem Zabbix-Server kommunizieren muss, um als online betrachtet zu werden (Standard: 1m; Bereich: 10s-15m). Wenn der Proxy innerhalb dieses Zeitraums nicht kommuniziert, wird der Proxy-Status auf *Offline* geändert und seine Hosts werden sofort auf andere Proxys verteilt. Der Proxy-Lastenausgleich beginnt nach dem 10-Fachen dieses Zeitraums. Unterstützt Zeitsuffixe (z. B. 30s, 1m) und Benutzermakros.
Minimum number of proxies	Mindestanzahl online verfügbarer Proxys, die erforderlich ist, damit die Proxy-Gruppe online bleibt (Standard: 1; Bereich: 1-1000). Unterstützt Benutzermakros. Dieser Wert sollte kleiner sein als die Gesamtanzahl der Proxys in der Gruppe. Wenn beispielsweise eine Gruppe 10 Proxys enthält und der Mindestwert auf 10 gesetzt wird, wird die Gruppe offline geschaltet, sobald ein Proxy ausfällt. Beachten Sie, dass online verfügbare Proxys in einer Offline-Gruppe weiterhin normal funktionieren, der Lastenausgleich bzw. die Hochverfügbarkeit jedoch nicht aktiv ist.
Description	Beschreibung der Proxy-Gruppe.
Proxies	Zeigt beim Bearbeiten einer Gruppe mit Proxys eine Liste mit bis zu fünf Proxys an (als Links oder als Klartext, abhängig von den Benutzerberechtigungen für Proxys).

Konfigurieren des Proxy-Lastenausgleichs

Um den Proxy-Lastenausgleich zu verwenden, müssen Sie eine Proxy-Gruppe im Zabbix Frontend konfigurieren (siehe oben) und sicherstellen, dass Hosts von einer Proxy-Gruppe und nicht von einzelnen Proxys überwacht werden (Sie können die Massenaktualisierung von Hosts verwenden, um Hosts von Proxys in die Proxy-Gruppe zu verschieben).

Wenn Sie den Zabbix Agent verwenden, konfigurieren Sie ihn außerdem wie folgt:

- Für passive Prüfungen listen Sie alle Proxys der Proxy-Gruppe im Parameter Server auf.
- Für aktive Prüfungen wird empfohlen, alle Proxys der Proxy-Gruppe oder den Zabbix Server im Parameter ServerActive aufzulisten. Beachten Sie, dass nur der Zabbix Agent 7.0 (oder höher) Proxy-Gruppen im aktiven Modus unterstützt.

Wenn der Parameter `ServerActive` nur einen Proxy aus einer Proxy-Gruppe (oder den Zabbix Server) enthält, kann der Agent trotzdem eine Verbindung zum richtigen Proxy herstellen. Wenn der Agent-Dienst gestartet wird und eine Verbindung zum angegebenen Proxy herstellt, empfängt und speichert der Agent die vollständige Liste der Proxy-IP-Adressen sowie deren aktuelle Auslastung innerhalb der Gruppe. Anschließend werden aktive Prüfungen basierend auf der aktuellen Proxy-Host-Zuweisung innerhalb der Proxy-Gruppe an den richtigen verfügbaren Proxy weitergeleitet.

Warning:
Wenn im Zabbix-Agent-Parameter `ServerActive` nur ein einzelner Proxy angegeben ist, kann dies zum Verlust von Überwachungsdaten führen, wenn der Agent gestartet oder neu gestartet wird, während der angegebene Proxy offline ist.

Note:
Bei Verwendung von Zabbix sender werden Datenanforderungen ebenfalls basierend auf der aktuellen Proxy-Host-Zuweisung innerhalb der Proxy-Gruppe an den richtigen verfügbaren Proxy weitergeleitet. Wenn Sie jedoch Werte mehrerer Hosts aus einer Eingabedatei senden, verwenden Sie die –g-Option, um zu verhindern, dass Daten an den falschen Proxy gesendet werden.

Der Zabbix Agent muss außerdem über die Firewall eine Verbindung zu allen Proxys in der Proxy-Gruppe herstellen können. Andernfalls können aktive Prüfungen während der Weiterleitung oder des Failovers hängen bleiben oder fehlschlagen. Beispiele:

- Während aktiver Prüfungen kann ein Proxy den Agent an einen anderen Proxy weiterleiten. Wenn dieser Proxy durch eine Firewall blockiert wird, bleibt die Kommunikation beim Warten auf eine Antwort hängen.
- In stabilen Hochverfügbarkeitsumgebungen ohne kürzlich erfolgte Neuverteilung kontaktieren Agents möglicherweise niemals Backup-Proxys. Wenn Firewall-Regeln geändert und nicht getestet wurden, kann das Failover fehlschlagen.

Testen des Proxy-Lastenausgleichs

So testen Sie den Proxy-Lastenausgleich:

1. Konfigurieren Sie eine Proxy-Gruppe.
2. Stellen Sie sicher, dass sich die Proxy-Gruppe in einem Online-Zustand befindet.

3. Stellen Sie sicher, dass Hosts von einer Proxy-Gruppe überwacht werden und nicht von einzelnen Proxys (Sie können die Massenaktualisierung von Hosts verwenden, um Hosts von Proxys in die Proxy-Gruppe zu verschieben).
4. Warten Sie einige Sekunden, bis die Konfiguration aktualisiert und die Hosts auf die Proxys in der Proxy-Gruppe verteilt wurden.

Beobachten Sie die Änderung, indem Sie die Host-Liste unter *Administration* > *Proxys* aktualisieren.

Wichtige Hinweise

- SNMP-Traps werden von Proxys in einer Proxy-Gruppe nicht unterstützt.
- Prüfungen, die von einer externen Konfiguration abhängen (z. B. Skripte für externe Prüfungen oder die ODBC-Konfiguration für Datenbankprüfungen), müssen auf allen Proxys in der Proxy-Gruppe dieselbe Konfiguration aufweisen.
- Datenbankprüfungen erfordern erweiterte Berechtigungen für das Datenbankobjekt bzw. den Datenbankserver.
- VMware-Hosts, die von einer Proxy-Gruppe überwacht werden, werden zufällig auf die Proxys in der Gruppe verteilt. Dies führt dazu, dass jeder Proxy alle VMware-Daten zwischenspeichert, wodurch die Last auf vCenter erhöht wird.
- Hosts, die auf Grundlage von Autoregistrierungsdaten eines Proxys in einer Proxy-Gruppe erstellt werden, werden so eingestellt, dass sie von dieser Proxy-Gruppe überwacht werden. Hosts, die jedoch auf Grundlage von Netzwerkermittlungsdaten eines Proxys in einer Proxy-Gruppe erstellt werden, werden so eingestellt, dass sie von diesem Proxy überwacht werden.

Synchronisierung der Überwachungskonfiguration

Übersicht

Diese Seite enthält Details zur Aktualisierung der Überwachungskonfiguration für den Proxy, d. h. wie Änderungen an der Überwachungskonfiguration auf dem Server mit dem Proxy synchronisiert werden.

Inkrementelle Aktualisierung

Die Aktualisierung der Proxy-Konfiguration erfolgt inkrementell.
Während einer Konfigurationssynchronisierung werden nur die geänderten Entitäten aktualisiert (wenn also keine Entitäten geändert wurden, wird nichts gesendet).
Dieser Ansatz ermöglicht es Ihnen, Ressourcen zu sparen und ein kleineres Intervall (nahezu sofort) für die Aktualisierung der Proxy-Konfiguration festzulegen.

Änderungen an der Proxy-Konfiguration werden mithilfe von Revisionsnummern nachverfolgt. Nur Entitäten mit Revisionen, die größer als die Revisionsnummer der Proxy-Konfiguration sind, werden in die an den Proxy gesendeten Konfigurationsdaten aufgenommen.

Die Entitäten für eine Konfigurationssynchronisierung sind:

Entität	Details
TLS-Daten für die automatische Registrierung	Alle TLS-Daten für die automatische Registrierung.
Ausdrücke	Alle Ausdrücke (reguläre Ausdrücke, Ausdruckstabellen).
globale Konfiguration	Globale Konfiguration, die in der Tabelle „config“ definiert ist
Host	Alle Eigenschaften, Schnittstellen, Inventardaten, Datenpunkte, Datenpunkt-Vorverarbeitung, Datenpunktparameter und Web-Szenarien eines Hosts.
Host-Makros	Alle auf einem Host definierten Makros sowie alle mit ihm verknüpften Vorlagen-IDs.
Proxy-Erkennungsregel	Erkennungsregeln und Prüfungen, die einem Proxy zugewiesen sind.

Das bedeutet:

- Wenn ein Datenpunkt auf einem **Host** geändert wird, wird die gesamte Konfiguration dieses Hosts synchronisiert.
- Wenn ein **regulärer Ausdruck** geändert wird, werden alle regulären Ausdrücke synchronisiert.

Eine Ausnahme bilden die Host-Makros, die ebenfalls gesendet werden, wenn irgendetwas auf dem Host geändert wurde.

Der Befehl `-R config_cache_reload` auf dem Proxy initiiert ebenfalls eine inkrementelle Aktualisierung.

Beachten Sie, dass beim Start/Neustart eines Proxys, bei einem HA-Failover, wenn sich das Sitzungstoken geändert hat oder wenn die Konfigurationsaktualisierung auf dem Proxy fehlgeschlagen ist, eine vollständige Konfigurationssynchronisierung durchgeführt wird, beispielsweise wenn die Verbindung während des Empfangs von Konfigurationsdaten unterbrochen wurde.

Konfigurationsparameter

Der Parameter **ProxyConfigFrequency** bestimmt, wie oft die Proxy-Konfiguration mit dem Server synchronisiert wird (standardmäßig alle 10 Sekunden).

Beachten Sie, dass ProxyConfigFrequency Folgendes ist:

- Server-Parameter für passive Proxies
- Proxy-Parameter für aktive Proxies

Bei aktiven Proxies ist ProxyConfigFrequency ein neuer Parameter seit Zabbix 6.4 und muss anstelle des inzwischen veralteten ConfigFrequency verwendet werden.

Attention:
Wenn sowohl ProxyConfigFrequency als auch ConfigFrequency verwendet werden, protokolliert der Proxy einen Fehler und beendet sich.

Berechnung von Warteschlangen während der Wartung

Attention:
Der Zabbix Proxy ist sich von Wartungszeiträumen nicht bewusst; siehe Berechnung von Warteschlangen während der Wartung für Details.

Server-Proxy-Datenaustauschprotokoll

Server - der Datenaustausch mit dem Proxy basiert auf dem JSON-Format.

Anfrage- und Antwortnachrichten müssen mit Header und Datenlänge beginnen.

Passiver Proxy

Konfigurationsanfrage

Der Server sendet zunächst eine leere `proxy config`-Anforderung. Diese Anforderung wird alle `ProxyConfigFrequency` (Server-Konfigurationsparameter) Sekunden gesendet.

Der Proxy antwortet mit der aktuellen Proxy-Version, dem Sitzungstoken und der Konfigurationsrevision. Der Server antwortet mit den Konfigurationsdaten, die aktualisiert werden müssen.

name	value type	description
server→proxy:		
request	*string*	'proxy config'
proxy→server:		
version	*string*	Proxy-Version (<major>.<minor>.<build>).
session	*string*	Sitzungstoken der Proxy-Konfiguration.
config_revision	*number*	Revision der Proxy-Konfiguration.
server→proxy:		
full_sync	*number*	1 - wenn vollständige Konfigurationsdaten gesendet werden; andernfalls nicht vorhanden (optional).
data	*array*	Objekt mit Tabellendaten. Nicht vorhanden, wenn die Konfiguration nicht geändert wurde (optional).
<table>	*object*	Ein oder mehrere Objekte mit <table>-Daten (optional, abhängig von den Änderungen).
fields	*array*	Array mit Feldnamen.
-	*string*	Feldname.
data	*array*	Array mit Zeilen.
-	*array*	Array mit Spalten.
-	*string*,*number*	Spaltenwert mit einem Typ, der vom Spaltentyp im Datenbankschema abhängt.
macro.secrets	*object*	Informationen zu geheimen Makros, nicht vorhanden, wenn es keine Änderungen an Vault-Makros gibt (optional).
config_revision	*number*	Revision des Konfigurations-Caches - wird mit den Konfigurationsdaten gesendet (optional).
del_hostids	*array*	Array der entfernten Hostids (optional).

name	value type	description
-	*number*	Host-Kennung.
del_macro_hostids	*array*	Array der Hostids, bei denen alle Makros entfernt wurden (optional).
-	*number*	Host-Kennung.
proxy→server:		
response	*string*	Information zum Erfolg der Anforderung ('success' oder 'failed').
version	*string*	Proxy-Version (<major>.<minor>.<build>).

Beispiel:

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", "ip
            "data": [
                [10084, "Zabbix server", 0, -1, 2, "", "", "Zabbix server", 1, 1, "", "", "", ""]
            ]
        },
        "interface": {
            "fields": ["interfaceid", "hostid", "main", "type", "useip", "ip", "dns", "port", "available"]
            "data": [
                [1, 10084, 1, 1, 1, "127.0.0.1", "", "10053", 1]
            ]
        },
        "interface_snmp": {
            "fields": ["interfaceid", "version", "bulk", "community", "securityname", "securitylevel", "au
            "data": []
        },
        "host_inventory": {
            "fields": ["hostid", "type", "type_full", "name", "alias", "os", "os_full", "os_short", "seria
            "data": [
                [10084, "", "", ..., "56.95387", "24.22067", ..., ""]
            ]
        },
        "items": {
            "fields": ["itemid", "type", "snmp_oid", "hostid", "key_", "delay", "history", "status", "valu
            "data": [
                [44161, 7, "", 10084, "agent.hostmetadata", "10s", "90d", 0, 1, "", "", "", "", 0, "", "",
                [44162, 0, "", 10084, "agent.ping", "10s", "90d", 0, 3, "", "", "", "", 0, "", "", "", "",
            ]
        },
        "item_rtdata": {
            "fields": ["itemid", "lastlogsize", "mtime"],
            "data": [
                [44161, 0, 0],
```

```
            [44162, 0, 0]
        ]
    },
    "item_preproc": {
        "fields": ["item_preprocid", "itemid", "step", "type", "params", "error_handler", "error_handl
        "data": []
    },
    "item_parameter": {
        "fields": ["item_parameterid", "itemid", "name", "value"],
        "data": []
    },
    "globalmacro": {
        "fields": ["globalmacroid", "macro", "value", "type"],
        "data": [
            [2, "{$SNMP_COMMUNITY}", "public", 0]
        ]
    },
    "hosts_templates": {
        "fields": ["hosttemplateid", "hostid", "templateid", "link_type"],
        "data": []
    },
    "hostmacro": {
        "fields": ["hostmacroid", "hostid", "macro", "value", "type", "automatic"],
        "data": [
            [5676, 10084, "{$M}", "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix:Content",
        ]
    },
    "drules": {
        "fields": ["druleid", "name", "iprange", "delay"],
        "data": [
            [2, "Local network", "127.0.0.1", "10s"]
        ]
    },
    "dchecks": {
        "fields": ["dcheckid", "druleid", "type", "key_", "snmp_community", "ports", "snmpv3_securityn
        "data": [
            [2, 2, 9, "system.uname", "", "10052", "", 0, "", "", 0, 0, 0, "", 1, 0]
        ]
    },
    "regexps": {
        "fields": ["regexpid", "name"],
        "data": [
            [1, "File systems for discovery"],
            [2, "Network interfaces for discovery"],
            [3, "Storage devices for SNMP discovery"],
            [4, "Windows service names for discovery"],
            [5, "Windows service startup states for discovery"]
        ]
    },
    "expressions": {
        "fields": ["expressionid", "regexpid", "expression", "expression_type", "exp_delimiter", "case
        "data": [
            [1, 1, "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|z
            [3, 3, "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$", 4, ",
            [5, 4, "^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.303
            [6, 5, "^(automatic|automatic delayed)$", 3, ",", 1],
            [7, 2, "^Software Loopback Interface", 4, ",", 1],
            [8, 2, "^(In)?[Ll]oop[Bb]ack[0-9._]*$", 4, ",", 1],
            [9, 2, "^NULL[0-9.]*$", 4, ",", 1],
            [10, 2, "^[Ll]o[0-9.]*$", 4, ",", 1],
            [11, 2, "^[Ss]ystem$", 4, ",", 1],
            [12, 2, "^Nu[0-9.]*$", 4, ",", 1]
```

```
            ]
        },
        "settings": {
            "fields": ["name", "type", "value_str", "value_int"],
            "data": [
                ["autoreg_tls_accept", 2, "", 1],
                ["hk_history_global", 2, "", 0],
                ["snmptrap_logging", 2, "", 1],
                ["proxy_secrets_provider", 2, "", 0],
                ["hk_history", 1, "31d", 0],
                ["timeout_db_monitor", 1, "3s", 0],
                ["timeout_external_check", 1, "3s", 0],
                ["timeout_http_agent", 1, "3s", 0],
                ["timeout_simple_check", 1, "3s", 0],
                ["timeout_snmp_agent", 1, "3s", 0],
                ["timeout_ssh_agent", 1, "3s", 0],
                ["timeout_telnet_agent", 1, "3s", 0],
                ["timeout_zabbix_agent", 1, "3s", 0],
                ["timeout_browser", 1, "30s", 0]
            ]
        },
        "httptest": {
            "fields": ["httptestid", "name", "delay", "agent", "authentication", "http_user", "http_passwo
            "data": []
        },
        "httptestitem": {
            "fields": ["httptestitemid", "httptestid", "itemid", "type"],
            "data": []
        },
        "httptest_field": {
            "fields": ["httptest_fieldid", "httptestid", "type", "name", "value"],
            "data": []
        },
        "httpstep": {
            "fields": ["httpstepid", "httptestid", "name", "no", "url", "timeout", "posts", "required", "s
            "data": []
        },
        "httpstepitem": {
            "fields": ["httpstepitemid", "httpstepid", "itemid", "type"],
            "data": []
        },
        "httpstep_field": {
            "fields": ["httpstep_fieldid", "httpstepid", "type", "name", "value"],
            "data": []
        },
        "config_autoreg_tls": {
            "fields": ["autoreg_tlsid", "tls_psk_identity", "tls_psk"],
            "data": [
                [1, "", ""]
            ]
        }
    },
    "macro.secrets": {
        "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix": {
            "Content": "738"
        }
    },
    "config_revision": 2
}
```


Proxy→Server:

```
{
  "response": "success",
  "version": "8.0.0"
}
```

Datenanforderung

Die Anforderung `proxy data` wird verwendet, um die Verfügbarkeit von Host-Schnittstellen sowie historische, Discovery- und Autoregistrierungsdaten vom Proxy abzurufen. Diese Anforderung wird alle `ProxyDataFrequency` Sekunden gesendet (Server-Konfigurationsparameter).

name	value type	description
server→proxy:		
request	*string*	'proxy data'
proxy→server:		
session	*string*	Daten-Sitzungstoken.
interface availability	*array*	*(optional)* Array von Objekten mit Schnittstellenverfügbarkeitsdaten.
interfaceid	*number*	Schnittstellen-ID.
available	*number*	Schnittstellenverfügbarkeit: **0**, *INTERFACE_AVAILABLE_UNKNOWN* - unbekannt **1**, *INTERFACE_AVAILABLE_TRUE* - verfügbar **2**, *INTERFACE_AVAILABLE_FALSE* - nicht verfügbar
error	*string*	Fehlermeldung der Schnittstelle oder leerer String.
history data	*array*	*(optional)* Array von Objekten mit Verlaufsdaten.
itemid	*number*	Datenpunkt-ID.
clock	*number*	Zeitstempel des Datenpunktwerts (Sekunden).
ns	*number*	Zeitstempel des Datenpunktwerts (Nanosekunden).
value	*string*	*(optional)* Datenpunktwert.
id	*number*	Wert-ID (aufsteigender Zähler, innerhalb einer Daten-Sitzung eindeutig).
timestamp	*number*	*(optional)* Zeitstempel von Datenpunkten des Log-Typs.
source	*string*	*(optional)* Quellwert des Eventlog-Datenpunkts.
severity	*number*	*(optional)* Schweregradwert des Eventlog-Datenpunkts.
eventid	*number*	*(optional)* Event-ID-Wert des Eventlog-Datenpunkts.
state	*string*	*(optional)* Datenpunktstatus: **0**, *ITEM_STATE_NORMAL* **1**, *ITEM_STATE_NOTSUPPORTED*
lastlogsize	*number*	*(optional)* Letzte Loggröße von Datenpunkten des Log-Typs.
mtime	*number*	*(optional)* Änderungszeit von Datenpunkten des Log-Typs.
discovery data	*array*	*(optional)* Array von Discovery-Datenobjekten.
clock	*number*	Zeitstempel der Discovery-Daten.
druleid	*number*	ID der Discovery-Regel.
dcheckid	*number*	ID der Discovery-Prüfung oder null für Discovery-Regel-Daten.

name	value type	description
type	*number*	Typ der Discovery-Prüfung: **-1** Discovery-Regel-Daten **0**, *SVC_SSH* - SSH-Dienstprüfung **1**, *SVC_LDAP* - LDAP-Dienstprüfung **2**, *SVC_SMTP* - SMTP-Dienstprüfung **3**, *SVC_FTP* - FTP-Dienstprüfung **4**, *SVC_HTTP* - HTTP-Dienstprüfung **5**, *SVC_POP* - POP-Dienstprüfung **6**, *SVC_NNTP* - NNTP-Dienstprüfung **7**, *SVC_IMAP* - IMAP-Dienstprüfung **8**, *SVC_TCP* - Prüfung der TCP-Port-Verfügbarkeit **9**, *SVC_AGENT* - Zabbix Agent **10**, *SVC_SNMPv1* - SNMPv1 Agent **11**, *SVC_SNMPv2* - SNMPv2 Agent **12**, *SVC_ICMPPING* - ICMP-Ping **13**, *SVC_SNMPv3* - SNMPv3 Agent **14**, *SVC_HTTPS* - HTTPS-Dienstprüfung **15**, *SVC_TELNET* - Telnet-Verfügbarkeitsprüfung
ip	*string*	IP-Adresse des Hosts.
dns	*string*	DNS-Name des Hosts.
port	*number*	*(optional)* Portnummer des Dienstes.
key_	*string*	*(optional)* Datenpunkt-Schlüssel für Discovery-Prüfung vom Typ **9** *SVC_AGENT*
value	*string*	*(optional)* Vom Dienst empfangener Wert, kann bei den meisten Diensten leer sein.
status	*number*	*(optional)* Dienststatus: **0**, *DOBJECT_STATUS_UP* - Dienst UP **1**, *DOBJECT_STATUS_DOWN* - Dienst DOWN
auto reg- is- tra- tion	*array*	*(optional)* Array von Autoregistrierungsdatenobjekten.
clock	*number*	Zeitstempel der Autoregistrierungsdaten.
host	*string*	Host-Name.
ip	*string*	*(optional)* IP-Adresse des Hosts.
dns	*string*	*(optional)* Aus der IP-Adresse aufgelöster DNS-Name.
port	*string*	*(optional)* Host-Port.
host_metadata	*string*	*(optional)* Vom Agent gesendete Host-Metadaten (basierend auf dem Agent-Konfigurationsparameter HostMetadata oder HostMetadataItem).
tasks	*array*	*(optional)* Array von Aufgaben.
type	*number*	Aufgabentyp: **0**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND_RESULT* - Ergebnis eines Remote-Befehls
status	*number*	Ausführungsstatus des Remote-Befehls: **0**, *ZBX_TM_REMOTE_COMMAND_COMPLETED* - Remote-Befehl erfolgreich abgeschlossen **1**, *ZBX_TM_REMOTE_COMMAND_FAILED* - Remote-Befehl fehlgeschlagen
error	*string*	*(optional)* Fehlermeldung.
parent_taskid	*number*	ID der übergeordneten Aufgabe.
more	*number*	*(optional)* 1 - es sind weitere Verlaufsdaten zu senden.
clock	*number*	*(optional)* Zeitstempel der Datenübertragung (Sekunden).
ns	*number*	*(optional)* Zeitstempel der Datenübertragung (Nanosekunden).
version	*string*	Proxy-Version (<major>.<minor>.<build>).
server→proxy:		
response	*string*	Information zum Erfolg der Anforderung ('success' oder 'failed').
tasks	*array*	*(optional)* Array von Aufgaben.

name	value type	description
type	*number*	Aufgabentyp: **1**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND* - Remote-Befehl
clock	*number*	Erstellungszeit der Aufgabe.
ttl	*number*	Zeit in Sekunden, nach der die Aufgabe abläuft.
commandtype	*number*	Typ des Remote-Befehls: **0**, *ZBX_SCRIPT_TYPE_CUSTOM_SCRIPT* - benutzerdefiniertes Skript verwenden **1**, *ZBX_SCRIPT_TYPE_IPMI* - IPMI verwenden **2**, *ZBX_SCRIPT_TYPE_SSH* - SSH verwenden **3**, *ZBX_SCRIPT_TYPE_TELNET* - Telnet verwenden **4**, *ZBX_SCRIPT_TYPE_GLOBAL_SCRIPT* - globales Skript verwenden (derzeit funktional gleichwertig mit benutzerdefiniertem Skript)
command	*string*	Auszuführender Remote-Befehl.
execute_on	*number*	Ausführungsziel für benutzerdefinierte Skripte: **0**, *ZBX_SCRIPT_EXECUTE_ON_AGENT* - Skript auf dem Agent ausführen **1**, *ZBX_SCRIPT_EXECUTE_ON_SERVER* - Skript auf dem Server ausführen **2**, *ZBX_SCRIPT_EXECUTE_ON_PROXY* - Skript auf dem Proxy ausführen
port	*number*	*(optional)* Port für Telnet- und SSH-Befehle.
authtype	*number*	*(optional)* Authentifizierungstyp für SSH-Befehle.
username	*string*	*(optional)* Benutzername für Telnet- und SSH-Befehle.
password	*string*	*(optional)* Passwort für Telnet- und SSH-Befehle.
publickey	*string*	*(optional)* Öffentlicher Schlüssel für SSH-Befehle.
privatekey	*string*	*(optional)* Privater Schlüssel für SSH-Befehle.
parent_taskid	*number*	ID der übergeordneten Aufgabe.
hostid	*number*	ID des Ziel-Hosts.

Beispiel:

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
      }
  ]
}
```


Tasks-Anfrage

Der Austausch über `proxy tasks` dient der Übermittlung und Bestätigung von Remote-Befehlsaufgaben zwischen dem Server und einem passiven Proxy. Wenn der Server den Proxy nach Aufgabenergebnissen abfragt, sendet er in dem durch `ProxyDataFrequency` definierten Intervall eine leere `proxy tasks`-Anfrage. Der Proxy antwortet mit seiner aktuellen Version und allen ausstehenden Aufgabenergebnissen (einschließlich Zeitstempeln). Der Server bestätigt anschließend den Empfang, indem er eine `response` zurückgibt, und kann neue Aufgaben für den Proxy zur Ausführung mitgeben.

name	value type	description
server→proxy:		
request	*string*	'proxy tasks'
proxy→server:		
version	*string*	Proxy-Version (<major>.<minor>.<build>).
clock	*number*	*(optional)* Zeitstempel der Datenübertragung (Sekunden).
ns	*number*	*(optional)* Zeitstempel der Datenübertragung (Nanosekunden).
server→proxy:		

name	value type	description
response	*string*	Erfolgsinformation der Anfrage (’success’ oder ’failed’).

Beispiel:

server→proxy:

```
{
  "request":"proxy tasks"
}
```

proxy→server:

```
{
  "version":"7.0.0",
  "clock":1721059872,
  "ns":848141522
}
```

server→proxy:

```
{
  "response":"success"
}
```

Aktiver Proxy

Konfigurationsanforderung

Die Anforderung `proxy config` wird vom aktiven Proxy gesendet, um Konfigurationsdaten des Proxys abzurufen. Diese Anforderung wird alle `ProxyConfigFrequency` Sekunden (Konfigurationsparameter des Proxys) gesendet.

name	value type	description
proxy→server:		
request	*string*	„proxy config“
host	*string *	Proxy-Name.
version	*string*	Proxy-Version (<major>.<minor>.<build>).
session	*string*	Sitzungstoken der Proxy-Konfiguration.
config_revision	*number*	Revision der Proxy-Konfiguration.
hostmap_revision	*number*	Aktuelle Revision der Zuordnung von Hosts zu Proxys.
server→proxy:		
full_sync	*number*	1 - wenn vollständige Konfigurationsdaten gesendet werden, andernfalls nicht vorhanden (optional).
data	*array*	Objekt mit Tabellendaten. Nicht vorhanden, wenn die Konfiguration nicht geändert wurde (optional).
<table>	*object*	Ein oder mehrere Objekte mit <table>-Daten (optional, abhängig von den Änderungen).
fields	*array*	Array mit Feldnamen.
-	*string*	Feldname.
data	*array*	Array mit Zeilen.
-	*array*	Array mit Spalten.
-	*string*,*number*	Spaltenwert mit einem Typ, der vom Spaltentyp im Datenbankschema abhängt.
macro.secrets	*object*	Informationen zu geheimen Makros, nicht vorhanden, wenn es keine Änderungen an Tresormakros gibt (optional).
proxy_group	*string*	Name der Proxy-Gruppe, zu der der Proxy gehört.
config_revision	*number*	Revision des Konfigurations-Caches - wird mit den Konfigurationsdaten gesendet (optional).
del_hostids	*array*	Array der entfernten Hostids (optional).
-	*number*	Host-ID.
del_macro_hostids	*array*	Array der Hostids, bei denen alle Makros entfernt wurden (optional).

name	value type	description
-	*number*	Host-ID.

Beispiel:

proxy→server:

```
{
  "request": "proxy config",
  "host": "Zabbix proxy",
  "version":"8.0.0",
  "session": "fd59a09ff4e9d1fb447de1f04599bcf6",
  "config_revision": 0
}
```

server→proxy:

```
{
"full_sync": 1,
"data": {
"hosts": {
"fields": ["hostid", "host", "status", "ipmi_authtype", "ipmi_privilege", "ipmi_username", "ipmi_password"
"data": [
[10084, "Zabbix server", 0, -1, 2, "", "", "Zabbix server", 1, 1, "", "", "", ""]
]
},
"interface": {
"fields": ["interfaceid", "hostid", "main", "type", "useip", "ip", "dns", "port", "available"],
"data": [
[1, 10084, 1, 1, 1, "127.0.0.1", "", "10053", 1]
]
},
"interface_snmp": {
"fields": ["interfaceid", "version", "bulk", "community", "securityname", "securitylevel", "authpassphrase
"data": []
},
"host_inventory": {
"fields": ["hostid", "type", "type_full", "name", "alias", "os", "os_full", "os_short", "serialno_a", "ser
"data": [
[10084, "", "", ..., "56.95387", "24.22067", ..., ""]
]
},
"items": {
"fields": ["itemid", "type", "snmp_oid", "hostid", "key_", "delay", "history", "status", "value_type", "tr
"data": [
[44161, 7, "", 10084, "agent.hostmetadata", "10s", "90d", 0, 1, "", "", "", "", 0, "", "", "", "", 0, null
[44162, 0, "", 10084, "agent.ping", "10s", "90d", 0, 3, "", "", "", "", 0, "", "", "", "", 0, 1, 0, "", nu
]
},
"item_rtdata": {
"fields": ["itemid", "lastlogsize", "mtime"],
"data": [
[44161, 0, 0],
[44162, 0, 0]
]
},
"item_preproc": {
"fields": ["item_preprocid", "itemid", "step", "type", "params", "error_handler", "error_handler_params"],
"data": []
},
"item_parameter": {
"fields": ["item_parameterid", "itemid", "name", "value"],
"data": []
```

```
},
"globalmacro": {
"fields": ["globalmacroid", "macro", "value", "type"],
"data": [
[2, "{$SNMP_COMMUNITY}", "public", 0]
]
},
"hosts_templates": {
"fields": ["hosttemplateid", "hostid", "templateid", "link_type"],
"data": []
},
"hostmacro": {
"fields": ["hostmacroid", "hostid", "macro", "value", "type", "automatic"],
"data": [
[5676, 10084, "{$M}", "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix:Content", 2, 0]
]
},
"drules": {
"fields": ["druleid", "name", "iprange", "delay"],
"data": [
[2, "Local network", "127.0.0.1", "10s"]
]
},
"dchecks": {
"fields": ["dcheckid", "druleid", "type", "key_", "snmp_community", "ports", "snmpv3_securityname", "snmpv
"data": [
[2, 2, 9, "system.uname", "", "10052", "", 0, "", "", 0, 0, 0, "", 1, 0]
]
},
"regexps": {
"fields": ["regexpid", "name"],
"data": [
[1, "File systems for discovery"],
[2, "Network interfaces for discovery"],
[3, "Storage devices for SNMP discovery"],
[4, "Windows service names for discovery"],
[5, "Windows service startup states for discovery"]
]
},
"expressions": {
"fields": ["expressionid", "regexpid", "expression", "expression_type", "exp_delimiter", "case_sensitive"]
"data": [
[1, 1, "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs)$", 3, ",", 0
[3, 3, "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$", 4, ",", 1],
[5, 4, "^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.30319_32)$", 4, ","
[6, 5, "^(automatic|automatic delayed)$", 3, ",", 1],
[7, 2, "^Software Loopback Interface", 4, ",", 1],
[8, 2, "^(In)?[Ll]oop[Bb]ack[0-9._]*$", 4, ",", 1],
[9, 2, "^NULL[0-9.]*$", 4, ",", 1],
[10, 2, "^[Ll]o[0-9.]*$", 4, ",", 1],
[11, 2, "^[Ss]ystem$", 4, ",", 1],
[12, 2, "^Nu[0-9.]*$", 4, ",", 1]
]
},
"settings": {
"fields": ["name", "type", "value_str", "value_int"],
"data": [
["autoreg_tls_accept", 2, "", 1],
["hk_history_global", 2, "", 0],
["snmptrap_logging", 2, "", 1],
["proxy_secrets_provider", 2, "", 0],
["hk_history", 1, "31d", 0],
```

```
["timeout_db_monitor", 1, "3s", 0],
["timeout_external_check", 1, "3s", 0],
["timeout_http_agent", 1, "3s", 0],
["timeout_simple_check", 1, "3s", 0],
["timeout_snmp_agent", 1, "3s", 0],
["timeout_ssh_agent", 1, "3s", 0],
["timeout_telnet_agent", 1, "3s", 0],
["timeout_zabbix_agent", 1, "3s", 0],
["timeout_browser", 1, "30s", 0]
]
},
"httptest": {
"fields": ["httptestid", "name", "delay", "agent", "authentication", "http_user", "http_password", "hostid
"data": []
},
"httptestitem": {
"fields": ["httptestitemid", "httptestid", "itemid", "type"],
"data": []
},
"httptest_field": {
"fields": ["httptest_fieldid", "httptestid", "type", "name", "value"],
"data": []
},
"httpstep": {
"fields": ["httpstepid", "httptestid", "name", "no", "url", "timeout", "posts", "required", "status_codes"
"data": []
},
"httpstepitem": {
"fields": ["httpstepitemid", "httpstepid", "itemid", "type"],
"data": []
},
"httpstep_field": {
"fields": ["httpstep_fieldid", "httpstepid", "type", "name", "value"],
"data": []
},
"config_autoreg_tls": {
"fields": ["autoreg_tlsid", "tls_psk_identity", "tls_psk"],
"data": [
[1, "", ""]
]
}
},
"macro.secrets": {
"AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix": {
"Content": "738"
}
},
"config_revision": 2
}
```


Datenanforderung

Die Anfrage `proxy data` wird vom Proxy gesendet, um Verfügbarkeit von Host-Schnittstellen, Historie, Discovery- und Autoregistrierungsdaten bereitzustellen. Diese Anfrage wird alle `DataSenderFrequency` Sekunden gesendet (Proxy-Konfigurationsparameter). Beachten Sie, dass ein aktiver Proxy den Zabbix Server weiterhin jede Sekunde nach Aufgaben für Remote-Befehle abfragt (mit einer leeren Anfrage `proxy data`).

name	value type	description
proxy→server:		
request	*string*	'proxy data'
host	*string*	Proxy-Name.
session	*string*	Daten-Session-Token.

name	value type	description
interface availability	*array*	*(optional)* Array von Objekten mit Schnittstellenverfügbarkeit.
interfaceid	*number*	Schnittstellen-ID.
available	*number*	Schnittstellenverfügbarkeit: **0**, *INTERFACE_AVAILABLE_UNKNOWN* - unbekannt **1**, *INTERFACE_AVAILABLE_TRUE* - verfügbar **2**, *INTERFACE_AVAILABLE_FALSE* - nicht verfügbar
error	*string*	Fehlermeldung der Schnittstelle oder leerer String.
history data	*array*	*(optional)* Array von Objekten mit Verlaufsdaten.
itemid	*number*	Datenpunkt-ID.
clock	*number*	Zeitstempel des Datenpunktwerts (Sekunden).
ns	*number*	Zeitstempel des Datenpunktwerts (Nanosekunden).
value	*string*	*(optional)* Datenpunktwert.
id	*number*	Wert-ID (aufsteigender Zähler, innerhalb einer Daten-Session eindeutig).
timestamp	*number*	*(optional)* Zeitstempel von Log-Typ-Datenpunkten.
source	*string*	*(optional)* Quellwert des Eventlog-Datenpunkts.
severity	*number*	*(optional)* Schweregradwert des Eventlog-Datenpunkts.
eventid	*number*	*(optional)* Event-ID-Wert des Eventlog-Datenpunkts.
state	*string*	*(optional)* Datenpunktstatus: **0**, *ITEM_STATE_NORMAL* **1**, *ITEM_STATE_NOTSUPPORTED*
lastlogsize	*number*	*(optional)* Letzte Loggröße von Datenpunkten des Log-Typs.
mtime	*number*	*(optional)* Änderungszeit von Datenpunkten des Log-Typs.
discovery data	*array*	*(optional)* Array von Discovery-Datenobjekten.
clock	*number*	Zeitstempel der Discovery-Daten.
druleid	*number*	ID der Discovery-Regel.
dcheckid	*number*	ID der Discovery-Prüfung oder null für Daten der Discovery-Regel.
type	*number*	Typ der Discovery-Prüfung: **-1** Daten der Discovery-Regel **0**, *SVC_SSH* - SSH-Dienstprüfung **1**, *SVC_LDAP* - LDAP-Dienstprüfung **2**, *SVC_SMTP* - SMTP-Dienstprüfung **3**, *SVC_FTP* - FTP-Dienstprüfung **4**, *SVC_HTTP* - HTTP-Dienstprüfung **5**, *SVC_POP* - POP-Dienstprüfung **6**, *SVC_NNTP* - NNTP-Dienstprüfung **7**, *SVC_IMAP* - IMAP-Dienstprüfung **8**, *SVC_TCP* - Prüfung der TCP-Port-Verfügbarkeit **9**, *SVC_AGENT* - Zabbix Agent **10**, *SVC_SNMPv1* - SNMPv1 Agent **11**, *SVC_SNMPv2* - SNMPv2 Agent **12**, *SVC_ICMPPING* - ICMP-Ping **13**, *SVC_SNMPv3* - SNMPv3 Agent **14**, *SVC_HTTPS* - HTTPS-Dienstprüfung **15**, *SVC_TELNET* - Telnet-Verfügbarkeitsprüfung
ip	*string*	IP-Adresse des Host.
dns	*string*	DNS-Name des Host.
port	*number*	*(optional)* Dienst-Portnummer.
key_	*string*	*(optional)* Datenpunkt-Schlüssel für Discovery-Prüfung vom Typ **9** *SVC_AGENT*
value	*string*	*(optional)* Vom Dienst empfangener Wert, kann bei den meisten Diensten leer sein.
status	*number*	*(optional)* Dienststatus: **0**, *DOBJECT_STATUS_UP* - Dienst UP **1**, *DOBJECT_STATUS_DOWN* - Dienst DOWN

name	value type	description
autoregistration	*array*	*(optional)* Array von Autoregistrierungsdatenobjekten.
clock	*number*	Zeitstempel der Autoregistrierungsdaten.
host	*string*	Host-Name.
ip	*string*	*(optional)* IP-Adresse des Host.
dns	*string*	*(optional)* Aus der IP-Adresse aufgelöster DNS-Name.
port	*string*	*(optional)* Host-Port.
host_metadata	*string*	*(optional)* Vom Agent gesendete Host-Metadaten (basierend auf dem Agent-Konfigurationsparameter HostMetadata oder HostMetadataItem).
tasks	*array*	*(optional)* Array von Aufgaben.
type	*number*	Aufgabentyp: **0**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND_RESULT* - Ergebnis eines Remote-Befehls
status	*number*	Ausführungsstatus des Remote-Befehls: **0**, *ZBX_TM_REMOTE_COMMAND_COMPLETED* - Remote-Befehl erfolgreich abgeschlossen **1**, *ZBX_TM_REMOTE_COMMAND_FAILED* - Remote-Befehl fehlgeschlagen
error	*string*	*(optional)* Fehlermeldung.
parent_taskid	*number*	ID der übergeordneten Aufgabe.
more	*number*	*(optional)* 1 - es sind weitere Verlaufsdaten zu senden
clock	*number*	*(optional)* Zeitstempel der Datenübertragung (Sekunden).
ns	*number*	*(optional)* Zeitstempel der Datenübertragung (Nanosekunden).
version	*string*	Proxy-Version (<major>.<minor>.<build>).
server→proxy:		
response	*string*	Information über den Erfolg der Anfrage (’success’ oder ’failed’).
upload	*string*	Upload-Steuerung für historische Daten (Historie, Autoregistrierung, Host-Verfügbarkeit, Netzwerk-Discovery). Mögliche Werte: **enabled** - normaler Betrieb **disabled** - der Server akzeptiert keine Daten (möglicherweise aufgrund eines internen Cache-Überlaufs)
tasks	*array*	*(optional)* Array von Aufgaben.
type	*number*	Aufgabentyp: **1**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND* - Remote-Befehl
clock	*number*	Erstellungszeit der Aufgabe.
ttl	*number*	Zeit in Sekunden, nach der die Aufgabe abläuft.
commandtype	*number*	Typ des Remote-Befehls: **0**, *ZBX_SCRIPT_TYPE_CUSTOM_SCRIPT* - benutzerdefiniertes Skript verwenden **1**, *ZBX_SCRIPT_TYPE_IPMI* - IPMI verwenden **2**, *ZBX_SCRIPT_TYPE_SSH* - SSH verwenden **3**, *ZBX_SCRIPT_TYPE_TELNET* - Telnet verwenden **4**, *ZBX_SCRIPT_TYPE_GLOBAL_SCRIPT* - globales Skript verwenden (derzeit funktional gleichwertig mit benutzerdefiniertem Skript)
command	*string*	Auszuführender Remote-Befehl.
execute_on	*number*	Ausführungsziel für benutzerdefinierte Skripte: **0**, *ZBX_SCRIPT_EXECUTE_ON_AGENT* - Skript auf Agent ausführen **1**, *ZBX_SCRIPT_EXECUTE_ON_SERVER* - Skript auf Server ausführen **2**, *ZBX_SCRIPT_EXECUTE_ON_PROXY* - Skript auf Proxy ausführen
port	*number*	*(optional)* Port für Telnet- und SSH-Befehle.
authtype	*number*	*(optional)* Authentifizierungstyp für SSH-Befehle.
username	*string*	*(optional)* Benutzername für Telnet- und SSH-Befehle.
password	*string*	*(optional)* Passwort für Telnet- und SSH-Befehle.
publickey	*string*	*(optional)* Öffentlicher Schlüssel für SSH-Befehle.
privatekey	*string*	*(optional)* Privater Schlüssel für SSH-Befehle.
parent_taskid	*number*	ID der übergeordneten Aufgabe.

name	value type	description
hostid	*number*	ID des Ziel-Host.

Beispiel:

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
}]
}
```


Konfigurationsparameter: Zabbix Proxy

Übersicht

Konfigurationsparameter ermöglichen die Anpassung des Verhaltens des Zabbix Proxy.

Geben Sie Parameterwerte in der Proxy-Konfigurationsdatei (`zabbix_proxy.conf`) an. Alle Parameter sind **optional**, sofern nicht ausdrücklich angegeben ist, dass der Parameter obligatorisch ist.

Diese Seite ist wie folgt strukturiert:

- Parameterübersicht (um vollständige Details zu einem Parameter anzuzeigen, klicken Sie auf seinen Namen)
- Parameterdetails

Parameter	Beschreibung
AllowRoot	Erlaubt die Ausführung des Proxy als `root`.
AllowUnsupportedDBVersions	Erlaubt dem Proxy, mit nicht unterstützten Datenbankversionen zu arbeiten.
APMListenIP	Gibt die IP-Adresse an, an der der APM-Collector auf eingehende Telemetriedaten lauschen soll.
APMListenPort	Gibt den Port an, an dem der APM-Collector auf eingehende Telemetriedaten lauschen soll.
APMTLSCAFile	Gibt den vollständigen Pfadnamen der Datei an, die die Zertifikate der übergeordneten CA(s) zur Überprüfung der Zertifikate von Anwendungen oder OpenTelemetry-Collectors enthält, die Daten an den APM-Collector senden.
APMTLSCertFile	Gibt den vollständigen Pfadnamen der Datei an, die das Zertifikat oder die Zertifikatskette des APM-Collectors enthält.
APMTLSKeyFile	Gibt den vollständigen Pfadnamen der Datei an, die den privaten Schlüssel des APM-Collectors enthält.
CacheSize	Legt die Größe des Konfigurations-Caches fest.
ConfigFrequency	Dieser Parameter ist **veraltet** (verwenden Sie stattdessen ProxyConfigFrequency). Legt das Intervall für den Abruf von Konfigurationsdaten vom Zabbix Server fest.
DataSenderFrequency	Legt das Intervall für das Senden gesammelter Daten an den Zabbix Server fest.
DBHost	Gibt den Datenbank-Host (oder das Socket-Verzeichnis) an.
DBName	Gibt den Datenbanknamen oder den Pfad zur Datenbankdatei für SQLite3 an.

Parameter	Beschreibung
DBPassword	Gibt das Datenbankpasswort an.
DBPort	Gibt den Port des Datenbank-Hosts von Zabbix an.
DBSchema	Gibt den Namen des Datenbankschemas an. Wird für PostgreSQL verwendet.
DBSocket	Gibt den Pfad zur MySQL-Socketdatei an.
DBUser	Gibt den Datenbankbenutzer an.
DBTLSConnect	Gibt einen Wert an, der eine TLS-Verbindung zur Datenbank erzwingt.
DBTLSCAFile	Gibt den vollständigen Pfadnamen der Datei an, die die Zertifikate der übergeordneten CA(s) zur Überprüfung des Datenbankzertifikats enthält.
DBTLSCertFile	Gibt den vollständigen Pfadnamen der Datei an, die das Zabbix-Proxy-Zertifikat zur Authentifizierung bei der Datenbank enthält.
DBTLSKeyFile	Gibt den vollständigen Pfadnamen der Datei an, die den privaten Schlüssel zur Authentifizierung bei der Datenbank enthält.
DBTLSCipher	Gibt die Liste der Verschlüsselungschiffren an, die der Zabbix Proxy für TLS-Protokolle bis einschließlich TLS v1.2 zulässt. Wird nur für MySQL unterstützt.
DBTLSCipher13	Gibt die Liste der Verschlüsselungssammlungen an, die der Zabbix Proxy für das TLS-v1.3-Protokoll zulässt. Wird nur für MySQL ab Version 8.0.16 unterstützt.
DebugLevel	Legt die Debug-Stufe fest.
EnableRemoteCommands	Erlaubt Remote-Befehle vom Zabbix Server.
ExternalScripts	Gibt den Speicherort externer Skripte an.
Fping6Location	Gibt den Speicherort von fping6 an.
FpingLocation	Gibt den Speicherort von fping an.
HistoryCacheSize	Legt die Größe des Verlaufs-Caches fest.
HistoryIndexCacheSize	Legt die Größe des Verlaufsindex-Caches fest.
Hostname	Gibt den Namen des Proxy an.
HostnameItem	Gibt den Datenpunkt an, der zum Festlegen von Hostname verwendet wird (falls nicht definiert).
HousekeepingFrequency	Legt das Intervall der Housekeeping-Prozedur fest (in Stunden).
Include	Gibt ein Verzeichnis oder einzelne Dateien an, die in die Konfigurationsdatei aufgenommen werden sollen.
JavaGateway	Gibt die IP-Adresse (oder den Hostnamen) des Zabbix Java gateway an.
JavaGatewayPort	Gibt den Port des Zabbix Java gateway an.
ListenBacklog	Legt die maximale Anzahl ausstehender Verbindungen in der TCP-Warteschlange fest.
ListenIP	Gibt die IP-Adresse an, an der der Trapper lauschen soll.
ListenPort	Gibt den Port an, an dem der Trapper lauschen soll.
LoadModule	Gibt das beim Start des Proxy zu ladende Modul an.
LoadModulePath	Gibt den vollständigen Pfad zum Speicherort der Proxy-Module an.
LogFile	Gibt die Protokolldatei des Proxy an.
LogFileSize	Legt die maximale Größe der Protokolldatei fest.
LogRemoteCommands	Aktiviert die Protokollierung ausgeführter Shell-Befehle als Warnungen.
LogSlowQueries	Legt die maximale Dauer fest, die eine Datenbankabfrage dauern darf, bevor sie protokolliert wird (in Millisekunden).
LogType	Gibt den Typ der Protokollausgabe an.
MaxConcurrentChecksPerPoller	Legt die maximale Anzahl asynchroner Prüfungen fest, die gleichzeitig von jedem HTTP-Agent-Poller, Agent-Poller oder SNMP-Poller ausgeführt werden können.
PidFile	Gibt die PID-Datei an.
ProxyBufferMode	Gibt den Speichermechanismus für Verlaufs-, Discovery- und Autoregistrierungsdaten an (Festplatte/Speicher/hybrid).
ProxyConfigFrequency	Legt das Intervall für den Abruf von Konfigurationsdaten vom Zabbix Server in Sekunden fest.
ProxyLocalBuffer	Legt fest, wie lange Daten lokal aufbewahrt werden (in Stunden), selbst wenn sie bereits mit dem Server synchronisiert wurden.
ProxyMemoryBufferAge	Legt das maximale Alter von Daten im Proxy-Speicherpuffer in Sekunden fest.
ProxyMemoryBufferSize	Legt die Größe des gemeinsam genutzten Speichercaches für gesammelte Verlaufs-, Discovery- und Autoregistrierungsdaten fest.
ProxyMode	Legt den Betriebsmodus des Proxy fest (aktiv/passiv).
ProxyOfflineBuffer	Legt fest, wie lange Daten bei fehlender Verbindung zum Zabbix Server aufbewahrt werden (in Stunden).
Server	Wenn ProxyMode auf `active` gesetzt ist: Gibt die IP-Adresse oder den DNS-Namen des Zabbix Server (address:port) oder des Clusters (address:port;address2:port) an, von dem Konfigurationsdaten abgerufen und an den Daten gesendet werden. Wenn ProxyMode auf `passive` gesetzt ist: Gibt die IP-Adressen oder DNS-Namen des Zabbix Server an.
SNMPTrapperFile	Gibt die temporäre Datei an, die zur Übergabe von Daten vom SNMP-Trap-Daemon an den Proxy verwendet wird.

Parameter	Beschreibung
SocketDir	Gibt das Verzeichnis zum Speichern von IPC-Sockets an, die von internen Zabbix-Diensten verwendet werden.
SourceIP	Gibt die Quell-IP-Adresse an.
SSHKeyLocation	Gibt den Speicherort öffentlicher und privater Schlüssel für SSH-Prüfungen und -Aktionen an.
SSLCertLocation	Gibt den Speicherort der SSL-Clientzertifikatsdateien für die Clientauthentifizierung an.
SSLKeyLocation	Gibt den Speicherort der privaten SSL-Schlüsseldateien für die Clientauthentifizierung an.
SSLCALocation	Gibt den Speicherort der Zertifizierungsstellendateien (CA) für die Überprüfung von SSL-Serverzertifikaten an.
StartAgentPollers	Legt die Anzahl vorab geforkter Instanzen asynchroner Zabbix-Agent-Poller fest.
StartAPMCollector	Auf 1 setzen, um den APM-Collector zu starten.
StartBrowserPollers	Legt die Anzahl vorab geforkter Instanzen von Browser-Datenpunkt-Pollern fest.
StartDBSyncers	Legt die Anzahl vorab geforkter Instanzen von Verlaufs-Synchronisierern fest.
StartDiscoverers	Legt die Anzahl vorab geforkter Instanzen von Discovery-Workern fest.
StartHTTPAgentPollers	Legt die Anzahl vorab geforkter Instanzen asynchroner HTTP-Agent-Poller fest.
StartHTTPPollers	Legt die Anzahl vorab geforkter Instanzen von HTTP-Pollern fest.
StartIPMIPollers	Legt die Anzahl vorab geforkter Instanzen von IPMI-Pollern fest.
StartJavaPollers	Legt die Anzahl vorab geforkter Instanzen von Java-Pollern fest.
StartODBCPollers	Legt die Anzahl vorab geforkter Instanzen von ODBC-Pollern fest.
StartPingers	Legt die Anzahl vorab geforkter Instanzen von ICMP-Pingern fest.
StartPollersUnreachable	Legt die Anzahl vorab geforkter Instanzen von Pollern für nicht erreichbare Hosts fest (einschließlich IPMI und Java).
StartPollers	Legt die Anzahl vorab geforkter Instanzen von Pollern fest.
StartPreprocessors	Legt die Anzahl vorab gestarteter Instanzen von Vorverarbeitungs-Workern fest.
StartSNMPPollers	Legt die Anzahl vorab geforkter Instanzen asynchroner SNMP-Poller fest.
StartSNMPTrapper	Auf „1“ setzen, um einen SNMP-Trapper-Prozess zu starten.
StartTelemetryQueryPollers	Legt die Anzahl vorab geforkter Instanzen asynchroner Telemetrieabfrage-Poller fest.
StartTrappers	Legt die Anzahl vorab geforkter Instanzen von Trappern fest.
StartVMwareCollectors	Legt die Anzahl vorab geforkter VMware-Collector-Instanzen fest.
StatsAllowedIP	Gibt die IP-Adressen oder DNS-Namen externer Zabbix-Instanzen an, die Statistiken anfordern dürfen.
TelemetryProvider	Legt die Quelle der Telemetriedaten fest.
Timeout	Gibt an, wie lange (in Sekunden) auf den Verbindungsaufbau und den Datenaustausch mit dem Zabbix Proxy, Agent, Webdienst und bei SNMP-Prüfungen gewartet werden soll (ausgenommen SNMP-`walk[OID]`- und `get[OID]`-Datenpunkte).
TLSAccept	Gibt die für eingehende Verbindungen vom Zabbix Server erforderliche Verschlüsselungsstufe an.
TLSCAFile	Gibt den vollständigen Pfadnamen der Datei an, die die Zertifikate der übergeordneten CA(s) zur Überprüfung von Peer-Zertifikaten enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSCertFile	Gibt den vollständigen Pfadnamen der Datei an, die das Serverzertifikat oder die Zertifikatskette enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSCipherAll	Gibt die GnuTLS-Prioritätszeichenfolge oder die OpenSSL-Chiffrenzeichenfolge (TLS 1.2) an. Überschreibt die Standardkriterien für die Auswahl der Verschlüsselungssammlung bei zertifikats- und PSK-basierter Verschlüsselung.
TLSCipherAll13	Gibt die Chiffrenzeichenfolge für OpenSSL 1.1.1 oder neuer unter TLS 1.3 an. Überschreibt die Standardkriterien für die Auswahl der Verschlüsselungssammlung bei zertifikats- und PSK-basierter Verschlüsselung.
TLSCipherCert	Gibt die GnuTLS-Prioritätszeichenfolge oder die OpenSSL-Chiffrenzeichenfolge (TLS 1.2) an. Überschreibt die Standardkriterien für die Auswahl der Verschlüsselungssammlung bei zertifikatsbasierter Verschlüsselung.
TLSCipherCert13	Gibt die Chiffrenzeichenfolge für OpenSSL 1.1.1 oder neuer unter TLS 1.3 an. Überschreibt die Standardkriterien für die Auswahl der Verschlüsselungssammlung bei zertifikatsbasierter Verschlüsselung.
TLSCipherPSK	Gibt die GnuTLS-Prioritätszeichenfolge oder die OpenSSL-Chiffrenzeichenfolge (TLS 1.2) an. Überschreibt die Standardkriterien für die Auswahl der Verschlüsselungssammlung bei PSK-basierter Verschlüsselung.
TLSCipherPSK13	Gibt die Chiffrenzeichenfolge für OpenSSL 1.1.1 oder neuer unter TLS 1.3 an. Überschreibt die Standardkriterien für die Auswahl der Verschlüsselungssammlung bei PSK-basierter Verschlüsselung.
TLSConnect	Gibt die für Verbindungen zum Zabbix Server erforderliche Verschlüsselungsstufe an.
TLSCRLFile	Gibt den vollständigen Pfadnamen der Datei an, die widerrufene Zertifikate enthält. Dieser Parameter wird für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet.

Parameter	Beschreibung
TLSKeyFile	Gibt den vollständigen Pfadnamen der Datei an, die den privaten Schlüssel des Proxy enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSListen	Erfordert ausschließlich TLS-Verbindungen am Trapper-Socket.
TLSPSKFile	Gibt den vollständigen Pfadnamen der Datei an, die den vorab gemeinsam genutzten Schlüssel des Proxy enthält und für verschlüsselte Kommunikation mit dem Zabbix Server verwendet wird.
TLSPSKIdentity	Gibt die Identitätszeichenfolge des vorab gemeinsam genutzten Schlüssels an, die für verschlüsselte Kommunikation mit dem Zabbix Server verwendet wird.
TLSServerCertIssuer	Gibt den zulässigen Aussteller des Serverzertifikats an.
TLSServerCertSubject	Gibt das zulässige Subjekt des Serverzertifikats an.
TmpDir	Gibt das temporäre Verzeichnis an.
TrapperTimeout	Gibt das Timeout in Sekunden an für: - den Abruf von Konfigurationsdaten vom Zabbix Server; - die Ausführung globaler Skripte oder Remote-Befehle.
UnavailableDelay	Legt das Intervall für die Überprüfung der Host-Verfügbarkeit während des Zeitraums der Nichtverfügbarkeit fest.
UnreachableDelay	Legt das Intervall für die Überprüfung der Host-Verfügbarkeit während des Zeitraums der Nichterreichbarkeit fest.
UnreachablePeriod	Legt fest, nach wie vielen Sekunden der Nichterreichbarkeit ein Host als nicht verfügbar behandelt wird.
User	Gibt die Berechtigungen auf einen bestimmten, vorhandenen Benutzer im System ab.
Vault	Gibt den Vault-Anbieter an.
VaultAppRoleID	Gibt die HashiCorp-Vault-AppRole-ID an.
VaultAppSecretID	Gibt die HashiCorp-Vault-AppRole-Secret-ID an.
VaultDBPath	Gibt den Speicherort an, aus dem Datenbankanmeldedaten anhand von Schlüsseln abgerufen werden.
VaultPrefix	Gibt ein benutzerdefiniertes Präfix für den Vault-Pfad oder die Abfrage an.
VaultTLSCertFile	Gibt den Namen der SSL-Zertifikatsdatei an, die für die Clientauthentifizierung verwendet wird.
VaultTLSKeyFile	Gibt den Namen der privaten SSL-Schlüsseldatei an, die für die Clientauthentifizierung verwendet wird.
VaultToken	Gibt das HashiCorp-Vault-Authentifizierungstoken an.
VaultURL	Gibt die URL des Vault-Servers an.
VMwareCacheSize	Legt die Größe des gemeinsam genutzten Speichers zum Speichern von VMware-Daten fest.
VMwareFrequency	Legt das Intervall für die Datenerfassung von einem einzelnen VMware-Dienst in Sekunden fest.
VMwarePerfFrequency	Legt das Intervall für den Abruf von Leistungszählerstatistiken von einem einzelnen VMware-Dienst in Sekunden fest.
VMwareTimeout	Legt die maximale Anzahl von Sekunden fest, die ein `vmware collector` auf eine Antwort vom VMware-Dienst wartet.
WebDriverURL	Gibt die URL der WebDriver-Schnittstelle an.

Beachten Sie:

- Die Standardwerte entsprechen den Daemon-Standards und nicht den Werten in den mitgelieferten Konfigurationsdateien.
- Werte unterstützen Umgebungsvariablen.
- Zabbix unterstützt Konfigurationsdateien nur in UTF-8-Kodierung ohne BOM.
- Kommentare, die mit # beginnen, werden nur am Anfang der Zeile unterstützt.

Parameterdetails

AllowRoot

Erlaubt dem Proxy, als `root` zu laufen. Wenn diese Option deaktiviert ist und der Proxy von `root` gestartet wird, versucht der Proxy stattdessen zum Benutzer `zabbix` zu wechseln. Hat keine Wirkung, wenn er unter einem normalen Benutzer gestartet wird.

Standard: 0
 Werte: 0 - nicht erlauben; 1 - erlauben

AllowUnsupportedDBVersions

Erlaubt dem Proxy, mit nicht unterstützten Datenbankversionen zu arbeiten.

Standard: 0
 Werte: 0 - nicht erlauben; 1 - erlauben

APMListenIP

Geben Sie die IP-Adresse an, auf der der APM collector auf eingehende Telemetriedaten warten soll.
 Wenn nicht angegeben, wartet der APM collector auf allen Netzwerkschnittstellen.

Standard: `0.0.0.0`

APMListenPort

Geben Sie den Port an, an dem der APM collector auf eingehende Telemetriedaten lauschen soll.
 Der Standardport ist der standardmäßige OpenTelemetry-Port (OTLP/gRPC), der standardmäßig von OpenTelemetry-SDKs und -Collectors verwendet wird.

Standard: 4317
 Bereich: 1024-32767

APMTLSCAFile

Geben Sie den vollständigen Pfadnamen der Datei an, die die Zertifikate der übergeordneten Zertifizierungsstelle(n) zur Überprüfung der Zertifikate von Anwendungen oder OpenTelemetry Collectors enthält, die Daten an den APM Collector senden.
 Wenn dieser Parameter festgelegt ist, akzeptiert der APM Collector nur verschlüsselte Verbindungen. Andernfalls akzeptiert er nur unverschlüsselte Verbindungen.

APMTLSCertFile

Geben Sie den vollständigen Pfadnamen der Datei an, die das Zertifikat oder die Zertifikatskette des APM-Collectors enthält.
 Dieser Parameter muss zusammen mit `APMTLSKeyFile` und nur dann festgelegt werden, wenn `APMTLSCAFile` festgelegt ist.

APMTLSKeyFile

Geben Sie den vollständigen Pfadnamen der Datei an, die den privaten Schlüssel des APM-Kollektors enthält.
 Dieser Parameter muss zusammen mit `APMTLSCertFile` gesetzt werden, und zwar nur, wenn `APMTLSCAFile` gesetzt ist.

CacheSize

Legt die Größe des Konfigurations-Caches in Bytes fest. Die Größe des gemeinsam genutzten Speichers zum Speichern von Host- und Datenpunktdaten.

Standard: 32M
 Bereich: 128K-64G

ConfigFrequency

Dieser Parameter ist **veraltet** (verwenden Sie stattdessen `ProxyConfigFrequency`).
Legt die Häufigkeit des Abrufs von Konfigurationsdaten vom Zabbix Server fest (in Sekunden).
Aktiver Proxy-Parameter. Wird für passive Proxies ignoriert (siehe Parameter `ProxyMode`).

Standard: 3600
 Bereich: 1-604800

DataSenderFrequency

Legt die Häufigkeit fest, mit der gesammelte Daten vom Proxy an den Zabbix Server gesendet werden. Beachten Sie, dass ein aktiver Proxy den Zabbix Server weiterhin jede Sekunde nach Aufgaben für Remote-Befehle abfragt.
Parameter für aktiven Proxy. Wird für passive Proxys ignoriert (siehe Parameter `ProxyMode`).

Standard: 1
 Bereich: 1-3600

DBHost

Geben Sie den Datenbank-Host (oder das Socket-Verzeichnis) an.

Für MySQL:

- `localhost` oder eine leere Zeichenfolge verwendet den standardmäßigen UNIX-Domain-Socket.

Für PostgreSQL:

- `localhost` wird über DNS aufgelöst (typischerweise zu `127.0.0.1`).
- Eine leere Zeichenfolge verwendet den standardmäßigen UNIX-Domain-Socket.
- Ein Pfad (zum Beispiel `/var/run/pgbouncer`) führt zur Verwendung des UNIX-Sockets an diesem Pfad.
- Eine durch Kommas getrennte Liste kann mehrere `host:port`-Werte enthalten. Jeder Host wird der Reihe nach versucht, bis eine Lese-/Schreibverbindung hergestellt ist. Beispiel: `DBHost=localhost:5431,127.0.0.1:20051,zabbix.domain,[::1]:30`

Standard: `localhost`

DBName

Geben Sie den Datenbanknamen oder den Pfad zur Datenbankdatei für SQLite3 an (die Multiprozessarchitektur von Zabbix erlaubt nicht die Verwendung einer In-Memory-Datenbank, z. B. `:memory:`, `file::memory:?cache=shared` oder `file:memdb1?mode=memory&cache=shared`).
Warnung: Versuchen Sie nicht, dieselbe Datenbank zu verwenden, die der Zabbix Server verwendet.

Obligatorisch: Ja

DBPassword

Geben Sie das Datenbankpasswort an. Kommentieren Sie diese Zeile aus, wenn kein Passwort verwendet wird. Wird bei SQLite ignoriert.

DBPort

Geben Sie den Port des Zabbix-Datenbank-Hosts an (siehe DBHost). Wird bei SQLite ignoriert.
 DBSocket und DBPort schließen sich in der Proxy-Konfiguration gegenseitig aus. Geben Sie nur einen von beiden an oder lassen Sie beide undefiniert.

Standard für MySQL: 3306

Standard für PostgreSQL: 5432

Bereich: 1024-65535

DBSchema

Geben Sie den Namen des Datenbankschemas an. Wird für PostgreSQL verwendet.

DBSocket

Geben Sie den Pfad zur MySQL-Socket-Datei an.
DBSocket und DBPort schließen sich in der Proxy-Konfiguration gegenseitig aus. Geben Sie nur einen von beiden an oder lassen Sie beide undefiniert.

DBUser

Geben Sie den Datenbankbenutzer an. Wird für SQLite ignoriert.

DBTLSConnect

Geben Sie einen Wert an, um eine TLS-Verbindung zur Datenbank zu erzwingen: - *required* - Verbindung über TLS herstellen; - *verify_ca* - Verbindung über TLS herstellen und das Zertifikat überprüfen; - *verify_full* - Verbindung über TLS herstellen, das Zertifikat überprüfen und prüfen, ob die durch DBHost angegebene Datenbankidentität mit ihrem Zertifikat übereinstimmt
Bei MySQL ab Version 5.7.11 und PostgreSQL werden die folgenden Werte unterstützt: ”required”, ”verify”, ”verify_full”.
Bei MariaDB ab Version 10.2.6 werden die Werte ”required” und ”verify_full” unterstützt.
Standardmäßig ist keine Option gesetzt, und das Verhalten hängt von der Datenbankkonfiguration ab.

DBTLSCAFile

Geben Sie den vollständigen Pfadnamen der Datei an, die die Zertifikate der obersten CA(s) für die Datenbankzertifikatsprüfung enthält.

Pflicht: nein (ja, wenn DBTLSConnect auf *verify_ca* oder *verify_full* gesetzt ist)

DBTLSCertFile

Geben Sie den vollständigen Pfadnamen der Datei an, die das Zabbix-Proxy-Zertifikat zur Authentifizierung an der Datenbank enthält.

DBTLSKeyFile

Geben Sie den vollständigen Pfadnamen der Datei an, die den privaten Schlüssel für die Authentifizierung an der Datenbank enthält.

DBTLSCipher

Geben Sie die Liste der Verschlüsselungs-Cipher an, die der Zabbix-Proxy für TLS-Protokolle bis einschließlich TLS v1.2 zulässt. Nur für MySQL unterstützt.

DBTLSCipher13

Geben Sie die Liste der Verschlüsselungs-Cipher Suites an, die der Zabbix Proxy für das TLS-v1.3-Protokoll zulässt. Nur für MySQL unterstützt, beginnend mit Version 8.0.16.

DebugLevel

Legt den Debug-Level fest: - *0* - grundlegende Informationen zum Starten und Stoppen von Zabbix-Prozessen; - *1* - kritische Informationen; - *2* - Fehlerinformationen; - *3* - Warnungen; - *4* - für das Debugging (liefert viele Informationen); - *5* - erweitertes Debugging (liefert noch mehr Informationen).

Standard: 3
 Bereich: 0-5

EnableRemoteCommands

Erlaubt Remote-Befehle vom Zabbix Server.

Standard: 0
 Werte: 0 - nicht erlauben; 1 - erlauben

ExternalScripts

Geben Sie den Speicherort des externen Skripts an (abhängig von der Installationsvariable `datadir` zur Kompilierzeit).

Standard: `/usr/local/share/zabbix/externalscripts`

Fping6Location

Geben Sie den Speicherort von fping6 an. Stellen Sie sicher, dass das fping6-Binary dem Benutzer root gehört und das SUID-Flag gesetzt ist. Lassen Sie den Wert leer (`Fping6Location=`), wenn Ihr `fping`-Dienstprogramm IPv6-Adressen verarbeiten kann.

Standard: `/usr/sbin/fping6`

FpingLocation

Geben Sie den Speicherort von fping an. Stellen Sie sicher, dass das fping-Binary dem Benutzer root gehört und das SUID-Flag gesetzt ist.

Standard: `/usr/sbin/fping`

HistoryCacheSize

Legt die Größe des History-Caches in Bytes fest. Die Größe des gemeinsam genutzten Speichers zum Speichern von Verlaufsdaten.

Standard: `16M`
 Bereich: 128K-16G

HistoryIndexCacheSize

Legt die Größe des History-Index-Caches in Bytes fest. Die Größe des gemeinsam genutzten Speichers für die Indizierung der im History-Cache gespeicherten History-Daten. Die Größe des Index-Caches benötigt ungefähr 100 Bytes, um einen Datenpunkt zwischenzuspeichern.

Standard: `4M`
 Bereich: 128K-16G

Hostname

Geben Sie einen eindeutigen, groß-/kleinschreibungssensitiven Proxy-Namen an. Stellen Sie sicher, dass der Proxy-Name dem Server bekannt ist.
Zulässige Zeichen: alphanumerisch, '.', ' ', '_' und '-'. Maximale Länge: 128

Standard: Festgelegt durch HostnameItem

HostnameItem

Geben Sie einen Datenpunkt zum Setzen von `Hostname` an (falls nicht definiert).
 Dieser Datenpunkt wird auf dem Proxy genauso ausgeführt wie auf einem Agent.
 Wird ignoriert, wenn `Hostname` gesetzt ist.
Unterstützt keine Benutzerparameter, Performancezähler oder Aliase; unterstützt jedoch `system.run[]`.

Standard: `system.hostname`

HousekeepingFrequency

Legt die Häufigkeit des Housekeeping-Vorgangs fest (in Stunden). Housekeeping entfernt veraltete Informationen aus der Datenbank.
Hinweis: Um die Last beim Start des Proxy zu verringern, wird Housekeeping für 30 Minuten nach dem Start des Proxy verschoben. Wenn `HousekeepingFrequency` also 1 ist, wird der allererste Housekeeping-Vorgang nach dem Start des Proxy nach 30 Minuten ausgeführt und danach stündlich wiederholt.
Es ist möglich, automatisches Housekeeping zu deaktivieren, indem HousekeepingFrequency auf 0 gesetzt wird. In diesem Fall kann der Housekeeping-Vorgang nur über die Laufzeitsteuerungsoption `housekeeper_execute` gestartet werden.
Siehe auch die Optionen und Details zur Laufzeitsteuerung sowie zum Housekeeping-Vorgang.

Standard: 1
 Bereich: 0-24

Include

Geben Sie ein Verzeichnis oder einzelne Dateien an, die in die Konfigurationsdatei aufgenommen werden sollen.
 Während der Zabbix-Installation wird dieser Parameter auf `/usr/local/etc` gesetzt, sofern Sie dies nicht zur Kompilierzeit ändern.

Allgemeine Regeln für den Parameter `Include`:

- Der Pfad kann relativ zum Speicherort der Datei `zabbix_proxy.conf` angegeben werden.
- Sie können mehrere `Include`-Parameter in einer Konfigurationsdatei angeben.
- Zabbix verarbeitet den `Include`-Parameter, sobald er gelesen wird. Platzieren Sie ihn daher am besten am Ende der Konfigurationsdatei.

Regeln für eingebundene Dateien:

- Jede Datei muss lesbar sein.
- Jede Datei muss eine korrekte Syntax haben. Andernfalls wird die Komponente nicht gestartet.

Regeln für das Einbinden von Verzeichnissen oder Dateimustern:

- Gehen Sie nicht von einer bestimmten Reihenfolge beim Einbinden aus (z. B. werden Dateien nicht alphabetisch eingebunden).
- Definieren Sie denselben Parameter nicht in mehr als einer eingebundenen Datei (z. B. um eine allgemeine Einstellung durch eine spezifische zu überschreiben), da Sie nicht vorhersagen können, welcher Wert verwendet wird.
- Achten Sie auf Sicherungskopien, die einige Texteditoren automatisch erstellen (z. B. kann beim Bearbeiten von `include/my_specific.conf` die Datei `include/my_specific.conf.BAK` erstellt werden), da beide Dateien eingebunden werden.

Um nur relevante Dateien aus einem Verzeichnis einzubinden, verwenden Sie den Platzhalter (*) für die Mustererkennung.

Beispiel:

```
Include=/absolute/path/to/config/files/*.conf
```

JavaGateway

Geben Sie die IP-Adresse oder den Hostnamen des Zabbix Java gateway an. Nur erforderlich, wenn Java-Poller gestartet sind.

JavaGatewayPort

Geben Sie den Port des Zabbix Java gateway an.

Standard: 10052
 Bereich: 1024-32767

ListenBacklog

Legt die maximale Anzahl ausstehender Verbindungen in der TCP-Warteschlange fest.
Der Standardwert ist eine fest codierte Konstante, die vom System abhängt.
Der maximal unterstützte Wert hängt ebenfalls vom System ab; zu hohe Werte können stillschweigend auf das „implementierungsabhängige Maximum“ gekürzt werden.

Standard: SOMAXCONN
 Bereich: 0 - INT_MAX

ListenIP

Geben Sie die Trap-Listener-IP-Adresse an. Eine durch Kommas getrennte Liste.
Der Trapper lauscht auf allen Netzwerkinterfaces, wenn dieser Parameter nicht angegeben ist.

Standard: 0.0.0.0

ListenPort

Geben Sie den Trapper-Listen-Port an.

Standard: 10051
 Bereich: 1024-32767

LoadModule

Geben Sie das beim Start des Proxys zu ladende Modul an.
Module werden verwendet, um die Funktionalität des Proxys zu erweitern.
Das Modul muss sich in dem durch LoadModulePath angegebenen Verzeichnis befinden, oder der Pfad muss dem Modulnamen vorangestellt werden.
Wenn der vorangestellte Pfad absolut ist (mit / beginnt), wird LoadModulePath ignoriert.
 Formate: - LoadModule=<module.so> - LoadModule=<path/module.so> - LoadModule=</abs_path/module.so>

Es können mehrere LoadModule-Parameter angegeben werden.

LoadModulePath

Der vollständige Pfad zum Speicherort der Proxy-Module. Die Standardeinstellung hängt von den Kompilierungsoptionen ab.

LogFile

Geben Sie die Proxy-Logdatei an.

Erforderlich: Ja, wenn LogType auf `file` gesetzt ist; andernfalls nein

LogFileSize

Legt die maximale Größe der Protokolldatei fest (in MB).
0 - automatische Protokollrotation deaktivieren.
Hinweis: Wenn die Größenbegrenzung der Protokolldatei erreicht wird und die Dateirotation aus irgendeinem Grund fehlschlägt, wird die vorhandene Protokolldatei gekürzt und neu begonnen.

Standard: 1
 Bereich: 0-1024

LogRemoteCommands

Aktivieren Sie die Protokollierung ausgeführter Shell-Befehle als Warnungen.

Standard: 0
 Werte: 0 - deaktiviert, 1 - aktiviert

LogSlowQueries

Legt die maximale Dauer fest, die eine Datenbankabfrage vor dem Protokollieren benötigen darf (in Millisekunden).
0 - langsame Abfragen nicht protokollieren.
Diese Option wird ab `DebugLevel=3` aktiviert.

Standard: 0
 Bereich: 0-3600000

LogType

Geben Sie den Typ der Protokollausgabe an: - *file* - Protokoll in die durch den Parameter `LogFile` angegebene Datei schreiben; - *system* - Protokoll in syslog schreiben; - *console* - Protokoll in die Standardausgabe schreiben.

Standardwert: `file`

MaxConcurrentChecksPerPoller

Legt die maximale Anzahl asynchroner Prüfungen fest, die von jedem HTTP-Agent-Poller, Agent-Poller oder SNMP-Poller gleichzeitig ausgeführt werden können. Siehe `StartHTTPAgentPollers`, `StartAgentPollers` und `StartSNMPPollers`.

Standard: 1000
 Bereich: 1-1000

PidFile

Geben Sie die PID-Datei an.

Standard: `/tmp/zabbix_proxy.pid`

ProxyBufferMode

Geben Sie den Speichermechanismus für Verlaufs-, Erkennungs- und Autoregistrierungsdaten an: - *disk* - Daten werden in der Datenbank gespeichert und aus der Datenbank hochgeladen; - *memory* - Daten werden im Speicher gespeichert und aus dem Speicher hochgeladen. Wenn der Puffer keinen Speicher mehr hat, werden die alten Daten verworfen. Beim Herunterfahren wird der Puffer verworfen. - *hybrid* - der Proxy-Puffer arbeitet normalerweise wie im Speichermodus, bis der Speicher erschöpft ist oder der älteste Datensatz das konfigurierte Alter überschreitet. Wenn das passiert, wird der Puffer in die Datenbank geleert und arbeitet wie im Festplattenmodus, bis alle Daten hochgeladen wurden, und beginnt dann wieder mit dem Speicher zu arbeiten. Beim Herunterfahren wird der Speicherpuffer in die Datenbank geleert.

Siehe auch: Proxy memory buffer.

Standard: `disk`
 Werte: disk; memory; hybrid

ProxyConfigFrequency

Legt die Häufigkeit fest, mit der Konfigurationsdaten vom Zabbix Server in Sekunden abgerufen werden.
Parameter für den aktiven Proxy. Wird für passive Proxys ignoriert (siehe Parameter `ProxyMode`).

Standard: 10
 Bereich: 1-604800

ProxyLocalBuffer

Legt die Dauer fest, für die Daten lokal aufbewahrt werden (in Stunden), auch wenn die Daten bereits mit dem Server synchronisiert wurden..
Dieser Parameter kann verwendet werden, wenn lokale Daten von Drittanbieteranwendungen genutzt werden sollen.

Standard: 0
 Bereich: 0-720

ProxyMemoryBufferAge

Legt das maximale Alter der Daten im Proxy-Speicherpuffer in Sekunden fest.
 Wenn diese Option aktiviert ist (nicht null) und Datensätze im Proxy-Speicherpuffer älter sind, erzwingt sie, dass der Proxy-Puffer in den Datenbankmodus wechselt, bis alle Datensätze auf den Server hochgeladen wurden. Dieser Parameter muss kleiner oder gleich dem Parameter `ProxyOfflineBuffer` sein.

Standard: 0
 Bereich: 0;600-864000

ProxyMemoryBufferSize

Legt die Größe des gemeinsam genutzten Speichercaches für gesammelte Verlaufs-, Discovery- und Auto-Registrierungsdaten fest (in Bytes). Wenn aktiviert (nicht null), hält der Proxy Verlaufs-, Discovery- und Auto-Registrierungsdaten im Speicher, sofern der Cache nicht voll ist oder die gespeicherten Datensätze nicht älter als das definierte `ProxyMemoryBufferAge` sind. Dieser Parameter kann nicht zusammen mit dem Parameter `ProxyLocalBuffer` verwendet werden.

Standard: 0
 Bereich: 0;128K-2G

ProxyMode

Legt den Betriebsmodus des Proxy fest.
0 - Proxy im aktiven Modus
1 - Proxy im passiven Modus
Hinweis: (sensible) Proxy-Konfigurationsdaten können für Parteien verfügbar werden, die Zugriff auf den Trapper-Port des Zabbix-Servers haben, wenn ein aktiver Proxy verwendet wird. Dies ist möglich, weil sich jeder als aktiver Proxy ausgeben und Konfigurationsdaten anfordern kann; eine Authentifizierung findet nicht statt.

Standard: 0
 Bereich: 0-1

ProxyOfflineBuffer

Legt die Dauer fest, für die Daten bei fehlender Verbindung zum Zabbix Server aufbewahrt werden (in Stunden).
Ältere Daten gehen verloren.

Standard: 1
 Bereich: 1-720

Server

Wenn `ProxyMode` auf `active` gesetzt ist: Geben Sie die IP-Adresse oder den DNS-Namen des Zabbix-Servers (Adresse:Port) oder Cluster (Adresse:Port;Adresse2:Port) an, von dem Konfigurationsdaten abgerufen und an den Daten gesendet werden sollen.
Wenn kein Port angegeben ist, wird der Standardport verwendet.
Cluster-Knoten müssen durch ein Semikolon getrennt werden.

Wenn `ProxyMode` auf `passive` gesetzt ist: Geben Sie IP-Adressen, optional in CIDR-Notation, oder DNS-Namen des Zabbix-Servers an. Eine durch Kommas getrennte Liste.
 Eingehende Verbindungen werden nur von den hier aufgeführten Adressen akzeptiert. Wenn IPv6-Unterstützung aktiviert ist, werden `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` als gleich behandelt.
`::/0` erlaubt jede IPv4- oder IPv6-Adresse. `0.0.0.0/0` kann verwendet werden, um jede IPv4-Adresse zuzulassen.

Beispiel:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Pflicht: ja

SNMPTrapperFile

Geben Sie die temporäre Datei an, die zum Übergeben von Daten vom SNMP-Trap-Daemon an den Proxy verwendet wird.
Sie muss mit der in `zabbix_trap_receiver.pl` oder der SNMPTT-Konfigurationsdatei angegebenen Datei übereinstimmen.

Standard: `/tmp/zabbix_traps.tmp`

SocketDir

Geben Sie das Verzeichnis an, in dem die von internen Zabbix-Diensten verwendeten IPC-Sockets gespeichert werden.

Standard: `/tmp`

SourceIP

Geben Sie die Quell-IP-Adresse an für:

- ausgehende Verbindungen zum Zabbix Server
- Agentless-Verbindungen (VMware, SSH, JMX, SNMP, Telnet und einfache Prüfungen)
- Verbindungen des HTTP-Agenten
- JavaScript-HTTP-Anfragen von Skript-Datenpunkten
- JavaScript-HTTP-Anfragen in der Vorverarbeitung
- Verbindungen zum Vault

SSHKeyLocation

Geben Sie den Speicherort der öffentlichen und privaten Schlüssel für SSH-Prüfungen und -Aktionen an.

SSLCertLocation

Geben Sie den Speicherort der SSL-Clientzertifikatdateien für die Clientauthentifizierung an.
Dieser Parameter wird beim Web-Monitoring, für die Kommunikation mit Vault und für die Telemetriedatenquelle ClickHouse verwendet.

SSLKeyLocation

Geben Sie den Speicherort der privaten SSL-Schlüsseldateien für die Client-Authentifizierung an.
Dieser Parameter wird beim Web-Monitoring, für die Kommunikation mit Vault und für die Telemetriedatenquelle ClickHouse verwendet.

SSLCALocation

Geben Sie den Speicherort der Dateien der Zertifizierungsstelle (CA) für die Überprüfung von SSL-Serverzertifikaten an.
Beachten Sie, dass der Wert dieses Parameters als `CURLOPT_CAPATH`-Option von libcurl festgelegt wird. Bei libcurl-Versionen vor 7.42.0 hat dies nur Auswirkungen, wenn libcurl für die Verwendung von OpenSSL kompiliert wurde. Weitere Informationen finden Sie auf der cURL-Webseite.
Dieser Parameter wird bei der Webüberwachung, der SMTP-Authentifizierung, HTTP-Agent-Datenpunkten, der Kommunikation mit Vault und für die Telemetriedatenquelle ClickHouse verwendet.

StartAgentPollers

Legt die Anzahl der vorab erzeugten Instanzen von Zabbix Agent `pollers` fest. Siehe `MaxConcurrentChecksPerPoller`.

Standard: 1
 Bereich: 0-1000

StartAPMCollector

Auf 1 setzen, um eine vorab gestartete Instanz eines APM-Collectors (die Prozesse `apm manager` und `apm worker`) zu starten.

Standard: 0
 Bereich: 0-1

StartBrowserPollers

Legt die Anzahl der vorab erzeugten Instanzen von Browser-Datenpunkt-`pollers` fest.

Standard: 1
 Bereich: 0-1000

StartDBSyncers

Legt die Anzahl der vorab per Fork erzeugten Instanzen von `history syncers` fest.
Hinweis: Seien Sie vorsichtig, wenn Sie diesen Wert ändern; eine Erhöhung kann mehr schaden als nützen.

Standard: 4
 Bereich: 1-100

StartDiscoverers

Legt die Anzahl der vorab erzeugten Instanzen von `discovery workers` fest.

Standard: 5
 Bereich: 0-1000

StartHTTPAgentPollers

Legt die Anzahl der vorab erzeugten Instanzen von `HTTP agent pollers` fest. Siehe `MaxConcurrentChecksPerPoller`.

Standard: 1
 Bereich: 0-1000

StartHTTPPollers

Legt die Anzahl der vorab erzeugten Instanzen von `HTTP pollers` fest.

Standard: 1
 Bereich: 0-1000

StartIPMIPollers

Legt die Anzahl der vorab erzeugten Instanzen von `IPMI pollers` fest.

Standard: 0
 Bereich: 0-1000

StartJavaPollers

Legt die Anzahl der vorab geforkten Instanzen von `Java pollers` fest.

Standard: 0
 Bereich: 0-1000

StartODBCPollers

Legt die Anzahl der vorab erzeugten Instanzen von `ODBC pollers` fest.

Standard: 1
 Bereich: 0-1000

StartPingers

Legt die Anzahl der vorab per Fork erzeugten Instanzen von `ICMP pingers` fest.

Standard: 1
 Bereich: 0-1000

StartPollersUnreachable

Legt die Anzahl der vorab erzeugten Instanzen von Pollern für nicht erreichbare Hosts fest (einschließlich IPMI und Java). Mindestens ein Poller für nicht erreichbare Hosts muss ausgeführt werden, wenn reguläre, IPMI- oder Java-Poller gestartet werden.

Standard: 1
 Bereich: 0-1000

StartPollers

Legt die Anzahl der vorab erzeugten Instanzen von `pollers` fest.

Standard: 5
 Bereich: 0-1000

StartPreprocessors

Setzen Sie die Anzahl der vorab gestarteten Instanzen der Preprocessing-worker-Threads auf mindestens die Anzahl der verfügbaren CPU-Kerne. Es sollten mehr Worker festgelegt werden, wenn das Preprocessing nicht CPU-gebunden ist und viele Netzwerkanfragen hat.

Standard: 16
 Bereich: 1-1000

StartSNMPPollers

Legt die Anzahl der vorab erzeugten Instanzen von SNMP pollers fest. Siehe MaxConcurrentChecksPerPoller.

Standard: 1
 Bereich: 0-1000

StartTelemetryQueryPollers

Legt die Anzahl der vorab gestarteten Instanzen asynchroner Telemetrieabfrage-Poller fest.
Siehe MaxConcurrentChecksPerPoller.

Standard: 1
 Bereich: 0-1000

StartSNMPTrapper

Auf "1" setzen, um einen SNMP trapper-Prozess zu starten.

Standard: 0
 Bereich: 0-1

StartTrappers

Legt die Anzahl der vorab erzeugten Instanzen von trappers fest.
Trappers akzeptieren eingehende Verbindungen von Zabbix sender und aktiven Agenten.

Standard: 5
 Bereich: 0-1000

StartVMwareCollectors

Legt die Anzahl der vorab erzeugten Instanzen von vmware collector fest.

Standard: 0
 Bereich: 0-250

StatsAllowedIP

Geben Sie IP-Adressen (optional in CIDR-Notation) oder DNS-Namen externer Zabbix-Instanzen an, die berechtigt sind, Statistiken anzufordern. Eine durch Kommas getrennte Liste.
 Wenn dieser Parameter nicht gesetzt ist, werden keine Statistik-Anfragen akzeptiert.
Wenn die IPv6-Unterstützung aktiviert ist, werden `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` als gleich behandelt, und `::/0` erlaubt jede IPv4- oder IPv6-Adresse. `0.0.0.0/0` kann verwendet werden, um jede IPv4-Adresse zuzulassen.

Beispiel:

```
StatsAllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

TelemetryProvider

Legen Sie die Telemetriedatenquelle im Format `<provider>[;<options>]` fest, wobei gilt:

- `<provider>` - eine der unterstützten Telemetriedatenquellen: `clickhouse`
- `<options>` - durch Kommas getrennte `key=value`-Paare. Werte können in Anführungszeichen gesetzt werden.

Optionen für ClickHouse:

- `url` - legt die HTTP- oder HTTPS-URL fest, die für die Verbindung zur Telemetriedatenquelle verwendet wird.
- `db` - legt den Namen der Datenbank fest.
- `username` - legt den Benutzernamen fest, der für die Anmeldung an der Datenbank verwendet wird.
- `password` - legt das Kennwort fest, das für die Anmeldung an der Datenbank verwendet wird.
- `ssl_cert_file`, `ssl_key_file`, `ssl_key_password`, `ssl_verify_peer`, `ssl_verify_host`, `ssl_ca_location`, `ssl_cert_location`, `ssl_key_location` - legen Sie diese Optionen fest, um eine sichere (TLS/SSL-)Verbindung zur Datenbank zu konfigurieren.
- `source_ip` - legt die Quell-IP-Adresse fest, die für ausgehende Verbindungen zur Telemetriedatenquelle verwendet wird.
- `vault_path` - legt den Pfad zu einem Vault-Geheimnis fest, das die Werte `username` und `password` enthält. Verwenden Sie diese Option nicht zusammen mit den Optionen `username` und `password`.

Beispiel für ClickHouse:

```
TelemetryProvider=clickhouse;url=http://localhost:8123,db=zabbix,username=zabbix,password=zabbix
```

Beispiel für ClickHouse mit Vault:

```
TelemetryProvider=clickhouse;url=http://localhost:8123,db=zabbix,vault_path=clickhouse
```

Timeout

Geben Sie an, wie lange in Sekunden auf den Verbindungsaufbau und den Datenaustausch mit dem Zabbix Server, Agent, Webdienst und Legacy-SNMP-Prüfungen (einzelne OID-Nummer oder Zeichenfolge) gewartet werden soll.

Dieser Parameter definiert die Dauer für verschiedene Kommunikationsvorgänge:

- Ausführung von Remote-Befehlen auf dem Zabbix Agent
- Ausführung von SSH-/Telnet-Befehlen
- Neuplanung von Datenpunkten, wenn die IPMI-Schnittstelle nicht mehr verfügbar ist
- Senden einer Antwort an den Zabbix Server, wenn der Datenaustausch aufgrund von Berechtigungs- oder Verschlüsselungsproblemen fehlschlägt
- Frist für IPC-Async-Sockets und Laufzeitsteuerungsoptionen
- DNS-Anfragen des asynchronen Pollers
- Antwort auf den Heartbeat der aktiven Prüfung
- Abruf von Zabbix Agent-Daten (Werten) von aktiven Agenten
- Abruf von Daten vom Zabbix Sender
- Senden der Liste aktiver Prüfungen an den Zabbix Agent

Dieses Timeout wird **nicht** für jene Prüfungen verwendet, für die im Frontend Einstellungen für flexibles Timeout konfiguriert sind (auf globaler Ebene, Proxy-Ebene oder pro Datenpunkt). Beispielsweise verwenden SNMP-`walk[OID]`- und `get[OID]`-Datenpunkte das im Frontend konfigurierte Timeout; Legacy-SNMP-Prüfungen verwenden weiterhin den Timeout-Wert des Servers.

Standard: 3
 Bereich: 1-30

TLSAccept

Geben Sie den Grad der Verschlüsselung an, der für eingehende Verbindungen vom Zabbix Server erforderlich ist. Wird für einen passiven Proxy verwendet, bei einem aktiven Proxy ignoriert. Es können mehrere Werte angegeben werden, getrennt durch Kommas: - *unencrypted* - Verbindungen ohne Verschlüsselung akzeptieren (Standard); - *psk* - Verbindungen mit TLS und einem vorab gemeinsam genutzten Schlüssel (PSK) akzeptieren; - *cert* - Verbindungen mit TLS und einem Zertifikat akzeptieren.

Pflicht: ja für einen passiven Proxy, wenn TLS-Zertifikat- oder PSK-Parameter definiert sind (auch für *unencrypted* Verbindung); andernfalls nein

TLSCAFile

Geben Sie den vollständigen Pfadnamen der Datei an, die die Zertifikate der obersten CA(s) für die Überprüfung von Peer-Zertifikaten enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

TLSCertFile

Geben Sie den vollständigen Pfadnamen der Datei an, die das Proxy-Zertifikat oder die Zertifikatskette enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

TLSCipherAll

Geben Sie die GnuTLS-Prioritätszeichenfolge oder die OpenSSL-(TLS 1.2)-Cipher-Zeichenfolge an. Überschreiben Sie die standardmäßigen Auswahlkriterien für die Ciphersuite für zertifikats- und PSK-basierte Verschlüsselung.

Beispiel:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

Geben Sie die Cipher-String für OpenSSL 1.1.1 oder neuer in TLS 1.3 an. Überschreiben Sie die Standardauswahlkriterien für die Ciphersuite für zertifikats- und PSK-basierte Verschlüsselung.

Beispiel für GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL
```

Beispiel für OpenSSL:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherCert

Geben Sie den GnuTLS-Prioritätsstring oder den OpenSSL-(TLS 1.2)-Cipher-String an. Überschreiben Sie die standardmäßigen Auswahlkriterien für die Cipher Suite bei zertifikatsbasierter Verschlüsselung.

Beispiel für GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Beispiel für OpenSSL:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

TLSCipherCert13

Geben Sie die Cipher-String für OpenSSL 1.1.1 oder neuer in TLS 1.3 an. Überschreiben Sie die standardmäßigen Auswahlkriterien für die Cipher Suite bei zertifikatsbasierter Verschlüsselung.

TLSCipherPSK

Geben Sie die GnuTLS-Prioritätszeichenfolge oder die Cipher-String von OpenSSL (TLS 1.2) an. Überschreiben Sie die standardmäßigen Auswahlkriterien für die Cipher Suite bei PSK-basierter Verschlüsselung.

Beispiel für GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Beispiel für OpenSSL:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

Geben Sie die Cipher-String für OpenSSL 1.1.1 oder neuer in TLS 1.3 an. Überschreiben Sie die Standardkriterien für die Auswahl der Cipher Suite für PSK-basierte Verschlüsselung.

Beispiel:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Geben Sie den für Proxy-Verbindungen zum Zabbix Server erforderlichen Verschlüsselungsgrad an. Wird für einen aktiven Proxy verwendet, bei einem passiven Proxy ignoriert. Es kann nur ein Wert angegeben werden: - *unencrypted* - ohne Verschlüsselung verbinden (Standard); - *psk* - mit TLS und einem vorab gemeinsam genutzten Schlüssel (PSK) verbinden; - *cert* - mit TLS und einem Zertifikat verbinden.

Pflicht: ja für aktiven Proxy, wenn TLS-Zertifikat- oder PSK-Parameter definiert sind (auch bei *unencrypted* Verbindung); andernfalls nein

TLSCRLFile

Geben Sie den vollständigen Pfadnamen der Datei an, die widerrufene Zertifikate enthält. Dieser Parameter wird für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet.

TLSKeyFile

Geben Sie den vollständigen Pfadnamen der Datei an, die den privaten Schlüssel des Proxy enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

TLSListen

TLS-only-Verbindungen auf dem Trapper-Socket erzwingen.

Unterstützte Werte:

- required - nur TLS-Verbindungen akzeptieren

TLSPSKFile

Geben Sie den vollständigen Pfadnamen der Datei an, die den vorab gemeinsam genutzten Schlüssel des Proxy enthält, der für verschlüsselte Kommunikation mit dem Zabbix-Server verwendet wird.

TLSPSKIdentity

Geben Sie die Identitätszeichenfolge des vorab gemeinsam genutzten Schlüssels an, die für verschlüsselte Kommunikation mit dem Zabbix Server verwendet wird.

TLSServerCertIssuer

Geben Sie den zulässigen Aussteller des Serverzertifikats an.

TLSServerCertSubject

Geben Sie den zulässigen Betreff des Serverzertifikats an.

TmpDir

Geben Sie das temporäre Verzeichnis an.

Standard: `/tmp`

TrapperTimeout

Geben Sie das Timeout in Sekunden an für:
 - das Abrufen von Konfigurationsdaten vom Zabbix-Server;
 - die Ausführung globaler Skripte oder die Ausführung entfernter Befehle.

Standard: 300
 Bereich: 1-300

UnavailableDelay

Legt die Häufigkeit der Überprüfung der Host-Verfügbarkeit während des Zeitraums der Nichtverfügbarkeit in Sekunden fest.

Standard: 60
 Bereich: 1-3600

UnreachableDelay

Legt die Häufigkeit der Überprüfung der Host-Verfügbarkeit während des [Unreachable]-Zeitraums (/manual/appendix/items/unreachability#ur interface) in Sekunden fest.

Standard: 15
 Bereich: 1-3600

UnreachablePeriod

Legt fest, nach wie vielen Sekunden der Unerreichbarkeit der Host als nicht verfügbar behandelt wird.

Standard: 45
 Bereich: 1-3600

Benutzer

Die Berechtigungen auf einen bestimmten, vorhandenen Benutzer auf dem System herabsetzen.
Wirkt nur, wenn es als `root` ausgeführt wird und `AllowRoot` deaktiviert ist.

Standard: `zabbix`

Vault

Geben Sie den Vault-Anbieter an: - *HashiCorp* - HashiCorp KV Secrets Engine Version 2; - *CyberArk* - CyberArk Central Credential Provider
muss mit dem im Frontend festgelegten Vault-Anbieter übereinstimmen.

Standard: `HashiCorp`

VaultAppRoleID

Geben Sie die HashiCorp Vault AppRole-ID an.
 Wenn festgelegt, wird die AppRole-Authentifizierung aktiviert. Kann nicht zusammen mit VaultToken verwendet werden.
 Wird nur für HashiCorp Vault unterstützt.

VaultAppSecretID

Geben Sie die Secret ID der HashiCorp Vault AppRole an.
 Nur für HashiCorp Vault unterstützt.

VaultDBPath

Geben Sie den Speicherort an, von dem Datenbankanmeldedaten anhand von Schlüsseln abgerufen werden. Je nach Vault handelt es sich dabei um einen Vault-Pfad oder eine Abfrage.

Die für HashiCorp verwendeten Schlüssel sind `password` und `username`.

Beispielpfad mit `VaultPrefix=/v1/secret/data/zabbix/`:

```
database
```

Beispielpfad ohne VaultPrefix:

```
secret/zabbix/database
```

Die für CyberArk verwendeten Schlüssel sind `Content` und `UserName`.

Beispiel:

```
AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix_proxy_database
```

Diese Option kann nur verwendet werden, wenn `DBUser` und `DBPassword` nicht angegeben sind.

VaultPrefix

Geben Sie ein benutzerdefiniertes Präfix für den Vault-Pfad oder die Abfrage an, abhängig vom Vault. Wenn nicht angegeben, werden die am besten geeigneten Standardwerte verwendet.
Beachten Sie, dass 'data' bei HashiCorp automatisch nach dem Mountpoint angehängt wird, wenn `VaultPrefix` nicht angegeben ist.
Beachten Sie, dass 'data' bei HashiCorp automatisch nach dem Mountpoint angehängt wird, wenn `VaultPrefix` nicht angegeben ist.

Beispiel für ein Präfix für HashiCorp:

```
/v1/secret/data/zabbix/
```

Beispiel für ein Präfix für Cyberark:

```
/AIMWebService/api/Accounts?
```

VaultTLSCertFile

Geben Sie den Namen der SSL-Zertifikatsdatei an, die für die Client-Authentifizierung verwendet wird. Die Zertifikatsdatei muss im PEM1-Format vorliegen.
Wenn die Zertifikatsdatei auch den privaten Schlüssel enthält, lassen Sie das Feld für die SSL-Schlüsseldatei leer.
Das Verzeichnis, das diese Datei enthält, wird durch den Konfigurationsparameter SSLCertLocation angegeben.
Diese Option kann weggelassen werden, wird jedoch für den CyberArkCCP-Vault empfohlen.

VaultTLSKeyFile

Geben Sie den Namen der SSL-Privatschlüsseldatei an, die für die Client-Authentifizierung verwendet wird. Die Privatschlüsseldatei muss im PEM1-Format vorliegen.
Das Verzeichnis, das diese Datei enthält, wird durch den Konfigurationsparameter SSLKeyLocation angegeben.
Diese Option kann weggelassen werden, wird jedoch für den CyberArkCCP-Vault empfohlen.

VaultToken

Geben Sie das HashiCorp-Vault-Authentifizierungstoken an, das ausschließlich für den Zabbix-Proxy mit Lesezugriff auf den im optionalen Konfigurationsparameter `VaultDBPath` angegebenen Pfad erstellt worden sein sollte.
Es ist ein Fehler, wenn `VaultToken` und die Umgebungsvariable `VAULT_TOKEN` gleichzeitig definiert sind.
 Kann nicht zusammen mit VaultAppRoleID verwendet werden.
 Wird nur für HashiCorp Vault unterstützt.

VaultURL

Geben Sie die URL des Vault-Servers an. Das systemweite Verzeichnis für CA-Zertifikate wird verwendet, wenn SSLCALocation nicht angegeben ist.

Standard: `https://127.0.0.1:8200`

VMwareCacheSize

Legt die Größe des gemeinsam genutzten Speichers zum Speichern von VMware-Daten fest.
Mit der internen Prüfung `zabbix[vmware,buffer,...]` kann die Auslastung des VMware-Cache überwacht werden (siehe Interne Prüfungen).
Beachten Sie, dass kein gemeinsam genutzter Speicher zugewiesen wird, wenn keine vmware-Collector-Instanzen für den Start konfiguriert sind.

Standard: `8M`
 Bereich: 256K-2G

VMwareFrequency

Legt die Häufigkeit der Datenerfassung von einem einzelnen VMware-Dienst fest (in Sekunden).
Diese Häufigkeit sollte auf das kleinste Aktualisierungsintervall eines beliebigen VMware-Überwachungsdatenpunkts gesetzt werden.

Standard: 60
 Bereich: 10-86400

VMwarePerfFrequency

Legt die Häufigkeit der Abfrage von Leistungsindikator-Statistiken von einem einzelnen VMware-Dienst fest (in Sekunden).
Diese Häufigkeit sollte auf das kleinste Aktualisierungsintervall eines beliebigen VMware-Monitoring-Datenpunkt gesetzt werden, der VMware-Leistungsindikatoren verwendet.

Standard: 60
 Bereich: 10-86400

VMwareTimeout

Legt die maximale Anzahl von Sekunden fest, die ein `vmware collector` auf eine Antwort vom VMware-Dienst (vCenter oder ESX-Hypervisor) wartet.

Standard: 10
 Bereich: 1-300

WebDriverURL

Geben Sie die URL der WebDriver-Schnittstelle an.

Beispiel (verwendet mit dem eigenständigen Selenium-WebDriver-Server):

```
WebDriverURL=http://localhost:4444
```

Agent

Übersicht Der Zabbix Agent wird auf einem Überwachungsziel bereitgestellt, um lokale Ressourcen und Anwendungen (Festplatten, Arbeitsspeicher, Prozessorstatistiken usw.) aktiv zu überwachen.

Der Agent erfasst Betriebsinformationen lokal und meldet die Daten zur weiteren Verarbeitung an den Zabbix Server. Bei Fehlern (z. B. wenn eine Festplatte voll läuft oder ein Dienstprozess abstürzt) kann der Zabbix Server die Administratoren des jeweiligen Computers, der den Fehler gemeldet hat, aktiv benachrichtigen.

Zabbix Agents sind aufgrund der Verwendung nativer Systemaufrufe zur Erfassung statistischer Informationen äußerst effizient.

Es ist möglich, mehrere Instanzen des Agents auf einem Host auszuführen.

Siehe auch:

- Protokolle für passive und aktive Prüfungen
- Agent als Root ausführen
- Konfigurationsparameter: Zabbix Agent
- Konfigurationsparameter: Zabbix Agent (Windows)
- Zabbix Agent unter Microsoft Windows
- Zabbix Agent unter IBM AIX

Passive und aktive Prüfungen

Zabbix Agents können passive und aktive Prüfungen durchführen:

- Passive Prüfungen - Der Zabbix Agent antwortet auf eine Anfrage vom Zabbix Server (oder Proxy). Zum Beispiel fordert der Server Daten an (z. B. CPU-Auslastung), und der Agent gibt das Ergebnis zurück.
- Aktive Prüfungen - Der Zabbix Agent sammelt und sendet Daten, ohne auf eine Anfrage vom Zabbix Server (oder Proxy) zu warten. Zuerst ruft er eine Liste der Überwachungsdatenpunkte vom Server ab (CPU-Auslastung, verfügbarer Speicher usw.), sammelt dann die erforderlichen Daten und sendet regelmäßig neue Werte zurück.

Der Typ der Agent-Prüfung wird durch Auswahl des entsprechenden Überwachungs-Datenpunkttyps konfiguriert. Der Zabbix Agent verarbeitet Datenpunkte vom Typ *Zabbix agent* oder *Zabbix agent (active)*.

Unterstützte Plattformen Informationen zu unterstützten Plattformen finden Sie auf der Seite Anforderungen.

Agent auf Unix-ähnlichen Systemen Der Zabbix Agent auf Unix-ähnlichen Systemen wird auf dem überwachten Host ausgeführt.

Installation

Der Zabbix Agent kann auf Linux-basierten Systemen mit einer der folgenden Methoden installiert werden:

- Zabbix-Pakete – wählen Sie die Komponente Agent aus (nachdem Sie Ihre Zabbix-Version, die Betriebssystemdistribution und die Betriebssystemversion ausgewählt haben) und befolgen Sie die bereitgestellten Anweisungen;
- Zabbix-Quellen – laden Sie die Quelldateien herunter und kompilieren Sie den Zabbix Agent, indem Sie die Quellen mit der Option `--enable-agent` konfigurieren.

> **Attention:**
> Im Allgemeinen funktionieren 32-Bit-Zabbix-Agents auf 64-Bit-Systemen, können jedoch in einigen Fällen fehlschlagen.

> **Note:**
> Vorkompilierte Binärdateien des Zabbix Agents stehen für macOS, IBM AIX, FreeBSD, OpenBSD und Solaris zum Download bereit. Legacy-Binärdateien, die mit der aktuellen Zabbix-Server-/Proxy-Version kompatibel sind, stehen für NetBSD und HP-UX bereit. Informationen zur Installation und Ausführung des Agents auf IBM AIX finden Sie unter Zabbix Agent auf IBM AIX.

Wenn als Paket installiert

Der Zabbix Agent läuft als Daemon-Prozess. Der Agent kann mit folgendem Befehl gestartet werden:

```
systemctl start zabbix-agent
```

Dies funktioniert auf den meisten GNU/Linux-Systemen. Auf anderen Systemen müssen Sie möglicherweise Folgendes ausführen:

```
/etc/init.d/zabbix-agent start
```

Um den Zabbix Agent zu stoppen, neu zu starten oder seinen Status zu prüfen, verwenden Sie die folgenden Befehle:

```
systemctl stop zabbix-agent
systemctl restart zabbix-agent
systemctl status zabbix-agent
```

Bei Installation aus einem Binärarchiv

Entpacken Sie das Archiv und suchen Sie die Binärdatei `zabbix_agentd` sowie die Konfigurationsdatei `zabbix_agentd.conf`. Ihr Speicherort innerhalb des Archivs hängt von der Plattform ab (z. B. `sbin/` und `conf/` oder `usr/sbin/` und `etc/zabbix/`). Führen Sie die Binärdatei direkt aus dem entpackten Verzeichnis aus und geben Sie die Konfigurationsdatei mit der Option `-c` an. Zum Beispiel:

```
/opt/zabbix/sbin/zabbix_agentd -c /opt/zabbix/etc/zabbix/zabbix_agentd.conf
```

Auf UNIX-ähnlichen Systemen ist der Zabbix Agent dafür ausgelegt, als Benutzer ohne Root-Rechte ausgeführt zu werden. Wenn er als Root gestartet wird, wechselt er zum Benutzer `zabbix` (oder zu dem Benutzer, der mit dem Parameter `[User](<PARAMS>#user)` angegeben ist), sofern in der Konfigurationsdatei nicht `[AllowRoot=1](<PARAMS>#allowroot)` festgelegt ist. Dieser Benutzer muss auf dem System vorhanden sein. Andernfalls wird der Agent nicht gestartet.

Unter IBM AIX kann der Agent auch mit dem System Resource Controller (SRC) ausgeführt und beim Systemstart gestartet werden. Weitere Informationen finden Sie unter Zabbix Agent unter IBM AIX.

Manuell starten

Sie können den Zabbix Agent starten, indem Sie die Binärdatei `zabbix_agentd` direkt ausführen, zum Beispiel:

```
zabbix_agentd
```

Dies funktioniert, wenn das Verzeichnis, das `zabbix_agentd` enthält, in `PATH` enthalten ist.

Agent auf Windows-Systemen Der Zabbix Agent unter Windows wird als Windows-Dienst ausgeführt.

Installation

Der Zabbix Agent kann unter Windows mit einer der folgenden Methoden installiert werden:

- Vorkompilierte Zabbix Agent-Binärdateien - laden Sie das MSI-Installationspaket des Zabbix Agent herunter und folgen Sie den Anweisungen auf der Seite Windows-Agent-Installation aus MSI;
- Zabbix-Quellcode - laden Sie die Quelldateien herunter und folgen Sie den Anweisungen auf der Seite Zabbix Agent unter Windows kompilieren.

Weitere Informationen zur Installation des Zabbix Agent (aus einem ZIP-Archiv) als Windows-Dienst finden Sie auf der Seite Zabbix Agent unter Microsoft Windows.

Optionen Die folgenden Befehlszeilenparameter können mit dem Zabbix Agent verwendet werden:

Parameter	Beschreibung
UNIX- und Windows-Agent	
`-c --config <config-file>`	Pfad zur Konfigurationsdatei. Mit dieser Option können Sie eine andere als die standardmäßige Konfigurationsdatei angeben.
`-f --foreground`	Zabbix Agent im Vordergrund ausführen (Standard: `true`).
`-p --print`	Bekannte Datenpunkte ausgeben und beenden. Beachten Sie, dass Sie die Konfigurationsdatei angeben müssen, wenn sie sich nicht am Standardspeicherort befindet, damit auch Ergebnisse von Benutzerparametern zurückgegeben werden.
`-t --test <item key>`	Den angegebenen Datenpunkt testen und beenden. Beachten Sie, dass Sie die Konfigurationsdatei angeben müssen, wenn sie sich nicht am Standardspeicherort befindet, damit auch Ergebnisse von Benutzerparametern zurückgegeben werden.
`-T --test-config`	Konfigurationsdatei validieren und beenden.
`-h --help`	Hilfeinformationen anzeigen.

Parameter	Beschreibung
`-V --version`	Versionsnummer anzeigen.
Nur UNIX-Agent	
`-R --runtime-control <option>`	Administrative Funktionen ausführen. Siehe Laufzeitsteuerung.
Nur Windows-Agent	
`-m --multiple-agents`	Mehrere Agent-Instanzen verwenden (mit den Optionen `-i`, `-d`, `-s`, `-x`). Um die Dienstnamen der Instanzen zu unterscheiden, enthält jeder Dienstname den Wert Hostname aus der angegebenen Konfigurationsdatei.
`-S --startup-type <value>`	Den Starttyp des Zabbix-Windows-Agent-Dienstes festlegen. Zulässige Werte: `automatic` - *(Standard)* den Dienst beim Start von Windows automatisch starten; `delayed` - den Dienst erst starten, nachdem die automatisch gestarteten Dienste vollständig gestartet wurden (verfügbar unter Windows Server 2008/Vista und späteren Versionen); `manual` - den Dienst manuell starten (durch einen Benutzer oder eine Anwendung); `disabled` - den Dienst deaktivieren, sodass er nicht von einem Benutzer oder einer Anwendung gestartet werden kann. Sie können diese Option zusammen mit der Option `-i` oder separat verwenden, um den Starttyp eines bereits installierten Dienstes zu ändern.
`-i --install`	Den Zabbix-Windows-Agent als Dienst installieren.
`-d --uninstall`	Den Zabbix-Windows-Agent-Dienst deinstallieren.
`-s --start`	Den Zabbix-Windows-Agent-Dienst starten.
`-x --stop`	Den Zabbix-Windows-Agent-Dienst stoppen.

Spezifische Beispiele für die Verwendung von Befehlszeilenparametern:

- Alle integrierten Agent-Datenpunkte mit Werten ausgeben.
- Einen Benutzerparameter mit dem im angegebenen Konfigurationsdatei definierten Schlüssel `mysql.ping` testen.
- Einen Zabbix-Agent-Dienst für Windows unter Verwendung des Standardpfads zur Konfigurationsdatei installieren.
- Einen Zabbix-Agent-Dienst [Hostname] für Windows unter Verwendung der Konfigurationsdatei im selben Ordner wie die ausführbare Agent-Datei installieren und den Dienstnamen durch Erweiterung um den Wert `Hostname` aus der Konfigurationsdatei eindeutig machen.
- Den Starttyp eines installierten Zabbix-Agent-Dienstes für Windows unter Verwendung der Konfigurationsdatei im selben Ordner wie die ausführbare Agent-Datei ändern.

```
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
```

Laufzeitsteuerung

Mit den Optionen zur Laufzeitsteuerung können Sie die Protokollstufe von Agent-Prozessen ändern.

Option	Beschreibung	Ziel
`log_level_increase[=<target>]`	Protokollstufe erhöhen. Wenn kein Ziel angegeben ist, sind alle Prozesse betroffen.	Das Ziel kann wie folgt angegeben werden: **Prozesstyp** – alle Prozesse des angegebenen Typs (z. B. `listener`) Siehe alle Agent-Prozesstypen. **Prozesstyp,N** – Prozesstyp und Nummer (z. B. `listener,3`) **PID** – Prozesskennung (1 bis 65535). Für größere Werte geben Sie das Ziel als „Prozesstyp,N“ an.
`log_level_decrease[=<target>]`	Protokollstufe verringern. Wenn kein Ziel angegeben ist, sind alle Prozesse betroffen.	
`userparameter_reload`	Werte der Optionen `UserParameter` und `Include` aus der aktuellen Konfigurationsdatei neu laden.	

Beispiele:

- Protokollstufe aller Prozesse erhöhen
- Protokollstufe des dritten Listener-Prozesses erhöhen
- Protokollstufe des Prozesses mit der PID 1234 erhöhen
- Protokollstufe aller Prozesse für aktive Prüfungen verringern

```
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:**
> Die Laufzeitsteuerung wird unter OpenBSD, NetBSD und Windows nicht unterstützt.

Agent-Prozesstypen Die Agent-Prozesstypen sind wie folgt:

- `active checks` - Prozess für die Durchführung aktiver Prüfungen;
- `collector` - Prozess für die Datenerfassung;
- `listener` - Prozess zum Abhören passiver Prüfungen.

Die Agent-Logdatei kann verwendet werden, um diese Prozesstypen zu beobachten.

Die Agent-Logdatei wird mit Lese- und Schreibberechtigungen nur für den Dateieigentümer erstellt. Zusätzlich ist die Datei für die Eigentümergruppe lesbar. Alle anderen Berechtigungen sind verweigert.

Benutzerprozess Der Zabbix Agent unter UNIX ist dafür ausgelegt, als Benutzer ohne Root-Rechte ausgeführt zu werden.
Er wird als der Benutzer ausgeführt, unter dem er gestartet wurde.
Daher können Sie den Agent problemlos als beliebigen Benutzer ohne Root-Rechte ausführen.

Wenn Sie versuchen, ihn als „root“ auszuführen, wechselt er zu einem fest codierten Benutzer „zabbix“, der auf Ihrem System vorhanden sein muss.
Sie können den Agent nur als „root“ ausführen, wenn Sie den Parameter `AllowRoot` in der Agent-Konfigurationsdatei entsprechend ändern.

Konfigurationsdatei Der Standardpfad zur Konfigurationsdatei hängt vom Betriebssystem und der Installationsmethode von Zabbix ab:

- Unter UNIX lautet der Standardpfad:
  - Bei der Installation aus den Quellen: `/usr/local/etc/zabbix_agentd.conf` oder der Pfad, der durch die Kompilierungsvariablen `--sysconfdir` oder `--prefix` festgelegt wurde.
  - Bei der Installation aus Paketen und bei der Appliance: `/etc/zabbix/zabbix_agentd.conf`.
- Unter Windows lautet der Standardpfad `C:\Program Files\Zabbix Agent\zabbix_agentd.conf`.

Wenn auf einem Host nur eine Instanz des Agent ausgeführt wird, kann sie die Standardkonfigurationsdatei oder eine in der Befehlszeile angegebene Konfigurationsdatei verwenden. Bei mehreren Instanzen muss jede Agent-Instanz über eine eigene Konfigurationsdatei verfügen (eine der Instanzen kann die Standardkonfigurationsdatei verwenden).

Weitere Informationen zur Konfiguration des Zabbix Agent finden Sie in den Optionen der Konfigurationsdatei für den Zabbix Agent unter Unix oder den Zabbix Agent unter Windows.

Locale Beachten Sie, dass der Agent ein UTF-8-Locale benötigt, damit einige textbasierte Agent-Datenpunkte den erwarteten Inhalt zurückgeben können. Die meisten modernen Unix-ähnlichen Systeme verwenden standardmäßig ein UTF-8-Locale; es gibt jedoch einige Systeme, auf denen dies ausdrücklich festgelegt werden muss.

Exit-Code Der Zabbix Agent gibt bei erfolgreichem Abschluss 0 und bei einem Fehler 1 zurück.

Passive und aktive Prüfprotokolle

Dieser Abschnitt enthält Protokolldetails zu passiven und aktiven Prüfungen, die von Zabbix Agent und Zabbix agent 2 durchgeführt werden.

Zabbix verwendet für die Kommunikation mit den Agents ein JSON-basiertes Kommunikationsprotokoll.

Die Protokolle von Zabbix Agent und Zabbix agent 2 sind seit Zabbix 7.0 vereinheitlicht. Der Unterschied zwischen Anfragen und Antworten von Zabbix Agent und Zabbix agent 2 wird durch den Wert des Tags ”variant” ausgedrückt.

Passive checks

Ein passiver Check ist eine einfache Datenanforderung. Zabbix Server oder Proxy fragt nach bestimmten Daten (zum Beispiel CPU-Auslastung), und der Zabbix Agent sendet das Ergebnis an den Server zurück.

Passive Checks werden asynchron ausgeführt - es ist nicht erforderlich, die Antwort auf eine Anfrage zu erhalten, bevor weitere Checks gestartet werden. Auch die DNS-Auflösung erfolgt asynchron.

Der Agent-Poller versucht, eine Verbindung zu allen Adressen herzustellen, die von der DNS-Abfrage zurückgegeben werden. Dadurch wird sichergestellt, dass der Poller bei einer nicht erreichbaren IP-Adresse die nächste verfügbare Adresse versucht, was die Wahrscheinlichkeit einer erfolgreichen Verbindung erhöht. Diese Verbesserung gilt sowohl für Zabbix Server als auch für Proxy.

Die maximale Parallelität asynchroner Checks beträgt 1000 (definiert durch MaxConcurrentChecksPerPoller).

Die Anzahl der asynchronen Agent-Poller wird durch den Parameter StartAgentPollers definiert.

Server-Anfrage

Für die Definition von Header und Datenlänge siehe Protokolldetails.

```
{
  "request": "passive checks",
  "data": [
    {
      "key": "agent.version",
      "timeout": 3
    }
  ]
}
```

Field		Type	Mandatory	Value
request		*string*	yes	"passive checks"
data		*array of object*	yes	Passive check item.
	key	*string*	yes	Item key with expanded macros.
	timeout	*number*	yes	Communication timeout.

Agent-Antwort

```
{
  "version": "8.0.0",
  "variant": 2,
  "data": [
    {
      "value": "8.0.0"
    }
  ]
}
```

Field		Type	Mandatory	Value
version		*string*	yes	The agent version number.
variant		*number*	yes	The agent variant (*1* - Zabbix agent, *2* - Zabbix agent 2).
data		*array of object*	yes	Contains the result of the check.
	value	*string*	no	The item value if the check was successful.
	error	*string*	no	The error message if the check was not successful.

Zum Beispiel für unterstützte Items:

1. Server öffnet eine TCP-Verbindung
2. Server sendet **<HEADER><DATALEN>{"request":"passive checks","data":[{"key":"agent.ping","timeout":3}]}**
3. Agent liest die Anfrage und antwortet mit **<HEADER><DATALEN>{"version":"8.0.0","variant":2,"data":[{"value":1}]}**
4. Server verarbeitet die Daten, um den Wert zu erhalten, in unserem Fall '1'
5. TCP-Verbindung wird geschlossen

Für nicht unterstützte Items:

1. Server öffnet eine TCP-Verbindung

2. Server sendet **<HEADER><DATALEN>{"request":"passive checks","data":[{"key":"vfs.fs.size[/nono]","timeout":3}]}**
3. Agent liest die Anfrage und antwortet mit **<HEADER><DATALEN>{"version":"8.0.0","variant":2,"data":[{"error":"Unsuppor item key."}]}**
4. Server verarbeitet die Daten, ändert den Item-Status auf nicht unterstützt mit der angegebenen Fehlermeldung
5. TCP-Verbindung wird geschlossen

Failover auf das alte Protokoll

Damit der Zabbix-Server oder Proxy mit Agents aus Versionen vor 7.2 zusammenarbeiten kann, die das Klartextprotokoll verwenden, ist ein Failover auf das alte Protokoll implementiert.

Passive Prüfungen werden nach einem Neustart oder bei einer Änderung der Schnittstellenkonfiguration über das JSON-Protokoll (7.0 und neuer) ausgeführt.
Wenn als Antwort kein gültiges JSON empfangen wird (der Agent sendet "ZBX_NOTSUPPORTED"), speichert Zabbix die Schnittstelle als altes Protokoll im Cache und **wiederholt** die Prüfung, indem nur der Datenpunkt-Schlüssel gesendet wird.

Beachten Sie, dass der Zabbix-Server/Proxy jede Stunde erneut versucht, mit allen Schnittstellen das neue Protokoll zu verwenden, und bei Bedarf auf das alte Protokoll zurückfällt.

Aktive Prüfungen

Aktive Prüfungen erfordern eine komplexere Verarbeitung.
Der Agent muss zunächst vom Server/Proxy eine Liste von Datenpunkten und/oder Remote-Befehlen zur unabhängigen Verarbeitung abrufen.

Die Server/Proxys, von denen die aktiven Prüfungen abgerufen werden, sind im Parameter `ServerActive` der Agent-Konfigurationsdatei aufgeführt.
Die Häufigkeit, mit der diese Prüfungen abgerufen werden, wird durch den Parameter `RefreshActiveChecks` in derselben Konfigurationsdatei festgelegt.
Wenn die Aktualisierung der aktiven Prüfungen jedoch fehlschlägt, wird sie nach fest codierten 60 Sekunden erneut versucht.

> **Note:**
> Seit Zabbix 6.4 erhält der Agent (im aktiven Modus) vom Server/Proxy nicht mehr alle zwei Minuten (Standard) eine vollständige Kopie der Konfiguration. Stattdessen wird, um den Netzwerkverkehr und den Ressourcenverbrauch zu verringern, alle 5 Sekunden (Standard) eine inkrementelle Konfigurationssynchronisierung durchgeführt, bei der der Server/Proxy eine vollständige Kopie der Konfiguration **nur dann** bereitstellt, wenn der Agent sie noch nicht erhalten hat oder sich etwas in der Host-Konfiguration, den globalen Makros oder den globalen regulären Ausdrücken geändert hat.

Der Agent sendet dann regelmäßig die neuen Werte an den Server(n). Wenn der Agent einen Remote-Befehl zur Ausführung erhalten hat, wird auch das Ausführungsergebnis gesendet. Beachten Sie, dass die Ausführung von Remote-Befehlen auf einem aktiven Agent seit Zabbix agent 7.0 unterstützt wird.

> **Note:**
> Wenn sich ein Agent hinter der Firewall befindet, sollten Sie möglicherweise nur aktive Prüfungen verwenden, da Sie in diesem Fall die Firewall nicht anpassen müssten, um eingehende Erstverbindungen zuzulassen.

Abrufen der Liste der Datenpunkte

Agent-Anfrage

Die Anfrage für aktive Prüfungen wird verwendet, um die aktiven Prüfungen zu ermitteln, die vom Agent verarbeitet werden sollen. Diese Anfrage wird vom Agent beim Start und anschließend in Intervallen von RefreshActiveChecks gesendet.

```
{
  "request": "active checks",
  "host": "Zabbix server",
  "host_metadata": "mysql,nginx",
  "interface": "zabbix.server.lan",
  "ip": "159.168.1.1",
  "port": 12050,
  "version": "8.0.0",
  "variant": 2,
  "config_revision": 1,
  "session": "e3dcbd9ace2c9694e1d7bbd030eeef6e"
}
```

Field	Type	Mandatory	Value
request	*string*	yes	`active checks`
host	*string*	yes	Host name.
host_metadata	*string*	no	The configuration parameter HostMetadata or HostMetadataItem metric value.
interface	*string*	no	The configuration parameter HostInterface or HostInterfaceItem metric value.
ip	*string*	no	The configuration parameter ListenIP first IP if set.
port	*number*	no	The configuration parameter ListenPort value if set and not default agent listening port.
version	*string*	yes	The agent version number.
variant	*number*	yes	The agent variant (*1* - Zabbix agent, *2* - Zabbix agent 2).
config_revision	*number*	no	Configuration identifier for incremental configuration sync.
session	*string*	no	Session identifier for incremental configuration sync.

Server-Antwort

Die Antwort auf aktive Prüfungen wird vom Server nach der Verarbeitung der Anfrage für aktive Prüfungen an den Agent zurückgesendet.

```
{
  "response": "success",
  "config_revision": 2,
  "data": [
    {
      "key": "system.uptime",
      "itemid": 1234,
      "delay": "10s",
      "lastlogsize": 0,
      "mtime": 0
    },
    {
      "key": "agent.version",
      "itemid": 5678,
      "delay": "10m",
      "lastlogsize": 0,
      "mtime": 0,
      "timeout": "30s"
    }
  ],
  "commands": [
    {
      "command": "df -h --output=source,size / | awk 'NR>1 {print $2}'",
      "id": 1324,
      "wait": 1
    }
  ]
}
```

Field	Type	Mandatory	Value
response	*string*	yes	`success` | `failed`
info	*string*	no	Error information in case of failure.
data	*array of objects*	no	Active check items. Omitted if host configuration is unchanged.
key	*string*	no	Item key with expanded macros.
itemid	*number*	no	Item identifier.
delay	*string*	no	Item update interval. Flexible/scheduling intervals are supported by both Zabbix agent and Zabbix agent 2 since Zabbix 7.0.
lastlogsize	*number*	no	Item lastlogsize.
mtime	*number*	no	Item mtime.
timeout	*string*	no	Item timeout.
refresh_unsupported	*number*	no	Unsupported item refresh interval.

Field	Type	Mandatory	Value
regexp	*array of objects*	no	Global regular expressions.
name	*string*	no	Global regular expression name.
expression	*string*	no	Global regular expression.
expression_type	*number*	no	Global regular expression type.
exp_delimiter	*string*	no	Global regular expression delimiter.
case_sensitive	*number*	no	Global regular expression case sensitivity setting.
commands	*array of objects*	no	Remote commands to execute. Included if remote command execution has been triggered by an action operation or manual script execution. Note that remote command execution on an active agent is supported since Zabbix agent 7.0. Older active agents will ignore any remote commands included in the active checks server response.
command	*string*	no	Remote command.
id	*number*	no	Remote command identifier.
wait	*number*	no	Remote command mode of execution (”0” (nowait) for commands from action operations; ”1” (wait) for commands from manual script execution).
timeout	*number*	no	Remote command execution timeout in server/proxy configuration.
config_revision	*number*	no	Configuration identifier for incremental configuration sync. Omitted if host configuration is unchanged. Incremented if host configuration is changed.

Der Server muss mit success antworten.

Zum Beispiel:

1. Agent öffnet eine TCP-Verbindung
2. Agent fragt die Liste der Prüfungen an
3. Server antwortet mit einer Liste von Datenpunkten und auszuführenden Remote-Befehlen
4. Agent analysiert die Antwort
5. TCP-Verbindung wird geschlossen
6. Agent beginnt mit der periodischen Datenerfassung und führt Remote-Befehle aus (seit Zabbix agent 7.0 unterstützt)

Attention:
Beachten Sie, dass (sensible) Konfigurationsdaten für Parteien zugänglich werden können, die Zugriff auf den Trapper-Port des Zabbix-Servers haben, wenn eine aktive Prüfung verwendet wird. Dies ist möglich, weil sich jeder als aktiver Agent ausgeben und Konfigurationsdaten für Datenpunkte anfordern kann; eine Authentifizierung findet nicht statt, sofern Sie keine Verschlüsselungs-Optionen verwenden.

Senden der gesammelten Daten

Agent sendet

Die Datenanforderung des Agenten enthält die gesammelten Datenpunktwerte und die Werte für ausgeführte Remote-Befehle (falls vorhanden).

```
{
  "request": "agent data",
  "data": [
    {
      "id": 1,
      "itemid": 5678,
      "value": "7.0.0",
      "clock": 1712830783,
      "ns": 76808644
    },
    {
      "id": 2,
      "itemid": 1234,
      "value": "69672",
      "clock": 1712830783,
      "ns": 77053975
    }
  ],
```

```
  "commands": [
    {
      "id": 1324,
      "value": "16G"
    }
  ],
  "session": "8495cd52070e6ca52b371f29c8574165",
  "host": "Zabbix server",
  "version": "8.0.0",
  "variant": 2
}
```

Field	Type	Mandatory	Value
request	*string*	yes	`agent data`
data	*array of objects*	yes	Datenpunktwerte.
id	*number*	yes	
itemid	*number*	yes	
value	*string*	no	
lastlogsize	*number*	no	
mtime	*number*	no	
state	*number*	no	
source	*string*	no	
eventid	*number*	no	
severity	*number*	no	
timestamp	*number*	no	
clock	*number*	yes	
ns	*number*	yes	
commands	*array of objects*	no	Ergebnis der Ausführung von Remote-Befehlen. Beachten Sie, dass die Ausführung von Remote-Befehlen auf einem aktiven Agenten seit Zabbix agent 7.0 unterstützt wird. Ältere aktive Agenten ignorieren alle Remote-Befehle, die in der Serverantwort auf aktive Prüfungen enthalten sind.
id	*number*	no	
value	*string*	no	
error	*string*	no	
session	*string*	yes	Eindeutiger Sitzungsbezeichner, der jedes Mal generiert wird, wenn der Agent gestartet wird (32 hexadezimale Ziffern lang).
host	*string*	yes	Hostname.
version	*string*	yes	Die Versionsnummer des Agenten.
variant	*number*	yes	Die Variante des Agenten (*1* - Zabbix agent, *2* - Zabbix agent 2).

Jedem Wert wird eine virtuelle ID zugewiesen. Die Wert-ID ist ein einfacher aufsteigender Zähler, der innerhalb einer Daten-Sitzung eindeutig ist (identifiziert durch das Sitzungstoken). Diese ID wird verwendet, um doppelte Werte zu verwerfen, die in Umgebungen mit schlechter Konnektivität gesendet werden könnten.

Serverantwort

Die Antwort auf die Agentendaten wird vom Server nach der Verarbeitung der Datenanforderung an den Agenten zurückgesendet.

```
{
  "response": "success",
  "info": "processed: 2; failed: 0; total: 2; seconds spent: 0.003534"
}
```

Field	Type	Mandatory	Value
response	*string*	yes	`success` | `failed`
info	*string*	yes	Ergebnisse der Verarbeitung der Datenpunkte.

Attention:
Wenn das Senden einiger Werte auf dem Server fehlschlägt (zum Beispiel, weil Host oder Datenpunkt deaktiviert oder gelöscht wurde), versucht der Agent nicht erneut, diese Werte zu senden.

Zum Beispiel:

1. Agent öffnet eine TCP-Verbindung
2. Agent sendet eine Liste von Werten
3. Server verarbeitet die Daten und sendet den Status zurück
4. TCP-Verbindung wird geschlossen

Attention:
Die Fehlermeldung wird serverseitig auf 2048 Zeichen gekürzt.

Heartbeat-Nachricht

Agent sendet

Die Heartbeat-Nachricht wird von einem aktiven Agent alle `HeartbeatFrequency` Sekunden an den Zabbix Server/Proxy gesendet (konfiguriert in der Konfigurationsdatei des Zabbix Agent oder Zabbix Agent 2).

Sie wird verwendet, um die Verfügbarkeit aktiver Prüfungen zu überwachen.

```
{
  "request": "active check heartbeat",
  "host": "Zabbix server",
  "heartbeat_freq": 60,
  "version": "8.0.0",
  "variant": 2
}
```

Field	Type	Mandatory	Value
request	*string*	yes	`active check heartbeat`
host	*string*	yes	Der Hostname.
heartbeat_freq	*number*	yes	Das Heartbeat-Intervall des Agent (Konfigurationsparameter HeartbeatFrequency).
version	*string*	yes	Die Versionsnummer des Agent.
variant	*number*	yes	Die Agent-Variante (*1* - Zabbix Agent, *2* - Zabbix Agent 2).

Antwort mit Umleitung

Wenn ein Host neu zugewiesen wurde, kann der Server den Agent anweisen, seinen Heartbeat (und nachfolgende aktive Prüfungen) an eine andere Proxy- oder Serverinstanz umzuleiten.

```
  {
    "response": "failed",
    "redirect": {
      "revision": 2,
      "address": "192.0.2.1:10055"
    }
  }
```

Field		Type	Mandatory	Value
response		*string*	yes	`success` | `failed`
redirect		*object*	yes	Anweisungen zur Umleitung.
	revision	*number*	yes	Bezeichner der Konfigurationsrevision.
	address	*string*	yes	Adresse des Ziel-Server/Proxy.

Älteres XML-Protokoll

Note:
Zabbix akzeptiert bis zu 16 MB Base64-codierte XML-Daten, aber ein einzelner decodierter Wert sollte nicht länger als 64 KB sein. Andernfalls wird er beim Decodieren auf 64 KB gekürzt.

Agent als root ausführen

Seit Zabbix **5.0.0** enthält die systemd-Dienstdatei für den Zabbix Agent in den offiziellen Paketen ausdrücklich Direktiven für `User` und `Group`. Beide sind auf `zabbix` gesetzt.

Es ist nicht mehr möglich, über die Datei `zabbix_agentd.conf` zu konfigurieren, unter welchem Benutzer der Zabbix Agent ausgeführt wird, da der Agent diese Konfiguration umgeht und als der in der systemd-Dienstdatei angegebene Benutzer ausgeführt wird. Um den Zabbix Agent als root auszuführen, müssen Sie die unten beschriebenen Änderungen vornehmen.

Zabbix Agent

Um den Standardbenutzer und die Standardgruppe für den Zabbix Agent zu überschreiben, führen Sie Folgendes aus:

```
systemctl edit zabbix-agent
```

Fügen Sie anschließend den folgenden Inhalt hinzu:

```
[Service]
User=root
Group=root
```

Laden Sie die Daemons neu und starten Sie den zabbix-agent-Dienst neu:

```
systemctl daemon-reload
systemctl restart zabbix-agent
```

Für den **Zabbix Agent** wird dadurch die Funktion zum Konfigurieren des Benutzers in der Agent-Konfigurationsdatei wieder aktiviert. Nun müssen Sie die Konfigurationsparameter `User=root` und `AllowRoot=1` in der Konfigurationsdatei festlegen.

Zabbix Agent 2

Um den Standardbenutzer und die Standardgruppe für Zabbix Agent 2 zu überschreiben, führen Sie Folgendes aus:

```
systemctl edit zabbix-agent2
```

Fügen Sie dann den folgenden Inhalt hinzu:

```
[Service]
User=root
Group=root
```

Laden Sie die Daemons neu und starten Sie den Dienst zabbix-agent2 neu:

```
systemctl daemon-reload
systemctl restart zabbix-agent2
```

Für **Zabbix Agent2** bestimmt dies vollständig den Benutzer, unter dem er ausgeführt wird. Weitere Änderungen sind nicht erforderlich.

Konfigurationsparameter: Zabbix Agent

Übersicht

Konfigurationsparameter ermöglichen die Anpassung des Verhaltens des Zabbix Agent.

Geben Sie Parameterwerte in der Agent-Konfigurationsdatei von Zabbix (`zabbix_agentd.conf`) an.
Alle Parameter sind **optional**, sofern nicht ausdrücklich angegeben ist, dass der Parameter obligatorisch ist.

Diese Seite ist wie folgt strukturiert:

- Parameterübersicht (um vollständige Details zu einem Parameter anzuzeigen, klicken Sie auf seinen Namen)
- Parameterdetails

Parameter	Beschreibung
Alias	Legt einen Alias für einen Datenpunktschlüssel fest.
AllowKey	Erlaubt die Ausführung von Datenpunktschlüsseln, die einem Muster entsprechen.
AllowKeyRegexp	Erlaubt die Ausführung von Datenpunktschlüsseln, die einem auf regulären Ausdrücken basierenden Muster entsprechen.
AllowRoot	Erlaubt die Ausführung des Agent als `root`.
BufferSend	Legt fest, wie häufig Werte aus dem Puffer an den Zabbix Server/Proxy gesendet werden.
BufferSize	Legt die maximale Anzahl von Werten im Speicherpuffer fest.
DebugLevel	Legt die Debug-Stufe fest.

Parameter	Beschreibung
DenyKey	Verhindert die Ausführung von Datenpunktschlüsseln, die einem Muster entsprechen.
DenyKeyRegexp	Verhindert die Ausführung von Datenpunktschlüsseln, die einem auf regulären Ausdrücken basierenden Muster entsprechen.
EnableRemoteCommands	Erlaubt Remote-Befehle vom Zabbix Server. Dieser Parameter ist veraltet; verwenden Sie stattdessen `AllowKey`, `AllowKeyRegexp`, `DenyKey` oder `DenyKeyRegexp`.
HeartbeatFrequency	Legt die Häufigkeit von Heartbeat-Nachrichten in Sekunden fest.
HostInterface	Gibt die Host-Schnittstelle an.
HostInterfaceItem	Gibt einen Datenpunkt zum Abrufen der Host-Schnittstelle an.
HostMetadata	Gibt die Host-Metadaten an.
HostMetadataItem	Gibt einen Zabbix-Agent-Datenpunkt zum Abrufen der Host-Metadaten an.
Hostname	Gibt den Hostnamen an.
HostnameItem	Gibt einen Zabbix-Agent-Datenpunkt zum Abrufen des Hostnamens an.
Include	Gibt ein Verzeichnis oder einzelne Dateien an, die in die Konfigurationsdatei aufgenommen werden sollen.
ListenBacklog	Legt die maximale Anzahl ausstehender Verbindungen in der TCP-Warteschlange fest.
ListenIP	Gibt die IP-Adresse an, an der der Agent Verbindungen vom Server/Proxy entgegennimmt.
ListenPort	Gibt den Port an, an dem der Agent Verbindungen entgegennimmt.
LoadModule	Gibt das beim Start des Agent zu ladende Modul an.
LoadModulePath	Gibt den vollständigen Pfad zum Speicherort der Agent-Module an.
LogFile	Gibt die Protokolldatei des Agent an.
LogFileSize	Legt die maximale Größe der Protokolldatei fest.
LogRemoteCommands	Aktiviert die Protokollierung ausgeführter Shell-Befehle als Warnungen.
LogType	Gibt den Typ der Protokollausgabe an.
MaxLinesPerSecond	Legt beim Verarbeiten aktiver `log`- oder `logrt`-Prüfungen die maximale Anzahl neuer Zeilen pro Sekunde fest, die an den Zabbix Server/Proxy gesendet werden.
PidFile	Gibt die PID-Datei an.
RefreshActiveChecks	Gibt die Häufigkeit der Aktualisierung der Liste aktiver Prüfungen an.
Server	Gibt die IP-Adressen oder DNS-Namen der Zabbix Server/Proxys an.
ServerActive	Gibt die Adresse des Zabbix Server/Proxy oder die Clusterkonfiguration an, von der aktive Prüfungen abgerufen werden.
SourceIP	Gibt die Quell-IP-Adresse an.
StartAgents	Legt die Anzahl vorab gestarteter Instanzen von `zabbix_agentd` zur Verarbeitung passiver Prüfungen fest.
Timeout	Gibt an, wie lange (in Sekunden) auf den Verbindungsaufbau und den Datenaustausch mit dem Zabbix Proxy oder Server gewartet werden soll.
TLSAccept	Gibt die für eingehende Verbindungen erforderliche Verschlüsselungsstufe an.
TLSCAFile	Gibt den vollständigen Pfadnamen der Datei an, die die Zertifikate der übergeordneten Zertifizierungsstelle(n) zur Überprüfung von Peer-Zertifikaten enthält und für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSCertFile	Gibt den vollständigen Pfadnamen der Datei an, die das Agent-Zertifikat oder die Zertifikatskette enthält und für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSCipherAll	Gibt die GnuTLS-Prioritätszeichenfolge oder die OpenSSL-(TLS-1.2-)Cipher-Zeichenfolge an. Überschreibt die Standardkriterien für die Auswahl der Cipher-Suite bei zertifikats- und PSK-basierter Verschlüsselung.
TLSCipherAll13	Gibt die Cipher-Zeichenfolge für OpenSSL 1.1.1 oder neuer unter TLS 1.3 an. Überschreibt die Standardkriterien für die Auswahl der Cipher-Suite bei zertifikats- und PSK-basierter Verschlüsselung.
TLSCipherCert	Gibt die GnuTLS-Prioritätszeichenfolge oder die OpenSSL-(TLS-1.2-)Cipher-Zeichenfolge an. Überschreibt die Standardkriterien für die Auswahl der Cipher-Suite bei zertifikatsbasierter Verschlüsselung.
TLSCipherCert13	Gibt die Cipher-Zeichenfolge für OpenSSL 1.1.1 oder neuer unter TLS 1.3 an. Überschreibt die Standardkriterien für die Auswahl der Cipher-Suite bei zertifikatsbasierter Verschlüsselung.
TLSCipherPSK	Gibt die GnuTLS-Prioritätszeichenfolge oder die OpenSSL-(TLS-1.2-)Cipher-Zeichenfolge an. Überschreibt die Standardkriterien für die Auswahl der Cipher-Suite bei PSK-basierter Verschlüsselung.
TLSCipherPSK13	Gibt die Cipher-Zeichenfolge für OpenSSL 1.1.1 oder neuer unter TLS 1.3 an. Überschreibt die Standardkriterien für die Auswahl der Cipher-Suite bei PSK-basierter Verschlüsselung.
TLSConnect	Gibt die für Verbindungen zum Zabbix Server oder Proxy erforderliche Verschlüsselungsstufe an.
TLSCRLFile	Gibt den vollständigen Pfadnamen der Datei an, die widerrufene Zertifikate enthält. Dieser Parameter wird für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet.

Parameter	Beschreibung
TLSKeyFile	Gibt den vollständigen Pfadnamen der Datei an, die den privaten Schlüssel des Agent enthält und für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSPSKFile	Gibt den vollständigen Pfadnamen der Datei an, die den Pre-Shared Key des Agent enthält und für die verschlüsselte Kommunikation mit dem Zabbix Server oder Proxy verwendet wird.
TLSPSKIdentity	Gibt die Identitätszeichenfolge des Pre-Shared Key an, die für die verschlüsselte Kommunikation mit dem Zabbix Server oder Proxy verwendet wird.
TLSServerCertIssuer	Gibt den zulässigen Aussteller des Server-(Proxy-)Zertifikats an.
TLSServerCertSubject	Gibt den zulässigen Betreff des Server-(Proxy-)Zertifikats an.
UnsafeUserParameters	Erlaubt die Übergabe aller Zeichen in Argumenten an benutzerdefinierte Parameter.
User	Gibt die Berechtigungen auf einen bestimmten, auf dem System vorhandenen Benutzer ab.
UserParameter	Gibt einen zu überwachenden benutzerdefinierten Parameter an.
UserParameterDir	Gibt den Standardsuchpfad für UserParameter-Befehle an.

Beachten Sie:

- Die Standardwerte entsprechen den Daemon-Standards und nicht den Werten in den mitgelieferten Konfigurationsdateien.
- Werte unterstützen Umgebungsvariablen.
- Zabbix unterstützt Konfigurationsdateien nur in der UTF-8-Kodierung ohne BOM.
- Kommentare, die mit # beginnen, werden nur am Anfang der Zeile unterstützt.

Parameterdetails

Alias

Legen Sie einen Alias für einen Datenpunktschlüssel fest.
Damit kann ein langer und komplexer Datenpunktschlüssel durch einen kürzeren und einfacheren ersetzt werden.
 Es können mehrere `Alias`-Parameter vorhanden sein.
Mehrere Parameter mit demselben `Alias`-Schlüssel sind nicht zulässig.
 Verschiedene `Alias`-Schlüssel können auf denselben Datenpunktschlüssel verweisen.
 Aliase können in `HostMetadataItem`, jedoch nicht im Parameter `HostnameItem` verwendet werden.

Beispiel 1: Abrufen der ID des Benutzers `zabbix`.

```
Alias=zabbix.userid:vfs.file.regexp[/etc/passwd,"^zabbix:.:([0-9]+)",,,,\1]
```

Nun kann der Kurzschlüssel `zabbix.userid` zum Abrufen von Daten verwendet werden.

Beispiel 2: Ermitteln der CPU-Auslastung mit Standard- und benutzerdefinierten Parametern.

```
Alias=cpu.util:system.cpu.util
Alias=cpu.util[*]:system.cpu.util[*]
```

Damit können Sie den Schlüssel `cpu.util` verwenden, um den Prozentsatz der CPU-Auslastung mit Standardparametern abzurufen, sowie `cpu.util[all, idle, avg15]`, um bestimmte Daten zur CPU-Auslastung abzurufen.

Beispiel 3: Ausführen mehrerer Regeln für die Low-Level-Discovery, die dieselben Discovery-Datenpunkte verarbeiten.

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

Nun können mehrere Discovery-Regeln mit `vfs.fs.discovery` eingerichtet werden, wobei für jede Regel unterschiedliche Parameter verwendet werden, z. B. `vfs.fs.discovery[foo]`, `vfs.fs.discovery[bar]` usw.

AllowKey

Erlaubt die Ausführung derjenigen Datenpunkt-Schlüssel, die einem Muster entsprechen. Das Schlüsselmuster ist ein Wildcard-Ausdruck, der das Zeichen * unterstützt, um eine beliebige Anzahl beliebiger Zeichen abzugleichen.
 Mehrere Regeln für den Schlüsselabgleich können in Kombination mit `DenyKey`, `AllowKeyRegexp` und `DenyKeyRegexp` definiert werden. Die Parameter werden nacheinander entsprechend ihrer Reihenfolge verarbeitet. Die Reihenfolge bestimmt, welche passende Regel Vorrang hat.
Siehe auch: Einschränken von Agent-Prüfungen.

AllowKeyRegexp

Erlaubt die Ausführung derjenigen Datenpunktschlüssel, die einem Muster entsprechen.
Das Schlüssel-Muster ist ein regulärer Ausdruck.
 Mehrere Regeln für den Datenpunktschlüssel-Abgleich können in Kombination mit `DenyKeyRegexp`, `AllowKey` und `DenyKey` definiert werden. Die Parameter werden entsprechend ihrer Reihenfolge nacheinander verarbeitet. Die Reihenfolge bestimmt, welche übereinstimmende Regel Vorrang hat. Siehe auch: Agent-Prüfungen einschränken.

Beispiel:

```
AllowKeyRegexp=^system\.run\[(?:/usr)?/bin/hostname(?: -[fsd])?\]$
```

AllowRoot

Erlaubt dem Agent, als `root` ausgeführt zu werden. Wenn diese Option deaktiviert ist und der Agent von `root` gestartet wird, versucht der Agent stattdessen, zum Benutzer `zabbix` zu wechseln. Hat keine Auswirkung, wenn der Start unter einem normalen Benutzer erfolgt.

Standard: 0
 Werte: 0 - nicht erlauben; 1 - erlauben

BufferSend

Legt die Häufigkeit fest, mit der Werte aus dem Puffer an den Zabbix-Server/Proxy gesendet werden (in Sekunden).

Standard: 5
 Bereich: 1-3600

BufferSize

Legt die maximale Anzahl von Werten im Speicherpuffer fest.
 Der Agent sendet alle gesammelten Daten an den Zabbix-Server oder Proxy, wenn der Puffer voll ist.

Standard: 100
 Bereich: 2-65535

DebugLevel

Legt den Debug-Level fest: - *0* - grundlegende Informationen zum Starten und Stoppen von Zabbix-Prozessen; - *1* - kritische Informationen; - *2* - Fehlerinformationen; - *3* - Warnungen; - *4* - für das Debugging (liefert viele Informationen); - *5* - erweitertes Debugging (liefert noch mehr Informationen).

Standard: 3
 Bereich: 0-5

DenyKey

Verhindert die Ausführung derjenigen Datenpunktschlüssel, die einem Muster entsprechen.
Das Schlüsselmuster ist ein Platzhalterausdruck, der das Zeichen * unterstützt, um eine beliebige Anzahl beliebiger Zeichen zu finden.
 Mehrere Regeln für den Datenpunktschlüsselabgleich können in Kombination mit `AllowKey`, `AllowKeyRegexp` und `DenyKeyRegexp` definiert werden.
Die Parameter werden entsprechend ihrer Reihenfolge nacheinander verarbeitet.
Die Reihenfolge bestimmt, welche übereinstimmende Regel Vorrang hat.
Siehe auch: Einschränken von Agent-Prüfungen.

Standard: `DenyKey=system.run[*]`

DenyKeyRegexp

Verhindert die Ausführung von Datenpunktschlüsseln, die einem Muster entsprechen. Das Schlüsselmuster ist ein regulärer Ausdruck.
 Mehrere Regeln für den Datenpunktabgleich können in Kombination mit `AllowKeyRegexp`, `AllowKey` und `DenyKey` definiert werden. Die Parameter werden entsprechend ihrer Reihenfolge nacheinander verarbeitet. Die Reihenfolge bestimmt, welche übereinstimmende Regel Vorrang hat.
 Alle `system.run[]`-Datenpunkte werden standardmäßig abgelehnt (`DenyKey=system.run[*]`), sofern keine andere `system.run[*]`-Regel angegeben ist.

Siehe auch: Einschränken von Agent-Prüfungen.

EnableRemoteCommands

Remote-Befehle vom Zabbix Server zulassen.
 Dieser Parameter ist **veraltet**. Verwenden Sie stattdessen `AllowKey=system.run[*]`, `AllowKeyRegexp=^system\.run\[.*\]$`, `DenyKey=system.run[*]` oder `DenyKeyRegexp=^system\.run\[.*\]$`.
Je nach Wert ist dies ein interner Alias für die Parameter `AllowKey`, `AllowKeyRegexp`, `DenyKey` und `DenyKeyRegexp`: - 0 - `DenyKey=system.run[*]` (`DenyKeyRegexp=^system\.run\[.*\]$`) - 1 - `AllowKey=system.run[*]` (`AllowKeyRegexp=^system\.`

Standard: 0
 Werte: 0 - nicht zulassen, 1 - zulassen

HeartbeatFrequency

Legt die Frequenz der Heartbeat-Nachrichten in Sekunden fest. Wird zur Überwachung der Verfügbarkeit aktiver Prüfungen verwendet.
0 - Heartbeat-Nachrichten deaktiviert.

Standard: 60
 Bereich: 0-3600

HostInterface

Geben Sie die Host-Schnittstelle (IP-Adresse oder DNS-Name) an, die während der Autoregistrierung des Hosts verwendet wird. Dieser Wert wird verwendet, um die Schnittstelle im neu erstellten Host zu befüllen, und ermöglicht es, entweder die IP- oder die DNS-Adresse explizit zu konfigurieren. Weitere Informationen finden Sie unter Using DNS as default interface.

Wenn nicht definiert, wird der Wert aus `HostInterfaceItem` übernommen.

Der Agent gibt einen Fehler aus und startet nicht, wenn der Wert die Grenze von 255 Zeichen überschreitet.

Bereich: 0-255 Zeichen

HostInterfaceItem

Geben Sie einen Datenpunkt an, um während der Autoregistrierung des Hosts die Host-Schnittstelle (IP-Adresse oder DNS-Name) zu bestimmen.
Dieser Wert wird nur verwendet, wenn `HostInterface` nicht definiert ist.
Weitere Informationen finden Sie unter DNS als Standardschnittstelle verwenden.

Während einer Autoregistrierungsanfrage protokolliert der Agent eine Warnmeldung, wenn der vom angegebenen Datenpunkt zurückgegebene Wert das Limit von 255 Zeichen überschreitet.

Der Datenpunkt `system.run[]` wird unabhängig von den Einstellungen `AllowKey`, `AllowKeyRegexp`, `DenyKey` oder `DenyKeyRegexp` unterstützt.

HostMetadata

Geben Sie die Metadaten an, anhand derer der Host während der Autoregistrierung (aktiver Agent) identifiziert oder unterschieden werden soll.
Mit `HostMetadata` können Hosts über den Hostnamen hinaus unterschieden werden.

Wenn nicht definiert, wird der Wert aus `HostMetadataItem` abgerufen.

Der Agent gibt einen Fehler aus und startet nicht, wenn der angegebene Wert das Limit von 2034 Bytes überschreitet oder keine UTF-8-Zeichenkette ist.
Wenn ein Parameter eine IP-Adresse oder einen DNS-Namen erwartet, werden Werte, die zwar gültiges UTF-8 enthalten, aber keine gültigen IP-Adressen oder DNS-Namen sind, ebenfalls abgelehnt und als ungültig gemeldet.

Mehrzeilige Metadaten werden nicht unterstützt – die Ausgabe wird beim ersten Zeilenumbruch abgeschnitten.

Bereich: 0–2034 Bytes

HostMetadataItem

Geben Sie einen Datenpunkt des Zabbix Agent an, über den Host-Metadaten abgerufen werden. Diese Option wird nur verwendet, wenn `HostMetadata` nicht definiert ist.

Der Wert von `HostMetadataItem` wird bei jedem Versuch der Autoregistrierung abgerufen und nur während des Host-Autoregistrierungsprozesses (aktiver Agent) verwendet. Mit `HostMetadataItem` können Hosts über den Hostnamen hinaus unterschieden werden.

Benutzerparameter und Aliase werden unterstützt. Der Datenpunkt `system.run[]` wird unabhängig von den Einstellungen `AllowKey`, `AllowKeyRegexp`, `DenyKey` oder `DenyKeyRegexp` unterstützt.

Während einer Autoregistrierungsanfrage protokolliert der Agent eine Warnung, wenn der vom angegebenen Datenpunkt zurückgegebene Wert das Limit von 65535 UTF-8-Codepunkten überschreitet. Der vom Datenpunkt zurückgegebene Wert muss eine UTF-8-Zeichenfolge sein; andernfalls wird er ignoriert. Wenn ein Parameter eine IP-Adresse oder einen DNS-Namen erwartet, werden Werte, die zwar gültiges UTF-8 enthalten, aber keine gültigen IP-Adressen oder DNS-Namen sind, ebenfalls abgelehnt und als ungültig gemeldet.

Mehrzeilige Metadaten werden nicht unterstützt – die Ausgabe wird beim ersten Zeilenumbruch abgeschnitten.

Hostname

Geben Sie eindeutige, groß-/kleinschreibungssensitive Hostnamen an. Eine durch Kommas getrennte Liste.
 Erforderlich für aktive Prüfungen und muss mit den auf dem Server konfigurierten Hostnamen übereinstimmen.
 Der Wert wird aus `HostnameItem` übernommen, wenn er nicht definiert ist.
Zulässige Zeichen: alphanumerisch, '.', ' ', '_' und '-'. Maximale Länge: 128 Zeichen pro Hostname, 2048 Zeichen für die gesamte Zeile.

Standard: Wird durch `HostnameItem` festgelegt

HostnameItem

Geben Sie einen Zabbix-Agent-Datenpunkt zum Abrufen des Hostnamens an.
 Diese Option wird nur verwendet, wenn Hostname nicht definiert ist.
 Benutzerparameter oder Aliase werden nicht unterstützt, aber der Datenpunkt `system.run[]` wird unabhängig von den Werten von `AllowKey`, `AllowKeyRegexp`, `DenyKey` oder `DenyKeyRegexp` unterstützt.

Standard: `system.hostname`

Include

Geben Sie ein Verzeichnis oder einzelne Dateien an, die in die Konfigurationsdatei aufgenommen werden sollen.
 Während der Zabbix-Installation wird dieser Parameter auf `/usr/local/etc` gesetzt, sofern Sie dies nicht zur Compile-Zeit ändern.

Allgemeine Regeln für den Parameter `Include`:

- Der Pfad kann relativ zum Speicherort der Datei `zabbix_agentd.conf` angegeben werden.
- Sie können mehrere `Include`-Parameter in einer Konfigurationsdatei angeben.
- Zabbix verarbeitet den Parameter `Include`, sobald er ihn liest. Platzieren Sie ihn daher am besten am Ende der Konfigurationsdatei.

Regeln für eingebundene Dateien:

- Jede Datei muss lesbar sein.
- Jede Datei muss eine korrekte Syntax haben. Andernfalls wird die Komponente nicht gestartet.

Regeln für das Einbinden von Verzeichnissen oder Dateimustern:

- Gehen Sie nicht von einer bestimmten Reihenfolge beim Einbinden aus (z. B. werden Dateien nicht alphabetisch eingebunden).
- Definieren Sie denselben Parameter nicht in mehr als einer eingebundenen Datei (z. B. um eine allgemeine Einstellung durch eine spezifische zu überschreiben), da Sie nicht vorhersagen können, welcher Wert verwendet wird.
- Achten Sie auf Sicherungskopien, die einige Texteditoren automatisch erstellen (z. B. kann beim Bearbeiten von `include/my_specific.conf` die Datei `include/my_specific.conf.BAK` erstellt werden), da beide Dateien eingebunden werden.

Um nur relevante Dateien aus einem Verzeichnis einzubinden, verwenden Sie den Platzhalter (`*`) für die Mustererkennung.

Beispiel:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenBacklog

Legt die maximale Anzahl ausstehender Verbindungen in der TCP-Warteschlange fest.
 Der Standardwert ist eine fest codierte Konstante, die vom System abhängt.
 Der maximal unterstützte Wert hängt ebenfalls vom System ab; zu hohe Werte können stillschweigend auf das „implementationsspezifische Maximum“ gekürzt werden.

Standard: `SOMAXCONN`
 Bereich: 0 - INT_MAX

ListenIP

Geben Sie die IP-Adresse an, auf der der Agent Verbindungen vom Server/Proxy entgegennimmt. Eine durch Kommas getrennte Liste.

Standard: `0.0.0.0`

ListenPort

Geben Sie den Agent-Listen-Port für Verbindungen vom Server/Proxy an.

Standard: 10050
 Bereich: 1024-32767

LoadModule

Geben Sie das Modul an, das beim Start des Agent geladen werden soll.
 Module werden verwendet, um die Funktionalität des Agent zu erweitern.
 Das Modul muss sich in dem durch `LoadModulePath` angegebenen Verzeichnis befinden, oder dem Modulnamen muss der Pfad vorangestellt sein. Wenn der vorangestellte Pfad absolut ist (beginnt mit `/`), wird `LoadModulePath` ignoriert.
 Formate: - `LoadModule=<module.so>` - `LoadModule=<path/module.so>` - `LoadModule=</abs_path/module.so>`

Es ist zulässig, mehrere `LoadModule`-Parameter anzugeben.

LoadModulePath

Geben Sie den vollständigen Pfad zum Speicherort der Agent-Module an.
 Die Standardeinstellung hängt von den Kompilierungsoptionen ab.

LogFile

Geben Sie die Logdatei des Agent an.

Erforderlich: Ja, wenn `LogType` auf `file` gesetzt ist; andernfalls nein

LogFileSize

Legt die maximale Größe der Protokolldatei fest (in MB).
0 - deaktiviert die automatische Protokollrotation.
Hinweis: Wenn die Größenbegrenzung der Protokolldatei erreicht wird und die Dateirotation aus irgendeinem Grund fehlschlägt, wird die vorhandene Protokolldatei gekürzt und neu begonnen.

Standard: 1
 Bereich: 0-1024

LogRemoteCommands

Aktivieren Sie die Protokollierung der ausgeführten Shell-Befehle als Warnungen.
 Befehle werden nur protokolliert, wenn sie remote ausgeführt werden.
 Protokolleinträge werden nicht erstellt, wenn `system.run[]` lokal durch die Parameter `HostMetadataItem`, `HostInterfaceItem` oder `HostnameItem` gestartet wird.

Standard: 0
 Werte: 0 - deaktiviert, 1 - aktiviert

LogType

Geben Sie den Typ der Logausgabe an: - *file* - schreibt das Log in die durch den Parameter `LogFile` angegebene Datei; - *system* - schreibt das Log in syslog; - *console* - schreibt das Log in die Standardausgabe.

Standardwert: `file`

MaxLinesPerSecond

Legt die maximale Anzahl neuer Zeilen pro Sekunde fest, die beim Verarbeiten aktiver `log`- oder `logrt`-Prüfungen an den Zabbix Server/Proxy gesendet werden.
 Der angegebene Wert wird durch den Parameter `maxlines` überschrieben, der im `log`- oder `logrt`-Datenpunktschlüssel angegeben ist.
Hinweis: Zabbix verarbeitet 10-mal mehr neue Zeilen als in `MaxLinesPerSecond` festgelegt, um in Protokolleinträgen nach der erforderlichen Zeichenfolge zu suchen.

Standard: 20
 Bereich: 1-1000

PidFile

Geben Sie die PID-Datei an.

Standard: `/tmp/zabbix_agentd.pid`

RefreshActiveChecks

Geben Sie die Häufigkeit der Aktualisierung der Liste der aktiven Prüfungen in Sekunden an.
 Beachten Sie, dass nach einem fehlgeschlagenen Versuch, die aktiven Prüfungen zu aktualisieren, der nächste Aktualisierungsversuch in 60 Sekunden unternommen wird.

Standard: 5
 Bereich: 1-86400

Server

Geben Sie die IP-Adressen, optional in CIDR-Notation, oder DNS-Namen von Zabbix-Servern/Proxys an. Eine durch Kommas getrennte Liste.
 Eingehende Verbindungen werden nur von den hier aufgeführten Hosts akzeptiert. Wenn die IPv6-Unterstützung aktiviert ist, werden `127.0.0.1`, `::127.0.0.1` und `::ffff:127.0.0.1` als gleich behandelt, und `::/0` erlaubt jede IPv4- oder IPv6-Adresse. `0.0.0.0/0` kann verwendet werden, um jede IPv4-Adresse zuzulassen. Beachten Sie, dass "IPv4-kompatible IPv6-Adressen" (`0000::/96`-Präfix) unterstützt werden, aber gemäß RFC4291 veraltet sind. Leerzeichen sind zulässig.

Beispiel:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Erforderlich: ja, wenn StartAgents nicht explizit auf 0 gesetzt ist

ServerActive

Geben Sie die Adresse des Zabbix-Servers/Proxys oder die Clusterkonfiguration an, von der aktive Prüfungen abgerufen werden sollen.
 Die Adresse des Servers/Proxys ist eine IP-Adresse oder ein DNS-Name mit optionalem Port, getrennt durch einen Doppelpunkt.
Die Clusterkonfiguration ist eine oder mehrere Adressen von Mitgliedern einer Server- oder Proxygruppe, getrennt durch Semikolon. Mehrere Zabbix-Server/Cluster und Zabbix-Proxys können angegeben werden, getrennt durch Komma. Außer bei Verwendung von Proxygruppen sollte von jedem Zabbix-Server/Cluster nicht mehr als ein Zabbix-Proxy angegeben werden. Wenn ein Zabbix-Proxy angegeben ist, sollte der Zabbix-Server/das Cluster für diesen Proxy nicht angegeben werden.
Mehrere durch Kommas getrennte Adressen können angegeben werden, um mehrere unabhängige Zabbix-Server parallel zu verwenden. Leerzeichen sind zulässig.
Wenn der Port nicht angegeben ist, wird der Standardport verwendet.
IPv6-Adressen müssen in eckige Klammern gesetzt werden, wenn für diesen Host ein Port angegeben ist. Wenn kein Port angegeben ist, sind eckige Klammern für IPv6-Adressen optional.
Wenn dieser Parameter nicht angegeben ist, sind aktive Prüfungen deaktiviert.

Beispiel für einen Zabbix-Proxy:

```
ServerActive=127.0.0.1:10051
```

Beispiel für eine Zabbix-Proxygruppe:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

Beispiel für mehrere Server:

```
ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
```

Beispiel für Hochverfügbarkeit:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

Beispiel für Hochverfügbarkeit mit zwei Clustern und einem Server:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

SourceIP

Geben Sie die Quell-IP-Adresse an für:

- ausgehende Verbindungen zum Zabbix Server oder Zabbix Proxy;
- Verbindungen, die beim Ausführen bestimmter Datenpunkte (`web.page.get`, `net.tcp.port` usw.) hergestellt werden.

StartAgents

Legt die Anzahl der vorab erzeugten Instanzen von `zabbix_agentd` für die Verarbeitung passiver Prüfungen fest.
 Wenn auf 0 gesetzt, sind passive Prüfungen deaktiviert und der Agent lauscht auf keinem TCP-Port.

Standard: 10
 Bereich: 0-100

Timeout

Geben Sie an, wie lange beim Aufbau der Verbindung und beim Datenaustausch mit dem Zabbix Proxy oder Server gewartet werden soll (in Sekunden).

Dieser Parameter legt die Dauer verschiedener Kommunikationsvorgänge fest, darunter:

- das Warten auf eine Antwort vom Zabbix Server;
- das Senden von Anfragen an den Zabbix Server, einschließlich Anfragen zur Datenpunktkonfiguration und Datenpunktdaten in aktiven Prüfungen;
- das Abrufen von Protokolldaten über logfile;
- das Senden von Heartbeat-Nachrichten;
- die maximale Dauer für `vfs.*`-Prüfungen;
- die Verwendung durch Zabbix Agent-Module;
- die Verwendung als Fallback in Szenarien, in denen ein Server oder Proxy älter als Version 7.0 Prüfungen ohne Timeouts sendet.

Dieses Timeout wird **nicht** für jene Agent-Prüfungen verwendet, die in der Frontend konfigurierbare Timeout-Einstellungen haben (auf globaler Ebene, auf Proxy-Ebene oder pro Datenpunkt).

Standard: 3
 Bereich: 1-30

TLSAccept

Geben Sie das für eingehende Verbindungen erforderliche Verschlüsselungsniveau an.
 Wird für passive Prüfungen verwendet.
 Es können mehrere Werte angegeben werden, getrennt durch Kommas: - *unencrypted* - Verbindungen ohne Verschlüsselung akzeptieren (Standard); - *psk* - Verbindungen mit TLS und einem vorab gemeinsam genutzten Schlüssel (PSK) akzeptieren; - *cert* - Verbindungen mit TLS und einem Zertifikat akzeptieren.

Pflicht: ja, wenn TLS-Zertifikat- oder PSK-Parameter definiert sind (auch für eine *unencrypted* Verbindung); andernfalls nein

TLSCAFile

Geben Sie den vollständigen Pfadnamen der Datei an, die die Zertifikate der obersten CA(s) für die Überprüfung von Peer-Zertifikaten enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

TLSCertFile

Geben Sie den vollständigen Pfadnamen der Datei an, die das Agent-Zertifikat oder die Zertifikatskette enthält und für verschlüsselte Kommunikation mit Zabbix-Komponenten verwendet wird.

TLSCipherAll

Geben Sie den GnuTLS-Prioritätsstring oder den OpenSSL-Cipher-String (TLS 1.2) an. Überschreiben Sie die standardmäßigen Auswahlkriterien für die Ciphersuite bei zertifikats- und PSK-basierter Verschlüsselung.

Beispiel:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

Geben Sie die Cipher-String für OpenSSL 1.1.1 oder neuer in TLS 1.3 an. Überschreiben Sie die standardmäßigen Auswahlkriterien für die Cipher Suite für zertifikats- und PSK-basierte Verschlüsselung.

Beispiel für GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL
```

Beispiel für OpenSSL:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherCert

Geben Sie die GnuTLS-Prioritätszeichenfolge oder die OpenSSL-(TLS 1.2)-Cipher-Zeichenfolge an. Überschreiben Sie die standardmäßigen Auswahlkriterien für die Cipher Suite bei zertifikatsbasierter Verschlüsselung.

Beispiel für GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Beispiel für OpenSSL:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

TLSCipherCert13

Geben Sie die Cipher-String für OpenSSL 1.1.1 oder neuer in TLS 1.3 an. Überschreiben Sie die standardmäßigen Auswahlkriterien für die Ciphersuite bei zertifikatsbasierter Verschlüsselung.

TLSCipherPSK

Geben Sie den GnuTLS-Prioritätsstring oder den OpenSSL-Cipher-String (TLS 1.2) an. Überschreiben Sie die standardmäßigen Auswahlkriterien für die Cipher Suite bei PSK-basierter Verschlüsselung.

Beispiel für GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Beispiel für OpenSSL:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

Geben Sie den Cipher-String für OpenSSL 1.1.1 oder neuer in TLS 1.3 an. Überschreiben Sie die standardmäßigen Auswahlkriterien für die Ciphersuite bei PSK-basierter Verschlüsselung.

Beispiel:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Geben Sie die für Agent-Verbindungen zum Zabbix Server oder Proxy erforderliche Verschlüsselungsstufe an.
 Wird für aktive Prüfungen verwendet.
 Es kann nur ein Wert angegeben werden: - *unencrypted* - Verbindung ohne Verschlüsselung herstellen (Standard); - *psk* - Verbindung mit TLS und einem vorab gemeinsam genutzten Schlüssel (PSK) herstellen; - *cert* - Verbindung mit TLS und einem Zertifikat herstellen.

Pflicht: ja, wenn TLS-Zertifikat- oder PSK-Parameter definiert sind (auch für eine *unencrypted* Verbindung); andernfalls nein

TLSCRLFile

Geben Sie den vollständigen Pfadnamen der Datei an, die widerrufene Zertifikate enthält. Dieser Parameter wird für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet.

TLSKeyFile

Geben Sie den vollständigen Pfadnamen der Datei an, die den privaten Schlüssel des Agent enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

TLSPSKFile

Geben Sie den vollständigen Pfadnamen der Datei an, die den vorab gemeinsam genutzten Schlüssel des Agent enthält und für verschlüsselte Kommunikation mit dem Zabbix-Server oder Proxy verwendet wird.

TLSPSKIdentity

Geben Sie die Identitätszeichenfolge des vorab gemeinsam genutzten Schlüssels an, die für verschlüsselte Kommunikation mit dem Zabbix-Server oder Proxy verwendet wird.

TLSServerCertIssuer

Geben Sie den zulässigen Zertifikatausssteller des Servers (Proxy) an.

TLSServerCertSubject

Geben Sie den zulässigen Zertifikatsbetreff des Servers (Proxy) an.

UnsafeUserParameters

Erlaubt, dass alle Zeichen in Argumenten an benutzerdefinierte Parameter übergeben werden. Die folgenden Zeichen sind nicht erlaubt: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
Zusätzlich sind Zeilenumbrüche nicht erlaubt.

Standard: 0
 Werte: 0 - nicht erlauben, 1 - erlauben

Benutzer

Die Berechtigungen auf einen bestimmten, vorhandenen Benutzer auf dem System herabsetzen.
Wirkt nur, wenn es als `root` ausgeführt wird und `AllowRoot` deaktiviert ist.

Standard: `zabbix`

UserParameter

Geben Sie einen benutzerdefinierten Parameter zur Überwachung an. Es können mehrere benutzerdefinierte Parameter vorhanden sein.
Format: `UserParameter=<key>,<shell command>`
Beachten Sie, dass der Shell-Befehl keinen leeren String oder nur EOL zurückgeben darf. Shell-Befehle dürfen relative Pfade verwenden, wenn der Parameter `UserParameterDir` angegeben ist.

Beispiel:

```
UserParameter=system.test,who|wc -l
UserParameter=check_cpu,./custom_script.sh
```

UserParameterDir

Geben Sie den standardmäßigen Suchpfad für `UserParameter`-Befehle an. Wenn diese Option verwendet wird, wechselt der Agent sein Arbeitsverzeichnis vor der Ausführung eines Befehls in das hier angegebene Verzeichnis. Dadurch können `UserParameter`-Befehle ein relatives Präfix `./` anstelle eines vollständigen Pfads verwenden.
Es ist nur ein Eintrag zulässig.

Beispiel:

```
UserParameterDir=/opt/myscripts
```

Siehe auch

1. Unterschiede in der Zabbix-Agent-Konfiguration für aktive und passive Prüfungen ab Version 2.0.0

Konfigurationsparameter: Zabbix Agent (Windows)

Übersicht

Konfigurationsparameter ermöglichen die Anpassung des Verhaltens des Zabbix Agent unter Windows.

Geben Sie die Parameterwerte in der Agent-Konfigurationsdatei von Zabbix (`zabbix_agentd.conf`) an. Alle Parameter sind **optional**, sofern nicht ausdrücklich angegeben ist, dass der Parameter obligatorisch ist.

Diese Seite ist wie folgt aufgebaut:

- Parameterübersicht (um die vollständigen Details eines Parameters anzuzeigen, klicken Sie auf seinen Namen).
- Parameterdetails.

Parameter	Beschreibung
Alias	Legt einen Alias für einen Datenpunktschlüssel fest.
AllowKey	Erlaubt die Ausführung der Datenpunktschlüssel, die einem Muster entsprechen.
AllowKeyRegexp	Erlaubt die Ausführung von Datenpunktschlüsseln, die einem auf regulären Ausdrücken basierenden Muster entsprechen.
BufferSend	Legt fest, wie häufig Werte aus dem Puffer an den Zabbix Server/Proxy gesendet werden.
BufferSize	Legt die maximale Anzahl von Werten im Speicherpuffer fest.
DebugLevel	Legt die Debug-Stufe fest.
DenyKey	Verweigert die Ausführung der Datenpunktschlüssel, die einem Muster entsprechen.
DenyKeyRegexp	Verweigert die Ausführung von Datenpunktschlüsseln, die einem auf regulären Ausdrücken basierenden Muster entsprechen.

Parameter	Beschreibung
EnableRemoteCommands	Erlaubt Remote-Befehle vom Zabbix Server. Dieser Parameter ist veraltet; verwenden Sie stattdessen `AllowKey`, `AllowKeyRegexp`, `DenyKey` oder `DenyKeyRegexp`.
HeartbeatFrequency	Legt die Häufigkeit von Heartbeat-Nachrichten in Sekunden fest.
HostInterface	Gibt die Host-Schnittstelle an.
HostInterfaceItem	Gibt einen Datenpunkt zum Abrufen der Host-Schnittstelle an.
HostMetadata	Gibt die Host-Metadaten an.
HostMetadataItem	Gibt einen Zabbix-Agent-Datenpunkt zum Abrufen der Host-Metadaten an.
Hostname	Gibt den Hostnamen an.
HostnameItem	Gibt einen Zabbix-Agent-Datenpunkt zum Abrufen des Hostnamens an.
Include	Gibt ein Verzeichnis oder einzelne Dateien an, die in die Konfigurationsdatei aufgenommen werden sollen.
ListenBacklog	Legt die maximale Anzahl ausstehender Verbindungen in der TCP-Warteschlange fest.
ListenIP	Gibt die IP-Adresse an, auf der der Agent Verbindungen vom Server/Proxy entgegennimmt.
ListenPort	Gibt den Port an, auf dem der Agent Verbindungen entgegennimmt.
LogFile	Gibt die Protokolldatei des Agent an.
LogFileSize	Legt die maximale Größe der Protokolldatei fest.
LogRemoteCommands	Aktiviert die Protokollierung ausgeführter Shell-Befehle als Warnungen.
LogType	Gibt den Typ der Protokollausgabe an.
MaxLinesPerSecond	Legt die maximale Anzahl neuer Zeilen pro Sekunde fest, die beim Verarbeiten aktiver Prüfungen von `log`, `logrt` oder `eventlog` an den Zabbix Server/Proxy gesendet werden.
PerfCounter	Definiert einen neuen Parameter <parameter_name>, der den Durchschnittswert des Systemleistungsindikators <perf_counter_path> für den angegebenen Zeitraum <period> (in Sekunden) darstellt.
PerfCounterEn	Definiert einen neuen Parameter <parameter_name>, der den Durchschnittswert des Systemleistungsindikators <perf_counter_path> für den angegebenen Zeitraum <period> (in Sekunden) darstellt. Im Gegensatz zu PerfCounter müssen die Pfade der Leistungsindikatoren auf Englisch angegeben werden.
RefreshActiveChecks	Gibt an, wie häufig die Liste der aktiven Prüfungen aktualisiert wird.
Server	Gibt die IP-Adressen oder DNS-Namen der Zabbix Server/Proxys an.
ServerActive	Gibt die Adresse des Zabbix Server/Proxy oder die Clusterkonfiguration an, von der aktive Prüfungen abgerufen werden.
SourceIP	Gibt die Quell-IP-Adresse an.
StartAgents	Legt die Anzahl vorab gestarteter Instanzen von `zabbix_agentd` zur Verarbeitung passiver Prüfungen fest.
Timeout	Gibt an, wie lange (in Sekunden) auf den Verbindungsaufbau und den Datenaustausch mit dem Zabbix Server oder Proxy gewartet wird.
TLSAccept	Gibt die erforderliche Verschlüsselungsstufe für eingehende Verbindungen an.
TLSCAFile	Gibt den vollständigen Pfadnamen der Datei an, die die Zertifikate der übergeordneten Zertifizierungsstelle(n) zur Überprüfung von Peer-Zertifikaten enthält und für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSCertFile	Gibt den vollständigen Pfadnamen der Datei an, die das Agent-Zertifikat oder die Zertifikatskette enthält und für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSConnect	Gibt die erforderliche Verschlüsselungsstufe für Verbindungen zum Zabbix Server oder Proxy an.
TLSCRLFile	Gibt den vollständigen Pfadnamen der Datei an, die widerrufene Zertifikate enthält. Dieser Parameter wird für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet.
TLSKeyFile	Gibt den vollständigen Pfadnamen der Datei an, die den privaten Schlüssel des Agent enthält und für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSPSKFile	Gibt den vollständigen Pfadnamen der Datei an, die den Pre-Shared Key des Agent enthält und für die verschlüsselte Kommunikation mit dem Zabbix Server oder Proxy verwendet wird.
TLSPSKIdentity	Gibt die Identitätszeichenfolge des Pre-Shared Key an, die für die verschlüsselte Kommunikation mit dem Zabbix Server oder Proxy verwendet wird.
TLSServerCertIssuer	Gibt den zulässigen Aussteller des Server-(Proxy-)Zertifikats an.
TLSServerCertSubject	Gibt den zulässigen Betreff des Server-(Proxy-)Zertifikats an.
UnsafeUserParameters	Erlaubt die Übergabe aller Zeichen in Argumenten an benutzerdefinierte Parameter.
UserParameter	Gibt einen zu überwachenden benutzerdefinierten Parameter an.
UserParameterDir	Gibt den Standardsuchpfad für UserParameter-Befehle an.

Beachten Sie:

- Die Standardwerte entsprechen den Daemon-Standards und nicht den Werten in den mitgelieferten Konfigurationsdateien.

- Werte unterstützen Umgebungsvariablen.
- Zabbix unterstützt Konfigurationsdateien nur in UTF-8-Kodierung ohne BOM.
- Kommentare, die mit # beginnen, werden nur am Anfang der Zeile unterstützt.

Parameterdetails

Alias

Legen Sie einen Alias für einen Datenpunktschlüssel fest. Damit kann ein langer und komplexer Datenpunktschlüssel durch einen kürzeren und einfacheren ersetzt werden.
 Es können mehrere `Alias`-Parameter vorhanden sein. Mehrere Parameter mit demselben `Alias`-Schlüssel sind nicht zulässig.
 Verschiedene `Alias`-Schlüssel können auf denselben Datenpunktschlüssel verweisen.
 Aliase können in `HostMetadataItem`, jedoch nicht in den Parametern `HostnameItem` oder `PerfCounter` verwendet werden.

Beispiel 1: Abrufen der Auslagerungsdateinutzung in Prozent vom Server.

```
Alias=pg_usage:perf_counter[\Paging File(_Total)\% Usage]
```

Nun kann der Kurzschlüssel `pg_usage` zum Abrufen von Daten verwendet werden.

Beispiel 2: Abrufen der CPU-Auslastung mit Standard- und benutzerdefinierten Parametern.

```
Alias=cpu.load:system.cpu.load
Alias=cpu.load[*]:system.cpu.load[*]
```

Damit können Sie den Schlüssel `cpu.load` verwenden, um die CPU-Auslastung mit Standardparametern abzurufen, sowie `cpu.load[percpu,avg15]`, um bestimmte Daten zur CPU-Auslastung abzurufen.

Beispiel 3: Ausführen mehrerer Low-Level-Discovery-Regeln, die dieselben Discovery-Datenpunkte verarbeiten.

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

Nun können mehrere Discovery-Regeln mit `vfs.fs.discovery` eingerichtet werden, wobei für jede Regel unterschiedliche Parameter verwendet werden können, z. B. `vfs.fs.discovery[foo]`, `vfs.fs.discovery[bar]` usw.

AllowKey

Erlaubt die Ausführung derjenigen Datenpunkt-Schlüssel, die einem Muster entsprechen. Das Schlüsselmuster ist ein Wildcard-Ausdruck, der das Zeichen * unterstützt, um eine beliebige Anzahl beliebiger Zeichen zu matchen.
 Mehrere Regeln für die Schlüsselübereinstimmung können in Kombination mit `DenyKey`, `AllowKeyRegexp` und `DenyKeyRegexp` definiert werden. Die Parameter werden nacheinander entsprechend ihrer Reihenfolge verarbeitet. Die Reihenfolge bestimmt, welche übereinstimmende Regel Vorrang hat. Siehe auch: Einschränken von Agent-Prüfungen.

AllowKeyRegexp

Erlaubt die Ausführung derjenigen Datenpunktschlüssel, die einem Muster entsprechen.
Das Schlüsselmuster ist ein regulärer Ausdruck.
 Mehrere Regeln für den Datenpunktschlüsselabgleich können in Kombination mit `DenyKeyRegexp`, `AllowKey` und `DenyKey` definiert werden. Die Parameter werden entsprechend ihrer Reihenfolge nacheinander verarbeitet. Die Reihenfolge bestimmt, welche übereinstimmende Regel Vorrang hat. Siehe auch: Einschränken von Agent-Prüfungen.

Beispiel:

```
AllowKeyRegexp=^system\.run\[(?:/usr)?/bin/hostname(?: -[fsd])?\]$
```

BufferSend

Legt die Häufigkeit fest, mit der Werte aus dem Puffer an den Zabbix Server/Proxy gesendet werden (in Sekunden).

Standard: 5
 Bereich: 1-3600

BufferSize

Legt die maximale Anzahl von Werten im Speicherpuffer fest.
 Der Agent sendet alle gesammelten Daten an den Zabbix-Server oder Proxy, wenn der Puffer voll ist.

Standard: 100
 Bereich: 2-65535

DebugLevel

Legt den Debug-Level fest: - *0* - grundlegende Informationen zum Starten und Stoppen von Zabbix-Prozessen; - *1* - kritische Informationen; - *2* - Fehlerinformationen; - *3* - Warnungen; - *4* - für das Debugging (liefert viele Informationen); - *5* - erweitertes Debugging (liefert noch mehr Informationen).

Standard: 3
 Bereich: 0-5

DenyKey

Verhindert die Ausführung derjenigen Datenpunktschlüssel, die einem Muster entsprechen.
Das Schlüsselmuster ist ein Platzhalterausdruck, der das Zeichen * unterstützt, um eine beliebige Anzahl beliebiger Zeichen zu finden.
 Mehrere Regeln für den Datenpunktschlüssel können in Kombination mit AllowKey, AllowKeyRegexp und DenyKeyRegexp definiert werden.
Die Parameter werden entsprechend ihrer Reihenfolge nacheinander verarbeitet.
Die Reihenfolge bestimmt, welche übereinstimmende Regel Vorrang hat.
Siehe auch: Einschränken von Agent-Prüfungen.

Standard: DenyKey=system.run[*]

DenyKeyRegexp

Verhindert die Ausführung von Datenpunktschlüsseln, die einem Muster entsprechen.
Das Schlüsselmuster ist ein regulärer Ausdruck.
 Mehrere Regeln für den Abgleich von Schlüsseln können in Kombination mit AllowKeyRegexp, AllowKey und DenyKey definiert werden. Die Parameter werden entsprechend ihrer Reihenfolge nacheinander verarbeitet. Die Reihenfolge bestimmt, welche übereinstimmende Regel Vorrang hat.
 Alle system.run[]-Datenpunkte werden standardmäßig abgelehnt (DenyKey=system.run[*]), sofern keine andere system.run[*]-Regel angegeben ist.

Siehe auch: Agent-Prüfungen einschränken.

EnableRemoteCommands

Ermöglicht Remote-Befehle vom Zabbix Server.
 Dieser Parameter ist **veraltet**. Verwenden Sie stattdessen AllowKey=system.run[*], AllowKeyRegexp=^system\.run\[.*\]$, DenyKey=system.run[*] oder DenyKeyRegexp=^system\.run\[.*\]$.
Dies ist abhängig vom Wert ein interner Alias für die Parameter AllowKey, AllowKeyRegexp, DenyKey und DenyKeyRegexp: - 0 - DenyKey=system.run[*] (DenyKeyRegexp=^system\.run\[.*\]$) - 1 - AllowKey=system.run[*] (AllowKeyRegexp=^system\.

Standard: 0
 Werte: 0 - nicht zulassen, 1 - zulassen

HeartbeatFrequency

Legt die Frequenz der Heartbeat-Nachrichten in Sekunden fest. Wird zur Überwachung der Verfügbarkeit aktiver Checks verwendet.
0 - Heartbeat-Nachrichten deaktiviert.

Standard: 60
 Bereich: 0-3600

HostInterface

Geben Sie die Host-Schnittstelle (IP-Adresse oder DNS-Name) an, die während der Autoregistrierung des Hosts verwendet wird. Dieser Wert wird verwendet, um die Schnittstelle im neu erstellten Host zu befüllen, und ermöglicht die explizite Konfiguration entweder einer IP- oder DNS-Adresse. Weitere Informationen finden Sie unter Using DNS as default interface.

Wenn nicht definiert, wird der Wert aus HostInterfaceItem übernommen.

Der Agent gibt einen Fehler aus und startet nicht, wenn der Wert die Grenze von 255 Zeichen überschreitet.

Bereich: 0-255 Zeichen

HostInterfaceItem

Geben Sie einen Datenpunkt an, um die Host-Schnittstelle (IP-Adresse oder DNS-Name) während der Autoregistrierung des Hosts zu bestimmen.
Dieser Wert wird nur verwendet, wenn HostInterface nicht definiert ist.
Weitere Informationen finden Sie unter DNS als Standardschnittstelle verwenden.

Während einer Autoregistrierungsanfrage protokolliert der Agent eine Warnmeldung, wenn der vom angegebenen Datenpunkt zurückgegebene Wert das Limit von 255 Zeichen überschreitet.

Der Datenpunkt system.run[] wird unabhängig von den Einstellungen AllowKey, AllowKeyRegexp, DenyKey oder DenyKeyRegexp unterstützt.

HostMetadata

Geben Sie die Metadaten an, die zur Identifizierung oder Unterscheidung des Hosts während der Autoregistrierung des Hosts (aktiver Agent) verwendet werden.
HostMetadata ermöglicht die Unterscheidung zwischen Hosts über den Hostnamen hinaus.

Wenn nicht definiert, wird der Wert aus HostMetadataItem abgerufen.

Der Agent gibt einen Fehler aus und wird nicht gestartet, wenn der angegebene Wert das Limit von 2034 Bytes überschreitet oder aus einer Nicht-UTF-8-Zeichenfolge besteht.
Wenn ein Parameter eine IP-Adresse oder einen DNS-Namen erwartet, werden Werte, die zwar gültiges UTF-8 enthalten, aber keine gültigen IP-Adressen oder DNS-Namen sind, ebenfalls abgelehnt und als ungültig gemeldet.

Mehrzeilige Metadaten werden nicht unterstützt – die Ausgabe wird beim ersten Zeilenumbruch abgeschnitten.

Bereich: 0–2034 Bytes

HostMetadataItem

Geben Sie einen Zabbix-Agent-Datenpunkt an, um Host-Metadaten abzurufen.
Diese Option wird nur verwendet, wenn `HostMetadata` nicht definiert ist.

Der Wert von `HostMetadataItem` wird bei jedem Versuch der Autoregistrierung abgerufen und nur während des Host-Autoregistrierungsprozesses (aktiver Agent) verwendet.
Mit `HostMetadataItem` können Hosts über den Hostnamen hinaus unterschieden werden.

Benutzerparameter, Performance-Counter und Aliase werden unterstützt.
Der Datenpunkt `system.run[]` wird unabhängig von den Einstellungen `AllowKey`, `AllowKeyRegexp`, `DenyKey` oder `DenyKeyRegexp` unterstützt.

Während einer Autoregistrierungsanfrage protokolliert der Agent eine Warnmeldung, wenn der vom angegebenen Datenpunkt zurückgegebene Wert das Limit von 65535 UTF-8-Codepunkten überschreitet.
Der vom Datenpunkt zurückgegebene Wert muss eine UTF-8-Zeichenfolge sein; andernfalls wird er ignoriert.
Wenn ein Parameter eine IP-Adresse oder einen DNS-Namen erwartet, werden Werte, die zwar gültiges UTF-8 enthalten, aber keine gültigen IP-Adressen oder DNS-Namen sind, ebenfalls abgelehnt und als ungültig gemeldet.

Mehrzeilige Metadaten werden nicht unterstützt – die Ausgabe wird beim ersten Zeilenumbruch abgeschnitten.

Hostname

Geben Sie eindeutige, groß-/kleinschreibungssensitive Hostnamen an. Eine durch Kommas getrennte Liste.
 Erforderlich für aktive Prüfungen und muss mit den auf dem Server konfigurierten Hostnamen übereinstimmen. Der Wert wird aus `HostnameItem` übernommen, wenn er nicht definiert ist.
Zulässige Zeichen: alphanumerisch, '.', ' ', '_' und '-'. Maximale Länge: 128 Zeichen pro Hostname, 2048 Zeichen für die gesamte Zeile.

Standard: Durch HostnameItem festgelegt

HostnameItem

Geben Sie einen Zabbix-Agent-Datenpunkt zum Abrufen des Hostnamens an.
 Diese Option wird nur verwendet, wenn `Hostname` nicht definiert ist.
 Benutzerparameter, Performance-Counter oder Aliase werden nicht unterstützt. Der Datenpunkt `system.run[]` wird jedoch unabhängig von den Werten von `AllowKey`, `AllowKeyRegexp`, `DenyKey` oder `DenyKeyRegexp` unterstützt.
Siehe auch eine detailliertere Beschreibung.

Standard: `system.hostname`

Include

Geben Sie ein Verzeichnis oder einzelne Dateien an, die in die Konfigurationsdatei aufgenommen werden sollen (standardmäßig unter `C:\Program Files\Zabbix Agent`, wenn der Zabbix Agent mithilfe von Windows-MSI-Installationspaketen installiert wurde; in dem bei der Installation angegebenen Ordner, wenn der Zabbix Agent als ZIP-Archiv installiert wurde).

Allgemeine Regeln für den Parameter `Include`:

- Der Pfad kann relativ zum Speicherort der Datei `zabbix_agentd.conf` angegeben werden.
- Sie können mehrere `Include`-Parameter in einer Konfigurationsdatei angeben.
- Zabbix verarbeitet den Parameter `Include`, sobald er gelesen wird. Ziehen Sie daher in Betracht, ihn am Ende der Konfigurationsdatei zu platzieren.

Regeln für eingebundene Dateien:

- Jede Datei muss lesbar sein.
- Jede Datei muss eine korrekte Syntax aufweisen. Andernfalls wird die Komponente nicht gestartet.

Regeln für das Einbinden von Verzeichnissen oder Dateimustern:

- Gehen Sie nicht von einer bestimmten Reihenfolge beim Einbinden aus (Dateien werden beispielsweise nicht alphabetisch eingebunden).
- Definieren Sie denselben Parameter nicht in mehreren eingebundenen Dateien (beispielsweise, um eine allgemeine Einstellung durch eine spezifische zu überschreiben), da nicht vorhersehbar ist, welcher Wert verwendet wird.
- Achten Sie auf Sicherungskopien, die einige Texteditoren automatisch erstellen (beispielsweise kann beim Bearbeiten von `include/my_specific.conf` die Datei `include/my_specific.conf.BAK` erstellt werden), da beide Dateien eingebunden werden.

Um nur relevante Dateien aus einem Verzeichnis einzubinden, verwenden Sie den Platzhalter (*) für den Musterabgleich.

Beispiel:

```
Include=C:\Program Files\Zabbix Agent\zabbix_agentd.d\*.conf
```

ListenBacklog

Legt die maximale Anzahl ausstehender Verbindungen in der TCP-Warteschlange fest.
 Der Standardwert ist eine fest codierte Konstante, die vom System abhängt.
 Der maximal unterstützte Wert hängt ebenfalls vom System ab; zu hohe Werte können stillschweigend auf das „implementationsspezifische Maximum“ gekürzt werden.

Standard: SOMAXCONN
 Bereich: 0 - INT_MAX

ListenIP

Geben Sie die IP-Adresse an, auf der der Agent Verbindungen vom Server/Proxy entgegennimmt. Eine durch Kommas getrennte Liste.

Standard: 0.0.0.0

ListenPort

Geben Sie den Listen-Port des Agent für Verbindungen vom Server/Proxy an.

Standard: 10050
 Bereich: 1024-32767

LogFile

Geben Sie die Logdatei des Agent an.

Standard: c:\zabbix_agentd.log
 Erforderlich: Ja, wenn LogType auf file gesetzt ist; andernfalls nein

LogFileSize

Legt die maximale Größe der Protokolldatei fest (in MB).
0 - automatische Protokollrotation deaktivieren.
Hinweis: Wenn das Limit für die Größe der Protokolldatei erreicht wird und die Dateirotation aus irgendeinem Grund fehlschlägt, wird die vorhandene Protokolldatei abgeschnitten und neu begonnen.

Standard: 1
 Bereich: 0-1024

LogRemoteCommands

Aktivieren Sie die Protokollierung der ausgeführten Shell-Befehle als Warnungen.
 Befehle werden nur protokolliert, wenn sie remote ausgeführt werden.
 Protokolleinträge werden nicht erstellt, wenn system.run[] lokal durch die Parameter HostMetadataItem, HostInterfaceItem oder HostnameItem gestartet wird.

Standard: 0
 Werte: 0 - deaktiviert, 1 - aktiviert

LogType

Geben Sie den Typ der Protokollausgabe an: - *file* - schreibt das Protokoll in die durch den Parameter LogFile angegebene Datei; - *system* - schreibt das Protokoll in das Windows-Ereignisprotokoll; - *console* - schreibt das Protokoll in die Standardausgabe.

Standardwert: file

MaxLinesPerSecond

Legt die maximale Anzahl neuer Zeilen pro Sekunde fest, die beim Verarbeiten von aktiven Prüfungen log, logrt oder eventlog an den Zabbix Server/Proxy gesendet werden.
 Der angegebene Wert wird durch den Parameter maxlines überschrieben, der im log, logrt oder eventlog Datenpunkt-Schlüssel angegeben ist.
Hinweis: Zabbix verarbeitet 10-mal mehr neue Zeilen als in MaxLinesPerSecond festgelegt, um die erforderliche Zeichenfolge in Log-Datenpunkten zu suchen.

Standard: 20
 Bereich: 1-1000

PerfCounter

Definieren Sie einen neuen Parameter <parameter_name>, der den Durchschnittswert des Systemleistungsindikators <perf_counter_path> für den angegebenen Zeitraum <period> (in Sekunden) darstellt.
Syntax: <parameter_name>,”<perf_counter_pat

Wenn Sie beispielsweise die durchschnittliche Anzahl von Prozessorunterbrechungen pro Sekunde für die letzte Minute erhalten möchten, können Sie einen neuen Parameter ”interrupts” wie folgt definieren:

```
PerfCounter = interrupts,"\Processor(0)\Interrupts/sec",60
```

Bitte beachten Sie die doppelten Anführungszeichen um den Pfad des Leistungsindikators. Der Parametername (interrupts) ist als Datenpunkt-Schlüssel zu verwenden, wenn ein Datenpunkt erstellt wird. Stichproben zur Berechnung des Durchschnittswerts werden jede Sekunde erfasst.
Sie können typeperf -qx ausführen, um die Liste aller in Windows verfügbaren Leistungsindikatoren anzuzeigen.

PerfCounterEn

Definieren Sie einen neuen Parameter <parameter_name>, der den Durchschnittswert des Systemleistungsindikators <perf_counter_path> für den angegebenen Zeitraum <period> (in Sekunden) liefert. Im Vergleich zu PerfCounter müssen die perfcounter-Pfade auf Englisch angegeben werden. Wird nur unter **Windows Server 2008/Vista** und höher unterstützt.
Syntax: <parameter_name>,"<perf_counter_path>",<period>

Wenn Sie beispielsweise die durchschnittliche Anzahl der Prozessorunterbrechungen pro Sekunde für die letzte Minute erhalten möchten, können Sie einen neuen Parameter "interrupts" wie folgt definieren:

```
PerfCounterEn = interrupts,"\Processor(0)\Interrupts/sec",60
```

Beachten Sie bitte die doppelten Anführungszeichen um den Pfad des Leistungsindikators. Der Parametername (interrupts) wird als item key verwendet, wenn ein Datenpunkt erstellt wird. Beispielwerte zur Berechnung des Durchschnittswerts werden jede Sekunde erfasst.
Die Liste der englischen Zeichenfolgen finden Sie, indem Sie den folgenden Registrierungsschlüssel anzeigen: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009`.

RefreshActiveChecks

Geben Sie die Häufigkeit der Aktualisierung der Liste der aktiven Prüfungen in Sekunden an.
 Beachten Sie, dass nach einem fehlgeschlagenen Versuch, die aktiven Prüfungen zu aktualisieren, der nächste Aktualisierungsversuch in 60 Sekunden unternommen wird.

Standard: 5
 Bereich: 1-86400

Server

Geben Sie die IP-Adressen, optional in CIDR-Notation, oder DNS-Namen von Zabbix-Servern/Proxys an. Eine durch Kommas getrennte Liste.
 Eingehende Verbindungen werden nur von den hier aufgeführten Hosts akzeptiert. Wenn die IPv6-Unterstützung aktiviert ist, werden `127.0.0.1`, `::127.0.0.1` und `::ffff:127.0.0.1` als gleich behandelt, und `::/0` erlaubt jede IPv4- oder IPv6-Adresse. `0.0.0.0/0` kann verwendet werden, um jede IPv4-Adresse zuzulassen. Beachten Sie, dass "IPv4-kompatible IPv6-Adressen" (`0000::/96`-Präfix) unterstützt werden, aber gemäß RFC4291 veraltet sind. Leerzeichen sind zulässig.

Beispiel:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Erforderlich: ja, wenn StartAgents nicht explizit auf 0 gesetzt ist

ServerActive

Geben Sie die Adresse des Zabbix-Servers/Proxy oder die Clusterkonfiguration an, von der aktive Prüfungen abgerufen werden sollen.
 Die Adresse des Servers/Proxy ist eine IP-Adresse oder ein DNS-Name mit optionalem Port, getrennt durch einen Doppelpunkt.
Die Clusterkonfiguration ist eine oder mehrere Adressen von Mitgliedern einer Server- oder Proxygruppe, getrennt durch Semikolon. Mehrere Zabbix-Server/Cluster und Zabbix-Proxys können angegeben werden, getrennt durch Komma. Sofern keine Proxygruppen verwendet werden, sollte von jedem Zabbix-Server/Cluster nicht mehr als ein Zabbix-Proxy angegeben werden. Wenn ein Zabbix-Proxy angegeben ist, darf der Zabbix-Server/das Cluster für diesen Proxy nicht angegeben werden.
Mehrere durch Kommas getrennte Adressen können angegeben werden, um mehrere unabhängige Zabbix-Server parallel zu verwenden. Leerzeichen sind zulässig.
Wenn der Port nicht angegeben ist, wird der Standardport verwendet.
IPv6-Adressen müssen in eckige Klammern gesetzt werden, wenn für diesen Host ein Port angegeben ist. Wenn kein Port angegeben ist, sind eckige Klammern für IPv6-Adressen optional.
Wenn dieser Parameter nicht angegeben ist, sind aktive Prüfungen deaktiviert.

Beispiel für Zabbix-Proxy:

```
ServerActive=127.0.0.1:10051
```

Beispiel für eine Zabbix-Proxygruppe:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

Beispiel für mehrere Server:

```
ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
```

Beispiel für Hochverfügbarkeit:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

Beispiel für Hochverfügbarkeit mit zwei Clustern und einem Server:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

Bereich: Die Anzahl der in `ServerActive` aufgeführten aktiven Server plus die Anzahl der für passive Prüfungen in `StartAgents` angegebenen vorab erzeugten Instanzen muss kleiner als 64 sein.

SourceIP

Geben Sie die Quell-IP-Adresse an für:

- ausgehende Verbindungen zum Zabbix Server oder Zabbix Proxy;
- Verbindungen, die beim Ausführen bestimmter Datenpunkte (`web.page.get`, `net.tcp.port` usw.) hergestellt werden.

StartAgents

Legt die Anzahl der vorab erzeugten Instanzen von `zabbix_agentd` für die Verarbeitung passiver Prüfungen fest.
 Wenn auf 0 gesetzt, sind passive Prüfungen deaktiviert und der Agent lauscht auf keinem TCP-Port.

Standard: 10
 Bereich: 0-100 (*)

Timeout

Geben Sie an, wie lange beim Aufbau der Verbindung und beim Austausch von Daten mit dem Zabbix Proxy oder Server gewartet werden soll (in Sekunden).

Dieser Parameter definiert die Dauer verschiedener Kommunikationsvorgänge, einschließlich:

- Warten auf eine Antwort vom Zabbix Server;
- Senden von Anfragen an den Zabbix Server, einschließlich Anfragen zur Datenpunktkonfiguration und Datenpunktdaten in aktiven Prüfungen;
- Abrufen von Protokolldaten über die Überwachung von Logdateien oder dem Windows-Ereignisprotokoll;
- Senden von Heartbeat-Nachrichten;
- maximale Dauer für `vfs.*`-Prüfungen;
- Verwendung durch Zabbix-Agent-Module;
- dient als Fallback in Szenarien, in denen ein Server oder Proxy älter als Version 7.0 Prüfungen ohne Timeouts sendet.

Dieses Timeout wird **nicht** für jene Agent-Prüfungen verwendet, die in der Frontend konfigurierbare Timeout-Einstellungen haben (auf globaler Ebene, auf Proxy-Ebene oder auf Datenpunktebene).

Standard: 3
 Bereich: 1-30

TLSAccept

Geben Sie das für eingehende Verbindungen erforderliche Verschlüsselungsniveau an.
 Wird für passive Prüfungen verwendet.
 Mehrere Werte können durch Komma getrennt angegeben werden: - *unencrypted* - Verbindungen ohne Verschlüsselung akzeptieren (Standard); - *psk* - Verbindungen mit TLS und einem vorab gemeinsam genutzten Schlüssel (PSK) akzeptieren; - *cert* - Verbindungen mit TLS und einem Zertifikat akzeptieren.

Pflicht: ja, wenn TLS-Zertifikat- oder PSK-Parameter definiert sind (auch für eine *unencrypted* Verbindung); andernfalls nein

TLSCAFile

Geben Sie den vollständigen Pfadnamen der Datei an, die die Zertifikate der obersten CA(s) für die Überprüfung von Peer-Zertifikaten enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

TLSCertFile

Geben Sie den vollständigen Pfadnamen der Datei an, die das Agent-Zertifikat oder die Zertifikatskette enthält und für verschlüsselte Kommunikation mit Zabbix-Komponenten verwendet wird.

TLSConnect

Geben Sie die für Agent-Verbindungen zum Zabbix-Server oder Proxy erforderliche Verschlüsselungsstufe an.
 Wird für aktive Prüfungen verwendet.
 Es kann nur ein Wert angegeben werden: - *unencrypted* - Verbindung ohne Verschlüsselung herstellen (Standard); - *psk* - Verbindung mit TLS und einem vorab gemeinsam genutzten Schlüssel (PSK) herstellen; - *cert* - Verbindung mit TLS und einem Zertifikat herstellen.

Pflicht: ja, wenn TLS-Zertifikat- oder PSK-Parameter definiert sind (auch für eine *unencrypted* Verbindung); andernfalls nein

TLSCRLFile

Geben Sie den vollständigen Pfadnamen der Datei an, die widerrufene Zertifikate enthält. Dieser Parameter wird für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet.

TLSKeyFile

Geben Sie den vollständigen Pfadnamen der Datei an, die den privaten Schlüssel des Agent enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

TLSPSKFile

Geben Sie den vollständigen Pfadnamen der Datei an, die den vorab gemeinsam genutzten Schlüssel des Agent enthält und für verschlüsselte Kommunikation mit dem Zabbix Server oder Proxy verwendet wird.

TLSPSKIdentity

Geben Sie die Identitätszeichenfolge des vorab gemeinsam genutzten Schlüssels an, die für verschlüsselte Kommunikation mit dem Zabbix-Server oder Proxy verwendet wird.

TLSServerCertIssuer

Geben Sie den zulässigen Aussteller des Server- (Proxy-) Zertifikats an.

TLSServerCertSubject

Geben Sie den zulässigen Zertifikatsbetreff des Servers (Proxy) an.

UnsafeUserParameters

Erlaubt, dass alle Zeichen in Argumenten an benutzerdefinierte Parameter übergeben werden. Die folgenden Zeichen sind nicht erlaubt: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
Zusätzlich sind Zeilenumbrüche nicht erlaubt.

Standard: 0
 Werte: 0 - nicht erlauben, 1 - erlauben

UserParameter

Geben Sie einen benutzerdefinierten Parameter an, der überwacht werden soll. Es können mehrere benutzerdefinierte Parameter vorhanden sein.
Format: `UserParameter=<key>,<shell command>`
Beachten Sie, dass der Shell-Befehl keinen leeren String oder nur EOL zurückgeben darf. Shell-Befehle dürfen relative Pfade verwenden, wenn der Parameter `UserParameterDir` angegeben ist.

Beispiel:

```
UserParameter=system.test,dir /b | find /c /v ""
UserParameter=check_cpu,.\custom\_script.bat
```

UserParameterDir

Geben Sie den standardmäßigen Suchpfad für `UserParameter`-Befehle an. Wenn diese Option verwendet wird, wechselt der Agent sein Arbeitsverzeichnis vor der Ausführung eines Befehls in das hier angegebene Verzeichnis. Dadurch können `UserParameter`-Befehle ein relatives Präfix `./` anstelle eines vollständigen Pfads verwenden. Es ist nur ein Eintrag zulässig.

Beispiel:

```
UserParameterDir=C:\opt\myscripts
```

Zabbix Agent unter Microsoft Windows

Agent konfigurieren

Beide Generationen der Zabbix Agenten werden als Windows-Dienst ausgeführt.
Ersetzen Sie für Zabbix Agent 2 in den nachstehenden Anweisungen `agentd` durch `agent2`.

Sie können eine einzelne Instanz des Zabbix Agent oder mehrere Instanzen des Agent auf einem Microsoft Windows Host ausführen. Eine einzelne Instanz kann entweder Folgendes verwenden:

- die Standardkonfigurationsdatei, die sich im selben Verzeichnis wie die Agent-Binärdatei befindet;
- eine in der Befehlszeile angegebene Konfigurationsdatei.

Bei mehreren Instanzen muss jede Agent-Instanz über eine eigene Konfigurationsdatei verfügen (eine der Instanzen kann die Standardkonfigurationsdatei verwenden).

Eine Beispielkonfigurationsdatei ist im Zabbix-Quellarchiv verfügbar als:

- `conf/zabbix_agentd.conf` für den Zabbix Agent;
- `conf/zabbix_agent2.conf` für Zabbix agent2.

Wenn Sie den Zabbix Agent/Agent 2 für Windows aus einem Archiv als Dienst installieren möchten, ohne die Konfigurationsdatei explizit anzugeben, müssen Sie vor der Installation des Agent:

- `conf/zabbix_agentd.conf` manuell in das Verzeichnis kopieren, in dem `zabbix_agentd.exe` installiert wird;
- `conf/zabbix_agent2.conf` und das Verzeichnis `conf/zabbix_agent2.d` manuell in das Verzeichnis kopieren, in dem `zabbix_agent2.exe` installiert wird.

Weitere Informationen zur Konfiguration des Zabbix Windows Agent finden Sie in den Optionen der Agent-Konfigurationsdatei oder der Konfigurationsdatei von Agent 2.

Hostname-Parameter

Um aktive Prüfungen auf einem Host durchzuführen, muss beim Zabbix Agent der Hostname definiert sein.
Außerdem muss der auf der Agent-Seite festgelegte Hostname exakt mit dem für den Host im Frontend konfigurierten *Host name* übereinstimmen.

Der Hostname auf der Agent-Seite kann entweder über den Parameter `Hostname` oder `HostnameItem` in der Agent-Konfigurationsdatei definiert werden. Wenn keiner dieser Parameter angegeben ist, werden die Standardwerte verwendet.

Der Standardwert für den Parameter `HostnameItem` ist der vom Agent-Schlüssel `system.hostname` zurückgegebene Wert.
Unter Windows wird das Ergebnis der Funktion `gethostname()` zurückgegeben, die Namespace-Provider abfragt, um den lokalen Hostnamen zu bestimmen.
Wenn kein Namespace-Provider antwortet, wird der NetBIOS-Name zurückgegeben.

Der Standardwert für `Hostname` ist der vom Parameter `HostnameItem` zurückgegebene Wert.
Wenn also beide Parameter nicht angegeben sind, entspricht der tatsächliche Hostname dem NetBIOS-Namen des Hosts. Der Zabbix Agent verwendet den NetBIOS-Hostnamen, um die Liste der aktiven Prüfungen vom Zabbix Server abzurufen und die Ergebnisse an ihn zu senden.

Der Schlüssel `system.hostname` unterstützt zwei optionale Parameter: *type* und *transform*.

Type bestimmt den Typ des Namens, den der Datenpunkt zurückgeben soll:

- *netbios* (Standard) - gibt den NetBIOS-Hostnamen zurück, der auf 15 Zeichen begrenzt ist und ausschließlich in GROSSBUCHSTABEN geschrieben wird;
- *host* - unterscheidet zwischen Groß- und Kleinschreibung und gibt den vollständigen, tatsächlichen Windows-Hostnamen (ohne Domäne) zurück;
- *shorthost* - gibt den Teil des Hostnamens vor dem ersten Punkt zurück.
  Wenn der Name keinen Punkt enthält, wird die vollständige Zeichenfolge zurückgegeben.
- *fqdn* - gibt den Fully Qualified Domain Name (ohne abschließenden Punkt) zurück.

Mit *Transform* können Sie eine zusätzliche Transformationsregel für den Hostnamen angeben:

- *none* (Standard) - verwendet die ursprüngliche Groß- und Kleinschreibung;
- *lower* - wandelt den Text in Kleinbuchstaben um.

Um die Konfiguration der Datei `zabbix_agentd.conf` zu vereinfachen und zu vereinheitlichen, können daher verschiedene Ansätze verwendet werden:

1. Lassen Sie die Parameter `Hostname` oder `HostnameItem` undefiniert. Der Zabbix Agent verwendet dann den NetBIOS-Hostnamen als Hostnamen.
2. Lassen Sie den Parameter `Hostname` undefiniert und definieren Sie `HostnameItem` wie folgt:
   - `HostnameItem=system.hostname[host]` - damit der Zabbix Agent den vollständigen, tatsächlichen Windows-Hostnamen unter Berücksichtigung der Groß- und Kleinschreibung als Hostnamen verwendet.
   - `HostnameItem=system.hostname[shorthost,lower]` - damit der Zabbix Agent nur den Teil des Hostnamens vor dem ersten Punkt verwendet und in Kleinbuchstaben umwandelt.
   - `HostnameItem=system.hostname[fqdn]` - damit der Zabbix Agent den Fully Qualified Domain Name als Hostnamen verwendet.

Der Hostname wird auch als Teil des Windows-Dienstnamens verwendet, der zum Installieren, Starten, Beenden und Deinstallieren des Windows-Dienstes dient. Wenn die Zabbix-Agent-Konfigurationsdatei beispielsweise `Hostname=Windows_db_server` angibt, wird der Agent als Windows-Dienst `Zabbix Agent [Windows_db_server]` installiert. Damit für jede Zabbix-Agent-Instanz ein anderer Windows-Dienstname verwendet wird, muss jede Instanz daher einen anderen Hostnamen verwenden.

Installation des Agent als Windows-Dienst

Kopieren Sie vor der Installation des Agent `conf/zabbix_agentd.conf` manuell in das Verzeichnis, in dem zabbix_agentd.exe installiert wird.

Um eine einzelne Instanz des Zabbix Agent mit der Standard-Konfigurationsdatei zu installieren:

```
zabbix_agentd.exe --install
```

Attention:
Auf einem 64-Bit-System ist eine 64-Bit-Version des Zabbix Agent erforderlich, damit alle Prüfungen im Zusammenhang mit der Ausführung von 64-Bit-Prozessen korrekt funktionieren.

Wenn Sie eine andere Konfigurationsdatei als die Standarddatei verwenden möchten, sollten Sie für die Dienstinstallation den folgenden Befehl verwenden:

```
zabbix_agentd.exe --config <your_configuration_file> --install
```

Es sollte ein vollständiger Pfad zur Konfigurationsdatei angegeben werden.

Mehrere Instanzen des Zabbix Agent können als Dienste wie folgt installiert werden:

```
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
```

Der installierte Dienst sollte nun in der Systemsteuerung sichtbar sein.

Starten des Agent

Um den Agent-Dienst zu starten, können Sie die Systemsteuerung verwenden oder dies über die Befehlszeile tun.

Um eine einzelne Instanz des Zabbix Agent mit der Standard-Konfigurationsdatei zu starten:

```
zabbix_agentd.exe --start
```

Um eine einzelne Instanz des Zabbix Agent mit einer anderen Konfigurationsdatei zu starten:

```
zabbix_agentd.exe --config <your_configuration_file> --start
```

Um eine von mehreren Instanzen des Zabbix Agent zu starten:

```
zabbix_agentd.exe --config <configuration_file_for_this_instance> --start --multiple-agents
```

Stoppen des Agent

Um den Agent-Dienst zu stoppen, können Sie die Systemsteuerung verwenden oder dies über die Befehlszeile tun.

Um eine einzelne Instanz des Zabbix Agent zu stoppen, die mit der Standard-Konfigurationsdatei gestartet wurde:

```
zabbix_agentd.exe --stop
```

Um eine einzelne Instanz des Zabbix Agent zu stoppen, die mit einer anderen Konfigurationsdatei gestartet wurde:

```
zabbix_agentd.exe --config <your_configuration_file> --stop
```

Um eine von mehreren Instanzen des Zabbix Agent zu stoppen:

```
zabbix_agentd.exe --config <configuration_file_for_this_instance> --stop --multiple-agents
```

Deinstallation des Agent-Windows-Dienstes

Um eine einzelne Instanz von Zabbix Agent mit der Standard- Konfigurationsdatei zu deinstallieren:

```
zabbix_agentd.exe --uninstall
```

Um eine einzelne Instanz von Zabbix Agent mit einer nicht standardmäßigen Konfigurationsdatei zu deinstallieren:

```
zabbix_agentd.exe --config <your_configuration_file> --uninstall
```

Um mehrere Instanzen von Zabbix Agent aus Windows-Diensten zu deinstallieren:

```
zabbix_agentd.exe --config <configuration_file_for_instance_1> --uninstall --multiple-agents
zabbix_agentd.exe --config <configuration_file_for_instance_2> --uninstall --multiple-agents
...
zabbix_agentd.exe --config <configuration_file_for_instance_N> --uninstall --multiple-agents
```

Einschränkungen

Der Zabbix Agent für Windows unterstützt keine nicht standardmäßigen Windows-Konfigurationen, bei denen CPUs ungleichmäßig über NUMA-Knoten verteilt sind. Wenn logische CPUs ungleichmäßig verteilt sind, sind CPU-Leistungsmetriken möglicherweise für einige CPUs nicht verfügbar. Wenn es beispielsweise 72 logische CPUs mit 2 NUMA-Knoten gibt, müssen beide Knoten jeweils 36 CPUs haben.

Zabbix Agent unter IBM AIX

Übersicht

Vorkompilierte Zabbix-Agent-Binärdateien für IBM AIX stehen als `.tar.gz`-Archiv zum Download zur Verfügung. Auf dieser Seite wird beschrieben, wie Sie den Agent aus dem Archiv installieren, ihn manuell oder mit dem System Resource Controller (SRC) ausführen, ihn beim Systemstart starten und aus dem SRC entfernen.

Agent installieren

Der AIX-Befehl `tar` unterstützt die Option `-z` nicht. Dekomprimieren Sie das Archiv daher mit `gunzip`. Verwenden Sie die Option `-o` von `tar`, damit die extrahierten Dateien dem Benutzer gehören, der sie extrahiert, und nicht den numerischen Benutzer- und Gruppen-IDs, die im Archiv gespeichert sind. Um den Agent beispielsweise unter `/opt/zabbix` abzulegen (als Root ausführen; `/opt/zabbix` darf noch nicht existieren):

```
gunzip -c zabbix_agent-7.4.7-aix-7.2-powerpc-openssl.tar.gz | tar -xof -
mv zabbix_agent-7.4.7-aix-7.2-powerpc-openssl /opt/zabbix
```

Im AIX-Archiv befindet sich die Agent-Binärdatei in `usr/sbin/` und die Konfigurationsdatei in `etc/zabbix/`. Passen Sie vor dem Start des Agents `/opt/zabbix/etc/zabbix/zabbix_agentd.conf` nach Bedarf an (mindestens die Parameter `Server`, `ServerActive` und `Hostname`).

Wenn der Agent als Root gestartet wird, wechselt er zum Benutzer `zabbix` (oder zu dem Benutzer, der mit dem Parameter `User` angegeben ist), sofern in der Konfigurationsdatei nicht `AllowRoot=1` gesetzt ist. Dieser Benutzer muss auf dem System vorhanden sein. Andernfalls wird der Agent nicht gestartet. Falls der Benutzer `zabbix` noch nicht existiert, erstellen Sie ihn zusammen mit einer Gruppe desselben Namens:

```
mkgroup zabbix
mkuser pgrp=zabbix login=false rlogin=false zabbix
```

Die Attribute `login=false` und `rlogin=false` verhindern, dass das Konto zum Anmelden verwendet wird. Der Agent kann dennoch unter diesem Benutzer ausgeführt werden.

Agent manuell starten

Der Agent kann manuell gestartet werden:

```
/opt/zabbix/usr/sbin/zabbix_agentd -c /opt/zabbix/etc/zabbix/zabbix_agentd.conf
```

Agent mit SRC ausführen

Agent registrieren

Der Agent kann auch beim System Resource Controller (SRC) registriert werden. Wenn er über SRC gestartet wird, muss der Agent im Vordergrundmodus (Option `-f`) ausgeführt werden. Andernfalls verliert SRC den Überblick über den Agent-Prozess und kann ihn nicht stoppen. Beispiel (als Root ausführen):

```
mkssys \
  -s zabbix_agentd \
  -p /opt/zabbix/usr/sbin/zabbix_agentd \
  -a "-c /opt/zabbix/etc/zabbix/zabbix_agentd.conf -f" \
  -u zabbix \
  -G tcpip \
  -S -n 15 -f 9 \
  -R
```

In diesem Beispiel:

- `-s` legt den Subsystemnamen fest, der mit den SRC-Befehlen verwendet wird;
- `-p` legt den vollständigen Pfad zur Agent-Binärdatei fest;
- `-a` legt die Befehlszeilenargumente des Agents als eine Zeichenfolge in Anführungszeichen fest; SRC teilt diese Zeichenfolge beim Starten des Agents in einzelne Argumente auf (es wird nur eine Option `-a` akzeptiert);
- `-u` legt den Benutzer fest, unter dem der Agent ausgeführt wird; der Benutzername wird beim Erstellen des Subsystems in eine numerische Benutzer-ID umgewandelt, daher muss der Benutzer bereits vorhanden sein;
- `-G` fügt den Agent zur Subsystemgruppe `tcpip` hinzu, sodass Gruppenbefehle auch auf ihn angewendet werden (beispielsweise stoppt `stopsrc -g tcpip` auch den Agent);
- `-S -n 15 -f 9` weist SRC an, den Agent mit Signalen zu steuern: `stopsrc` sendet `SIGTERM` (15) für ein normales Stoppen und `SIGKILL` (9) für ein erzwungenes Stoppen (`stopsrc -f`);
- `-R` weist SRC an, den Agent neu zu starten, wenn er unerwartet beendet wird (beispielsweise wenn einer seiner Prozesse abstürzt); nach `stopsrc` wird der Agent nicht neu gestartet.

Da die Benutzer-ID in der Subsystemdefinition gespeichert wird, muss die Definition aktualisiert werden, wenn der Benutzer `zabbix` mit einer anderen Benutzer-ID neu erstellt wird:

```
chssys -s zabbix_agentd -u zabbix
```

Agent starten und stoppen

Der Agent kann mit `startsrc`, `lssrc` und `stopsrc` gestartet, abgefragt und gestoppt werden:

```
startsrc -s zabbix_agentd
lssrc -s zabbix_agentd
stopsrc -s zabbix_agentd
```

Vermeiden Sie das erzwungene Stoppen (`stopsrc -f`). Dabei wird nur der Hauptprozess des Agents beendet; die anderen Agent-Prozesse laufen weiter und halten den Listening-Port geöffnet. Daher kann der Agent erst wieder gestartet werden, wenn diese Prozesse manuell beendet wurden (zum Beispiel mit `kill -9`).

Agent beim Systemstart starten

Die Registrierung des Agents bei SRC bewirkt nicht, dass er beim Systemstart gestartet wird. Um den Agent automatisch beim Systemstart zu starten, fügen Sie einen Eintrag zu `/etc/inittab` hinzu:

```
mkitab "zabbix_agentd:2:once:/usr/bin/startsrc -s zabbix_agentd >/dev/console 2>&1"
```

Entfernen des Agent aus SRC

Um den Agent aus dem Systemstart und aus SRC zu entfernen:

```
stopsrc -s zabbix_agentd
rmitab zabbix_agentd
rmssys -s zabbix_agentd
```

Agent 2

Übersicht Zabbix Agent 2 ist eine neue Generation des Zabbix Agent, geschrieben in Go (wobei ein Teil des C-Codes aus dem Zabbix Agent wiederverwendet wird). Er ist dafür konzipiert, um:

- die Anzahl der TCP-Verbindungen zu reduzieren;
- eine verbesserte Parallelität von Prüfungen bereitzustellen;
- sich mithilfe von Plugins, die einfache Prüfungen mit minimalem Code bereitstellen, leicht erweitern zu lassen und komplexe Prüfungen zu unterstützen, die aus lang laufenden Skripten und eigenständiger Datenerfassung mit periodischer Berichterstattung bestehen;
- als Ersatz für den Zabbix Agent zu dienen und alle bisherigen Funktionen zu unterstützen.

Alle Metriken in Zabbix Agent 2 werden von Plugins erfasst.

Diese Funktionen sind spezifisch für Plugins:

- Geplante und flexible Intervalle für passive und aktive Prüfungen
- Verwaltung der Aufgabenwarteschlange unter Berücksichtigung von Zeitplan und Aufgabenparallelität
- Zeitüberschreitungen auf Plugin-Ebene
- Kompatibilitätsprüfung von Zabbix Agent 2 und seinen Plugins beim Start

Es gibt zwei Arten von Plugins:

- Integrierte Plugins
- Ladbare Plugins

Sie können auch Ihre eigenen Plugins erstellen.

Eigenständig erstellte Plugins müssen nicht in Agent 2 eingebaut sein. Sie können sie als ladbare Plugins hinzufügen und dadurch die Erstellung von Plugins für neue Metriken vereinfachen.

Ladbare Plugins können zur Laufzeit hinzugefügt werden, ohne Zabbix Agent 2 neu zu kompilieren.

Siehe auch:

- Vergleich von Agent und Agent 2
- Konfigurationsparameter: Zabbix Agent 2
- Konfigurationsparameter: Zabbix Agent 2 (Windows)
- Konfigurationsparameter: Plugins
- Zabbix Agent 2 Plugin-Protokoll

Passive und aktive Prüfungen

Zabbix Agent 2 unterstützt passive und aktive Prüfungen, ähnlich wie der Zabbix Agent. Zusätzlich unterstützen aktive Prüfungen des Zabbix Agent 2 flexible/planbare Intervalle und die Prüfungskonkurrenz innerhalb eines aktiven Servers.

Note:
Standardmäßig plant Zabbix Agent 2 nach einem Neustart die erste Datensammlung für aktive Prüfungen zu einem bedingt zufälligen Zeitpunkt innerhalb des Aktualisierungsintervalls des Datenpunkts, um Spitzen bei der Ressourcennutzung zu vermeiden. Um aktive Prüfungen, die kein *Scheduling*-Aktualisierungsintervall haben, unmittelbar nach dem Neustart des Agents durchzuführen, setzen Sie den Parameter `ForceActiveChecksOnStart` (auf globaler Ebene) oder den `Plugins.<Plugin name>.System.ForceActiveChecksOnStart`-Pluginparameter (betrifft nur Prüfungen bestimmter Plugins) in der Konfigurationsdatei. Der Parameter auf Plugin-Ebene überschreibt, sofern festgelegt, den globalen Parameter.

Prüfkonkurrenz

Prüfungen aus verschiedenen Plugins können gleichzeitig ausgeführt werden. Die Anzahl der gleichzeitigen Prüfungen innerhalb eines Plugins ist durch die Einstellung der Plugin-Kapazität begrenzt. Jedes Plugin kann eine fest codierte Kapazitätseinstellung haben (1000 ist standardmäßig der Wert), die mit der Einstellung `Plugins.<PluginName>.System.Capacity=N` im Konfigurations-Parameter `Plugins` verringert werden kann.

Unterstützte Plattformen Informationen zu unterstützten Plattformen finden Sie auf der Seite Anforderungen.

Agent 2 auf Unix-ähnlichen Systemen Zabbix Agent 2 auf Unix-ähnlichen Systemen wird auf dem überwachten Host ausgeführt.

Installation

Zabbix Agent 2 kann auf Linux-basierten Systemen mit einer der folgenden Methoden installiert werden:

- Zabbix-Pakete - wählen Sie nach Auswahl Ihrer Zabbix-Version, der OS-Distribution und der OS-Version die Komponente Agent 2 aus und folgen Sie den Anweisungen;
- Zabbix-Quellen - laden Sie die Quelldateien herunter und kompilieren Sie den Agent durch Konfigurieren mit der Option `--enable-agent2`.

Note:
Die Überwachungsfunktionen von Zabbix Agent 2 können durch ladbare Plugins erweitert werden, die separat verfügbar sind. Weitere Details finden Sie unter Loadable plugins.

Wenn als Paket installiert

Zabbix Agent 2 läuft als Vordergrundprozess und ist für die Ausführung im Hintergrund auf einen externen Dienstmanager (z. B. systemd) angewiesen. Zabbix Agent 2 verfügt unter Linux nicht über eine integrierte Unterstützung für die Daemonisierung.

Der Agent kann mit folgendem Befehl gestartet werden:

```
systemctl start zabbix-agent2
```

Zum Stoppen, Neustarten oder Prüfen des Status von Zabbix Agent 2 verwenden Sie die folgenden Befehle:

```
systemctl stop zabbix-agent2
systemctl restart zabbix-agent2
systemctl status zabbix-agent2
```

Manuell starten

Sie können den Zabbix Agent starten, indem Sie die `zabbix_agent2`-Binärdatei suchen und direkt ausführen; zum Beispiel:

```
zabbix_agent2
```

Agent 2 auf Windows-Systemen Zabbix Agent 2 wird als eigenständiger Prozess ausgeführt; er kann jedoch auch als Windows-Dienst ausgeführt werden.

Installation

Zabbix Agent 2 kann unter Windows mit einer der folgenden Methoden installiert werden:

- Vorkompilierte Zabbix-Agent-Binärdateien - laden Sie das MSI-Installationspaket des Agenten herunter und folgen Sie den Anweisungen auf der Seite Installation des Windows-Agenten aus MSI;
- Zabbix-Quellcode - laden Sie die Quelldateien herunter und folgen Sie den Anweisungen auf der Seite Zabbix Agent 2 unter Windows kompilieren.

Note:
Die Überwachungsfunktionen von Zabbix Agent 2 können durch ladbare Plugins erweitert werden, die separat verfügbar sind. Weitere Informationen finden Sie unter Ladbare Plugins.

Weitere Details zur Installation von Zabbix Agent 2 (aus einem ZIP-Archiv) als Windows-Dienst finden Sie auf der Seite Zabbix Agent unter Microsoft Windows.

Optionen Die folgenden Befehlszeilenparameter können mit Zabbix Agent 2 verwendet werden:

Parameter	Beschreibung
UNIX- und Windows-Agent	
`-c --config <config-file>`	Pfad zur Konfigurationsdatei. Mit dieser Option können Sie eine andere als die standardmäßig verwendete Konfigurationsdatei angeben.
`-f --foreground`	Zabbix Agent im Vordergrund ausführen (Standard: true).
`-p --print`	Bekannte Datenpunkte ausgeben und beenden. Beachten Sie, dass Sie auch die Ergebnisse von Benutzerparametern zurückgeben können. Dazu müssen Sie die Konfigurationsdatei angeben, wenn sie sich nicht am Standardspeicherort befindet.
`-t --test <item key>`	Den angegebenen Datenpunkt testen und beenden. Beachten Sie, dass Sie auch die Ergebnisse von Benutzerparametern zurückgeben können. Dazu müssen Sie die Konfigurationsdatei angeben, wenn sie sich nicht am Standardspeicherort befindet.
`-T --test-config`	Konfigurationsdatei validieren und beenden.
`-h --help`	Hilfeinformationen ausgeben und beenden.
`-v --verbose`	Debugging-Informationen ausgeben. Verwenden Sie diese Option zusammen mit den Optionen `-p` und `-t`.
`-V --version`	Version und Lizenzinformationen des Agent ausgeben.
`-R --runtime-control <option>`	Administrative Funktionen ausführen. Siehe Laufzeitsteuerung.
Nur Windows-Agent	
`-m --multiple-agents`	Mehrere Agent-Instanzen verwenden (mit den Optionen `-i`, `-d`, `-s`, `-x`). Zur Unterscheidung der Dienstnamen der Instanzen wird jeder Dienstname den Wert Hostname aus der angegebenen Konfigurationsdatei enthalten.
`-S --startup-type <value>`	Starttyp des Zabbix-Windows-Agent-Dienstes festlegen. Zulässige Werte: `automatic` - *(Standard)* den Dienst beim Start von Windows automatisch starten; `delayed` - den Start des Dienstes verzögern, bis die automatisch gestarteten Dienste ihren Start abgeschlossen haben; `manual` - den Dienst manuell durch einen Benutzer oder eine Anwendung starten; `disabled` - den Dienst deaktivieren, sodass er nicht durch einen Benutzer oder eine Anwendung gestartet werden kann. Sie können diese Option zusammen mit der Option `-i` oder separat verwenden, um den Starttyp eines bereits installierten Dienstes zu ändern.
`-i --install`	Zabbix-Windows-Agent als Dienst installieren.
`-d --uninstall`	Zabbix-Windows-Agent-Dienst deinstallieren.
`-s --start`	Zabbix-Windows-Agent-Dienst starten.
`-x --stop`	Zabbix-Windows-Agent-Dienst stoppen.

Spezifische Beispiele für die Verwendung von Befehlszeilenparametern:

- Alle integrierten Agent-Datenpunkte mit ihren Werten ausgeben.
- Einen Benutzerparameter mit dem im angegebenen Konfigurationsdatei definierten Schlüssel `mysql.ping` testen.
- Einen Zabbix-Agent-Dienst für Windows unter Verwendung des Standardpfads zur Konfigurationsdatei installieren.
- Den Starttyp eines installierten Zabbix-Agent-Dienstes für Windows mithilfe der Konfigurationsdatei ändern, die sich im selben Ordner wie die ausführbare Agent-Datei befindet.

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

Laufzeitsteuerung

Die Laufzeitsteuerung bietet einige Optionen zur Fernsteuerung.

Option	Beschreibung
`log_level_increase`	Protokollierungsstufe erhöhen.
`log_level_decrease`	Protokollierungsstufe verringern.
`metrics`	Verfügbare Metriken auflisten.
`periodic_prof_enable`	Periodisches Profiling aktivieren. Alle Metriken erfassen Daten sofort und geben sie aus, mit Ausnahme der CPU-Daten. Periodische CPU-Daten werden über einen bestimmten Zeitraum in einer ausgeblendeten temporären Datei erfasst. Nach Ablauf des Intervalls wird die Datei als CPU-Profil veröffentlicht und die konfigurierte Anzahl abgeschlossener Dateien wird beibehalten. Die temporäre Datei wird nicht auf `ProfilerMaxFilesPerProfile` angerechnet.
`periodic_prof_disable`	Periodisches Profiling deaktivieren.
`periodic_prof_execute`	Profile nach einer 5-sekündigen Stichprobenerfassung schreiben. Wenn das periodische Profiling aktiviert ist, beendet dieser Befehl das aktuelle CPU-Intervall, schreibt ein separates 5-sekündiges CPU-Profil und startet eine neue periodische CPU-Datei. Explizit angeforderte Profile werden beim Abschluss des Befehls gemäß `ProfilerMaxFilesPerProfile` rotiert. Während dieser Befehl Profile erfasst, werden alle Laufzeitbefehle des Profilers mit „profiler is busy“ abgelehnt.
`periodic_prof_set_interval`	Profiling-Intervall festlegen.
`version`	Agent-Version anzeigen.
`userparameter_reload`	Werte der Optionen `UserParameter` und `Include` aus der aktuellen Konfigurationsdatei neu laden.
`help`	Hilfeinformationen zur Laufzeitsteuerung anzeigen.

Beispiele:

- Protokollierungsstufe für Agent 2 erhöhen.
- Optionen der Laufzeitsteuerung anzeigen.

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

Konfigurationsdatei Der Standardpfad zur Konfigurationsdatei hängt vom Betriebssystem und der Installationsmethode von Zabbix ab:

- Unter UNIX lautet der Standardpfad:
  - Bei der Installation aus den Quellen: `/usr/local/etc/zabbix_agent2.conf` oder der Pfad, der durch die Kompilierungsvariablen `--sysconfdir` oder `--prefix` festgelegt wurde.
  - Bei der Installation aus Paketen: `/etc/zabbix/zabbix_agent2.conf`.
- Unter Windows lautet der Standardpfad `C:\Program Files\Zabbix Agent 2\zabbix_agent2.conf`.

Wenn eine einzelne Instanz des Agent auf einem Host ausgeführt wird, kann sie die Standardkonfigurationsdatei oder eine über die Befehlszeile angegebene Konfigurationsdatei verwenden. Bei mehreren Instanzen muss jede Agent-Instanz über eine eigene Konfigurationsdatei verfügen (eine der Instanzen kann die Standardkonfigurationsdatei verwenden).

Die Konfigurationsparameter von Zabbix agent 2 entsprechen größtenteils denen des Zabbix agent, mit einigen Ausnahmen:

Parameter	Beschreibung
Zabbix-agent-2-spezifische Konfigurationsparameter	
`ControlSocket`	Der Pfad zum Control-Socket zur Laufzeit. Zabbix agent 2 verwendet einen Control-Socket für Laufzeitbefehle.

Parameter		Beschreibung
	EnablePersistentBuffer, PersistentBufferFile, PersistentBufferPeriod	Diese Parameter werden verwendet, um den persistenten Speicher für aktive Datenpunkte in Zabbix agent 2 zu konfigurieren.
	ForceActiveChecksOnStart	Legt fest, ob der Agent unmittelbar nach einem Neustart aktive Prüfungen durchführen oder diese gleichmäßig über die Zeit verteilen soll.
	Plugins	Plugins können eigene Parameter haben, und zwar im Format Plugins.<Plugin name>.<Parameter>=<value>. Ein allgemeiner Plugin-Parameter ist System.Capacity, der das Limit für gleichzeitig ausführbare Prüfungen festlegt.
	StatusPort	Der Port, an dem Zabbix agent 2 auf HTTP-Statusanfragen sowie die Anzeige einer Liste der konfigurierten Plugins und einiger interner Parameter wartet.
Zabbix-agent-spezifische Konfigurationsparameter, die von Zabbix agent 2 nicht unterstützt werden		
	AllowRoot, User	Werden von Zabbix agent 2 nicht unterstützt, da dieser keine Daemonisierung unterstützt.
	EnableRemoteCommands	Veraltet; stattdessen werden bei beiden Agents die Parameter AllowKey, AllowKeyRegexp, DenyKey und DenyKeyRegexp verwendet.
	ListenBacklog	Wird im Zabbix agent verwendet, um die maximale Anzahl ausstehender Verbindungen in der TCP-Warteschlange festzulegen. Wird von Zabbix agent 2 nicht unterstützt.
	LoadModule, LoadModulePath	Ladbare Module werden von Zabbix agent 2 nicht unterstützt.
	MaxLinesPerSecond, LogRemoteCommands	In Zabbix agent 2 werden stattdessen Plugins.Log.MaxLinesPerSecond und Plugins.SystemRun.LogRemoteCommands verwendet.
	StartAgents	Dieser Parameter wird im Zabbix agent verwendet, um die Parallelität passiver Prüfungen zu erhöhen oder passive Prüfungen zu deaktivieren. In Zabbix agent 2 wird die Parallelität auf Plugin-Ebene konfiguriert und kann durch eine Kapazitätseinstellung begrenzt werden. Passive Prüfungen sind in Zabbix Agent 2 deaktiviert, wenn der Parameter Server nicht angegeben ist.

Weitere Informationen finden Sie in den Optionen der Konfigurationsdatei für Zabbix agent2.

Exit-Codes Zabbix Agent 2 kann auch mit älteren OpenSSL-Versionen (1.0.1, 1.0.2) kompiliert werden.

In diesem Fall stellt Zabbix Mutexes für das Locking in OpenSSL bereit. Wenn das Sperren oder Entsperren eines Mutex fehlschlägt, wird eine Fehlermeldung in den Standardfehlerstrom (STDERR) ausgegeben, und Agent 2 beendet sich mit dem Rückgabecode 2 bzw. 3.

Agent vs agent 2 Vergleich

Dieser Abschnitt beschreibt die Unterschiede zwischen dem Zabbix Agent und dem Zabbix agent 2.

Parameter	Zabbix agent	Zabbix agent 2
Programmiersprache	C	Go mit einigen Teilen in C
Daemonisierung	ja	nur über systemd (unter Windows ja)
Unterstützte Erweiterungen	Benutzerdefinierte ladbare Module in C.	Benutzerdefinierte Plugins in Go.
Anforderungen		
Unterstützte Plattformen	Linux, IBM AIX, FreeBSD, NetBSD, OpenBSD, HP-UX, Mac OS X, Solaris: 9, 10, 11, Windows: alle Desktop- und Server-Versionen seit XP	Linux, Windows: alle Desktop- und Server-Versionen, auf denen eine unterstützte Go-Version installiert werden kann.
Unterstützte Kryptobiblio-theken	GnuTLS 3.1.18 und neuer. OpenSSL 1.0.1, 1.0.2, 1.1.0, 1.1.1, 3.0.x. LibreSSL - getestet mit den Versionen 2.7.4, 2.8.2 (es gelten bestimmte Einschränkungen, siehe die Seite Verschlüsselung für Details).	Linux: OpenSSL 1.0.1 und neuer. MS Windows: OpenSSL 1.1.1 oder neuer. Die OpenSSL-Bibliothek muss mit aktivierter PSK-Unterstützung kompiliert sein. LibreSSL wird nicht unterstützt.
Überwachungsprozesse		
Prozesse	Ein separater Prozess für aktive Prüfungen für jeden Server/Proxy-Eintrag.	Ein einzelner Prozess mit automatisch erstellten Threads. Die maximale Anzahl der Threads wird durch die Umgebungsvariable `GOMAXPROCS` bestimmt.
Metriken	**UNIX**: siehe eine Liste der unterstützten Datenpunkte. **Windows**: siehe eine Liste zusätzlicher Windows-spezifischer Datenpunkte.	**UNIX**: Alle von Zabbix agent unterstützten Metriken. Zusätzlich bietet der agent 2 eine Zabbix-native Überwachungslösung für: Docker, Memcached, MySQL, PostgreSQL, Redis, systemd und andere Überwachungsziele - siehe eine vollständige Liste der agent 2-spezifischen Datenpunkte. **Windows**: Alle von Zabbix agent unterstützten Metriken sowie `net.tcp.service*`-Prüfungen für HTTPS, LDAP. Zusätzlich bietet der agent 2 eine Zabbix-native Überwachungslösung für: PostgreSQL, Redis.
Parallelität	Aktive Prüfungen für einen einzelnen Server werden nacheinander ausgeführt.	Prüfungen aus verschiedenen Plugins oder mehrere Prüfungen innerhalb eines Plugins können parallel ausgeführt werden.
Traps von Drittanbietern	nein	ja
Zusätzliche Funktionen		
Persistenter Speicher	nein	ja
Persistente Dateien für `log*[]`-Metriken	ja (nur unter Unix)	nein
Hochladen von Logdaten	Kann während der Protokollerfassung durchgeführt werden, um den Puffer freizugeben.	Die Protokollerfassung wird gestoppt, wenn der Puffer voll ist. Daher muss der Parameter `BufferSize` mindestens `MaxLinesPerSecond` x 2 betragen.
Wechselt zur Laufzeit den Benutzer	ja (nur Unix-ähnliche Systeme)	nein (durch systemd gesteuert)
Vom Benutzer konfigurierbare Cipher Suites	ja	nein

Siehe auch:

- *Beschreibung der Zabbix-Prozesse*: Zabbix agent, Zabbix agent 2
- *Konfigurationsparameter*: Zabbix agent UNIX / Windows, Zabbix agent 2 UNIX / Windows

Integrierte Plugins

Diese Plugins sind in Zabbix agent 2 integriert und sofort einsatzbereit. Klicken Sie auf den Plugin-Namen, um zum Plugin-Repository mit zusätzlichen Informationen zu gelangen.

Plugin name	Description	Supported item keys	Comments
Agent	Metriken des verwendeten Zabbix agent.	agent.hostname, agent.ping, agent.version	Unterstützte Schlüssel haben dieselben Parameter wie Zabbix agent keys.
CPU	System-CPU-Überwachung (Anzahl der CPUs/CPU-Kerne, erkannte CPUs, Auslastungsprozentsatz).	system.cpu.discovery, system.cpu.num, system.cpu.util	Unterstützte Schlüssel haben dieselben Parameter wie Zabbix agent keys.
Docker	Überwachung von Docker-Containern.	docker.container_info, docker.container_stats, docker.containers, docker.containers.discovery, docker.data_usage, docker.images, docker.images.discovery, docker.info, docker.ping	Siehe auch: Konfigurationsparameter
File	Erfassung von Dateimetriiken.	vfs.file.cksum, vfs.file.contents, vfs.file.exists, vfs.file.md5sum, vfs.file.regexp, vfs.file.regmatch, vfs.file.size, vfs.file.time	Unterstützte Schlüssel haben dieselben Parameter wie Zabbix agent keys.
Kernel	Kernel-Überwachung.	kernel.maxfiles, kernel.maxproc	Unterstützte Schlüssel haben dieselben Parameter wie Zabbix agent keys.
Log	Überwachung von Protokolldateien.	log, log.count, logrt, logrt.count	Unterstützte Schlüssel haben dieselben Parameter wie Zabbix agent keys. Siehe auch: Plugin-Konfigurationsparameter (Unix/Windows)
Memcached	Überwachung des Memcached-Servers.	memcached.ping, memcached.stats	
Modbus	Liest Modbus-Daten.	modbus.get	Unterstützte Schlüssel haben dieselben Parameter wie Zabbix agent keys.
MQTT	Empfängt veröffentlichte Werte von MQTT-Themen.	mqtt.get	Um eine verschlüsselte Verbindung zum MQTT-Broker zu konfigurieren, geben Sie TLS-Parameter in der Konfigurationsdatei des Agent als benannte Sitzung oder Standard-Parameter an. Derzeit können TLS-Parameter nicht als Parameter für item keys übergeben werden.
MySQL	Überwachung von MySQL und seinen 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	Um eine verschlüsselte Verbindung zur Datenbank zu konfigurieren, geben Sie TLS-Parameter in der Konfigurationsdatei des Agent als benannte Sitzung oder Standard-Parameter an. Derzeit können TLS-Parameter nicht als Parameter für item keys übergeben werden.

Plugin name	Description	Supported item keys	Comments
NetIf	Überwachung von Netzwerkschnittstellen.	net.if.collisions, net.if.discovery, net.if.get, net.if.in, net.if.out, net.if.total	Unterstützte Schlüssel haben dieselben Parameter wie Zabbix agent keys.
Oracle	Überwachung der Oracle Database.	oracle.diskgroups.stats, oracle.diskgroups.discovery, oracle.archive.info, oracle.archive.discovery, oracle.cdb.info, oracle.custom.query, oracle.datafiles.stats, oracle.db.discovery, oracle.fra.stats, oracle.instance.info, oracle.pdb.info, oracle.pdb.discovery, oracle.pga.stats, oracle.ping, oracle.proc.stats, oracle.redolog.info, oracle.sga.stats, oracle.sessions.stats, oracle.sys.metrics, oracle.sys.params, oracle.ts.stats, oracle.ts.discovery, oracle.user.info, oracle.version	Installieren Sie vor der Verwendung des Plugins den Oracle Instant Client.
Proc	CPU-Auslastungsprozentsatz von Prozessen.	proc.cpu.util	Der unterstützte Schlüssel hat dieselben Parameter wie Zabbix agent key.
Redis	Überwachung des Redis-Servers.	redis.config, redis.info, redis.ping, redis.slowlog.count	Um eine verschlüsselte Verbindung zu Redis zu konfigurieren, geben Sie TLS-Parameter in der Konfigurationsdatei des Agent als benannte Sitzung oder Standard-Parameter an. TLS-Parameter können nicht als Parameter für item keys übergeben werden. Beachten Sie, dass eine falsche oder anderweitig ungültige TLS-Konfiguration den Start von Zabbix agent 2 verhindern kann. Überprüfen Sie daher Zertifikatsdateien, Berechtigungen und Pfade, bevor Sie TLS aktivieren.

Smart	S.M.A.R.T.-Überwachung.	smart.attribute.discovery, smart.disk.discovery, smart.disk.get	Die mindestens erforderliche smartctl-Version ist 7.1.
			Für den Benutzer, der Zabbix agent 2 ausführt, sind Sudo-/Root-Zugriffsrechte für smartctl erforderlich. Das Plugin verwendet nur die folgenden Befehle:
			`/usr/sbin/smartctl -a *`
			`/usr/sbin/smartctl --scan *`
			`/usr/sbin/smartctl -j -V`
			
			Unterstützte keys können mit Zabbix agent 2 nur unter Linux/Windows verwendet werden, sowohl als passive als auch als aktive Prüfung.
			Siehe auch: Konfigurationsparameter

Plugin name	Description	Supported item keys	Comments
SW	Auflistung installierter Pakete.	system.sw.packages, system.sw.packages.get	Die unterstützten Schlüssel haben dieselben Parameter wie Zabbix agent key.
Swap	Größe des Swap-Speichers in Bytes/Prozent.	system.swap.size	Der unterstützte Schlüssel hat dieselben Parameter wie Zabbix agent key.
SystemRun	Führt den angegebenen Befehl aus.	system.run	Der unterstützte Schlüssel hat dieselben Parameter wie Zabbix agent key. Siehe auch: Plugin-Konfigurationsparameter (Unix/Windows)
Systemd	Überwachung von systemd-Diensten.	systemd.unit.discovery, systemd.unit.get, systemd.unit.info	
TCP	Prüfung der Verfügbarkeit von TCP-Verbindungen.	net.tcp.port	Der unterstützte Schlüssel hat dieselben Parameter wie Zabbix agent key.
UDP	Überwachung der Verfügbarkeit und Leistung von UDP-Diensten.	net.udp.service, net.udp.service.perf	Unterstützte Schlüssel haben dieselben Parameter wie Zabbix agent keys.
Uname	Abruf von Informationen über das System.	system.hostname, system.sw.arch, system.uname	Unterstützte Schlüssel haben dieselben Parameter wie Zabbix agent keys.
Uptime	Erfassung von System-Uptime-Metriken.	system.uptime	Der unterstützte Schlüssel hat dieselben Parameter wie Zabbix agent key.
VFSDev	Erfassung von VFS-Metriken.	vfs.dev.discovery, vfs.dev.get, vfs.dev.read, vfs.dev.write	Unterstützte Schlüssel haben dieselben Parameter wie Zabbix agent keys.
WebCertificate	Überwachung von TLS/SSL-Website-Zertifikaten.	web.certificate.get	
WebPage	Überwachung von Webseiten.	web.page.get, web.page.perf, web.page.regexp	Unterstützte Schlüssel haben dieselben Parameter wie Zabbix agent keys.
ZabbixAsync	Asynchrone Metrikenerfassung.	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	Unterstützte Schlüssel haben dieselben Parameter wie Zabbix agent keys.
ZabbixStats	Interne Metriken von Zabbix Server/Proxy oder Anzahl verzögerter Datenpunkte in einer Warteschlange.	zabbix.stats	Unterstützte Schlüssel haben dieselben Parameter wie Zabbix agent keys.

Plugin name	Description	Supported item keys	Comments
ZabbixSync	Synchrone Metrikenerfassung.	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.	Unterstützte Schlüssel haben dieselben Parameter wie Zabbix agent keys.

Ladbare Plugins

Ladbare Plugins sind für Zabbix Agent 2 nicht standardmäßig verfügbar und müssen separat installiert werden:

- Unter Linux können Sie Pakete verwenden (z. B. Ember+ unter Ubuntu mit `apt install zabbix-agent2-plugin-ember-plus` installieren) oder Plugins erstellen
- Unter Windows können Sie über MSI installieren oder Plugins erstellen

Attention:
Lesen Sie vor der Installation eines Plugins die zugehörige README-Datei. Sie kann spezifische Anforderungen und Installationsanweisungen enthalten.

Klicken Sie auf den Namen des Plugins, um zum Plugin-Repository zu gelangen. Dort finden Sie die README-Datei des Plugins mit zusätzlichen Informationen.

Plugin name	Description	Supported item keys	Comments
Ceph	Überwachung von Ceph.	ceph.df.details, ceph.osd.stats, ceph.osd.discovery, ceph.osd.dump, ceph.ping, ceph.pool.discovery, ceph.status	Siehe auch Konfigurationsparameter des Ceph-Plugins.
Ember+	Überwachung von Ember+.	ember.get	Siehe auch Konfigurationsparameter des Ember+-Plugins.

Plugin name	Description	Supported item keys	Comments
MongoDB	Überwachung von MongoDB-Servern und -Clustern (dokumentenbasierte, verteilte Datenbank).	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	Um verschlüsselte Verbindungen zur Datenbank zu konfigurieren, geben Sie die TLS-Parameter in der Agent-Konfigurationsdatei als Parameter einer benannten Sitzung an. Derzeit können TLS-Parameter nicht als Parameter des Datenpunktschlüssels übergeben werden. Siehe auch Konfigurationsparameter des MongoDB-Plugins.
MSSQL	Überwachung der MSSQL-Datenbank.	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	Um eine verschlüsselte Verbindung zur Datenbank zu konfigurieren, geben Sie die TLS-Parameter in der Agent-Konfigurationsdatei als Parameter einer benannten Sitzung oder als Standardparameter an. Derzeit können TLS-Parameter nicht als Parameter des Datenpunktschlüssels übergeben werden. Siehe auch Konfigurationsparameter des MSSQL-Plugins.
NVIDIA GPU	Überwachung von NVIDIA-GPUs.	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	Siehe auch Konfigurationsparameter des NVIDIA-GPU-Plugins.

Plugin name	Description	Supported item keys	Comments
PostgreSQL	Überwachung von PostgreSQL und seinen 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	Um verschlüsselte Verbindungen zur Datenbank zu konfigurieren, geben Sie die TLS-Parameter in der Agent-Konfigurationsdatei als Parameter einer benannten Sitzung oder als Standardparameter an. Derzeit können TLS-Parameter nicht als Parameter des Datenpunktschlüssels übergeben werden. Siehe auch Konfigurationsparameter des PostgreSQL-Plugins.

Note:
Ladbare Plugins können mit folgenden Optionen gestartet werden:
 - `-V --version` - Versions- und Lizenzinformationen des Plugins ausgeben;
 - `-h --help` - Hilfeinformationen ausgeben.
 - `-t, --test <item key>` - Plugin zu Testzwecken starten (Plugin-Konfiguration wird ignoriert).

Konfigurationsparameter: Zabbix Agent 2

Übersicht

Konfigurationsparameter ermöglichen die Anpassung des Verhaltens von Zabbix Agent 2.

Zabbix Agent 2 ist eine neue Generation des Zabbix Agent und kann anstelle des Zabbix Agent verwendet werden.

Geben Sie die Parameterwerte in der Konfigurationsdatei von Agent 2 (`zabbix_agent2.conf`) an. Alle Parameter sind **optional**, sofern nicht ausdrücklich angegeben ist, dass der Parameter obligatorisch ist.

Diese Seite ist wie folgt aufgebaut:

- Parameterübersicht (um vollständige Details zu einem Parameter anzuzeigen, klicken Sie auf seinen Namen)
- Parameterdetails

Parameter	Beschreibung
Alias	Legt einen Alias für einen Datenpunktschlüssel fest.
AllowKey	Erlaubt die Ausführung von Datenpunktschlüsseln, die einem Muster entsprechen.
AllowKeyRegexp	Erlaubt die Ausführung von Datenpunktschlüsseln, die einem auf regulären Ausdrücken basierenden Muster entsprechen.
BufferSend	Legt fest, wie häufig Werte aus dem Puffer an den Zabbix Server/Proxy gesendet werden.
BufferSize	Legt die maximale Anzahl von Werten im Speicherpuffer fest.
ControlSocket	Gibt den Control-Socket für das Senden von Laufzeitbefehlen mit der Option „-R“ an.
DebugLevel	Legt die Debug-Stufe fest.
DenyKey	Verhindert die Ausführung von Datenpunktschlüsseln, die einem Muster entsprechen.
DenyKeyRegexp	Verhindert die Ausführung von Datenpunktschlüsseln, die einem auf regulären Ausdrücken basierenden Muster entsprechen.
EnablePersistentBuffer	Aktiviert die Verwendung eines lokalen persistenten Speichers für aktive Datenpunkte.

Parameter	Beschreibung
EnableProfiler	Aktiviert die regelmäßige Go-Profilerstellung beim Start des Agent.
ForceActiveChecksOnStart	Führt aktive Prüfungen unmittelbar nach dem Neustart für die erste empfangene Konfiguration aus.
HeartbeatFrequency	Legt die Frequenz der Heartbeat-Nachrichten in Sekunden fest.
HostInterface	Gibt die Host-Schnittstelle an.
HostInterfaceItem	Gibt einen Datenpunkt zum Abrufen der Host-Schnittstelle an.
HostMetadata	Gibt die Host-Metadaten an.
HostMetadataItem	Gibt einen Zabbix-Agent-Datenpunkt zum Abrufen der Host-Metadaten an.
Hostname	Gibt den Hostnamen an.
HostnameItem	Gibt einen Zabbix-Agent-Datenpunkt zum Abrufen des Hostnamens an.
Include	Gibt ein Verzeichnis oder einzelne Dateien an, die in die Konfigurationsdatei aufgenommen werden sollen.
ListenIP	Gibt die IP-Adresse an, an der der Agent Verbindungen vom Server/Proxy entgegennimmt.
ListenPort	Gibt den Port an, an dem der Agent Verbindungen entgegennimmt.
LogFile	Gibt die Protokolldatei des Agent an.
LogFileSize	Legt die maximale Größe der Protokolldatei fest.
LogType	Gibt den Typ der Protokollausgabe an.
PersistentBufferFile	Gibt die Datei zum Speichern der SQLite-Datenbank an.
PersistentBufferPeriod	Legt den Zeitraum fest, in dem Daten gespeichert werden, wenn keine Verbindung zum Zabbix Server oder Proxy besteht.
PidFile	Gibt die PID-Datei an.
Plugins.<PluginName>.System.Capacity	Legt das Limit der Prüfungen pro Plugin fest, die gleichzeitig ausgeführt werden können.
Plugins.Log.MaxLinesPerSecond	Legt die maximale Anzahl neuer Zeilen pro Sekunde fest, die bei der Verarbeitung aktiver `log`- oder `logrt`-Prüfungen an den Zabbix Server/Proxy gesendet werden.
Plugins.SystemRun.LogRemoteCommands	Aktiviert die Protokollierung der ausgeführten Shell-Befehle als Warnungen.
PluginSocket	Gibt den Pfad zum UNIX-Socket für die Kommunikation mit ladbaren Plugins an.
PluginTimeout	Legt das Timeout für Verbindungen zu ladbaren Plugins fest.
ProfilerDir	Das Verzeichnis, in das Profilerstellungsdaten geschrieben werden.
ProfilerMaxFilesPerProfile	Die maximale Anzahl von Dateien, die für jeden Profiltyp aufbewahrt werden.
ProfilerInterval	Das Schreibintervall für Profilerstellungsdaten.
RefreshActiveChecks	Gibt an, wie häufig die Liste der aktiven Prüfungen aktualisiert wird.
Server	Gibt die IP-Adressen oder DNS-Namen der Zabbix Server/Proxys an.
ServerActive	Gibt die Adresse des Zabbix Server/Proxy oder die Clusterkonfiguration an, von der aktive Prüfungen abgerufen werden.
SourceIP	Gibt die Quell-IP-Adresse an.
StatusPort	Gibt den Port an, an dem HTTP-Statusanfragen entgegengenommen werden (`http://localhost:<port>/status`).
Timeout	Gibt an, wie lange (in Sekunden) auf den Verbindungsaufbau und den Datenaustausch mit dem Zabbix Server oder Proxy gewartet wird.
TLSAccept	Gibt die für eingehende Verbindungen erforderliche Verschlüsselungsstufe an.
TLSCAFile	Gibt den vollständigen Pfadnamen der Datei an, die die Zertifikate der übergeordneten Zertifizierungsstelle(n) zur Überprüfung von Peer-Zertifikaten enthält und für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSCertFile	Gibt den vollständigen Pfadnamen der Datei an, die das Agent-Zertifikat oder die Zertifikatskette enthält und für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSCipherAll	Gibt die OpenSSL-Chiffre-Zeichenfolge (TLS 1.2) an. Überschreibt die Standardkriterien für die Auswahl der Chiffresuite für zertifikats- und PSK-basierte Verschlüsselung.
TLSCipherAll13	Gibt die OpenSSL-Chiffre-Zeichenfolge (TLS 1.3) für OpenSSL 1.1.1 oder neuer an. Überschreibt die Standardkriterien für die Auswahl der Chiffresuite für zertifikats- und PSK-basierte Verschlüsselung.
TLSCipherCert	Gibt die OpenSSL-Chiffre-Zeichenfolge (TLS 1.2) an. Überschreibt die Standardkriterien für die Auswahl der Chiffresuite für zertifikatsbasierte Verschlüsselung.
TLSCipherCert13	Gibt die OpenSSL-Chiffre-Zeichenfolge (TLS 1.3) für OpenSSL 1.1.1 oder neuer an. Überschreibt die Standardkriterien für die Auswahl der Chiffresuite für zertifikatsbasierte Verschlüsselung.
TLSCipherPSK	Gibt die OpenSSL-Chiffre-Zeichenfolge (TLS 1.2) an. Überschreibt die Standardkriterien für die Auswahl der Chiffresuite für PSK-basierte Verschlüsselung.
TLSCipherPSK13	Gibt die OpenSSL-Chiffre-Zeichenfolge (TLS 1.3) für OpenSSL 1.1.1 oder neuer an. Überschreibt die Standardkriterien für die Auswahl der Chiffresuite für PSK-basierte Verschlüsselung.
TLSConnect	Gibt die für Verbindungen zum Zabbix Server oder Proxy erforderliche Verschlüsselungsstufe an.
TLSCRLFile	Gibt den vollständigen Pfadnamen der Datei an, die widerrufene Zertifikate enthält. Dieser Parameter wird für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet.

Parameter	Beschreibung
TLSKeyFile	Gibt den vollständigen Pfadnamen der Datei an, die den privaten Schlüssel des Agent enthält und für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSPSKFile	Gibt den vollständigen Pfadnamen der Datei an, die den Pre-Shared Key des Agent enthält und für die verschlüsselte Kommunikation mit dem Zabbix Server oder Proxy verwendet wird.
TLSPSKIdentity	Gibt die Identitätszeichenfolge des Pre-Shared Key an, die für die verschlüsselte Kommunikation mit dem Zabbix Server oder Proxy verwendet wird.
TLSServerCertIssuer	Gibt den zulässigen Aussteller des Server-(Proxy-)Zertifikats an.
TLSServerCertSubject	Gibt den zulässigen Betreff des Server-(Proxy-)Zertifikats an.
UnsafeUserParameters	Erlaubt die Übergabe aller Zeichen in Argumenten an benutzerdefinierte Parameter.
UserParameter	Gibt einen zu überwachenden benutzerdefinierten Parameter an.
UserParameterDir	Gibt den Standardsuchpfad für UserParameter-Befehle an.

Beachten Sie:

- Die Standardwerte spiegeln die Prozessstandards wider, nicht die Werte in den mitgelieferten Konfigurationsdateien.
- Werte unterstützen Umgebungsvariablen.
- Zabbix unterstützt Konfigurationsdateien nur in der UTF-8-Kodierung ohne BOM.
- Kommentare, die mit # beginnen, werden nur am Zeilenanfang unterstützt.

Parameterdetails

Alias

Legen Sie einen Alias für einen Datenpunktschlüssel fest.
Damit kann ein langer und komplexer Datenpunktschlüssel durch einen kürzeren und einfacheren ersetzt werden.
 Es können mehrere `Alias`-Parameter vorhanden sein.
Mehrere Parameter mit demselben `Alias`-Schlüssel sind nicht zulässig.
 Verschiedene `Alias`-Schlüssel können auf denselben Datenpunktschlüssel verweisen.
 Aliase können in `HostMetadataItem`, jedoch nicht im Parameter `HostnameItem` verwendet werden.

Beispiel 1: Abrufen der ID des Benutzers `zabbix`.

```
Alias=zabbix.userid:vfs.file.regexp[/etc/passwd,"^zabbix:.:([0-9]+)",,,,\1]
```

Nun kann der Kurzschlüssel `zabbix.userid` zum Abrufen von Daten verwendet werden.

Beispiel 2: Ermitteln der CPU-Auslastung mit Standard- und benutzerdefinierten Parametern.

```
Alias=cpu.util:system.cpu.util
Alias=cpu.util[*]:system.cpu.util[*]
```

Damit können Sie den Schlüssel `cpu.util` verwenden, um den Prozentsatz der CPU-Auslastung mit Standardparametern abzurufen, sowie `cpu.util[all, idle, avg15]`, um bestimmte Daten zur CPU-Auslastung abzurufen.

Beispiel 3: Ausführen mehrerer Regeln für die Low-Level-Erkennung, die dieselben Erkennungsdatenpunkte verarbeiten.

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

Nun können mehrere Erkennungsregeln mit `vfs.fs.discovery` eingerichtet werden, wobei für jede Regel unterschiedliche Parameter verwendet werden, z. B. `vfs.fs.discovery[foo]`, `vfs.fs.discovery[bar]` usw.

AllowKey

Erlaubt die Ausführung derjenigen item-Schlüssel, die einem Muster entsprechen. Das Schlüsselmuster ist ein Platzhalterausdruck, der das Zeichen `*` unterstützt, um eine beliebige Anzahl beliebiger Zeichen zu matchen.
Mehrere Schlüssel-Matchregeln können in Kombination mit `DenyKey` definiert werden. Mehrere Schlüssel-Matchregeln können in Kombination mit `DenyKey`, `AllowKeyRegexp` und `DenyKeyRegexp` definiert werden. Die Parameter werden nacheinander entsprechend ihrer Reihenfolge verarbeitet. Die Reihenfolge bestimmt, welche übereinstimmende Regel Vorrang hat. Siehe auch: Einschränken von Agent-Prüfungen.

AllowKeyRegexp

Ermöglicht die Ausführung derjenigen Datenpunktschlüssel, die einem Muster entsprechen.
Das Schlüssel-Muster ist ein regulärer Ausdruck.
 Mehrere Regeln für den Datenpunktschlüssel-Abgleich können in Kombination mit `DenyKeyRegexp`, `AllowKey` und `DenyKey` definiert werden.
Die Parameter werden entsprechend ihrer Reihenfolge nacheinander verarbeitet.
Die Reihenfolge bestimmt, welche übereinstimmende Regel Vorrang hat.
Siehe auch: Einschränken von Agent-Prüfungen.

Beispiel:

```
AllowKeyRegexp=^system\.run\[(?:/usr)?/bin/hostname(?: -[fsd])?\]$
```

BufferSend

Legt die Häufigkeit fest, mit der Werte aus dem Puffer an den Zabbix-Server/Proxy gesendet werden (in Sekunden).
 Beachten Sie, dass die Daten früher gesendet werden, wenn der Puffer voll ist.

Standard: 5
 Bereich: 1-3600

BufferSize

Legt die maximale Anzahl von Werten im Speicherpuffer fest.
 Der Agent sendet alle gesammelten Daten an den Zabbix-Server oder Proxy, wenn der Puffer voll ist. Dieser Parameter sollte nur verwendet werden, wenn der persistente Puffer deaktiviert ist (`EnablePersistentBuffer=0`).

Standard: 1000
 Bereich: 2-65535

ControlSocket

Geben Sie den Control-Socket an, um Laufzeitbefehle mit der Option `-R` zu senden.

Standard: `/tmp/agent.sock`

DebugLevel

Legt den Debug-Level fest: - *0* - grundlegende Informationen zum Starten und Stoppen von Zabbix-Prozessen; - *1* - kritische Informationen; - *2* - Fehlerinformationen; - *3* - Warnungen; - *4* - für das Debugging (liefert viele Informationen); - *5* - erweitertes Debugging (liefert noch mehr Informationen).

Standard: 3
 Bereich: 0-5

DenyKey

Verhindert die Ausführung derjenigen Datenpunktschlüssel, die einem Muster entsprechen.
 Das Schlüsselmuster ist ein Platzhalterausdruck, der das Zeichen * unterstützt, um eine beliebige Anzahl beliebiger Zeichen zu finden.
Mehrere Regeln für den Abgleich von Schlüsseln können in Kombination mit `AllowKey` definiert werden. Mehrere Regeln für den Abgleich von Schlüsseln können in Kombination mit `AllowKey`, `AllowKeyRegexp` und `DenyKeyRegexp` definiert werden. Die Parameter werden entsprechend ihrer Reihenfolge nacheinander verarbeitet. Die Reihenfolge bestimmt, welche übereinstimmende Regel Vorrang hat. Siehe auch: Einschränken von Agent-Prüfungen.

Standard: `DenyKey=system.run[*]`

DenyKeyRegexp

Verhindert die Ausführung derjenigen Datenpunkt-Schlüssel, die einem Muster entsprechen.
Das Schlüssel-Muster ist ein regulärer Ausdruck.
 Mehrere Regeln für den Abgleich von Schlüsseln können in Kombination mit `AllowKeyRegexp`, `AllowKey` und `DenyKey` definiert werden. Die Parameter werden entsprechend ihrer Reihenfolge nacheinander verarbeitet. Die Reihenfolge bestimmt, welche übereinstimmende Regel Vorrang hat.
 Alle `system.run[]`-Datenpunkte werden standardmäßig abgelehnt (`DenyKey=system.run[*]`), sofern keine andere `system.run[*]`-Regel angegeben ist.

Siehe auch: Einschränken von Agent-Prüfungen.

Beispiel:

```
DenyKeyRegexp=^system\.run\[.*(?:rm -rf|poweroff|reboot).*\]$
```

EnablePersistentBuffer

Aktiviert die Verwendung des lokalen persistenten Speichers für aktive Datenpunkte.
 Wenn der persistente Speicher deaktiviert ist, wird der Speicherpuffer verwendet.

Standard: 0
 Werte: 0 - deaktiviert, 1 - aktiviert

EnableProfiler

Aktiviert die regelmäßige Go-Profilerstellung beim Start des Agent.

Die Profilerstellung kann zur Laufzeit auch mit der Option `-R` aktiviert oder deaktiviert werden.
Während `periodic_prof_execute` Profile erfasst, werden alle Laufzeitbefehle des Profilers als ausgelastet abgelehnt.

Standard: 0
 Werte: 0 - deaktiviert, 1 - aktiviert

ForceActiveChecksOnStart

Führe aktive Prüfungen unmittelbar nach dem Neustart für die zuerst empfangene Konfiguration aus.
 Auch als Konfigurationsparameter pro Plugin verfügbar, zum Beispiel: `Plugins.Uptime.System.ForceActiveChecksOnStart=1`

Standard: 0
 Werte: 0 - deaktiviert, 1 - aktiviert

HeartbeatFrequency

Legt die Frequenz der Heartbeat-Nachrichten in Sekunden fest.
 Wird zur Überwachung der Verfügbarkeit aktiver Checks verwendet.
0 - Heartbeat-Nachrichten deaktiviert.

Standard: 60
 Bereich: 0-3600

HostInterface

Geben Sie die Host-Schnittstelle (IP-Adresse oder DNS-Name) an, die während der Autoregistrierung des Host verwendet wird.
 Dieser Wert wird verwendet, um die Schnittstelle im neu erstellten Host zu befüllen, und ermöglicht die explizite Konfiguration entweder einer IP- oder DNS-Adresse.
 Weitere Informationen finden Sie unter Using DNS as default interface.

Wenn nicht definiert, wird der Wert aus `HostInterfaceItem` übernommen.

Der Agent gibt einen Fehler aus und startet nicht, wenn der Wert die Grenze von 255 Zeichen überschreitet.

Bereich: 0-255 Zeichen

HostInterfaceItem

Geben Sie einen Datenpunkt an, um die Host-Schnittstelle (IP-Adresse oder DNS-Name) während der Autoregistrierung des Hosts zu bestimmen.
 Dieser Wert wird nur verwendet, wenn `HostInterface` nicht definiert ist.
 Weitere Informationen finden Sie unter DNS als Standardschnittstelle verwenden.

Während einer Autoregistrierungsanfrage protokolliert der Agent eine Warnmeldung, wenn der vom angegebenen Datenpunkt zurückgegebene Wert das Limit von 255 Zeichen überschreitet.

Der Datenpunkt `system.run[]` wird unabhängig von den Einstellungen `AllowKey`, `AllowKeyRegexp`, `DenyKey` oder `DenyKeyRegexp` unterstützt.

HostMetadata

Geben Sie die Metadaten an, die verwendet werden, um den Host während der Autoregistrierung zu identifizieren oder zu unterscheiden.
 `HostMetadata` ermöglicht die Unterscheidung zwischen Hosts über den Hostnamen hinaus.

Wenn nicht definiert, wird der Wert aus `HostMetadataItem` abgerufen.

Der Agent gibt einen Fehler aus und wird nicht gestartet, wenn der angegebene Wert das Limit von 2034 Byte überschreitet oder keine UTF-8-Zeichenfolge ist. Wenn ein Parameter eine IP-Adresse oder einen DNS-Namen erwartet, werden Werte, die zwar gültiges UTF-8 enthalten, aber keine gültigen IP-Adressen oder DNS-Namen sind, ebenfalls abgelehnt und als ungültig gemeldet.

Mehrzeilige Metadaten werden nicht unterstützt – die Ausgabe wird beim ersten Zeilenumbruch abgeschnitten.

Bereich: 0–2034 Byte

HostMetadataItem

Geben Sie einen Datenpunkt an, aus dem Host-Metadaten abgerufen werden.
 Diese Option wird nur verwendet, wenn `HostMetadata` nicht definiert ist.

Der Wert von `HostMetadataItem` wird bei jedem Versuch der Autoregistrierung abgerufen und nur während des Autoregistrierungsprozesses des Hosts verwendet. `HostMetadataItem` ermöglicht die Unterscheidung zwischen Hosts über den Hostnamen hinaus.

Benutzerparameter und Aliase werden unterstützt. Der Datenpunkt `system.run[]` wird unabhängig von den Einstellungen `AllowKey`, `AllowKeyRegexp`, `DenyKey` oder `DenyKeyRegexp` unterstützt.

Während einer Autoregistrierungsanfrage protokolliert der Agent eine Warnung, wenn der vom angegebenen Datenpunkt zurückgegebene Wert das Limit von 65535 UTF-8-Codepunkten überschreitet. Der vom Datenpunkt zurückgegebene Wert muss eine UTF-8-Zeichenfolge sein; andernfalls wird er ignoriert. Wenn ein Parameter eine IP-Adresse oder einen DNS-Namen erwartet, werden Werte, die zwar gültiges UTF-8 enthalten, aber keine gültigen IP-Adressen oder DNS-Namen sind, ebenfalls abgelehnt und als ungültig gemeldet.

Mehrzeilige Metadaten werden nicht unterstützt – die Ausgabe wird beim ersten Zeilenumbruch abgeschnitten.

Hostname

Geben Sie eindeutige, groß-/kleinschreibungsabhängige Hostnamen an. Eine durch Kommas getrennte Liste.
 Erforderlich für aktive Prüfungen und muss mit den auf dem Server konfigurierten Hostnamen übereinstimmen.
 Der Wert wird aus

HostnameItem übernommen, wenn er nicht definiert ist.
Zulässige Zeichen: alphanumerisch, '.', ' ', '_' und '-'. Maximale Länge: 128 Zeichen pro Hostname, 2048 Zeichen für die gesamte Zeile.

Standard: Festgelegt durch HostnameItem

HostnameItem

Geben Sie einen Datenpunkt zum Abrufen des Hostnamens an.
 Diese Option wird nur verwendet, wenn Hostname nicht definiert ist.
 Benutzerparameter oder Aliase werden nicht unterstützt, aber der Datenpunkt system.run[] wird unabhängig von den Werten AllowKey, AllowKeyRegexp, DenyKey oder DenyKeyRegexp unterstützt.

Standard: system.hostname

Include

Geben Sie ein Verzeichnis oder einzelne Dateien an, die in die Konfigurationsdatei aufgenommen werden sollen.
 Während der Zabbix-Installation wird dieser Parameter auf /usr/local/etc gesetzt, sofern Sie dies nicht zur Kompilierzeit ändern.

Allgemeine Regeln für den Parameter Include:

- Der Pfad kann relativ zum Speicherort der Datei zabbix_agent2.conf angegeben werden.
- Sie können mehrere Include-Parameter in einer Konfigurationsdatei angeben.
- Zabbix verarbeitet den Include-Parameter, sobald er gelesen wird. Platzieren Sie ihn daher am besten am Ende der Konfigurationsdatei.

Regeln für eingebundene Dateien:

- Jede Datei muss lesbar sein.
- Jede Datei muss eine korrekte Syntax haben. Andernfalls wird die Komponente nicht gestartet.

Regeln für das Einbinden von Verzeichnissen oder Dateimustern:

- Gehen Sie nicht von einer bestimmten Reihenfolge beim Einbinden aus (z. B. werden Dateien nicht alphabetisch eingebunden).
- Definieren Sie denselben Parameter nicht in mehr als einer eingebundenen Datei (z. B. um eine allgemeine Einstellung durch eine spezifische zu überschreiben), da Sie nicht vorhersagen können, welcher Wert verwendet wird.
- Achten Sie auf Sicherungskopien, die einige Texteditoren automatisch erstellen (z. B. kann beim Bearbeiten von include/my_specific.conf die Datei include/my_specific.conf.BAK erstellt werden), da beide Dateien eingebunden werden.

Um nur relevante Dateien aus einem Verzeichnis einzubinden, verwenden Sie den Platzhalter (*) für die Mustererkennung.

Beispiel:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenIP

Geben Sie die IP-Adresse an, auf der der Agent Verbindungen vom Server/Proxy entgegennimmt. Eine durch Kommas getrennte Liste.
 Die erste IP-Adresse wird an den Zabbix-Server gesendet, wenn eine Verbindung zu ihm hergestellt wird, um die Liste der aktiven Prüfungen abzurufen.

Standard: 0.0.0.0

ListenPort

Geben Sie den Listen-Port des Agent für Verbindungen vom Server/Proxy an.

Standard: 10050
 Bereich: 1024-32767

LogFile

Geben Sie die Logdatei des Agent an.

Standard: /tmp/zabbix_agent2.log
 Erforderlich: Ja, wenn LogType auf file gesetzt ist; andernfalls nein

LogFileSize

Setzen Sie die maximale Größe der Logdatei (in MB).
0 - automatische Logrotation deaktivieren.
Hinweis: Wenn die Größenbegrenzung der Logdatei erreicht wird und die Dateirotation aus irgendeinem Grund fehlschlägt, wird die vorhandene Logdatei gekürzt und neu begonnen.

Standard: 1
 Bereich: 0-1024

LogType

Geben Sie den Typ der Protokollausgabe an: - *file* - schreibt das Protokoll in die durch den Parameter LogFile angegebene Datei; - *system* - schreibt das Protokoll in syslog; - *console* - schreibt das Protokoll in die Standardausgabe.

Standardwert: file

PersistentBufferFile

Geben Sie die Datei an, in der die SQLite-Datenbank gespeichert wird. Es muss ein vollständiger Dateiname sein.
 Dieser Parameter wird nur verwendet, wenn der persistente Puffer aktiviert ist (EnablePersistentBuffer=1).

PersistentBufferPeriod

Legt den Zeitraum fest, für den Daten gespeichert werden, wenn keine Verbindung zum Zabbix Server oder Proxy besteht. Ältere Daten gehen verloren.
 Logdaten werden beibehalten. Dieser Parameter wird nur verwendet, wenn der persistente Puffer aktiviert ist (EnablePersistentBuffer=1).

Standard: 1h
 Bereich: 1m-365d

PidFile

Geben Sie die PID-Datei an.

Standard: /tmp/zabbix_agent2.pid

Plugins.<PluginName>.System.Capacity

Legt das Limit der gleichzeitig ausführbaren Prüfungen pro <PluginName>-Plugin fest.

Standard: 1000 Bereich: 1-1000

Plugins.Log.MaxLinesPerSecond

Legt die maximale Anzahl neuer Zeilen pro Sekunde fest, die beim Verarbeiten von aktiven log- oder logrt-Prüfungen an den Zabbix Server/Proxy gesendet werden.
 Der angegebene Wert wird durch den Parameter maxlines überschrieben, der im log- und logrt-Datenpunkt-Schlüssel angegeben ist.
Hinweis: Zabbix verarbeitet 10-mal mehr neue Zeilen als in MaxLinesPerSecond festgelegt, um in Protokolleinträgen nach der erforderlichen Zeichenfolge zu suchen.

Standard: 20
 Bereich: 1-1000

Plugins.SystemRun.LogRemoteCommands

Aktiviert die Protokollierung der ausgeführten Shell-Befehle als Warnungen.
 Die Befehle werden nur protokolliert, wenn sie remote ausgeführt werden.
 Protokolleinträge werden nicht erstellt, wenn system.run[] lokal über die Parameter HostMetadataItem, HostInterfaceItem oder HostnameItem gestartet wird.

Standard: 0
 Werte: 0 - deaktiviert, 1 - aktiviert

PluginSocket

Geben Sie den Pfad zum UNIX-Socket für die Kommunikation mit ladbaren Plugins an.

Standard: /tmp/agent.plugin.sock

PluginTimeout

Geben Sie das Timeout für Verbindungen zu ladbaren Plugins in Sekunden an.

Standard: Timeout
 Bereich: 1-30

ProfilerDir

Das Verzeichnis, in das Profildaten geschrieben werden.

Standard: /tmp/zabbix_agent2_profiles

ProfilerMaxFilesPerProfile

Die maximale Anzahl der Dateien, die für jeden Profiltyp aufbewahrt werden sollen.
Das Limit wird nach jedem Profildump auf abgeschlossene Dateien angewendet.
Das aktive CPU-Profil wird in einer ausgeblendeten temporären Datei erfasst und nicht auf das Limit angerechnet.
Dateien, die von periodic_prof_execute erstellt werden, unterliegen ebenfalls dem Limit.

Standard: 10
 Bereich: 1-100

ProfilerInterval

Das Schreibintervall für Profildaten in Sekunden.

Standard: 3600
 Bereich: 1-86400

RefreshActiveChecks

Geben Sie die Häufigkeit der Aktualisierung der Liste der aktiven Prüfungen in Sekunden an.
 Beachten Sie, dass nach einem fehlgeschlagenen Versuch, die aktiven Prüfungen zu aktualisieren, der nächste Aktualisierungsversuch nach 60 Sekunden erfolgt.

Standard: 5
 Bereich: 1-86400

Server

Geben Sie die IP-Adressen, optional in CIDR-Notation, oder DNS-Namen von Zabbix-Servern/Proxys an. Eine durch Kommas getrennte Liste.
 Eingehende Verbindungen werden nur von den hier aufgeführten Hosts akzeptiert. Wenn die IPv6-Unterstützung aktiviert ist, werden `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` als gleich behandelt, und `::/0` erlaubt jede IPv4- oder IPv6-Adresse. `0.0.0.0/0` kann verwendet werden, um jede IPv4-Adresse zuzulassen. Leerzeichen sind erlaubt. Wenn dieser Parameter nicht angegeben wird, werden passive Prüfungen deaktiviert, und der Agent lauscht auf keinem TCP-Port.

Beispiel:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

ServerActive

Geben Sie die Adresse des Zabbix Server/Proxy oder die Clusterkonfiguration an, von der aktive Prüfungen abgerufen werden sollen.
 Die Adresse des Server/Proxy ist eine IP-Adresse oder ein DNS-Name mit optionalem, durch einen Doppelpunkt getrenntem Port.
Die Clusterkonfiguration ist eine oder mehrere Adressen von Mitgliedern einer Server- oder Proxygruppe, getrennt durch Semikolon. Es können mehrere Zabbix Server/Cluster und Zabbix Proxies angegeben werden, getrennt durch Komma. Sofern keine Proxygruppen verwendet werden, sollte von jedem Zabbix Server/Cluster nicht mehr als ein Zabbix Proxy angegeben werden. Wenn ein Zabbix Proxy angegeben ist, darf der Zabbix Server/Cluster für diesen Proxy nicht angegeben werden.
Mehrere durch Kommas getrennte Adressen können angegeben werden, um mehrere unabhängige Zabbix Server parallel zu verwenden. Leerzeichen sind zulässig.
Wenn der Port nicht angegeben ist, wird der Standardport verwendet.
IPv6-Adressen müssen in eckige Klammern gesetzt werden, wenn für diesen Host ein Port angegeben ist. Wenn kein Port angegeben ist, sind eckige Klammern für IPv6-Adressen optional.
Wenn dieser Parameter nicht angegeben ist, sind aktive Prüfungen deaktiviert.

Beispiel für Zabbix Proxy:

```
ServerActive=127.0.0.1:10051
```

Beispiel für eine Zabbix Proxygruppe:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

Beispiel für mehrere Server:

```
ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
```

Beispiel für Hochverfügbarkeit:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

Beispiel für Hochverfügbarkeit mit zwei Clustern und einem Server:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

SourceIP

Geben Sie die Quell-IP-Adresse an für:

- ausgehende Verbindungen zum Zabbix Server oder Zabbix Proxy.
- Verbindungen, die beim Ausführen einiger Datenpunkte (`web.page.get`, `net.tcp.port` usw.) hergestellt werden.

StatusPort

Geben Sie den Port für das Lauschen auf HTTP-Statusanfragen an (`http://localhost:<port>/status`).

Bereich: 1024-32767

Timeout

Geben Sie an, wie lange beim Aufbau der Verbindung und beim Datenaustausch mit dem Zabbix Proxy oder Server gewartet werden soll (in Sekunden).

Dieser Parameter legt die Dauer verschiedener Kommunikationsvorgänge fest, darunter:

- das Warten auf eine Antwort vom Zabbix Server;
- das Senden von Anfragen an den Zabbix Server, einschließlich Anfragen zur Datenpunktkonfiguration und Datenpunktdaten in aktiven Prüfungen;
- das Abrufen von Protokolldaten über logfile;

- das Senden von Heartbeat-Nachrichten;
- die maximale Dauer für `vfs.*`-Prüfungen;
- die Verwendung als Fallback in Szenarien, in denen ein Server oder Proxy älter als Version 7.0 Prüfungen ohne Timeouts sendet.

Dieses Timeout wird **nicht** für jene Agent-Prüfungen verwendet, die in der Frontend konfigurierbare Timeout-Einstellungen haben (auf globaler Ebene, auf Proxy-Ebene oder auf Datenpunktebene).

Standard: 3
 Bereich: 1-30

TLSAccept

Geben Sie die für eingehende Verbindungen erforderliche Verschlüsselungsstufe an.
 Wird für passive Prüfungen verwendet.
 Mehrere Werte können durch Kommas getrennt angegeben werden: - *unencrypted* - Verbindungen ohne Verschlüsselung akzeptieren (Standard); - *psk* - Verbindungen mit TLS und einem vorab gemeinsam genutzten Schlüssel (PSK) akzeptieren; - *cert* - Verbindungen mit TLS und einem Zertifikat akzeptieren.

Pflicht: ja, wenn TLS-Zertifikat- oder PSK-Parameter definiert sind (auch für eine *unencrypted* Verbindung); andernfalls nein

TLSCAFile

Geben Sie den vollständigen Pfadnamen der Datei an, die die Zertifikate der obersten CA(s) für die Überprüfung von Peer-Zertifikaten enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

TLSCertFile

Geben Sie den vollständigen Pfadnamen der Datei an, die das Agent-Zertifikat oder die Zertifikatskette enthält und für verschlüsselte Kommunikation mit Zabbix-Komponenten verwendet wird.

TLSCipherAll

Geben Sie die OpenSSL-Cipher-String (TLS 1.2) an. Überschreiben Sie die standardmäßigen Auswahlkriterien für die Ciphersuite für zertifikats- und PSK-basierte Verschlüsselung.

Beispiel:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherAll13

Geben Sie den OpenSSL-Cipher-String (TLS 1.3) für OpenSSL 1.1.1 oder neuer an. Überschreiben Sie die standardmäßigen Auswahlkriterien für die Ciphersuite für zertifikats- und PSK-basierte Verschlüsselung.

Beispiel:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherCert

Geben Sie die OpenSSL-Cipher-String an (TLS 1.2). Überschreiben Sie die standardmäßigen Auswahlkriterien für die Ciphersuite bei zertifikatsbasierter Verschlüsselung.

Beispiel:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

Beachten Sie, dass dieser Parameter nicht zusammen mit `TLSAccept=cert,psk` verwendet werden kann; für eine Zertifikatsverbindung (`TLSConnect=cert`) verwenden Sie stattdessen `TLSCipherAll`.

TLSCipherCert13

Geben Sie den OpenSSL-Cipher-String (TLS 1.3) für OpenSSL 1.1.1 oder neuer an. Überschreiben Sie die standardmäßigen Auswahlkriterien für die Cipher Suite bei zertifikatsbasierter Verschlüsselung.

Beachten Sie, dass dieser Parameter nicht zusammen mit `TLSAccept=cert,psk` verwendet werden kann; für eine Zertifikatsverbindung (`TLSConnect=cert`) verwenden Sie stattdessen `TLSCipherAll13`.

TLSCipherPSK

Geben Sie die OpenSSL-Cipher-String (TLS 1.2) an. Überschreiben Sie die standardmäßigen Auswahlkriterien für die Ciphersuite für PSK-basierte Verschlüsselung.

Beispiel:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

Geben Sie den OpenSSL-Cipher-String (TLS 1.3) für OpenSSL 1.1.1 oder neuer an. Überschreiben Sie die standardmäßigen Auswahlkriterien für die Cipher Suite bei PSK-basierter Verschlüsselung.

Beispiel:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Geben Sie den Verschlüsselungsgrad an, der für Agent-Verbindungen zum Zabbix Server oder Proxy erforderlich ist.
 Wird für aktive Prüfungen verwendet.
 Es kann nur ein Wert angegeben werden: - *unencrypted* - Verbindung ohne Verschlüsselung herstellen (Standard); - *psk* - Verbindung mit TLS und einem vorab gemeinsam genutzten Schlüssel (PSK) herstellen; - *cert* - Verbindung mit TLS und einem Zertifikat herstellen.

Pflicht: ja, wenn TLS-Zertifikat- oder PSK-Parameter definiert sind (auch bei *unencrypted*-Verbindung); andernfalls nein

TLSCRLFile

Geben Sie den vollständigen Pfadnamen der Datei an, die widerrufene Zertifikate enthält. Dieser Parameter wird für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet.

TLSKeyFile

Geben Sie den vollständigen Pfadnamen der Datei an, die den privaten Schlüssel des Agent enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

TLSPSKFile

Geben Sie den vollständigen Pfadnamen der Datei an, die den vorab gemeinsam genutzten Schlüssel des Agent enthält und für verschlüsselte Kommunikation mit dem Zabbix-Server oder Proxy verwendet wird.

TLSPSKIdentity

Geben Sie die Identitätszeichenfolge des Pre-Shared Keys an, die für verschlüsselte Verbindungen mit dem Zabbix Server oder Proxy verwendet wird.

TLSServerCertIssuer

Geben Sie den zulässigen Aussteller des Server- (Proxy-) Zertifikats an.

TLSServerCertSubject

Geben Sie den zulässigen Zertifikatsbetreff des Servers (Proxy) an.

UnsafeUserParameters

Erlaubt, dass alle Zeichen in Argumenten an benutzerdefinierte Parameter übergeben werden. Die folgenden Zeichen sind nicht erlaubt: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
Zusätzlich sind Zeilenumbrüche nicht erlaubt.

Standard: 0
 Werte: 0 - nicht erlauben, 1 - erlauben

UserParameter

Geben Sie einen benutzerdefinierten Parameter an, der überwacht werden soll. Es können mehrere benutzerdefinierte Parameter vorhanden sein.
Format: `UserParameter=<key>,<shell command>`
Beachten Sie, dass der Shell-Befehl keinen leeren String oder nur EOL zurückgeben darf. Shell-Befehle dürfen relative Pfade verwenden, wenn der Parameter `UserParameterDir` angegeben ist.

Beispiel:

```
UserParameter=system.test,who|wc -l
UserParameter=check_cpu,./custom_script.sh
```

UserParameterDir

Geben Sie den standardmäßigen Suchpfad für `UserParameter`-Befehle an. Wenn diese Option verwendet wird, wechselt der Agent vor der Ausführung eines Befehls in das hier angegebene Arbeitsverzeichnis. Dadurch können UserParameter-Befehle ein relatives Präfix `./` anstelle eines vollständigen Pfads verwenden.
Es ist nur ein Eintrag zulässig.

Beispiel:

```
UserParameterDir=/opt/myscripts
```

Konfigurationsparameter: Zabbix Agent 2 (Windows)

Übersicht

Konfigurationsparameter ermöglichen die Anpassung des Verhaltens von Zabbix Agent 2 unter Windows.

Geben Sie die Parameterwerte in der Konfigurationsdatei von Agent 2 (`zabbix_agent2.conf`) an. Alle Parameter sind **optional**, sofern nicht ausdrücklich angegeben ist, dass der Parameter obligatorisch ist.

Diese Seite ist wie folgt aufgebaut:

- Parameterübersicht (um die vollständigen Details eines Parameters anzuzeigen, klicken Sie auf seinen Namen)
- Parameterdetails

Parameter	Beschreibung
Alias	Legt einen Alias für einen Datenpunktschlüssel fest.
AllowKey	Erlaubt die Ausführung der Datenpunktschlüssel, die einem Muster entsprechen.
AllowKeyRegexp	Erlaubt die Ausführung der Datenpunktschlüssel, die einem auf regulären Ausdrücken basierenden Muster entsprechen.
BufferSend	Legt fest, wie häufig Werte aus dem Puffer an den Zabbix Server gesendet werden.
BufferSize	Legt die maximale Anzahl von Werten im Speicherpuffer fest.
ControlSocket	Gibt den Control-Socket an, der zum Senden von Laufzeitbefehlen mit der Option `-R` verwendet wird.
DebugLevel	Legt die Debug-Stufe fest.
DenyKey	Verhindert die Ausführung der Datenpunktschlüssel, die einem Muster entsprechen.
DenyKeyRegexp	Verhindert die Ausführung der Datenpunktschlüssel, die einem auf regulären Ausdrücken basierenden Muster entsprechen.
EnablePersistentBuffer	Aktiviert die Verwendung eines lokalen persistenten Speichers für aktive Datenpunkte.
EnableProfiler	Aktiviert die regelmäßige Go-Profilerstellung beim Start von Agent.
ForceActiveChecksOnStart	Führt aktive Prüfungen unmittelbar nach dem Neustart für die erste empfangene Konfiguration aus.
HeartbeatFrequency	Legt die Häufigkeit von Heartbeat-Nachrichten in Sekunden fest.
HostInterface	Gibt die Host-Schnittstelle an.
HostInterfaceItem	Gibt einen Datenpunkt zum Abrufen der Host-Schnittstelle an.
HostMetadata	Gibt die Host-Metadaten an.
HostMetadataItem	Gibt einen Zabbix-Agent-Datenpunkt zum Abrufen der Host-Metadaten an.
Hostname	Gibt den Hostnamen an.
HostnameItem	Gibt einen Zabbix-Agent-Datenpunkt zum Abrufen des Hostnamens an.
Include	Gibt ein Verzeichnis oder einzelne Dateien an, die in die Konfigurationsdatei aufgenommen werden sollen.
ListenIP	Gibt die IP-Adresse an, unter der Agent Verbindungen vom Server/Proxy entgegennimmt.
ListenPort	Gibt den Port an, unter dem Agent Verbindungen entgegennimmt.
LogFile	Gibt die Protokolldatei von Agent an.
LogFileSize	Legt die maximale Größe der Protokolldatei fest.
LogType	Gibt den Typ der Protokollausgabe an.
PerfCounter	Gibt einen neuen Parameter <parameter_name> an, der den Durchschnittswert des Systemleistungsindikators <perf_counter_path> für den angegebenen Zeitraum <period> (in Sekunden) darstellt.
PerfCounterEn	Gibt einen neuen Parameter <parameter_name> an, der den Durchschnittswert des Systemleistungsindikators <perf_counter_path> für den angegebenen Zeitraum <period> (in Sekunden) darstellt. Im Gegensatz zu PerfCounter müssen die Pfade der Leistungsindikatoren auf Englisch angegeben werden.
PersistentBufferFile	Gibt die Datei an, in der Zabbix Agent 2 die SQLite-Datenbank speichern soll.
PersistentBufferPeriod	Legt fest, wie lange Daten gespeichert werden, wenn keine Verbindung zum Server oder Proxy besteht.
Plugins.<PluginName>.System.Capacity	Legt die Anzahl der Prüfungen pro Plugin fest, die gleichzeitig ausgeführt werden können.
Plugins.Log.MaxLinesPerSecond	Legt die maximale Anzahl neuer Zeilen fest, die Agent bei der Verarbeitung aktiver `log`-, `logrt`- und `eventlog`-Prüfungen pro Sekunde an den Zabbix Server oder Proxy sendet.
Plugins.SystemRun.LogRemoteCommands	Aktiviert die Protokollierung der ausgeführten Shell-Befehle als Warnungen.
PluginSocket	Gibt den Pfad zur benannten Windows-Pipe für die Kommunikation mit ladbaren Plugins an.
PluginTimeout	Legt das Timeout für Verbindungen mit ladbaren Plugins in Sekunden fest.
ProfilerDir	Das Verzeichnis, in das Profilerstellungsdaten geschrieben werden.
ProfilerMaxFilesPerProfile	Die maximale Anzahl von Dateien, die für jeden Profiltyp aufbewahrt werden.
ProfilerInterval	Das Schreibintervall für Profilerstellungsdaten.

Parameter	Beschreibung
RefreshActiveChecks	Gibt an, wie häufig die Liste der aktiven Prüfungen aktualisiert wird.
Server	Gibt die IP-Adressen oder DNS-Namen der Zabbix Server/Proxys an.
ServerActive	Gibt die Adresse des Zabbix Servers/Proxys oder die Clusterkonfiguration an, von der aktive Prüfungen abgerufen werden.
SourceIP	Gibt die Quell-IP-Adresse an.
StatusPort	Legt den Port fest, unter dem Agent HTTP-Statusanfragen entgegennimmt (`http://localhost:<port>/status`).
Timeout	Gibt an, wie lange (in Sekunden) auf den Verbindungsaufbau und den Datenaustausch mit dem Zabbix Server oder Proxy gewartet wird.
TLSAccept	Gibt die für eingehende Verbindungen erforderliche Verschlüsselungsstufe an.
TLSCAFile	Gibt den vollständigen Pfadnamen der Datei an, die die Zertifikate der übergeordneten Zertifizierungsstelle(n) zur Überprüfung von Peer-Zertifikaten enthält und für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSCertFile	Gibt den vollständigen Pfadnamen der Datei an, die das Agent-Zertifikat oder die Zertifikatskette enthält und für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSCipherAll13	Gibt die OpenSSL-Chiffre-Zeichenfolge (TLS 1.3) an. Überschreibt die Standardkriterien für die Auswahl der Chiffresuite für zertifikats- und PSK-basierte Verschlüsselung.
TLSCipherCert13	Gibt die OpenSSL-Chiffre-Zeichenfolge (TLS 1.3) an. Überschreibt die Standardkriterien für die Auswahl der Chiffresuite für zertifikatsbasierte Verschlüsselung.
TLSCipherPSK13	Gibt die OpenSSL-Chiffre-Zeichenfolge (TLS 1.3) an. Überschreibt die Standardkriterien für die Auswahl der Chiffresuite für PSK-basierte Verschlüsselung.
TLSConnect	Gibt die für Verbindungen zum Zabbix Server oder Proxy erforderliche Verschlüsselungsstufe an.
TLSCRLFile	Gibt den vollständigen Pfadnamen der Datei an, die widerrufene Zertifikate enthält. Dieser Parameter wird für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet.
TLSKeyFile	Gibt den vollständigen Pfadnamen der Datei an, die den privaten Schlüssel von Agent enthält und für die verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.
TLSPSKFile	Gibt den vollständigen Pfadnamen der Datei an, die den Pre-Shared Key von Agent enthält und für die verschlüsselte Kommunikation mit dem Zabbix Server oder Proxy verwendet wird.
TLSPSKIdentity	Gibt die Identitätszeichenfolge des Pre-Shared Keys an, die für die verschlüsselte Kommunikation mit dem Zabbix Server oder Proxy verwendet wird.
TLSServerCertIssuer	Gibt den zulässigen Aussteller des Server-(Proxy-)Zertifikats an.
TLSServerCertSubject	Gibt den zulässigen Betreff des Server-(Proxy-)Zertifikats an.
UnsafeUserParameters	Erlaubt die Übergabe aller Zeichen in Argumenten an benutzerdefinierte Parameter.
UserParameter	Gibt einen zu überwachenden benutzerdefinierten Parameter an.
UserParameterDir	Gibt den Standardsuchpfad für `UserParameter`-Befehle an.

Beachten Sie Folgendes:

- Die Standardwerte spiegeln die Prozessstandards wider, nicht die Werte in den mitgelieferten Konfigurationsdateien.
- Werte unterstützen Umgebungsvariablen.
- Zabbix unterstützt Konfigurationsdateien nur in der UTF-8-Kodierung ohne BOM.
- Kommentare, die mit # beginnen, werden nur am Anfang der Zeile unterstützt.

Parameterdetails

Alias

Legen Sie einen Alias für einen Datenpunktschlüssel fest.
Damit kann ein langer und komplexer Datenpunktschlüssel durch einen kürzeren und einfacheren ersetzt werden.
 Es können mehrere `Alias`-Parameter vorhanden sein.
Mehrere Parameter mit demselben `Alias`-Schlüssel sind nicht zulässig.
 Verschiedene `Alias`-Schlüssel können auf denselben Datenpunktschlüssel verweisen.
 Aliase können in `HostMetadataItem`, jedoch nicht im Parameter `HostnameItem` verwendet werden.

Beispiel 1: Abrufen der Auslagerungsdateinutzung in Prozent vom Server.

```
Alias=pg_usage:perf_counter[\Paging File(_Total)\% Usage]
```

Nun kann der Kurzschlüssel `pg_usage` zum Abrufen von Daten verwendet werden.

Beispiel 2: Abrufen der CPU-Auslastung mit Standard- und benutzerdefinierten Parametern.

```
Alias=cpu.load:system.cpu.load
Alias=cpu.load[*]:system.cpu.load[*]
```

Damit können Sie den Schlüssel `cpu.load` verwenden, um die CPU-Auslastung mit Standardparametern abzurufen, sowie `cpu.load[percpu,avg15]`, um bestimmte Daten zur CPU-Auslastung abzurufen.

Beispiel 3: Ausführen mehrerer Low-Level-Discovery-Regeln, die dieselben Discovery-Datenpunkte verarbeiten.

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

Nun können mehrere Discovery-Regeln mit `vfs.fs.discovery` eingerichtet werden, wobei für jede Regel unterschiedliche Parameter verwendet werden können, z. B. `vfs.fs.discovery[foo]`, `vfs.fs.discovery[bar]` usw.

AllowKey

Erlaubt die Ausführung derjenigen Datenpunkt-Schlüssel, die mit einem Muster übereinstimmen. Das Schlüsselmuster ist ein Wildcard-Ausdruck, der das Zeichen * unterstützt, um eine beliebige Anzahl beliebiger Zeichen abzugleichen.
 Mehrere Regeln für den Schlüsselabgleich können in Kombination mit `DenyKey`, `AllowKeyRegexp` und `DenyKeyRegexp` definiert werden. Die Parameter werden nacheinander entsprechend ihrer Reihenfolge verarbeitet. Die Reihenfolge bestimmt, welche übereinstimmende Regel Vorrang hat. Siehe auch: Restricting agent checks.

AllowKeyRegexp

Ermöglicht die Ausführung derjenigen Datenpunktschlüssel, die einem Muster entsprechen.
Das Schlüsselmuster ist ein regulärer Ausdruck.
 Mehrere Regeln für den Datenpunktschlüsselabgleich können in Kombination mit `DenyKeyRegexp`, `AllowKey` und `DenyKey` definiert werden. Die Parameter werden entsprechend ihrer Reihenfolge nacheinander verarbeitet. Die Reihenfolge bestimmt, welche übereinstimmende Regel Vorrang hat. Siehe auch: Einschränken von Agent-Prüfungen.

Beispiel:

```
AllowKeyRegexp=^system\.run\[(?:/usr)?/bin/hostname(?: -[fsd])?\]$
```

BufferSend

Legt die Häufigkeit fest, mit der Werte aus dem Puffer an den Zabbix Server/Proxy gesendet werden (in Sekunden).
Beachten Sie, dass die Daten früher gesendet werden, wenn der Puffer voll ist.

Standard: 5
 Bereich: 1-3600

BufferSize

Legt die maximale Anzahl von Werten im Speicherpuffer fest.
 Der Agent sendet alle gesammelten Daten an den Zabbix Server oder Proxy, wenn der Puffer voll ist.
Dieser Parameter sollte nur verwendet werden, wenn der persistente Puffer deaktiviert ist (`EnablePersistentBuffer=0`).

Standard: 1000
 Bereich: 2-65535

ControlSocket

Geben Sie den Control-Socket an, der verwendet wird, um Laufzeitbefehle mit der Option `-R` zu senden.

Standard: `\\.\pipe\agent.sock`

DebugLevel

Legt den Debug-Level fest: - *0* - grundlegende Informationen über das Starten und Beenden von Zabbix-Prozessen; - *1* - kritische Informationen; - *2* - Fehlerinformationen; - *3* - Warnungen; - *4* - für Debugging (liefert viele Informationen); - *5* - erweitertes Debugging (liefert noch mehr Informationen).

Standard: 3
 Bereich: 0-5

DenyKey

Verhindert die Ausführung von Datenpunktschlüsseln, die einem Muster entsprechen. Das Schlüsselmuster ist ein Platzhalterausdruck, der das Zeichen * unterstützt, um eine beliebige Anzahl beliebiger Zeichen zu finden.
 Mehrere Regeln für den Abgleich von Schlüsseln können in Kombination mit `AllowKey`, `AllowKeyRegexp` und `DenyKeyRegexp` definiert werden. Die Parameter werden entsprechend ihrer Reihenfolge nacheinander verarbeitet. Die Reihenfolge bestimmt, welche übereinstimmende Regel Vorrang hat.

Siehe auch: Einschränken von Agent-Prüfungen.

Standard: `DenyKey=system.run[*]`

DenyKeyRegexp

Verhindert die Ausführung derjenigen Datenpunktschlüssel, die einem Muster entsprechen.
Das Schlüssel-Muster ist ein regulärer Ausdruck.
 Mehrere Regeln für den Datenpunktschlüssel können in Kombination mit

AllowKeyRegexp, AllowKey und DenyKey definiert werden. Die Parameter werden entsprechend ihrer Reihenfolge nacheinander verarbeitet. Die Reihenfolge bestimmt, welche übereinstimmende Regel Vorrang hat.
 Alle system.run[]-Datenpunkte werden standardmäßig abgelehnt (DenyKey=system.run[*]), sofern keine andere system.run[*]-Regel angegeben ist.

Siehe auch: Agent-Prüfungen einschränken.

EnablePersistentBuffer

Aktiviert die Verwendung des lokalen persistenten Speichers für aktive Datenpunkte. Wenn der persistente Speicher deaktiviert ist, wird der Speicherpuffer verwendet.

Standard: 0
 Werte: 0 - deaktiviert, 1 - aktiviert

EnableProfiler

Aktiviert die regelmäßige Go-Profilerstellung beim Start des Agent.

Die Profilerstellung kann zur Laufzeit auch mit der Option −R aktiviert oder deaktiviert werden.
Während periodic_prof_execute Profile erfasst, werden alle Laufzeitbefehle des Profilers mit dem Status „beschäftigt“ abgelehnt.

Standard: 0
 Werte: 0 - deaktiviert, 1 - aktiviert

ForceActiveChecksOnStart

Führt aktive Prüfungen unmittelbar nach dem Neustart für die zuerst empfangene Konfiguration aus. Auch als Konfigurationsparameter pro Plugin verfügbar, zum Beispiel: Plugins.Uptime.System.ForceActiveChecksOnStart=1

Standard: 0
 Werte: 0 - deaktiviert, 1 - aktiviert

HeartbeatFrequency

Legt die Frequenz der Heartbeat-Nachrichten in Sekunden fest. Wird zur Überwachung der Verfügbarkeit aktiver Checks verwendet.
0 - Heartbeat-Nachrichten deaktiviert.

Standard: 60
 Bereich: 0-3600

HostInterface

Geben Sie die Host-Schnittstelle (IP-Adresse oder DNS-Name) an, die während der Autoregistrierung des Hosts verwendet wird. Dieser Wert wird verwendet, um die Schnittstelle im neu erstellten Host zu befüllen, und ermöglicht es, entweder die IP- oder die DNS-Adresse explizit zu konfigurieren. Weitere Informationen finden Sie unter DNS als Standardschnittstelle verwenden.

Wenn nicht definiert, wird der Wert aus HostInterfaceItem übernommen.

Der Agent gibt einen Fehler aus und startet nicht, wenn der Wert die Grenze von 255 Zeichen überschreitet.

Bereich: 0-255 Zeichen

HostInterfaceItem

Geben Sie einen Datenpunkt an, um während der Autoregistrierung des Hosts die Hostschnittstelle (IP-Adresse oder DNS-Name) zu bestimmen.
Dieser Wert wird nur verwendet, wenn HostInterface nicht definiert ist.
Weitere Informationen finden Sie unter DNS als Standardschnittstelle verwenden.

Während einer Autoregistrierungsanfrage protokolliert der Agent eine Warnmeldung, wenn der vom angegebenen Datenpunkt zurückgegebene Wert das Limit von 255 Zeichen überschreitet.

Der Datenpunkt system.run[] wird unabhängig von den Einstellungen AllowKey, AllowKeyRegexp, DenyKey oder DenyKeyRegexp unterstützt.

HostMetadata

Geben Sie die Metadaten an, die zur Identifizierung oder Unterscheidung des Hosts während der Autoregistrierung des Hosts (aktiver Agent) verwendet werden.
HostMetadata ermöglicht die Unterscheidung von Hosts über den Hostnamen hinaus.

Wenn nicht definiert, wird der Wert aus HostMetadataItem abgerufen.

Der Agent gibt einen Fehler aus und wird nicht gestartet, wenn der angegebene Wert das Limit von 2034 Byte überschreitet oder aus einer Nicht-UTF-8-Zeichenfolge besteht.
Wenn ein Parameter eine IP-Adresse oder einen DNS-Namen erwartet, werden Werte, die zwar gültiges UTF-8 enthalten, aber keine gültigen IP-Adressen oder DNS-Namen sind, ebenfalls abgelehnt und als ungültig gemeldet.

Mehrzeilige Metadaten werden nicht unterstützt – die Ausgabe wird beim ersten Zeilenumbruch abgeschnitten.

Bereich: 0–2034 Byte

HostMetadataItem

Geben Sie einen Datenpunkt des Zabbix Agent an, um Host-Metadaten abzurufen.
Diese Option wird nur verwendet, wenn `HostMetadata` nicht definiert ist.

Der Wert von `HostMetadataItem` wird bei jedem Versuch der Autoregistrierung abgerufen und nur während des Autoregistrierungsprozesses des Hosts (aktiver Agent) verwendet.
Mit `HostMetadataItem` können Hosts über den Hostnamen hinaus unterschieden werden.

Benutzerparameter und Aliase werden unterstützt.
Der Datenpunkt `system.run[]` wird unabhängig von den Einstellungen `AllowKey`, `AllowKeyRegexp`, `DenyKey` oder `DenyKeyRegexp` unterstützt.

Während einer Autoregistrierungsanfrage protokolliert der Agent eine Warnung, wenn der vom angegebenen Datenpunkt zurückgegebene Wert das Limit von 65535 UTF-8-Codepunkten überschreitet.
Der vom Datenpunkt zurückgegebene Wert muss eine UTF-8-Zeichenfolge sein; andernfalls wird er ignoriert.
Wenn ein Parameter eine IP-Adresse oder einen DNS-Namen erwartet, werden Werte, die zwar gültiges UTF-8 enthalten, aber keine gültigen IP-Adressen oder DNS-Namen sind, ebenfalls abgelehnt und als ungültig gemeldet.

Mehrzeilige Metadaten werden nicht unterstützt – die Ausgabe wird beim ersten Zeilenumbruch abgeschnitten.

Hostname

Geben Sie eindeutige, groß-/kleinschreibungssensitive Hostnamen an. Eine durch Kommas getrennte Liste.
 Erforderlich für aktive Prüfungen und muss mit den auf dem Server konfigurierten Hostnamen übereinstimmen. Der Wert wird aus HostnameItem übernommen, wenn er nicht definiert ist.
 Zulässige Zeichen: alphanumerische Zeichen, '.', ' ', '_' und '-'. Maximale Länge: 128 Zeichen pro Hostname, 2048 Zeichen für die gesamte Zeile.

Standard: Durch HostnameItem festgelegt

HostnameItem

Geben Sie einen Datenpunkt zum Abrufen des Hostnamens an.
Diese Option wird nur verwendet, wenn `Hostname` nicht definiert ist.
Benutzerparameter oder Aliase werden nicht unterstützt, aber der Datenpunkt `system.run[]` wird unabhängig von den Werten für `AllowKey`, `AllowKeyRegexp`, `DenyKey` oder `DenyKeyRegexp` unterstützt.

Standard: `system.hostname`

Include

Geben Sie ein Verzeichnis oder einzelne Dateien an, die in die Konfigurationsdatei aufgenommen werden sollen (standardmäßig unter `C:\Program Files\Zabbix Agent 2`, wenn der Zabbix Agent mithilfe von Windows-MSI-Installationspaketen installiert wurde; in dem bei der Installation angegebenen Ordner, wenn der Zabbix Agent als ZIP-Archiv installiert wurde). Während der Zabbix-Installation wird dieser Parameter auf `/usr/local/etc` gesetzt, sofern Sie dies nicht während der Kompilierung ändern.

Allgemeine Regeln für den Parameter `Include`:

- Der Pfad kann relativ zum Speicherort der Datei `zabbix_agent2.conf` angegeben werden (z. B. `Include=.\zabbix_agent2.d\plu`
- Sie können mehrere `Include`-Parameter in einer Konfigurationsdatei angeben.
- Zabbix verarbeitet den Parameter `Include`, sobald er gelesen wird. Ziehen Sie daher in Betracht, ihn am Ende der Konfigurationsdatei zu platzieren.

Regeln für eingebundene Dateien:

- Jede Datei muss lesbar sein.
- Jede Datei muss eine korrekte Syntax aufweisen. Andernfalls wird die Komponente nicht gestartet.

Regeln für das Einbinden von Verzeichnissen oder Dateimustern:

- Gehen Sie nicht von einer bestimmten Reihenfolge beim Einbinden aus (Dateien werden beispielsweise nicht in alphabetischer Reihenfolge eingebunden).
- Definieren Sie denselben Parameter nicht in mehreren eingebundenen Dateien (beispielsweise, um eine allgemeine Einstellung durch eine spezifische zu überschreiben), da nicht vorhersehbar ist, welcher Wert verwendet wird.
- Achten Sie auf Sicherungskopien, die einige Texteditoren automatisch erstellen (beispielsweise kann beim Bearbeiten von `include/my_specific.conf` die Datei `include/my_specific.conf.BAK` erstellt werden), da beide Dateien eingebunden werden.

Um nur relevante Dateien aus einem Verzeichnis einzubinden, verwenden Sie den Platzhalter (*) für den Musterabgleich.

Beispiel:

```
Include=C:\Program Files\Zabbix Agent2\zabbix_agent2.d\*.conf
```

ListenIP

Geben Sie die IP-Adresse an, auf der der Agent Verbindungen vom Server/Proxy entgegennimmt. Eine durch Kommas getrennte Liste.
 Die erste IP-Adresse wird an den Zabbix-Server gesendet, wenn eine Verbindung zu ihm hergestellt wird, um die Liste der aktiven Prüfungen abzurufen.

Standard: `0.0.0.0`

ListenPort

Geben Sie den Listen-Port des Agent für Verbindungen vom Server/Proxy an.

Standard: 10050
 Bereich: 1024-32767

LogFile

Geben Sie die Protokolldatei des Agent an.

Standard: `c:\zabbix_agent2.log`
 Erforderlich: Ja, wenn LogType auf `file` gesetzt ist; andernfalls nein

LogFileSize

Setzen Sie die maximale Größe der Logdatei (in MB).
0 - automatische Logrotation deaktivieren.
Hinweis: Wenn die Größenbegrenzung der Logdatei erreicht wird und die Dateirotation aus irgendeinem Grund fehlschlägt, wird die vorhandene Logdatei abgeschnitten und neu begonnen.

Standard: 1
 Bereich: 0-1024

LogType

Geben Sie den Typ der Protokollausgabe an: - *file* - schreibt das Protokoll in die durch den Parameter `LogFile` angegebene Datei; - *console* - schreibt das Protokoll in die Standardausgabe.

Standardwert: `file`

PerfCounter

Definieren Sie einen neuen Parameter <parameter_name>, der den Durchschnittswert des Systemleistungsindikators <perf_counter_path> für den angegebenen Zeitraum <period> (in Sekunden) darstellt.
Syntax: <parameter_name>,”<perf_counter_pat

Wenn Sie beispielsweise die durchschnittliche Anzahl der Prozessorunterbrechungen pro Sekunde für die letzte Minute erhalten möchten, können Sie einen neuen Parameter ”interrupts” wie folgt definieren:

```
PerfCounter = interrupts,"\Processor(0)\Interrupts/sec",60
```

Bitte beachten Sie die doppelten Anführungszeichen um den Pfad des Leistungsindikators. Der Parametername (interrupts) ist als Datenpunkt-Schlüssel zu verwenden, wenn ein Datenpunkt erstellt wird. Beispielwerte für die Berechnung des Durchschnittswerts werden jede Sekunde erfasst.
Sie können `typeperf -qx` ausführen, um die Liste aller in Windows verfügbaren Leistungsindikatoren anzuzeigen.

PerfCounterEn

Definieren Sie einen neuen Parameter <parameter_name>, der den Durchschnittswert des Systemleistungsindikators <perf_counter_path> für den angegebenen Zeitraum <period> (in Sekunden) liefert. Im Vergleich zu PerfCounter müssen die perfcounter-Pfade auf Englisch angegeben werden. Wird nur unter **Windows Server 2008/Vista** und höher unterstützt.
Syntax: <parameter_name>,”<perf_counter_path>”,<period>

Wenn Sie beispielsweise die durchschnittliche Anzahl der Prozessorunterbrechungen pro Sekunde für die letzte Minute erhalten möchten, können Sie einen neuen Parameter ”interrupts” wie folgt definieren:

```
PerfCounterEn = interrupts,"\Processor(0)\Interrupts/sec",60
```

Beachten Sie bitte die doppelten Anführungszeichen um den Pfad des Leistungsindikators. Der Parametername (interrupts) wird beim Erstellen eines Datenpunkts als item key verwendet. Die zur Berechnung des Durchschnittswerts verwendeten Samples werden jede Sekunde erfasst.
Die Liste der englischen Zeichenfolgen finden Sie, indem Sie den folgenden Registrierungsschlüssel anzeigen: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009`.

PersistentBufferFile

Geben Sie die Datei an, in der der Zabbix Agent 2 die SQLite-Datenbank speichern soll. Es muss ein vollständiger Dateiname sein.
 Dieser Parameter wird nur verwendet, wenn der persistente Puffer aktiviert ist (`EnablePersistentBuffer=1`).

PersistentBufferPeriod

Legt die Dauer fest, für die Daten gespeichert werden, wenn keine Verbindung zum Server oder Proxy besteht. Ältere Daten gehen verloren. Logdaten werden beibehalten.
 Dieser Parameter wird nur verwendet, wenn der persistente Puffer aktiviert ist (`EnablePersistentBuffer=1`).

Standard: `1h`
 Bereich: 1m-365d

Plugins.<PluginName>.System.Capacity

Legt das Limit der gleichzeitig ausführbaren Prüfungen pro <`PluginName`>-Plugin fest.

Standard: 1000 Bereich: 1-1000

Plugins.Log.MaxLinesPerSecond

Legt die maximale Anzahl neuer Zeilen fest, die der Agent pro Sekunde an den Zabbix Server oder Proxy sendet, wenn aktive Prüfungen vom Typ `log`, `logrt` und `eventlog` verarbeitet werden. Der angegebene Wert wird durch den Parameter `maxlines` überschrieben, sofern dieser im `log`, `logrt` oder `eventlog` Datenpunkt-Schlüssel angegeben ist.
Hinweis: Zabbix verarbeitet 10-mal mehr neue Zeilen als in `MaxLinesPerSecond` festgelegt, um in Log-Datenpunkten nach der erforderlichen Zeichenfolge zu suchen.

Standard: 20
 Bereich: 1-1000

Plugins.SystemRun.LogRemoteCommands

Aktivieren Sie die Protokollierung der ausgeführten Shell-Befehle als Warnungen. Die Befehle werden nur protokolliert, wenn sie remote ausgeführt werden. Protokolleinträge werden nicht erstellt, wenn `system.run[]` lokal über die Parameter `HostMetadataItem`, `HostInterfaceItem` oder `HostnameItem` gestartet wird.

Standard: `0`
 Werte: 0 - deaktiviert, 1 - aktiviert

PluginSocket

Geben Sie den Pfad zur Windows-Named-Pipe für die Kommunikation mit ladbaren Plugins an.

Standard: `\\.\pipe\agent.plugin.sock`

PluginTimeout

Legt das Timeout für Verbindungen mit ladbaren Plugins in Sekunden fest.

Standard: `Timeout`
 Bereich: 1-30

ProfilerDir

Das Verzeichnis, in das Profildaten geschrieben werden.

Standard: `c:\zabbix_pprof`

ProfilerMaxFilesPerProfile

Die maximale Anzahl der Dateien, die für jeden Profiltyp aufbewahrt werden sollen.
Das Limit wird nach jedem Profildump auf abgeschlossene Dateien angewendet.
Das aktive CPU-Profil wird in einer ausgeblendeten temporären Datei erfasst und nicht auf das Limit angerechnet.
Dateien, die von `periodic_prof_execute` erstellt werden, unterliegen ebenfalls dem Limit.

Standard: 10
 Bereich: 1-100

ProfilerInterval

Das Schreibintervall für Profildaten in Sekunden.

Standard: 3600
 Bereich: 1-86400

RefreshActiveChecks

Geben Sie die Häufigkeit der Aktualisierung der Liste der aktiven Prüfungen in Sekunden an.
 Beachten Sie, dass nach einem fehlgeschlagenen Versuch, die aktiven Prüfungen zu aktualisieren, der nächste Aktualisierungsversuch nach 60 Sekunden erfolgt.

Standard: 5
 Bereich: 1-86400

Server

Geben Sie die IP-Adressen, optional in CIDR-Notation, oder DNS-Namen von Zabbix-Servern/Proxys an. Eine durch Kommas getrennte Liste.
 Wenn der Parameter nicht angegeben ist, sind passive Prüfungen deaktiviert und der Agent lauscht auf keinem TCP-Port.
 Eingehende Verbindungen werden nur von den hier aufgeführten Hosts akzeptiert. Wenn die IPv6-Unterstützung aktiviert ist, werden `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` gleich behandelt, und `::/0` erlaubt jede IPv4- oder IPv6-Adresse. `0.0.0.0/0` kann verwendet werden, um jede IPv4-Adresse zuzulassen. Leerzeichen sind erlaubt.

Beispiel:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

ServerActive

Geben Sie die Adresse des Zabbix Server/Proxy oder die Clusterkonfiguration an, von der aktive Prüfungen abgerufen werden sollen.
 Wenn der Parameter nicht angegeben ist, sind aktive Prüfungen deaktiviert.

Die Adresse des Server/Proxy ist eine IP-Adresse oder ein DNS-Name mit optionalem Port, getrennt durch einen Doppelpunkt. Wenn der Port nicht angegeben ist, wird der Standardport verwendet.
 IPv6-Adressen müssen in eckige Klammern gesetzt werden, wenn für diesen Host ein Port angegeben ist. Wenn kein Port angegeben ist, sind eckige Klammern für IPv6-Adressen optional.

Die Clusterkonfiguration besteht aus einer oder mehreren Adressen von Mitgliedern einer Server- oder Proxy-Gruppe, getrennt durch Semikolon.

Mehrere Zabbix Server/Cluster und Zabbix Proxies können angegeben werden, getrennt durch Komma.

Außer bei Verwendung von Proxy-Gruppen sollte von jedem Zabbix Server/Cluster nicht mehr als ein Zabbix Proxy angegeben werden. Wenn ein Zabbix Proxy angegeben ist, sollte der Zabbix Server/Cluster für diesen Proxy nicht angegeben werden.

Mehrere durch Kommas getrennte Adressen können angegeben werden, um mehrere unabhängige Zabbix Server parallel zu verwenden. Leerzeichen sind erlaubt.

Beispiel für einen Zabbix Proxy:

```
ServerActive=127.0.0.1:10051
```

Beispiel für eine Zabbix Proxy-Gruppe:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

Beispiel für mehrere Server:

```
ServerActive=127.0.0.1:20051,zabbix.domain,\[::1\]:30051,::1,\[12fc::1\]
```

Beispiel für Hochverfügbarkeit:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

Beispiel für Hochverfügbarkeit mit zwei Clustern und einem Server:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

SourceIP

Geben Sie die Quell-IP-Adresse an für:

- ausgehende Verbindungen zum Zabbix Server oder Zabbix Proxy;
- Verbindungen, die beim Ausführen bestimmter Datenpunkte (`web.page.get`, `net.tcp.port` usw.) hergestellt werden.

StatusPort

Legt den Port fest, auf dem der Agent auf HTTP-Statusanfragen lauscht (`http://localhost:<port>/status`).

Bereich: 1024-32767

Timeout

Geben Sie an, wie lange in Sekunden auf den Aufbau der Verbindung und den Datenaustausch mit dem Zabbix Proxy oder Server gewartet werden soll.

Dieser Parameter definiert die Dauer verschiedener Kommunikationsvorgänge, einschließlich:

- Warten auf eine Antwort vom Zabbix Server;
- Senden von Anfragen an den Zabbix Server, einschließlich Anfragen zur Datenpunktkonfiguration und Datenpunktdaten in active checks;
- Abrufen von Protokolldaten über die Überwachung von Logdateien oder Windows-Ereignisprotokollen;
- Senden von Heartbeat-Nachrichten;
- maximale Dauer für `vfs.*`-Prüfungen;
- als Fallback in Szenarien zu dienen, in denen ein Server oder Proxy älter als Version 7.0 Prüfungen ohne Timeouts sendet.

Dieses Timeout wird **nicht** für jene Agent-Prüfungen verwendet, die in der Frontend konfigurierbare Timeout-Einstellungen haben (auf globaler Ebene, auf Proxy-Ebene oder pro Datenpunkt).

Standard: 3
 Bereich: 1-30

TLSAccept

Geben Sie den für eingehende Verbindungen erforderlichen Verschlüsselungsgrad an.
 Wird für passive Prüfungen verwendet.

Es können mehrere Werte angegeben werden, getrennt durch Kommas:

- *unencrypted* - Verbindungen ohne Verschlüsselung akzeptieren (Standard);
- *psk* - Verbindungen mit TLS und einem vorab gemeinsam genutzten Schlüssel (PSK) akzeptieren;
- *cert* - Verbindungen mit TLS und einem Zertifikat akzeptieren.

Pflicht: ja, wenn TLS-Zertifikat- oder PSK-Parameter definiert sind (auch für eine *unencrypted* Verbindung); andernfalls nein

TLSCAFile

Geben Sie den vollständigen Pfadnamen der Datei an, die die Zertifikate der obersten CA(s) für die Überprüfung von Peer-Zertifikaten enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

TLSCertFile

Geben Sie den vollständigen Pfadnamen der Datei an, die das Agent-Zertifikat oder die Zertifikatskette enthält und für verschlüsselte Kommunikation mit Zabbix-Komponenten verwendet wird.

TLSCipherAll13

Geben Sie die OpenSSL-Cipher-String (TLS 1.3) an. Überschreiben Sie die standardmäßigen Auswahlkriterien für die Ciphersuite für zertifikats- und PSK-basierte Verschlüsselung.

Beispiel:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherCert13

Geben Sie die OpenSSL-Cipher-String (TLS 1.3) an. Überschreiben Sie die standardmäßigen Auswahlkriterien für die Ciphersuite bei zertifikatsbasierter Verschlüsselung.

Beachten Sie, dass dieser Parameter nicht zusammen mit `TLSAccept=cert,psk` verwendet werden kann; für eine Zertifikatsverbindung (`TLSConnect=cert`) verwenden Sie stattdessen `TLSCipherAll13`.

TLSCipherPSK13

Geben Sie die OpenSSL-Cipher-String (TLS 1.3) an. Überschreiben Sie die standardmäßigen Auswahlkriterien für die Ciphersuite bei PSK-basierter Verschlüsselung.

Beispiel:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Geben Sie die für Agent-Verbindungen zum Zabbix-Server oder Proxy erforderliche Verschlüsselungsstufe an.
 Wird für aktive Prüfungen verwendet.
 Es kann nur ein Wert angegeben werden: - *unencrypted* - Verbindung ohne Verschlüsselung herstellen (Standard); - *psk* - Verbindung mit TLS und einem vorab gemeinsam genutzten Schlüssel (PSK) herstellen; - *cert* - Verbindung mit TLS und einem Zertifikat herstellen.

Pflicht: ja, wenn TLS-Zertifikat- oder PSK-Parameter definiert sind (auch für eine *unencrypted* Verbindung); andernfalls nein

TLSCRLFile

Geben Sie den vollständigen Pfadnamen der Datei an, die widerrufene Zertifikate enthält. Dieser Parameter wird für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet.

TLSKeyFile

Geben Sie den vollständigen Pfadnamen der Datei an, die den privaten Schlüssel des Agent enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

TLSPSKFile

Geben Sie den vollständigen Pfadnamen der Datei an, die den vorab gemeinsam genutzten Schlüssel des Agent enthält und für verschlüsselte Kommunikation mit dem Zabbix-Server oder Proxy verwendet wird.

TLSPSKIdentity

Geben Sie die Identitätszeichenfolge des vorab gemeinsam genutzten Schlüssels an, die für verschlüsselte Kommunikation mit dem Zabbix-Server oder Proxy verwendet wird.

TLSServerCertIssuer

Geben Sie den zulässigen Aussteller des Server- (Proxy-) Zertifikats an.

TLSServerCertSubject

Geben Sie das zulässige Zertifikats-Subject des Servers (Proxy) an.

UnsafeUserParameters

Erlaubt, dass alle Zeichen in Argumenten an benutzerdefinierte Parameter übergeben werden. Die folgenden Zeichen sind nicht erlaubt: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
Zusätzlich sind Zeilenumbruchzeichen nicht erlaubt.

Standard: 0
 Werte: 0 - nicht erlauben, 1 - erlauben

UserParameter

Geben Sie einen benutzerdefinierten Parameter an, der überwacht werden soll. Es können mehrere benutzerdefinierte Parameter vorhanden sein.
Format: `UserParameter=<key>,<shell command>`
Beachten Sie, dass der Shell-Befehl keinen leeren String oder nur EOL zurückgeben darf. Shell-Befehle dürfen relative Pfade verwenden, wenn der Parameter `UserParameterDir` angegeben ist.

Beispiel:

```
UserParameter=system.test,dir /b | find /c /v ""
UserParameter=check_cpu,.\custom\_script.bat
```

UserParameterDir

Geben Sie den standardmäßigen Suchpfad für `UserParameter`-Befehle an. Wenn diese Option verwendet wird, wechselt der Agent vor der Ausführung eines Befehls in das hier angegebene Arbeitsverzeichnis. Dadurch können UserParameter-Befehle ein relatives Präfix `.\` anstelle eines vollständigen Pfads verwenden.
Es ist nur ein Eintrag zulässig.

Beispiel:

```
UserParameterDir=C:\opt\myscripts
```

Konfigurationsparameter: Plugins

Siehe die Konfigurationsparameter für 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

Prinzipien der Plugin-Konfiguration

Diese Seite beschreibt Prinzipien und Best Practices für die Plugin-Konfiguration.

Alle Plugins werden mithilfe des Parameters `Plugins.*` konfiguriert. Dieser kann entweder Bestandteil der Konfigurationsdatei von Zabbix Agent 2 oder der Konfigurationsdatei eines Plugins sein. Wenn ein Plugin eine separate Konfigurationsdatei verwendet, muss der Pfad zu dieser Datei im Parameter `Include` der Konfigurationsdatei von Zabbix Agent 2 angegeben werden.

Ein typischer Plugin-Parameter hat folgende Struktur:

`Plugins.<PluginName>.<Parameter>=<Value>`

Zusätzlich gibt es zwei spezielle Parametergruppen:

- `Plugins.<PluginName>.Default.<Parameter>=<Value>` wird zum Definieren von Standardparameterwerten verwendet.
- `Plugins.<PluginName>.<SessionName>.<Parameter>=<Value>` wird zum Definieren separater Parametersätze für verschiedene Überwachungsziele mithilfe von benannten Sitzungen verwendet.

Alle Parameternamen müssen die folgenden Anforderungen erfüllen:

- Es wird empfohlen, die Namen Ihrer Plugins großzuschreiben.
- Der Parameter muss großgeschrieben werden.
- Sonderzeichen sind nicht zulässig.
- Die Verschachtelung ist nicht auf eine maximale Ebene begrenzt.
- Die Anzahl der Parameter ist nicht begrenzt.

Um beispielsweise aktive Prüfungen, die kein *Scheduling*-Aktualisierungsintervall haben, direkt nach dem Neustart des Agent nur für das Uptime-Plugin auszuführen, setzen Sie `Plugins.Uptime.System.ForceActiveChecksOnStart=1` in der Konfigurationsdatei. Um auf ähnliche Weise ein benutzerdefiniertes Limit für gleichzeitige Prüfungen für das CPU-Plugin festzulegen, setzen Sie `Plugins.CPU.System.Capacity=N` in der Konfigurationsdatei.

Standardwerte

Sie können Standardwerte für die verbindungsbezogenen Parameter (URI, Benutzername, Passwort usw.) in der Konfigurationsdatei im folgenden Format festlegen:

```
Plugins.<PluginName>.Default.<Parameter>=<Value>
```

Beispielsweise `Plugins.Mysql.Default.Username=zabbix`, `Plugins.MongoDB.Default.Uri=tcp://127.0.0.1:27017` usw.

Wenn für einen solchen Parameter weder in einem Datenpunktschlüssel noch in den Parametern der benannten Sitzung ein Wert angegeben ist, verwendet das Plugin den Standardwert. Wenn auch für einen Standardparameter kein Wert definiert ist, werden fest codierte Standardwerte verwendet.

> **Note:**
> Wenn ein Datenpunktschlüssel keine Parameter enthält, versucht Zabbix Agent 2, die Metrik mithilfe der im Abschnitt mit den Standardparametern definierten Werte zu erfassen.

Benannte Sitzungen

Benannte Sitzungen stellen eine zusätzliche Ebene von Plugin-Parametern dar und können verwendet werden, um für jede der überwachten Instanzen separate Sätze von Authentifizierungsparametern anzugeben. Jeder Parameter einer benannten Sitzung sollte die folgende Struktur haben:

```
Plugins.<PluginName>.Sessions.<SessionName>.<Parameter>=<Value>
```

Ein Sitzungsname kann als Schlüsselparameter eines connString-Elements verwendet werden, anstatt URI, Benutzername und/oder Passwort separat anzugeben.

In Datenpunkt-Schlüsseln kann der erste Parameter entweder ein connString oder eine URI sein. Wenn der erste Schlüsselparameter mit keinem Sitzungsnamen übereinstimmt, wird er als URI behandelt. Beachten Sie, dass die Übergabe eingebetteter URI-Anmeldedaten im Datenpunkt-Schlüssel nicht unterstützt wird; verwenden Sie stattdessen benannte Sitzungsparameter.

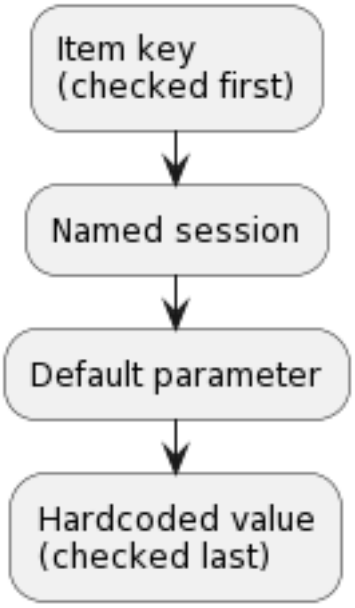
Die Liste der verfügbaren benannten Sitzungsparameter hängt vom Plugin ab.

Es ist möglich, Sitzungsparameter zu überschreiben, indem neue Werte in den Datenpunkt-Schlüsselparametern angegeben werden (siehe Beispiel).

Wenn ein Parameter für die benannte Sitzung nicht definiert ist, verwendet Zabbix Agent 2 den im Standard-Plugin-Parameter definierten Wert.

Parameterpriorität

Zabbix Agent 2-Plugins suchen nach verbindungsbezogenen Parameterwerten in der folgenden Reihenfolge:



1. Der erste Parameter des Datenpunktschlüssels wird mit den Sitzungsnamen verglichen. Wenn keine Übereinstimmung gefunden wird, wird er als tatsächlicher Wert behandelt; in diesem Fall wird Schritt 3 übersprungen. Wenn eine Übereinstimmung gefunden wird, muss der Parameterwert (in der Regel eine URI) in der benannten Sitzung definiert sein.
2. Andere Parameter werden, sofern definiert, aus dem Datenpunktschlüssel übernommen.
3. Wenn ein Parameter des Datenpunktschlüssels (zum Beispiel password) leer ist, sucht das Plugin nach dem entsprechenden Parameter der benannten Sitzung.
4. Wenn der Sitzungsparameter ebenfalls nicht angegeben ist, wird der im entsprechenden Standardparameter definierte Wert verwendet.
5. Wenn alles andere fehlschlägt, verwendet das Plugin den fest codierten Standardwert.

Beispiel 1

Überwachung von zwei Instanzen "MySQL1" und "MySQL2".

Konfigurationsparameter:

```
Plugins.Mysql.Sessions.MySQL1.Uri=tcp://127.0.0.1:3306
Plugins.Mysql.Sessions.MySQL1.User=mysql1_user
Plugins.Mysql.Sessions.MySQL1.Password=unique_password
Plugins.Mysql.Sessions.MySQL2.Uri=tcp://192.0.2.1:3306
Plugins.Mysql.Sessions.MySQL2.User=mysql2_user
Plugins.Mysql.Sessions.MySQL2.Password=different_password
```

Als Ergebnis dieser Konfiguration kann jeder Sitzungsname als connString in einem item key verwendet werden, z. B. `mysql.ping[MySQL1]` oder `mysql.ping[MySQL2]`.

Beispiel 2

Einige der Parameter im Datenpunkt-Schlüssel angeben.

Konfigurationsparameter:

```
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
```

Datenpunkt-Schlüssel: `pgsql.ping[session1,new_username,,postgres]`

Als Ergebnis dieser Konfiguration verbindet sich der Agent mit PostgreSQL unter Verwendung der folgenden Parameter:

- URI aus dem Sitzungsparameter: `192.0.2.234:5432`
- Benutzername aus dem Datenpunkt-Schlüssel: `new_username`
- Passwort aus dem Sitzungsparameter (da es im Datenpunkt-Schlüssel ausgelassen wird): `session_password`
- Datenbankname aus dem Datenpunkt-Schlüssel: `postgres`

Beispiel 3

Erfassung eines Metrikwerts unter Verwendung der Standardkonfigurationsparameter.

Konfigurationsparameter:

```
Plugins.PostgreSQL.Default.Uri=tcp://192.0.2.234:5432
Plugins.PostgreSQL.Default.User=zabbix
Plugins.PostgreSQL.Default.Password=password
```

Item-Schlüssel: `pgsql.ping[,,,postgres]`

Als Ergebnis dieser Konfiguration verbindet sich der Agent mit PostgreSQL unter Verwendung der folgenden Parameter:

- Standard-URI: `192.0.2.234:5432`
- Standardbenutzername: `zabbix`
- Standardpasswort: `password`
- Datenbankname aus dem Item-Schlüssel: `postgres`

Verbindungen

Einige Plugins unterstützen das gleichzeitige Erfassen von Metriken von mehreren Instanzen. Sowohl lokale als auch entfernte Instanzen können überwacht werden. TCP- und Unix-Socket-Verbindungen werden unterstützt.

Es wird empfohlen, Plugins so zu konfigurieren, dass Verbindungen zu Instanzen im offenen Zustand gehalten werden. Die Vorteile sind eine geringere Netzwerkauslastung, geringere Latenz sowie ein reduzierter CPU- und Speicherverbrauch aufgrund der geringeren Anzahl an Verbindungen. Die Client-Bibliothek übernimmt dies.

> **Note:**
> Der Zeitraum, für den ungenutzte Verbindungen offen bleiben sollen, kann über den Parameter `Plugins.<PluginName>.KeepAlive` festgelegt werden. Beispiel: `Plugins.Memcached.KeepAlive`

Ceph-Plugin

Übersicht

Konfigurationsparameter ermöglichen die Anpassung des Verhaltens des Ceph-Plugins.

Es wird empfohlen, sie in einer eigenen Konfigurationsdatei (z. B. `ceph.conf`) anzugeben, statt in der Konfigurationsdatei von Agent 2. Verwenden Sie dann die Direktive Include, um diese Datei in die Konfiguration von Zabbix Agent 2 einzubinden.

Alle Parameter sind **optional**, sofern nicht ausdrücklich angegeben ist, dass der Parameter obligatorisch ist.

Diese Seite ist wie folgt aufgebaut:

- Parameterübersicht (um die vollständigen Details eines Parameters anzuzeigen, klicken Sie auf seinen Namen)
- Parameterdetails

Parameter	Beschreibung
Plugins.Ceph.Default.ApiKey	Geben Sie den standardmäßigen API-Schlüssel für die Verbindung zu Ceph an.
Plugins.Ceph.Default.Mode	Geben Sie den standardmäßigen Modus für die Verbindung zu Ceph an.
Plugins.Ceph.Default.URI	Geben Sie die standardmäßige URI für die Verbindung zu Ceph an.
Plugins.Ceph.Default.User	Geben Sie den standardmäßigen Benutzernamen für die Verbindung zu Ceph an.
Plugins.Ceph.InsecureSkipVerify	Geben Sie an, ob ein HTTP-Client die Zertifikatskette und den Hostnamen des Servers überprüfen soll.
Plugins.Ceph.KeepAlive	Legen Sie die maximale Wartezeit fest, bevor ungenutzte Plugin-Verbindungen geschlossen werden.
Plugins.Ceph.Sessions.<SessionName>.ApiKey	Geben Sie den API-Schlüssel der benannten Sitzung für die Verbindung zu Ceph an.
Plugins.Ceph.Sessions.<SessionName>.Mode	Geben Sie den Modus der benannten Sitzung für die Verbindung zu Ceph an.
Plugins.Ceph.Sessions.<SessionName>.URI	Geben Sie die URI der benannten Sitzung für die Verbindung zu Ceph an.
Plugins.Ceph.Sessions.<SessionName>.User	Geben Sie den Benutzernamen der benannten Sitzung für die Verbindung zu Ceph an.
Plugins.Ceph.Timeout	Legen Sie das Timeout für die Ausführung von Anfragen fest.

Beachten Sie:

- Die Standardwerte spiegeln die Prozess-Standardeinstellungen wider, nicht die Werte in den mitgelieferten Konfigurationsdateien;
- Werte unterstützen Umgebungsvariablen;
- Zabbix unterstützt Konfigurationsdateien nur in UTF-8-Kodierung ohne BOM;
- Kommentare, die mit "#" beginnen, werden nur am Anfang der Zeile unterstützt.

Parameterdetails

Plugins.Ceph.Default.ApiKey

Geben Sie den Standard-API-Schlüssel für die Verbindung zu Ceph an.
 Wird verwendet, wenn im Datenpunkt-Schlüssel oder in einer benannten Sitzung kein Wert angegeben ist.

Plugins.Ceph.Default.Mode

Geben Sie den Standardmodus für die Verbindung zu Ceph an.
 Der native Modus wird nur unter Linux und ab Ceph 16 unterstützt. Der restful-Modus funktioniert nicht mit Ceph Version 20 oder neuer.
 Beachten Sie, dass sich der Satz der Benutzeranmeldedaten für jeden Modus unterscheidet und die Modi nicht miteinander kompatibel sind.

Standard: `restful` (veraltet)
 Bereich: native/restful

Plugins.Ceph.Default.URI

Geben Sie die Standard-URI für die Verbindung zu Ceph an.
 Wird verwendet, wenn kein Wert in einem Datenpunkt-Schlüssel oder einer benannten Sitzung angegeben ist.

Sollte keine eingebetteten Anmeldedaten enthalten (diese werden ignoriert).
Muss dem URI-Format entsprechen.
Nur das `https`-Schema wird unterstützt; ein Schema kann weggelassen werden.
Ein Port kann weggelassen werden (Standard=8003).

Standard: `https://localhost:8003`

Beispiele:

```
https://127.0.0.1:8003
localhost
```

Plugins.Ceph.Default.User

Geben Sie den Standardbenutzernamen für die Verbindung zu Ceph an.
 Wird verwendet, wenn in einem Datenpunkt-Schlüssel oder einer benannten Sitzung kein Wert angegeben ist.

Plugins.Ceph.InsecureSkipVerify

Geben Sie an, ob ein HTTP-Client die Zertifikatskette und den Hostnamen des Servers überprüfen soll.
 Wenn *true*, akzeptiert TLS jedes vom Server vorgelegte Zertifikat und jeden Hostnamen in diesem Zertifikat. In diesem Modus ist TLS anfällig für Man-in-the-Middle-Angriffe (nur zu Testzwecken verwenden).

Standard: `false`
 Bereich: false/true

Plugins.Ceph.KeepAlive

Legt die maximale Wartezeit (in Sekunden) fest, bevor ungenutzte Plugin-Verbindungen geschlossen werden.

Standard: 300
 Bereich: 60-900

Plugins.Ceph.Sessions.<SessionName>.ApiKey

Geben Sie den API-Schlüssel der benannten Sitzung für die Verbindung zu Ceph an.
 <SessionName> - Name einer Sitzung, die in Datenpunkt-Schlüsseln verwendet werden soll.

Plugins.Ceph.Sessions.<SessionName>.Mode

Geben Sie den benannten Sitzungsmodus für die Verbindung zu Ceph an.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunkt-Schlüsseln.
 Der native Modus wird nur unter Linux und ab Ceph 16 unterstützt. Der restful-Modus funktioniert nicht mit Ceph-Version 20 oder neuer.
 Beachten Sie, dass sich der Satz von Benutzeranmeldedaten je nach Modus unterscheidet und die Modi nicht miteinander kompatibel sind.

Standard: `restful` (veraltet)
 Bereich: native/restful

Plugins.Ceph.Sessions.<SessionName>.URI

Geben Sie die URI der benannten Sitzung für die Verbindung zu Ceph an.
 <SessionName> - Name einer Sitzung, die in Datenpunkt-Schlüsseln verwendet werden soll.
Sollte keine eingebetteten Anmeldedaten enthalten (diese werden ignoriert).
Muss dem URI-Format entsprechen.
 Es wird nur das Schema `https` unterstützt; ein Schema kann weggelassen werden.
Ein Port kann weggelassen werden (Standard=8003).

Standard: `https://localhost:8003`

Beispiele:

```
https://127.0.0.1:8003
localhost
```

Plugins.Ceph.Sessions.<SessionName>.User

Geben Sie den Benutzernamen der benannten Sitzung an, um eine Verbindung zu Ceph herzustellen.
 <SessionName> - Name einer Sitzung festlegen, die in Datenpunkt-Schlüsseln verwendet wird.

Plugins.Ceph.Timeout

Legt das Zeitlimit für die Ausführung von Anfragen fest (Dauer, die auf den Abschluss einer Anfrage gewartet wird, bevor sie beendet wird) in Sekunden.

Standard: `global timeout`
 Bereich: 1-30

Siehe auch

- Anweisungen zum Konfigurieren von Plugins
- Beschreibung der allgemeinen Konfigurationsparameter von Zabbix Agent 2: Zabbix Agent 2 (UNIX) / Zabbix Agent 2 (Windows)

Docker-Plugin

Übersicht

Konfigurationsparameter ermöglichen die Anpassung des Verhaltens des Docker-Plugins.

Es wird empfohlen, sie in einer eigenen Konfigurationsdatei (z. B. `docker.conf`) anzugeben, statt in der Konfigurationsdatei von Agent 2. Verwenden Sie dann die Direktive Include, um diese Datei in die Konfiguration von Zabbix Agent 2 einzubinden.

Alle Parameter sind **optional**, sofern nicht ausdrücklich angegeben ist, dass der Parameter obligatorisch ist.

Diese Seite ist wie folgt aufgebaut:

- Parameterübersicht (um die vollständigen Details eines Parameters anzuzeigen, klicken Sie auf seinen Namen)
- Parameterdetails

Parameter	Beschreibung
Plugins.Docker.Endpoint	Geben Sie den UNIX-Socket-Speicherort des Docker-Daemons an..
Plugins.Docker.Timeout	Geben Sie das Timeout für die Ausführung der Anfrage an.

Beachten Sie:

- Die Standardwerte spiegeln die Prozess-Standardwerte wider, nicht die Werte in den mitgelieferten Konfigurationsdateien;
- Werte unterstützen Umgebungsvariablen;
- Zabbix unterstützt Konfigurationsdateien nur in UTF-8-Kodierung ohne BOM;
- Kommentare, die mit "#" beginnen, werden nur am Anfang der Zeile unterstützt.

Parameterdetails

Plugins.Docker.Endpoint

Geben Sie den UNIX-Socket-Speicherort des Docker-Daemons an.
 Muss ein Schema enthalten (nur `unix://` wird unterstützt).

Standard: `unix:///var/run/docker.sock`

Plugins.Docker.Timeout

Geben Sie das Timeout für die Ausführung der Anfrage an (die Dauer in Sekunden, die auf den Abschluss einer Anfrage gewartet wird, bevor sie beendet wird).

Standard: `global timeout`
 Bereich: 1–30

Siehe auch

- Beschreibung der allgemeinen Konfigurationsparameter von Zabbix Agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Anweisungen zum Konfigurieren von Plugins

Ember+ Plugin

Übersicht

Konfigurationsparameter ermöglichen die Anpassung des Verhaltens des Ember+ Plugins.

Es wird empfohlen, sie in einer eigenen Konfigurationsdatei (z. B. `emberplus.conf`) statt in der Konfigurationsdatei von Agent 2 anzugeben. Verwenden Sie dann die Direktive Include, um diese Datei in die Konfiguration von Zabbix Agent 2 einzubinden.

Das Ember+ Plugin ist ein ladbares Plugin und ist verfügbar sowie vollständig beschrieben im Ember+ plugin repository.

Dieses Plugin ist derzeit nur aus dem Quellcode kompilierbar (sowohl für Unix als auch für Windows).

Alle Parameter sind **optional**, sofern nicht ausdrücklich angegeben ist, dass der Parameter obligatorisch ist.

Diese Seite ist wie folgt aufgebaut:

- Parameterübersicht (um die vollständigen Details eines Parameters anzuzeigen, klicken Sie auf seinen Namen)
- Parameterdetails

Parameter	Beschreibung
Plugins.EmberPlus.Default.Uri	Geben Sie die Standard-URI für die Verbindung zu Ember+ an.
Plugins.EmberPlus.KeepAlive	Legt die maximale Wartezeit fest, bevor ungenutzte Plugin-Verbindungen geschlossen werden.
Plugins.EmberPlus.Sessions.<SessionName>.Uri	Geben Sie die URI der benannten Sitzung für die Verbindung zu Ember+ an.
Plugins.EmberPlus.System.Path	Geben Sie den Pfad zur ausführbaren Datei des Ember+ Plugins an.
Plugins.EmberPlus.Timeout	Legt das Zeitlimit für die Ausführung von Anfragen fest.

Beachten Sie:

- Die Standardwerte spiegeln die Prozess-Standardeinstellungen wider, nicht die Werte in den mitgelieferten Konfigurationsdateien.
- Werte unterstützen Umgebungsvariablen;
- Zabbix unterstützt Konfigurationsdateien nur in UTF-8-Kodierung ohne BOM.
- Kommentare, die mit "#" beginnen, werden nur am Zeilenanfang unterstützt.

Parameterdetails

Plugins.EmberPlus.Default.Uri

Geben Sie die Standard-URI für die Verbindung zu Ember+ an.
 Das einzige unterstützte Schema ist `tcp://`. Ein Schema kann weggelassen werden. Eingebettete Anmeldedaten werden ignoriert.

Standard: `tcp://localhost:9998`

Plugins.EmberPlus.KeepAlive

Legt die maximale Wartezeit (in Sekunden) fest, bevor ungenutzte Plugin-Verbindungen geschlossen werden.

Standard: 300
 Bereich: 60–900

Plugins.EmberPlus.Sessions.<SessionName>.Uri

Geben Sie die URI der benannten Sitzung für die Verbindung zu Ember+ an.
 Das einzige unterstützte Schema ist `tcp://`. Ein Schema kann weggelassen werden. Eingebettete Anmeldedaten werden ignoriert.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunkt-Schlüsseln.

Standard: `tcp://localhost:9998`

Plugins.EmberPlus.System.Path

Geben Sie den Pfad zur ausführbaren Datei des Ember+-Plugins an.

Beispiel: `Plugins.EmberPlus.System.Path=/usr/libexec/zabbix/zabbix-agent2-plugin-ember-plus`

Plugins.EmberPlus.Timeout

Legt die Dauer in Sekunden fest, die beim ersten Verbindungsaufbau und bei nachfolgenden Vorgängen in der Sitzung auf eine Antwort des Servers gewartet wird.

Standard: `global timeout`
 Bereich: 1–30

Optionen

Parameter	Beschreibung
-V --version	Gibt die Plugin-Version und Lizenzinformationen aus.
-h --help	Gibt Hilfsinformationen aus (Kurzform).
-t, --test <item key>	Startet das Plugin zum Testen (Plugin-Konfiguration wird ignoriert).

Siehe auch:

- Beschreibung der allgemeinen Konfigurationsparameter von Zabbix Agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Anweisungen zur Konfiguration von Plugins

Memcached-Plugin

Übersicht

Konfigurationsparameter ermöglichen die Anpassung des Verhaltens des Memcached-Plugins.

Es wird empfohlen, sie in einer eigenen Konfigurationsdatei (z. B. `memcached.conf`) anzugeben, statt in der Konfigurationsdatei von Agent 2. Verwenden Sie dann die Direktive Include, um diese Datei in die Konfiguration von Zabbix Agent 2 einzubinden.

Alle Parameter sind **optional**, sofern nicht ausdrücklich angegeben ist, dass der Parameter obligatorisch ist.

Diese Seite ist wie folgt aufgebaut:

- Parameterübersicht (um die vollständigen Details eines Parameters anzuzeigen, klicken Sie auf seinen Namen)
- Parameterdetails

Parameter	Beschreibung
Plugins.Memcached.Default.Password	Geben Sie das Standardpasswort für die Verbindung zu Memcached an.
Plugins.Memcached.Default.Uri	Geben Sie die Standard-URI für die Verbindung zu Memcached an.
Plugins.Memcached.Default.User	Geben Sie den Standardbenutzernamen für die Verbindung zu Memcached an.
Plugins.Memcached.KeepAlive	Legt die maximale Wartezeit fest, bevor ungenutzte Plugin-Verbindungen geschlossen werden.
Plugins.Memcached.Sessions.<Session>.Password	Geben Sie das Passwort der benannten Sitzung für die Verbindung zu Memcached an.
Plugins.Memcached.Sessions.<Session>.Uri	Geben Sie die URI der benannten Sitzung für die Verbindung zu Memcached an.
Plugins.Memcached.Sessions.<Session>.User	Geben Sie den Benutzernamen der benannten Sitzung für die Verbindung zu Memcached an.
Plugins.Memcached.Timeout	Legt das Timeout für die Ausführung von Anfragen fest.

Beachten Sie:

- Die Standardwerte spiegeln die Prozess-Standardeinstellungen wider, nicht die Werte in den mitgelieferten Konfigurationsdateien;
- Werte unterstützen Umgebungsvariablen;
- Zabbix unterstützt Konfigurationsdateien nur in UTF-8-Kodierung ohne BOM;
- Kommentare, die mit ”#” beginnen, werden nur am Zeilenanfang unterstützt.

Parameterdetails

Plugins.Memcached.Default.Password

Geben Sie das Standardpasswort für die Verbindung zu Memcached an.
 Wird verwendet, wenn in einem Datenpunkt-Schlüssel oder einer benannten Sitzung kein Wert angegeben ist.

Plugins.Memcached.Default.Uri

Geben Sie die Standard-URI für die Verbindung zu Memcached an.
 Wird verwendet, wenn kein Wert in einem Datenpunkt-Schlüssel oder einer benannten Sitzung angegeben ist.
 Eingebettete Anmeldedaten werden ignoriert.
 Muss dem URI-Format entsprechen. Unterstützte Schemata: `tcp`, `unix`.
 Ein Schema kann weggelassen werden.
 Ein Port kann weggelassen werden (Standard=11211).

Standard: `tcp://localhost:11211`

Beispiele:

```
tcp://localhost:11211
localhost
unix:/var/run/memcached.sock
```

Plugins.Memcached.Default.User

Geben Sie den Standardbenutzernamen für die Verbindung zu Memcached an.
 Wird verwendet, wenn in einem Datenpunkt-Schlüssel oder einer benannten Sitzung kein Wert angegeben ist.

Plugins.Memcached.KeepAlive

Legt die maximale Wartezeit (in Sekunden) fest, bevor ungenutzte Plugin-Verbindungen geschlossen werden.

Standard: 300
 Bereich: 60–900

Plugins.Memcached.Sessions.<SessionName>.Password

Geben Sie das Passwort für die benannte Sitzung an, um eine Verbindung zu Memcached herzustellen.
 <SessionName> - Name einer Sitzung, der in Datenpunkt-Schlüsseln verwendet wird.

Plugins.Memcached.Sessions.<SessionName>.Uri

Geben Sie die URI der benannten Sitzung für die Verbindung zu Memcached an.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunkt-Schlüsseln.
 Eingebettete Anmeldedaten werden ignoriert.
 Muss dem URI-Format entsprechen. Unterstützte Schemata: `tcp`, `unix`.
 Ein Schema kann weggelassen werden.
 Ein Port kann weggelassen werden (Standard=11211).

Beispiele:

```
tcp://localhost:11211
localhost
unix:/var/run/memcached.sock
```

Plugins.Memcached.Sessions.<SessionName>.User

Geben Sie den Benutzernamen der benannten Sitzung für die Verbindung zu Memcached an.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunktschlüsseln.

Plugins.Memcached.Timeout

Legt das Zeitlimit für die Ausführung der Anfrage fest (die Dauer in Sekunden, die auf den Abschluss einer Anfrage gewartet wird, bevor sie beendet wird).

Standard: `global timeout`
 Bereich: 1–30

Siehe auch:

- Beschreibung der allgemeinen Konfigurationsparameter von Zabbix Agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Anweisungen zum Konfigurieren von Plugins

Modbus-Plugin

Übersicht

Konfigurationsparameter ermöglichen die Anpassung des Verhaltens des Modbus-Plugins.

Es wird empfohlen, sie in einer eigenen Konfigurationsdatei (z. B. `modbus.conf`) anzugeben, statt in der Konfigurationsdatei von Agent 2. Verwenden Sie dann die Direktive Include, um diese Datei in die Zabbix Agent 2-Konfiguration einzubinden.

Alle Parameter sind **optional**, sofern nicht ausdrücklich angegeben ist, dass der Parameter obligatorisch ist.

Diese Seite ist wie folgt aufgebaut:

- Parameterübersicht (um die vollständigen Details eines Parameters anzuzeigen, klicken Sie auf seinen Namen)
- Parameterdetails

Parameter	Beschreibung
Plugins.Modbus.Sessions.<SessionName>.Endpoint	Geben Sie den benannten Sitzungs-Endpunkt für die Verbindung zu Modbus an.
Plugins.Modbus.Sessions.<SessionName>.SlaveID	Geben Sie die benannte Slave-ID der Sitzung für die Verbindung zu Modbus an.
Plugins.Modbus.Sessions.<SessionName>.Timeout	Legt das Timeout der benannten Sitzung in Sekunden fest.

Beachten Sie:

- Die Standardwerte spiegeln die Prozess-Standardeinstellungen wider, nicht die Werte in den mitgelieferten Konfigurationsdateien;

- Werte unterstützen Umgebungsvariablen;
- Zabbix unterstützt Konfigurationsdateien nur in UTF-8-Kodierung ohne BOM;
- Kommentare, die mit "#" beginnen, werden nur am Zeilenanfang unterstützt.

Parameterdetails

Plugins.Modbus.Sessions.<SessionName>.Endpoint

Geben Sie den benannten Sitzungs-Endpunkt für die Verbindung zu Modbus an.
 Die Verbindungszeichenfolge besteht aus einem Protokollschema, einer Host-Adresse und einem Port oder einem seriellen Portnamen sowie Attributen.
 <SessionName> - Name einer Sitzung, die in Datenpunkt-Schlüsseln verwendet wird.

Plugins.Modbus.Sessions.<SessionName>.SlaveID

Geben Sie die Slave-ID der benannten Sitzung für die Verbindung zu Modbus an.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunkt-Schlüsseln.

Beispiel: `Plugins.Modbus.Sessions.MB1.SlaveID=20`

Hinweis: Dieser benannte Sitzungsparameter wird nur geprüft, wenn der im Datenpunkt-Schlüssel angegebene Wert für die Slave-ID leer ist.

Plugins.Modbus.Sessions.<SessionName>.Timeout

Legt das Timeout der benannten Sitzung in Sekunden fest.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunkt-Schlüsseln.

Beispiel: `Plugins.Modbus.Sessions.MB1.Timeout=2`

Wenn Sie das Timeout für die Ausführung einer Anfrage festlegen müssen (wie lange gewartet werden soll, bis eine Anfrage abgeschlossen ist, bevor sie beendet wird), verwenden Sie das Formular für die Datenpunkt-Konfiguration.

Siehe auch

- Beschreibung der allgemeinen Konfigurationsparameter von Zabbix Agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Anweisungen zum Konfigurieren von Plugins

MongoDB-Plugin

Übersicht

Konfigurationsparameter ermöglichen die Anpassung des Verhaltens des MongoDB-Plugins.

Es wird empfohlen, sie in einer eigenen Konfigurationsdatei (z. B. `mongodb.conf`) anzugeben, statt in der Konfigurationsdatei von Agent 2. Verwenden Sie dann die Direktive Include, um diese Datei in die Konfiguration von Zabbix Agent 2 einzubinden.

Das MongoDB-Plugin ist ein ladbares Plugin und ist im MongoDB-Plugin-Repository verfügbar und dort vollständig beschrieben.

Alle Parameter sind **optional**, sofern nicht ausdrücklich angegeben ist, dass der Parameter obligatorisch ist.

Beim Verbinden mit MongoDB prüft das Plugin seine Konfiguration in einer bestimmten Reihenfolge, die festlegt, welcher Wert verwendet wird. Die Reihenfolge von der höchsten zur niedrigsten Priorität lautet:

1. Parameter, die in der Verbindungs-URI angegeben sind (z. B. haben Abfrageparameter, die nach ? in `Plugins.MongoDB.Sessions.<s` definiert sind, die höchste Priorität).
2. Parameter im Zabbix Agent 2-Datenpunkt-Schlüssel (z. B. mongodb.collection.stats).
3. Parameter, die in einer benannten Sitzung definiert sind (z. B. `Plugins.MongoDB.Sessions.<SessionName>.User`, `Plugins.MongoDB.Sessions.<SessionName>.Password`).
4. Standardwerte (z. B. `Plugins.MongoDB.Default.User`, `Plugins.MongoDB.Default.Password`).

Beispiel:

- Wenn Sie die Datenbank für die Authentifizierung einer Verbindung zum MongoDB-Server im Parameter `*.<SessionName>.Uri` angeben (z. B. `Plugins.MongoDB.Sessions.<session_name>.Uri=mongodb://user:password@127.0.0.1:27017/?authS` dann betrachtet das Plugin den Parameter `*.<SessionName>.AuthSource` als auf `admin` gesetzt, selbst wenn der Parameter `*.<SessionName>.AuthSource` auf einen anderen Wert gesetzt ist.
- Wenn Sie die URI im Parameter `*.<SessionName>.Uri` angeben (z. B. `Plugins.MongoDB.Sessions.<session_name>.Uri=mong` aber Benutzername und Passwort im Zabbix Agent 2-Datenpunkt-Schlüssel angeben, verwendet das Plugin den Benutzernamen und das Passwort aus dem Datenpunkt-Schlüssel, selbst wenn `Plugins.MongoDB.Sessions.<SessionName>.User` und `Plugins.MongoDB.Sessions.<SessionName>.Password` konfiguriert sind.

Attention:
Wenn das Schema `mongodb+srv://` verwendet wird, kann die URI Parameter aus einem DNS-TXT-Record abrufen. Parameter, die aus diesem TXT-Record gewonnen werden, überschreiben alle anderen Konfigurationen, einschließlich URI-Parametern, Zabbix Agent 2-Datenpunkt-Parametern und Sitzungsparametern.
 Für den in der Verbindungszeichenfolge verwendeten Hostnamen darf nur ein TXT-Record vorhanden sein.

Diese Seite ist wie folgt aufgebaut:

- Parameterübersicht (um die vollständigen Details eines Parameters anzuzeigen, klicken Sie auf seinen Namen)
- Parameterdetails

Parameter	Beschreibung
Plugins.MongoDB.Default.AuthMechanism	Geben Sie den standardmäßigen Authentifizierungsmechanismus für die Verbindung zum MongoDB-Server an.
Plugins.MongoDB.Default.AuthSource	Geben Sie die standardmäßige Datenbank für die Authentifizierung der Verbindung zum MongoDB-Server an.
Plugins.MongoDB.Default.Disable	Deaktivieren Sie die automatische Erkennung von MongoDB-Replica-Set-Knoten.
Plugins.MongoDB.Default.Password	Geben Sie das standardmäßige Passwort an.
Plugins.MongoDB.Default.ReplicaSet	Geben Sie den standardmäßigen Namen des Replica Sets an, der für die automatische Erkennung von MongoDB-Replica-Set-Knoten verwendet wird.
Plugins.MongoDB.Default.ReadPreference	Geben Sie die standardmäßige Reihenfolge für die Verbindung zu MongoDB-Replica-Set-Knoten oder als Fallback an.
Plugins.MongoDB.Default.TLSConnect	Geben Sie die für die Kommunikation zwischen Zabbix Agent 2 und den überwachten Datenbanken erforderliche Verschlüsselungsstufe an.
Plugins.MongoDB.Default.TLSCAFile	Geben Sie den vollständigen Pfadnamen der Datei an, die die Zertifikate der obersten CA(s) für die Überprüfung des Peer-Zertifikats enthält.
Plugins.MongoDB.Default.TLSCertFile	Geben Sie den vollständigen Pfadnamen der Datei an, die das Agent-Zertifikat oder die Zertifikatskette enthält.
Plugins.MongoDB.Default.TLSKeyFile	Geben Sie den vollständigen Pfadnamen der Datei an, die den privaten Schlüssel der Datenbank enthält.
Plugins.MongoDB.Default.Uri	Geben Sie die standardmäßige URI an.
Plugins.MongoDB.Default.User	Geben Sie den standardmäßigen Benutzernamen an.
Plugins.MongoDB.KeepAlive	Legen Sie die maximale Wartezeit fest, bevor ungenutzte Plugin-Verbindungen geschlossen werden.
Plugins.MongoDB.Sessions.<SessionName>.AuthMechanism	Geben Sie den Authentifizierungsmechanismus der benannten Sitzung für die Verbindung zum MongoDB-Server an.
Plugins.MongoDB.Sessions.<SessionName>.AuthSource	Geben Sie die Datenbank der benannten Sitzung für die Authentifizierung der Verbindung zum MongoDB-Server an.
Plugins.MongoDB.Sessions.<SessionName>.Disable	Deaktivieren Sie die automatische Erkennung von MongoDB-Replica-Set-Knoten für die benannte Sitzung.
Plugins.MongoDB.Sessions.<SessionName>.Password	Geben Sie das Passwort der benannten Sitzung an.
Plugins.MongoDB.Sessions.<SessionName>.ReplicaSet	Geben Sie den Namen des Replica Sets der benannten Sitzung an, der für die automatische Erkennung von MongoDB-Replica-Set-Knoten verwendet wird.
Plugins.MongoDB.Sessions.<SessionName>.ReadPreference	Geben Sie die Reihenfolge der benannten Sitzung für die Verbindung zu MongoDB-Replica-Set-Knoten oder als Fallback an.
Plugins.MongoDB.Sessions.<SessionName>.TLSConnect	Geben Sie die für die Kommunikation zwischen Zabbix Agent 2 und den überwachten Datenbanken erforderliche Verschlüsselungsstufe für die benannte Sitzung an.
Plugins.MongoDB.Sessions.<SessionName>.TLSCAFile	Geben Sie den vollständigen Pfadnamen der Datei an, die die Zertifikate der obersten CA(s) für die Überprüfung des Peer-Zertifikats für die benannte Sitzung enthält.
Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile	Geben Sie den vollständigen Pfadnamen der Datei an, die das Agent-Zertifikat oder die Zertifikatskette für die benannte Sitzung enthält.
Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile	Geben Sie den vollständigen Pfadnamen der Datei an, die den privaten Schlüssel der Datenbank für die benannte Sitzung enthält.
Plugins.MongoDB.Sessions.<SessionName>.Uri	Geben Sie die URI der benannten Sitzung an.
Plugins.MongoDB.Sessions.<SessionName>.User	Geben Sie den Benutzernamen der benannten Sitzung an.
Plugins.MongoDB.System.Path	Geben Sie den Pfad zur ausführbaren Datei des Plugins an.
Plugins.MongoDB.Timeout	Legen Sie das Timeout für die Ausführung von Anfragen fest.

Beachten Sie:

- Die Standardwerte spiegeln die Prozess-Standardwerte wider, nicht die Werte in den mitgelieferten Konfigurationsdateien.
- Werte unterstützen Umgebungsvariablen.

- Zabbix unterstützt Konfigurationsdateien nur in UTF-8-Kodierung ohne BOM.
- Kommentare, die mit "#" beginnen, werden nur am Zeilenanfang unterstützt.

Parameterdetails

Plugins.MongoDB.Default.AuthMechanism

Geben Sie den standardmäßigen Authentifizierungsmechanismus für die Verbindung mit dem MongoDB-Server an.
 <SessionName> - Name einer Sitzung festlegen, die in Datenpunkt-Schlüsseln verwendet wird.

Bereich: `SCRAM-SHA-1`, `SCRAM-SHA-256` oder `MONGODB-X509`

Plugins.MongoDB.Default.AuthSource

Geben Sie die Standarddatenbank für die Authentifizierung der Verbindung zum MongoDB-Server an.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunktschlüsseln.

Standard: `admin`

Wenn der Authentifizierungsmechanismus `MONGODB-X509` verwendet wird, wird dieser Parameter automatisch auf `$external` gesetzt; eine Änderung auf einen anderen Wert führt beim Start zu einem Fehler.

Plugins.MongoDB.Default.Direct

Deaktiviert die automatische Erkennung von MongoDB-Replica-Set-Knoten.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunkt-Schlüsseln.

Standard: `false`
 Bereich: `true` - nur mit dem angegebenen Server verbinden und andere Mitglieder des Replica Sets ignorieren, `false` - alle Knoten im Replica Set automatisch erkennen.

Wenn das Schema `tcp://` (veraltet) verwendet wird, wird der Parameter standardmäßig auf `true` gesetzt.

Plugins.MongoDB.Default.Password

Geben Sie das Standardpasswort an.
 <SessionName> - Namen einer Sitzung zur Verwendung in Datenpunktschlüsseln festlegen.

Plugins.MongoDB.Default.ReplicaSet

Geben Sie den Standardnamen des Replica Sets an, der für die automatische Erkennung von MongoDB-Replica-Set-Knoten verwendet wird.
 <SessionName> - Name einer Sitzung definieren, die in Datenpunkt-Schlüsseln verwendet wird.
 Wenn dieser Parameter mit dem Schema `mongodb+srv://` verwendet wird, verhält er sich so, als wären mehrere URLs angegeben worden; ein DNS-Server kann einen TXT-Eintrag haben, der diesen (oder einen anderen) Parameter überschreibt.
 Nicht verfügbar, wenn der Parameter `Plugins.MongoDB.Sessions.<SessionName>.Direct` auf `true` gesetzt ist; der Agent kann dann nicht gestartet werden.

Plugins.MongoDB.Default.ReadPreference

Gibt die Standardreihenfolge für die Verbindung mit Knoten eines MongoDB-Replikatsatzes oder für den Fallback an.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunktschlüsseln.
 Nicht verfügbar, wenn der Parameter `Plugins.MongoDB.Sessions.<SessionName>.Direct` auf `true` gesetzt ist; der Agent kann nicht gestartet werden.

Bereich: `primary`, `primaryPreferred`, `secondary`, `secondaryPreferred`, `nearest`, `PrimaryMode`, `PrimaryPreferredMode`, `SecondaryMode`, `SecondaryPreferredMode` oder `NearestMode`

Die Werte `primary`, `primaryPreferred`, `secondary`, `secondaryPreferred` und `nearest` werden bevorzugt. Werte, die mit `Mode` enden, werden als Legacy-Kompatibilitätsaliase unterstützt.

Plugins.MongoDB.Default.TLSConnect

Geben Sie den für die Kommunikation zwischen Zabbix Agent 2 und überwachten Datenbanken erforderlichen Verschlüsselungsgrad an.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunktschlüsseln.
 Unterstützt seit Plugin-Version 1.2.1.

Bereich: `required` - TLS-Verbindung erforderlich, `verify_ca` - Zertifikate überprüfen, `verify_full` - Zertifikate und IP-Adresse überprüfen

Plugins.MongoDB.Default.TLSCAFile

Geben Sie den vollständigen Pfadnamen der Datei an, die die Zertifikate der obersten CA(s) für die Überprüfung des Peer-Zertifikats enthält und für verschlüsselte Verbindungen zwischen Zabbix Agent 2 und den überwachten Datenbanken verwendet wird.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunktschlüsseln.

Pflicht: `yes`, wenn `Plugins.MongoDB.Sessions.<SessionName>.TLSConnect` auf `verify_ca` oder `verify_full` gesetzt ist

Plugins.MongoDB.Default.TLSCertFile

Geben Sie den vollständigen Pfadnamen der Datei an, die das Agent-Zertifikat oder die Zertifikatskette enthält und für verschlüsselte Kommunikation zwischen Zabbix Agent 2 und überwachten Datenbanken verwendet wird.
 <SessionName> - Name einer Sitzung für die Verwendung in Datenpunkt-Schlüsseln definieren.

Pflicht: `yes`, wenn `Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile` angegeben ist

Plugins.MongoDB.Default.TLSKeyFile

Geben Sie den vollständigen Pfadnamen der Datei an, die den privaten Schlüssel der Datenbank enthält und für verschlüsselte Kommunikation zwischen Zabbix Agent 2 und überwachten Datenbanken verwendet wird.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunkt-Schlüsseln.

Erforderlich: `yes`, wenn `Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile` angegeben ist

Plugins.MongoDB.Default.Uri

Geben Sie die Standard-URI an.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunkt-Schlüsseln.
 Muss dem URI-Format entsprechen.
 Unterstützte Schemas: `mongodb://` und `mongodb+srv://`; das Schema `tcp://` ist veraltet und wird aus Gründen der Abwärtskompatibilität mit vorhandenen Konfigurationen beibehalten.
 Ein Port kann weggelassen werden (Standard=27017).
Beispiele: “

Standard: `tcp://127.0.0.1:27017`

Beispiele:

```
mongodb://127.0.0.1:27017
mongodb+srv://example.com
localhost
```

Plugins.MongoDB.Default.User

Geben Sie den Standardbenutzernamen an.
 <SessionName> - Namen einer Sitzung für die Verwendung in Datenpunktschlüsseln definieren.

Plugins.MongoDB.KeepAlive

Legt die maximale Wartezeit (in Sekunden) fest, bevor ungenutzte Plugin-Verbindungen geschlossen werden.

Standard: 300
 Bereich: 60-900

Plugins.MongoDB.Sessions.<SessionName>.AuthMechanism

Geben Sie den benannten Sitzungs-Authentifizierungsmechanismus für die Verbindung zum MongoDB-Server an.
 <SessionName> - Name einer Sitzung, die in Datenpunkt-Schlüsseln verwendet wird.

Bereich: `SCRAM-SHA-1`, `SCRAM-SHA-256` oder `MONGODB-X509`

Plugins.MongoDB.Sessions.<SessionName>.AuthSource

Geben Sie die benannte Sitzungsdatenbank für die Authentifizierung der Verbindung zum MongoDB-Server an.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunktschlüsseln.

Standard: `admin`

Wenn der Authentifizierungsmechanismus `MONGODB-X509` verwendet wird, wird dieser Parameter automatisch auf `$external` gesetzt; eine Änderung auf einen anderen Wert führt beim Start zu einem Fehler.

Plugins.MongoDB.Sessions.<SessionName>.Direct

Deaktiviert die automatische Erkennung von MongoDB-Replica-Set-Knoten für die benannte Sitzung.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunkt-Schlüsseln.

Standard: `false`
 Bereich: `true` - nur mit dem angegebenen Server verbinden und andere Mitglieder des Replica Sets ignorieren, `false` - alle Knoten im Replica Set automatisch erkennen.

Wenn das Schema `tcp://` (veraltet) verwendet wird, wird der Parameter standardmäßig auf `true` gesetzt.

Plugins.MongoDB.Sessions.<SessionName>.Password

Geben Sie das Passwort der benannten Sitzung an.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunktschlüsseln.

Plugins.MongoDB.Sessions.<SessionName>.ReplicaSet

Geben Sie den Namen des Replica Sets der benannten Sitzung an, der für die automatische Erkennung von MongoDB-Replica-Set-Knoten verwendet wird.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunkt-Schlüsseln.
 Wenn dieser Parameter mit dem Schema `mongodb+srv://` verwendet wird, verhält er sich so, als wären mehrere URLs angegeben worden; ein DNS-Server kann einen TXT-Eintrag haben, der diesen Parameter (oder einen anderen) überschreiben kann.
 Nicht verfügbar, wenn der Parameter `Plugins.MongoDB.Sessions.<SessionName>.Direct` auf `true` gesetzt ist; der Agent wird dann nicht gestartet.

Plugins.MongoDB.Sessions.<SessionName>.ReadPreference

Geben Sie die benannte Sitzungsreihenfolge für die Verbindung mit MongoDB-Replikatsatzknoten oder den Fallback an.
 <SessionName> - Legt den Namen einer Sitzung fest, die in Datenpunktschlüsseln verwendet wird.
 Nicht verfügbar, wenn der Parameter `Plugins.MongoDB.Sessions.<SessionName>.Direct` auf `true` gesetzt ist; der Agent kann nicht gestartet werden.

Bereich: `primary`, `primaryPreferred`, `secondary`, `secondaryPreferred`, `nearest`, `PrimaryMode`, `PrimaryPreferredMode`, `SecondaryMode`, `SecondaryPreferredMode` oder `NearestMode`

Die Werte `primary`, `primaryPreferred`, `secondary`, `secondaryPreferred` und `nearest` werden bevorzugt. Werte, die mit `Mode` enden, werden als Legacy-Kompatibilitätsaliase unterstützt.

Plugins.MongoDB.Sessions.<SessionName>.TLSConnect

Geben Sie das erforderliche Verschlüsselungsniveau für die Kommunikation zwischen Zabbix Agent 2 und den überwachten Datenbanken für die benannte Sitzung an.
 <SessionName> - legt den Namen einer Sitzung fest, der in Datenpunkt-Schlüsseln verwendet wird.
 Unterstützt seit Plugin-Version 1.2.1.

Bereich: `required` - TLS-Verbindung erforderlich, `verify_ca` - Zertifikate überprüfen, `verify_full` - Zertifikate und IP-Adresse überprüfen

Plugins.MongoDB.Sessions.<SessionName>.TLSCAFile

Geben Sie den vollständigen Pfadnamen der Datei an, die die Zertifikate der obersten CA(s) für die Überprüfung des Peer-Zertifikats enthält und für verschlüsselte Verbindungen zwischen Zabbix Agent 2 und den überwachten Datenbanken für die benannte Sitzung verwendet wird.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunktschlüsseln.

Pflicht: `yes`, wenn `Plugins.MongoDB.Sessions.<SessionName>.TLSConnect` auf `verify_ca` oder `verify_full` gesetzt ist

Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile

Geben Sie den vollständigen Pfadnamen der Datei an, die das Agent-Zertifikat oder die Zertifikatskette enthält und für verschlüsselte Kommunikation zwischen Zabbix Agent 2 und den überwachten Datenbanken für die benannte Sitzung verwendet wird.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunkt-Schlüsseln.

Erforderlich: `yes`, wenn `Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile` angegeben ist

Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile

Geben Sie den vollständigen Pfadnamen der Datei an, die den privaten Schlüssel der Datenbank enthält und für verschlüsselte Kommunikation zwischen Zabbix Agent 2 und überwachten Datenbanken für die benannte Sitzung verwendet wird.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunkt-Schlüsseln.

Pflicht: `yes`, wenn `Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile` angegeben ist

Plugins.MongoDB.Sessions.<SessionName>.Uri

Geben Sie die URI der benannten Sitzung an.
 <SessionName> - Name einer Sitzung für die Verwendung in Datenpunkt-Schlüsseln definieren.
 Muss dem URI-Format entsprechen.
 Unterstützte Schemas: `mongodb://` und `mongodb+srv://`; das Schema `tcp://` ist veraltet und wird aus Gründen der Abwärtskompatibilität mit vorhandenen Konfigurationen beibehalten.
 Ein Port kann weggelassen werden (Standard=27017).
Beispiele: “

Standard: `tcp://127.0.0.1:27017`

Beispiele:

```
mongodb://127.0.0.1:27017
mongodb+srv://example.com
localhost
```

Plugins.MongoDB.Sessions.<SessionName>.User

Geben Sie den Benutzernamen der benannten Sitzung an.
 <SessionName> - Definieren Sie den Namen einer Sitzung zur Verwendung in Datenpunkt-Schlüsseln.

Plugins.MongoDB.System.Path

Geben Sie den Pfad zur ausführbaren Datei des MongoDB-Plugins an.

Beispielverwendung: `Plugins.MongoDB.System.Path=/usr/libexec/zabbix/zabbix-agent2-plugin-mongodb`

Plugins.MongoDB.Timeout

Legt das Timeout für die Ausführung von Anfragen fest (die Dauer in Sekunden, die auf den Abschluss einer Anfrage gewartet wird, bevor sie beendet wird).

Standard: `global timeout<br>` Bereich: 1-30

Optionen

Parameter	Beschreibung
-V --version	Gibt die Plugin-Version und Lizenzinformationen aus.
-h --help	Gibt Hilfsinformationen aus (Kurzform).
-t, --test <item key>	Startet das Plugin zum Testen (Plugin-Konfiguration wird ignoriert).

Siehe auch

- Beschreibung der allgemeinen Konfigurationsparameter von Zabbix Agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Anweisungen zum Konfigurieren von Plugins

MQTT-Plugin

Übersicht

Konfigurationsparameter ermöglichen die Anpassung des Verhaltens des MQTT-Plugins.

Es wird empfohlen, sie in einer eigenen Konfigurationsdatei (z. B. `mqtt.conf`) anzugeben, statt in der Konfigurationsdatei von Agent 2. Verwenden Sie dann die Direktive Include, um diese Datei in die Konfiguration von Zabbix Agent 2 einzubinden.

Alle Parameter sind **optional**, sofern nicht ausdrücklich angegeben ist, dass der Parameter obligatorisch ist.

Diese Seite ist wie folgt aufgebaut:

- Parameterübersicht (um die vollständigen Details eines Parameters anzuzeigen, klicken Sie auf seinen Namen)
- Parameterdetails

Parameter	Beschreibung
Plugins.MQTT.Default.Password	Geben Sie das Standardpasswort für die Verbindung zu MQTT an.
Plugins.MQTT.Default.TLSCAFile	Geben Sie den vollständigen Pfadnamen der Datei an, die Zertifikate der obersten CA(s) für die Überprüfung des Peer-Zertifikats enthält.
Plugins.MQTT.Default.TLSCertFile	Geben Sie den vollständigen Pfadnamen der Datei an, die das Agent-Zertifikat oder die Zertifikatskette enthält.
Plugins.MQTT.Default.TLSKeyFile	Geben Sie den vollständigen Pfadnamen der Datei an, die den privaten MQTT-Schlüssel enthält.
Plugins.MQTT.Default.Topic	Geben Sie das Standard-Topic für das MQTT-Abonnement an.
Plugins.MQTT.Default.Url	Geben Sie die Standard-Verbindungszeichenfolge des MQTT-Brokers an.
Plugins.MQTT.Default.User	Geben Sie den Standardbenutzernamen für die Verbindung zu MQTT an.
Plugins.MQTT.Sessions.<SessionName>.Password	Geben Sie das Passwort der benannten Sitzung für die Verbindung zu MQTT an.
Plugins.MQTT.Sessions.<SessionName>.TLSCAFile	Geben Sie den vollständigen Pfadnamen der Datei an, die Zertifikate der obersten CA(s) für die Überprüfung des Peer-Zertifikats für die benannte Sitzung enthält.
Plugins.MQTT.Sessions.<SessionName>.TLSCertFile	Geben Sie den vollständigen Pfadnamen der Datei an, die das Agent-Zertifikat oder die Zertifikatskette für die benannte Sitzung enthält.
Plugins.MQTT.Sessions.<SessionName>.TLSKeyFile	Geben Sie den vollständigen Pfadnamen der Datei an, die den privaten MQTT-Schlüssel für die benannte Sitzung enthält.
Plugins.MQTT.Sessions.<SessionName>.Topic	Geben Sie das Topic der benannten Sitzung für das MQTT-Abonnement an.
Plugins.MQTT.Sessions.<SessionName>.Url	Geben Sie die Verbindungszeichenfolge des MQTT-Brokers für die benannte Sitzung an.
Plugins.MQTT.Sessions.<SessionName>.User	Geben Sie den Benutzernamen der benannten Sitzung für die Verbindung zu MQTT an.

Beachten Sie:

- Die Standardwerte spiegeln die Prozess-Standardwerte wider, nicht die Werte in den mitgelieferten Konfigurationsdateien;
- Werte unterstützen Umgebungsvariablen;
- Zabbix unterstützt Konfigurationsdateien nur in UTF-8-Kodierung ohne BOM;

- Kommentare, die mit ”#” beginnen, werden nur am Zeilenanfang unterstützt.

Parameterdetails

Plugins.MQTT.Default.Password

Geben Sie das Standardpasswort für die Verbindung zu MQTT an.
 Wird verwendet, wenn in einem Datenpunkt-Schlüssel oder einer benannten Sitzung kein Wert angegeben ist.

Plugins.MQTT.Default.TLSCAFile

Geben Sie den vollständigen Pfadnamen der Datei an, die die CA-Zertifikate der obersten Ebene für die Überprüfung des Peer-Zertifikats bei verschlüsselter Kommunikation zwischen Zabbix Agent 2 und dem MQTT-Broker enthält.
 Wird verwendet, wenn in einer benannten Sitzung kein Wert angegeben ist.

Plugins.MQTT.Default.TLSCertFile

Geben Sie den vollständigen Pfadnamen der Datei an, die das Agent-Zertifikat oder die Zertifikatskette für verschlüsselte Kommunikation zwischen Zabbix Agent 2 und dem MQTT-Broker enthält.
 Wird verwendet, wenn in einer benannten Sitzung kein Wert angegeben ist.

Plugins.MQTT.Default.TLSKeyFile

Geben Sie den vollständigen Pfadnamen der Datei an, die den privaten MQTT-Schlüssel für die verschlüsselte Kommunikation zwischen Zabbix Agent 2 und dem MQTT-Broker enthält.
 Wird verwendet, wenn in einer benannten Sitzung kein Wert angegeben ist.

Plugins.MQTT.Default.Topic

Geben Sie das Standard-Topic für das MQTT-Subscription an.
 Wird verwendet, wenn im Datenpunkt-Schlüssel oder in der benannten Sitzung kein Wert angegeben ist.
 Das Topic kann Platzhalter enthalten (”+”,”#”).

Beispiele:

```
path/to/file
path/to/#
path/+/topic
```

Plugins.MQTT.Default.Url

Geben Sie die Standard-Verbindungszeichenfolge für den MQTT-Broker an.
 Wird verwendet, wenn in einem Datenpunkt-Schlüssel oder einer benannten Sitzung kein Wert angegeben ist.
 Sollte keine Abfrageparameter enthalten.
 Muss dem URL-Format entsprechen.
 Unterstützte Schemas: `tcp` (Standard), `ws`, `tls`
 Ein Schema kann weggelassen werden.
 Ein Port kann weggelassen werden (Standard=1883)

Standard: `tcp://localhost:1883`

Beispiele:

```
tcp://host:1883
localhost
ws://host:8080
```

Plugins.MQTT.Default.User

Geben Sie den Standard-Benutzernamen für die Verbindung zu MQTT an.
 Wird verwendet, wenn im Datenpunkt-Schlüssel oder in einer benannten Sitzung kein Wert angegeben ist.

Plugins.MQTT.Sessions.<SessionName>.Password

Geben Sie das Passwort der benannten Sitzung für die Verbindung zu MQTT an.
 <SessionName> - Name einer Sitzung, der in Datenpunkt-Schlüsseln verwendet wird.

Plugins.MQTT.Sessions.<SessionName>.TLSCAFile

Geben Sie den vollständigen Pfadnamen der Datei an, die die Zertifikate der obersten CA(s) für die Überprüfung des Peer-Zertifikats für die benannte Sitzung enthält und für verschlüsselte Kommunikation zwischen Zabbix Agent 2 und dem MQTT-Broker verwendet wird.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunkt-Schlüsseln.

Plugins.MQTT.Sessions.<SessionName>.TLSCertFile

Geben Sie den vollständigen Pfad zur Datei an, die das Agent-Zertifikat oder die Zertifikatskette für die benannte Sitzung enthält. Diese wird für verschlüsselte Kommunikation zwischen Zabbix Agent 2 und dem MQTT-Broker verwendet.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunkt-Schlüsseln.

Plugins.MQTT.Sessions.<SessionName>.TLSKeyFile

Geben Sie den vollständigen Pfadnamen der Datei an, die den MQTT-privaten Schlüssel für die benannte Sitzung enthält und für verschlüsselte Kommunikation zwischen Zabbix Agent 2 und dem MQTT-Broker verwendet wird.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunkt-Schlüsseln.

Plugins.MQTT.Sessions.<SessionName>.Topic

Geben Sie das benannte Sitzungsthema für das MQTT-Subscription an.
 <SessionName> - Name einer Sitzung für die Verwendung in Datenpunkt-Schlüsseln definieren.
 Das Thema kann Platzhalter ("+","#") enthalten.

Beispiele:

```
path/to/file
path/to/#
path/+/topic
```

Plugins.MQTT.Sessions.<SessionName>.Url

Geben Sie die Verbindungszeichenfolge des MQTT-Brokers für die benannte Sitzung an.
 <SessionName> - Name einer Sitzung definieren, der in Datenpunkt-Schlüsseln verwendet wird.
 Darf keine Abfrageparameter enthalten.
 Muss dem URL-Format entsprechen.
 Unterstützte Schemas: `tcp` (Standard), `ws`, `tls`
 Ein Schema kann weggelassen werden.
 Ein Port kann weggelassen werden (Standard=1883)

Standard: `tcp://localhost:1883`

Beispiele:

```
tcp://host:1883
localhost
ws://host:8080
```

Plugins.MQTT.Sessions.<SessionName>.User

Geben Sie den Benutzernamen der benannten Sitzung für die Verbindung zu MQTT an.
 <SessionName> - Name einer Sitzung definieren, die in Datenpunkt-Schlüsseln verwendet wird.

Wenn Sie das Timeout für die Ausführung von Anfragen festlegen müssen (also wie lange auf den Abschluss einer Anfrage gewartet werden soll, bevor sie beendet wird), verwenden Sie das Formular Datenpunktkonfiguration.

Siehe auch

- Beschreibung der allgemeinen Konfigurationsparameter von Zabbix Agent 2: Zabbix Agent 2 (UNIX) / Zabbix Agent 2 (Windows)
- Anweisungen zum Konfigurieren von Plugins

MSSQL-Plugin

Übersicht

Konfigurationsparameter ermöglichen die Anpassung des Verhaltens des MSSQL-Plugins.

Es wird empfohlen, sie in einer eigenen Konfigurationsdatei anzugeben (z. B. `mssql.conf`) und nicht in der Konfigurationsdatei von Agent 2. Verwenden Sie dann die Direktive Include, um diese Datei in die Konfiguration von Zabbix Agent 2 einzubinden.

Das MSSQL-Plugin ist ein ladbares Plugin und ist im Repository des MSSQL-Plugins verfügbar und dort vollständig beschrieben.

Alle Parameter sind **optional**, sofern nicht ausdrücklich angegeben ist, dass der Parameter obligatorisch ist.

Diese Seite ist wie folgt aufgebaut:

- Parameterübersicht (um die vollständigen Details eines Parameters anzuzeigen, klicken Sie auf seinen Namen)
- Parameterdetails

Parameter	Beschreibung
Plugins.MSSQL.CustomQueriesDir	Geben Sie den Pfad zu dem Verzeichnis an, das benutzerdefinierte `.sql`-Dateien mit benutzerdefinierten Abfragen enthält, die das Plugin ausführen kann.
Plugins.MSSQL.CustomQueriesEnabled	Aktivieren Sie die Ausführung des Datenpunkt-Schlüssels `mssql.custom.query`.
Plugins.MSSQL.Default.CACertPath	Geben Sie standardmäßig den Dateipfad zum öffentlichen Schlüsselzertifikat der Zertifizierungsstelle (CA) an, die das Zertifikat des MSSQL-Servers ausgestellt hat.
Plugins.MSSQL.Default.Database	Geben Sie standardmäßig den Datenbanknamen an.
Plugins.MSSQL.Default.Encrypt	Geben Sie den Standardwert für die Verschlüsselungsstufe an.
Plugins.MSSQL.Default.HostnameInCertificate	Geben Sie standardmäßig den Common Name (CN) des Zertifikats des MSSQL-Servers an.

Parameter	Beschreibung
Plugins.MSSQL.Default.Password	Geben Sie das Standardpasswort an.
Plugins.MSSQL.Default.TLSMinVersion	Geben Sie standardmäßig die minimale TLS-Version an.
Plugins.MSSQL.Default.TrustServerCertificate	Vertrauen Sie dem Serverzertifikat standardmäßig, ohne es zu validieren.
Plugins.MSSQL.Default.Uri	Geben Sie die Standard-URI an.
Plugins.MSSQL.Default.User	Geben Sie den Standardbenutzernamen an.
Plugins.MSSQL.KeepAlive	Geben Sie die maximale Wartezeit (in Sekunden) an, bevor ungenutzte Plugin-Verbindungen geschlossen werden.
Plugins.MSSQL.Sessions.<SessionName>.CACertPath	Geben Sie den Dateipfad der benannten Sitzung zum öffentlichen Schlüsselzertifikat der Zertifizierungsstelle (CA) an, die das Zertifikat des MSSQL-Servers ausgestellt hat.
Plugins.MSSQL.Sessions.<SessionName>.Database	Geben Sie den Datenbanknamen der benannten Sitzung an.
Plugins.MSSQL.Sessions.<SessionName>.Encrypt	Geben Sie die Verschlüsselungsstufe der benannten Sitzung an.
Plugins.MSSQL.Sessions.<SessionName>.HostNameInCertificate	Geben Sie den Common Name (CN) des Zertifikats des MSSQL-Servers für die benannte Sitzung an.
Plugins.MSSQL.Sessions.<SessionName>.Password	Geben Sie das Passwort der benannten Sitzung an.
Plugins.MSSQL.Sessions.<SessionName>.TLSMinVersion	Geben Sie die minimale TLS-Version der benannten Sitzung an.
Plugins.MSSQL.Sessions.<SessionName>.TrustServerCertificate	Vertrauen Sie dem Serverzertifikat der benannten Sitzung, ohne es zu validieren.
Plugins.MSSQL.Sessions.<SessionName>.Uri	Geben Sie die URI der benannten Sitzung an.
Plugins.MSSQL.Sessions.<SessionName>.User	Geben Sie den Benutzernamen der benannten Sitzung an.
Plugins.MSSQL.System.Path	Geben Sie den Pfad zur ausführbaren Datei des MSSQL-Plugins an.
Plugins.MSSQL.Timeout	Legen Sie die Dauer in Sekunden fest, die beim ersten Verbindungsaufbau und bei nachfolgenden Vorgängen in der Sitzung auf eine Antwort des Servers gewartet wird.

Beachten Sie:

- Die Standardwerte spiegeln die Prozess-Standardeinstellungen wider, nicht die Werte in den mitgelieferten Konfigurationsdateien.
- Werte unterstützen Umgebungsvariablen;
- Zabbix unterstützt Konfigurationsdateien nur in UTF-8-Kodierung ohne BOM.
- Kommentare, die mit "#" beginnen, werden nur am Zeilenanfang unterstützt.

Parameterdetails

Plugins.MSSQL.CustomQueriesDir

Geben Sie den Pfad zum Verzeichnis an, das benutzerdefinierte `.sql`-Dateien mit benutzerdefinierten Abfragen enthält, die das Plugin ausführen kann.
 Das Plugin lädt beim Start alle verfügbaren .sql-Dateien im konfigurierten Verzeichnis. Das bedeutet, dass Änderungen an den Dateien mit benutzerdefinierten Abfragen erst nach einem Neustart des Plugins wirksam werden. Das Plugin wird zusammen mit Zabbix Agent 2 gestartet und gestoppt.

Standardwerte:
`/usr/local/share/zabbix/custom-queries/mssql` - für Unix-Systeme,
`*:\Program Files\Zabbix Agent 2\Custom Queries\MSSQL` - für Windows-Systeme (wobei `*` der Laufwerksname aus der Umgebungsvariable `ProgramFiles` ist)

Plugins.MSSQL.CustomQueriesEnabled

Aktiviert die Ausführung des Datenpunkt-Schlüssels `mssql.custom.query`.
 Wenn diese Option deaktiviert ist, werden keine Abfragen aus dem Pfad des Verzeichnisses für benutzerdefinierte Abfragen geladen.

Standard: `false`
 Bereich: `true/false`

Plugins.MSSQL.Default.CACertPath

Geben Sie den Standard-Dateipfad zum öffentlichen Schlüsselzertifikat der Zertifizierungsstelle (CA) an, die das Zertifikat des MSSQL-Servers ausgestellt hat.
 Das Zertifikat muss im PEM-Format vorliegen.

Plugins.MSSQL.Default.Database

Geben Sie den Namen der Standarddatenbank an.

Plugins.MSSQL.Default.Encrypt

Geben Sie den Standardverschlüsselungsgrad an.

Bereich:
true - die Datenübertragung zwischen Plugin und Server ist verschlüsselt;
false - die Datenübertragung zwischen Plugin und Server ist über das Login-Paket hinaus nicht verschlüsselt;
strict - die Datenübertragung zwischen Plugin und Server ist Ende-zu-Ende verschlüsselt mit TDS8;
disable - die Datenübertragung zwischen Plugin und Server ist nicht verschlüsselt.

Plugins.MSSQL.Default.HostNameInCertificate

Geben Sie standardmäßig den Common Name (CN) des MSSQL-Serverzertifikats an.

Plugins.MSSQL.Default.Password

Geben Sie das Standardkennwort für einen geschützten MSSQL-Server an.

Plugins.MSSQL.Default.TLSMinVersion

Geben Sie standardmäßig die minimale TLS-Version an.

Bereich: `1.0`, `1.1`, `1.2`, `1.3`

Plugins.MSSQL.Default.TrustServerCertificate

Das Serverzertifikat standardmäßig vertrauen, ohne es zu validieren.

Bereich: `true`, `false`

Plugins.MSSQL.Default.Uri

Geben Sie die Standard-URI für die Verbindung mit dem MSSQL-Server an.
 Eingebettete Anmeldedaten werden ignoriert.
 Das einzige unterstützte Schema ist `sqlserver://`.
 Das Schema kann weggelassen werden.

Standard: `sqlserver://localhost:1433`

Plugins.MSSQL.Default.User

Geben Sie den Standard-Benutzernamen für einen geschützten MSSQL-Server an.

Plugins.MSSQL.KeepAlive

Geben Sie die maximale Wartezeit (in Sekunden) an, bevor ungenutzte Plugin-Verbindungen geschlossen werden.

Standard: 300
 Bereich: 60–900

Plugins.MSSQL.Sessions.<SessionName>.CACertPath

Geben Sie den Dateipfad der benannten Sitzung zur öffentlichen Schlüsselzertifikatsdatei der Zertifizierungsstelle (CA) an, die das Zertifikat des MSSQL-Servers ausgestellt hat.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunktschlüsseln.
 Das Zertifikat muss im PEM-Format vorliegen.

Plugins.MSSQL.Sessions.<SessionName>.Database

Geben Sie den Namen der Datenbank für die benannte Sitzung an.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunktschlüsseln.

Plugins.MSSQL.Sessions.<SessionName>.Encrypt

Gibt den Verschlüsselungsgrad der benannten Sitzung an.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunkt-Schlüsseln.

Bereich:
true - die Datenübertragung zwischen Plugin und Server ist verschlüsselt;
false - die Datenübertragung zwischen Plugin und Server ist über das Login-Paket hinaus nicht verschlüsselt;
strict - die Datenübertragung zwischen Plugin und Server ist Ende-zu-Ende verschlüsselt unter Verwendung von TDS8;
disable - die Datenübertragung zwischen Plugin und Server ist nicht verschlüsselt.

Plugins.MSSQL.Sessions.<SessionName>.HostNameInCertificate

Geben Sie den Common Name (CN) des MSSQL-Serverzertifikats für die benannte Sitzung an.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunkt-Schlüsseln.

Plugins.MSSQL.Sessions.<SessionName>.Password

Geben Sie das Passwort der benannten Sitzung für einen geschützten MSSQL-Server an.
 <SessionName> - Name einer Sitzung, die in Datenpunktschlüsseln verwendet wird.

Plugins.MSSQL.Sessions.<SessionName>.TLSMinVersion

Geben Sie die minimale TLS-Version für die benannte Sitzung an.
 <SessionName> - Name einer Sitzung, die in Datenpunkt-Schlüsseln verwendet wird.

Bereich: `1.0`, `1.1`, `1.2`, `1.3`

Plugins.MSSQL.Sessions.<SessionName>.TrustServerCertificate

Dem Serverzertifikat der benannten Sitzung vertrauen, ohne es zu validieren.
 <SessionName> - Namen einer Sitzung zur Verwendung in Datenpunkt-Schlüsseln definieren.

Bereich: `true`, `false`

Plugins.MSSQL.Sessions.<SessionName>.Uri

Geben Sie die URI der benannten Sitzung für die Verbindung zum MSSQL-Server an.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunkt-Schlüsseln.
 Eingebettete Anmeldedaten werden ignoriert.
 Das einzige unterstützte Schema ist `sqlserver://`.
 Das Schema kann weggelassen werden.

Standard: `sqlserver://localhost:1433`

Plugins.MSSQL.Sessions.<SessionName>.User

Geben Sie den Benutzernamen der benannten Sitzung für einen geschützten MSSQL-Server an.
 <SessionName> - Name einer Sitzung, die in Datenpunkt-Schlüsseln verwendet wird.

Plugins.MSSQL.System.Path

Geben Sie den Pfad zur ausführbaren Datei des MSSQL-Plugins an.
 Globale Einstellung für das MSSQL-Plugin. Wird auf alle Verbindungen angewendet.

Beispiel: `Plugins.MSSQL.System.Path=/usr/libexec/zabbix/zabbix-agent2-plugin-mssql`

Plugins.MSSQL.Timeout

Legt die Dauer in Sekunden fest, die beim ersten Verbindungsaufbau und bei nachfolgenden Vorgängen in der Sitzung auf eine Antwort des Servers gewartet wird.

Standard: `global timeout`
 Bereich: 1-30

Optionen

Parameter	Beschreibung
-V --version	Gibt die Plugin-Version und Lizenzinformationen aus.
-h --help	Gibt Hilfsinformationen aus (Kurzform).
-t, --test <item key>	Startet das Plugin zum Testen (Plugin-Konfiguration wird ignoriert).

Siehe auch

- Beschreibung der allgemeinen Konfigurationsparameter von Zabbix Agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Anweisungen zum Konfigurieren von Plugins

MySQL-Plugin

Übersicht

Konfigurationsparameter ermöglichen die Anpassung des Verhaltens des MySQL-Plugins.

Es wird empfohlen, sie in einer eigenen Konfigurationsdatei anzugeben (z. B. `mysql.conf`) und nicht in der Konfigurationsdatei von Agent 2. Verwenden Sie dann die Direktive Include, um diese Datei in die Konfiguration von Zabbix Agent 2 einzubinden.

Alle Parameter sind **optional**, sofern nicht ausdrücklich angegeben ist, dass der Parameter obligatorisch ist.

Diese Seite ist wie folgt aufgebaut:

- Parameterübersicht (um die vollständigen Details eines Parameters anzuzeigen, klicken Sie auf seinen Namen)
- Parameterdetails

Parameter	Beschreibung
Plugins.Mysql.CallTimeout	Geben Sie die maximale Zeit in Sekunden an, die auf die Ausführung einer Anfrage gewartet werden soll.
Plugins.Mysql.CustomQueriesPath	Geben Sie den vollständigen Pfad des Verzeichnisses an, das `.sql`-Dateien mit benutzerdefinierten Abfragen enthält.
Plugins.Mysql.CustomQueriesEnabled	Aktivieren Sie die Ausführung des Datenpunkt-Schlüssels `mysql.custom.query`.
Plugins.Mysql.Default.Password	Geben Sie das Standardpasswort an.
Plugins.Mysql.Default.TLSCipher	Geben Sie den Standard-Verschlüsselungsgrad für die Kommunikation zwischen Zabbix Agent 2 und den überwachten Datenbanken an.
Plugins.Mysql.Default.TLSCAFile	Geben Sie den vollständigen Pfad der Datei an, die Zertifikate der obersten CA(s) für die Überprüfung des Peer-Zertifikats enthält.
Plugins.Mysql.Default.TLSCertFile	Geben Sie den vollständigen Pfad der Datei an, die das Zertifikat oder die Zertifikatskette des Agenten enthält.

Parameter	Beschreibung
Plugins.Mysql.Default.TLSKeyFile	Geben Sie den vollständigen Pfad der Datei an, die den privaten Schlüssel der Datenbank enthält.
Plugins.Mysql.Default.Uri	Geben Sie die Standard-URI an.
Plugins.Mysql.Default.User	Geben Sie den Standardbenutzernamen an.
Plugins.Mysql.KeepAlive	Geben Sie die maximale Wartezeit in Sekunden an, bevor ungenutzte Plugin-Verbindungen geschlossen werden.
Plugins.Mysql.Sessions.<SessionName>.Password	Geben Sie das Passwort der benannten Sitzung an.
Plugins.Mysql.Sessions.<SessionName>.TLSConnect	Geben Sie den Verschlüsselungsgrad der benannten Sitzung für die Kommunikation zwischen Zabbix Agent 2 und den überwachten Datenbanken an.
Plugins.Mysql.Sessions.<SessionName>.TLSCAFile	Geben Sie den vollständigen Pfad der Datei an, die Zertifikate der obersten CA(s) für die Überprüfung des Peer-Zertifikats der benannten Sitzung enthält.
Plugins.Mysql.Sessions.<SessionName>.TLSCertFile	Geben Sie den vollständigen Pfad der Datei an, die das Zertifikat oder die Zertifikatskette der benannten Sitzung des Agenten enthält.
Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile	Geben Sie den vollständigen Pfad der Datei an, die den privaten Schlüssel der Datenbank der benannten Sitzung enthält.
Plugins.Mysql.Sessions.<SessionName>.Uri	Geben Sie die URI der benannten Sitzung an.
Plugins.Mysql.Sessions.<SessionName>.User	Geben Sie den Benutzernamen der benannten Sitzung an.
Plugins.Mysql.Timeout	Legen Sie die maximale Wartezeit in Sekunden fest, wenn eine Verbindung hergestellt werden muss.

Beachten Sie:

- Die Standardwerte spiegeln die Prozess-Standardeinstellungen wider, nicht die Werte in den mitgelieferten Konfigurationsdateien;
- Werte unterstützen Umgebungsvariablen;
- Zabbix unterstützt Konfigurationsdateien nur in UTF-8-Kodierung ohne BOM;
- Kommentare, die mit ”#” beginnen, werden nur am Zeilenanfang unterstützt.

Parameterdetails

Plugins.Mysql.CallTimeout

Geben Sie die maximale Wartezeit in Sekunden an, bis eine Anfrage abgeschlossen ist.

Standard: `global timeout`
 Bereich: 1-30

Plugins.Mysql.CustomQueriesPath

Geben Sie den vollständigen Pfadnamen des Verzeichnisses an, das `.sql`-Dateien mit benutzerdefinierten Abfragen enthält.

Standardwerte:
`/usr/local/share/zabbix/custom-queries/mysql` - für Unix-Systeme,
`*:\Program Files\Zabbix Agent 2\Custom Queries\Mysql` - für Windows-Systeme (wobei `*` der Laufwerksname aus der Umgebungsvariable `ProgramFiles` ist)

Plugins.Mysql.CustomQueriesEnabled

Die Ausführung des Datenpunkt-Schlüssels `mysql.custom.query` aktivieren.
 Wenn diese Option deaktiviert ist, werden keine Abfragen aus dem Verzeichnispfad für benutzerdefinierte Abfragen geladen.

Standard: `false`
 Bereich: `true`/`false`

Plugins.Mysql.Default.Password

Geben Sie das Standardpasswort für einen geschützten MySQL-Server an.

Plugins.Mysql.Default.TLSConnect

Geben Sie die standardmäßige Verschlüsselungsstufe für die Kommunikation zwischen Zabbix Agent 2 und den überwachten Datenbanken an.

Bereich:
required - TLS-Verbindung erforderlich;
verify_ca - Zertifikate überprüfen;
verify_full - Zertifikate und IP-Adresse überprüfen

Plugins.Mysql.Default.TLSCAFile

Geben Sie den vollständigen Pfadnamen der Datei an, die die Zertifikate der obersten CA(s) für die Überprüfung des Peer-Zertifikats enthält und für verschlüsselte Verbindungen zwischen Zabbix Agent 2 und den überwachten Datenbanken verwendet wird.

Pflicht: `yes`, wenn `Plugins.Mysql.Default.TLSConnect` auf `verify_ca` oder `verify_full` gesetzt ist

Plugins.Mysql.Default.TLSCertFile

Geben Sie den vollständigen Pfadnamen der Datei an, die das Agent-Zertifikat oder die Zertifikatskette enthält und für verschlüsselte Kommunikation zwischen Zabbix Agent 2 und den überwachten Datenbanken verwendet wird.

Erforderlich: `yes`, wenn `Plugins.Mysql.Default.TLSKeyFile` angegeben ist

Plugins.Mysql.Default.TLSKeyFile

Geben Sie den vollständigen Pfadnamen der Datei an, die den privaten Schlüssel der Datenbank enthält und für verschlüsselte Kommunikation zwischen Zabbix Agent 2 und den überwachten Datenbanken verwendet wird.

Erforderlich: `yes`, wenn `Plugins.Mysql.Default.TLSCertFile` angegeben ist

Plugins.Mysql.Default.Uri

Geben Sie die Standard-URI für die Verbindung zum MySQL-Server an.
 Eingebettete Anmeldedaten werden ignoriert.
 Unterstützte Schemas: `tcp`, `unix`
 Das Schema kann weggelassen werden.
 Der Port kann weggelassen werden (Standard=3306).

Beispiele:

```
tcp://localhost:3306
localhost
unix:/var/run/mysql.sock
```

Plugins.Mysql.Default.User

Geben Sie den Standard-Benutzernamen für einen geschützten MySQL-Server an.

Plugins.Mysql.KeepAlive

Geben Sie die maximale Wartezeit (in Sekunden) an, bevor ungenutzte Plugin-Verbindungen geschlossen werden.

Standard: 300
 Bereich: 60−900

Plugins.Mysql.Sessions.<SessionName>.Password

Geben Sie das Passwort für die benannte Sitzung für einen geschützten MySQL-Server an.
 <SessionName> - Name einer Sitzung, die in Datenpunkt-Schlüsseln verwendet wird.

Plugins.Mysql.Sessions.<SessionName>.TLSConnect

Geben Sie den Verschlüsselungsgrad für benannte Sitzungen für die Kommunikation zwischen Zabbix Agent 2 und überwachten Datenbanken an.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunkt-Schlüsseln.

Bereich:
required - TLS-Verbindung erforderlich;
verify_ca - Zertifikate überprüfen;
verify_full - Zertifikate und IP-Adresse überprüfen

Plugins.Mysql.Sessions.<SessionName>.TLSCAFile

Geben Sie den vollständigen Pfadnamen der Datei an, die die CA-Zertifikate der obersten Ebene für die Peer-Zertifikatsprüfung der benannten Sitzung enthält und für verschlüsselte Verbindungen zwischen Zabbix Agent 2 und den überwachten Datenbanken verwendet wird.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunkt-Schlüsseln.

Pflicht: `yes`, wenn `Plugins.Mysql.Sessions.<SessionName>.TLSConnect` auf `verify_ca` oder `verify_full` gesetzt ist

Plugins.Mysql.Sessions.<SessionName>.TLSCertFile

Geben Sie den vollständigen Pfadnamen der Datei an, die das Zertifikat oder die Zertifikatskette des benannten Session-Agent enthält und für verschlüsselte Kommunikation zwischen Zabbix Agent 2 und den überwachten Datenbanken verwendet wird.
 <SessionName> - definiert den Namen einer Session zur Verwendung in Datenpunktschlüsseln.

Pflicht: `yes`, wenn `Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile` angegeben ist

Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile

Geben Sie den vollständigen Pfadnamen der Datei an, die den privaten Schlüssel der benannten Sitzung für die Datenbank enthält und für verschlüsselte Kommunikation zwischen Zabbix Agent 2 und den überwachten Datenbanken verwendet wird.
 <SessionName> - Definieren Sie den Namen einer Sitzung zur Verwendung in Datenpunktschlüsseln.

Erforderlich: `yes`, wenn `Plugins.Mysql.Sessions.<SessionName>.TLSCertFile` angegeben ist

Plugins.Mysql.Sessions.<SessionName>.Uri

Geben Sie die URI der benannten Sitzung für die Verbindung zum MySQL-Server an.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunkt-Schlüsseln.
 Eingebettete Anmeldedaten werden ignoriert.
 Unterstützte Schemata: `tcp`, `unix`
 Das Schema kann weggelassen werden.
 Der Port kann weggelassen werden (Standard=3306).

Beispiele:

```
tcp://localhost:3306
localhost
unix:/var/run/mysql.sock
```

Plugins.Mysql.Sessions.<SessionName>.User

Geben Sie den Benutzernamen der benannten Sitzung für einen geschützten MySQL-Server an.
 <SessionName> - Name einer Sitzung zur Verwendung in Datenpunktschlüsseln definieren.

Plugins.Mysql.Timeout

Legt die maximale Wartezeit in Sekunden fest, wenn eine Verbindung hergestellt werden muss.

Standard: `global timeout`
 Bereich: 1-30

Siehe auch

- Beschreibung der allgemeinen Konfigurationsparameter von Zabbix Agent 2: Zabbix Agent 2 (UNIX) / Zabbix Agent 2 (Windows)
- Anweisungen zum Konfigurieren von Plugins

NVIDIA GPU-Plugin

Übersicht

Konfigurationsparameter ermöglichen die Anpassung des Verhaltens des NVIDIA GPU-Plugins.

Es wird empfohlen, sie in einer eigenen Konfigurationsdatei (z. B. `nvidiagpu.conf`) anzugeben, statt in der Konfigurationsdatei von Agent 2. Verwenden Sie dann die Direktive Include, um diese Datei in die Zabbix Agent 2-Konfiguration einzubinden.

Das NVIDIA GPU-Plugin ist ein ladbares Plugin und ist im Repository des NVIDIA GPU-Plugins verfügbar und dort vollständig beschrieben.

Alle Parameter sind **optional**, sofern nicht ausdrücklich angegeben ist, dass der Parameter obligatorisch ist.

Diese Seite ist wie folgt aufgebaut:

- Parameterübersicht (um die vollständigen Details eines Parameters anzuzeigen, klicken Sie auf seinen Namen)
- Parameterdetails

Parameter	Beschreibung
Plugins.NVIDIA.System.Path	Geben Sie den Pfad zur ausführbaren Datei des NVIDIA GPU-Plugins an.
Plugins.NVIDIA.Timeout	Geben Sie die maximale Zeit (in Sekunden) an, die bei Verbindungsversuchen und nachfolgenden Vorgängen in der Sitzung auf eine Server-Antwort gewartet werden soll.

Beachten Sie:

- Die Standardwerte spiegeln die Prozess-Standardeinstellungen wider, nicht die Werte in den mitgelieferten Konfigurationsdateien;
- Werte unterstützen Umgebungsvariablen;
- Zabbix unterstützt Konfigurationsdateien nur in UTF-8-Kodierung ohne BOM;
- Kommentare, die mit "#" beginnen, werden nur am Zeilenanfang unterstützt.

Parameterdetails

Plugins.NVIDIA.System.Path

Geben Sie den Pfad zur ausführbaren Datei des NVIDIA-GPU-Plugins an.

Beispiel:

```
Plugins.NVIDIA.System.Path=/path/to/executable/nvidia
```

Plugins.NVIDIA.Timeout

Geben Sie die maximale Zeit (in Sekunden) an, die bei Verbindungsversuchen und nachfolgenden Vorgängen in der Sitzung auf eine Server-Antwort gewartet wird.
 Der globale Datenpunkt-Typ-Timeout oder der individuelle Datenpunkt-Timeout überschreibt diesen Wert, wenn er größer ist.

Standard: `global timeout`
 Bereich: 1-30

Siehe auch

- Beschreibung der allgemeinen Konfigurationsparameter von Zabbix Agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Anweisungen zum Konfigurieren von Plugins

Oracle-Plugin

Übersicht

Konfigurationsparameter ermöglichen die Anpassung des Verhaltens des Oracle-Plugins.

Es wird empfohlen, sie in einer eigenen Konfigurationsdatei (z. B. `oracle.conf`) statt in der Konfigurationsdatei von Agent 2 anzugeben. Verwenden Sie dann die Direktive Include, um diese Datei in die Konfiguration von Zabbix Agent 2 einzubinden.

Alle Parameter sind **optional**, sofern nicht ausdrücklich angegeben ist, dass der Parameter obligatorisch ist.

Diese Seite ist wie folgt aufgebaut:

- Parameterübersicht (um die vollständigen Details eines Parameters anzuzeigen, klicken Sie auf seinen Namen)
- Parameterdetails

Parameter	Beschreibung
Plugins.Oracle.CallTimeout	Die maximale Wartezeit in Sekunden, bis eine Anfrage abgeschlossen ist. Dieser Parameter hat derzeit keine Wirkung; stattdessen wird die Einstellung item timeout verwendet.
Plugins.Oracle.ConnectTimeout	Geben Sie die maximale Wartezeit in Sekunden an, bis eine Verbindung hergestellt ist.
Plugins.Oracle.CustomQueriesPath	Geben Sie den vollständigen Pfad des Verzeichnisses an, das `.sql`-Dateien mit benutzerdefinierten Abfragen enthält.
Plugins.Oracle.CustomQueriesEnabled	Aktivieren Sie die Ausführung des item-Schlüssels `oracle.custom.query`.
Plugins.Oracle.Default.Password	Geben Sie das Standardpasswort an.
Plugins.Oracle.Default.Service	Geben Sie den Standard-Servicenamen an.
Plugins.Oracle.Default.URI	Geben Sie die Standard-URI an.
Plugins.Oracle.Default.User	Geben Sie den Standardbenutzernamen an.
Plugins.Oracle.KeepAlive	Legen Sie die maximale Wartezeit in Sekunden fest, bevor ungenutzte Plugin-Verbindungen geschlossen werden.
Plugins.Oracle.ResolveTNS	Geben Sie an, wie die Verbindungszeichenfolge (ConnString) für die Verbindung mit dem Oracle-Server interpretiert werden soll.
Plugins.Oracle.Sessions.<SessionName>.Password	Geben Sie das Passwort der benannten Sitzung an.
Plugins.Oracle.Sessions.<SessionName>.Service	Geben Sie den Servicenamen der benannten Sitzung an.
Plugins.Oracle.Sessions.<SessionName>.URI	Geben Sie die URI der benannten Sitzung an.
Plugins.Oracle.Sessions.<SessionName>.User	Geben Sie den Benutzernamen der benannten Sitzung an.

Beachten Sie:

- Die Standardwerte spiegeln die Prozess-Standardeinstellungen wider, nicht die Werte in den mitgelieferten Konfigurationsdateien.
- Werte unterstützen Umgebungsvariablen.
- Zabbix unterstützt Konfigurationsdateien nur in UTF-8-Kodierung ohne BOM.
- Kommentare, die mit ”#” beginnen, werden nur am Zeilenanfang unterstützt.

Parameterdetails

Plugins.Oracle.CallTimeout

Die maximale Wartezeit in Sekunden, bis eine Anfrage abgeschlossen ist.
 Dieser Parameter hat derzeit keine Auswirkung. Die maximale Wartezeit wird stattdessen durch das Datenpunkt-Timeout bestimmt.

Standard: `Timeout`[1]
 Bereich: 1–30

Plugins.Oracle.ConnectTimeout

Geben Sie die maximale Wartezeit in Sekunden an, bis eine Verbindung hergestellt wird.

Standard: `Timeout`[1]
 Bereich: 1–30

Plugins.Oracle.CustomQueriesPath

Geben Sie den vollständigen Pfadnamen des Verzeichnisses an, das `.sql`-Dateien mit benutzerdefinierten Abfragen enthält.

Standardwerte:
/usr/local/share/zabbix/custom-queries/oracle - für Unix-Systeme,
*:\Program Files\Zabbix Agent 2\Custom Queries\Oracle - für Windows-Systeme (wobei * der Laufwerksname aus der Umgebungsvariable ProgramFiles ist)

Beispiel:

```
/etc/zabbix/oracle/sql
```

Plugins.Oracle.CustomQueriesEnabled

Aktiviert die Ausführung des Datenpunkt-Schlüssels oracle.custom.query.
 Wenn diese Option deaktiviert ist, werden keine Abfragen aus dem Verzeichnispfad für benutzerdefinierte Abfragen geladen.

Standard: false
 Bereich: true/false

Plugins.Oracle.Default.Password

Geben Sie das Standardpasswort für die Verbindung zu Oracle an.
 Wird verwendet, wenn in einem Datenpunkt-Schlüssel oder einer benannten Sitzung kein Wert angegeben ist.

Plugins.Oracle.Default.Service

Geben Sie den Standarddienstnamen für die Verbindung zu Oracle an (SID wird nicht unterstützt).
 Wird verwendet, wenn in einem Datenpunkt-Schlüssel oder einer benannten Sitzung kein Wert angegeben ist.

Plugins.Oracle.Default.Uri

Geben Sie die Standard-URI für die Verbindung zu Oracle an.
 Wird verwendet, wenn in einem Datenpunkt-Schlüssel oder einer benannten Sitzung kein Wert angegeben ist.
 Eingebettete Anmeldedaten werden ignoriert.
 Muss dem URI-Format entsprechen.
 Unterstützte Schemas: tcp, tcps
 Das Schema kann weggelassen werden.
 Der Port kann weggelassen werden (Standard=1521).
 Es ist auch möglich, den TNS-Schlüssel oder den TNS-Wert als Verbindungszeichenfolge anzugeben. Der TNS-Wert muss ohne Leerzeichen zusammengesetzt werden.

Beispiele:

```
tcp://127.0.0.1:1521
localhost
zbx_tns_example (TNS key)
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=xe))) (TNS val
```

Plugins.Oracle.Default.User

Geben Sie den Standardbenutzernamen für die Verbindung zu Oracle an.
 Wird verwendet, wenn in einem Datenpunkt-Schlüssel oder einer benannten Sitzung kein Wert angegeben ist.

Plugins.Oracle.KeepAlive

Legt die maximale Wartezeit (in Sekunden) fest, bevor ungenutzte Plugin-Verbindungen geschlossen werden.

Standard: 300
 Bereich: 60-900

Plugins.Oracle.ResolveTNS

Geben Sie an, wie die Verbindungszeichenfolge (ConnString) für die Verbindung zum Oracle-Server interpretiert werden soll.
 Wenn der Wert auf false gesetzt ist, werden das Standardschema und der Standardport zu ConnString hinzugefügt (falls sie fehlen).
 Wenn die Option auf true gesetzt ist, werden das Standardschema und der Standardport weggelassen (sofern sie nicht ausdrücklich in ConnString angegeben sind), und ConnString wird unverändert an den Oracle-Client übergeben. Wenn der Oracle-Client ConnString in der Datei tnsnames.ora findet, wird die gefundene Verbindungsbeschreibung für die Verbindung zum Oracle-Server verwendet.

Standard: true
 Bereich: true/false

Plugins.Oracle.Sessions.<SessionName>.Password

Geben Sie das Passwort der benannten Sitzung für die Verbindung zu Oracle an.
 <SessionName> - Name einer Sitzung definieren, die in Datenpunkt-Schlüsseln verwendet wird.

Plugins.Oracle.Sessions.<SessionName>.Service

Geben Sie den Namen des benannten Sitzungsdienstes für die Verbindung zu Oracle an (SID wird nicht unterstützt).
 <SessionName> - Name einer Sitzung definieren, der in Datenpunktschlüsseln verwendet wird.

Plugins.Oracle.Sessions.<SessionName>.Uri

Geben Sie die URI der benannten Sitzung für die Verbindung zu Oracle an.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunkt-Schlüsseln.
 Eingebettete Anmeldedaten werden ignoriert.
 Muss dem URI-Format

entsprechen.
 Unterstützte Schemas: `tcp`, `tcps`
 Das Schema kann weggelassen werden.
 Der Port kann weggelassen werden (Standard=1521).
 Es ist auch möglich, den TNS-Schlüssel oder den TNS-Wert als Verbindungszeichenfolge anzugeben. Der TNS-Wert muss ohne Leerzeichen zusammengesetzt sein.

Beispiele:

```
tcp://127.0.0.1:1521
localhost
zbx_tns_example (TNS key)
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=xe))) (TNS val
```

Plugins.Oracle.Sessions.<SessionName>.User

Geben Sie den Benutzernamen der benannten Sitzung für die Verbindung zu Oracle an.
 <SessionName> - Name einer Sitzung festlegen, die in Datenpunkt-Schlüsseln verwendet wird.

Fußnoten

[1] Das Timeout wird durch die Option `Timeout` in der Konfigurationsdatei (`zabbix_agent2.conf`) des Zabbix-Agent 2 (UNIX) oder des Zabbix-Agent 2 (Windows) festgelegt.

Siehe auch

- Beschreibung der allgemeinen Konfigurationsparameter von Zabbix Agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Anweisungen zum Konfigurieren von Plugins

PostgreSQL-Plugin

Übersicht

Konfigurationsparameter ermöglichen die Anpassung des Verhaltens des PostgreSQL-Plugins.

Es wird empfohlen, sie in einer eigenen Konfigurationsdatei (z. B. `postgresql.conf`) anzugeben, statt in der Konfigurationsdatei von Agent 2. Verwenden Sie dann die Direktive Include, um diese Datei in die Konfiguration von Zabbix Agent 2 einzubinden.

Das PostgreSQL-Plugin ist ein ladbares Plugin und ist im Repository des PostgreSQL-Plugins verfügbar und dort vollständig beschrieben.

Alle Parameter sind **optional**, sofern nicht ausdrücklich angegeben ist, dass der Parameter obligatorisch ist.

Diese Seite ist wie folgt aufgebaut:

- Parameterübersicht (um die vollständigen Details eines Parameters anzuzeigen, klicken Sie auf seinen Namen)
- Parameterdetails

Parameter	Beschreibung
Plugins.PostgreSQL.CallTimeout	Legt die maximale Wartezeit in Sekunden fest, bis eine Anfrage abgeschlossen ist.
Plugins.PostgreSQL.CustomQueriesPath	Geben Sie den vollständigen Pfadnamen des Verzeichnisses an, das `.sql`-Dateien mit benutzerdefinierten Abfragen enthält.
Plugins.PostgreSQL.CustomQueriesEnabled	Aktiviert die Ausführung des item-Schlüssels `postgresql.custom.query`.
Plugins.PostgreSQL.Default.CacheMode	Geben Sie den Standard-Cachemodus für die PostgreSQL-Verbindung an.
Plugins.PostgreSQL.Default.Database	Geben Sie den Standard-Datenbanknamen an.
Plugins.PostgreSQL.Default.Password	Geben Sie das Standardpasswort an.
Plugins.PostgreSQL.Default.TLSCAFile	Geben Sie den vollständigen Pfadnamen der Datei an, die das Zertifikat der obersten CA(s) für die Peer-Zertifikatsprüfung enthält.
Plugins.PostgreSQL.Default.TLSCertFile	Geben Sie den vollständigen Pfadnamen der Datei an, die das PostgreSQL-Zertifikat oder die Zertifikatskette enthält.
Plugins.PostgreSQL.Default.TLSConnect	Geben Sie den Standard-Verschlüsselungsgrad für die Kommunikation zwischen Zabbix Agent 2 und den überwachten Datenbanken an.
Plugins.PostgreSQL.Default.TLSKeyFile	Geben Sie den vollständigen Pfadnamen der Datei an, die den privaten PostgreSQL-Schlüssel enthält.
Plugins.PostgreSQL.Default.Uri	Geben Sie die Standard-URI an.
Plugins.PostgreSQL.Default.User	Geben Sie den Standard-Benutzernamen an.
Plugins.PostgreSQL.KeepAlive	Legt die maximale Wartezeit in Sekunden fest, bevor ungenutzte Plugin-Verbindungen geschlossen werden.
Plugins.PostgreSQL.Sessions.<SessionName>.CacheMode	Geben Sie den Cachemodus der benannten Sitzung für die PostgreSQL-Verbindung an.
Plugins.PostgreSQL.Sessions.<SessionName>.Database	Geben Sie den Datenbanknamen der benannten Sitzung an.
Plugins.PostgreSQL.Sessions.<SessionName>.Password	Geben Sie das Passwort der benannten Sitzung an.

Parameter	Beschreibung
Plugins.PostgreSQL.Sessions.<Session name>.TLSCAFile	Geben Sie den vollständigen Pfadnamen der Datei an, die das Zertifikat der obersten CA(s) für die Peer-Zertifikatsprüfung enthält.
Plugins.PostgreSQL.Sessions.<Session name>.TLSCertFile	Geben Sie den vollständigen Pfadnamen der Datei an, die das PostgreSQL-Zertifikat oder die Zertifikatskette enthält.
Plugins.PostgreSQL.Sessions.<Session name>.TLSConnect	Geben Sie den verschlüsselungsgrad der benannten Sitzung für die Kommunikation zwischen Zabbix Agent 2 und den überwachten Datenbanken an.
Plugins.PostgreSQL.Sessions.<Session name>.TLSKeyFile	Geben Sie den vollständigen Pfadnamen der Datei an, die den privaten PostgreSQL-Schlüssel enthält.
Plugins.PostgreSQL.Sessions.<Session name>.Uri	Geben Sie die URI der benannten Sitzung an.
Plugins.PostgreSQL.Sessions.<Session name>.User	Geben Sie den Benutzernamen der benannten Sitzung an.

|Plugins.PostgreSQL.System.Path|Geben Sie den Pfad zur ausführbaren Datei des PostgreSQL-Plugins an.| |Plugins.PostgreSQL.Timeout|Legt das Zeitlimit für die Ausführung von Anfragen fest.|

Beachten Sie:

- Die Standardwerte spiegeln die Prozess-Standardeinstellungen wider, nicht die Werte in den mitgelieferten Konfigurationsdateien.
- Werte unterstützen Umgebungsvariablen;
- Zabbix unterstützt Konfigurationsdateien nur in UTF-8-Kodierung ohne BOM.
- Kommentare, die mit ”#” beginnen, werden nur am Zeilenanfang unterstützt.

Parameterdetails

Plugins.PostgreSQL.CallTimeout

Legt die maximale Wartezeit in Sekunden fest, bis eine Anfrage abgeschlossen ist.

Standard: `global timeout`
 Bereich: 1-30

Plugins.PostgreSQL.CustomQueriesPath

Geben Sie den vollständigen Pfadnamen des Verzeichnisses an, das `.sql`-Dateien mit benutzerdefinierten Abfragen enthält.

Standardwerte:
`/usr/local/share/zabbix/custom-queries/postgreSQL` - für Unix-Systeme,
`*:\Program Files\Zabbix Agent 2\Custom Queries\PostgreSQL` - für Windows-Systeme (wobei `*` der Laufwerksname aus der Umgebungsvariable `ProgramFiles` ist)

Beispiel:

```
/etc/zabbix/oracle/sql
```

Plugins.PostgreSQL.CustomQueriesEnabled

Aktiviert die Ausführung des Datenpunkt-Schlüssels `postgresql.custom.query`.
 Wenn diese Option deaktiviert ist, werden keine Abfragen aus dem Verzeichnispfad für benutzerdefinierte Abfragen geladen.

Standard: `false`
 Bereich: `true`/`false`

Plugins.PostgreSQL.Default.CacheMode

Geben Sie den Standard-Cache-Modus für die PostgreSQL-Verbindung an.

Standard: `prepare`
 Bereich:
`prepare` - vorbereitete Anweisungen auf dem PostgreSQL-Server erstellen;
`describe` - die anonyme vorbereitete Anweisung verwenden, um eine Anweisung zu beschreiben, ohne eine Anweisung auf dem Server zu erstellen.

Beachten Sie, dass `describe` vor allem dann nützlich ist, wenn die Umgebung keine vorbereiteten Anweisungen zulässt, z. B. beim Einsatz eines Connection Poolers wie PgBouncer.

Plugins.PostgreSQL.Default.Database

Geben Sie die Standarddatenbank für die Verbindung zu PostgreSQL an.
 Wird verwendet, wenn in einem Datenpunkt-Schlüssel oder einer benannten Sitzung kein Wert angegeben ist.

Plugins.PostgreSQL.Default.Password

Geben Sie das Standardpasswort für die Verbindung zu PostgreSQL an.
 Wird verwendet, wenn im Datenpunkt-Schlüssel oder in der benannten Sitzung kein Wert angegeben ist.

Plugins.PostgreSQL.Default.TLSCAFile

Geben Sie den vollständigen Pfadnamen der Datei an, die das Zertifikat der obersten CA(s) für die Überprüfung des Peer-Zertifikats bei verschlüsselter Kommunikation zwischen Zabbix Agent 2 und den überwachten Datenbanken enthält.
 Wird verwendet, wenn in einer benannten Sitzung kein Wert angegeben ist.

Pflicht: `yes`, wenn `Plugins.PostgreSQL.Default.TLSConnect` auf `verify_ca` oder `verify_full` gesetzt ist

Plugins.PostgreSQL.Default.TLSCertFile

Geben Sie den vollständigen Pfadnamen der Datei an, die das PostgreSQL-Zertifikat oder die Zertifikatskette für verschlüsselte Kommunikation zwischen Zabbix Agent 2 und überwachten Datenbanken enthält.
 Wird verwendet, wenn in einer benannten Sitzung kein Wert angegeben ist.

Erforderlich: `yes`, wenn `Plugins.PostgreSQL.Default.TLSKeyFile` angegeben ist

Plugins.PostgreSQL.Default.TLSConnect

Geben Sie den standardmäßigen Verschlüsselungsgrad für die Kommunikation zwischen Zabbix Agent 2 und den überwachten Datenbanken an.
 Wird verwendet, wenn in einer benannten Sitzung kein Wert angegeben ist.

Bereich:
`required` - Verbindung mit TLS als Transportmodus ohne Identitätsprüfung herstellen;
`verify_ca` - Verbindung mit TLS herstellen und das Zertifikat überprüfen;
`verify_full` - Verbindung mit TLS herstellen, das Zertifikat überprüfen und verifizieren, dass die von DBHost angegebene Datenbankidentität (CN) mit ihrem Zertifikat übereinstimmt.

Ein nicht definierter Verschlüsselungstyp bedeutet eine unverschlüsselte Verbindung.

Plugins.PostgreSQL.Default.TLSKeyFile

Geben Sie den vollständigen Pfadnamen der Datei an, die den privaten PostgreSQL-Schlüssel für verschlüsselte Kommunikation zwischen Zabbix Agent 2 und überwachten Datenbanken enthält.
 Wird verwendet, wenn in einer benannten Sitzung kein Wert angegeben ist.

Erforderlich: `yes`, wenn `Plugins.PostgreSQL.Default.TLSCertFile` angegeben ist

Plugins.PostgreSQL.Default.Uri

Geben Sie den Standard-URI für die Verbindung zu PostgreSQL an.
 Wird verwendet, wenn in einem Datenpunkt-Schlüssel oder einer benannten Sitzung kein Wert angegeben ist.
 Eingebettete Anmeldedaten werden ignoriert.
 Muss dem URI-Format entsprechen.
 Unterstützte Schemas: `tcp`, `unix`

Beispiele:

```
tcp://127.0.0.1:5432
tcp://localhost
unix:/var/run/postgresql/.s.PGSQL.5432
```

Plugins.PostgreSQL.Default.User

Geben Sie den Standard-Benutzernamen für die Verbindung zu PostgreSQL an.
 Wird verwendet, wenn in einem Datenpunkt-Schlüssel oder einer benannten Sitzung kein Wert angegeben ist.

Plugins.PostgreSQL.KeepAlive

Legt die maximale Wartezeit (in Sekunden) fest, bevor ungenutzte Plugin-Verbindungen geschlossen werden.

Plugins.PostgreSQL.Sessions.<SessionName>.CacheMode

Geben Sie den benannten Sitzungscachemodus für die PostgreSQL-Verbindung an.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunkt-Schlüsseln.

Standard: `prepare`
 Bereich:
 `prepare` - erstellt vorbereitete Anweisungen auf dem PostgreSQL-Server;
 `describe` - verwendet die anonyme vorbereitete Anweisung, um eine Anweisung zu beschreiben, ohne eine Anweisung auf dem Server zu erstellen.

Beachten Sie, dass `describe` vor allem dann nützlich ist, wenn die Umgebung keine vorbereiteten Anweisungen zulässt, z. B. beim Einsatz eines Connection Poolers wie PgBouncer.

Plugins.PostgreSQL.Sessions.<SessionName>.Database

Geben Sie die Datenbank für die benannte Sitzung an, um eine Verbindung zu PostgreSQL herzustellen.
 <SessionName> - Name einer Sitzung festlegen, die in Datenpunkt-Schlüsseln verwendet wird.

Plugins.PostgreSQL.Sessions.<SessionName>.Password

Geben Sie das Passwort für die benannte Sitzung an, um eine Verbindung zu PostgreSQL herzustellen.
 <SessionName> - Definieren Sie den Namen einer Sitzung zur Verwendung in Datenpunktschlüsseln.
 Muss dem Passwortformat entsprechen.

Plugins.PostgreSQL.Sessions.<SessionName>.TLSCAFile

Geben Sie den vollständigen Pfadnamen der Datei an, die das Zertifikat der obersten CA(s) für die Überprüfung des Peer-Zertifikats bei verschlüsselter Kommunikation zwischen Zabbix Agent 2 und den überwachten Datenbanken enthält.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunkt-Schlüsseln.

Pflicht: `yes`, wenn `Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect` auf `verify_ca` oder `verify_full` gesetzt ist

Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile

Geben Sie den vollständigen Pfadnamen der Datei an, die das PostgreSQL-Zertifikat oder die Zertifikatskette für verschlüsselte Verbindungen zwischen Zabbix Agent 2 und überwachten Datenbanken enthält.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunkt-Schlüsseln.

Pflicht: `yes`, wenn `Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile` angegeben ist

Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect

Geben Sie die benannte Verschlüsselungsstufe auf Sitzungsebene für die Kommunikation zwischen Zabbix Agent 2 und den überwachten Datenbanken an.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunkt-Schlüsseln.

Bereich:
 `required` - Verbindung über TLS als Transportmodus ohne Identitätsprüfung;
 `verify_ca` - Verbindung über TLS und Zertifikatsprüfung;
 `verify_full` - Verbindung über TLS, Zertifikatsprüfung und Prüfung, dass die von DBHost angegebene Datenbankidentität (CN) mit ihrem Zertifikat übereinstimmt.

Ein nicht definierter Verschlüsselungstyp bedeutet eine unverschlüsselte Verbindung.

Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile

Geben Sie den vollständigen Pfadnamen der Datei an, die den privaten PostgreSQL-Schlüssel für die verschlüsselte Kommunikation zwischen Zabbix Agent 2 und den überwachten Datenbanken enthält.
 <SessionName> - definieren Sie den Namen einer Sitzung zur Verwendung in Datenpunkt-Schlüsseln.

Erforderlich: `yes`, wenn `Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile` angegeben ist.

Plugins.PostgreSQL.Sessions.<SessionName>.Uri

Geben Sie die URI der benannten Sitzung für die Verbindung zu PostgreSQL an.
 <SessionName> - definiert den Namen einer Sitzung zur Verwendung in Datenpunkt-Schlüsseln.
 Eingebettete Anmeldedaten werden ignoriert.
 Muss dem URI-Format entsprechen.
 Unterstützte Schemas: `tcp`, `unix`

Beispiele:

```
tcp://127.0.0.1:5432
tcp://localhost
unix:/var/run/postgresql/.s.PGSQL.5432
```

Plugins.PostgreSQL.Sessions.<SessionName>.User

Geben Sie den Benutzernamen der benannten Sitzung für die Verbindung zu PostgreSQL an.
 <SessionName> - Name einer Sitzung definieren, die in Datenpunktschlüsseln verwendet wird.

Plugins.PostgreSQL.System.Path

Geben Sie den Pfad zur ausführbaren Datei des PostgreSQL-Plugins an.

Erforderlich: `yes`

Beispiel:

```
Plugins.PostgreSQL.System.Path=/usr/libexec/zabbix/zabbix-agent2-plugin-postgresql
```

Plugins.PostgreSQL.Timeout

Legt das Timeout für die Ausführung von Anfragen fest (die Dauer in Sekunden, die auf den Abschluss einer Anfrage gewartet wird, bevor sie beendet wird).

Standard: `global timeout`
 Bereich: 1-30

Optionen

Parameter	Beschreibung
-V --version	Gibt die Plugin-Version und Lizenzinformationen aus.
-h --help	Gibt Hilfeinformationen aus (Kurzform).

Parameter	Beschreibung
-t, --test <item key>	Startet das Plugin zum Testen (Plugin-Konfiguration wird ignoriert).

Siehe auch

- Beschreibung der allgemeinen Konfigurationsparameter von Zabbix Agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Anweisungen zum Konfigurieren von Plugins

Redis-Plugin

Übersicht

Konfigurationsparameter ermöglichen die Anpassung des Verhaltens des Redis-Plugins.

Es wird empfohlen, sie in einer eigenen Konfigurationsdatei (z. B. `redis.conf`) statt in der Konfigurationsdatei von Agent 2 anzugeben. Verwenden Sie dann die Direktive Include, um diese Datei in die Konfiguration von Zabbix Agent 2 einzubinden.

Alle Parameter sind **optional**, sofern nicht ausdrücklich angegeben ist, dass der Parameter obligatorisch ist.

Diese Seite ist wie folgt aufgebaut:

- Parameterübersicht (um die vollständigen Details eines Parameters anzuzeigen, klicken Sie auf seinen Namen)
- Parameterdetails

Parameter	Beschreibung
Plugins.Redis.Default.Password	Geben Sie das Standardpasswort für die Verbindung zu Redis an.
Plugins.Redis.Default.TLSConnect	Geben Sie die erforderliche Verschlüsselungsstufe für die Kommunikation von Zabbix Agent 2 an.
Plugins.Redis.Default.TLSCAFile	Geben Sie den vollständigen Pfadnamen der Datei an, die die Zertifikate der obersten CA(s) enthält.
Plugins.Redis.Default.TLSCertFile	Geben Sie den vollständigen Pfadnamen der Datei an, die das Agent-Zertifikat enthält.
Plugins.Redis.Default.TLSKeyFile	Geben Sie den vollständigen Pfadnamen der Datei an, die den privaten Schlüssel des Agent enthält.
Plugins.Redis.Default.Uri	Geben Sie die Standard-URI für die Verbindung zu Redis an.
Plugins.Redis.Default.User	Geben Sie den Standardbenutzernamen für die Verbindung zu Redis an.
Plugins.Redis.KeepAlive	Legen Sie die maximale Wartezeit (in Sekunden) fest, bevor ungenutzte Plugin-Verbindungen geschlossen werden.
Plugins.Redis.Sessions.<Session>.Password	Geben Sie das Passwort der benannten Sitzung für die Verbindung zu Redis an.
Plugins.Redis.Sessions.<Session>.TLSConnect	Geben Sie die erforderliche Verschlüsselungsstufe für die Kommunikation von Zabbix Agent 2 für die benannte Sitzung an.
Plugins.Redis.Sessions.<Session>.TLSCAFile	Geben Sie den vollständigen Pfadnamen der Datei an, die die Zertifikate der obersten CA(s) für die benannte Sitzung enthält.
Plugins.Redis.Sessions.<Session>.TLSCertFile	Geben Sie den vollständigen Pfadnamen der Datei an, die das Agent-Zertifikat für die benannte Sitzung enthält.
Plugins.Redis.Sessions.<Session>.TLSKeyFile	Geben Sie den vollständigen Pfadnamen der Datei an, die den privaten Schlüssel des Agent für die benannte Sitzung enthält.
Plugins.Redis.Sessions.<Session>.Uri	Geben Sie die URI der benannten Sitzung für die Verbindung zu Redis an.
Plugins.Redis.Sessions.<Session>.User	Geben Sie den Benutzernamen der benannten Sitzung für die Verbindung zu Redis an.
Plugins.Redis.Timeout	Legen Sie das Timeout für die Ausführung von Anfragen fest.

Beachten Sie:

- Die Standardwerte spiegeln die Prozess-Standardwerte wider, nicht die Werte in den mitgelieferten Konfigurationsdateien.
- Werte unterstützen Umgebungsvariablen.
- Zabbix unterstützt Konfigurationsdateien nur in UTF-8-Kodierung ohne BOM.
- Kommentare, die mit ”#” beginnen, werden nur am Zeilenanfang unterstützt.

Parameterdetails

Plugins.Redis.Default.Password

Geben Sie das Standardpasswort für die Verbindung zu Redis an.
 Wird verwendet, wenn in einem Datenpunkt-Schlüssel oder einer benannten Sitzung kein Wert angegeben ist.

Plugins.Redis.Default.TLSConnect

Geben Sie das für die Kommunikation von Zabbix Agent 2 erforderliche Verschlüsselungsniveau an.
 Ein leerer oder nicht gesetzter Wert wird als kein TLS behandelt.

Bereich: `required` - verschlüsselt, nicht verifiziert (nur für Tests), `verify_ca` - verschlüsselt, Serverzertifikat, `verify_full` - verschlüsselt, Serverzertifikat und Servername mit SAN verifiziert

Plugins.Redis.Default.TLSCAFile

Geben Sie den vollständigen Pfadnamen der Datei an, die die Zertifikate der obersten CA(s) enthält.

Plugins.Redis.Default.TLSCertFile

Geben Sie den vollständigen Pfadnamen der Datei an, die das Agent-Zertifikat enthält.

Plugins.Redis.Default.TLSKeyFile

Geben Sie den vollständigen Pfadnamen der Datei an, die den privaten Schlüssel des Agent enthält.

Plugins.Redis.Default.Uri

Geben Sie den Standard-URI für die Verbindung zu Redis an.
 Wird verwendet, wenn kein Wert in einem Datenpunkt-Schlüssel oder einer benannten Sitzung angegeben ist.
 Eingebettete Anmeldedaten werden ignoriert. Muss dem URI-Format entsprechen.
 Unterstützte Schemas: `tcp`, `unix`
 Ein Schema kann weggelassen werden.
 Ein Port kann weggelassen werden (Standard=6379).

Standard: `tcp://localhost:6379`

Beispiele:

```
tcp://localhost:6379
localhost
unix:/var/run/redis.sock
```

Plugins.Redis.Default.User

Geben Sie den Standard-Benutzernamen für die Verbindung zu Redis an.
 Wird verwendet, wenn in einem Datenpunkt-Schlüssel oder einer benannten Sitzung kein Wert angegeben ist.

Standard: `default`

Plugins.Redis.KeepAlive

Legt die maximale Wartezeit (in Sekunden) fest, bevor ungenutzte Plugin-Verbindungen geschlossen werden.

Standard: 300
 Bereich: 60–900

Plugins.Redis.Sessions.<SessionName>.Password

Geben Sie das Passwort der benannten Sitzung für die Verbindung zu Redis an.
 <SessionName> - Definieren Sie den Sitzungsnamen, der in Datenpunkt-Schlüsseln verwendet werden soll.

Plugins.Redis.Sessions.<SessionName>.TLSConnect

Geben Sie den für die Kommunikation von Zabbix Agent 2 für die benannte Sitzung erforderlichen Verschlüsselungsgrad an.
 <SessionName> - definiert den Sitzungsnamen, der in Datenpunkt-Schlüsseln verwendet werden soll.
 Ein leerer oder nicht gesetzter Wert wird als kein TLS behandelt.

Bereich: `required` - verschlüsselt, nicht verifiziert (nur zum Testen), `verify_ca` - verschlüsselt, Server-Zertifikat, `verify_full` - verschlüsselt, Server-Zertifikat und Servername mit SAN verifiziert

Plugins.Redis.Sessions.<SessionName>.TLSCAFile

Geben Sie den vollständigen Pfad zur Datei an, die die Zertifikate der obersten CA(s) für die benannte Sitzung enthält.
 <SessionName> - definiert den Namen der Sitzung, der in Datenpunkt-Schlüsseln verwendet werden soll.

Pflicht: `yes`, wenn `Plugins.Redis.Sessions.<SessionName>.TLSConnect` auf `verify_ca` oder `verify_full` gesetzt ist

Plugins.Redis.Sessions.<SessionName>.TLSCertFile

Geben Sie den vollständigen Pfadnamen der Datei an, die das Agent-Zertifikat für die benannte Sitzung enthält.
 <SessionName> - definiert den Sitzungsnamen, der in Datenpunkt-Schlüsseln verwendet werden soll.

Plugins.Redis.Sessions.<SessionName>.TLSKeyFile

Geben Sie den vollständigen Pfadnamen der Datei an, die den privaten Schlüssel des Agent für die benannte Sitzung enthält.
 <SessionName> - definiert den Sitzungsnamen, der in Datenpunkt-Schlüsseln verwendet werden soll.

Plugins.Redis.Sessions.<SessionName>.Uri

Geben Sie die URI der benannten Sitzung an, um eine Verbindung zu Redis herzustellen.
 <SessionName> - definiert den Sitzungsnamen, der in Datenpunkt-Schlüsseln verwendet werden soll.
 Eingebettete Anmeldedaten werden ignoriert. Muss dem URI-Format entsprechen.
 Unterstützte Schemas: `tcp`, `unix`
 Ein Schema kann weggelassen werden.
 Ein Port kann weggelassen werden (Standard=6379).

Standard: `tcp://localhost:6379`

Beispiele:

```
tcp://localhost:6379
localhost
unix:/var/run/redis.sock
```

Plugins.Redis.Sessions.<SessionName>.User

Geben Sie den Benutzernamen der benannten Sitzung an, um eine Verbindung zu Redis herzustellen.
 <SessionName> - definiert den Sitzungsnamen, der in Datenpunktschlüsseln verwendet werden soll.

Plugins.Redis.Timeout

Legt das Timeout für die Ausführung von Anfragen fest (die Dauer in Sekunden, die auf den Abschluss einer Anfrage gewartet wird, bevor sie beendet wird).

Standard: `global timeout`
 Bereich: 1-30

Siehe auch

- Beschreibung der allgemeinen Konfigurationsparameter von Zabbix Agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Anweisungen zum Konfigurieren von Plugins

SMART-Plugin

Übersicht

Konfigurationsparameter ermöglichen die Anpassung des Verhaltens des SMART-Plugins.

Es wird empfohlen, sie in einer eigenen Konfigurationsdatei (z. B. `smart.conf`) statt in der Konfigurationsdatei von Agent 2 anzugeben. Verwenden Sie dann die Direktive Include, um diese Datei in die Konfiguration von Zabbix Agent 2 einzubinden.

Alle Parameter sind **optional**, sofern nicht ausdrücklich angegeben ist, dass der Parameter obligatorisch ist.

Diese Seite ist wie folgt aufgebaut:

- Parameterübersicht (um die vollständigen Details eines Parameters anzuzeigen, klicken Sie auf seinen Namen)
- Parameterdetails

Parameter	Beschreibung
Plugins.Smart.Path	Geben Sie den Pfad zur ausführbaren Datei `smartctl` an.
Plugins.Smart.Timeout	Legt das Timeout für die Ausführung der Anfrage fest (die Dauer in Sekunden, die auf den Abschluss einer Anfrage gewartet wird, bevor sie beendet wird).

Beachten Sie:

- Die Standardwerte entsprechen den Prozess-Standardeinstellungen, nicht den Werten in den mitgelieferten Konfigurationsdateien;
- Werte unterstützen Umgebungsvariablen;
- Der Pfad zur ausführbaren Datei `smartctl` muss entweder durch Hinzufügen ihres Verzeichnisses zur Umgebungsvariable `PATH` des Systems oder durch Konfiguration von `Plugins.Smart.Path` festgelegt werden; dies gilt sowohl für Linux als auch für Windows;
- Zabbix unterstützt Konfigurationsdateien nur in UTF-8-Kodierung ohne BOM;
- Kommentare, die mit "#" beginnen, werden nur am Anfang der Zeile unterstützt.

Parameterdetails

Plugins.Smart.Path

Geben Sie den Pfad zur ausführbaren Datei `smartctl` an.

Standard: `smartctl`

Plugins.Smart.Timeout

Legt das Timeout für die Ausführung der Anfrage fest (die Dauer in Sekunden, die auf den Abschluss einer Anfrage gewartet wird, bevor sie beendet wird).

Standard: `global timeout<br>` Bereich: 1–30

Siehe auch

- Beschreibung der allgemeinen Konfigurationsparameter von Zabbix Agent 2: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Anweisungen zum Konfigurieren von Plugins

Zabbix Agent 2-Plugin-Protokoll

Das Zabbix Agent 2-Protokoll basiert auf Code, Größe und Datenmodell.

Code

Type	Size	Comments
Byte	4	Payload-Typ, derzeit wird nur JSON unterstützt.

Größe

Type	Size	Comments
Byte	4	Größe der aktuellen Nutzlast in Bytes.

Nutzdaten

Type	Size	Comments
Byte	Definiert durch das Feld *Size*	JSON-formatierte Daten.

Definition der Payload-Daten

Gemeinsame Daten

Diese Parameter sind in allen Anfragen/Antworten vorhanden:

Name	Type	Comments
id	uint32	Für Anfragen - der fortlaufende Bezeichner, der verwendet wird, um Anfragen mit Antworten zu verknüpfen. Innerhalb einer Nachrichtenrichtung eindeutig (d. h. vom Agent zum Plugin oder vom Plugin zum Agent). Für Antworten - die ID der entsprechenden Anfrage.
type	uint32	Der Anfragetyp.

Protokollanforderung

Eine von einem Plugin gesendete Anforderung, um eine Protokollmeldung in die Protokolldatei des Agent zu schreiben.

direction	plugin → agent
response	no

Parameter, die für Protokollanforderungen spezifisch sind:

Name	Type	Comments
severity	uint32	Die Schwere der Meldung (Protokollstufe).

Name	Type	Comments
message	string	Die zu protokollierende Meldung.

Beispiel:

```
{"id":0,"type":1,"severity":3,"message":"message"}
```

Registrierungsanfrage

Eine Anfrage, die vom Agent während der Startphase des Agent gesendet wird, um bereitgestellte Metriken abzurufen und ein Plugin zu registrieren.

direction	Agent → plugin
response	yes

Für Registrierungsanfragen spezifische Parameter:

Name	Type	Comments
version	string	Die Protokollversion <major>.<minor>

Beispiel:

```
{"id":1,"type":2,"version":"1.0"}
```

Register-Antwort

Antwort des Plugins auf die Register-Anfrage.

direction	plugin → Agent
response	n/a

Für Register-Antworten spezifische Parameter:

Name	Type	Comments
name	string	Der Name des Plugins.
metrics	array of strings (optional)	Die Metriken mit Beschreibungen, wie sie im Plugin verwendet werden. Gibt RegisterMetrics() zurück. Nicht vorhanden, wenn ein Fehler zurückgegeben wird.
interfaces	uint32 (optional)	Die Bitmaske der vom Plugin unterstützten Schnittstellen. Nicht vorhanden, wenn ein Fehler zurückgegeben wird.
error	string (optional)	Eine Fehlermeldung, die zurückgegeben wird, wenn ein Plugin nicht gestartet werden kann. Nicht vorhanden, wenn Metriken zurückgegeben werden.

Beispiele:

```
{"id":2,"type":3,"metrics":["external.test", "External exporter Test."], "interfaces": 4}
```

oder

```
{"id":2,"type":3,"error":"error message"}
```

Start-Anfrage

Eine Anfrage zum Ausführen der Start-Funktion der Runner-Schnittstelle.

direction	Agent → plugin
response	no

Die Anfrage hat keine spezifischen Parameter, sie enthält nur Parameter für gemeinsame Daten.

Beispiel:

```
{"id":3,"type":4}
```

Anfrage beenden

Eine vom Agent an ein Plugin gesendete Anfrage zum Herunterfahren eines Plugins.

direction	Agent → plugin
response	no

Die Anfrage hat keine spezifischen Parameter, sie enthält nur Parameter für gemeinsame Daten.

Beispiel:

```
{"id":3,"type":5}
```

Export-Anfrage

Eine Anfrage zum Ausführen der Export-Funktion der Exporter-Schnittstelle.

direction	agent → plugin
response	no

Für Export-Anfragen spezifische Parameter:

Name	Type	Comments
key	string	Der Plugin-Schlüssel.
parameters	array of strings (optional)	Die Parameter für die Export-Funktion.

Beispiel:

```
{"id":4,"type":6,"key":"test.key","parameters":["foo","bar"]}
```

Export-Antwort

Antwort der Export-Funktion der Exporter-Schnittstelle.

direction	plugin → agent
response	n/a

Für Export-Antworten spezifische Parameter:

Name	Type	Comments
value	string (optional)	Antwortwert der Export-Funktion. Nicht vorhanden, wenn ein Fehler zurückgegeben wird.
error	string (optional)	Fehlermeldung, wenn die Export-Funktion nicht erfolgreich ausgeführt wurde. Nicht vorhanden, wenn value zurückgegeben wird.

Beispiele:

```
{"id":5,"type":7,"value":"response"}
```

oder

```
{"id":5,"type":7,"error":"error message"}
```

Anfrage konfigurieren

Eine Anfrage zum Ausführen der Funktion *Configure* der Schnittstelle *Configurator*.

direction	Agent → plugin
response	n/a

Für *Configure*-Anfragen spezifische Parameter:

Name	Type	Comments
global_options	JSON object	JSON object mit globalen Konfigurationsoptionen des Agent.
private_options	JSON object (optional)	JSON object mit privaten Konfigurationsoptionen des Plugins, sofern angegeben.

Beispiel:

```
{"id":6,"type":8,"global_options":{...},"private_options":{...}}
```

Anfrage validieren

Eine Anfrage zum Ausführen der Funktion *Validate* der Schnittstelle *Configurator*.

direction	agent → plugin
response	yes

Für *Validate*-Anfragen spezifische Parameter:

Name	Type	Comments
private_options	JSON object (optional)	JSON object mit privaten Konfigurationsoptionen des Plugins, sofern angegeben.

Beispiel:

```
{"id":7,"type":9,"private_options":{...}}
```

Antwort validieren

Antwort der Funktion *Validate* der Schnittstelle *Configurator*.

direction	plugin → Agent
response	n/a

Für *Validate*-Antworten spezifische Parameter:

Name	Type	Comments
error	string (optional)	Eine Fehlermeldung, die zurückgegeben wird, wenn die Funktion Validate nicht erfolgreich ausgeführt wird. Fehlt bei erfolgreicher Ausführung.

Beispiel:

```
{"id":8,"type":10}
```

oder

```
{"id":8,"type":10,"error":"error message"}
```

Java gateway

Übersicht Das Zabbix Java gateway kann aus dem Quellcode oder über Pakete installiert werden.

Die native Unterstützung für die Überwachung von JMX-Anwendungen erfolgt in Form eines Zabbix-Daemons namens „Zabbix Java gateway“. Zabbix Java gateway ist ein in Java geschriebener Daemon. Um den Wert eines bestimmten JMX-Zählers auf einem Host zu ermitteln, fragt der Zabbix Server das Zabbix Java gateway ab, das die JMX Management API verwendet, um die betreffende Anwendung remote abzufragen. Für die Anwendung muss keine zusätzliche Software installiert werden. Sie muss lediglich mit der Befehlszeilenoption `-Dcom.sun.management.jmxremote` gestartet werden.

Das Java gateway akzeptiert eingehende Verbindungen vom Zabbix Server oder Proxy und kann nur als „passiver Proxy“ verwendet werden. Im Gegensatz zum Zabbix Proxy kann es auch von einem Zabbix Proxy verwendet werden (Zabbix Proxies können nicht verkettet werden). Der Zugriff auf jedes Java gateway wird direkt in der Konfigurationsdatei des Zabbix Servers oder Zabbix Proxys konfiguriert. Daher kann pro Zabbix Server oder Zabbix Proxy nur ein Java gateway konfiguriert werden. Wenn ein Host Datenpunkte vom Typ *JMX agent* und Datenpunkte anderer Typen enthält, werden nur die *JMX agent*-Datenpunkte zur Abfrage an das Java gateway übergeben.

Wenn ein Datenpunkt über das Java gateway aktualisiert werden muss, stellt der Zabbix-Server oder Proxy eine Verbindung zum Java gateway her und fordert den Wert an. Das Java gateway ruft den Wert seinerseits ab und übergibt ihn an den Server oder Proxy zurück. Daher speichert das Java gateway keine Werte zwischen.

Der Zabbix-Server oder Proxy verfügt über einen bestimmten Prozesstyp, der Verbindungen zum Java gateway herstellt und durch die Option `StartJavaPollers` gesteuert wird. Intern startet das Java gateway mehrere Threads, die durch die Option `START_POLLERS` gesteuert werden. Auf der Serverseite wird eine Verbindung beendet, wenn sie länger als ”Timeout” Sekunden dauert. Das Java gateway ist möglicherweise weiterhin damit beschäftigt, den Wert vom JMX-Zähler abzurufen. Um dies zu lösen, stellt das Java gateway die Option `TIMEOUT` bereit, mit der Sie ein Timeout für JMX-Netzwerkoperationen festlegen können.

Siehe auch:

- Konfigurationsparameter: Zabbix Java gateway
- Einrichtung aus Debian-/Ubuntu-Paketen
- Einrichtung aus RHEL-Paketen
- Einrichtung aus Quellen

Server oder Proxy werden versuchen, Anfragen an ein einzelnes JMX-Ziel so weit wie möglich zu bündeln (beeinflusst durch die Datenpunkt-Intervalle) und sie für eine bessere Leistung in einer einzigen Verbindung an das Java gateway zu senden.

Es wird empfohlen, `StartJavaPollers` kleiner oder gleich `START_POLLERS` zu setzen; andernfalls kann es Situationen geben, in denen im Java gateway keine Threads verfügbar sind, um eingehende Anfragen zu bearbeiten. In solchen Fällen verwendet das Java gateway `ThreadPoolExecutor.CallerRunsPolicy`, was bedeutet, dass der Hauptthread die eingehende Anfrage bearbeitet und vorübergehend keine neuen Anfragen annimmt.

Wenn Sie versuchen, Java-Anwendungen auf Wildfly-Basis mit dem Zabbix Java gateway zu überwachen, installieren Sie bitte die neueste `jboss-client.jar`, die auf der Wildfly-Downloadseite verfügbar ist.

Konfigurationsparameter: Zabbix Java gateway

Wenn Sie die Skripte `startup.sh` und `shutdown.sh` zum Starten des Zabbix Java gateway verwenden, können Sie die erforderlichen Konfigurationsparameter in der Datei `settings.sh` angeben. Die Start- und Herunterfahrskripte lesen die Einstellungsdatei ein und übernehmen die Umwandlung von Shell-Variablen (in der ersten Spalte aufgeführt) in Java-Eigenschaften (in der zweiten Spalte aufgeführt).

Wenn Sie Zabbix Java gateway manuell starten, indem Sie `java` direkt ausführen, geben Sie die entsprechenden Java-Eigenschaften in der Befehlszeile an.

Variable	Property	Mandatory	Range	Default	Description
LISTEN_IP	zabbix.listenIP	no		0.0.0.0	IP-Adresse, auf der gelauscht werden soll.
LISTEN_PORT	zabbix.listenPort	no	1024-32767	10052	Port, auf dem gelauscht werden soll.

Variable	Property	Mandatory	Range	Default	Description
PID_FILE	zabbix.pidFile	no		/tmp/zabbix_java.pid	Name der PID-Datei. Wenn sie weggelassen wird, wird Zabbix Java Gateway als Konsolenanwendung gestartet.
PROPERTIES_FILE	zabbix.propertiesFile	no			Name der Eigenschaftsdatei. Kann verwendet werden, um zusätzliche Eigenschaften im Schlüssel-Wert-Format so festzulegen, dass sie in der Befehlszeile nicht sichtbar sind, oder um vorhandene zu überschreiben. Beispiel: ”javax.net.ssl.trustStorePassword=<p
START_POLLERS	zabbix.startPollers	no	1-1000	5	Anzahl der zu startenden Worker-Threads.
TIMEOUT	zabbix.timeout	no	1-30	3	Wie lange auf Netzwerkoperationen gewartet werden soll (in Sekunden).

Warning:
Port 10052 ist nicht bei IANA registriert.

Einrichtung aus Debian-/Ubuntu-Paketen

Übersicht

Wenn die Installation aus Paketen erfolgt ist, helfen Ihnen die folgenden Informationen bei der Einrichtung des Zabbix Java gateway.

Konfigurieren und Ausführen von Java gateway

Die Konfiguration von Java gateway kann in der Datei angepasst werden:

```
/etc/zabbix/zabbix_java_gateway.conf
```

Weitere Details finden Sie unter den Parametern der Zabbix Java gateway-Konfiguration.

Um Zabbix Java gateway zu starten:

```
systemctl restart zabbix-java-gateway
```

Um Zabbix Java gateway beim Systemstart automatisch zu starten:

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

Konfigurieren des Servers für die Verwendung mit Java gateway

Wenn Java gateway ausgeführt wird, müssen Sie dem Zabbix Server mitteilen, wo er das Zabbix Java gateway findet. Dies geschieht durch Angabe der Parameter `JavaGateway` und `JavaGatewayPort` in der Server-Konfigurationsdatei. Wenn der Host, auf dem die JMX-Anwendung ausgeführt wird, von einem Zabbix Proxy überwacht wird, geben Sie die Verbindungsparameter stattdessen in der Proxy-Konfigurationsdatei an.

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

Standardmäßig startet der Server keine Prozesse, die mit der JMX-Überwachung zusammenhängen. Wenn Sie diese Funktion verwenden möchten, müssen Sie jedoch die Anzahl der vorab erzeugten Instanzen von Java-Pollern angeben. Dies tun Sie auf dieselbe Weise, wie Sie reguläre Poller und Trapper angeben.

```
StartJavaPollers=5
```

Vergessen Sie nicht, den Server oder Proxy neu zu starten, sobald Sie mit der Konfiguration fertig sind.

Debugging des Java gateway

Die Zabbix-Logdatei des Java gateway lautet:

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

Wenn Sie die Protokollierung erhöhen möchten, bearbeiten Sie die Datei:

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

und ändern Sie `info` in `level="info"` zu debug oder sogar `trace` (für eine tiefgehende Fehleranalyse):

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

</configuration>
```

JMX-Monitoring

Weitere Details finden Sie auf der Seite JMX monitoring.

Einrichtung aus den Quelltexten

Übersicht

Wenn Sie aus den Quellen installiert haben, hilft Ihnen die folgende Information bei der Einrichtung des Zabbix Java gateway.

Übersicht der Dateien

Wenn Sie das Java gateway aus den Quellen erstellt haben, sollten Sie eine Sammlung von Shell-Skripten, JAR- und Konfigurationsdateien unter `$PREFIX/sbin/zabbix_java` vorfinden. Die Funktion dieser Dateien wird im Folgenden zusammengefasst.

```
bin/zabbix-java-gateway-$VERSION.jar
```

Die JAR-Datei des Java gateway selbst.

```
lib/logback-core-1.5.25.jar
lib/logback-classic-1.5.25.jar
lib/slf4j-api-2.0.16.jar
lib/android-json-4.3_r3.1.jar
```

Abhängigkeiten des Java gateway: die Bibliotheken Logback, SLF4J und Android JSON.

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

Konfigurationsdateien für Logback.

```
shutdown.sh
startup.sh
```

Hilfsskripte zum Starten und Stoppen des Java gateway.

```
settings.sh
```

Konfigurationsdatei, die von den oben genannten Start- und Stoppskripten eingelesen wird.

Konfigurieren und Starten von Java gateway

Standardmäßig lauscht Java gateway auf Port 10052. Wenn Sie Java gateway auf einem anderen Port ausführen möchten, können Sie dies im Skript `settings.sh` angeben. Siehe die Beschreibung der Java gateway-Konfigurationsdatei, um zu erfahren, wie Sie dies und andere Optionen angeben.

> **Warning:**
> Port 10052 ist nicht bei IANA registriert.

Sobald Sie mit den Einstellungen vertraut sind, können Sie Java gateway starten, indem Sie das Skript `startup.sh` ausführen:

```
./startup.sh
```

Ebenso können Sie, wenn Sie Java gateway nicht mehr benötigen, das Skript `shutdown.sh` ausführen, um es zu stoppen:

```
./shutdown.sh
```

Beachten Sie, dass Java gateway im Gegensatz zu Server oder Proxy leichtgewichtig ist und keine Datenbank benötigt.

Konfigurieren des Servers für die Verwendung mit Java gateway

Wenn Java gateway ausgeführt wird, müssen Sie dem Zabbix Server mitteilen, wo sich das Zabbix Java gateway befindet. Dies geschieht durch Angabe der Parameter `JavaGateway` und `JavaGatewayPort` in der Serverkonfigurationsdatei. Wenn der Host, auf dem die JMX-Anwendung ausgeführt wird, von einem Zabbix Proxy überwacht wird, geben Sie die Verbindungsparameter stattdessen in der Proxykonfigurationsdatei an.

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

Standardmäßig startet der Server keine Prozesse im Zusammenhang mit der JMX-Überwachung. Wenn Sie diese Funktion jedoch verwenden möchten, müssen Sie die Anzahl der vorab erzeugten Instanzen von Java-Pollern angeben. Dies tun Sie auf dieselbe Weise, wie Sie reguläre Poller und Trapper angeben.

```
StartJavaPollers=5
```

Vergessen Sie nicht, den Server oder Proxy neu zu starten, sobald Sie mit der Konfiguration fertig sind.

Debugging Java gateway

Falls es Probleme mit Java gateway gibt oder eine Fehlermeldung, die Sie in Bezug auf einen Datenpunkt im Frontend sehen, nicht aussagekräftig genug ist, sollten Sie sich die Protokolldatei von Java gateway ansehen.

Standardmäßig protokolliert Java gateway seine Aktivitäten in die Datei `/tmp/zabbix_java.log` mit dem Protokollierungsgrad "info". Manchmal reicht diese Information nicht aus, und es werden Informationen auf dem Protokollierungsgrad "debug" benötigt. Um den Protokollierungsgrad zu erhöhen, ändern Sie die Datei `lib/logback.xml` und setzen Sie das Attribut `level` des `<root>`-Tags auf `debug`:

```
<root level="debug">
  <appender-ref ref="FILE" />
</root>
```

Beachten Sie, dass im Gegensatz zu Zabbix server oder Zabbix proxy kein Neustart von Zabbix Java gateway nach dem Ändern der Datei `logback.xml` erforderlich ist - Änderungen in `logback.xml` werden automatisch übernommen. Wenn Sie mit dem Debugging fertig sind, können Sie den Protokollierungsgrad wieder auf "info" zurücksetzen.

Wenn Sie in eine andere Datei oder in ein völlig anderes Medium wie eine Datenbank protokollieren möchten, passen Sie die Datei `logback.xml` entsprechend Ihren Anforderungen an. Weitere Details finden Sie im Logback Manual.

Manchmal ist es zu Debugging-Zwecken nützlich, Java gateway als Konsolenanwendung statt als Daemon zu starten. Dazu kommentieren Sie die Variable `PID_FILE` in `settings.sh` aus. Wenn `PID_FILE` weggelassen wird, startet das Skript `startup.sh` Java gateway als Konsolenanwendung und veranlasst Logback außerdem, die Datei `lib/logback-console.xml` zu verwenden, die nicht nur in die Konsole protokolliert, sondern auch den Protokollierungsgrad "debug" aktiviert.

Abschließend beachten Sie bitte, dass Java gateway für die Protokollierung SLF4J verwendet. Daher können Sie Logback durch ein Framework Ihrer Wahl ersetzen, indem Sie eine geeignete JAR-Datei im Verzeichnis lib ablegen. Weitere Details finden Sie im SLF4J Manual.

JMX-Überwachung

Weitere Details finden Sie auf der Seite JMX monitoring.

Einrichtung aus RHEL-Paketen

Überblick

Wenn die Installation aus Paketen erfolgt ist, helfen Ihnen die folgenden Informationen bei der Einrichtung des Zabbix Java gateway.

Konfigurieren und Ausführen von Java gateway

Die Konfigurationsparameter des Zabbix Java gateway können in der Datei angepasst werden:

```
/etc/zabbix/zabbix_java_gateway.conf
```

Weitere Informationen finden Sie unter den Konfigurations-Parametern für Zabbix Java gateway.

Um Zabbix Java gateway zu starten:

```
systemctl restart zabbix-java-gateway
```

Um Zabbix Java gateway beim Systemstart automatisch zu starten:

RHEL 7 und neuer:

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

RHEL vor 7:

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

Konfiguration des Servers für die Verwendung mit Java gateway

Wenn Java gateway ausgeführt wird, müssen Sie dem Zabbix Server mitteilen, wo sich das Zabbix Java gateway befindet. Dies geschieht durch Angabe der Parameter `JavaGateway` und `JavaGatewayPort` in der Serverkonfigurationsdatei. Wenn der Host, auf dem die JMX-Anwendung ausgeführt wird, von einem Zabbix Proxy überwacht wird, geben Sie die Verbindungsparameter stattdessen in der Proxykonfigurationsdatei an.

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

Standardmäßig startet der Server keine Prozesse im Zusammenhang mit der JMX-Überwachung. Wenn Sie diese Funktion verwenden möchten, müssen Sie jedoch die Anzahl der vorab erzeugten Instanzen von Java-Pollern angeben. Dies tun Sie auf dieselbe Weise wie bei der Angabe von regulären Pollern und Trappern.

```
StartJavaPollers=5
```

Vergessen Sie nicht, den Server oder Proxy neu zu starten, sobald Sie mit der Konfiguration fertig sind.

Debugging des Java gateway

Die Zabbix Java gateway-Protokolldatei ist:

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

Wenn Sie die Protokollierung erhöhen möchten, bearbeiten Sie die Datei:

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

und ändern Sie `info` in `level="info"` zu debug oder sogar `trace` (für eine tiefgehende Fehlerbehebung):

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

</configuration>
```

JMX-Überwachung

Weitere Details finden Sie auf der Seite JMX monitoring.

Absender

Übersicht Zabbix sender ist ein Befehlszeilenprogramm, das zum Senden von Leistungsdaten zur Verarbeitung an den Zabbix Server oder Proxy verwendet werden kann.

Das Dienstprogramm wird üblicherweise in langfristig ausgeführten Benutzerskripten zum regelmäßigen Senden von Verfügbarkeits- und Leistungsdaten verwendet.

Zum direkten Senden von Ergebnissen an den Zabbix Server oder Proxy muss ein Datenpunkt vom Typ *Zabbix trapper* konfiguriert werden.

Siehe auch:

- Zabbix-Sender-Protokoll
- Dynamische Zabbix-Sender-Bibliothek für Windows
- Python-Bibliothek für Zabbix mit integrierter Funktionalität, um wie Zabbix sender zu agieren

Ausführen von Zabbix sender Ein Beispiel für die Ausführung des Zabbix UNIX senders:

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

Dabei gilt:

- `z` - Host des Zabbix Servers (die IP-Adresse kann ebenfalls verwendet werden);
- `s` - technischer Name des überwachten Hosts (wie im Zabbix Frontend registriert);
- `k` - Schlüssel des Datenpunkts;
- `o` - zu sendender Wert.

> **Attention:**
> Optionen, die Leerzeichen enthalten, müssen in doppelte Anführungszeichen eingeschlossen werden.

Zabbix sender kann verwendet werden, um mehrere Werte aus einer Eingabedatei zu senden. Weitere Informationen finden Sie in der Manpage zu Zabbix sender.

Wenn eine Konfigurationsdatei angegeben ist, verwendet Zabbix sender alle in dem Agent-Konfigurationsparameter `ServerActive` definierten Adressen, um Daten zu senden. Wenn das Senden an eine Adresse fehlschlägt, versucht der Sender, die Daten an die anderen Adressen zu senden. Wenn das Senden von Batchdaten an eine Adresse fehlschlägt, werden nachfolgende Batches nicht an diese Adresse gesendet.

Zabbix sender akzeptiert Zeichenfolgen in UTF-8-Kodierung (sowohl für Unix-ähnliche Systeme als auch für Windows), wobei sich am Anfang der Datei keine Byte Order Mark (BOM) befinden darf.

Zabbix sender kann unter Windows ähnlich ausgeführt werden:

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

`zabbix_sender`-Szenarien für das Senden in Echtzeit sammeln mehrere Werte, die in kurzer Folge übergeben werden, und senden sie in einer einzigen Verbindung an den Server. Ein Wert, dessen Abstand zum vorherigen Wert höchstens 0,2 Sekunden beträgt, kann in denselben Stapel aufgenommen werden; die maximale Abfragezeit beträgt jedoch weiterhin 1 Sekunde.

> **Note:**
> Zabbix sender wird beendet, wenn die angegebene Konfigurationsdatei einen ungültigen Parametereintrag enthält (der nicht der Notation `parameter=value` folgt).

Ausführen von Zabbix sender mit Low-Level-Discovery Ein Beispiel für die Ausführung von Zabbix sender zum Senden eines JSON-formatierten Werts für Low-Level-Discovery:

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

Damit dies funktioniert, muss die Low-Level-Discovery-Regel den Item-Typ *Zabbix trapper* haben (in diesem Beispiel mit dem Schlüssel `trapper.discovery.item`).

Zabbix sender-Protokoll

Zabbix Server und Zabbix Proxy verwenden ein JSON-basiertes Kommunikationsprotokoll, um Daten von Zabbix sender zu empfangen. Daten können mithilfe eines trapper item oder eines HTTP agent item mit aktivierter Trapping-Funktion empfangen werden.

Anforderungs- und Antwortnachrichten müssen mit header and data length beginnen.

Zabbix sender-Anfrage

```
{
    "request": "sender data",
    "data": [
        {
            "host": "<hostname>",
            "key": "trap",
            "value": "test value"
        }
    ]
}
```

Zabbix Server-Antwort

```
{
    "response": "success",
    "info": "processed: 1; failed: 0; total: 1; seconds spent: 0.060753"
}
```

Zabbix sender-Anfrage mit Zeitstempel

Alternativ kann Zabbix sender eine Anfrage mit Zeitstempel und Nanosekunden senden.

```
{
    "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
}
```

Zabbix Server-Antwort

```
{
    "response": "success",
    "info": "processed: 2; failed: 0; total: 2; seconds spent: 0.060904"
}
```

Zabbix sender dynamische Linkbibliothek für Windows

In einer Windows-Umgebung können Anwendungen Daten an den Zabbix Server/Proxy senden, indem sie die Zabbix sender dynamische Linkbibliothek (`zabbix_sender.dll`) verwenden, anstatt einen externen Prozess (`zabbix_sender.exe`) starten zu müssen.

zabbix_sender.h und zabbix_sender.lib werden benötigt, um Benutzeranwendungen mit zabbix_sender.dll zu kompilieren.

Bezug

Es gibt zwei Möglichkeiten, zabbix_sender.dll zu erhalten.

1. Laden Sie die Dateien zabbix_sender.h, zabbix_sender.lib und zabbix_sender.dll als ZIP-Archiv herunter.

Achten Sie bei der Auswahl der Download-Optionen darauf, unter *Encryption* "No encryption" und unter *Packaging* "Archive" auszuwählen. Laden Sie dann Zabbix Agent (nicht Zabbix agent 2) herunter.

Die Dateien zabbix_sender.h, zabbix_sender.lib und zabbix_sender.dll befinden sich im heruntergeladenen ZIP-Archiv im Verzeichnis `bin\dev`. Entpacken Sie die Dateien an den gewünschten Ort.

2. Erstellen Sie zabbix_sender.dll aus dem Quellcode (siehe Anleitung).

Die dynamische Linkbibliothek mit den Entwicklungsdateien befindet sich im Verzeichnis `bin\winXX\dev`. Um sie zu verwenden, binden Sie die Header-Datei zabbix_sender.h ein und verlinken Sie mit der Bibliothek zabbix_sender.lib.

Siehe auch

- Beispiel eines einfachen Zabbix sender-Dienstprogramms, das mit der Zabbix sender dynamischen Linkbibliothek implementiert wurde, um die Verwendung der Bibliothek zu veranschaulichen;
- Datei zabbix_sender.h mit den Schnittstellenfunktionen der Zabbix sender dynamischen Linkbibliothek. Diese Datei enthält eine Dokumentation, die den Zweck jeder Schnittstellenfunktion, ihre Argumente und den Rückgabewert erläutert.

Abrufen

Übersicht Zabbix get ist ein Befehlszeilenwerkzeug, das zur Kommunikation mit dem Zabbix Agent verwendet werden kann, um die erforderlichen Informationen vom Agent abzurufen.

Das Dienstprogramm wird in der Regel zur Fehlerbehebung bei Zabbix Agenten verwendet.

Siehe auch die Python-Bibliothek für Zabbix, die über eine integrierte Funktionalität verfügt, um sich wie Zabbix get zu verhalten.

Zabbix get ausführen Ein Beispiel für die Ausführung von Zabbix get unter UNIX, um den Prozessorlastwert vom Agent abzurufen:

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

Ein weiteres Beispiel für die Ausführung von Zabbix get zum Erfassen einer Zeichenfolge von einer Website:

```
cd bin
./zabbix_get -s 192.168.1.1 -p 10050 -k "web.page.regexp[www.example.com,,,\"USA: ([a-zA-Z0-9.-]+)\",,\1]"
```

Beachten Sie, dass der Datenpunkt-Schlüssel hier ein Leerzeichen enthält, daher werden Anführungszeichen verwendet, um den Datenpunkt-Schlüssel für die Shell zu kennzeichnen. Die Anführungszeichen sind nicht Teil des Datenpunkt-Schlüssels; sie werden von der Shell entfernt und nicht an den Zabbix Agent übergeben.

Wenn ein Datenpunkt-Schlüssel nicht unterstützt wird, gibt Zabbix get den Exit-Code 1 zurück.

Zabbix get akzeptiert die folgenden Befehlszeilenparameter:

```
-s --host <host name or IP>             Hostnamen oder IP-Adresse eines Hosts angeben
-p --port <port number>                 Portnummer des auf dem Host laufenden Agent angeben (Standard: 100
-I --source-address <IP address>        Quell-IP-Adresse angeben
-t --timeout <seconds>                  Timeout angeben. Gültiger Bereich: 1-600 Sekunden (Standard: 30 Se
-k --key <item key>                     Schlüssel des abzurufenden Datenpunkts angeben
-P --protocol <value>                   Protokoll für die Kommunikation mit dem Agent. Werte:
                                            auto - Verbindung über das JSON-Protokoll herstellen, bei Fehl
                                            json - Verbindung über das JSON-Protokoll herstellen
                                            plaintext - Verbindung über das Klartextprotokoll herstellen,
-h --help                               Diese Hilfemeldung anzeigen
-V --version                            Versionsnummer anzeigen

--tls-connect <value>                   Verbindungsart zum Agent. Werte:
                                            unencrypted - ohne Verschlüsselung verbinden (Standard)
                                            psk - Verbindung über TLS und einen vorab gemeinsam genutzten
                                            cert - Verbindung über TLS und ein Zertifikat herstellen
--tls-ca-file <CA file>                 Vollständiger Pfad zu einer Datei mit den Zertifikaten der oberste
--tls-crl-file <CRL file>               Vollständiger Pfad zu einer Datei mit widerrufenen Zertifikaten
--tls-agent-cert-issuer <cert issuer>   Zulässiger Aussteller des Agent-Zertifikats
--tls-agent-cert-subject <cert subject> Zulässiger Betreff des Agent-Zertifikats
--tls-cert-file <cert file>             Vollständiger Pfad zu einer Datei mit dem Zertifikat oder der Zert
--tls-key-file <key file>               Vollständiger Pfad zu einer Datei mit dem privaten Schlüssel
--tls-psk-identity <PSK-identity>       Eindeutige, groß-/kleinschreibungssensitive Zeichenfolge zur Ident
--tls-psk-file <PSK-file>               Vollständiger Pfad zu einer Datei mit dem vorab gemeinsam genutzte
--tls-cipher13 <cipher-string>          Cipher-String für OpenSSL 1.1.1 oder neuer für TLS 1.3. Überschrei
--tls-cipher <cipher-string>            GnuTLS-Prioritätszeichenfolge (für TLS 1.2 und höher) oder OpenSSL
```

Weitere Informationen finden Sie auch auf der Zabbix get manpage.

Zabbix get kann unter Windows auf ähnliche Weise ausgeführt werden:

```
zabbix_get.exe [options]
```

JS

Übersicht zabbix_js ist ein Befehlszeilenwerkzeug, das für das Testen eingebetteter Skripte verwendet werden kann.

Dieses Werkzeug führt ein Benutzerskript mit einem String-Parameter aus und gibt das Ergebnis aus. Skripte werden mit der eingebetteten Zabbix-Skripting-Engine ausgeführt.

Bei Kompilierungs- oder Ausführungsfehlern gibt zabbix_js den Fehler auf stderr aus und beendet sich mit dem Code 1.

```
zabbix_js -s script-file -p input-param [-l log-level] [-t timeout]
zabbix_js -s script-file -i input-file [-l log-level] [-t timeout]
zabbix_js -h
zabbix_js -V
```

Verwendung zabbix_js akzeptiert die folgenden Befehlszeilenparameter:

```
-s, --script script-file          Gibt den Dateinamen des auszuführenden Skripts an
                                  Wenn als Dateiname "-" angegeben wird, wird das Skript von stdin gelesen
-i, --input input-file            Gibt den Dateinamen der Eingabeinformationen an
                                  Der Inhalt wird als Wert an das Skript übergeben
                                  Gegenseitig ausschließend mit -p
                                  Wenn als Dateiname "-" angegeben wird, wird das Skript von stdin gelesen
-p, --param input-param           Gibt den Eingabeparameter an: die Variable, die als Wert an das Skript ü
                                  Gegenseitig ausschließend mit -i
-l, --loglevel log-level          Gibt den Protokollierungsgrad an
                                  Bereich: 0-5
-t, --timeout timeout             Gibt das Zeitlimit in Sekunden an
                                  Gültiger Bereich: 1-600 Sekunden (Standard: 10 Sekunden)
-h, --help                        Zeigt Hilfeinformationen an
-V, --version                     Zeigt die Versionsnummer an
-w <webdriver url>                Aktiviert die Browserüberwachung
```

Beispiele: Beispiel 1: Ausführen eines Skripts mit einer Variable, die einen Wert enthält

Das Skript (script-file.js):

```
return value;
```

Verwendung:

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

Ausgabe: example_value

Beispiel 2: Ausführen eines Skripts mit einer Datei, die Eingabeinformationen enthält

Die Datei mit dem Skript (script-file.js):

```
return value;
```

Die Datei mit den Eingabeinformationen (example.txt):

```
Beispiel für Eingabeinformationen aus der Datei
```

Verwendung:

```
zabbix_js -s script-file.js -i example.txt
```

Ausgabe: Beispiel für Eingabeinformationen aus der Datei

Beispiel 3: Ausführen eines Skripts mit von stdin gelesener Eingabe

Die Datei mit dem Skript (script-file.js):

```
return value;
```

Verwendung:

```
zabbix_js -s script-file.js -i -
```

Eingabe (stdin):

```
Beispiel für Eingabe von stdin
```

Ausgabe: `Beispiel für Eingabe von stdin`

Webdienst

Übersicht Der Zabbix web service ist ein Prozess, der für die Kommunikation mit externen Webdiensten verwendet wird. Derzeit wird der Zabbix web service zum Erstellen und Senden von geplanten Berichten verwendet; in Zukunft ist die Ergänzung weiterer Funktionen geplant.

Der Zabbix Server verbindet sich über HTTP(S) mit dem Webdienst. Für den Zabbix web service muss Google Chrome auf demselben Host installiert sein; auf einigen Distributionen kann der Dienst auch mit Chromium funktionieren (siehe bekannte Probleme).

Installation Das offizielle Paket für den Zabbix-Webservice ist im Zabbix-Repository verfügbar.

Um den Zabbix-Webservice aus den Quellen zu kompilieren, geben Sie die Konfigurationsoption `--enable-webservice` an.

Um den Zabbix-Webservice zu konfigurieren, aktualisieren Sie die Parameter der Konfigurationsdatei des Webservice.

Attention:
Es wird dringend empfohlen, die Verschlüsselung zwischen dem Zabbix-Server und dem Zabbix-Webservice mithilfe von Zertifikaten einzurichten. Standardmäßig werden zwischen dem Zabbix-Server und dem Zabbix-Webservice übertragene Daten nicht verschlüsselt, was zu unbefugtem Zugriff führen kann.

Konfigurationsparameter: Zabbix web service

Übersicht

Der Zabbix web service ist ein Prozess, der für die Kommunikation mit externen Webdiensten verwendet wird.

Die von der Konfigurationsdatei des Zabbix web service (`zabbix_web_service.conf`) unterstützten Parameter sind in diesem Abschnitt aufgeführt.

Die Parameter werden ohne zusätzliche Informationen aufgelistet. Klicken Sie auf den Parameter, um die vollständigen Details anzuzeigen.

Parameter	Beschreibung
AllowedIP	Eine Liste durch Kommas getrennter IP-Adressen, optional in CIDR-Notation, oder DNS-Namen von Zabbix-Servern und Zabbix-Proxys.
DebugLevel	Die Debug-Stufe.
IgnoreURLCertErrors	Legt die Behandlung von TLS-Zertifikatsvalidierungsfehlern beim Zugriff auf die Frontend-URL fest.
Include	Sie können einzelne Dateien oder alle Dateien in einem Verzeichnis in die Konfigurationsdatei einbinden.
ListenPort	Der Dienst lauscht auf diesem Port auf Verbindungen vom Server.
LogFile	Der Name der Protokolldatei.
LogFileSize	Die maximale Größe der Protokolldatei.
LogType	Der Typ der Protokollausgabe.
Timeout	Die maximale Zeit (in Sekunden), die für die Formatierung des PDF-Berichts eines Dashboards aufgewendet wird.
TLSAccept	Welche eingehenden Verbindungen akzeptiert werden sollen.
TLSCAFile	Der vollständige Pfad zu einer Datei mit den Zertifikaten der obersten CA(s) zur Überprüfung von Peer-Zertifikaten, verwendet für verschlüsselte Kommunikation zwischen Zabbix-Komponenten.
TLSCertFile	Der vollständige Pfad zu einer Datei mit dem Dienstzertifikat oder der Zertifikatskette, verwendet für verschlüsselte Kommunikation zwischen Zabbix-Komponenten.
TLSKeyFile	Der vollständige Pfad zu einer Datei mit dem privaten Schlüssel des Dienstes, verwendet für verschlüsselte Kommunikation zwischen Zabbix-Komponenten.

Alle Parameter sind optional, sofern nicht ausdrücklich angegeben ist, dass der Parameter obligatorisch ist.

Beachten Sie:

- Die Standardwerte spiegeln die Prozess-Standardeinstellungen wider, nicht die Werte in den mitgelieferten Konfigurationsdateien;
- Werte unterstützen Umgebungsvariablen;
- Zabbix unterstützt Konfigurationsdateien nur in UTF-8-Kodierung ohne BOM;
- Kommentare, die mit # beginnen, werden nur am Zeilenanfang unterstützt.

Parameterdetails

AllowedIP

Eine Liste durch Kommas getrennter IP-Adressen, optional in CIDR-Notation, oder DNS-Namen von Zabbix-Servern und Zabbix-Proxys. Eingehende Verbindungen werden nur von den hier aufgeführten Hosts akzeptiert.
Wenn die IPv6-Unterstützung aktiviert ist, werden `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` gleich behandelt, und `::/0` erlaubt jede IPv4- oder IPv6-Adresse. `0.0.0.0/0` kann verwendet werden, um jede IPv4-Adresse zuzulassen.

Beispiel:

`127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com`

Pflicht: ja

DebugLevel

Geben Sie den Debug-Level an: - *0* - grundlegende Informationen über das Starten und Beenden von Zabbix-Prozessen; - *1* - kritische Informationen; - *2* - Fehlerinformationen; - *3* - Warnungen; - *4* - für das Debugging (liefert viele Informationen); - *5* - erweitertes Debugging (liefert noch mehr Informationen).

Standard: 3
 Bereich: 0-5

IgnoreURLCertErrors

Legt fest, wie TLS-Zertifikatsvalidierungsfehler beim Zugriff auf die Frontend-URL behandelt werden:
0 - Zertifikatsfehler nicht ignorieren;
1 - Zertifikatsfehler ignorieren.

Standard: 0
 Bereich: 0-1

Include

Geben Sie ein Verzeichnis oder einzelne Dateien an, die in die Konfigurationsdatei aufgenommen werden sollen.
 Während der Zabbix-Installation wird dieser Parameter auf `/usr/local/etc` gesetzt, sofern Sie dies nicht zur Kompilierzeit ändern.

Allgemeine Regeln für den Parameter `Include`:

- Der Pfad kann relativ zum Speicherort der Datei `zabbix_web_service.conf` angegeben werden.
- Sie können mehrere `Include`-Parameter in einer Konfigurationsdatei angeben.
- Zabbix verarbeitet den `Include`-Parameter, sobald er ihn liest. Platzieren Sie ihn daher am besten am Ende der Konfigurationsdatei.

Regeln für eingebundene Dateien:

- Jede Datei muss lesbar sein.
- Jede Datei muss eine korrekte Syntax haben. Andernfalls wird die Komponente nicht gestartet.

Regeln für das Einbinden von Verzeichnissen oder Dateimustern:

- Verlassen Sie sich nicht auf eine bestimmte Reihenfolge beim Einbinden (z. B. werden Dateien nicht alphabetisch eingebunden).
- Definieren Sie denselben Parameter nicht in mehr als einer eingebundenen Datei (z. B. um eine allgemeine Einstellung durch eine spezifische zu überschreiben), da Sie nicht vorhersagen können, welcher Wert verwendet wird.
- Achten Sie auf Sicherungskopien, die einige Texteditoren automatisch erstellen (z. B. kann beim Bearbeiten von `include/my_specific.conf` die Datei `include/my_specific.conf.BAK` erstellt werden), da beide Dateien eingebunden werden.

Um nur relevante Dateien aus einem Verzeichnis einzubinden, verwenden Sie den Platzhalter (`*`) für die Mustererkennung.

Beispiel:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenPort

Der Dienst lauscht auf diesem Port auf Verbindungen vom Server.

Standard: 10053
 Bereich: 1024-32767

LogFile

Der Name der Protokolldatei.

Beispiel:

`/tmp/zabbix_web_service.log`

Erforderlich: Ja, wenn LogType auf `file` gesetzt ist; andernfalls nein

LogFileSize

Die maximale Größe einer Protokolldatei in MB.
0 - automatische Protokollrotation deaktivieren.
Hinweis: Wenn das Größenlimit der Protokolldatei erreicht wird und die Dateirotation aus irgendeinem Grund fehlschlägt, wird die vorhandene Protokolldatei gekürzt und neu begonnen.

Standard: 1
 Bereich: 0-1024

LogType

Der Typ der Protokollausgabe: - *file* - Protokoll in die durch den Parameter LogFile angegebene Datei schreiben; - *system* - Protokoll in syslog schreiben; - *console* - Protokoll in die Standardausgabe schreiben.

Standardwert: `file`

Timeout

Die maximale Zeit (in Sekunden), die für die Formatierung des PDF-Berichts eines Dashboards aufgewendet wird.

Standard: 10
 Bereich: 1-30

TLSAccept

Welche eingehenden Verbindungen akzeptiert werden sollen: - *unencrypted* - Verbindungen ohne Verschlüsselung akzeptieren (Standard); - *cert* - Verbindungen mit TLS und einem Zertifikat akzeptieren.

Standard: `unencrypted`

TLSCAFile

Der vollständige Pfadname der Datei, die die Zertifikate der obersten CA(s) für die Überprüfung von Peer-Zertifikaten enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

TLSCertFile

Der vollständige Pfadname der Datei, die das Dienstzertifikat oder die Zertifikatskette enthält und für verschlüsselte Kommunikation mit Zabbix-Komponenten verwendet wird.

TLSKeyFile

Der vollständige Pfadname der Datei, die den privaten Schlüssel des Dienstes enthält und für verschlüsselte Kommunikation zwischen Zabbix-Komponenten verwendet wird.

Bridge-Adapter

Der Zabbix Bridge-Adapter ermöglicht Ihrem Zabbix Server die Kommunikation mit der Zabbix Mobile App.

Die Zabbix Mobile App verbindet sich nicht direkt mit Ihrem Zabbix Server. Stattdessen übertragen zwei Komponenten Daten zwischen ihnen:

- Zabbix Bridge – ein zentral gehosteter Dienst, der Daten mit der Zabbix Mobile App auf Ihrem Mobilgerät austauscht.
- Zabbix Bridge-Adapter – eine Komponente, die Sie installieren und ausführen. Sie verbindet Ihren Zabbix Server mit Zabbix Bridge.

Der Bridge-Adapter kann auf demselben System wie Ihr Zabbix Server oder auf einem anderen System mithilfe von Paketen installiert werden.

Um den Zabbix Bridge-Adapter zu konfigurieren, aktualisieren Sie seine Konfigurationsdatei (Standardpfad: `/etc/zabbix/zabbix_bridge_`

Informationen zum Absichern des Zabbix Bridge-Adapters finden Sie unter Sicherer mobiler Zugriff.

Datenfluss Wenn Sie die mobile Zabbix-App verwenden, werden die Daten wie folgt übertragen:

1. Sie führen eine Aktion in der mobilen Zabbix-App aus, die anschließend eine Abfrage an Zabbix Bridge sendet.
2. Zabbix Bridge übergibt die Abfrage als JSON-RPC-Anfrage an den Zabbix-Bridge-Adapter.
3. Der Zabbix-Bridge-Adapter prüft, ob die angeforderte Zabbix-API-Methode zulässig ist.
   Wenn die Methode nicht zulässig ist, weist der Bridge-Adapter die Anfrage zurück.
   Wenn die Methode zulässig ist, sendet der Bridge-Adapter die Anfrage an die Zabbix-API.
4. Die Zabbix-API verarbeitet die Anfrage und gibt eine Antwort an den Zabbix-Bridge-Adapter zurück.
5. Der Zabbix-Bridge-Adapter gibt die Antwort an Zabbix Bridge zurück.
6. Zabbix Bridge übergibt die Antwort an die mobile Zabbix-App, die die Daten auf Ihrem Mobilgerät anzeigt.

Konfigurationsparameter: Zabbix-Bridge-Adapter

Übersicht

Konfigurationsparameter ermöglichen die Anpassung des Verhaltens des Zabbix-Bridge-Adapters.

Geben Sie die Parameterwerte in der Konfigurationsdatei (`zabbix_bridge_adapter.conf`) an. Alle Parameter sind **optional**, sofern nicht ausdrücklich angegeben ist, dass der Parameter obligatorisch ist.

Diese Seite ist wie folgt strukturiert:

- Parameterübersicht (um vollständige Details zu einem Parameter anzuzeigen, klicken Sie auf seinen Namen)
- Parameterdetails

Parameter	Beschreibung
AllowedAPIMethods	Geben Sie an, welche Zabbix-API-Methoden über den Zabbix-Bridge-Adapter aufgerufen werden können.
DataDir	Geben Sie das Verzeichnis an, in dem die Zustandsdatei und die Verschlüsselungsschlüssel des Bridge-Adapters erstellt und gespeichert werden.
DebugLevel	Legen Sie die Debug-Stufe fest.
Include	Geben Sie ein Verzeichnis oder einzelne Dateien an, die in die Konfigurationsdatei aufgenommen werden sollen.
ListenIP	Geben Sie die IP-Adresse an, unter der der Bridge-Adapter eingehende HTTP(S)-Verbindungen vom Zabbix-Server entgegennimmt.
ListenHealthPort	Geben Sie den Port an, unter dem der Bridge-Adapter eingehende HTTP-Verbindungen für Zustandsprüfungen entgegennimmt.
ListenPort	Geben Sie den Port an, unter dem der Bridge-Adapter eingehende HTTP(S)-Verbindungen vom Zabbix-Server entgegennimmt.
LogFile	Geben Sie den vollständigen Pfad zur Protokolldatei des Bridge-Adapters an.
LogFileSize	Legen Sie die maximale Größe der Protokolldatei (in MB) fest.
LogType	Geben Sie den Typ der Protokollausgabe an.
PidFile	Geben Sie die PID-Datei an.
TLSConfig.AllowedCN	Geben Sie eine durch Kommas getrennte Liste der zulässigen Common Names von Clients an (nur `mTLS`).
TLSConfig.CAFile	Geben Sie den vollständigen Pfad zu einer Datei an, die das bzw. die übergeordnete(n) CA-Zertifikat(e) enthält, die zur Überprüfung von Clientzertifikaten im `mTLS`-Modus verwendet werden.
TLSConfig.CertFile	Geben Sie den vollständigen Pfad zu einer Datei an, die das Zertifikat oder die Zertifikatskette enthält.
TLSConfig.KeyFile	Geben Sie den vollständigen Pfad zu einer Datei an, die den privaten Schlüssel enthält.
TLSConfig.Mode	Legen Sie den TLS-Modus für eingehende Verbindungen an `ListenPort` fest.
ServerId	Geben Sie den eindeutigen Bezeichner des Zabbix-Servers an. **Obligatorisch**.
ZabbixAPITLSCAFile	Geben Sie den vollständigen Pfadnamen einer Datei an, die das bzw. die übergeordnete(n) CA-Zertifikat(e) enthält, die der Bridge-Adapter zur Überprüfung des Zabbix-API-Zertifikats verwendet.
ZabbixAPITLSConnect	Legen Sie den TLS-Modus für ausgehende Verbindungen zur Zabbix-API fest.
ZabbixAPIURL	Geben Sie die URL des Zabbix-API-Endpunkts an. **Obligatorisch**.
ZabbixBridgeURL	Geben Sie die Zabbix-Bridge-URL für den Abruf der Zabbix-Bridge-Konfiguration an (der Abruf erfolgt beim Start des Bridge-Adapters).

Beachten Sie Folgendes:

- Die Standardwerte entsprechen den Prozessstandards, nicht den Werten in den mitgelieferten Konfigurationsdateien.
- Die Werte unterstützen Umgebungsvariablen.
- Zabbix unterstützt Konfigurationsdateien nur in UTF-8-Kodierung ohne BOM.
- Kommentare, die mit # beginnen, werden nur am Anfang der Zeile unterstützt.

Parameterdetails

AllowedAPIMethods

Geben Sie an, welche Zabbix-API-Methoden über den Zabbix bridge adapter aufgerufen werden können.
 Für mehrere Methoden wiederholen Sie den Parameter, eine Methode pro Zeile.

Jeder Parameterwert muss entweder ein exakter Methodenname (z. B. `problem.get`) oder ein Platzhaltermuster sein, das nur den gesamten API-Namen (z. B. `*.get`) oder den gesamten Methodennamen (z. B. `problem.*`) ersetzt.

Jeder andere Wert ist ungültig und verhindert den Start des Zabbix bridge adapter. Dies umfasst einen alleinstehenden Platzhalter (`*`), ein partielles Platzhaltermuster (`host*` oder `host*.*`) oder einen leeren Wert (`AllowedAPIMethods=`).

Beispiel:

```
AllowedAPIMethods=apiinfo.version
AllowedAPIMethods=problem.*
AllowedAPIMethods=*.get
```

Der Mindestsatz an Methoden, den die mobile Zabbix-App benötigt:

```
AllowedAPIMethods=apiinfo.version
AllowedAPIMethods=device.onboard
AllowedAPIMethods=device.offboard
AllowedAPIMethods=user.get
AllowedAPIMethods=role.get
AllowedAPIMethods=problem.get
AllowedAPIMethods=event.get
AllowedAPIMethods=event.acknowledge
AllowedAPIMethods=trigger.get
AllowedAPIMethods=alert.get
AllowedAPIMethods=host.get
AllowedAPIMethods=hostgroup.get
```

Standard: nicht festgelegt (alle Zabbix-API-Methoden sind nicht zulässig)

DataDir

Geben Sie das Verzeichnis an, in dem die Statusdatei des Bridge-Adapters und die Verschlüsselungsschlüssel erstellt und gespeichert werden.
 Das Verzeichnis muss über die Berechtigungen 0700 verfügen, und alle darin gespeicherten Dateien müssen über die Berechtigungen 0600 verfügen.

Standard: `/var/lib/zabbix-bridge-adapter`

DebugLevel

Legen Sie die Debug-Stufe fest:

- 0 – grundlegende Informationen zum Starten und Beenden von Zabbix-Prozessen
- 1 – kritische Informationen
- 2 – Fehlerinformationen
- 3 – Warnungen
- 4 – zum Debuggen (erzeugt viele Informationen)
- 5 – erweitertes Debuggen (erzeugt noch mehr Informationen)

Standard: 3
 Bereich: 0-5

Include

Geben Sie ein Verzeichnis oder einzelne Dateien an, die in die Konfigurationsdatei aufgenommen werden sollen.
 Bei der Installation von Zabbix wird dieser Parameter auf `/usr/local/etc` gesetzt, sofern Sie dies nicht während der Kompilierung ändern.

Allgemeine Regeln für den Parameter `Include`:

- Der Pfad kann relativ zum Speicherort der Datei `zabbix_bridge_adapter.conf` angegeben werden.
- Sie können mehrere `Include`-Parameter in einer Konfigurationsdatei angeben.

- Zabbix verarbeitet den Parameter `Include`, sobald er gelesen wird. Ziehen Sie daher in Betracht, ihn am Ende der Konfigurationsdatei zu platzieren.

Regeln für eingeschlossene Dateien:

- Jede Datei muss lesbar sein.
- Jede Datei muss eine korrekte Syntax aufweisen. Andernfalls wird die Komponente nicht gestartet.

Regeln für das Einschließen von Verzeichnissen oder Dateimustern:

- Gehen Sie nicht von einer bestimmten Reihenfolge beim Einschließen aus (Dateien werden beispielsweise nicht alphabetisch eingeschlossen).
- Definieren Sie denselben Parameter nicht in mehreren eingeschlossenen Dateien (beispielsweise, um eine allgemeine Einstellung durch eine spezifische zu überschreiben), da Sie nicht vorhersagen können, welcher Wert verwendet wird.
- Achten Sie auf Sicherungskopien, die einige Texteditoren automatisch erstellen (beispielsweise kann beim Bearbeiten von `include/my_specific.conf` die Datei `include/my_specific.conf.BAK` erstellt werden), da beide Dateien eingeschlossen werden.

Um nur relevante Dateien aus einem Verzeichnis einzuschließen, verwenden Sie den Platzhalter (`*`) für den Musterabgleich.

Beispiel:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenIP

Geben Sie die IP-Adresse an, an der der Bridge-Adapter auf eingehende HTTP(S)-Verbindungen vom Zabbix Server wartet.
 Wenn nicht angegeben, wartet der Bridge-Adapter an allen IP-Adressen.

ListenHealthPort

Geben Sie den Listen-Port des Bridge-Adapters für eingehende HTTP-Health-Check-Verbindungen an.
 Stellt die Endpunkte `/health` und `/ready` bereit.
 Wenn nicht angegeben, wird der Health-Service nicht gestartet.

Standard: 8081
 Bereich: 1024-32767

ListenPort

Geben Sie den Listen-Port des Bridge-Adapters für eingehende HTTP(S)-Verbindungen vom Zabbix Server an.

Standard: 10054
 Bereich: 1024-32767

LogFile

Geben Sie den vollständigen Pfad zur Protokolldatei des Bridge-Adapters an.

Obligatorisch: **Ja**, wenn LogType auf `file` gesetzt ist

Standard: `/tmp/zabbix_bridge_adapter.log`

LogFileSize

Legt die maximale Größe der Protokolldatei (in MB) fest.
 0 - deaktiviert die automatische Protokollrotation.

Standard: 1
 Bereich: 0-1024

LogType

Geben Sie den Typ der Protokollausgabe an:

- `file` - schreibt das Protokoll in die durch den Parameter LogFile angegebene Datei.
- `system` - schreibt das Protokoll in Syslog.
- `console` - schreibt das Protokoll in die Standardausgabe.

Standard: `file`

PidFile

Geben Sie die PID-Datei an.

Standard: `/tmp/zabbix_bridge_adapter.pid`

ServerId

Geben Sie den eindeutigen Bezeichner des Zabbix-Servers an.
 Verfügbar unter *Berichte* > *Systeminformationen* in der Zabbix-Benutzeroberfläche.

Obligatorisch: **Ja**

TLSConfig.AllowedCN

Geben Sie eine durch Kommas getrennte Liste der zulässigen Common Names der Clients an (nur mTLS).
 Wenn nicht angegeben, überprüft der Bridge-Adapter den Common Name des Clients nicht.

TLSConfig.CAFile

Geben Sie den vollständigen Pfad zu einer Datei an, die das/die übergeordnete(n) CA-Zertifikat(e) enthält, die zur Überprüfung von Clientzertifikaten im mTLS-Modus verwendet werden.

TLSConfig.CertFile

Geben Sie den vollständigen Pfad zu einer Datei an, die das Zertifikat oder die Zertifikatskette enthält.

TLSConfig.KeyFile

Geben Sie den vollständigen Pfad zu einer Datei an, die den privaten Schlüssel enthält.

TLSConfig.Mode

Legen Sie den TLS-Modus für eingehende Verbindungen an `ListenPort` fest:

- `none` - Verbindungen ohne Verschlüsselung akzeptieren.
- `tls` - Standard-TLS mit mindestens v1.2.
- `mtls` - erweitert TLS um gegenseitige Authentifizierung.

Standard: `none`

ZabbixAPITLSCAFile

Geben Sie den vollständigen Pfadnamen einer Datei an, die das bzw. die übergeordneten CA-Zertifikat(e) enthält, die der Bridge-Adapter zur Überprüfung des Zabbix-API-Zertifikats verwendet.

Obligatorisch: **Ja**, wenn `ZabbixAPITLSConnect` auf `verify_ca` oder `verify_full` gesetzt ist.

ZabbixAPITLSConnect

Legt den TLS-Modus für ausgehende Verbindungen zur Zabbix-API fest:

- `disabled` (oder leer) – verwendet einfaches HTTP; `ZabbixAPIURL` muss `http://` verwenden.
- `required` – verwendet HTTPS ohne Zertifikatsüberprüfung (nur zu Testzwecken).
- `verify_ca` – verwendet HTTPS und überprüft das Zabbix-API-Zertifikat.
- `verify_full` – verwendet HTTPS und überprüft das Zabbix-API-Zertifikat sowie den Hostnamen.

ZabbixAPIURL

Geben Sie die URL des Zabbix-API-Endpunkts an.

Obligatorisch: **Ja**

Beispiel:

```
ZabbixAPIURL=https://example.com/zabbix/api_jsonrpc.php
```

ZabbixBridgeURL

Geben Sie die Zabbix-Bridge-URL für den Abruf der Zabbix-Bridge-Konfiguration an (wird beim Start des Bridge-Adapters abgerufen).

Standard: `ZabbixBridgeURL=https://test.aws.zabbix-lab.win/.well-known/bridge-configuration`

MCP-Server

Übersicht Der Zabbix-MCP-Server ist ein Prozess, der es KI-Agenten ermöglicht, mithilfe des Model Context Protocol (MCP) mit Zabbix zu arbeiten. Der MCP-Server kommuniziert mit MCP-Clients und der Zabbix-API.

Mit dem MCP-Server kann ein KI-Agent:

- Hostgruppen, Vorlagengruppen, Hosts, Vorlagen, Datenpunkte, LLD-Regeln, Schnittstellen, Makros, Auslöser, Probleme, Wartungszeiträume und Daten aus dem Datenpunktverlauf abrufen (mit Ausnahme binärer Datenpunkte);
- Probleme aktualisieren: Kommentare hinzufügen, bestätigen oder die Bestätigung aufheben, den Schweregrad ändern, unterdrücken oder die Unterdrückung aufheben oder Probleme schließen;
- Wartungszeiträume erstellen, aktualisieren und löschen.

Zabbix-API-Tokens werden zur Autorisierung verwendet; die Tokens werden mit jeder Anfrage gesendet. Der Zugriff ist durch die Berechtigungen des Tokenbesitzers eingeschränkt: Nur Daten aus Hostgruppen, die für diesen Benutzer verfügbar sind, sind sichtbar, und Aktionen, zu deren Ausführung der Benutzer nicht berechtigt ist, werden abgelehnt.

Der MCP-Server ist zustandslos: Zwischen Anfragen wird keine Sitzung aufrechterhalten.

> **Note:**
> Der Zabbix-MCP-Server funktioniert möglicherweise mit älteren oder neueren Zabbix-Versionen. Die Kompatibilität wird jedoch offiziell nur für dieselbe Zabbix-Version unterstützt, die auch der MCP-Server selbst verwendet.

Siehe auch:

- Konfigurationsparameter: MCP-Server
- Tools und Ressourcen

Installation Das offizielle Paket zabbix-mcp-server ist im Zabbix-Repository verfügbar.

Um den Zabbix MCP server aus den Quellen zu installieren, erstellen Sie ihn mit `make`.

Um den Zabbix MCP server zu konfigurieren, aktualisieren Sie die Parameter der Konfigurationsdatei des MCP servers. Mindestens die Parameter `FrontendURL` und `AllowedIP` müssen festgelegt werden.

Die standardmäßige Konfigurationsdatei (`zabbix_mcp_server.conf`) befindet sich unter `/etc/zabbix/`; sie verweigert den Zugriff auf die Tools, die am häufigsten sensible Daten offenlegen. Um ein Tool zu aktivieren, entfernen Sie die entsprechende Zeile `DenyTool`.

Siehe auch: Bewährte Sicherheitspraktiken

Ausführung Um den MCP-Server zu starten, führen Sie folgenden Befehl aus:

```
zabbix_mcp_server -c /etc/zabbix/zabbix_mcp_server.conf
```

Um die Konfigurationsdatei zu überprüfen und den Vorgang zu beenden, ohne den Server zu starten, verwenden Sie die Option `-T`:

```
zabbix_mcp_server -c /etc/zabbix/zabbix_mcp_server.conf -T
```

Die folgenden Befehlszeilenoptionen werden unterstützt:

Option	Beschreibung
`-c <file>`	Der Pfad zur Konfigurationsdatei. Der Standardwert ist `/etc/zabbix/zabbix_mcp_server.conf`.
`-T`	Die Konfigurationsdatei validieren und den Vorgang beenden.
`-h`	Hilfeinformationen anzeigen.
`-V`	Die Versionsnummer anzeigen.

Wenn der MCP-Server aus den Quellen installiert wird, muss eine systemd-Unit manuell erstellt werden, zum Beispiel:

```
[Unit]
Description=Zabbix MCP server
After=network-online.target

[Service]
User=zabbix
ExecStart=/usr/local/sbin/zabbix_mcp_server -c /etc/zabbix/zabbix_mcp_server.conf
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

Verbinden eines MCP-Clients Jeder MCP-Client, der den Streamable-HTTP-Transport unterstützt, kann mit dem MCP-Server verbunden werden. Die folgenden Einstellungen sind erforderlich:

Einstellung	Wert
Transport	Streamable HTTP

Einstellung	Wert
URL	`https://<host>:<ListenPort>/mcp`
Header	`Authorization: Bearer <Zabbix API token>`, mit jeder Anfrage gesendet

Jeder Benutzer muss sein eigenes API-Token konfigurieren.

Um die Einrichtung zu überprüfen, senden Sie eine einzelne Anfrage an den MCP-Server:

```
curl -sS https://mcp.example.com:8443/mcp \
  -H "Authorization: Bearer $ZABBIX_API_TOKEN" \
  -H "Content-Type: application/json" \
  -H "Accept: application/json, text/event-stream" \
  -d '{"jsonrpc":"2.0","id":1,"method":"tools/list"}'
```

Der HTTP-Statuscode 200 und eine Liste der verfügbaren Tools in der Antwort bestätigen, dass der MCP-Server erreichbar ist und das Token akzeptiert; die Liste zeigt, welche Tools verfügbar sind. Andere Statuscodes weisen auf die folgenden Probleme hin:

- 400 - der `Accept`-Header enthält nicht beide vom Transport benötigten Medientypen;
- 401 - der `Authorization`-Header mit dem Token fehlt;
- 403 - die Clientadresse ist nicht im Parameter `AllowedIP` aufgeführt.

Genehmigung und Tool-Zugriff Wenn der MCP-Client die Formularabfrage unterstützt, muss jede Änderung vom Benutzer genehmigt werden, bevor sie an Zabbix gesendet wird. Der MCP-Client zeigt dem Benutzer an, was geändert wird, und die Änderung wird nur angewendet, wenn der Benutzer sie bestätigt. Wenn der Benutzer die Anfrage ablehnt oder verwirft, wird in Zabbix nichts geändert. Eine Genehmigung ist fünf Minuten lang und nur für die exakt angezeigte Änderung gültig. Innerhalb dieses Zeitraums kann die Genehmigung wiederverwendet werden.

Wenn der MCP-Client keine Formularabfrage unterstützt (einschließlich Clients, die nur URL-Abfragen unterstützen), werden Änderungen sofort und ohne Genehmigung angewendet. Beschränken Sie in diesem Fall die Schreib-Tools und die Berechtigungen der Zabbix-API.

Eine Genehmigung ist besonders wichtig, da die Tool-Ausgabe nicht vertrauenswürdigen Text aus überwachten Systemen enthalten kann, der einen Agent beeinflussen könnte.

Verwenden Sie AllowTool, DenyTool, AllowToolRegexp und DenyToolRegexp, um zu steuern, welche Tools der MCP-Server bereitstellt. Verweigerte Tools können weder aufgerufen noch aufgelistet werden, und auf die zugehörigen Ressourcen kann nicht zugegriffen werden.

> **Warning:**
> Die Regeln AllowTool, DenyTool, AllowToolRegexp und DenyToolRegexp beschränken den MCP-Zugriff, ändern jedoch nicht die Berechtigungen des Zabbix-Tokens. Wenn die Frontend-Adresse einem KI-Agent offengelegt wird, kann der Agent versuchen, direkte API-Aufrufe zu senden. Halten Sie den Speicherort des Zabbix-Frontends und die Konfiguration des MCP-Servers für den Agent unzugänglich, damit Anfragen über den MCP-Server und dessen Sicherheitskontrollen geleitet werden müssen.

Best Practices für die Sicherheit API-Tokens

Authentifizierung und Autorisierung werden vollständig von Zabbix übernommen. Der MCP-Client sendet mit jeder Anfrage ein Zabbix-API-Token; der MCP-Server leitet es weiter, ohne es zu speichern oder zu protokollieren, während Zabbix das Token validiert, Berechtigungen durchsetzt und Aktionen im Audit-Protokoll erfasst.

Verwenden Sie für jeden Agent und Benutzer ein separates, mit den geringstmöglichen Berechtigungen ausgestattetes und ablaufendes Token. Speichern Sie Tokens an einem sicheren Ort und wechseln Sie API-Tokens regelmäßig.

TLS-Verschlüsselung

Um zu verhindern, dass Token im Klartext übertragen werden, setzen Sie `TLSAccept=cert` und geben Sie das Zertifikat sowie den Schlüssel (`TLSCertFile` und `TLSKeyFile`) an, oder betreiben Sie den MCP-Server hinter einem vertrauenswürdigen TLS-terminierenden Reverse-Proxy und setzen Sie `TLSAccept=unencrypted`. Unverschlüsseltes HTTP ist nur in einem vertrauenswürdigen Netzwerk akzeptabel.

Zugriffsbeschränkungen

Beschränken Sie den Parameter `AllowedIP` auf die Adressen vertrauenswürdiger MCP-Clients und verwenden Sie Zabbix-Berechtigungen, um einzuschränken, was API-Token tun können.

Der MCP-Server bietet keine Ratenbegrenzung für Anfragen. Jede MCP-Anfrage führt normalerweise zu einem Zabbix-API-Aufruf, obwohl ein in einer Schleife ausgeführter Agent dennoch eine erhebliche Last erzeugen kann.

Schutz sensibler Daten

Datenpunktschlüssel können Zugangsdaten enthalten (zum Beispiel in SSH-, JMX-, ODBC- und HTTP-Agent-Datenpunkten), Makrowerte können Passwörter und Community-Strings enthalten, Schnittstellen geben interne Adressen preis, und Werte von Protokoll- oder Textdatenpunkten können beliebige Daten enthalten. Aus diesem Grund sind die Tools `item_get`, `lld_get`, `interface_get`, `macro_get` und `history_get` in der ausgelieferten Konfigurationsdatei standardmäßig deaktiviert.

Um zu verhindern, dass sensible Makrowerte dem LLM-Anbieter des verbundenen Agents offengelegt werden, verwenden Sie geheime Makros; ihre Werte werden von der API niemals zurückgegeben.

Konfigurationsparameter: Zabbix MCP-Server

Übersicht

Der Zabbix MCP-Server ist ein Prozess, der es KI-Agenten ermöglicht, mithilfe des Model Context Protocol (MCP) mit Zabbix zu arbeiten.

Die vom Zabbix MCP-Server in der Konfigurationsdatei (`zabbix_mcp_server.conf`) unterstützten Parameter sind in diesem Abschnitt aufgeführt.

Die Parameter werden ohne zusätzliche Informationen aufgeführt. Klicken Sie auf den Parameter, um die vollständigen Details anzuzeigen.

Parameter	Beschreibung
AllowedIP	Eine durch Kommas getrennte Liste von IP-Adressen, optional in CIDR-Notation, oder DNS-Namen von MCP-Clients.
AllowTool	Eine Wildcard-Regel, die ein Tool zulässt.
AllowToolRegexp	Eine Regel mit regulärem Ausdruck, die Tools zulässt.
DebugLevel	Die Debug-Stufe.
DenyTool	Eine Wildcard-Regel, die ein Tool ablehnt.
DenyToolRegexp	Eine Regel mit regulärem Ausdruck, die Tools ablehnt.
FrontendURL	Der Zabbix-API-Endpunkt.
IgnoreURLCertErrors	Gibt an, wie Validierungsfehler von TLS-Zertifikaten beim Zugriff auf die Zabbix-API behandelt werden.
ListenIP	Die IP-Adresse, an der auf Verbindungen gewartet wird.
ListenPort	Der Port, an dem auf Verbindungen gewartet wird.
LogFile	Der Name der Protokolldatei.
LogFileSize	Die maximale Größe der Protokolldatei.
LogType	Der Typ der Protokollausgabe.
MaxResponseSize	Die maximale Größe einer von einem Lesetool zurückgegebenen Antwort.
PidFile	Der Name der PID-Datei.
Timeout	Die maximale Zeit (in Sekunden), die auf eine Antwort von der Zabbix-API gewartet wird.
TLSAccept	Welche eingehenden Verbindungen akzeptiert werden.
TLSCertFile	Der vollständige Pfadname einer Datei, die das Serverzertifikat oder die Zertifikatskette enthält.
TLSKeyFile	Der vollständige Pfadname einer Datei, die den privaten Serverschlüssel enthält.
ToolPrefix	Ein Präfix, das den Namen aller Tools vorangestellt wird.

Alle Parameter sind optional, sofern nicht ausdrücklich angegeben ist, dass der Parameter obligatorisch ist.

Beachten Sie Folgendes:

- Die Standardwerte entsprechen den Prozessstandards und nicht den Werten in den mitgelieferten Konfigurationsdateien.
- Werte unterstützen Umgebungsvariablen im Format `${NAME}`.
- Zabbix unterstützt Konfigurationsdateien nur in der UTF-8-Kodierung ohne BOM.
- Kommentare, die mit # beginnen, werden nur am Anfang der Zeile unterstützt.

Parameterdetails

AllowedIP

Eine Liste durch Kommas getrennter IP-Adressen, optional in CIDR-Notation, oder DNS-Namen von MCP-Clients. Eingehende Verbindungen werden nur von den hier aufgeführten Hosts akzeptiert.

Beispiel:

```
AllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,mcp-client.example.com
```

Obligatorisch: ja

AllowTool

Eine Wildcard-Regel, die ein Tool zulässt. Der Parameter kann mehrfach angegeben werden.
 Wildcard-Muster verwenden die Syntax des Datenpunktschlüssels, wobei * für beliebige Zeichen steht. Die Regel stimmt mit dem einfachen Toolnamen überein und ignoriert ToolPrefix.
 Alle Regeln `AllowTool`, `DenyTool`, `AllowToolRegexp` und `DenyToolRegexp` bilden eine einzige Liste in der Reihenfolge, in der sie angegeben werden; die erste übereinstimmende Regel entscheidet. Ein Tool, auf das keine Regel zutrifft, ist zulässig.

Beispiel:

```
AllowTool=host_get
```

AllowToolRegexp

Eine Regel mit regulären Ausdrücken, die Tools zulässt. Der Parameter kann mehrfach angegeben werden.
 Reguläre Ausdrücke verwenden die RE2-Syntax und stimmen mit jedem Teil des einfachen Toolnamens überein, sofern sie nicht verankert sind.

Beispiel:

```
AllowToolRegexp=^(host|hostgroup|trigger|problem)_get$
```

DebugLevel

Geben Sie die Debug-Stufe an: - *0* – grundlegende Informationen zum Starten und Beenden von Zabbix-Prozessen; - *1* – kritische Informationen; - *2* – Fehlerinformationen; - *3* – Warnungen; - *4* – zum Debuggen (erzeugt viele Informationen); - *5* – erweitertes Debugging (erzeugt noch mehr Informationen).

Standard: 3
 Bereich: 0-5

DenyTool

Eine Wildcard-Regel, die ein Tool verweigert. Der Parameter kann mehrfach angegeben werden.
 Ein verweigertes Tool wird nicht in die Liste der Tools aufgenommen und kann nicht aufgerufen werden. Weitere Informationen finden Sie unter AllowTool.

Beispiel:

```
DenyTool=maintenance_delete
```

DenyToolRegexp

Eine Regel mit regulärem Ausdruck, die Tools ablehnt. Der Parameter kann mehrfach angegeben werden.
 Weitere Informationen finden Sie unter AllowToolRegexp.

Beispiel:

```
DenyToolRegexp=^problem_(close|suppress)$
```

FrontendURL

Der Endpunkt der Zabbix-API: die URL des Zabbix-Frontends, die mit `api_jsonrpc.php` endet.

Beispiel:

```
FrontendURL=https://zabbix.example.com/api_jsonrpc.php
```

Obligatorisch: ja

IgnoreURLCertErrors

Gibt an, wie TLS-Zertifikatsvalidierungsfehler beim Zugriff auf die Zabbix-API behandelt werden:
 false – Zertifikatsfehler nicht ignorieren;
 true – Zertifikatsfehler ignorieren (nur in einer Entwicklungsumgebung verwenden).

Standard: `false`

ListenIP

Die IP-Adresse, auf der Verbindungen entgegengenommen werden.

Standard: `0.0.0.0`

ListenPort

Der Port, an dem auf Verbindungen gelauscht wird. Der MCP-Endpunktpfad ist `/mcp`.

Standard: 8443
 Bereich: 1024-32767

LogFile

Der Name der Protokolldatei.

Standardwert: `/tmp/zabbix_mcp_server.log`

Obligatorisch: Nein

LogFileSize

Die maximale Größe einer Protokolldatei in MB.
0 – automatische Protokollrotation deaktivieren.

Standard: 1
 Bereich: 0-1024

LogType

Der Typ der Protokollausgabe: - *file* – schreibt das Protokoll in die durch den Parameter LogFile angegebene Datei; - *system* – schreibt das Protokoll in syslog; - *console* – schreibt das Protokoll in die Standardausgabe.

Standard: `file`

MaxResponseSize

Die maximale Größe einer von einem Lesetool zurückgegebenen Antwort. Sobald dieses Limit oder das `limit`-Argument des Tools zuerst erreicht wird, wird die Antwort beendet.
 Unterstützt Einheitensuffixe.

Standard: `64K`
 Bereich: 16K-256K

PidFile

Der Name der PID-Datei.

Standard: `/tmp/zabbix_mcp_server.pid`

Zeitüberschreitung

Die maximale Zeit (in Sekunden), die auf eine Antwort von der Zabbix API gewartet wird.

Standard: 10
 Bereich: 3-30

TLSAccept

Welche eingehenden Verbindungen akzeptiert werden: - *unverschlüsselt* – Verbindungen ohne Verschlüsselung akzeptieren (Standard); - *Zertifikat* – Verbindungen mit TLS und einem Zertifikat akzeptieren.

Standard: `unencrypted`

TLSCertFile

Der vollständige Pfadname der Datei, die das Serverzertifikat oder die Zertifikatskette enthält.

Obligatorisch: Ja, wenn `TLSAccept` auf `cert` gesetzt ist; andernfalls nein

TLSKeyFile

Der vollständige Pfadname der Datei, die den privaten Schlüssel des Servers enthält.

Obligatorisch: Ja, wenn `TLSAccept` auf `cert` gesetzt ist; andernfalls nein

ToolPrefix

Ein Präfix, das den Namen aller Tools vorangestellt wird, beispielsweise um Namenskonflikte mit Tools anderer MCP-Server zu vermeiden. Mit `ToolPrefix=zbx_` wird das Tool `host_get` beispielsweise als `zbx_host_get` bereitgestellt.
 Zulässige Zeichen: Buchstaben, Ziffern, Unterstrich (`_`), Bindestrich (`-`) und Punkt (`.`).
 Zugriffsregeln für Tools entsprechen den Tool-Namen ohne Präfix.

Beispiel:

`ToolPrefix=zbx_`

Tools und Ressourcen

Übersicht

Der Zabbix MCP server stellt Zabbix-Funktionen für MCP-Clients als *Tools* und *Ressourcen* bereit.

Die Tool-Namen beschreiben die Aktivität. Wenn der Parameter ToolPrefix festgelegt ist, wird das Präfix jedem Tool-Namen vorangestellt; die Tool-Zugriffsregeln stimmen mit den unten aufgeführten einfachen Namen überein.

Lesewerkzeuge

Werkzeug	Zabbix-API-Methode	Argumente	Sortiert nach	Ressource
hostgroup_get	hostgroup.get	groupids, names, search, limit	name	zabbix://hostgroup/<i
templategroup_get	templategroup.get	groupids, names, search, limit	name	zabbix://templategrou
host_get	host.get	hostids, groupids, search, limit	name	zabbix://host/<id>
template_get	template.get	templateids, groupids, search, limit	name	zabbix://template/<id
item_get	item.get	hostids, itemids, search, limit	name	zabbix://item/<id>
lld_get	discoveryrule.get	hostids, ruleids, limit	name	-
interface_get	hostinterface.get	hostids, interfaceids, limit	Schnittstellen-ID	-
macro_get	usermacro.get	hostids oder global, macro_names, limit	Makro	-
problem_get	problem.get	eventids, groupids, hostids, severities, acknowledged, suppressed, recent, limit	Ereignis-ID, absteigend	-
maintenance_get	maintenance.get	maintenanceids, groupids, hostids, limit	name	-
history_get	history.get	itemids, time_from, time_till, type, limit	clock, absteigend	-
trigger_get	trigger.get	triggerids, hostids, search, limit	Beschreibung	zabbix://trigger/<id>

Beachten Sie Folgendes:

- host_get und template_get geben auch die Auslöser jedes Objekts (eine begrenzte Anzahl von Feldern), Tags und verknüpfte Vorlagen zurück. Datenpunkte, Discovery-Regeln, Schnittstellen und Makros werden von den entsprechenden Werkzeugen zurückgegeben.
- history_get erfordert einen Zeitraum von höchstens 7 Tagen, der frühestens vor 7 Tagen beginnt. Das Argument type muss float, unsigned, character, text, log oder json sein und mit dem Werttyp der Datenpunkte übereinstimmen. Bei einem falschen Typ wird ein leeres Ergebnis zurückgegeben, das nicht von einem Datenpunkt ohne Daten unterschieden werden kann. Der Verlauf binärer Datenpunkte (Werttyp 5) kann nicht über den MCP-Server abgerufen werden.
- Lesewerkzeuge fordern eine feste Liste von Feldern an; alle Felder (output: extend) werden niemals angefordert.
- Die Ergebnisse von Werkzeugen mit einer Ressource enthalten das Feld uri, sodass ein Agent von einer Liste zu einer einzelnen Entität wechseln kann.

Antwortformat

Alle Lese-Tools geben die Antwort im selben Format zurück:

```
{
  "results": [ ... ],
  "count": 2,
  "truncated": true,
  "truncated_by": "count",
  "limit": 2,
  "note": "The result count limit was reached ..."
}
```

Eine Antwort wird durch das Argument limit (Standard: 100, Maximum: 1000) und durch die serialisierte Größe der Antwort (MaxResponseSize, Standard: 64K) begrenzt. Die Antwort wird beendet, sobald eines der Limits erreicht ist. Die Antwort wird nur zwischen Ergebnissen gekürzt, niemals innerhalb eines Ergebnisses. Eine gekürzte Antwort gibt an, welches Limit erreicht wurde. Jedes Tool verfügt über eine feste Sortierreihenfolge, daher enthält eine gekürzte Antwort immer den Anfang des vollständigen Ergebnissatzes. Paging wird nicht unterstützt. Um andere Ergebnisse abzurufen, schränken Sie die Abfrage ein.

Problem-Tools

Alle Problem-Tools rufen event.acknowledge mit den entsprechenden Aktionsflags auf.
Das Argument eventids ist erforderlich; Ereignis-IDs können mit problem_get abgerufen werden.
Mit jedem Problem-Tool kann eine Nachricht (Kommentar) hinzugefügt werden.
Das Argument message ist optional, außer bei problem_annotate, wo es erforderlich ist; wenn es angegeben wird, wird die Nachricht im selben Zabbix-API-Aufruf aufgezeichnet.

Tool	Effekt	Zusätzliche Argumente
problem_acknowledge	Das Problem bestätigen.	-
problem_unacknowledge	Die Bestätigung entfernen.	-
problem_close	Das Problem manuell schließen (nur für Auslöser, die das manuelle Schließen erlauben).	-

Tool	Effekt	Zusätzliche Argumente
problem_annotate	Eine Nachricht aufzeichnen.	message (erforderlich)
problem_change_severity	Den Schweregrad des Problems ändern.	severity (0-5)
problem_suppress	Das Problem bis zu einem bestimmten Zeitpunkt oder auf unbestimmte Zeit unterdrücken.	suppress_until
problem_unsuppress	Die Unterdrückung entfernen.	-

Probleme können nicht erstellt oder gelöscht werden; alle Problem-Tools aktualisieren vorhandene Probleme.

Wartungswerkzeuge

Werkzeug	Zabbix-API-Methode	Argumente
maintenance_create	maintenance.create	name, groupids und/oder hostids, active_since, active_till, timeperiods, description, tags, tags_evaltype
maintenance_update	maintenance.update	maintenanceid und beliebige der oben genannten Argumente. Listenargumente ersetzen den gesamten Wertesatz; eine leere Liste löscht den Wertesatz.
maintenance_delete	maintenance.delete	maintenanceids

Wenn ein Wartungszeitraum aktualisiert wird, ersetzen Listenargumente (z. B. hostids, groupids und timeperiods) die vorhandenen Werte, anstatt sie zu ergänzen. Wenn beispielsweise ein Wartungszeitraum die Hosts A und B umfasst und maintenance_update mit hostids aufgerufen wird, das nur Host C enthält, umfasst der Wartungszeitraum anschließend nur Host C.

maintenance_create ist die einzige Operation, die nicht idempotent ist, und maintenance_delete ist die einzige Operation, die nicht rückgängig gemacht werden kann. Um ihre Verwendung zu verhindern, verweigern Sie diese Werkzeuge mit dem Parameter DenyTool.

Verhalten von Schreibwerkzeugen

Problem- und Wartungswerkzeuge sind *Schreibwerkzeuge*. Sie funktionieren wie folgt:

- Die Argumente werden vor jedem Zabbix-API-Aufruf validiert. Wenn die Validierung fehlschlägt, nennt der Fehler die nicht erfüllte Einschränkung.
- Wenn der Client dies unterstützt, wird die Änderung erst angewendet, nachdem sie vom Benutzer genehmigt wurde.
- Die Änderung wird von Zabbix unter Verwendung des Tokens des Aufrufers durchgeführt. Zabbix prüft die Berechtigungen und protokolliert die Änderung im Audit-Log für den Besitzer des Tokens.
- Zabbix-Fehler, einschließlich Berechtigungsfehlern, werden unverändert als Fehlerobjekt mit den Feldern code, message und data zurückgegeben.
- Schreibwerkzeuge geben das Zabbix-Ergebnis direkt zurück; die Antwortlimits von Lesewerkzeugen gelten nicht.
- Schreibaufrufe werden vom MCP-Server niemals wiederholt.

Ressourcen

Ressourcen verwenden das URI-Schema zabbix://<kind>/<id>. Nur zabbix://about ist in der Antwort von resources/list enthalten; Entitäts-IDs können nicht aufgelistet werden und werden aus den Ergebnissen der Tools abgerufen. Die Antwort von resources/templates/list listet die Arten von Entitäten auf.

URI	Inhalt
zabbix://about	Die API-URL, die Frontend-URL, die Zabbix-Version, die Identität, auf die das Token des Aufrufers aufgelöst wird, und die Liste der bereitgestellten Tools.
zabbix://host/<hostid>	Der Host-Datensatz ohne Datenpunkte, Auslöser, Schnittstellen und Makros.
zabbix://hostgroup/<groupid>	Der Hostgruppen-Datensatz ohne die Liste der Mitglieder.
zabbix://template/<templateid>	Der Vorlagen-Datensatz ohne Datenpunkte, Auslöser und Discovery-Regeln.
zabbix://templategroup/<groupid>	Der Vorlagengruppen-Datensatz ohne die Liste der Mitglieder.
zabbix://item/<itemid>	Die Definition des Datenpunkts ohne den aktuellen Wert.
zabbix://trigger/<triggerid>	Die Definition des Auslösers ohne Hosts, Abhängigkeiten, Funktionen und Tags.

Eine Ressource, deren zugehöriges Tool verweigert wird, existiert nicht. Eine nicht vorhandene Entität und eine Entität, auf die der Eigentümer des Tokens keinen Zugriff hat, werden beide als *Ressource nicht gefunden* gemeldet; diese Fälle sind absichtlich nicht voneinander zu unterscheiden. Die Ressource zabbix://about ist von den Zugriffsregeln für Tools nicht betroffen.

Umgebungsvariablen

Übersicht Umgebungsvariablen ermöglichen die Konfiguration von Zabbix-Komponenten, ohne Werte in Konfigurationsdateien fest zu kodieren. Dadurch lassen sich Konfigurationen in dynamischen Umgebungen wie Docker einfach verwalten, in denen Variablen zur Laufzeit übergeben werden können, um sich an unterschiedliche Setups anzupassen.

Im einfachsten Fall können Sie den Wert des Konfigurationsparameters DebugLevel des Zabbix Server auf eine Umgebungsvariable setzen und diese dann verwenden, um den Server beim Start zu konfigurieren:

```
### Zabbix Server-Konfigurationsdatei:
DebugLevel=${NEW_DEBUG_LEVEL}

### Zabbix Server starten:
NEW_DEBUG_LEVEL=5 /usr/sbin/zabbix_server
```

Umgebungsvariablen werden von den folgenden Zabbix-Komponenten unterstützt:

- Server
- Proxy
- Agent (UNIX oder Windows)
- Agent 2 (UNIX oder Windows), einschließlich Plugins
- Webdienst
- Zabbix sender (bei Verwendung der Option -c, --config)

Wichtige Hinweise

- Wenn ein Konfigurationsparameter auf eine Umgebungsvariable gesetzt ist, die beim Starten der Komponente nicht angegeben wird, wird der Standardwert des Parameters verwendet.
- Bei der Verwendung von Laufzeitbefehlen (z. B. um den Log-Level des Agent zu erhöhen) müssen alle zuvor verwendeten Umgebungsvariablen angegeben werden. Der Grund dafür ist, dass Zabbix-Komponenten ihre Konfigurationsdatei verwenden, um Laufzeitbefehle auszuführen; wenn Umgebungsvariablen weggelassen werden, werden die Standardwerte der Konfigurationsparameter verwendet. Siehe Beispiele.
- Der Laufzeitbefehl `userparameter_reload` unterstützt das erneute Laden von Umgebungsvariablen nicht. Beim Neuladen werden Variablen ignoriert, und nur Parameter mit regulären Werten werden neu geladen.
- Die aktuellen Umgebungsvariablen des Prozesses, die in Konfigurationsdateien verwendet wurden, werden nach dem Start der Zabbix-Komponente gelöscht. Dadurch wird sichergestellt, dass Kindprozesse (z. B. von Zabbix ausgeführte Remote-Skripte) nicht auf diese Variablen zugreifen können. Beachten Sie jedoch, dass die anfänglichen Prozessvariablen weiterhin abgerufen werden können (z. B. über die Datei `/proc/<PID>/environ`).

Syntax Umgebungsvariablen müssen die folgende Syntax verwenden: `${alphanumerics/underscores}`.

Der Variablenname darf nur Buchstaben (a-z, A-Z), Unterstriche (_) und Ziffern (0-9) enthalten und darf nicht mit einer Ziffer beginnen.

Variablen, die nicht der erforderlichen Syntax entsprechen oder mit einem regulären Wert kombiniert werden, werden als reguläre Werte behandelt, was zu Fehlern führen kann.

Korrekte Variablensyntax:

```
DebugLevel=${NEW_DEBUG_LEVEL}
Hostname=${ZBX_HOSTNAME}
LogFile=${LogFile_001}
```

Falsche Variablensyntax:

```
DebugLevel=${5_DebugLevel}
Hostname=${ZBX.HOSTNAME 1}
LogFile=/${HOME}/zabbix/zabbix_server.log
```

> **Note:**
> Unter Windows sind die Namen von Umgebungsvariablen nicht groß-/kleinschreibungsempfindlich.

Beispiele Die folgenden Beispiele zeigen, wie Umgebungsvariablen mit Zabbix-Komponenten konfiguriert und verwendet werden.

Beispiel 1: Konfigurieren und Testen des Zabbix Agent

1. Legen Sie Umgebungsvariablen in der Agent-Konfigurationsdatei fest:

```
Hostname=${ZBX_HOSTNAME}
ServerActive=${ServerActive}
```

2. Testen Sie die Konfigurationsdatei:

```
ZBX_HOSTNAME="New Zabbix agent" ServerActive=127.0.0.1 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agent
```

3. Starten Sie den Agent mit Umgebungsvariablen:

```
ZBX_HOSTNAME="New Zabbix agent" ServerActive=127.0.0.1 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agent
```

Bei der Verwendung von Laufzeitbefehlen (z. B. zum Erhöhen der Agent-Protokollstufe) müssen alle zuvor verwendeten Umgebungsvariablen angegeben werden:

```
ZBX_HOSTNAME="New Zabbix agent" ServerActive=127.0.0.1 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agent
```

Der Grund dafür ist, dass der Agent seine Konfigurationsdatei zum Ausführen von Laufzeitbefehlen verwendet. Wenn Umgebungsvariablen weggelassen werden, werden die Standardwerte der Konfigurationsparameter verwendet.

Alternativ können Sie die Umgebungsvariablen nach dem Festlegen in der Agent-Konfigurationsdatei für Prozesse verfügbar machen (z. B. mit dem Befehl `export`). Dadurch wird das Risiko unerwarteten Verhaltens aufgrund fehlender oder falsch festgelegter Variablen verringert.

```
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
```

Beispiel 2: Konfigurieren des Zabbix Agent für einen Container

Wenn Sie Ihr eigenes benutzerdefiniertes Image für Zabbix-Komponenten (z. B. den Zabbix Agent) erstellen und konfigurieren, können Sie Konfigurationsparameter mithilfe von Umgebungsvariablen definieren und den Container anschließend mit diesen Variablen starten.

1. Legen Sie beim Vorbereiten des Images Umgebungsvariablen in der Agent-Konfigurationsdatei fest:

```
Hostname=${ZBX_HOSTNAME}
BufferSize=${BUFSZ}
ListenPort=${LISTENPORT}
UserParameter=${_UsrPar01}
UserParameter=${_UsrPar02}
```

2. Starten Sie nach dem Erstellen des Container-Images den Agent-Container (z. B. Docker) mit Umgebungsvariablen:

```
docker run --name my-zabbix-agent -e ZBX_HOSTNAME="new-hostname" -e BUFSZ=1000 -e LISTENPORT=20050 -e _Usr
```

3. Wenn Sie Laufzeitbefehle verwenden (z. B. um die Agent-Protokollebene zu erhöhen), greifen Sie auf die Container-Shell zu und führen Sie den Laufzeitbefehl aus:

```
docker exec -it <containerid> sh
/usr/sbin/zabbix_agentd -R log_level_increase
```

Attention:
Der Laufzeitbefehl `userparameter_reload` unterstützt das erneute Laden von Umgebungsvariablen nicht. Beim erneuten Laden werden Variablen ignoriert, und nur Parameter mit regulären Werten werden erneut geladen.

Konfiguration

Übersicht Die Konfiguration in Zabbix umfasst die Festlegung, welche Hosts und Systeme überwacht werden sollen, welche Daten erfasst werden und wie Benachrichtigungen zugestellt werden, wenn Probleme auftreten.

Verwenden Sie die Seitenleiste, um zu den relevanten Konfigurationsbereichen zu navigieren.

Wichtige Konfigurationsaufgaben

- Einen neuen Host oder eine Hostgruppe einrichten → Hosts and host groups
- Festlegen, welche Daten von Hosts erfasst werden sollen → Items
- Regeln zum Erkennen von Problemen erstellen → Problem detection with triggers
- Konfigurieren, wie Benachrichtigungen zugestellt werden → Notifications upon events
- Überwachungsdaten über Dashboards, Diagramme und Karten anzeigen → Visualization
- Konfigurationen über mehrere Hosts hinweg wiederverwenden → Templates and template groups, Templates out of the box
- Benutzerzugriff und Berechtigungen steuern → Users and user groups

Best Practices für die Konfiguration Konfigurationsentscheidungen beeinflussen Leistung und Wartbarkeit. Auch wenn jede Zabbix-Umgebung anders ist, können diese Best Practices als Grundlage dienen und bei Bedarf angepasst werden.

Separater Host für jede Monitoring-Instanz

Ein Host in Zabbix ist eine logische Einheit und nicht unbedingt ein physischer Rechner oder ein physisches Gerät. Erstellen Sie für jede unabhängige Instanz einen separaten Host, z. B. für eine virtuelle Maschine, eine Datenbank, einen Container oder einen Netzwerkswitch.

Dieser Ansatz bietet:

1. Sauberere Monitoring-Daten mit getrennten Datenpunkten, Auslösern und Alarmbenachrichtigungen für jeden Host.
2. Eine fein abgestufte Benutzerzugriffssteuerung über user-roles, abgestimmt auf das Prinzip der geringsten Berechtigung.

Vorlagen und Massenaktualisierung für ähnliche Hosts verwenden

Für ähnliche Hosts wie *Network switch 1* und *Network switch 2* können Sie eine Vorlage mit den erforderlichen Metriken (*Network switch template*) erstellen und die Hosts in einer Host-Gruppe (*Network switches*) zusammenfassen.

Beim Hinzufügen neuer Switches können Sie in *Data Collection* > *Hosts* die Massenaktualisierung verwenden, um die Vorlage zu verknüpfen, anstatt Hosts manuell zu klonen.

Master- und abhängige Datenpunkte konfigurieren, um Anfragen zu minimieren

Um die Anzahl der Anfragen an die Zielentität zu minimieren, konfigurieren Sie einen Master-Datenpunkt, der Daten in einer einzigen Anfrage erfasst, und verwenden Sie dann abhängige Datenpunkte mit Vorverarbeitung, um bestimmte Werte zu extrahieren.

Der Master-Datenpunkt kann beispielsweise eine JSON- oder XML-Antwort mit mehreren Metriken erfassen oder eine Datenbankabfrage ausführen, die mehrere Spalten zurückgibt (z. B. Anzahl offener Verbindungen, abgebrochene Verbindungen, maximal zulässige gleichzeitige Verbindungen und die seit dem Start insgesamt kumulierte Anzahl von Verbindungen), und die abhängigen Datenpunkte parsen und speichern dann jeden Wert separat.

Verwerfen Sie den Verlauf des Master-Datenpunkts nach der Erfassung und behalten Sie nur die Daten der abhängigen Datenpunkte.

Bei Bedarf einen Zabbix-Proxy bereitstellen

Wenn sich alle Ihre Hosts im selben lokalen Netzwerk wie der Zabbix-Server befinden und keine Bedenken hinsichtlich Skalierbarkeit oder Leistung bestehen, benötigen Sie möglicherweise keinen Proxy. In größeren oder komplexeren Umgebungen ermöglicht ein Proxy eine gleichmäßigere Lastverteilung.

Fügen Sie einen Zabbix-Proxy hinzu, wenn:

1. Sie mehrere Hosts mit verschiedenen Methoden zur Metrikerfassung hinter einer Firewall überwachen. Der Proxy sammelt Daten von Hosts und leitet sie an den Zabbix-Server weiter, wodurch die Anzahl der geöffneten Firewall-Ports reduziert wird.
2. Sie Remote-Standorte, Zweigstellen oder Netzwerke überwachen. Zabbix-Proxys an Remote-Standorten setzen die Datensammlung während Netzwerkunterbrechungen fort und senden die Daten zurück, sobald die Verbindung wiederhergestellt ist.
3. Sie umfangreiche Bereitstellungen verwalten. Wenn Sie viele Hosts haben und/oder eine hohe Anzahl von Werten pro Sekunde erfassen, stellen Sie einen Proxy bereit, um die Serverlast zu reduzieren und die Leistung zu verbessern.

Unabhängige Abrufung geheimer Makros konfigurieren

Sie können geheime Benutzermakros entweder als geheimen Text oder als geheime Vault-Makros verwenden.

Standardmäßig werden die Werte geheimer Makros vom Zabbix Server abgerufen und an Proxies weitergegeben. Für erhöhte Sicherheit bei geheimen Vault-Makros konfigurieren Sie, dass die Makrowerte unabhängig vom Zabbix Server und von Proxies abgerufen werden.

Konfigurieren einer Vorlage

Übersicht

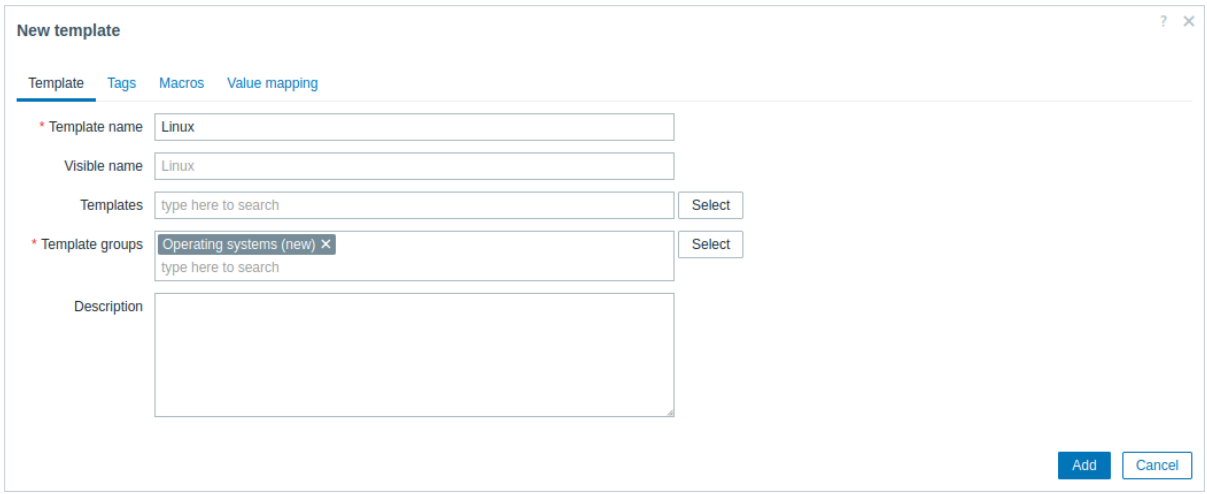
Zum Konfigurieren einer Vorlage müssen Sie zunächst eine Vorlage erstellen, indem Sie ihre allgemeinen Parameter definieren, und ihr anschließend Entitäten (Datenpunkte, Auslöser, Graphen usw.) hinzufügen.

Erstellen einer Vorlage

Gehen Sie zum Erstellen einer Vorlage wie folgt vor:

1. Gehen Sie zu *Datensammlung* > *Vorlagen*.
2. Klicken Sie auf *Vorlage erstellen*.
3. Bearbeiten Sie die Vorlagenattribute.

Die Registerkarte **Vorlage** enthält allgemeine Vorlagenattribute.



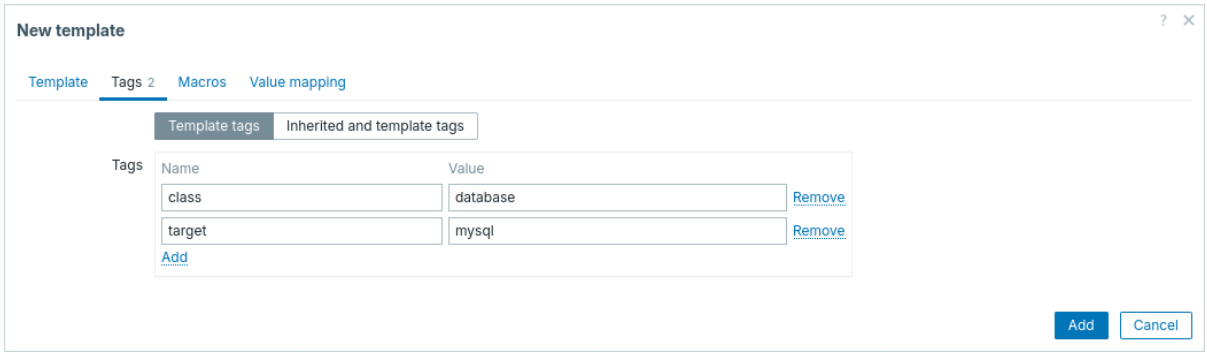
Alle Pflichtfelder sind mit einem roten Sternchen gekennzeichnet.

Vorlagenattribute:

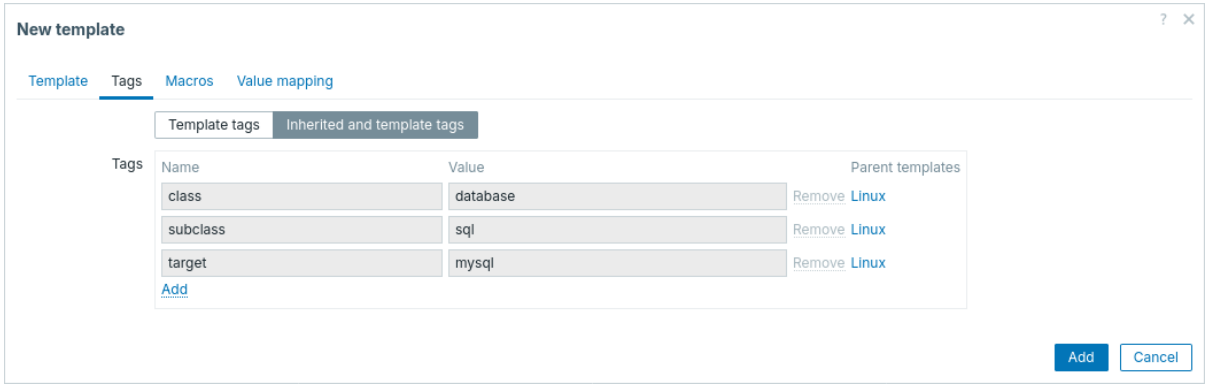
Parameter	Beschreibung
Vorlagenname	Eindeutiger Vorlagenname. Alphanumerische Zeichen, Leerzeichen, Punkte, Bindestriche und Unterstriche sind zulässig. Führende und nachgestellte Leerzeichen sind nicht zulässig.
Anzeigename	Wenn Sie diesen Namen festlegen, wird er in Listen, Karten usw. angezeigt.
Vorlagen	Verknüpfen Sie eine oder mehrere Vorlagen mit dieser Vorlage. Alle Entitäten (Datenpunkte, Auslöser usw.) werden von den verknüpften Vorlagen übernommen. Um eine neue Vorlage zu verknüpfen, geben Sie ihren Namen in das Feld *Vorlagen* ein. Daraufhin wird eine Dropdown-Liste mit passenden Vorlagen angezeigt. Alternativ können Sie auf *Auswählen* klicken, um das Pop-up-Fenster *Vorlagen* zu öffnen, das zunächst keine Vorlagen enthält. Um die Liste zu füllen, geben Sie entweder einen Namen einer Vorlagengruppe in das Feld *Vorlagengruppe* ein (und wählen Sie eine Gruppe aus der Vorschlagsliste aus) oder klicken Sie neben dem Feld *Vorlagengruppe* auf die Schaltfläche *Auswählen*, um das Pop-up-Fenster *Vorlagengruppen* zu öffnen. Sobald Sie eine Vorlagengruppe auswählen, wird das Pop-up-Fenster *Vorlagen* aktualisiert und zeigt die zu dieser Gruppe gehörenden Vorlagen an. Die ausgewählten Vorlagen werden verknüpft, wenn die Konfiguration gespeichert oder aktualisiert wird. Um die Verknüpfung mit einer Vorlage aufzuheben, verwenden Sie eine der beiden Optionen im Block *Vorlagen*: *Verknüpfung aufheben* – hebt die Verknüpfung mit der Vorlage auf, behält jedoch deren Entitäten (Datenpunkte, Auslöser usw.) bei; *Verknüpfung aufheben und leeren* – hebt die Verknüpfung mit der Vorlage auf und entfernt alle ihre Entitäten (Datenpunkte, Auslöser usw.).
Vorlagengruppen	Vorlagengruppen, zu denen die Vorlage gehört.
Beschreibung	Beschreibung der Vorlage.

Parameter	Beschreibung
Hersteller und Version	Hersteller und Version der Vorlage; wird nur beim Aktualisieren vorhandener Vorlagen angezeigt (Out-of-the-box-Vorlagen von Zabbix, importierte Vorlagen oder über die Template API geänderte Vorlagen), wenn die Vorlagenkonfiguration diese Informationen enthält. Kann im Zabbix Frontend nicht geändert werden. Bei Out-of-the-box-Vorlagen wird die Version wie folgt angezeigt: Hauptversion von Zabbix, Trennzeichen ("-"), Revisionsnummer (wird mit jeder neuen Version der Vorlage erhöht und mit jeder Hauptversion von Zabbix zurückgesetzt). Beispiele: 7.0-0, 7.0-5, 8.0-0, 8.0-3.

Auf der Registerkarte **Tags** können Sie Vorlagen-Tags auf Vorlagenebene definieren. Alle Probleme von Hosts, die mit dieser Vorlage verknüpft sind, werden mit den hier eingegebenen Werten getaggt.



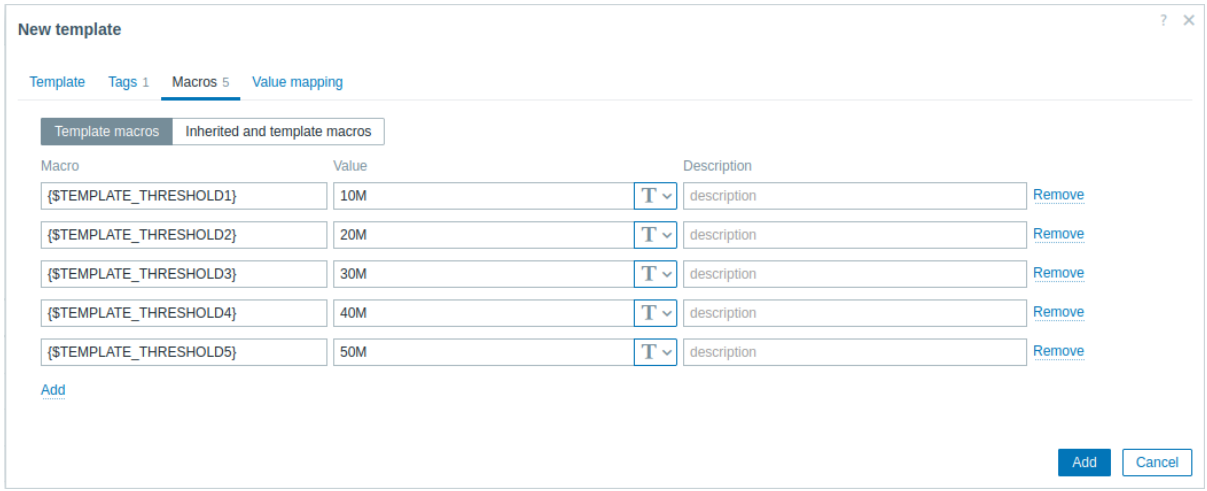
Wenn Sie die Option *Geerbte und Vorlagen-Tags* auswählen, können Sie auch Tags aus verknüpften Vorlagen anzeigen, die von dieser Vorlage übernommen werden.



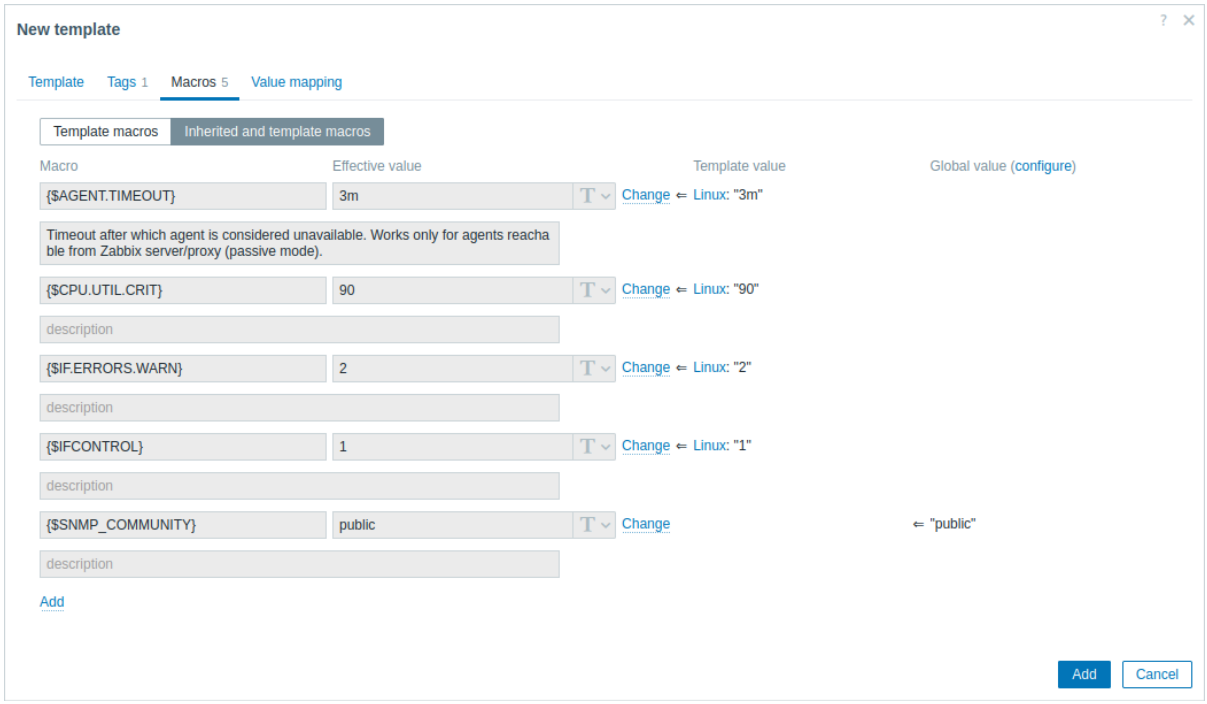
Zur Vereinfachung werden Links zu den jeweiligen Vorlagen bereitgestellt.

Benutzermakros, {INVENTORY.*}-Makros, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} und {HOST.ID}-Makros werden in Tags unterstützt.

Auf der Registerkarte **Makros** können Sie Vorlagen-Benutzermakros als Name-Wert-Paare definieren. Beachten Sie, dass Makrowerte als Klartext, geheimer Text oder Vault-Geheimnis gespeichert werden können. Das Hinzufügen einer Beschreibung wird ebenfalls unterstützt.



Wenn Sie die Option *Geerbte und Vorlagenmakros* auswählen, können Sie auch Makros aus verknüpften Vorlagen und globale Makros anzeigen, die von dieser Vorlage übernommen werden, sowie die Werte, in die die Makros aufgelöst werden.



Zur Vereinfachung werden Links zu den jeweiligen Vorlagen sowie ein Link zur Konfiguration globaler Makros bereitgestellt. Es ist auch möglich, ein Makro einer verknüpften Vorlage oder ein globales Makro auf Vorlagenebene zu bearbeiten und dadurch eine Kopie des Makros in der Vorlage zu erstellen.

Auf der Registerkarte **Wertzuordnung** können Sie eine benutzerfreundliche Darstellung von Daten aus Datenpunkten in Wertzuordnungen konfigurieren.

Schaltflächen:

Add	Fügt die Vorlage hinzu. Die hinzugefügte Vorlage sollte in der Liste angezeigt werden.
Update	Aktualisiert die Eigenschaften einer vorhandenen Vorlage.
Clone	Erstellt eine weitere Vorlage auf Grundlage der Eigenschaften der aktuellen Vorlage. Dies **umfasst** die Entitäten (Datenpunkte, Auslöser usw.), die sowohl von verknüpften Vorlagen übernommen als auch direkt an die aktuelle Vorlage angehängt wurden, **schließt jedoch** Hersteller und Version der aktuellen Vorlage für die geklonte Vorlage aus, damit diese von der ursprünglichen Vorlage unterschieden werden kann.
Delete	Löscht die Vorlage; Entitäten der Vorlage (Datenpunkte, Auslöser usw.) verbleiben bei den verknüpften Hosts.
Delete and clear	Löscht die Vorlage und alle ihre Entitäten von den verknüpften Hosts.
Cancel	Bricht die Bearbeitung der Vorlageneigenschaften ab.

Hinzufügen von Datenpunkten, Auslösern, Diagrammen

Attention:
Datenpunkte müssen zuerst zu einer Vorlage hinzugefügt werden. Auslöser und Diagramme können nicht ohne den entsprechenden Datenpunkt hinzugefügt werden.

Sie können vorhandene Datenpunkte zu einer Vorlage hinzufügen oder neue Datenpunkte erstellen.

So fügen Sie Datenpunkte zur Vorlage hinzu:

1. Gehen Sie zu *Datenerfassung* > *Hosts* (oder *Vorlagen*).
2. Klicken Sie in der Zeile des gewünschten Hosts/der gewünschten Vorlage auf *Datenpunkte*.
3. Markieren Sie die Kontrollkästchen der Datenpunkte, die Sie zur Vorlage hinzufügen möchten.
4. Klicken Sie unterhalb der Datenpunktliste auf *Kopieren*.
5. Wählen Sie die Vorlage (oder die Vorlagengruppe) aus, in die die Datenpunkte kopiert werden sollen, und klicken Sie auf *Kopieren*.
 Alle ausgewählten Datenpunkte sollten in die Vorlage kopiert werden.

Das Hinzufügen von Auslösern und Diagrammen erfolgt auf ähnliche Weise (jeweils aus der Liste der Auslöser bzw. Diagramme), wobei erneut zu beachten ist, dass sie nur hinzugefügt werden können, wenn die erforderlichen Datenpunkte zuerst hinzugefügt wurden.

Dashboards hinzufügen

Um Dashboards zu einer Vorlage unter *Datenerfassung* > *Vorlagen* hinzuzufügen, gehen Sie wie folgt vor:

1. Klicken Sie in der Zeile der Vorlage auf *Dashboards*.
2. Konfigurieren Sie ein Dashboard gemäß den Richtlinien zum Konfigurieren von Dashboards.

Attention:
Beim Konfigurieren von Widgets auf einem Vorlagen-Dashboard (statt auf einem globalen Dashboard) sind die hostbezogenen Parameter nicht verfügbar, und einige Parameter haben eine andere Bezeichnung. Das liegt daran, dass Vorlagen-Dashboards nur Daten von dem Host anzeigen, mit dem die Vorlage verknüpft ist. Beispielsweise sind die Parameter *Host groups*, *Exclude host groups* und *Hosts* im Widget *Problems* nicht verfügbar, der Parameter *Host groups* im Widget *Host availability* ist nicht verfügbar, und der Parameter *Show hosts in maintenance* wurde in *Show data in maintenance* umbenannt usw. Weitere Informationen zur Verfügbarkeit von Parametern in Vorlagen-Dashboard-Widgets finden Sie bei den spezifischen Parametern der einzelnen Dashboard-Widgets.

Note:
Einzelheiten zum Zugriff auf Host-Dashboards, die aus Vorlagen-Dashboards erstellt werden, finden Sie im Abschnitt Host-Dashboards.

Konfigurieren von Low-Level-Discovery-Regeln

Siehe den Abschnitt Low-Level-Discovery im Handbuch.

Hinzufügen von Webszenarien

Um Webszenarien zu einer Vorlage unter *Datenerfassung* > *Vorlagen* hinzuzufügen, gehen Sie wie folgt vor:

1. Klicken Sie in der Zeile der Vorlage auf *Web*.
2. Konfigurieren Sie ein Webszenario nach der üblichen Methode zum Konfigurieren von Webszenarien.

Konfigurieren einer Vorlagengruppe

Übersicht

Vorlagengruppen werden für die logische Gruppierung von Vorlagen und die Zuweisung von Benutzerberechtigungen zu ihnen verwendet.

Jeder Vorlage muss mindestens eine Vorlagengruppe zugewiesen sein. Eine Vorlage kann mehreren Vorlagengruppen angehören, und jede Vorlagengruppe kann mehrere Vorlagen enthalten.

Beachten Sie, dass in Zabbix alle Berechtigungen auf Gruppen basieren: Benutzergruppen, Hostgruppen und Vorlagengruppen. Selbst wenn ein einzelner Benutzer Zugriff auf eine einzelne Vorlage benötigt, wird dieser gewährt, indem der Benutzer zu einer Benutzergruppe hinzugefügt und dieser Benutzergruppe Berechtigungen für die Vorlagengruppe zugewiesen werden, die diese Vorlage enthält.

Konfiguration

Attention:
Nur Superadmin-Benutzer können Vorlagengruppen erstellen.

Es gibt zwei Möglichkeiten, eine Vorlagengruppe im Zabbix Frontend zu erstellen.

Option eins:

1. Gehen Sie zu *Datenerfassung* > *Vorlagengruppen*.
2. Klicken Sie oben rechts auf dem Bildschirm auf *Vorlagengruppe erstellen*.
3. Geben Sie den Gruppennamen im Formular ein.



Option zwei: Wenn Sie eine Vorlage konfigurieren, geben Sie im Eingabefeld *Vorlagengruppen* einen nicht vorhandenen Gruppennamen ein.

Sobald die Vorlagengruppe erstellt wurde, können Sie auf den Vorlagennamen in der Liste unter *Datenerfassung* > *Vorlagengruppen* klicken, um den Gruppennamen zu bearbeiten, die Gruppe zu klonen oder die Gruppe zu löschen.

Das Löschen einer Vorlagengruppe entfernt nur die logische Gruppe, nicht die Vorlagen in der Gruppe. Es ist nicht möglich, eine Vorlagengruppe zu löschen, die für eine vorhandene Vorlage die einzige Gruppe ist.

Erstellen von Untergruppen für Vorlagen

Eine Untergruppe für Vorlagen (oder verschachtelte Vorlagengruppe) ist ein untergeordnetes Element der übergeordneten Vorlagengruppe, die sie enthält.

Eine Untergruppe wird erstellt, indem im Eingabefeld für den Gruppennamen der Schrägstrich '/' verwendet wird, um ihre Beziehung zu den übergeordneten Gruppen anzugeben. Zum Beispiel:

- die Eingabe von `Linux servers/Databases` erstellt die Untergruppe `Linux servers/Databases` der übergeordneten Gruppe `Linux servers`.
- die Eingabe von `Linux servers/Databases/MySQL/Tokyo` erstellt die entsprechende Untergruppe innerhalb der verschachtelten übergeordneten Gruppen `Linux servers`, `Linux servers/Databases`, `Linux servers/Databases/MySQL`.

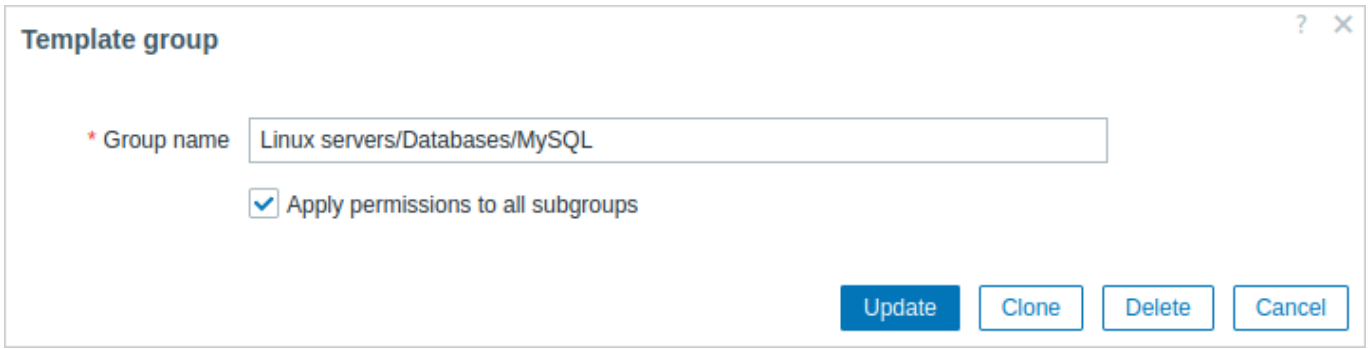
Beim Erstellen einer Untergruppe ist die Verwendung von führenden oder nachgestellten Schrägstrichen sowie mehrerer Schrägstriche hintereinander nicht zulässig. Das Escaping von '/' wird nicht unterstützt.

Es ist nicht erforderlich, vor dem Erstellen einer Untergruppe übergeordnete Vorlagengruppen zu erstellen. Sie können wählen, ob Sie mit dem Erstellen einer Untergruppe beginnen (zum Beispiel `Linux servers/Databases`) oder mit einer beliebigen übergeordneten Vorlagengruppe (in unserem Beispiel `Linux servers`). Wenn Sie mit dem Erstellen einer Untergruppe beginnen, werden übergeordnete Vorlagengruppen **nicht** automatisch erstellt.

Berechtigungen für Vorlagengruppen

- Beim Erstellen einer Untergruppe zu einer vorhandenen übergeordneten Vorlagengruppe (zum Beispiel beim Erstellen von `Linux servers/Databases`, wenn `Linux servers` bereits vorhanden ist) werden die Benutzergruppen-Berechtigungen für die Untergruppe vom übergeordneten Element übernommen.
- Beim Erstellen einer übergeordneten Vorlagengruppe zu einer vorhandenen Untergruppe (zum Beispiel beim Erstellen von `Linux servers`, wenn `Linux servers/Databases` bereits vorhanden ist) werden keine Berechtigungen für das übergeordnete Element gesetzt.

Beim Bearbeiten einer beliebigen Vorlagengruppe können Sie außerdem die zusätzliche Option *Berechtigungen auf alle Untergruppen anwenden* festlegen.



Wenn Sie dieses Kontrollkästchen aktivieren und auf *Aktualisieren* klicken, wird für alle aktuellen und zukünftigen Untergruppen der bearbeiteten Vorlagengruppe dieselbe Berechtigungsstufe angewendet.

Wenn also einer beliebigen Benutzergruppe unterschiedliche Berechtigungen für die Untergruppen der bearbeiteten Vorlagengruppe zugewiesen wurden, erhalten durch Aktivieren des Kontrollkästchens alle aktuellen und zukünftigen Untergruppen dieselben Benutzerberechtigungen wie die bearbeitete Gruppe.

Beachten Sie, dass diese Option nicht in der Datenbank gespeichert wird und vorhandene Berechtigungen überschreibt. Alle über diese Option vorgenommenen Änderungen können nur manuell rückgängig gemacht werden.

Verknüpfen/Entknüpfen

Übersicht

Verknüpfen ist ein Prozess, bei dem Vorlagen auf Hosts angewendet werden, während beim Entknüpfen die Verknüpfung einer Vorlage mit einem Host entfernt wird.

Verknüpfen einer Vorlage

Gehen Sie wie folgt vor, um eine Vorlage mit dem Host zu verknüpfen:

1. Gehen Sie zu *Datensammlung* > *Hosts*.
2. Klicken Sie auf den gewünschten Host.
3. Beginnen Sie im Feld *Vorlagen*, den Namen der Vorlage einzugeben. Eine Liste mit passenden Vorlagen wird angezeigt. Scrollen Sie nach unten, um eine Vorlage auszuwählen. Alternativ können Sie neben dem Feld auf *Auswählen* klicken und eine oder mehrere Vorlagen aus der Liste in einem Popup-Fenster auswählen.
4. Klicken Sie im Formular mit den Host-Attributen auf *Hinzufügen/Aktualisieren*.

Der Host verfügt nun über alle Entitäten der Vorlage. Dazu gehören Datenpunkte, Auslöser, Graphen, Regeln für die Low-Level-Discovery, Web-Szenarien sowie Dashboards.

Attention:
Das Verknüpfen mehrerer Vorlagen mit demselben Host schlägt fehl, wenn diese Vorlagen Datenpunkte mit demselben Datenpunktschlüssel enthalten. Da Auslöser und Graphen Datenpunkte verwenden, können sie ebenfalls nicht aus mehreren Vorlagen mit einem einzelnen Host verknüpft werden, wenn identische Datenpunktschlüssel verwendet werden.

Wenn Entitäten (Datenpunkte, Auslöser usw.) aus der Vorlage hinzugefügt werden:

- zuvor auf dem Host vorhandene identische Entitäten werden als Entitäten der Vorlage aktualisiert, und **alle vorhandenen Anpassungen auf Host-Ebene an der Entität gehen verloren**;
- Entitäten aus der Vorlage werden hinzugefügt;
- alle direkt verknüpften Entitäten, die vor der Verknüpfung mit der Vorlage nur auf dem Host vorhanden waren, bleiben unverändert.

In den Listen werden alle Entitäten aus der Vorlage nun mit dem Namen der Vorlage als Präfix angezeigt. Dadurch wird angegeben, dass sie zu dieser bestimmten Vorlage gehören. Der Name der Vorlage selbst (grau angezeigt) ist ein Link, über den Sie auf die Liste dieser Entitäten auf Vorlagenebene zugreifen können.

Note:
Für einige Datenpunkte, z. B. externe Prüfungen, HTTP-Agent-Prüfungen, einfache Prüfungen, SSH-Prüfungen und Telnet-Prüfungen, ist eine Host-Schnittstelle optional. Wenn zum Zeitpunkt der Verknüpfung einer Vorlage keine Schnittstelle für den Host definiert ist, werden diese Datenpunkte ohne Schnittstelle hinzugefügt. Wenn Sie später eine Host-Schnittstelle hinzufügen, wird sie vorhandenen Datenpunkten nicht automatisch zugewiesen. Um die neu hinzugefügte Host-Schnittstelle allen Datenpunkten der Vorlage gleichzeitig zuzuweisen, heben Sie die Verknüpfung der Vorlage mit dem Host auf und verknüpfen Sie sie anschließend erneut. Um den Verlauf der Datenpunkte beizubehalten, verwenden Sie die Option *Verknüpfung aufheben*, nicht *Verknüpfung aufheben und löschen*.

Wenn einer Entität nicht der Name der Vorlage vorangestellt ist, bedeutet dies, dass sie bereits vor der Verknüpfung mit der Vorlage auf dem Host vorhanden war und nicht von der Vorlage hinzugefügt wurde.

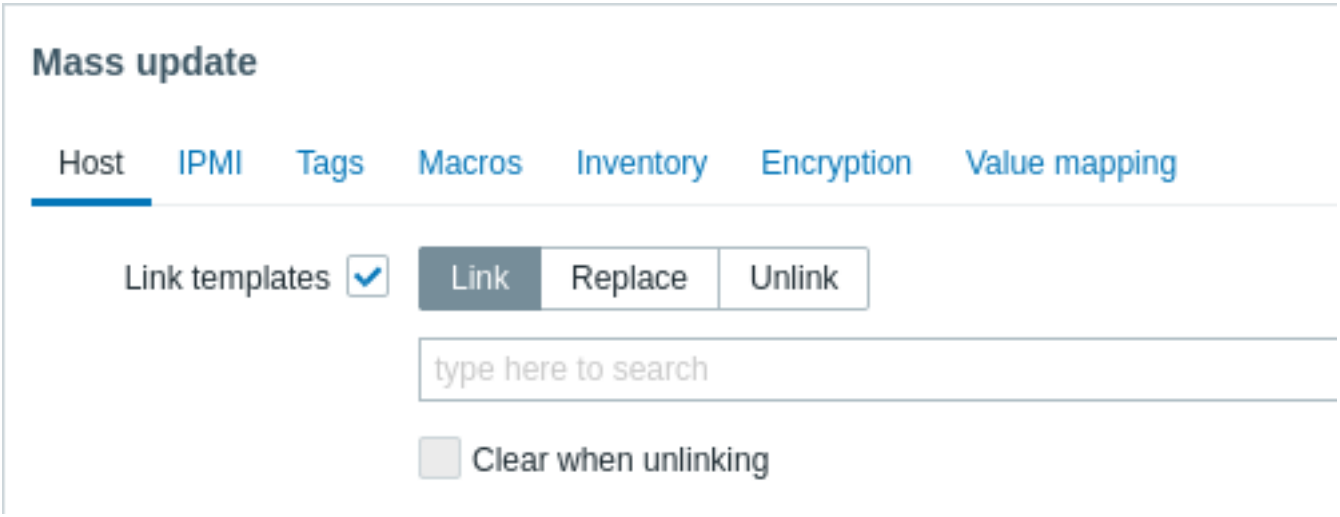
Eindeutigkeitskriterien für Entitäten

Beim Hinzufügen von Entitäten (Datenpunkten, Auslösern usw.) aus einer Vorlage ist es wichtig zu wissen, welche dieser Entitäten bereits auf dem Host vorhanden sind und aktualisiert werden müssen und welche Entitäten sich unterscheiden. Die Eindeutigkeitskriterien zur Entscheidung über Gleichheit/Unterschied sind:

- für Datenpunkte - der Datenpunktschlüssel;
- für Auslöser - Auslösername und Ausdruck;
- für benutzerdefinierte Diagramme - Diagrammname und seine Datenpunkte.

Verknüpfen von Vorlagen mit mehreren Hosts

Um die Vorlagenverknüpfung mehrerer Hosts zu aktualisieren, wählen Sie unter *Datensammlung* > *Hosts* einige Hosts aus, indem Sie deren Kontrollkästchen markieren. Klicken Sie anschließend unterhalb der Liste auf **Massenaktualisierung** und wählen Sie dann *Vorlagen verknüpfen*:



Um zusätzliche Vorlagen zu verknüpfen, beginnen Sie im Autovervollständigungsfeld mit der Eingabe des Vorlagennamens, bis eine Dropdown-Liste mit den passenden Vorlagen angezeigt wird. Scrollen Sie nach unten, um die zu verknüpfende Vorlage auszuwählen.

Mit der Option *Ersetzen* können Sie eine neue Vorlage verknüpfen und gleichzeitig alle Vorlagen trennen, die zuvor mit den Hosts verknüpft waren. Mit der Option *Verknüpfung aufheben* können Sie festlegen, welche Vorlagen getrennt werden sollen. Mit der Option *Beim Aufheben der Verknüpfung löschen* können Sie nicht nur zuvor verknüpfte Vorlagen trennen, sondern auch alle von ihnen geerbten Entitäten (Datenpunkte, Auslöser usw.) entfernen.

Note:
Zabbix bietet eine umfangreiche Auswahl vordefinierter Vorlagen. Sie können diese als Referenz verwenden. Vermeiden Sie es jedoch, sie unverändert in Produktionsumgebungen einzusetzen, da sie möglicherweise zu viele Datenpunkte enthalten und Daten zu häufig abfragen. Wenn Sie sie verwenden möchten, passen Sie sie an Ihre tatsächlichen Anforderungen an.

Verknüpfte Entitäten bearbeiten

Wenn Sie versuchen, einen Datenpunkt oder einen Auslöser zu bearbeiten, der aus der Vorlage verknüpft wurde, stellen Sie möglicherweise fest, dass viele wichtige Optionen für die Bearbeitung deaktiviert sind. Das ist sinnvoll, da die Idee von Vorlagen darin besteht, Dinge zentral auf Vorlagenebene mit einem einzigen Eingriff zu bearbeiten. Sie können jedoch beispielsweise einen Datenpunkt auf einzelnen Hosts aktivieren/deaktivieren und das Aktualisierungsintervall, die Verlaufsdauer sowie einige andere Parameter festlegen.

Wenn Sie die Entität vollständig bearbeiten möchten, müssen Sie sie auf Vorlagenebene bearbeiten (die Verknüpfung zur Vorlagenebene wird im Formularnamen angezeigt), wobei Sie beachten sollten, dass sich diese Änderungen auf alle Hosts auswirken, mit denen diese Vorlage verknüpft ist.

Attention:
Alle Anpassungen an den Entitäten, die auf Vorlagenebene vorgenommen werden, überschreiben die vorherigen Anpassungen der Entitäten auf Host-Ebene.

Trennen einer Vorlage

Um eine Vorlage von einem Host zu trennen, gehen Sie wie folgt vor:

1. Gehen Sie zu *Datenerfassung* > *Hosts*.
2. Klicken Sie auf den gewünschten Host und suchen Sie das Feld *Vorlagen*.
3. Klicken Sie neben der Vorlage auf *Trennen* oder *Trennen und löschen*, um die Vorlage zu trennen.
4. Klicken Sie im Formular mit den Host-Attributen auf *Aktualisieren*.

Wenn Sie die Option *Trennen* wählen, wird lediglich die Verknüpfung mit der Vorlage entfernt, während alle ihre Entitäten beim Host verbleiben. Dazu gehören Datenpunkte, Auslöser, Diagramme, Low-Level-Discovery-Regeln und Web-Szenarien, jedoch keine Dashboards. Beachten Sie, dass auch von verknüpften Vorlagen geerbte Wertzuordnungen und Tags entfernt werden.

Wenn Sie die Option *Trennen und löschen* wählen, werden sowohl die Verknüpfung mit der Vorlage als auch alle ihre Entitäten (Datenpunkte, Auslöser usw.) entfernt.

Verschachtelung

Übersicht

Verschachtelung ist eine Möglichkeit, bei der eine Vorlage eine oder mehrere andere Vorlagen umfasst.

Da es sinnvoll ist, Entitäten für verschiedene Dienste, Anwendungen usw. in einzelnen Vorlagen zu trennen, kann es sein, dass Sie am Ende recht viele Vorlagen haben, die alle mit zahlreichen Hosts verknüpft werden müssen. Um dies zu vereinfachen, können mehrere Vorlagen in einer einzigen Vorlage miteinander verknüpft werden.

Der Vorteil der Verschachtelung besteht darin, dass Sie nur eine Vorlage mit dem Host verknüpfen müssen, und der Host automatisch alle Entitäten aus den Vorlagen erbt, die mit dieser einen Vorlage verknüpft sind. Wenn wir zum Beispiel *T1* und *T2* mit *T3* verknüpfen, ergänzen wir *T3* um alle Entitäten aus *T1* und *T2*, aber nicht umgekehrt. Wenn wir *T1* mit *T2* und *T3* verknüpfen, ergänzen wir *T2* und *T3* um Entitäten aus *T1*.

Verschachtelte Vorlagen konfigurieren

Um Vorlagen zu verknüpfen, müssen Sie eine vorhandene Vorlage verwenden (oder eine neue erstellen) und dann:

1. Öffnen Sie das Konfigurationsformular der Vorlage.
2. Suchen Sie das Feld *Vorlagen*.
3. Klicken Sie auf *Auswählen*, um das Popup-Fenster *Vorlagen* zu öffnen.
4. Wählen Sie im Popup-Fenster die erforderlichen Vorlagen aus und klicken Sie dann auf *Auswählen*, um die Vorlagen zur Liste hinzuzufügen.
5. Klicken Sie im Konfigurationsformular der Vorlage auf *Hinzufügen* oder *Aktualisieren*.

Damit werden nun alle Entitäten der konfigurierten Vorlage sowie alle Entitäten der verknüpften Vorlagen in der Vorlagenkonfiguration angezeigt. Dazu gehören Datenpunkte, Auslöser, Diagramme, Regeln zur Low-Level-Erkennung und Webszenarien, jedoch keine Dashboards. Die Dashboards verknüpfter Vorlagen werden jedoch dennoch von Hosts übernommen.

Um eine der verknüpften Vorlagen zu trennen, klicken Sie im Konfigurationsformular der Vorlage auf *Verknüpfen aufheben* oder *Verknüpfen aufheben und löschen* und anschließend auf *Aktualisieren*.

Die Option *Verknüpfen aufheben* entfernt lediglich die Zuordnung zur verknüpften Vorlage, ohne deren Entitäten (Datenpunkte, Auslöser usw.) zu entfernen.

Die Option *Verknüpfen aufheben und löschen* entfernt sowohl die Zuordnung zur verknüpften Vorlage als auch alle deren Entitäten (Datenpunkte, Auslöser usw.).

Massenaktualisierung

Übersicht

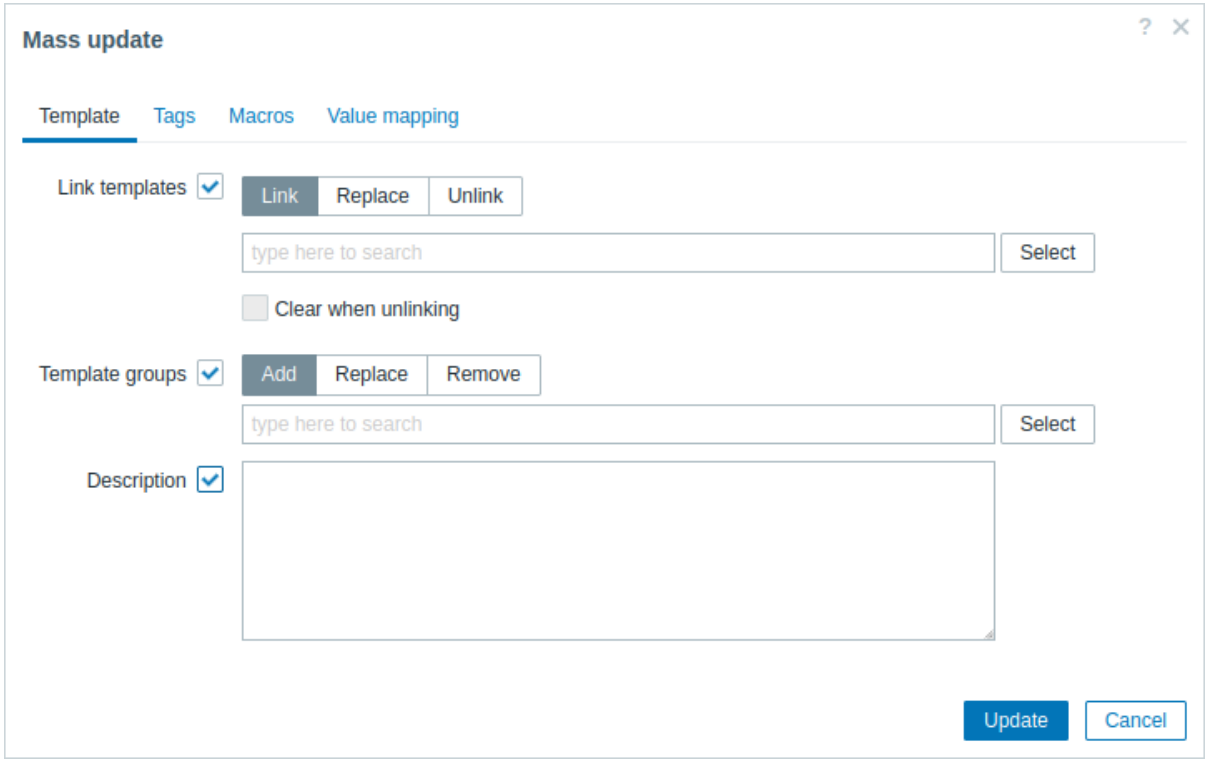
Manchmal möchten Sie ein bestimmtes Attribut für mehrere Vorlagen gleichzeitig ändern. Anstatt jede einzelne Vorlage zur Bearbeitung zu öffnen, können Sie dafür die Funktion zur Massenaktualisierung verwenden.

Massenaktualisierung verwenden

Um einige Vorlagen massenhaft zu aktualisieren, gehen Sie wie folgt vor:

1. Markieren Sie die Kontrollkästchen vor den Vorlagen, die Sie aktualisieren möchten, in der Vorlagenliste.
2. Klicken Sie unterhalb der Liste auf *Massenaktualisierung*.
3. Wechseln Sie zur Registerkarte mit den erforderlichen Attributen (*Vorlage*, *Tags*, *Makros* oder *Wertzuordnung*).
4. Markieren Sie die Kontrollkästchen der zu aktualisierenden Attribute und geben Sie für diese einen neuen Wert ein.

Die Registerkarte **Vorlage** enthält allgemeine Optionen für die Massenaktualisierung von Vorlagen.



Bei Auswahl der jeweiligen Schaltfläche für die Aktualisierung *Vorlagen verknüpfen* stehen folgende Optionen zur Verfügung:

- *Verknüpfen* – geben Sie an, welche zusätzlichen Vorlagen verknüpft werden sollen;
- *Ersetzen* – geben Sie an, welche Vorlagen verknüpft werden sollen, während gleichzeitig alle zuvor verknüpften Vorlagen getrennt werden;
- *Trennen* – geben Sie an, welche Vorlagen getrennt werden sollen.

Um die zu verknüpfenden bzw. zu trennenden Vorlagen anzugeben, beginnen Sie im Autovervollständigungsfeld mit der Eingabe des Vorlagennamens, bis eine Dropdown-Liste mit den passenden Vorlagen angezeigt wird. Scrollen Sie nach unten, um die gewünschten Vorlagen auszuwählen.

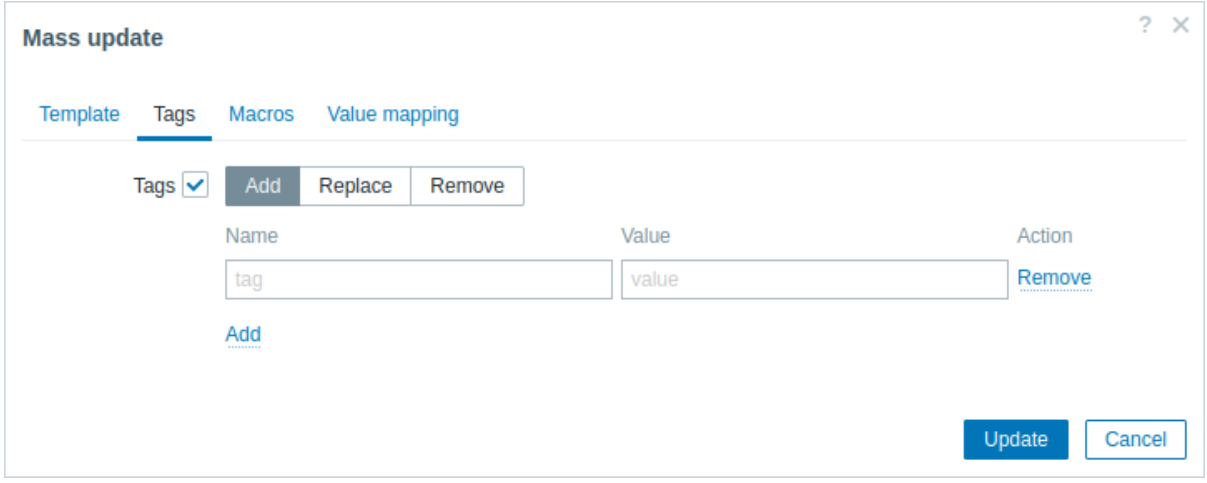
Mit der Option *Beim Trennen löschen* können Sie nicht nur zuvor verknüpfte Vorlagen trennen, sondern auch alle von ihnen geerbten Entitäten (Datenpunkte, Auslöser, Diagramme usw.) entfernen.

Bei Auswahl der jeweiligen Schaltfläche für die Aktualisierung *Vorlagengruppen* stehen folgende Optionen zur Verfügung:

- *Hinzufügen* – geben Sie zusätzliche Vorlagengruppen aus den vorhandenen Gruppen an oder geben Sie vollständig neue Vorlagengruppen für die Vorlagen ein;
- *Ersetzen* – entfernen Sie die Vorlage aus allen vorhandenen Vorlagengruppen und ersetzen Sie diese durch die in diesem Feld angegebenen Vorlagengruppen (vorhandene oder neue Vorlagengruppen);
- *Entfernen* – entfernen Sie bestimmte Vorlagengruppen aus den Vorlagen.

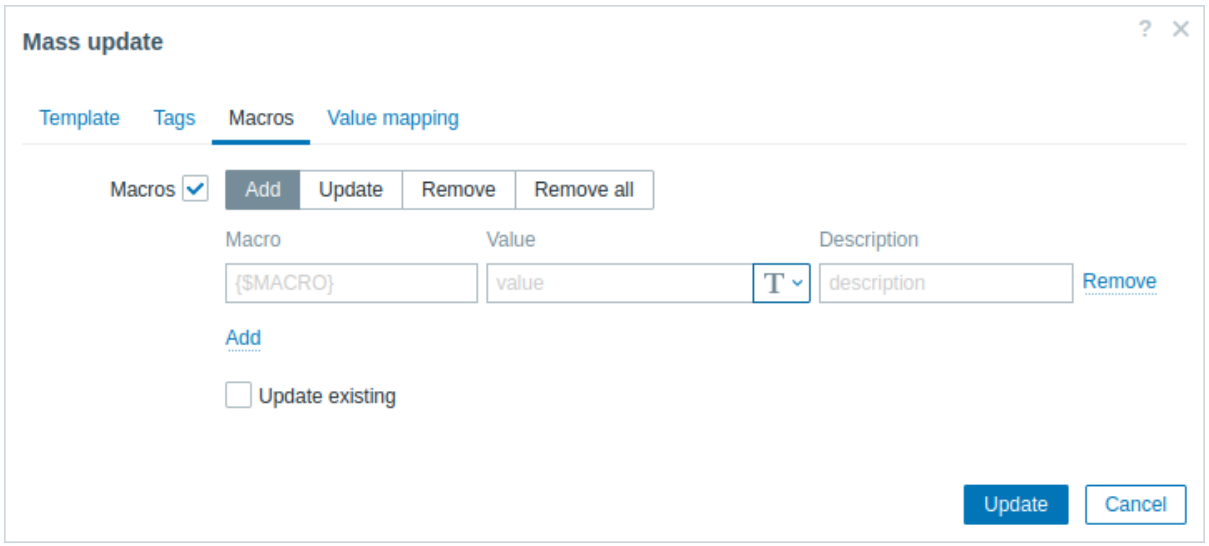
Diese Felder unterstützen die Autovervollständigung. Wenn Sie mit der Eingabe beginnen, wird eine Dropdown-Liste mit passenden Vorlagengruppen angezeigt. Wenn die Vorlagengruppe neu ist, wird sie ebenfalls in der Dropdown-Liste angezeigt und nach der Zeichenfolge mit *(neu)* gekennzeichnet. Scrollen Sie nach unten, um eine Auswahl zu treffen.

Die Registerkarte **Tags** ermöglicht die Massenaktualisierung von Tags auf Vorlagenebene.



Benutzermakros, {INVENTORY.*}-Makros, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} und {HOST.ID}-Makros werden in Tags unterstützt. Beachten Sie, dass Tags mit demselben Namen, aber unterschiedlichen Werten nicht als „Duplikate“ betrachtet werden und derselben Vorlage hinzugefügt werden können.

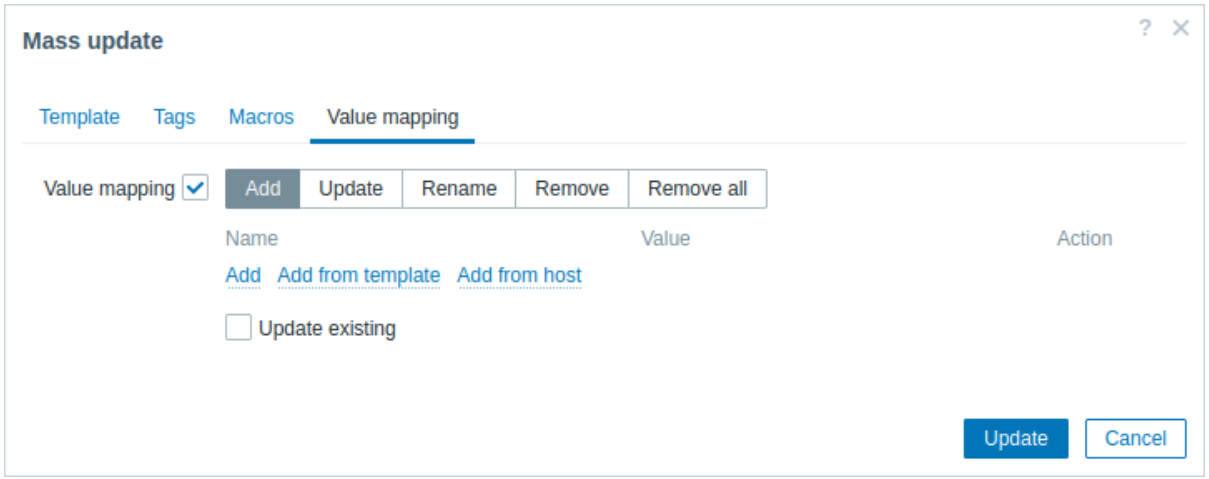
Der Tab **Makros** ermöglicht die Massenaktualisierung von Makros auf Vorlagenebene.



Bei der Auswahl der jeweiligen Schaltfläche zur Aktualisierung von Makros stehen folgende Optionen zur Verfügung:

- *Hinzufügen* – zusätzliche Benutzermakros für die Vorlagen angeben. Wenn das Kontrollkästchen *Vorhandene aktualisieren* aktiviert ist, werden Wert, Typ und Beschreibung für den angegebenen Makronamen aktualisiert. Wenn es nicht aktiviert ist und ein Makro mit diesem Namen bereits auf der Vorlage bzw. den Vorlagen vorhanden ist, wird es nicht aktualisiert.
- *Aktualisieren* – Werte, Typen und Beschreibungen der in dieser Liste angegebenen Makros ersetzen. Wenn das Kontrollkästchen *Fehlende hinzufügen* aktiviert ist, wird ein Makro, das zuvor auf einer Vorlage nicht vorhanden war, als neues Makro hinzugefügt. Wenn es nicht aktiviert ist, werden nur Makros aktualisiert, die bereits auf einer Vorlage vorhanden sind.
- *Entfernen* – angegebene Makros aus Vorlagen entfernen. Wenn das Kontrollkästchen *Ausgewählte ausnehmen* aktiviert ist, werden alle Makros außer den in der Liste angegebenen entfernt. Wenn es nicht aktiviert ist, werden nur die in der Liste angegebenen Makros entfernt.
- *Alle entfernen* – alle Benutzermakros aus Vorlagen entfernen. Wenn das Kontrollkästchen *Ich bestätige, alle Makros zu entfernen* nicht aktiviert ist, wird ein neues Popup-Fenster geöffnet, in dem Sie aufgefordert werden, das Entfernen aller Makros zu bestätigen.

Die Registerkarte **Wertzuordnung** ermöglicht die Massenaktualisierung von Wertzuordnungen.



Für die Aktualisierung von Wertzuordnungen stehen Schaltflächen mit den folgenden Optionen zur Verfügung:

- *Hinzufügen* – Wertzuordnungen zu den Vorlagen hinzufügen. Wenn Sie *Vorhandene aktualisieren* markieren, werden alle Eigenschaften der Wertzuordnung mit diesem Namen aktualisiert. Andernfalls wird eine Wertzuordnung mit diesem Namen nicht aktualisiert, wenn sie bereits vorhanden ist.
- *Aktualisieren* – vorhandene Wertzuordnungen aktualisieren. Wenn Sie *Fehlende hinzufügen* markieren, wird eine Wertzuordnung, die zuvor nicht in einer Vorlage vorhanden war, als neue Wertzuordnung hinzugefügt. Andernfalls werden nur die Wertzuordnungen aktualisiert, die bereits in einer Vorlage vorhanden sind.
- *Umbenennen* – einer vorhandenen Wertzuordnung einen neuen Namen geben.

- *Entfernen* – die angegebenen Wertzuordnungen aus den Vorlagen entfernen. Wenn Sie *Ausgewählte ausnehmen* markieren, werden alle Wertzuordnungen entfernt, **außer** den angegebenen.
- *Alle entfernen* – alle Wertzuordnungen aus den Vorlagen entfernen. Wenn das Kontrollkästchen *Ich bestätige, alle Wertzuordnungen zu entfernen* nicht markiert ist, wird ein neues Popup-Fenster geöffnet, in dem Sie aufgefordert werden, das Entfernen zu bestätigen.

Die Optionen *Aus Vorlage hinzufügen* und *Aus Host hinzufügen* sind für Vorgänge zum Hinzufügen bzw. Aktualisieren von Wertzuordnungen verfügbar.
Damit können Sie Wertzuordnungen aus einer Vorlage bzw. einem Host auswählen.

Wenn alle erforderlichen Änderungen vorgenommen wurden, klicken Sie auf *Update*. Die Attribute werden entsprechend für alle ausgewählten Vorlagen aktualisiert.

Hosts und Hostgruppen

Was ist ein „Host“?

In Zabbix bezieht sich ein „Host“ auf jedes physische oder virtuelle Gerät, jede Anwendung, jeden Dienst oder jede andere logisch zusammengehörige Sammlung überwachter Parameter.

Das Erstellen von Hosts ist eine der ersten Überwachungsaufgaben in Zabbix. Wenn Sie beispielsweise einige Parameter auf einem Server „x“ überwachen möchten, müssen Sie zuerst einen Host mit dem Namen etwa „Server X“ erstellen; anschließend können Sie ihm Überwachungs-Datenpunkte hinzufügen.

Hosts sind in Host-Gruppen organisiert.

Weiter zu:

- Host-Assistent für eine geführte Schritt-für-Schritt-Oberfläche zum Erstellen und Konfigurieren eines neuen oder bestehenden Hosts.
- Konfigurieren eines Hosts für einen klassischen Ansatz zum Erstellen und Konfigurieren eines Hosts.

Host-Assistent

Übersicht

Der Host Wizard ist eine geführte, schrittweise Oberfläche zum Einrichten Ihres Überwachungsziels (Gerät, Anwendung, Dienst usw.) in Zabbix. Er führt Sie durch die folgenden Schritte:

- Auswahl einer Vorlage
- Erstellen oder Auswählen eines Host
- Installieren von Zabbix Agent oder Agent 2
- Hinzufügen einer Host-Schnittstelle
- Anwenden zusätzlicher Konfiguration auf Ihr Überwachungsziel oder den Zabbix-Host (falls von der Vorlage erforderlich)

Um auf den Host Wizard im Zabbix Frontend zuzugreifen, gehen Sie wie folgt vor:

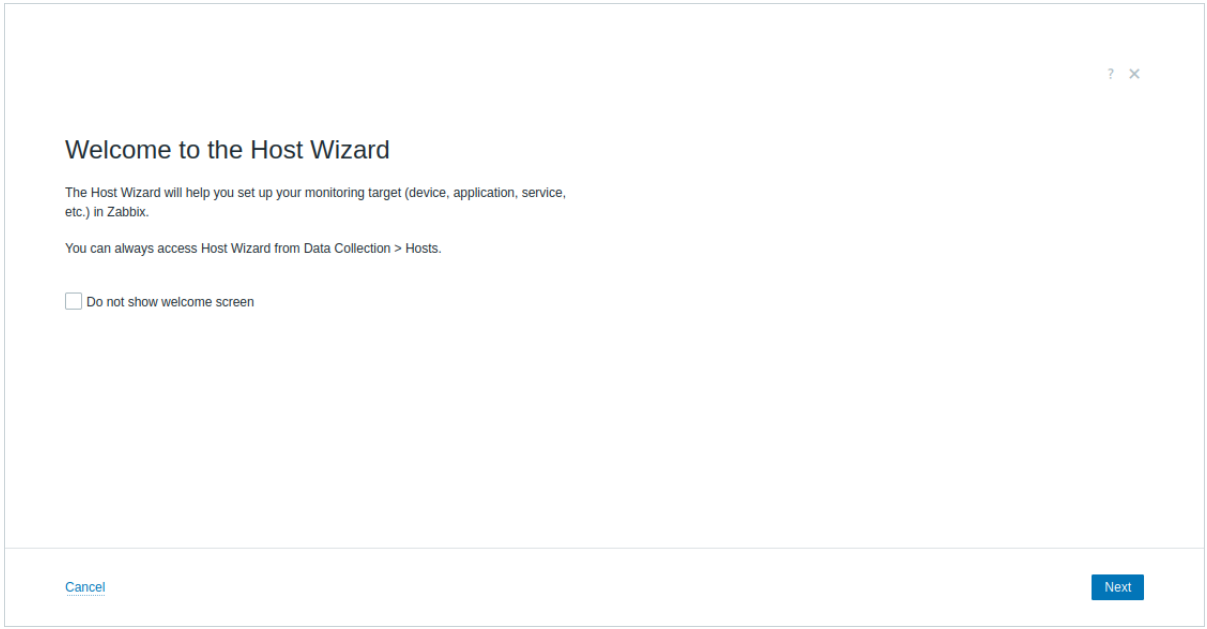
1. Gehen Sie zu: *Datenerfassung* > *Hosts*.
2. Klicken Sie oben rechts auf dem Bildschirm auf *Host Wizard*.
3. Folgen Sie den Anweisungen im Assistenten.

Sie können einen vorhandenen Host auch bearbeiten, indem Sie in verschiedenen Frontend-Bereichen auf seinen Namen (oder das Drei-Punkte-Symbol daneben) klicken, um über das Host-Menü auf den Host Wizard zuzugreifen. Beachten Sie, dass beim Bearbeiten eines vorhandenen Host einige Schritte übersprungen werden, wenn er bereits die von der ausgewählten Vorlage erforderliche Konfiguration enthält.

Alternativ können Sie die klassische Vorgehensweise zum Erstellen und Konfigurieren eines Host verwenden.

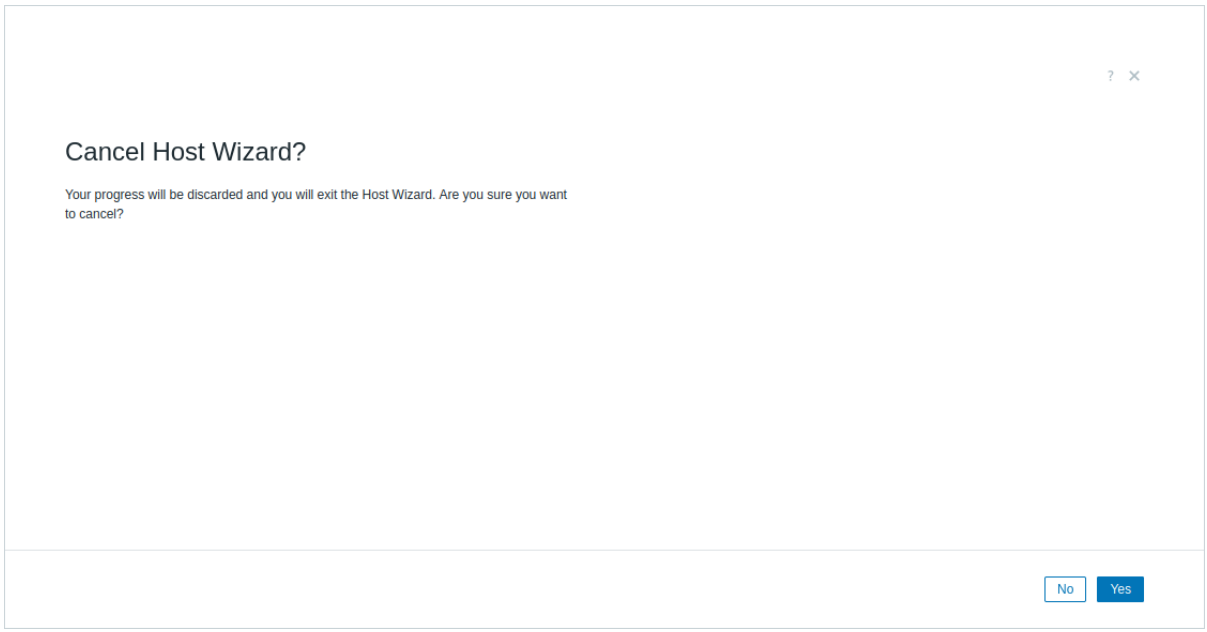
Willkommen im Host-Assistenten

Wenn Sie den Host-Assistenten starten, wird ein Willkommensbildschirm angezeigt:



Um diesen Bildschirm in zukünftigen Sitzungen zu überspringen, können Sie das Kontrollkästchen *Willkommensbildschirm nicht anzeigen* aktivieren und auf *Weiter* klicken.

Wenn Sie in den nächsten Schritten des Host-Assistenten mit der Konfiguration von Einstellungen beginnen und versuchen, den Assistenten durch Klicken auf das Schließen-Symbol oder durch Drücken von Esc zu schließen, wird ein Bestätigungsbildschirm angezeigt:



Wenn Sie auf *Ja* klicken, wird der Host-Assistent beendet, ohne dass Ihre Fortschritte gespeichert werden. Wenn Sie auf *Nein*, Esc oder das Schließen-Symbol klicken, kehren Sie zum letzten Schritt zurück.

Vorlage auswählen

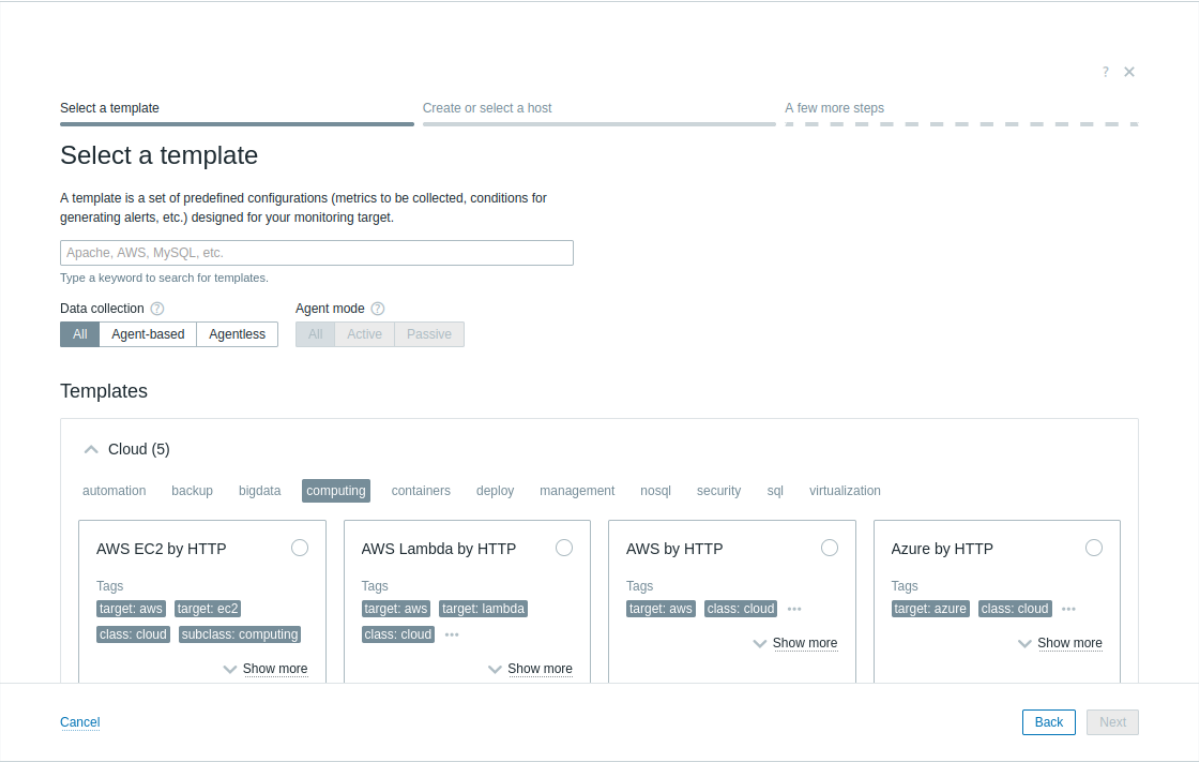
Der erste Schritt beim Einrichten Ihres Überwachungsziels besteht darin, eine Vorlage auszuwählen. Dabei handelt es sich um eine Sammlung vordefinierter Konfigurationen (zu erfassende Metriken, Bedingungen für die Generierung von Warnungen usw.), die für Ihr Überwachungsziel entwickelt wurden.

Die Out-of-the-Box-Vorlagen von Zabbix bieten eine Vielzahl vordefinierter Überwachungskonfigurationen für Betriebssysteme, Anwendungen, Datenbanken, Netzwerkgeräte, Dienste und mehr.

In diesem Schritt können Sie:

- Vorlagen nach Klasse (Cloud, Datenbank, Netzwerk usw.) und Unterklasse (z. B. Automatisierung, Backup) durchsuchen, die auf Tags der Vorlagen basieren.
- Vorlagen anhand von Schlüsselwörtern im Vorlagennamen oder in den Namen und Werten der Vorlagen-Tags suchen.
- Vorlagen nach Datenerfassungsmethode filtern (Agent-basiert oder Agent-los; Agent-basierte Vorlagen enthalten mindestens einen Zabbix-Agent-Datenpunkt).

- Vorlagen nach Agent-Modus filtern (aktiv oder passiv; aktive Vorlagen enthalten mindestens einen Zabbix-Agent-Datenpunkt (aktiv), während passive Vorlagen mindestens einen passiven Datenpunkt enthalten; Einzelheiten finden Sie unter Passive und aktive Agent-Prüfungen).
- Wenn Sie den Host Wizard zum Konfigurieren eines vorhandenen Hosts verwenden, können Sie Vorlagen auch danach filtern, ob sie bereits mit dem Host verknüpft sind.



Attention:
Bei der Auswahl einer Vorlage wird möglicherweise folgende Meldung angezeigt: *Einige Vorlagen (n) sind nicht mit dem Host Wizard kompatibel. Erfahren Sie, wie Sie sie aktualisieren können. Benutzerdefinierte Vorlagen werden nicht unterstützt.* Dies bedeutet, dass einige Out-of-the-Box-Vorlagen noch nicht mit dem Host Wizard kompatibel sind und aktualisiert werden müssen. Weitere Informationen finden Sie in den Anweisungen zum Aktualisieren von Vorlagen.

Die von Ihnen ausgewählte Vorlage bestimmt die nächsten Schritte im Host Wizard. Wenn Sie beispielsweise die Vorlage *MySQL by Zabbix agent* auswählen, die den Zabbix-Agent zur Datenerfassung verwendet, führt Sie der Assistent durch den Installationsprozess des Agents.

Host erstellen oder auswählen

Eine Vorlage muss mit einem Host verknüpft werden. Dabei handelt es sich um eine Entität in Zabbix, die Ihr Überwachungsziel darstellt. Nach der Verknüpfung erhält der Host alle Entitäten der Vorlage, z. B. Datenpunkte (zu erfassende Metriken) und Auslöser (Bedingungen zum Generieren von Alarmen), die für die Überwachung erforderlich sind.

Jeder Host muss außerdem mindestens einer Hostgruppe angehören. Diese wird verwendet, um Hosts zu organisieren und ihnen Benutzerberechtigungen zuzuweisen.

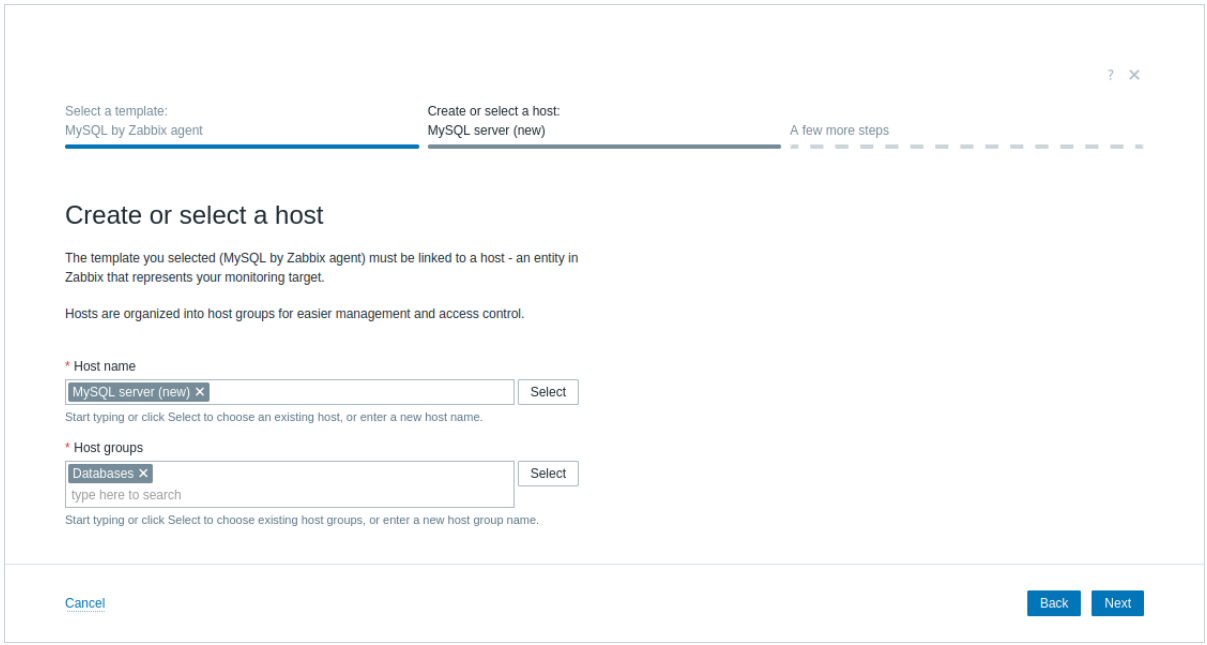
In diesem Schritt können Sie:

- Einen neuen Host und eine Hostgruppe erstellen, indem Sie für beide Namen eingeben.
- Einen neuen Host erstellen, indem Sie seinen Namen eingeben, und eine vorhandene Hostgruppe auswählen, der der neue Host zugewiesen wird.
- Einen vorhandenen Host auswählen. Sie können diesen Host außerdem zusätzlichen Hostgruppen zuweisen, ohne ihn aus den aktuellen Gruppen zu entfernen.

Note:
Durch die Auswahl eines vorhandenen Hosts wird dessen aktuelle Konfiguration nicht entfernt oder überschrieben, sofern dies in späteren Schritten nicht ausdrücklich angegeben wird. Ermittelte Hosts können nicht ausgewählt werden.

Sie könnten beispielsweise einen neuen Host mit dem Namen *MySQL server* erstellen, der den lokal installierten MySQL-Server darstellt, den Sie überwachen möchten. Die zuvor ausgewählte Vorlage *MySQL by Zabbix agent* wird mit diesem Host verknüpft. Zusätzlich könnten Sie die vorhandene Hostgruppe *Databases* auswählen (oder eine neue Gruppe erstellen, z. B. *MySQL servers*),

um den Host zusammen mit anderen datenbankbezogenen Hosts zu organisieren und die Verwaltung von Berechtigungen später zu vereinfachen.



Zabbix Agent installieren

Agent-basierte Vorlagen (wie *MySQL by Zabbix agent*) erfordern die Installation von Zabbix Agent oder Agent 2 – einem Prozess, der auf Ihrem Überwachungsziel bereitgestellt wird, um lokale Ressourcen und Anwendungen aktiv zu überwachen (falls dies von der Vorlage benötigt wird).

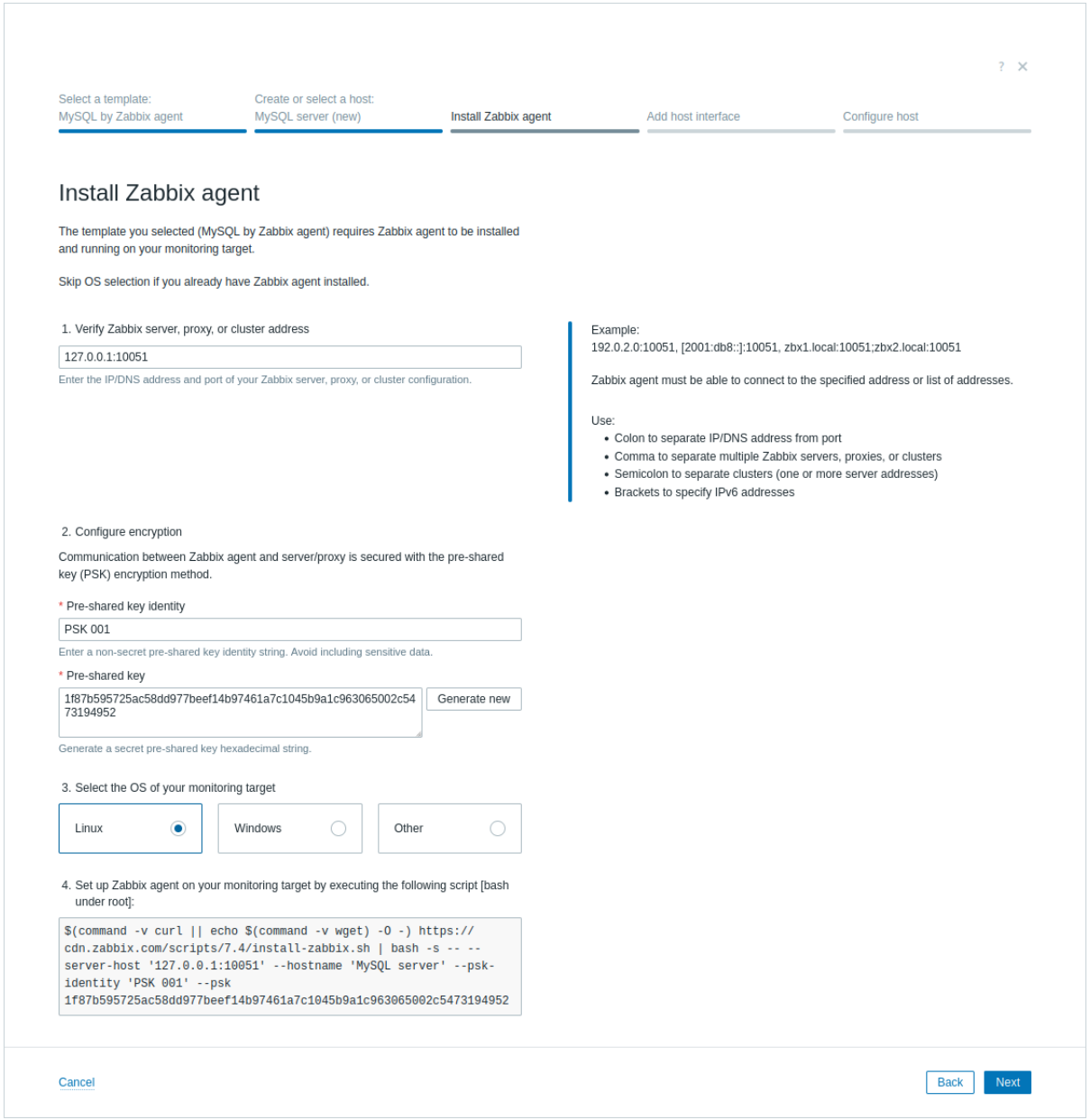
In diesem Schritt:

- Überprüfen Sie die Adresse des Zabbix Servers, Proxys oder Clusters (z. B. 192.0.2.1:10051). Dieser Wert wird verwendet, um die Parameter Server und ServerActive für den Zabbix Agent zu konfigurieren.
- Geben Sie eine neue, eindeutige und nicht geheime Identität für den Pre-Shared Key (z. B. *PSK 001* oder *mysql-agent-psk1*) ein und generieren Sie den Pre-Shared Key. Diese Werte werden verwendet, um die Verschlüsselung mittels Pre-Shared Key (PSK) für den Zabbix Host (z. B. *MySQL server*) und den Zabbix Agent zu konfigurieren.

Attention:
Beim Bearbeiten eines vorhandenen Hosts überschreibt die PSK-Konfiguration alle vorhandenen Verschlüsselungseinstellungen auf dem Host.

- Wählen Sie das Betriebssystem des Computers aus, auf dem sich Ihr Überwachungsziel befindet.
- Installieren Sie den Zabbix Agent auf diesem System, indem Sie das bereitgestellte Skript darauf ausführen oder die Installationsanweisungen befolgen.

Sie könnten beispielsweise *Pre-shared key identity* auf *PSK 001* setzen und einen neuen *Pre-shared key* generieren. Anschließend könnten Sie *Linux* als Betriebssystem auswählen und das bereitgestellte Skript auf diesem System ausführen. Kehren Sie nach der Installation des Zabbix Agent zum Host Wizard zurück.

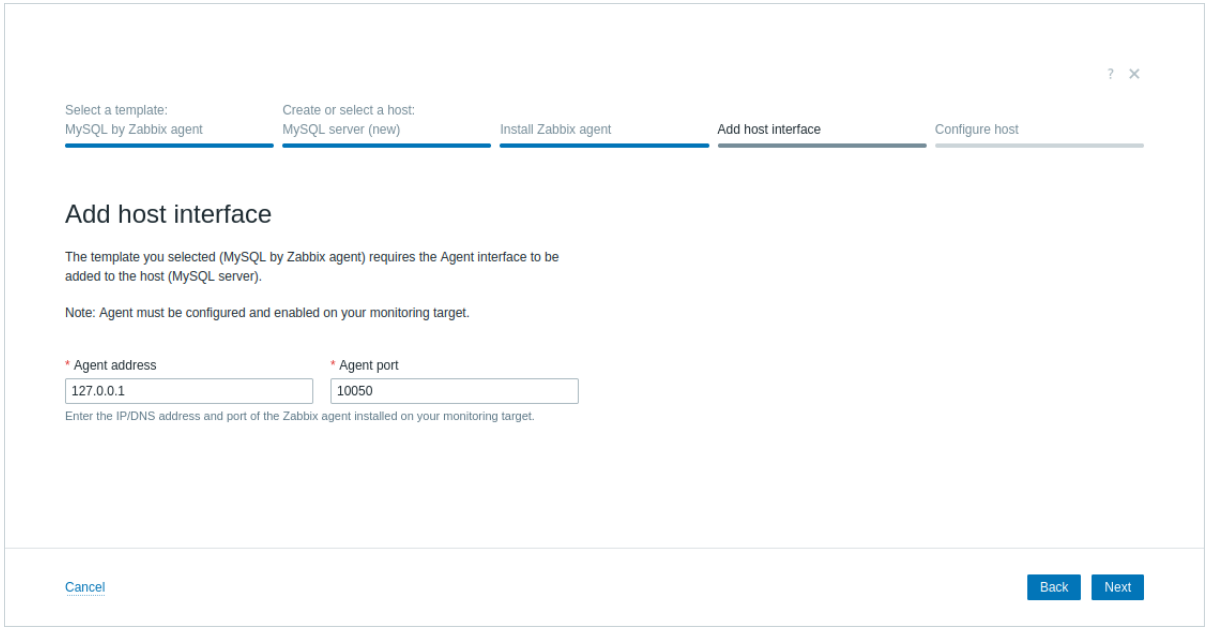


Host-Schnittstelle hinzufügen

Die Host-Schnittstelle definiert, wie der Zabbix Server eine Verbindung zu Ihrem Überwachungsziel über das Netzwerk herstellt. Sie definiert Verbindungsparameter wie IP-Adresse, DNS-Name, Port und Schnittstellentyp (Agent, SNMP, JMX oder IPMI), abhängig von der für die ausgewählte Vorlage erforderlichen Datenerfassungsmethode.

Geben Sie in diesem Schritt die von der ausgewählten Vorlage benötigten Details zur Host-Schnittstelle ein.

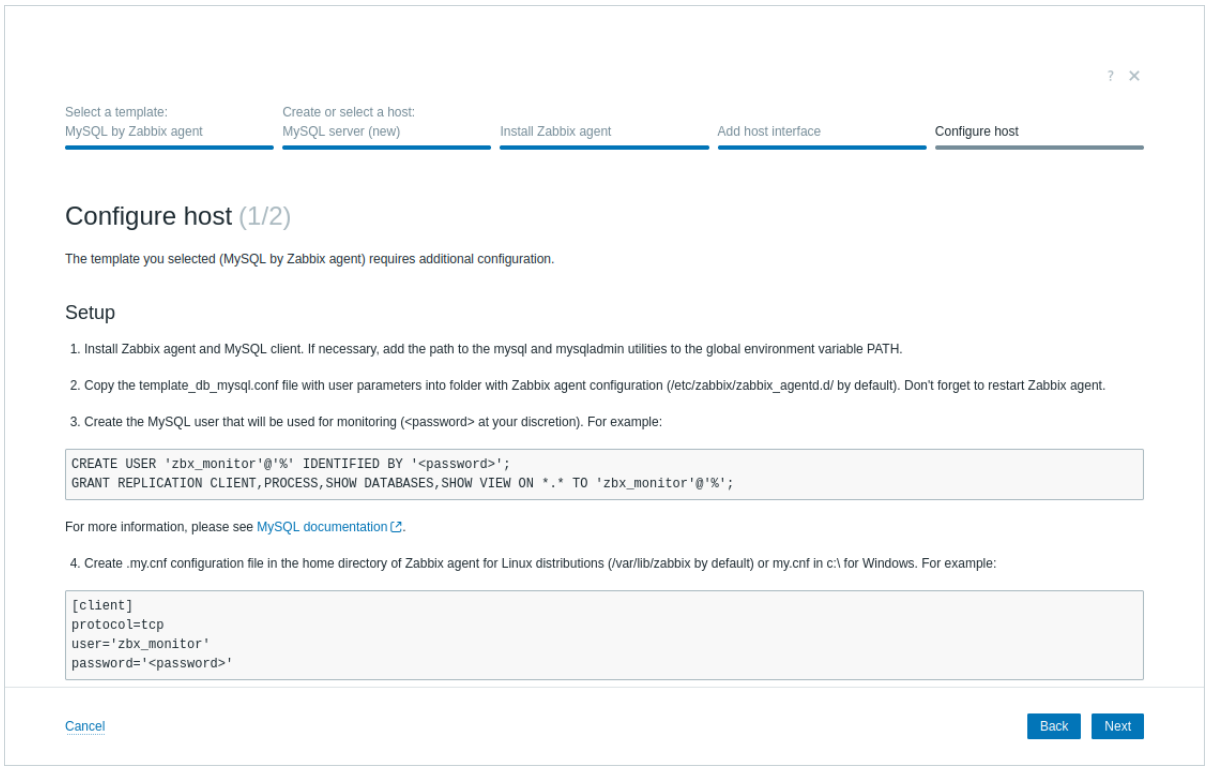
Sie können beispielsweise die standardmäßige Agent-Adresse (`127.0.0.1`) und den Port (`10050`) verwenden, wenn Zabbix Server, Zabbix Agent und der MySQL-Server alle lokal auf demselben Rechner ausgeführt werden.



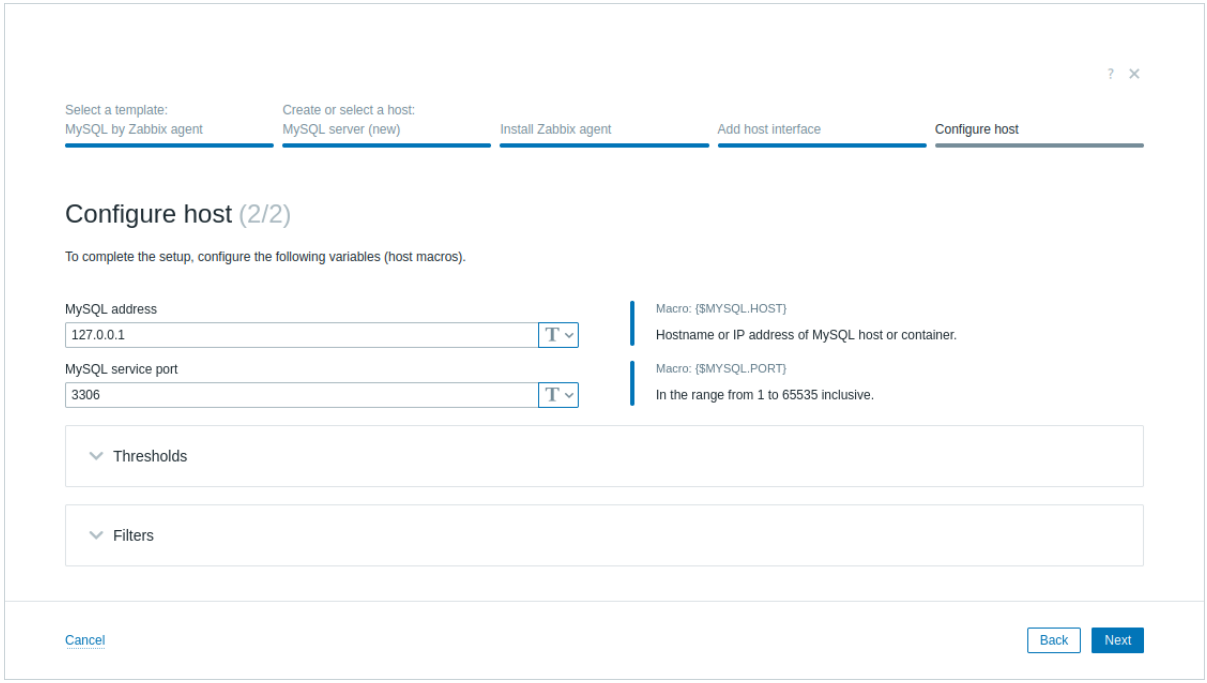
Host konfigurieren

Einige Vorlagen erfordern eine zusätzliche Konfiguration, bevor der Host erstellt werden kann. Dazu können gehören:

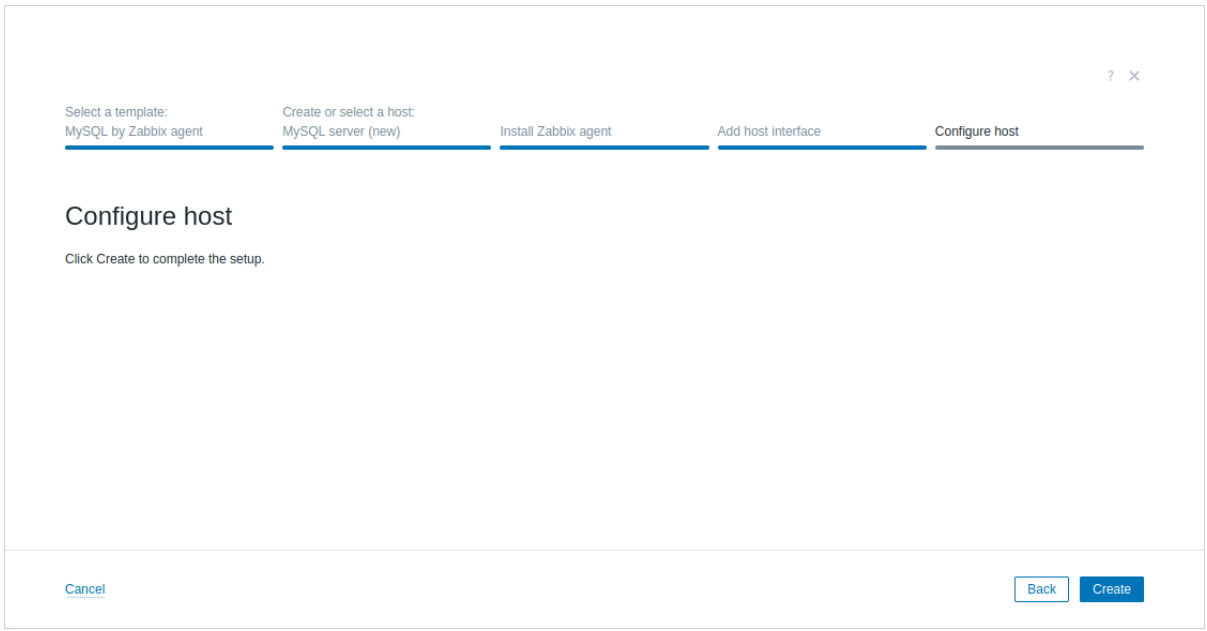
- Manuelle Konfiguration Ihres Überwachungsziels (z. B. das Aktivieren bestimmter Dienste, das Erstellen von Dienstbenutzern oder das Erteilen von Berechtigungen):



- Definieren von Host-Makros – Variablen, die das Verhalten von Datenpunkten, Verbindungseinstellungen, Authentifizierungsdaten und andere Parameter steuern:



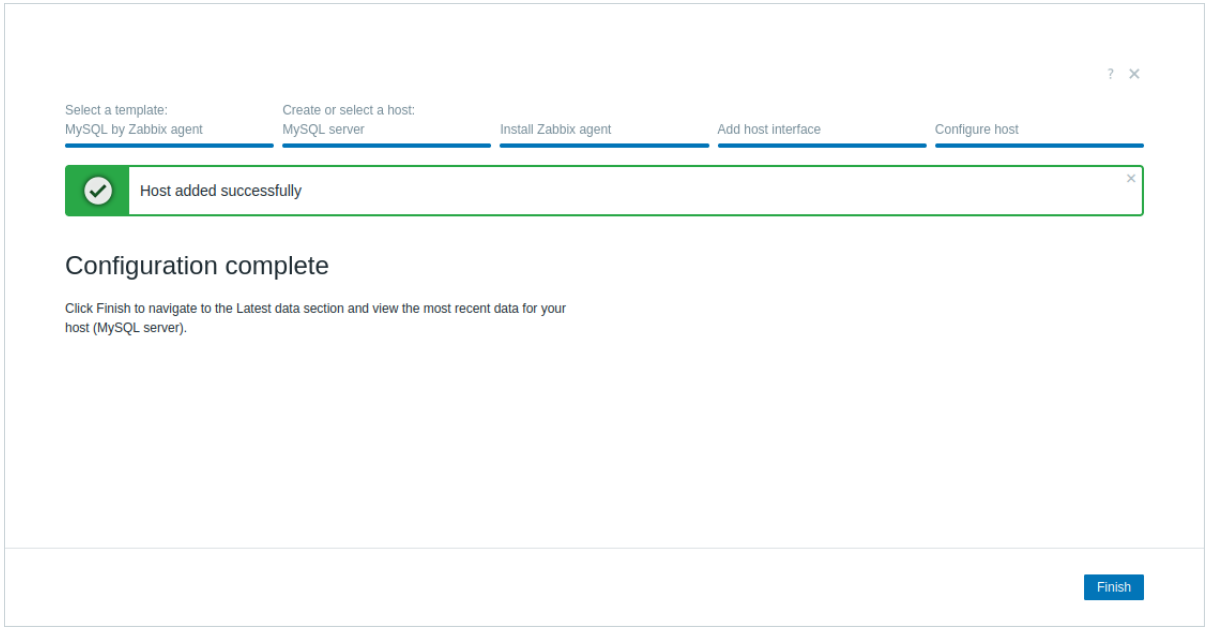
Nachdem Sie alle erforderlichen Konfigurationsschritte abgeschlossen haben, klicken Sie auf *Create*, um den Host zu Zabbix hinzuzufügen (oder auf *Update*, um die Konfiguration eines vorhandenen Hosts zu aktualisieren):



Konfiguration abgeschlossen

Zu diesem Zeitpunkt überwacht Zabbix Ihr Ziel bereits.

Klicken Sie bei neuen Hosts auf *Fertigstellen*, um zum Abschnitt Neueste Daten zu wechseln und die neuesten Daten für Ihren Host anzuzeigen. Klicken Sie bei bestehenden Hosts auf *Fertigstellen*, um den Host-Assistenten zu schließen und zu dem Bildschirm zurückzukehren, von dem aus er geöffnet wurde.



Konfigurieren eines Hosts

Übersicht

Um einen Host im Zabbix Frontend zu konfigurieren, gehen Sie wie folgt vor:

1. Gehen Sie zu *Datenerfassung* > *Hosts* oder *Überwachung* > *Hosts*.
2. Klicken Sie oben rechts auf dem Bildschirm auf *Host erstellen* (oder auf den Hostnamen, um einen vorhandenen Host zu bearbeiten).
3. Geben Sie die Parameter des Hosts im Formular ein.

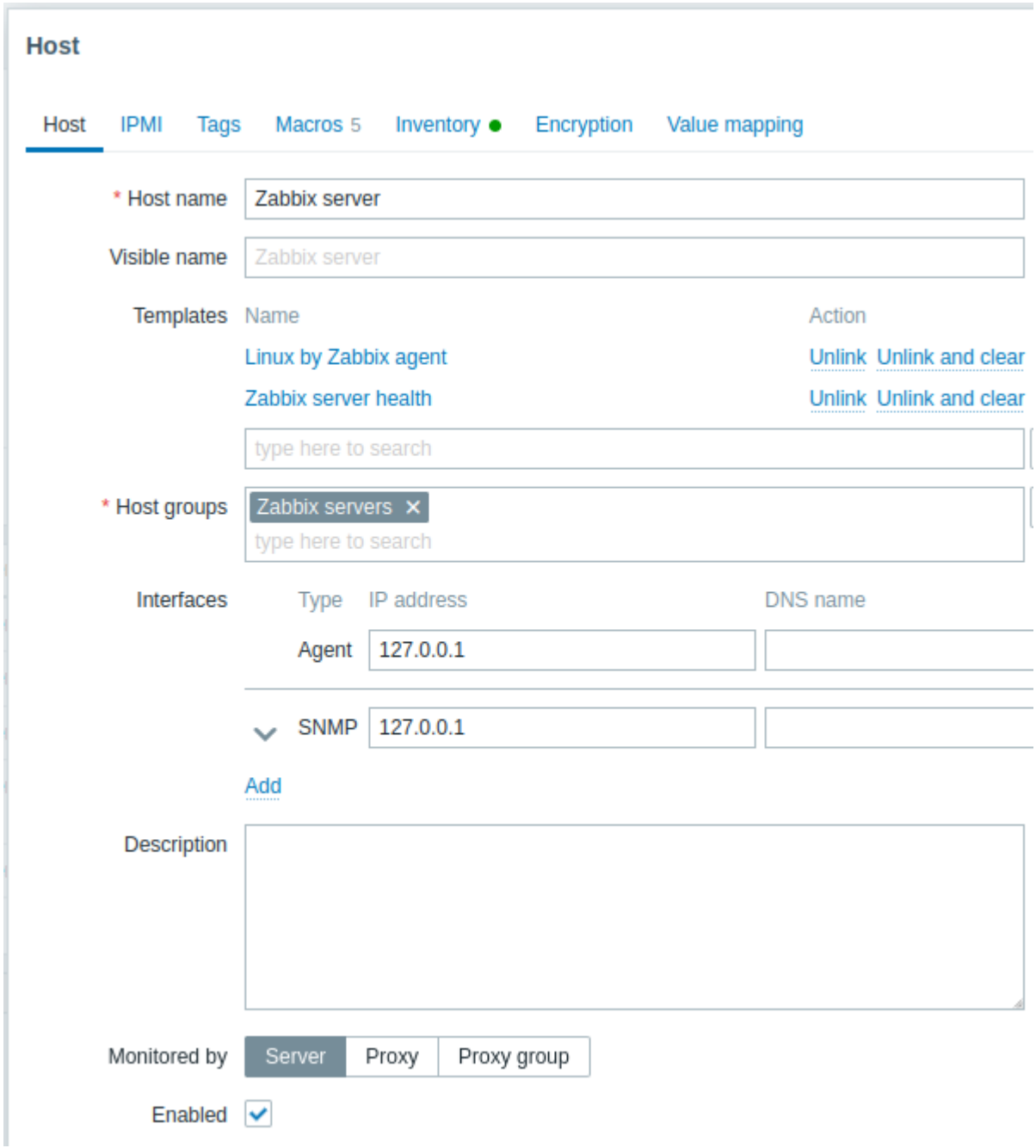
Sie können auch die Schaltfläche *Klonen* im Konfigurationsformular eines vorhandenen Hosts verwenden, um einen neuen Host zu erstellen. Dieser Host verfügt über alle Eigenschaften des vorhandenen Hosts, einschließlich verknüpfter Vorlagen, Entitäten (Datenpunkte, Auslöser usw.) aus diesen Vorlagen sowie der direkt dem vorhandenen Host zugeordneten Entitäten.

Beachten Sie, dass beim Klonen eines Hosts alle Vorlageneinträge so übernommen werden, wie sie ursprünglich in der Vorlage definiert sind. Alle Änderungen an diesen Entitäten, die auf der Ebene des vorhandenen Hosts vorgenommen wurden (z. B. geändertes Datenpunkt-Intervall, angepasster regulärer Ausdruck oder hinzugefügte Prototypen zur Low-Level-Discovery-Regel), werden nicht auf den neuen Host übernommen; stattdessen gelten die Werte der Vorlage.

Alternativ können Sie den Host-Assistenten verwenden, um einen Host über eine geführte, schrittweise Oberfläche zu konfigurieren.

Konfiguration

Die Registerkarte **Host** enthält allgemeine Hostattribute:



Alle Pflichtfelder sind mit einem roten Sternchen gekennzeichnet.

Parameter	Beschreibung
Host name	Geben Sie einen eindeutigen Hostnamen ein. Alphanumerische Zeichen, Leerzeichen, Punkte, Bindestriche und Unterstriche sind zulässig. Führende und nachgestellte Leerzeichen sind jedoch nicht zulässig. *Hinweis:* Wenn der Zabbix Agent auf dem Host ausgeführt wird, den Sie konfigurieren, muss der Parameter *Hostname* in der Agent-Konfigurationsdatei denselben Wert wie der hier eingegebene Hostname haben. Der Wert dieses Parameters wird für die Verarbeitung aktiver Prüfungen benötigt.
Visible name	Geben Sie einen eindeutigen sichtbaren Namen für den Host ein. Wenn Sie diesen Namen festlegen, wird er in Listen, Karten usw. anstelle des technischen Hostnamens angezeigt. Dieses Attribut unterstützt UTF-8.

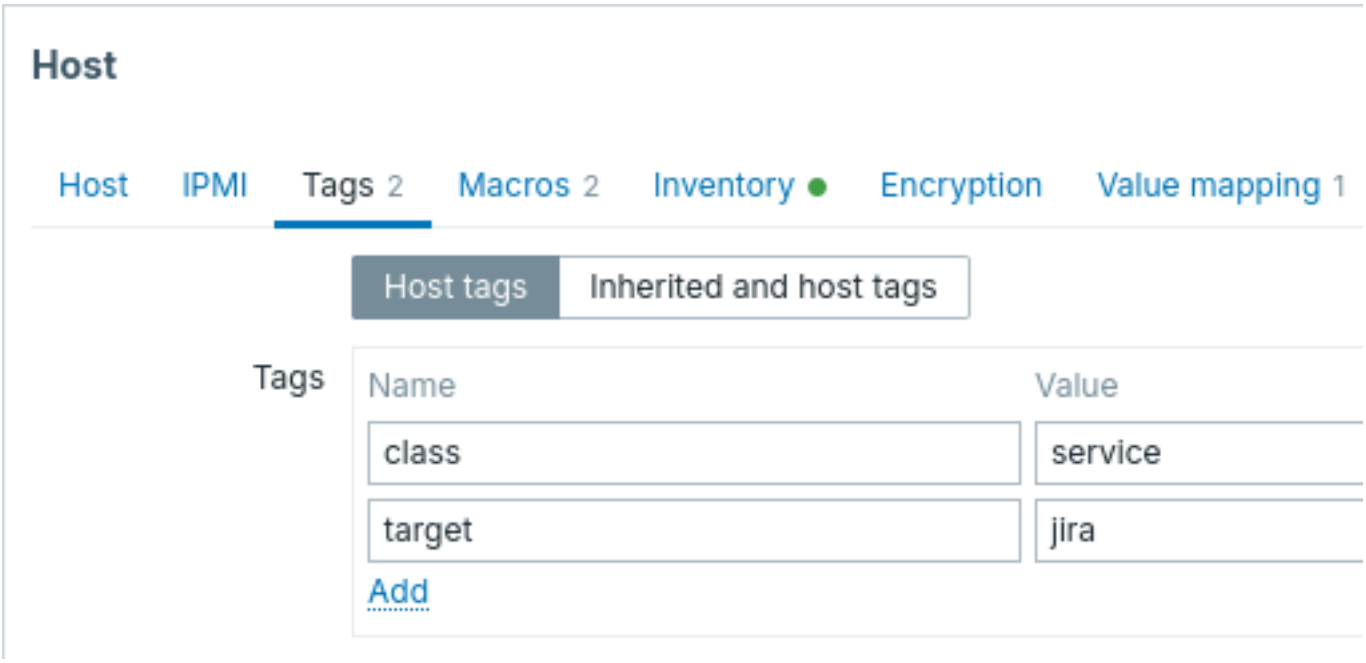
Parameter	Beschreibung
Templates	Verknüpfen Sie Vorlagen mit dem Host. Alle Entitäten (Datenpunkte, Auslöser usw.) werden von der Vorlage übernommen. Um eine neue Vorlage zu verknüpfen, beginnen Sie, den Vorlagennamen in das Texteingabefeld einzugeben. Eine Liste übereinstimmender Vorlagen wird angezeigt. Scrollen Sie nach unten, um eine Vorlage auszuwählen. Alternativ können Sie neben dem Feld auf *Select* klicken und Vorlagen aus der Liste in einem Popup-Fenster auswählen. Alle ausgewählten Vorlagen werden mit dem Host verknüpft, sobald das Host-Konfigurationsformular gespeichert oder aktualisiert wird. Um die Verknüpfung mit einer Vorlage aufzuheben, verwenden Sie eine der beiden Optionen im Block *Linked templates*: *Unlink* – hebt die Verknüpfung mit der Vorlage auf, behält jedoch deren Entitäten (Datenpunkte, Auslöser usw.) bei; *Unlink and clear* – hebt die Verknüpfung mit der Vorlage auf und entfernt alle deren Entitäten (Datenpunkte, Auslöser usw.). Die aufgelisteten Vorlagennamen sind anklickbare Links, die zum Vorlagenkonfigurationsformular führen.
Host groups	Wählen Sie die Hostgruppen aus, denen der Host angehört. Ein Host muss mindestens einer Hostgruppe angehören. Eine neue Gruppe kann erstellt und mit dem Host verknüpft werden, indem Sie einen noch nicht vorhandenen Gruppennamen hinzufügen.
Interfaces	Für einen Host werden mehrere Hostschnittstellentypen unterstützt: *Agent*, *SNMP*, *JMX* und *IPMI*. Standardmäßig sind keine Schnittstellen definiert. Um eine neue Schnittstelle hinzuzufügen, klicken Sie im Block *Interfaces* auf *Add*, wählen Sie den Schnittstellentyp aus und geben Sie die Informationen zu *IP/DNS*, *Connect to* und *Port* ein. *Hinweis:* Schnittstellen, die in Datenpunkten verwendet werden, können nicht entfernt werden; der Link *Remove* ist für diese Schnittstellen deaktiviert. Die „IP“ oder der „DNS“-Name einer SNMP-Schnittstelle wird auch für SNMP-Traps verwendet. Beim Abgleich wird nur die ausgewählte „IP“ oder der ausgewählte „DNS“-Name in der Hostschnittstelle verwendet. Weitere Informationen zum Konfigurieren einer SNMP-Schnittstelle (v1, v2 und v3) finden Sie unter SNMP-Überwachung konfigurieren.
IP address	IP-Adresse des Hosts (optional).
DNS name	DNS-Name des Hosts (optional).
Connect to	Durch Klicken auf die entsprechende Schaltfläche legen Sie fest, was der Zabbix Server zum Abrufen von Daten von Agents verwenden soll: **IP** – Verbindung zur IP-Adresse des Hosts herstellen (empfohlen) **DNS** – Verbindung zum DNS-Namen des Hosts herstellen
Port	TCP/UDP-Portnummer. Standardwerte sind: 10050 für den Zabbix Agent, 161 für den SNMP Agent, 12345 für JMX und 623 für IPMI.
Default	Aktivieren Sie das Optionsfeld, um die Standardschnittstelle festzulegen.
Description	Geben Sie die Beschreibung des Hosts ein.
Monitored by	Wählen Sie aus, ob der Host überwacht wird von: **Server** – der Host wird vom Zabbix Server überwacht; **Proxy** – der Host wird von einem eigenständigen Proxy überwacht; **Proxy group** – der Host wird von einer Proxygruppe überwacht.
Proxy	Der Name des zugewiesenen Proxys wird angezeigt (nur wenn der Zabbix Server einen Proxy aus der ausgewählten Proxygruppe zugewiesen hat). Dieses Feld wird nur angezeigt, wenn der Host von einer Proxygruppe überwacht wird.

Parameter	Beschreibung
Enabled	Wenn das Kontrollkästchen aktiviert ist, ist der Host aktiviert und zur Überwachung bereit. Wenn das Kontrollkästchen deaktiviert ist, ist der Host deaktiviert und wird nicht überwacht: Bei passiven Datenanforderungen, die vom Zabbix Server/Proxy initiiert werden (z. B. Zabbix Agent, SNMP Agent, einfache Prüfungen), wird die Überwachung nach der Konfigurationssynchronisierung deaktiviert. Mit dem Host verknüpfte Auslöser und Aktionen werden ebenfalls erst nach dem Neuladen der Konfiguration deaktiviert. Bei aktiven Prüfungen des Zabbix Agents wird die Überwachung innerhalb des Zeitraums (etwa 5 Sekunden) beendet, in dem der Zabbix Agent die Information erhält, dass der Host deaktiviert wurde. Während dieses kurzen Zeitraums sammelt der Host weiterhin lokal Daten für die aktiven Prüfungen und versucht, diese an den Server/Proxy zu senden. Da der Host jedoch als *Disabled* markiert ist, weist der Server/Proxy die Daten zurück. Wenn Sie den Host deaktivieren, werden seine Datenpunkte sofort aus dem Verlaufscache entfernt (mit Ausnahme ihrer letzten Werte, die für Protokolle beibehalten werden).

Die Registerkarte **IPMI** enthält IPMI-Verwaltungsattribute.

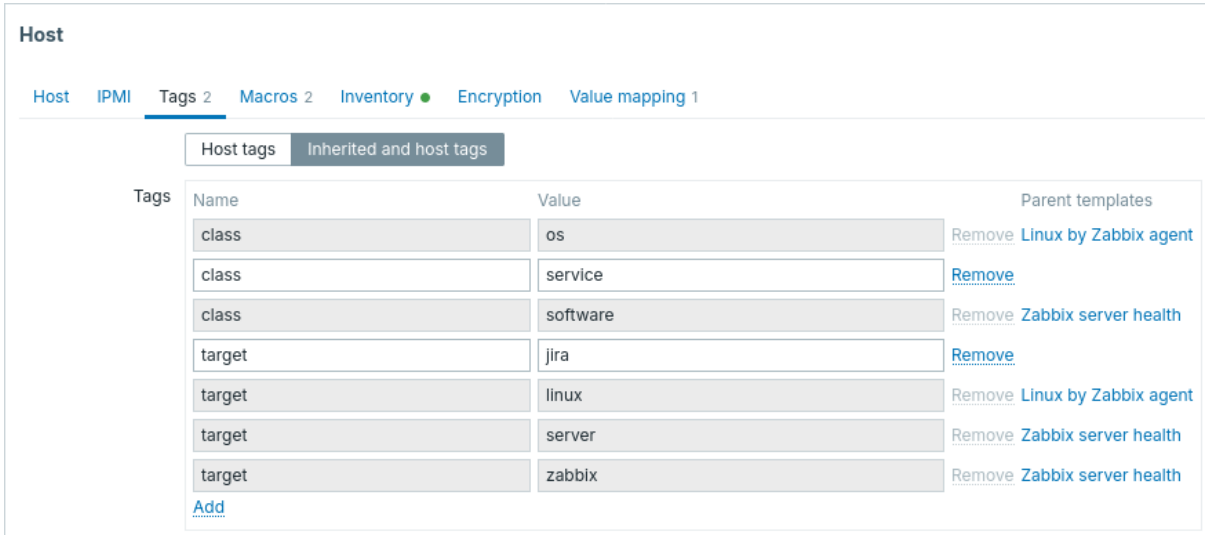
Parameter	Beschreibung
Authentication algorithm	Wählen Sie den Authentifizierungsalgorithmus aus.
Privilege level	Wählen Sie die Berechtigungsstufe aus.
Username	Benutzername für die Authentifizierung. Benutzermakros können verwendet werden.
Password	Passwort für die Authentifizierung. Benutzermakros können verwendet werden.

Auf der Registerkarte **Tags** können Sie Hostebenen-Tags definieren. Alle Probleme dieses Hosts werden mit den hier eingegebenen Werten getaggt.



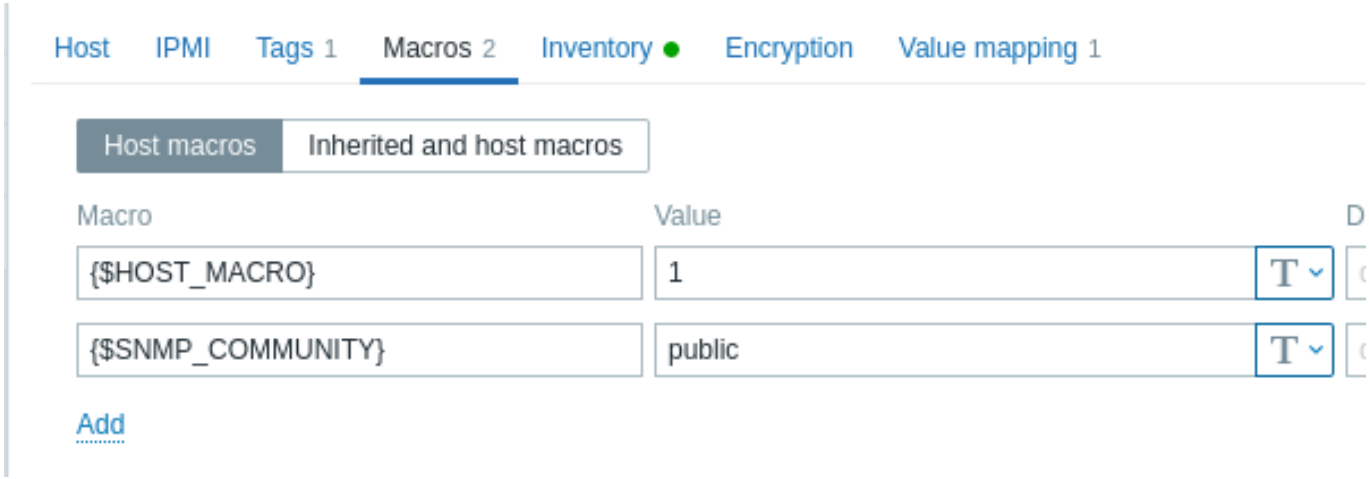
Benutzermakros, `{INVENTORY.*}`-Makros, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.IP}`, `{HOST.PORT}` und `{HOST.ID}`-Makros werden in Tags unterstützt.

Wenn Sie die Option *Inherited and host tags* auswählen, können Sie hier auch Tags auf Vorlagenebene anzeigen. Dort werden alle für den Host definierten Benutzertags sowie deren Herkunft angezeigt.

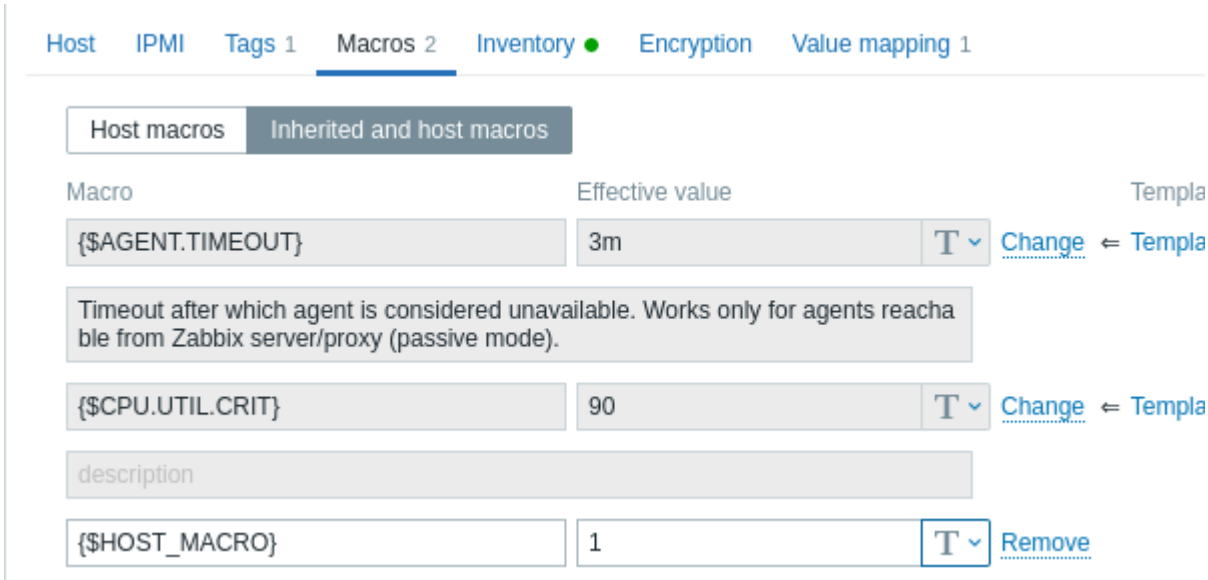


Zur Vereinfachung werden Links zu den entsprechenden Vorlagen bereitgestellt.

Auf der Registerkarte **Macros** können Sie Hostebenen-Benutzermakros als Name-Wert-Paare definieren. Beachten Sie, dass Makrowerte als Klartext, geheimer Text oder Vault-Secret gespeichert werden können. Das Hinzufügen einer Beschreibung wird ebenfalls unterstützt.

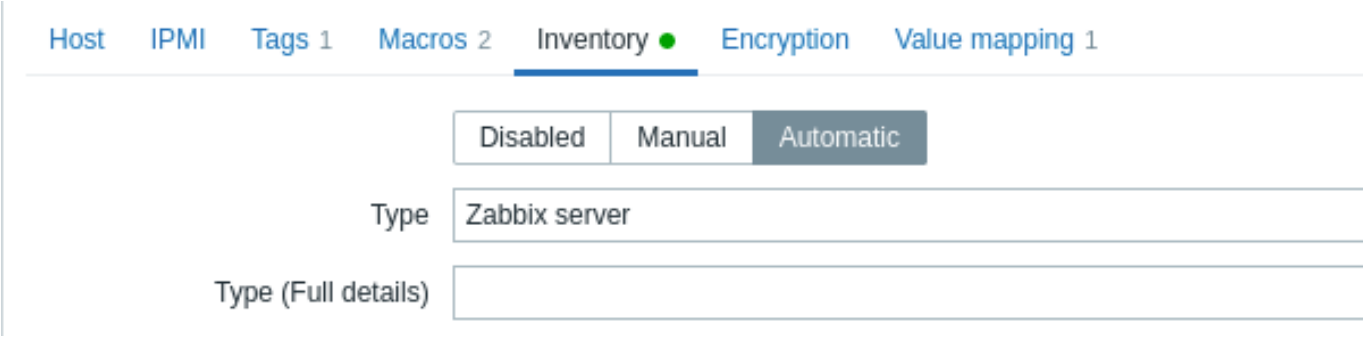


Wenn Sie die Option *Inherited and host macros* auswählen, können Sie hier auch Benutzermakros auf Vorlagen- und globaler Ebene anzeigen. Dort werden alle für den Host definierten Benutzermakros zusammen mit dem aufgelösten Wert und ihrer Herkunft angezeigt.



Zur Vereinfachung werden Links zu den entsprechenden Vorlagen und zur globalen Makrokonfiguration bereitgestellt. Es ist auch möglich, ein Vorlagen- oder globales Makro auf Hostebene zu bearbeiten, wodurch effektiv eine Kopie des Makros auf dem Host erstellt wird.

Auf der Registerkarte **Inventory** können Sie Inventar-Informationen für den Host manuell eingeben. Sie können außerdem die Inventarbefüllung auf *Automatic* setzen oder die Inventarbefüllung für diesen Host deaktivieren.



Wenn das Inventar aktiviert ist (manuell oder automatisch), wird neben dem Namen der Registerkarte ein grüner Punkt angezeigt.

Verschlüsselung

Die Registerkarte **Verschlüsselung** ermöglicht es Ihnen, verschlüsselte Verbindungen mit dem Host zu erzwingen.

Parameter	Beschreibung
Verbindungen zum Host	Wie der Zabbix Server oder Proxy eine Verbindung zum Zabbix Agent auf einem Host herstellt: ohne Verschlüsselung (Standard), mit PSK (Pre-Shared Key) oder mit Zertifikat.
Verbindungen vom Host	Wählen Sie aus, welche Arten von Verbindungen vom Host aus zulässig sind (d. h. vom Zabbix Agent und Zabbix sender). Mehrere Verbindungstypen können gleichzeitig ausgewählt werden (nützlich zum Testen und zum Wechsel zu einem anderen Verbindungstyp). Standard ist "Keine Verschlüsselung".
Aussteller	Zulässiger Aussteller des Zertifikats. Das Zertifikat wird zuerst mit der CA (Zertifizierungsstelle) validiert. Wenn es gültig und von der CA signiert ist, kann das Feld *Aussteller* verwendet werden, um die zulässige CA weiter einzuschränken. Dieses Feld ist vorgesehen, wenn Ihre Zabbix-Installation Zertifikate von mehreren CAs verwendet. Wenn dieses Feld leer ist, wird jede CA akzeptiert.
Betreff	Zulässiger Betreff des Zertifikats. Das Zertifikat wird zuerst mit der CA validiert. Wenn es gültig und von der CA signiert ist, kann das Feld *Betreff* verwendet werden, um nur einen Wert der Zeichenfolge *Betreff* zuzulassen. Wenn dieses Feld leer ist, wird jedes gültige, von der konfigurierten CA signierte Zertifikat akzeptiert.
PSK identity	Identitätszeichenfolge des Pre-Shared Key. Geben Sie in der PSK identity keine sensiblen Informationen an, da sie unverschlüsselt über das Netzwerk übertragen wird, um dem Empfänger mitzuteilen, welche PSK verwendet werden soll.
PSK	Pre-Shared Key (Hex-String). Maximale Länge: 512 Hex-Zeichen (256-Byte-PSK), wenn Zabbix die Bibliothek GnuTLS oder OpenSSL verwendet, 64 Hex-Zeichen (32-Byte-PSK), wenn Zabbix die Bibliothek mbed TLS (PolarSSL) verwendet. Beispiel: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952

Wertzuordnung

Auf der Registerkarte **Wertzuordnung** können Sie eine benutzerfreundliche Darstellung von Datenpunkten in Wertzuordnungen konfigurieren.

Konfigurieren einer Hostgruppe

Übersicht

Host-Gruppen werden für die logische Gruppierung von Hosts und die Zuweisung von Benutzerberechtigungen zu ihnen verwendet.

Jedem Host muss mindestens eine Host-Gruppe zugewiesen sein. Ein Host kann mehreren Host-Gruppen angehören, und jede Host-Gruppe kann mehrere Hosts enthalten.

Beachten Sie, dass in Zabbix alle Berechtigungen auf Gruppen basieren: Benutzergruppen, Host-Gruppen und Vorlagengruppen. Selbst wenn ein einzelner Benutzer Zugriff auf einen einzelnen Host benötigt, wird dieser gewährt, indem der Benutzer zu einer Benutzergruppe hinzugefügt und dieser Benutzergruppe Berechtigungen für die Host-Gruppe zugewiesen werden, die diesen Host enthält.

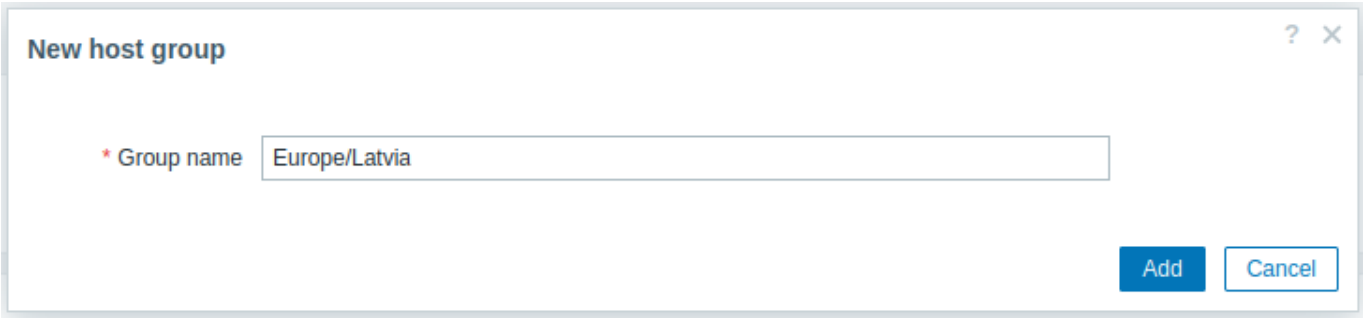
Konfiguration

Attention:
Nur Superadmin-Benutzer können Host-Gruppen erstellen.

Es gibt zwei Möglichkeiten, eine Host-Gruppe im Zabbix Frontend zu erstellen.

Option eins:

1. Gehen Sie zu *Datenerfassung* > *Host-Gruppen*.
2. Klicken Sie oben rechts auf dem Bildschirm auf *Host-Gruppe erstellen*.
3. Geben Sie den Gruppennamen im Formular ein.



Option zwei: Geben Sie beim Konfigurieren eines Hosts einen nicht vorhandenen Gruppennamen in das Eingabefeld *Host-Gruppen* ein.

Sobald die Host-Gruppe erstellt wurde, können Sie auf den Gruppennamen in der Liste unter *Datenerfassung* > *Host-Gruppen* klicken, um den Gruppennamen zu bearbeiten, die Gruppe zu klonen oder die Gruppe zu löschen.

Das Löschen einer Host-Gruppe entfernt nur die logische Gruppe, nicht die Hosts in der Gruppe. Es ist nicht möglich, eine Host-Gruppe zu löschen, die die einzige Gruppe für einen vorhandenen Host ist.

Erstellen von Host-Untergruppen

Eine Host-Untergruppe (oder verschachtelte Host-Gruppe) ist ein untergeordnetes Element der übergeordneten Host-Gruppe, die sie enthält.

Eine Untergruppe wird erstellt, indem im Eingabefeld für den Gruppennamen der Schrägstrich '/' verwendet wird, um ihre Beziehung zu der/den übergeordneten Gruppe(n) anzugeben. Zum Beispiel:

- die Eingabe von `Europe/Latvia` erstellt die Untergruppe `Europe/Latvia` der übergeordneten Gruppe `Europe`.
- die Eingabe von `Europe/Latvia/Riga/Zabbix servers` erstellt die entsprechende Untergruppe innerhalb der verschachtelten übergeordneten Gruppen `Europe`, `Europe/Latvia`, `Europe/Latvia/Riga`.

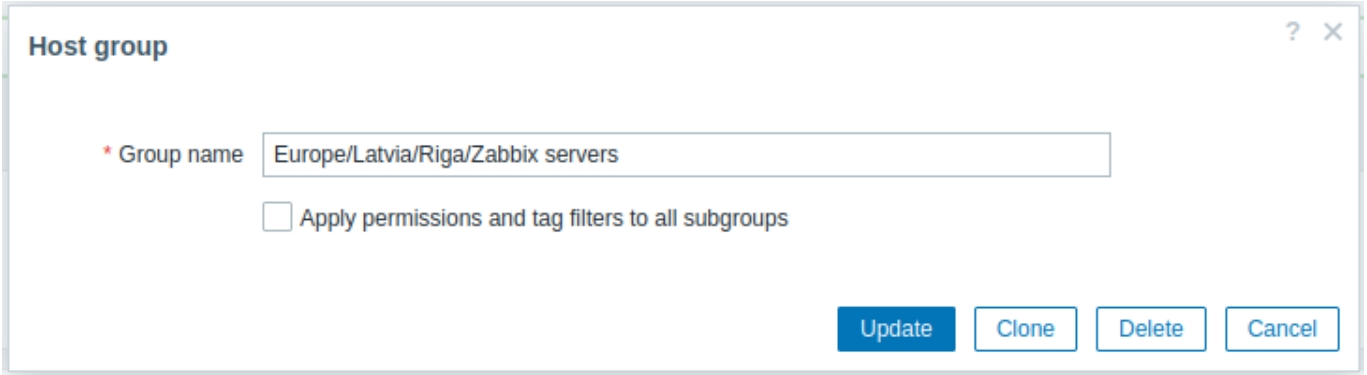
Beim Erstellen einer Untergruppe ist die Verwendung von führenden oder nachgestellten Schrägstrichen sowie mehrerer Schrägstriche hintereinander nicht zulässig. Das Escaping von '/' wird nicht unterstützt.

Es ist nicht erforderlich, vor dem Erstellen einer Untergruppe übergeordnete Host-Gruppe(n) zu erstellen. Sie können wählen, ob Sie mit dem Erstellen einer Untergruppe (zum Beispiel `Europe/Latvia`) oder einer beliebigen übergeordneten Host-Gruppe bzw. mehrerer übergeordneter Host-Gruppen (in unserem Beispiel `Europe`) beginnen. Wenn Sie mit dem Erstellen einer Untergruppe beginnen, werden übergeordnete Host-Gruppe(n) **nicht** automatisch erstellt.

Berechtigungen für Host-Gruppen

- Beim Erstellen einer Untergruppe zu einer vorhandenen übergeordneten Host-Gruppe (zum Beispiel beim Erstellen von `Europe/Latvia`, wenn `Europe` bereits vorhanden ist) werden die Benutzergruppen-Berechtigungen für die Untergruppe von der übergeordneten Gruppe übernommen.
- Beim Erstellen einer übergeordneten Host-Gruppe zu einer vorhandenen Untergruppe (zum Beispiel beim Erstellen von `Europe`, wenn `Europe/Latvia` bereits vorhanden ist) werden keine Berechtigungen für die übergeordnete Gruppe gesetzt.

Beim Bearbeiten einer beliebigen Host-Gruppe können Sie außerdem die zusätzliche Option *Berechtigungen und Tag-Filter auf alle Untergruppen anwenden* festlegen.



Wenn Sie dieses Kontrollkästchen aktivieren und auf *Update* klicken, werden dieselben Berechtigungsstufen und Tag-Filter auf alle aktuellen und zukünftigen Untergruppen der bearbeiteten Host-Gruppe angewendet.

Wenn also einer beliebigen Benutzergruppe unterschiedliche Berechtigungen für die Untergruppen der bearbeiteten Host-Gruppe zugewiesen wurden, erhalten durch Aktivieren des Kontrollkästchens alle aktuellen und zukünftigen Untergruppen dieselben Benutzerberechtigungen und tagbasierten Berechtigungen wie die bearbeitete Gruppe.

Beachten Sie, dass diese Option nicht in der Datenbank gespeichert wird und vorhandene Berechtigungen überschreibt. Alle über diese Option vorgenommenen Änderungen können nur manuell rückgängig gemacht werden.

Inventar

Übersicht

Sie können das Inventar von vernetzten Geräten in Zabbix verwalten.

Im Zabbix-Frontend gibt es ein spezielles Menü *Inventar*. Dort werden jedoch zunächst keine Daten angezeigt, und dort geben Sie auch keine Daten ein. Inventardaten werden entweder manuell bei der Konfiguration eines Hosts oder automatisch mithilfe einiger Optionen zur automatischen Befüllung erstellt.

Inventar erstellen

Manueller Modus

Beim Konfigurieren eines Hosts können Sie auf der Registerkarte *Inventar* solche Details wie den Gerätetyp, die Seriennummer, den Standort, die verantwortliche Person, URLs usw. eingeben – Daten, die die Inventarinformationen ausfüllen.

Wenn die Host-Inventarinformationen eine URL enthalten und diese mit „http“ oder „https“ beginnt, wird im Abschnitt *Inventar* ein anklickbarer Link angezeigt.

Automatischer Modus

Das Host-Inventar kann auch automatisch ausgefüllt werden. Dazu muss beim Konfigurieren eines Hosts der Inventarmodus auf der Registerkarte *Inventar* auf *Automatisch* gesetzt werden.

Anschließend können Sie Datenpunkte für Hosts konfigurieren, um beliebige Host-Inventarfelder mit ihren Werten zu füllen. Geben Sie dazu in der Datenpunktkonfiguration das Zielfeld über das entsprechende Attribut an (mit der Bezeichnung *Datenpunkt füllt Host-Inventarfeld*).

Datenpunkte, die für die automatisierte Erfassung von Inventardaten besonders nützlich sind:

- system.hw.chassis[full|type|vendor|model|serial] - Standardwert ist [full], Root-Berechtigungen erforderlich
- system.hw.cpu[all|cpunum,full|maxfreq|vendor|model|curfreq] - Standardwert ist [all,full]
- system.hw.devices[pci|usb] - Standardwert ist [pci]
- system.hw.macaddr[interface,short|full] - Standardwert ist [all,full], interface ist ein regulärer Ausdruck
- system.sw.arch
- system.sw.os[name|short|full] - Standardwert ist [name]
- system.sw.packages[regexp,manager,short|full] - Standardwert ist [all,all,full].

Auswahl des Inventarmodus

Der Inventarmodus kann im Host-Konfigurationsformular ausgewählt werden.

Der Inventarmodus für neue Hosts wird standardmäßig basierend auf der Einstellung *Standardinventarmodus für Hosts* unter *Administration* > *Allgemein* > *Sonstiges* ausgewählt.

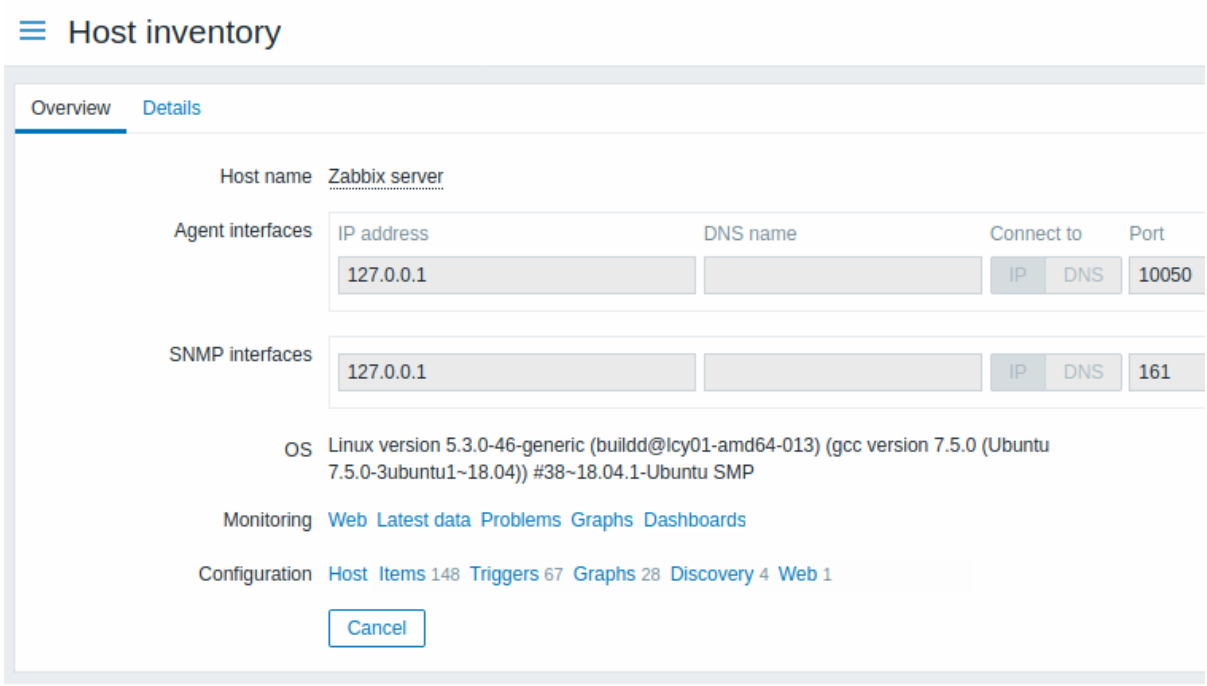
Für Hosts, die durch Netzwerkerkennungs- oder Autoregistrierungsaktionen hinzugefügt werden, kann eine Operation *Inventarmodus für Host festlegen* definiert werden, mit der der manuelle oder automatische Modus ausgewählt wird. Diese Operation überschreibt die Einstellung *Standardinventarmodus für Hosts*.

Bestandsübersicht

Die Details aller vorhandenen Bestandsdaten sind im Menü *Inventory* verfügbar.

In *Inventory* > *Overview* können Sie eine Host-Anzahl nach verschiedenen Feldern des Bestands abrufen.

In *Inventory* > *Hosts* können Sie alle Hosts sehen, für die Bestandsinformationen vorhanden sind. Wenn Sie auf den Hostnamen klicken, werden die Bestandsdetails in einem Formular angezeigt.



Die Registerkarte **Overview** zeigt:

Parameter	Beschreibung
Host name	Name des Hosts. Wenn Sie auf den Namen klicken, wird ein Menü mit den für den Host definierten Skripten geöffnet. Der Hostname wird mit einem orangefarbenen Symbol angezeigt, wenn sich der Host in Wartung befindet.
Visible name	Angezeigter Name des Hosts (falls definiert).
Host (Agent, SNMP, JMX, IPMI) interfaces	Dieser Block enthält Details zu den für den Host konfigurierten Schnittstellen.
OS	Bestandsfeld des Betriebssystems des Hosts (falls definiert).
Hardware	Bestandsfeld der Host-Hardware (falls definiert).
Software	Bestandsfeld der Host-Software (falls definiert).
Description	Hostbeschreibung.
Monitoring	Links zu Überwachungsbereichen mit Daten für diesen Host: *Web*, *Latest data*, *Problems*, *Graphs*, *Dashboards*.
Configuration	Links zu Konfigurationsbereichen für diesen Host: *Host*, *Items*, *Triggers*, *Graphs*, *Discovery*, *Web*. Die Anzahl der konfigurierten Entitäten wird nach jedem Link angezeigt.

Die Registerkarte **Details** zeigt alle Bestandsfelder an, die ausgefüllt sind (nicht leer).

Inventar-Makros

Es stehen Host-Inventar-Makros `{INVENTORY.*}` zur Verwendung in Benachrichtigungen zur Verfügung, zum Beispiel:

"Server in {INVENTORY.LOCATION1} hat ein Problem, die verantwortliche Person ist {INVENTORY.CONTACT1}, Telefonnummer {INVENTORY.POC.PRIMARY.PHONE.A1}."

Weitere Informationen finden Sie auf der Seite Unterstützte Makros.

Massenaktualisierung

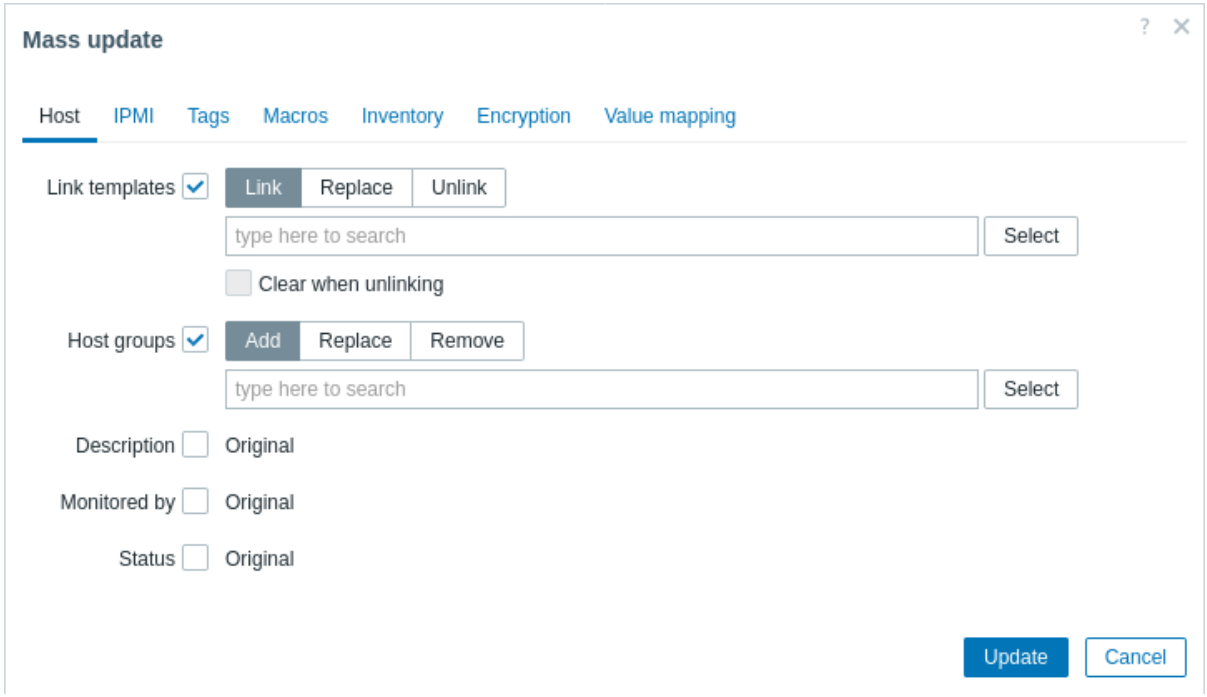
Übersicht

Manchmal möchten Sie ein bestimmtes Attribut für mehrere Hosts gleichzeitig ändern. Anstatt jeden einzelnen Host zur Bearbeitung zu öffnen, können Sie dafür die Funktion zur Massenaktualisierung verwenden.

Massenaktualisierung verwenden

Gehen Sie wie folgt vor, um mehrere Hosts gleichzeitig zu aktualisieren:

- Aktivieren Sie die Kontrollkästchen vor den Hosts, die Sie in der Host- liste aktualisieren möchten.
- Klicken Sie unterhalb der Liste auf *Massenaktualisierung*.
- Navigieren Sie zu der Registerkarte mit den erforderlichen Attributen (*Host*, *IPMI*, *Tags*, *Makros*, *Inventar*, *Verschlüsselung* oder *Wertzuordnung*).
- Aktivieren Sie die Kontrollkästchen der zu aktualisierenden Attribute und geben Sie einen neuen Wert dafür ein.



Bei der Auswahl der entsprechenden Schaltfläche zur Aktualisierung der **Vorlagen**verknüpfung stehen folgende Optionen zur Verfügung:

- *Verknüpfen* – geben Sie an, welche zusätzlichen Vorlagen verknüpft werden sollen.
- *Ersetzen* – geben Sie an, welche Vorlagen verknüpft werden sollen, während alle zuvor mit den Hosts verknüpften Vorlagen getrennt werden.
- *Trennen* – geben Sie an, welche Vorlagen getrennt werden sollen.

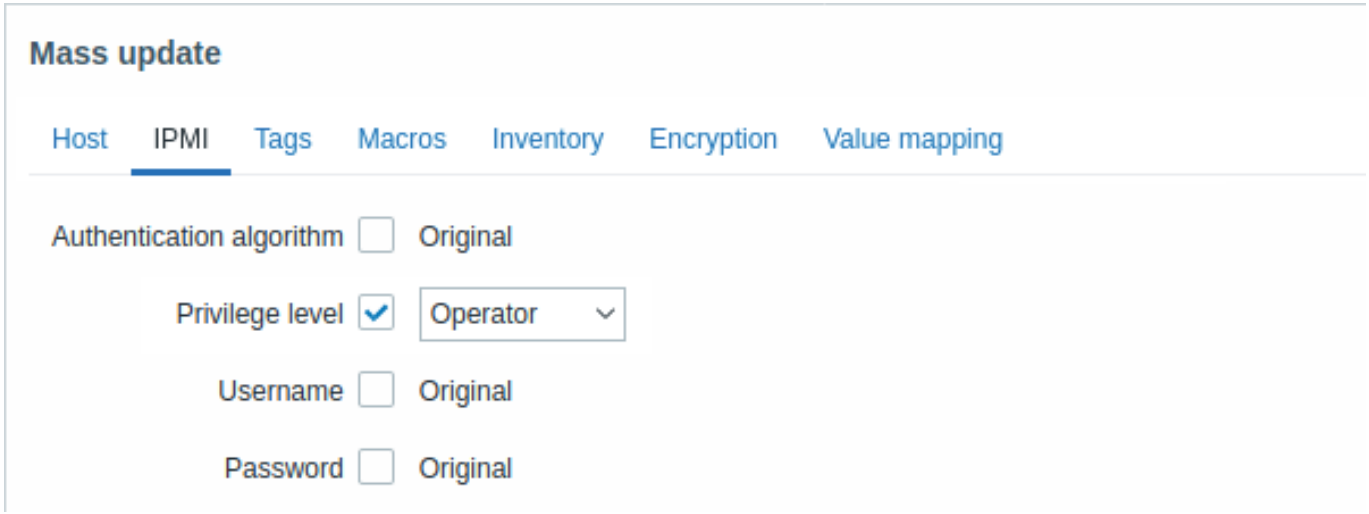
Um die zu verknüpfenden bzw. zu trennenden Vorlagen anzugeben, beginnen Sie im Autovervollständigungsfeld mit der Eingabe des Vorlagennamens, bis eine Dropdown-Liste mit den passenden Vorlagen angezeigt wird. Scrollen Sie nach unten, um die gewünschte Vorlage auszuwählen.

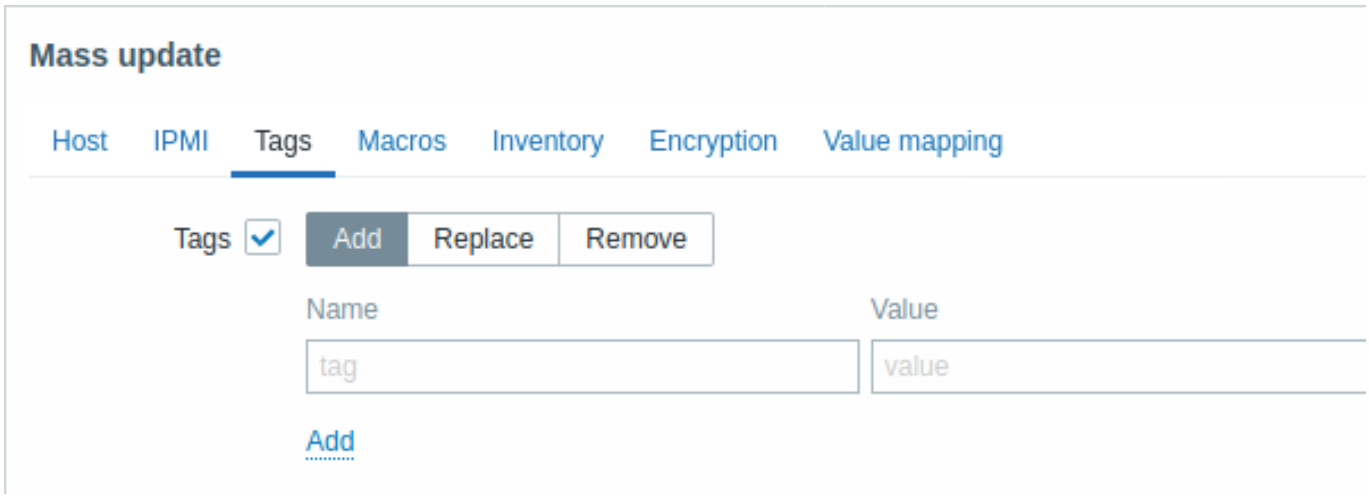
Mit der Option *Beim Trennen löschen* können Sie nicht nur zuvor verknüpfte Vorlagen trennen, sondern auch alle von ihnen geerbten Entitäten (Datenpunkte, Auslöser, Diagramme usw.) entfernen.

Bei der Auswahl der entsprechenden Schaltfläche zur Aktualisierung der **Hostgruppen** stehen folgende Optionen zur Verfügung:

- *Hinzufügen* – geben Sie zusätzliche Hostgruppen aus den vorhandenen Gruppen an oder geben Sie vollständig neue Hostgruppen für die Hosts ein.
- *Ersetzen* – entfernen Sie den Host aus allen vorhandenen Hostgruppen und ersetzen Sie diese durch die in diesem Feld angegebenen Gruppen (vorhandene oder neue Hostgruppen).
- *Entfernen* – entfernen Sie bestimmte Hostgruppen von den Hosts.

Diese Felder unterstützen die Autovervollständigung. Wenn Sie mit der Eingabe beginnen, wird eine Dropdown-Liste mit passenden Hostgruppen angezeigt. Ist die Hostgruppe neu, wird sie ebenfalls in der Dropdown-Liste angezeigt und durch *(neu)* nach der Zeichenfolge gekennzeichnet. Scrollen Sie nach unten, um sie auszuwählen.

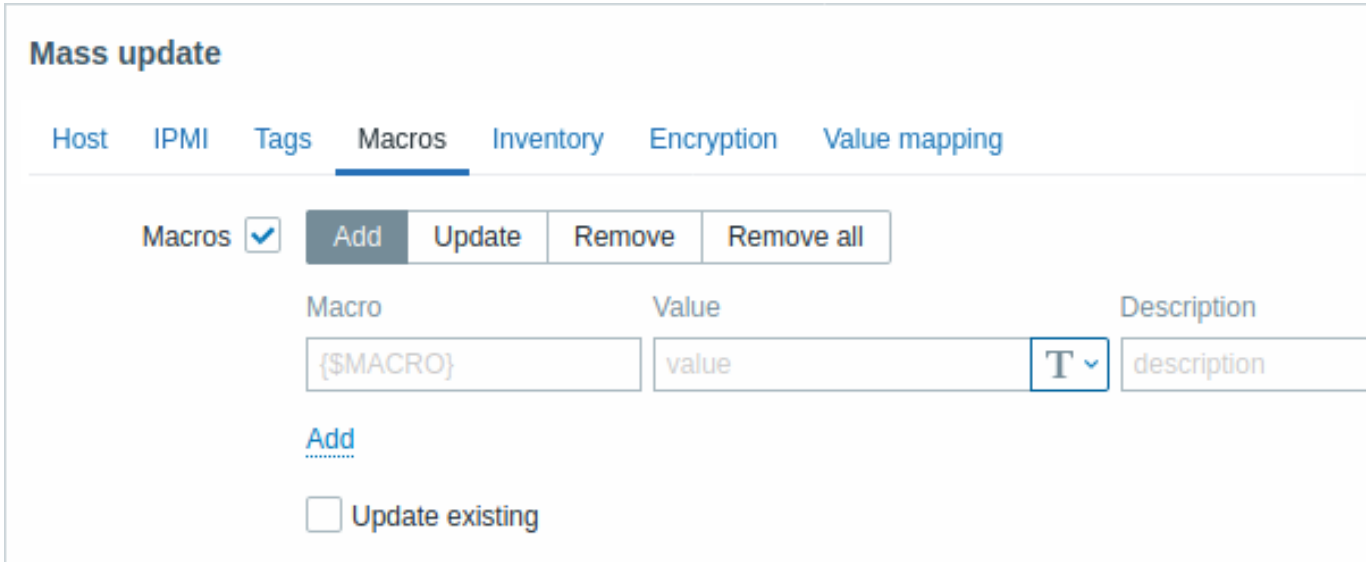




Bei der Auswahl der entsprechenden Schaltfläche zur Aktualisierung von Tags stehen folgende Optionen zur Verfügung:

- *Hinzufügen* – fügen Sie neue Tags für die Hosts hinzu.
- *Ersetzen* – entfernen Sie alle vorhandenen Tags vom Host und ersetzen Sie sie durch die unten angegebenen Tags.
- *Entfernen* – entfernen Sie die angegebenen Tags von den Hosts.

Benutzermakros, `{INVENTORY.*}`-Makros, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.IP}`, `{HOST.PORT}` und `{HOST.ID}` werden in Tags unterstützt. Beachten Sie, dass Tags mit demselben Namen, aber unterschiedlichen Werten nicht als „Duplikate“ betrachtet werden und demselben Host hinzugefügt werden können.

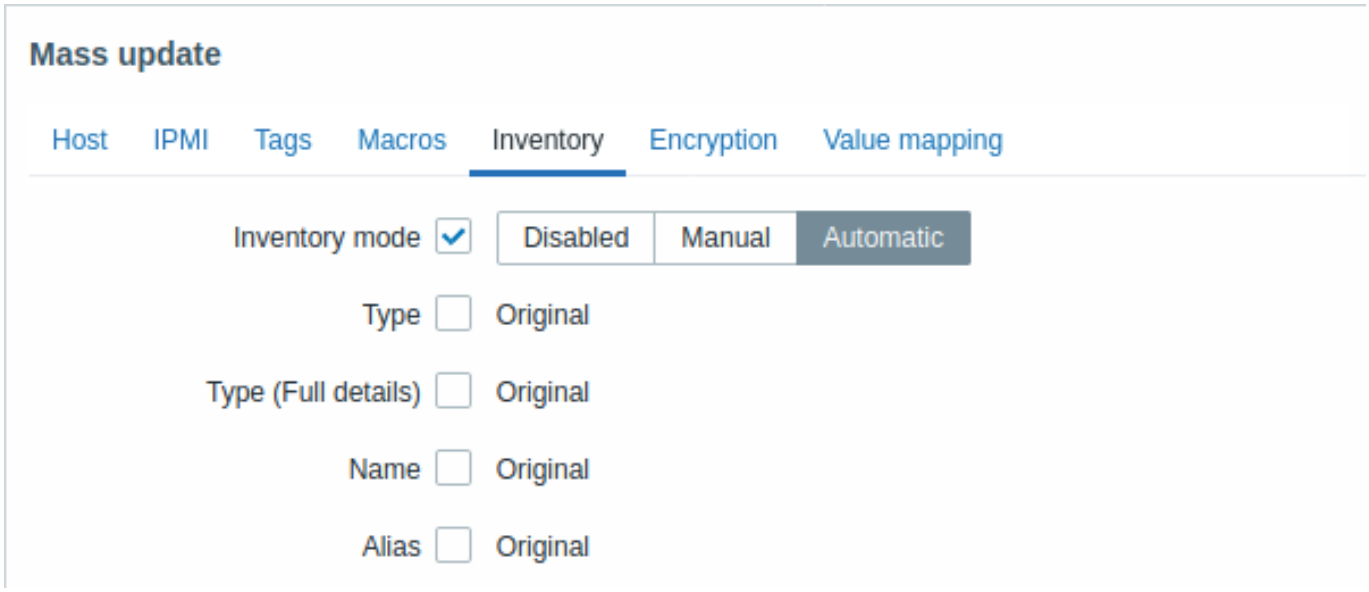


Bei der Auswahl der entsprechenden Schaltfläche zur Aktualisierung von Makros stehen folgende Optionen zur Verfügung:

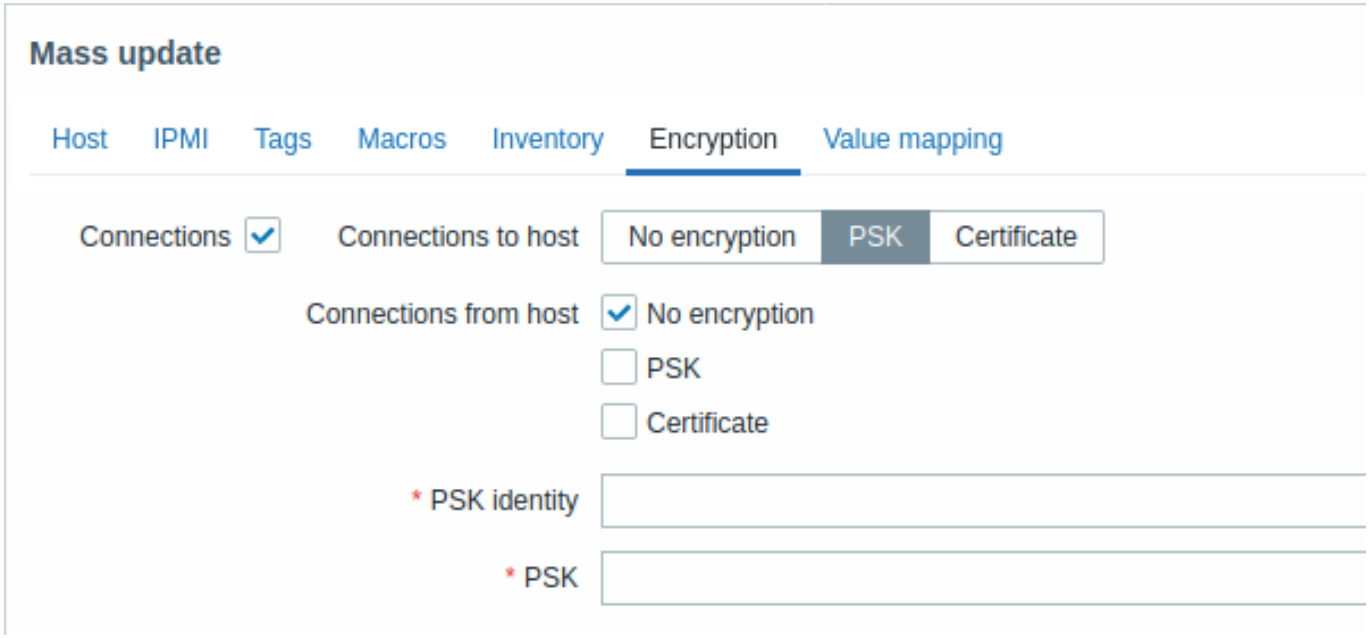
- *Hinzufügen* – geben Sie zusätzliche Benutzermakros für die Hosts an. Wenn das Kontrollkästchen *Vorhandene aktualisieren* aktiviert ist, werden Wert, Typ und Beschreibung für den angegebenen Makronamen aktualisiert. Wenn es nicht aktiviert ist

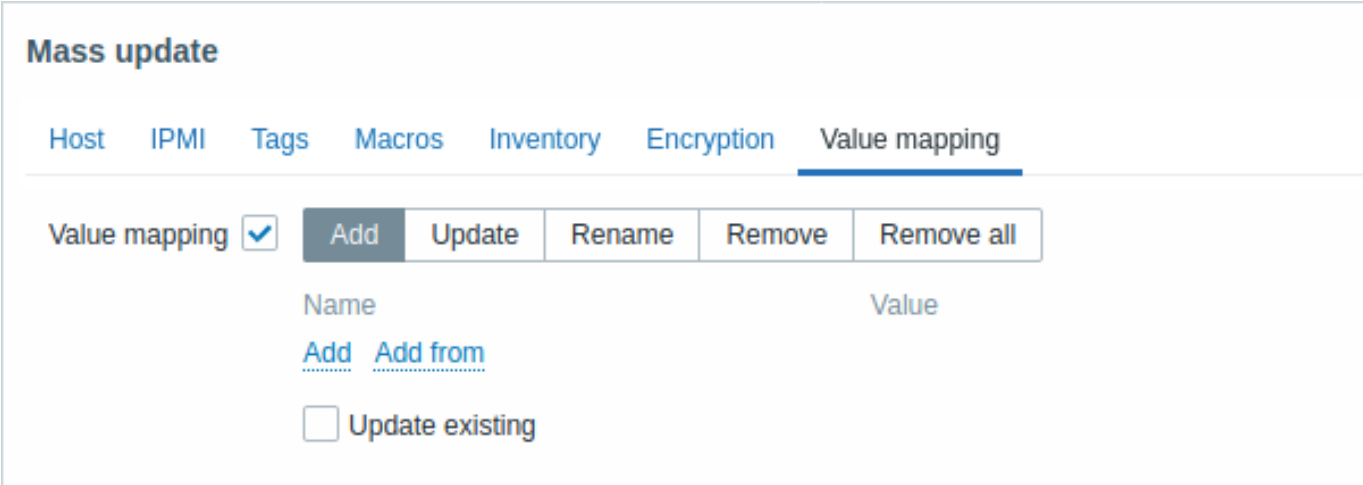
und auf dem bzw. den Hosts bereits ein Makro mit diesem Namen vorhanden ist, wird es nicht aktualisiert.

- *Aktualisieren* – ersetzen Sie die Werte, Typen und Beschreibungen der in dieser Liste angegebenen Makros. Wenn das Kontrollkästchen *Fehlende hinzufügen* aktiviert ist, wird ein Makro, das zuvor auf einem Host nicht vorhanden war, als neues Makro hinzugefügt. Wenn es nicht aktiviert ist, werden nur Makros aktualisiert, die bereits auf einem Host vorhanden sind.
- *Entfernen* – entfernen Sie die angegebenen Makros von den Hosts. Wenn das Kontrollkästchen *Ausgewählte ausnehmen* aktiviert ist, werden alle außer den in der Liste angegebenen Makros entfernt. Wenn es nicht aktiviert ist, werden nur die in der Liste angegebenen Makros entfernt.
- *Alle entfernen* – entfernen Sie alle Benutzermakros von den Hosts. Wenn das Kontrollkästchen *Ich bestätige, alle Makros zu entfernen* nicht aktiviert ist, wird ein neues Popup-Fenster geöffnet, in dem Sie aufgefordert werden, das Entfernen aller Makros zu bestätigen.



Damit Inventarfelder massenaktualisiert werden können, muss der *Inventarmodus* auf „Manuell“ oder „Automatisch“ gesetzt sein.





Für die Aktualisierung von Wertzuordnungen stehen Schaltflächen mit den folgenden Optionen zur Verfügung:

- *Hinzufügen* – fügen Sie den Hosts Wertzuordnungen hinzu. Wenn Sie *Vorhandene aktualisieren* aktivieren, werden alle Eigenschaften der Wertzuordnung mit diesem Namen aktualisiert. Andernfalls wird eine bereits vorhandene Wertzuordnung mit diesem Namen nicht aktualisiert.
- *Aktualisieren* – aktualisieren Sie vorhandene Wertzuordnungen. Wenn Sie *Fehlende hinzufügen* aktivieren, wird eine Wertzuordnung, die zuvor auf einem Host nicht vorhanden war, als neue Wertzuordnung hinzugefügt. Andernfalls werden nur Wertzuordnungen aktualisiert, die bereits auf einem Host vorhanden sind.
- *Umbenennen* – geben Sie einer vorhandenen Wertzuordnung einen neuen Namen.
- *Entfernen* – entfernen Sie die angegebenen Wertzuordnungen von den Hosts. Wenn Sie *Ausgewählte ausnehmen* aktivieren, werden alle Wertzuordnungen **außer** den angegebenen entfernt.
- *Alle entfernen* – entfernen Sie alle Wertzuordnungen von den Hosts. Wenn das Kontrollkästchen *Ich bestätige, alle Wertzuordnungen zu entfernen* nicht aktiviert ist, wird ein neues Popup-Fenster geöffnet, in dem Sie aufgefordert werden, das Entfernen zu bestätigen.

Wenn Sie alle erforderlichen Änderungen vorgenommen haben, klicken Sie auf *Aktualisieren*. Die Attribute werden für alle ausgewählten Hosts entsprechend aktualisiert.

Datenpunkte

Übersicht

Ein Datenpunkt ist eine einzelne Metrik.

Datenpunkte werden zum Sammeln von Daten verwendet. Sobald Sie einen Host konfiguriert haben, müssen Sie Datenpunkte hinzufügen, um tatsächliche Daten zu erhalten. Eine Möglichkeit, schnell viele Datenpunkte hinzuzufügen, besteht darin, eine der vordefinierten Vorlagen mit einem Host zu verknüpfen. Für eine optimierte Systemleistung kann es jedoch erforderlich sein, die Vorlagen fein abzustimmen, sodass sie genau so viele Datenpunkte und so häufige Überwachung enthalten wie nötig.

Um festzulegen, welche Art von Daten von einem Host erfasst werden soll, verwenden Sie den Datenpunkt-Schlüssel. Ein Datenpunkt mit dem Schlüsselnamen **system.cpu.load** erfasst beispielsweise Daten zur Prozessorlast, während ein Datenpunkt mit dem Schlüsselnamen **net.if.in** Informationen zum eingehenden Datenverkehr sammelt.

Zusätzliche Parameter können in eckigen Klammern nach dem Schlüsselnamen angegeben werden. Zum Beispiel gibt system.cpu.load**[avg5]** die durchschnittliche Prozessorlast der letzten 5 Minuten zurück, während net.if.in**[eth0]** den eingehenden Datenverkehr auf der Schnittstelle „eth0“ anzeigt.

> **Note:**
> In den einzelnen Abschnitten zu Datenpunkt-Typen finden Sie alle unterstützten Datenpunkt-Typen und Datenpunkt-Schlüssel.

Fahren Sie mit Erstellen und Konfigurieren eines Datenpunkts fort.

Erstellen eines Datenpunkts

Übersicht

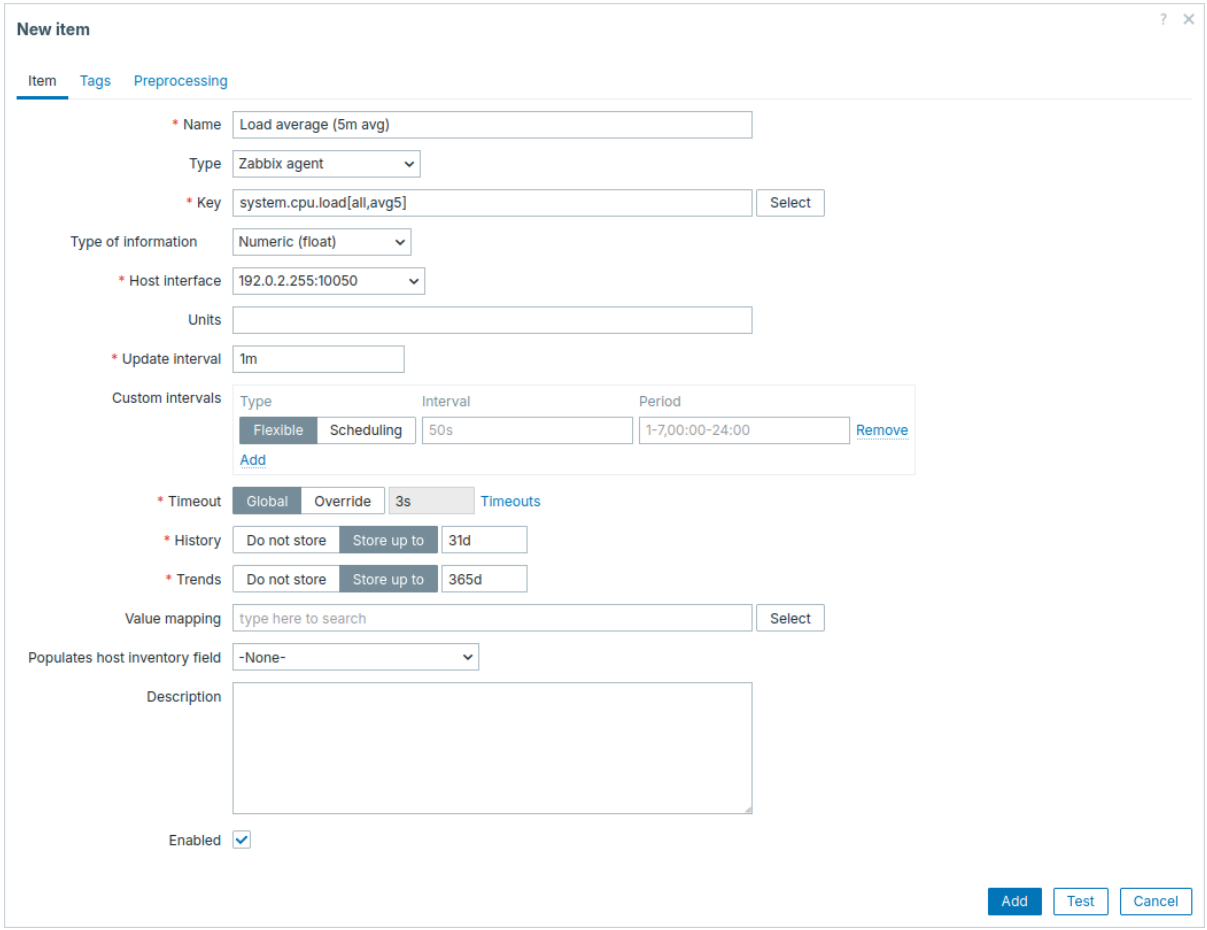
Um einen Datenpunkt im Zabbix Frontend zu erstellen, gehen Sie wie folgt vor:

1. Gehen Sie zu *Datensammlung* > *Hosts*.
2. Klicken Sie in der Zeile des Hosts auf *Datenpunkte*.
3. Klicken Sie in der oberen rechten Ecke des Bildschirms auf *Datenpunkt erstellen*.
4. Geben Sie die Parameter des Datenpunkts im Formular ein.

Alternativ können Sie einen Datenpunkt erstellen, indem Sie einen vorhandenen Datenpunkt öffnen, auf die Schaltfläche *Klonen* klicken und ihn unter einem anderen Namen speichern.

Konfiguration

Die Registerkarte **Datenpunkt** enthält allgemeine Attribute des Datenpunkts.



Alle Pflichtfelder sind mit einem roten Sternchen gekennzeichnet.

Parameter	Beschreibung
Name	Name des Datenpunkts. Benutzermakros werden unterstützt.
Typ	Datenpunkttyp. Siehe die einzelnen Abschnitte zu Datenpunkttypen.
Schlüssel	Datenpunktschlüssel (bis zu 2048 Zeichen). Die unterstützten Datenpunktschlüssel sind in den einzelnen Abschnitten zu den Datenpunkttypen aufgeführt. Der Schlüssel muss innerhalb eines einzelnen Hosts eindeutig sein. Wenn der Schlüsseltyp „Zabbix agent“, „Zabbix agent (aktiv)“ oder „Einfache Prüfung“ lautet, muss der Schlüsselwert vom Zabbix agent oder Zabbix server unterstützt werden. Siehe auch: das korrekte Schlüsselformat.

Parameter	Beschreibung
Informationstyp	Der Datentyp, der zur Validierung des Datenpunktwerts und zu seiner Speicherung in der Datenbank verwendet wird, jeweils nach etwaigen Konvertierungen: **Numerisch (vorzeichenlos)** – vorzeichenlose 64-Bit-Ganzzahl; **Numerisch (Gleitkommazahl)** – 64-Bit-Gleitkommazahl; **Zeichen** – kurzer Text; **Protokoll** – langer Text mit oder ohne Protokolleigenschaften (Zeitstempel, Quelle, Schweregrad, Ereignis-ID des Protokolls); **Text** – langer Text; **Binär** – Binärzahl (nur für abhängige Datenpunkte unterstützt); **JSON** – strukturierte JSON-Daten, nativ in der Datenbank gespeichert (für berechnete Datenpunkte nicht unterstützt). Datenpunkte, die Werte mit dem Datentyp „Binär“ oder „JSON“ zurückgeben, werden in Formeln für berechnete Datenpunkte oder Ausdrücken von Auslösern nicht unterstützt. Unter Datenlimits für Datenpunkte finden Sie die Speichergrenzen für die einzelnen Datentypen und Datenbank-Backends. Bei Datenpunkten, die Daten in einem einzigen bestimmten Format zurückgeben, wird automatisch ein passender Datentyp ausgewählt – beispielsweise geben `system.cpu.load` und `system.cpu.util` beide eine 64-Bit-Gleitkommazahl zurück, daher wird *Numerisch (Gleitkommazahl)* automatisch ausgewählt.
Host-Schnittstelle	Wählen Sie die Host-Schnittstelle aus. Dieses Feld ist verfügbar, wenn ein Datenpunkt auf Host-Ebene bearbeitet wird.
Einheiten	Wenn ein Einheitensymbol festgelegt ist, wendet Zabbix eine Nachverarbeitung auf den empfangenen Datenpunktwert an und zeigt ihn mit dem angegebenen Einheitenpräfix an. Unterstützte Einheitensymbole mit spezieller Formatierung (und Beispiele für empfangenen Datenpunktwert → angezeigten Wert): **B** – Byte (1024 → 1 KB) **Bps** – Byte pro Sekunde (1024 → 1 KBps) **s** – Sekunden, angezeigt mit bis zu drei größten Einheiten ungleich null (881764 → 10d 4h 56m) **uptime** – verstrichene Zeit im Format hh:mm:ss oder N Tage, hh:mm:ss (881764 → 10 days, 04:56:04) **unixtime** – Unix-Zeitstempel, formatiert als yyyy.mm.dd hh:mm:ss (881764 → 1970-01-11 04:56:04 AM); für eine korrekte Formatierung muss der empfangene Datenpunktwert *Numerisch (vorzeichenlos)* sein. Bei anderen Einheiten (z. B. Hz, W usw.) wird der empfangene Wert, wenn er 1000 überschreitet, durch 1000 geteilt und mit einem entsprechenden Präfix angezeigt (5000 → 5 KHz, 881764 → 881.76 KW). Wenn dem Einheitensymbol ein ! vorangestellt wird, sind Einheitenkonvertierung und Präfixierung deaktiviert (1024 !B → 1024 B, 61 !s → 61 s). Weitere Beispiele und Informationen zu Einheitensymbolen und Einheitenkonvertierung finden Sie unter Suffixe für Datenpunktwerte.
Aktualisierungsintervall	Ruft alle N Sekunden einen neuen Wert für diesen Datenpunkt ab. Das maximal zulässige Aktualisierungsintervall beträgt 86400 Sekunden (1 Tag). Zeitsuffixe werden unterstützt, z. B. 30s, 1m, 2h, 1d. Benutzermakros werden unterstützt. Ein einzelnes Makro muss das gesamte Feld ausfüllen. Mehrere Makros in einem Feld oder Makros zusammen mit Text werden nicht unterstützt. *Hinweis*: Das Aktualisierungsintervall kann nur auf „0“ gesetzt werden, wenn benutzerdefinierte Intervalle mit einem Wert ungleich null vorhanden sind. Wenn es auf „0“ gesetzt ist und ein benutzerdefiniertes Intervall (flexibel oder geplant) mit einem Wert ungleich null vorhanden ist, wird der Datenpunkt während der Dauer des benutzerdefinierten Intervalls abgefragt. *Beachten Sie*, dass die erste Abfrage des Datenpunkts nach seiner Aktivierung oder nach einer Änderung des Aktualisierungsintervalls früher als der konfigurierte Wert erfolgen kann. Neue Datenpunkte werden innerhalb von 60 Sekunden nach ihrer Erstellung geprüft, sofern sie nicht über ein geplantes oder flexibles Aktualisierungsintervall verfügen und das *Aktualisierungsintervall* auf 0 gesetzt ist. Ein vorhandener passiver Datenpunkt kann sofort durch Klicken auf die Schaltfläche *Jetzt ausführen* button auf seinen Wert abgefragt werden.

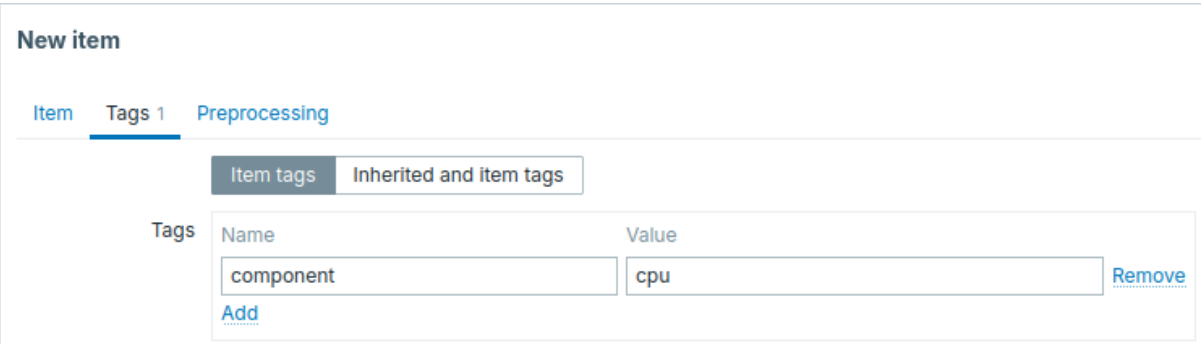
Parameter	Beschreibung
Benutzerdefinierte Intervalle	Sie können benutzerdefinierte Regeln für die Prüfung des Datenpunkts erstellen: **Flexibel** – erstellt eine Ausnahme für das *Aktualisierungsintervall* (Intervall mit einer anderen Häufigkeit). **Planung** – erstellt einen benutzerdefinierten Abfragezeitplan. Detaillierte Informationen finden Sie unter Benutzerdefinierte Intervalle. Zeitsuffixe werden im Feld *Intervall* unterstützt, z. B. 30s, 1m, 2h, 1d. Benutzermakros werden unterstützt. Ein einzelnes Makro muss das gesamte Feld ausfüllen. Mehrere Makros in einem Feld oder Makros zusammen mit Text werden nicht unterstützt.
Zeitüberschreitung	Legt die Zeitüberschreitung für die Datenpunktprüfung fest (verfügbar für unterstützte Datenpunkttypen). Wählen Sie die Option für die Zeitüberschreitung aus: **Global** – die Proxy-/globale Zeitüberschreitung wird verwendet (im ausgegrauten Feld *Zeitüberschreitung* angezeigt). **Überschreiben** – eine benutzerdefinierte Zeitüberschreitung wird verwendet (im Feld *Zeitüberschreitung* festgelegt; zulässiger Bereich: 1–600 s). Zeitsuffixe, z. B. 30s, 1m, und Benutzermakros werden unterstützt. Über den Link *Zeitüberschreitungen* können Sie Proxy-Zeitüberschreitungen oder globale Zeitüberschreitungen konfigurieren (wenn kein Proxy verwendet wird). Beachten Sie, dass der Link *Zeitüberschreitungen* nur Benutzern vom Typ *Super-Admin* angezeigt wird, die über Berechtigungen für die Frontend-Bereiche *Administration* > *Allgemein* oder *Administration* > *Proxies* verfügen.
Historie	Wählen Sie eine der folgenden Optionen aus: **Nicht speichern** – die Datenpunkthistorie wird nicht gespeichert. Dies ist für Master-Datenpunkte nützlich, wenn nur abhängige Datenpunkte die Historie speichern müssen. Diese Einstellung kann nicht durch globale Einstellungen des Housekeepers überschrieben werden. **Speichern bis** – gibt an, wie lange die detaillierte Historie in der Datenbank aufbewahrt wird (1 Stunde bis 25 Jahre). Ältere Daten werden vom Housekeeper entfernt. Die Dauer wird in Sekunden gespeichert. Zeitsuffixe werden unterstützt, z. B. 2h, 1d. Benutzermakros werden unterstützt. Der Wert *Speichern bis* kann global unter *Administration* > *Housekeeping* überschrieben werden.  Wenn eine globale Überschreibungseinstellung vorhanden ist, wird ein oranges Informationssymbol angezeigt. Wenn Sie den Mauszeiger darauf positionieren, wird eine Warnmeldung angezeigt, z. B. *Durch globale Housekeeper-Einstellungen überschrieben (1d)*. Es wird empfohlen, die aufgezeichneten Werte so kurz wie möglich aufzubewahren, um die Größe der Werthistorie in der Datenbank zu reduzieren. Statt eine lange Wertehistorie zu speichern, können Sie Trenddaten länger aufbewahren. Siehe auch Historie und Trends.
Trends	Wählen Sie eine der folgenden Optionen aus: **Nicht speichern** – Trends werden nicht gespeichert. Diese Einstellung kann nicht durch globale Einstellungen des Housekeepers überschrieben werden. **Speichern bis** – gibt an, wie lange aggregierte (stündlicher Minimalwert, Maximalwert, Durchschnitt, Anzahl) Historie in der Datenbank aufbewahrt wird (1 Tag bis 25 Jahre). Ältere Daten werden vom Housekeeper entfernt. Die Dauer wird in Sekunden gespeichert. Zeitsuffixe werden unterstützt, z. B. 24h, 1d. Benutzermakros werden unterstützt. Der Wert *Speichern bis* kann global unter *Administration* > *Housekeeping* überschrieben werden.  Wenn eine globale Überschreibungseinstellung vorhanden ist, wird ein oranges Informationssymbol angezeigt. Wenn Sie den Mauszeiger darauf positionieren, wird eine Warnmeldung angezeigt, z. B. *Durch globale Housekeeper-Einstellungen überschrieben (7d)*. *Hinweis:* Die Aufbewahrung von Trends ist für nicht numerische Daten – Zeichen, Protokoll und Text – nicht verfügbar. Siehe auch Historie und Trends.
Wertzuordnung	Wendet eine Wertzuordnung auf diesen Datenpunkt an. Eine Wertzuordnung ändert die empfangenen Werte nicht, sondern dient nur zur Anzeige von Daten. Sie funktioniert mit Datenpunkten vom Typ *Numerisch (vorzeichenlos)*, *Numerisch (Gleitkommazahl)* und *Zeichen*. Beispiel: „Windows-Dienststatus“.

Parameter	Beschreibung
Zeitformat des Protokolls	Nur für Datenpunkte vom Typ **Protokoll** verfügbar. Unterstützte Platzhalter: **y**: *Jahr (1970–2038)*. **M**: *Monat (01–12)*. **d**: *Tag (01–31)*. **h**: *Stunde (00–23)*. **m**: *Minute (00–59)*. **s**: *Sekunde (00–59)*. Wenn das Feld leer bleibt, wird der Zeitstempel auf 0 in der Unix-Zeit gesetzt, was den 1. Januar 1970 darstellt. Betrachten Sie beispielsweise die folgende Zeile aus der Protokolldatei des Zabbix agent: „ 23480:20100328:154718.045 Zabbix agent started. Zabbix 1.8.2 (revision 11211).“ Sie beginnt mit sechs Zeichenpositionen für die PID, gefolgt von Datum, Uhrzeit und dem restlichen Text. Das Zeitformat des Protokolls für diese Zeile wäre „pppppp:yyyyMMdd:hhmmss“. Beachten Sie, dass die Zeichen „p“ und „:“ Platzhalter sind und beliebige Zeichen außer „yMdhms“ sein können.
Füllt Inventarfeld des Hosts	Sie können ein Inventarfeld des Hosts auswählen, das mit dem Wert des Datenpunkts gefüllt wird. Dies funktioniert, wenn die automatische Inventarisierung für den Host aktiviert ist und Datenpunkte vom Typ *Numerisch (vorzeichenlos)*, *Numerisch (Gleitkommazahl)*, *Zeichen* oder *Text* verwendet werden.
Beschreibung	Geben Sie eine Beschreibung des Datenpunkts ein. Benutzermakros werden unterstützt.
Aktiviert	Aktivieren Sie das Kontrollkästchen, um den Datenpunkt zu aktivieren und verarbeiten zu lassen. Wenn Sie den Datenpunkt deaktivieren, wird er sofort aus dem Verlaufscache entfernt (mit Ausnahme seines letzten Werts, der für Protokolle beibehalten wird).
Neueste Daten	Klicken Sie auf den Link, um die neuesten Daten für den Datenpunkt anzuzeigen. Dieser Link ist nur verfügbar, wenn ein bereits vorhandener Datenpunkt bearbeitet wird.

Note:
Datenpunkttypspezifische Felder werden auf den entsprechenden Seiten beschrieben.

Note:
Beim Bearbeiten eines vorhandenen Datenpunkts auf Vorlagen-Ebene auf Host-Ebene sind mehrere Felder schreibgeschützt. Über den Link in der Kopfzeile des Formulars können Sie zur Vorlagenebene wechseln und die Felder dort bearbeiten. Beachten Sie dabei, dass Änderungen auf Vorlagenebene den Datenpunkt für alle Hosts ändern, mit denen die Vorlage verknüpft ist.

Auf der Registerkarte **Tags** können Sie Tags auf Datenpunktebene definieren.



Vorverarbeitung von Datenpunktwerten

Auf der Registerkarte **Vorverarbeitung** können Sie Transformationsregeln für die empfangenen Werte definieren.

Item-Test

Attention:
Um das Testen von Datenpunkten durchzuführen, stellen Sie sicher, dass die Systemzeit auf dem Server und dem Proxy synchronisiert ist. Falls die Serverzeit hinterherhinkt, kann das Testen von Datenpunkten die Fehlermeldung ”The task has been expired.” zurückgeben. Wenn auf dem Server und dem Proxy unterschiedliche Zeitzonen eingestellt sind, hat dies jedoch keinen Einfluss auf das Testergebnis.

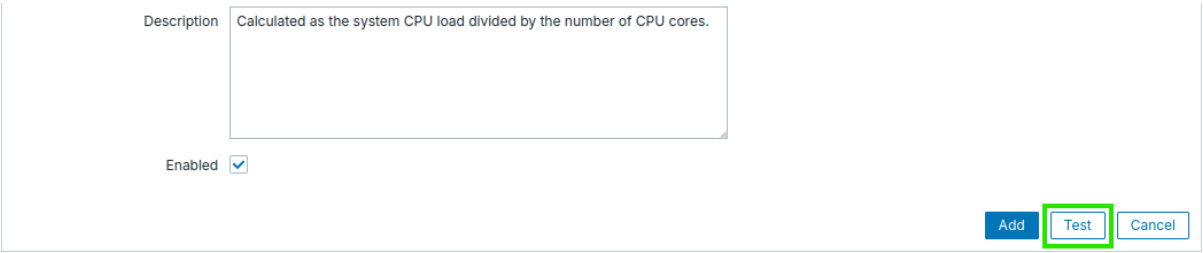
Es ist möglich, einen Datenpunkt zu testen und, wenn er korrekt konfiguriert ist, einen echten Wert zurückzuerhalten. Das Testen kann sogar erfolgen, bevor ein Datenpunkt gespeichert wird.

Das Testen ist für Host- und Vorlagen-Datenpunkte, Datenpunktprototypen und Low-Level-Discovery-Regeln verfügbar. Das Testen ist für aktive Datenpunkte nicht verfügbar.

Das Testen von Datenpunkten ist für die folgenden passiven Datenpunkttypen verfügbar:

- Zabbix Agent
- SNMP Agent (v1, v2, v3)
- IPMI Agent
- SSH-Prüfungen
- Telnet-Prüfungen
- JMX Agent
- Einfache Prüfungen (außer `icmpping*`, `vmware.*` Datenpunkte)
- Zabbix intern
- Berechnete Datenpunkte
- Externe Prüfungen
- Datenbankmonitor
- HTTP Agent
- Skript
- Browser

Um einen Datenpunkt zu testen, klicken Sie unten im Konfigurationsformular des Datenpunkts auf die Schaltfläche *Test*. Beachten Sie, dass die Schaltfläche *Test* für Datenpunkte, die nicht getestet werden können, deaktiviert ist (z. B. aktive Prüfungen, ausgeschlossene einfache Prüfungen).

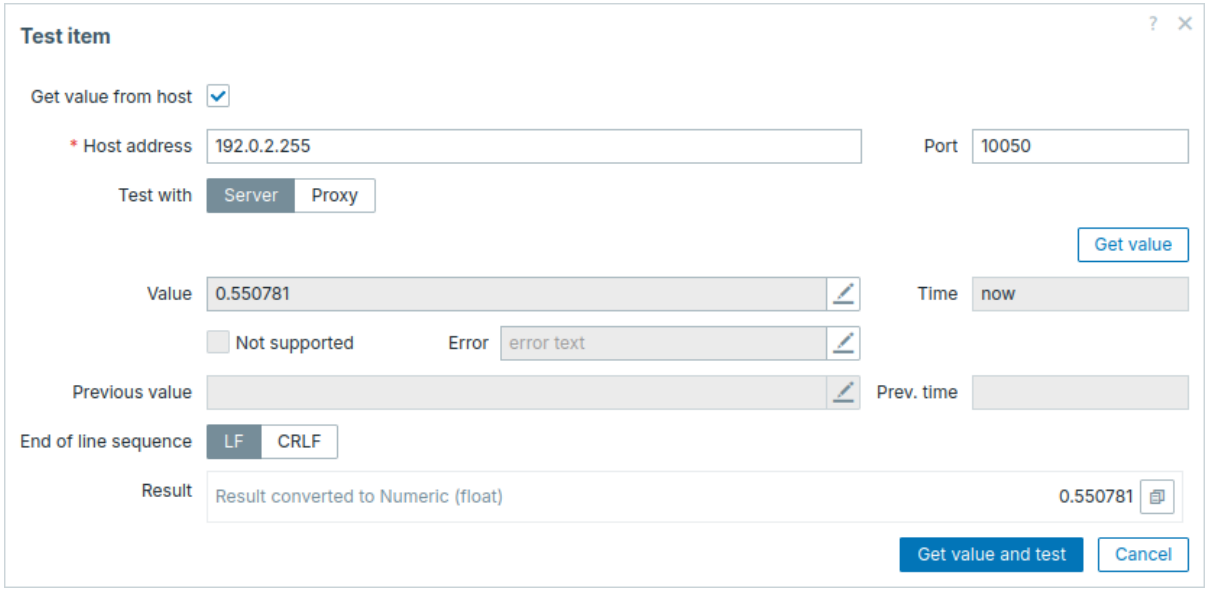


Das Formular zum Testen von Datenpunkten enthält Felder für die erforderlichen Host-Parameter (Host-Adresse, Port, Test mit Server/Proxy (Proxy-Name)) und datenpunktspezifische Details (z. B. SNMPv2-Community oder SNMPv3-Sicherheitsanmeldedaten). Diese Felder sind kontextabhängig:

- Die Werte werden, wenn möglich, vorab ausgefüllt, d. h. bei Datenpunkten, die einen Agent erfordern, indem die Informationen aus der ausgewählten Agent-Schnittstelle des Hosts übernommen werden.
- Die Werte müssen für Vorlagen-Datenpunkte manuell ausgefüllt werden.
- Makrowerte im Klartext werden aufgelöst.
- Felder, deren Wert (oder ein Teil des Werts) ein Secret- oder Vault-Makro ist, sind leer und müssen manuell eingegeben werden. Wenn ein Datenpunktparameter einen Secret-Makro-Wert enthält, wird die folgende Warnmeldung angezeigt: "Item contains user-defined macros with secret values. Values of these macros should be entered manually."
- Die Felder sind deaktiviert, wenn sie im Kontext des Datenpunkttyps nicht benötigt werden (z. B. sind das Feld für die Host-Adresse und das Proxy-Feld für berechnete Datenpunkte deaktiviert)

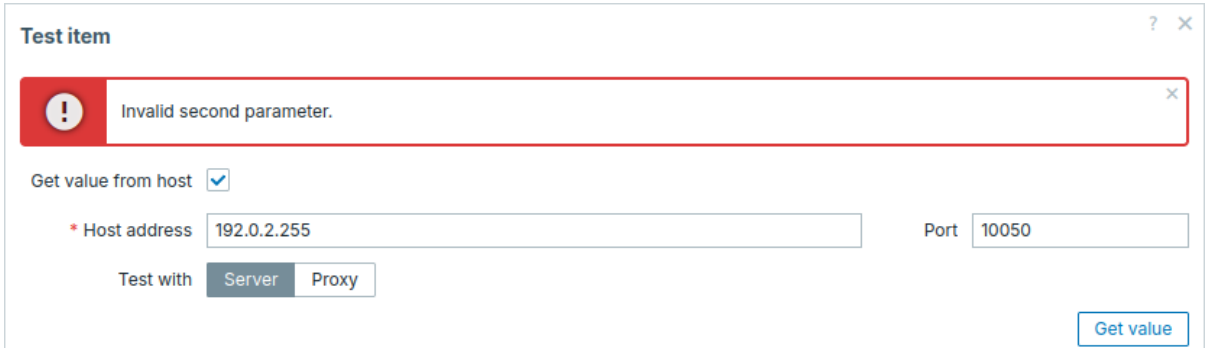
Um den Datenpunkt zu testen, klicken Sie auf *Get value*. Wenn der Wert erfolgreich abgerufen wurde, wird er das Feld *Value* füllen, wobei der aktuelle Wert (falls vorhanden) in das Feld *Previous value* verschoben wird. Gleichzeitig wird das Feld *Prev. time* berechnet, d. h. die Zeitdifferenz zwischen den beiden Werten (Klicks), und es wird versucht, eine EOL-Sequenz zu erkennen und auf CRLF umzuschalten, wenn im abgerufenen Wert "\n\r" erkannt wird.

Klicken Sie auf *Get value and test*, um die Vorverarbeitung zu testen.



Von einem Host abgerufene Werte und Testergebnisse werden auf eine maximale Größe von 512 KB begrenzt, wenn sie an das Frontend gesendet werden. Wenn ein Testergebnis gekürzt wurde, wird ein Warnsymbol angezeigt; fahren Sie mit der Maus darüber, um Details anzuzeigen. Wenn ein Wert gekürzt wird, kann dies bei aufeinanderfolgenden Tests, die große Werte vergleichen (z. B. JSON), zu einem Fehler bei der Datentypvalidierung führen; in solchen Fällen wird eine Fehlermeldung angezeigt. Beachten Sie, dass Daten, die größer als 512 KB sind, von Zabbix Server weiterhin vollständig verarbeitet werden.

Wenn die Konfiguration fehlerhaft ist, wird eine Fehlermeldung angezeigt, die die mögliche Ursache beschreibt.



Ein erfolgreich von einem Host abgerufener Wert kann auch zum Testen von Vorverarbeitungsschritten verwendet werden.

Schaltflächen des Formulars

Mit den Schaltflächen am unteren Rand des Formulars können Sie mehrere Vorgänge ausführen.

Add	Einen Datenpunkt hinzufügen. Diese Schaltfläche ist nur für neue Datenpunkte verfügbar.
Update	Die Eigenschaften eines Datenpunkts aktualisieren.
Clone	Einen weiteren Datenpunkt basierend auf den Eigenschaften des aktuellen Datenpunkts erstellen.
Execute now	Eine Prüfung auf einen neuen Datenpunktwert sofort ausführen. Wird nur für **passive** Prüfungen unterstützt (siehe weitere Details). *Hinweis*: Beim sofortigen Prüfen eines Werts wird der Konfigurations-Cache nicht aktualisiert. Daher berücksichtigt der Wert möglicherweise sehr aktuelle Änderungen an der Datenpunktkonfiguration nicht.
Test	Testen, ob die Datenpunktkonfiguration korrekt ist, indem ein Wert abgerufen wird.
Clear history and trends	Den Verlauf und die Trends des Datenpunkts löschen.
Delete	Den Datenpunkt löschen.
Cancel	Die Bearbeitung der Datenpunkteigenschaften abbrechen.

Datenpunkt-Datenlimits

Die Datenlimits für Datenpunkte hängen vom Datenpunkttyp und vom Datenbank-Backend ab.

Numerische Werte (unsigned) werden unverändert gespeichert. Wenn ein Gleitkommawert empfangen wird, wird nur der Ganzzahlteil gespeichert (z. B. 1.23 → 1).

Numerische Werte (float) werden unverändert gespeichert und unterstützen ungefähr 15 bis 17 Stellen Genauigkeit, mit einem Bereich von ungefähr -1.79E+308 bis 1.79E+308. Werte in wissenschaftlicher Notation werden ebenfalls unterstützt (z. B. 1.23E+7, 1e308, 1.1E-4).

Textwerte werden vor dem Speichern abgeschnitten, um dem Limit des Datenbank-Wertyps zu entsprechen:

Database	Type of information		
	Character	**Log**	**Text**
MySQL	255 Zeichen	65536 Bytes	65536 Bytes
PostgreSQL	255 Zeichen	65536 Zeichen	65536 Zeichen
SQLite (only Zabbix Proxy)	255 Zeichen	65536 Zeichen	65536 Zeichen

Binärwerte sind auf 16 MiB (16777216 Bytes) begrenzt. Werte, die dieses Limit überschreiten, werden verworfen, und ein entsprechender Fehler wird im Frontend angezeigt.

Werte mit dem JSON-Datentyp sind auf 128 MiB (134217728 Bytes) begrenzt. Werte, die dieses Limit überschreiten, werden verworfen, und ein entsprechender Fehler wird im Frontend angezeigt. Für MySQL/MariaDB sollten Sie erwägen, tägliche Partitionen für die Tabelle `history_json` einzurichten (die zum Speichern von JSON-Werten verwendet wird), da sie schnell groß werden und den Housekeeper über einen längeren Zeitraum blockieren kann. Wenn Ihr Anwendungsfall große JSON-Werte (1 MiB oder mehr) umfasst, lesen Sie, wie Sie Ihr System für Unterstützung großer JSON-Werte konfigurieren.

Wenn Zabbix beliebige Datenpunkte-Daten in die Datenbank schreibt, verwendet es INSERT-Abfragen und protokolliert diese (wenn `DebugLevel` auf 4 oder 5 gesetzt ist). Große Protokolleinträge werden auf 64 KB gekürzt.

Limit für benutzerdefinierte Skripte

Die verfügbare Länge benutzerdefinierter Skripte hängt von der verwendeten Datenbank ab:

Datenbank	Limit in Zeichen	Limit in Byte
MySQL	65535	65535
PostgreSQL	65535	nicht begrenzt
SQLite (nur Zabbix Proxy)	65535	nicht begrenzt

Item-Timeout

Der Item-Timeout legt fest, wie lange Zabbix warten soll, bevor die Prüfung als fehlgeschlagen abgebrochen wird.

Wenn der Timeout erreicht wird, wird die Prüfung abgebrochen, auch wenn die Datenerfassung noch nicht abgeschlossen ist. Wenn Daten nur teilweise empfangen werden, wird das Item als nicht unterstützt markiert und eine Fehlermeldung protokolliert (zum Beispiel, wenn bei einer SNMP-Prüfung Daten erfolgreich nur für einen von mehreren OIDs erfasst werden).

Für viele Item-Typen können Sie **flexible** Item-Timeouts festlegen:

- pro einzelnes Item
- pro Item-Typ (auf Proxy-Ebene)
- pro Item-Typ (auf globaler-Ebene)

Ein benutzerdefinierter Timeout *pro einzelnes Item* ist nützlich, wenn Sie für ein bestimmtes Item einen längeren Timeout festlegen möchten, während die Timeouts für andere Items niedrig bleiben.

Timeout-Priorität

1. Der Timeout für ein einzelnes Item hat Vorrang vor allen anderen Timeouts.
2. Timeouts auf Proxy-Ebene haben Vorrang vor globalen Timeouts.

Der Timeout aus der Zabbix-Server- oder Proxy-Konfiguration spielt bei Prüfungen, für die flexible Timeouts festgelegt sind, **keine** Rolle.

Flexible Timeout-Unterstützung

Flexible Datenpunkt-Timeouts werden für diese Datenpunkt-Typen unterstützt:

- Zabbix agent (sowohl passive als auch aktive Prüfungen)
- SNMP agent (außer veraltete SNMP-Prüfungen[1])
- Simple check (außer `icmpping*` und VMware-Datenpunkte[2])
- SSH agent
- Telnet agent
- External check

- Database monitor
- Script
- HTTP agent
- Browser

[1] Für veraltete SNMP-Prüfungen (Abfrage einer einzelnen OID) gelten die Timeout-Einstellungen des Server oder Proxy.
 [2] Für `icmpping*`-Datenpunkte wird der Timeout-Wert direkt im Datenpunkt-Schlüssel angegeben; außerdem gilt ein fest codierter maximal möglicher Timeout von 600 Sekunden. VMware-Überwachungsdatenpunkte verwenden ihren eigenen VMwareTimeout-Parameter aus der Server- oder Proxy-Konfiguration.

Nicht unterstützte Datenpunkte

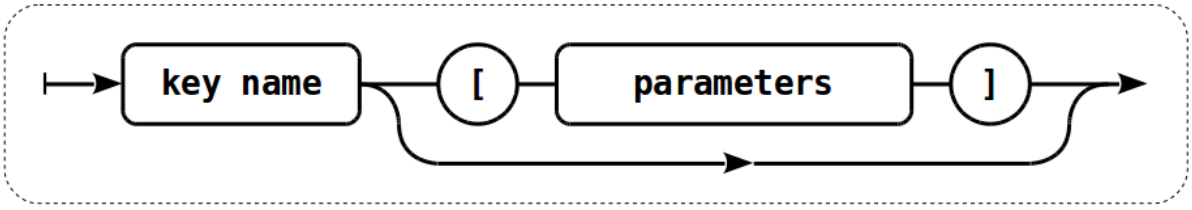
Nicht unterstützte Datenpunkte werden mit dem Status *Nicht unterstützt* gemeldet und weiterhin in ihrem normalen *Aktualisierungsintervall* erneut geprüft.

Ein Datenpunkt wird nicht unterstützt, wenn sein Wert aus irgendeinem Grund nicht abgerufen werden kann (z. B. Verbindungsfehler, keine Poller zur Verarbeitung des Datenpunkts konfiguriert). Außerdem bleiben Datenpunkte, die keine Verlaufsdaten erhalten, im Status *Nicht unterstützt*. Dies ist das erwartete Verhalten — ein Datenpunkt wechselt erst dann in den Status *Normal*, wenn er neue, gültige Verlaufsdaten erhält.

Wie Probleme werden auch nicht unterstützte Datenpunkte nur dann erneut ausgewertet, wenn neue Daten empfangen werden — selbst wenn für diesen Datenpunkt keine Verlaufsdaten mehr verfügbar sind. Mit anderen Worten: Datenpunkte und Auslöser ändern ihren Zustand ausschließlich beim Empfang neuer Daten. Enthält ein Auslöserausdruck jedoch eine Funktion für Datum und Uhrzeit und/oder nodata(), wird der Auslöser alle 30 Sekunden neu berechnet (siehe Berechnungszeit von Auslösern für Details).

Format des Datenpunkt-Schlüssels

Das Format des Datenpunkt-Schlüssels, einschließlich der Schlüsselparameter, muss den Syntaxregeln entsprechen. Die folgenden Abbildungen zeigen die unterstützte Syntax. Zulässige Elemente und Zeichen an jeder Stelle können durch das Folgen der Pfeile bestimmt werden - wenn ein Block über die Linie erreicht werden kann, ist er zulässig, andernfalls - ist er nicht zulässig.



Um einen gültigen Datenpunkt-Schlüssel zu erstellen, beginnt man mit der Angabe des Schlüsselnamens, danach hat man die Wahl, entweder Parameter anzugeben oder nicht - wie durch die beiden Linien dargestellt, denen gefolgt werden kann.

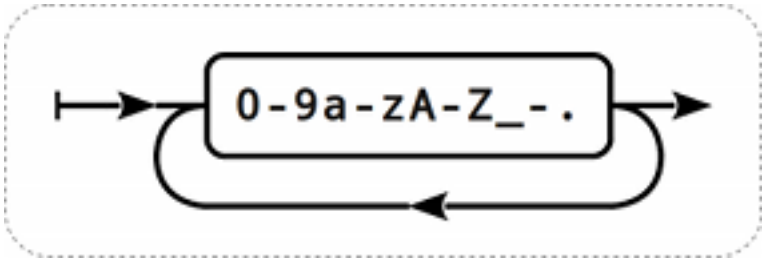
Schlüsselname

Der Schlüsselname selbst hat einen begrenzten Bereich zulässiger Zeichen, die einfach aufeinander folgen. Zulässige Zeichen sind:

`0-9a-zA-Z_-.`

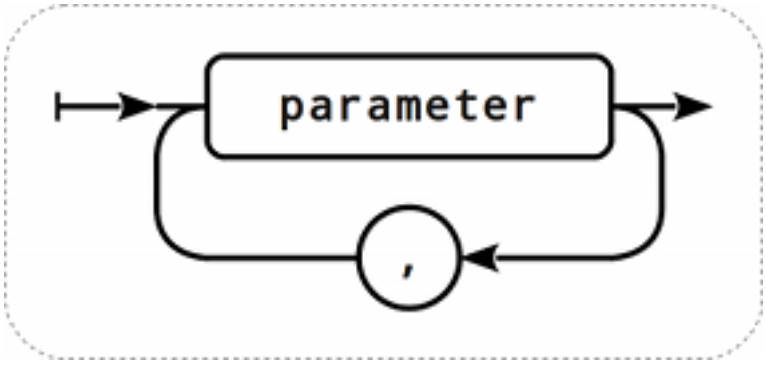
Das bedeutet:

- alle Zahlen;
- alle Kleinbuchstaben;
- alle Großbuchstaben;
- Unterstrich;
- Bindestrich;
- Punkt.

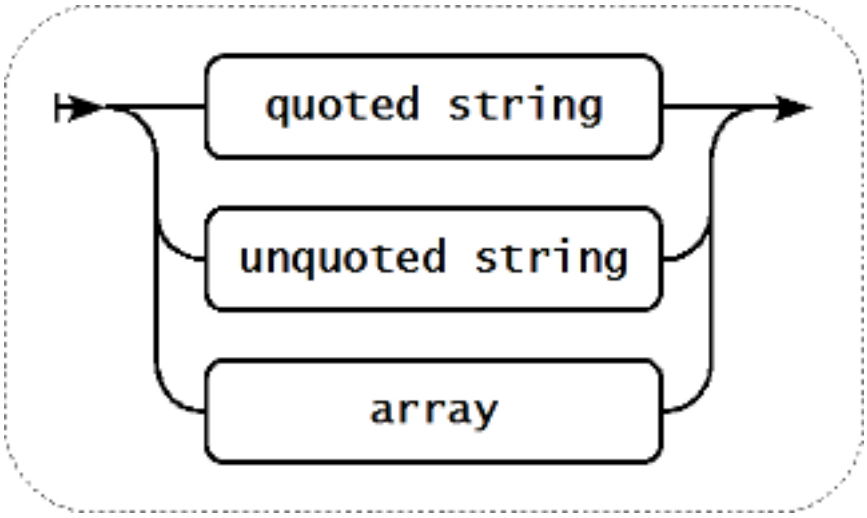


Schlüsselparameter

Ein Datenpunktschlüssel kann mehrere Parameter haben, die durch Kommas getrennt sind.



Jeder Schlüsselparameter kann entweder eine Zeichenfolge in Anführungszeichen, eine Zeichenfolge ohne Anführungszeichen oder ein Array sein.

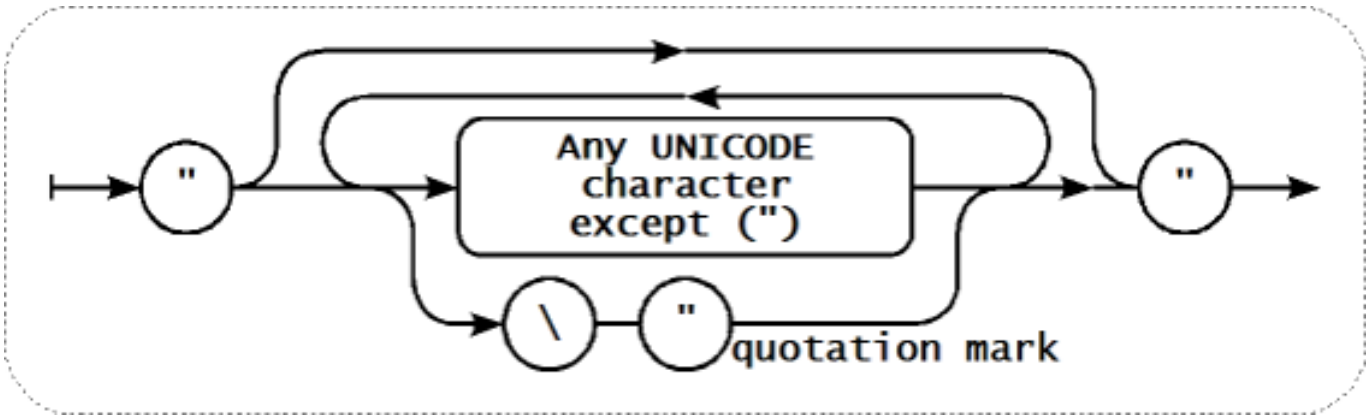


Der Parameter kann auch leer gelassen werden, sodass der Standardwert verwendet wird. In diesem Fall muss die entsprechende Anzahl von Kommas hinzugefügt werden, wenn weitere Parameter angegeben werden. Zum Beispiel würde der Datenpunktschlüssel **icmpping[„200„500]** festlegen, dass das Intervall zwischen einzelnen Pings 200 Millisekunden beträgt, das Timeout 500 Millisekunden, und alle anderen Parameter auf ihren Standardwerten belassen werden.

Es ist möglich, Makros in die Parameter einzuschließen. Dabei kann es sich um Benutzermakros oder einige der integrierten Makros handeln. Um zu sehen, welche integrierten Makros in Datenpunktschlüsselparametern unterstützt werden, suchen Sie auf der Seite Unterstützte Makros nach „item key parameters“.

Parameter – Zeichenkette in Anführungszeichen

Wenn der Schlüsselparameter eine Zeichenkette in Anführungszeichen ist, sind beliebige Unicode-Zeichen zulässig. Wenn die Zeichenkette des Schlüsselparameters ein Anführungszeichen enthält, muss dieser Parameter in Anführungszeichen gesetzt werden, und jedes Anführungszeichen, das Teil der Parameterzeichenkette ist, muss mit einem Backslash-Zeichen (\) maskiert werden. Wenn die Zeichenkette des Schlüsselparameters ein Komma enthält, muss dieser Parameter in Anführungszeichen gesetzt werden.

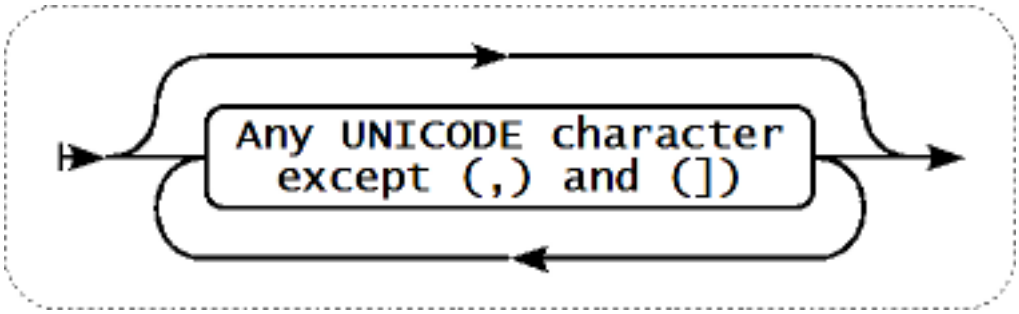


Warning:
Verwenden Sie zum Setzen von Anführungszeichen bei Datenpunkt-Schlüsselparametern ausschließlich doppelte Anführungszeichen. Einfache Anführungszeichen werden nicht unterstützt.

Attention:
Mehrstufige Parameter-Arrays, z. B. `[a,[b,[c,d]],e]`, sind nicht zulässig.

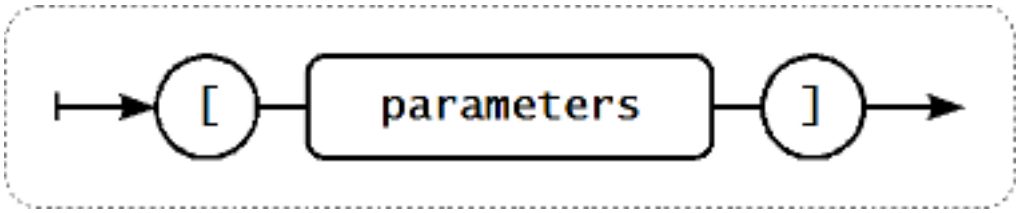
Parameter - nicht in Anführungszeichen gesetzte Zeichenkette

Wenn der Schlüsselparameter eine nicht in Anführungszeichen gesetzte Zeichenkette ist, ist jedes Unicode-Zeichen zulässig, außer Komma und schließender eckiger Klammer (]). Ein nicht in Anführungszeichen gesetzter Parameter darf nicht mit einer öffnenden eckigen Klammer ([) beginnen.



Parameter - array

Wenn der Schlüsselparameter ein Array ist, wird es erneut in eckige Klammern gesetzt, wobei die einzelnen Parameter den Regeln und der Syntax für die Angabe mehrerer Parameter entsprechen.



Benutzerdefinierte Intervalle

Übersicht

Es ist möglich, benutzerdefinierte Regeln für die Zeitpunkte zu erstellen, zu denen ein Datenpunkt überprüft wird. Dafür gibt es zwei Methoden: *Flexible Intervalle*, mit denen Sie das standardmäßige Aktualisierungsintervall neu definieren können, und *Zeitplanung*, mit der eine Datenpunktprüfung zu einem bestimmten Zeitpunkt oder einer bestimmten Abfolge von Zeitpunkten ausgeführt werden kann.

Flexible Intervalle

Flexible Intervalle ermöglichen es, das standardmäßige Aktualisierungsintervall für bestimmte Zeiträume neu zu definieren.

Ein flexibles Intervall wird mit *Interval* und *Period* definiert:

- *Interval* – das Aktualisierungsintervall für den angegebenen Zeitraum.
  Zeit-Suffixe werden unterstützt, z. B. 30s, 1m, 2h, 1d.
- *Period* – der Zeitraum, in dem das flexible Intervall aktiv ist (eine ausführliche Beschreibung des Formats von *Period* finden Sie unter Zeitzeiträume).

Wenn sich mehrere flexible Intervalle überschneiden, wird für den Überschneidungszeitraum der kleinste Wert von *Interval* verwendet.
Beachten Sie, dass keine Abfrage durchgeführt wird, wenn der kleinste Wert der sich überschneidenden flexiblen Intervalle „0“ ist.
Außerhalb der flexiblen Intervalle wird das standardmäßige Aktualisierungsintervall verwendet.

Beachten Sie, dass der Datenpunkt genau einmal geprüft wird, wenn das flexible Intervall der Länge des Zeitraums entspricht.
Wenn das flexible Intervall länger als der Zeitraum ist, wird der Datenpunkt möglicherweise einmal oder überhaupt nicht geprüft (daher wird eine solche Konfiguration nicht empfohlen).
Wenn das flexible Intervall kürzer als der Zeitraum ist, wird der Datenpunkt mindestens einmal geprüft.

Wenn das flexible Intervall auf „0“ gesetzt ist, wird der Datenpunkt während des Zeitraums des flexiblen Intervalls nicht abgefragt und setzt die Abfragen gemäß dem standardmäßigen *Update interval* fort, sobald der Zeitraum beendet ist.

Beispiele:

Interval	Period	Description
10	1-5,09:00-18:00	Der Datenpunkt wird während der Arbeitszeit alle 10 Sekunden geprüft.
0	1-7,00:00-7:00	Der Datenpunkt wird nachts nicht geprüft.
0	7-7,00:00-24:00	Der Datenpunkt wird sonntags nicht geprüft.
60	1-7,12:00-12:01	Der Datenpunkt wird jeden Tag um 12:00 Uhr geprüft. Beachten Sie, dass dies als Workaround für geplante Prüfungen verwendet wurde. Für solche Prüfungen wird empfohlen, Planungsintervalle zu verwenden.

Zeitplanintervalle

Zeitplanintervalle werden verwendet, um Datenpunkte zu bestimmten Zeiten zu prüfen. Während flexible Intervalle dazu dienen, das Standard-Aktualisierungsintervall eines Datenpunkts neu zu definieren, werden Zeitplanintervalle verwendet, um einen unabhängigen Prüfzeitplan festzulegen, der parallel ausgeführt wird.

Ein Zeitplanintervall wird wie folgt definiert: `md<filter>wd<filter>h<filter>m<filter>s<filter>`, wobei:

- **md** - Monatstage
- **wd** - Wochentage
- **h** - Stunden
- **m** - Minuten
- **s** - Sekunden

`<filter>` wird verwendet, um Werte für sein Präfix (Tage, Stunden, Minuten, Sekunden) anzugeben, und ist wie folgt definiert: `[<from>[-<to>]][/<step>][,<filter>]`, wobei:

- `<from>` und `<to>` definieren den Bereich der übereinstimmenden Werte (einschließlich). Wenn `<to>` weggelassen wird, dann entspricht der Filter einem Bereich `<from> - <from>`. Wenn `<from>` ebenfalls weggelassen wird, dann entspricht der Filter allen möglichen Werten.
- `<step>` definiert die Schrittweite der Zahlenwerte innerhalb des Bereichs. Standardmäßig hat `<step>` den Wert 1, was bedeutet, dass alle Werte des definierten Bereichs übereinstimmen.

Während die Filterdefinitionen optional sind, muss mindestens ein Filter verwendet werden. Ein Filter muss entweder einen Bereich oder den Wert <*step*> definiert haben.

Ein leerer Filter entspricht entweder ’0’, wenn kein Filter einer niedrigeren Ebene definiert ist, oder andernfalls allen möglichen Werten. Wenn zum Beispiel der Stundenfilter weggelassen wird, dann stimmt nur die Stunde ’0’ überein, vorausgesetzt, dass Minuten- und Sekundenfilter ebenfalls weggelassen werden; andernfalls entspricht ein leerer Stundenfilter allen Stundenwerten.

Gültige `<from>`- und `<to>`-Werte für das jeweilige Filterpräfix sind:

Prefix	Beschreibung	<*from*>	<*to*>
md	Monatstage	1-31	1-31
wd	Wochentage	1-7	1-7
h	Stunden	0-23	0-23
m	Minuten	0-59	0-59
s	Sekunden	0-59	0-59

Der Wert `<from>` muss kleiner oder gleich dem Wert `<to>` sein. Der Wert `<step>` muss größer oder gleich 1 und kleiner oder gleich `<to>` - `<from>` sein.

Einstellige Werte für Monatstage, Stunden, Minuten und Sekunden können mit einer führenden 0 versehen werden. Zum Beispiel sind `md01-31` und `h/02` gültige Intervalle, `md01-031` und `wd01-07` jedoch nicht.

Im Zabbix Frontend werden mehrere Zeitplanintervalle in separaten Zeilen eingegeben. In der Zabbix API werden sie zu einer einzelnen Zeichenkette zusammengefügt, wobei ein Semikolon `;` als Trennzeichen verwendet wird.

Wenn eine Zeit von mehreren Intervallen abgedeckt wird, wird sie nur einmal ausgeführt. Zum Beispiel wird `wd1h9;h9` am Montag um 9:00 Uhr nur einmal ausgeführt.

Beispiele:

Intervall	Wird ausgeführt
m0-59	jede Minute
h9-17/2	alle 2 Stunden beginnend mit 9:00 (9:00, 11:00 ...)
m0,30 or m/30	stündlich um hh:00 und hh:30
m0,5,10,15,20,25,30,35,40,45,50,55 or m/5	alle fünf Minuten
wd1-5h9	jeden Montag bis Freitag um 9:00
wd1-5h9-18	jeden Montag bis Freitag um 9:00,10:00,...,18:00
h9,10,11 or h9-11	jeden Tag um 9:00, 10:00 und 11:00
md1h9m30	jeden 1. Tag jedes Monats um 9:30
md1wd1h9m30	jeden 1. Tag jedes Monats um 9:30, wenn es ein Montag ist
h9m/30	jeden Tag um 9:00, 9:30
h9m0-59/30	jeden Tag um 9:00, 9:30
h9,10m/30	jeden Tag um 9:00, 9:30, 10:00, 10:30
h9-10m30	jeden Tag um 9:30, 10:30
h9m10-40/30	jeden Tag um 9:10, 9:40
h9,10m10-40/30	jeden Tag um 9:10, 9:40, 10:10, 10:40
h9-10m10-40/30	jeden Tag um 9:10, 9:40, 10:10, 10:40
h9m10-40	jeden Tag um 9:10, 9:11, 9:12, ... 9:40
h9m10-40/1	jeden Tag um 9:10, 9:11, 9:12, ... 9:40
h9-12,15	jeden Tag um 9:00, 10:00, 11:00, 12:00, 15:00
h9-12,15m0	jeden Tag um 9:00, 10:00, 11:00, 12:00, 15:00
h9-12,15m0s30	jeden Tag um 9:00:30, 10:00:30, 11:00:30, 12:00:30, 15:00:30
h9-12s30	jeden Tag um 9:00:30, 9:01:30, 9:02:30 ... 12:58:30, 12:59:30
h9m/30;h10 (*API-specific syntax*)	jeden Tag um 9:00, 9:30, 10:00
h9m/30 h10 (*add this as another row in frontend*)	jeden Tag um 9:00, 9:30, 10:00

Zeitzonen für Proxys und Agent angleichen

Beachten Sie, dass Zabbix-Proxys und Agent bei der Verarbeitung von Planungsintervallen ihre lokalen Zeitzonen verwenden.

Aus diesem Grund wird empfohlen, bei Planungsintervallen, die auf Datenpunkte angewendet werden, die von einem Zabbix Proxy überwacht werden, oder auf aktive Agent-Datenpunkte, die Zeitzone der jeweiligen Proxys oder des Agent auf dieselbe wie die des Zabbix Server festzulegen. Andernfalls kann die queue Verzögerungen von Datenpunkten falsch melden.

Die Zeitzone für Zabbix Proxy oder Agent kann über die Umgebungsvariable `TZ` in der `systemd`-Unit-Datei festgelegt werden:

```
[Service]
...
Environment="TZ=Europe/Amsterdam"
```

Datenpunkt-Wert-Vorverarbeitung

Übersicht

Die Vorverarbeitung ermöglicht es Ihnen, Transformationen auf die empfangenen Datenpunktwerte anzuwenden, bevor sie in der Datenbank gespeichert werden. Diese Transformationen/Vorverarbeitungsschritte werden vom Zabbix Server oder Proxy ausgeführt (wenn Datenpunkte über einen Proxy überwacht werden).

Diese Funktion unterstützt eine Vielzahl von Anwendungsfällen, zum Beispiel:

- Umwandlung von Byte in Bit (z. B. durch Multiplikation von Netzwerkverkehrswerten mit „8“);
- Berechnung von Statistiken pro Sekunde für inkrementell ansteigende Werte;
- Anwendung regulärer Ausdrücke zum Extrahieren oder Ändern von Werten;
- Ausführung benutzerdefinierter Skripte auf Werten;
- Verwerfen unveränderter Werte zur Optimierung der Datenbankspeicherung.

Für einen Datenpunkt können ein oder mehrere Vorverarbeitungsschritte konfiguriert werden. Diese Schritte werden in der Reihenfolge ausgeführt, in der sie konfiguriert sind.

Attention:
Wenn ein Vorverarbeitungsschritt fehlschlägt, wird ein Datenpunkt nicht unterstützt. Dies kann durch die Fehlerbehandlung *Benutzerdefiniert bei Fehler* vermieden werden (für die meisten Transformationen verfügbar), mit der Sie Werte verwerfen oder benutzerdefinierte Werte festlegen können.

 Bei Log-Datenpunkten setzen Log-Metadaten (ohne Wert) den Status „nicht unterstützt“ des Datenpunkts immer zurück, sodass er wieder unterstützt wird. Dies geschieht auch dann, wenn der ursprüngliche Fehler nach dem Empfang eines Log-Werts vom Agent aufgetreten ist.

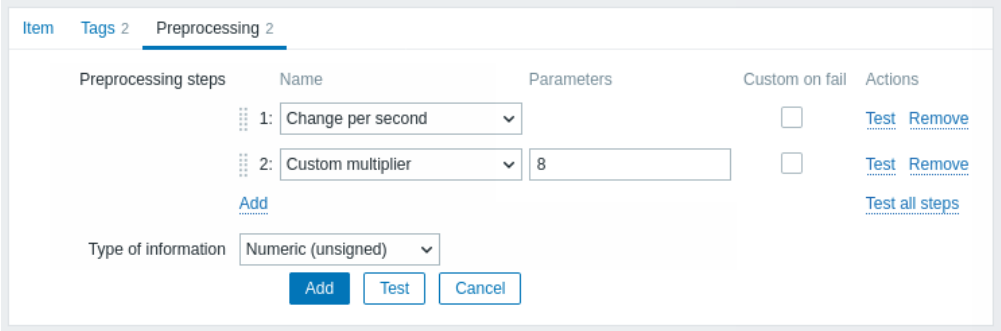
Alle an die Vorverarbeitung übergebenen Werte werden zunächst als Zeichenfolgen behandelt. Die Konvertierung in den gewünschten Werttyp (wie in der Datenpunktkonfiguration definiert) erfolgt am Ende der Vorverarbeitungspipeline. Bestimmte Vorverarbeitungsschritte können jedoch bei Bedarf frühere Konvertierungen auslösen. Ausführliche technische Informationen finden Sie unter Details zur Vorverarbeitung.

Um sicherzustellen, dass Ihre Vorverarbeitungskonfiguration wie erwartet funktioniert, können Sie sie testen.

Siehe auch: Beispiele zur Vorverarbeitung

Konfiguration

Vorverarbeitungsschritte werden auf der Registerkarte **Preprocessing** des Datenpunkt-Konfigurationsformulars definiert.



Klicken Sie auf *Add*, um eine unterstützte Transformation auszuwählen.

Das Feld *Type of information* wird am unteren Rand der Registerkarte angezeigt, sobald mindestens ein Vorverarbeitungsschritt definiert ist. Bei Bedarf kann der Informationstyp geändert werden, ohne die Registerkarte *Preprocessing* zu verlassen. Eine detaillierte Beschreibung der Parameter finden Sie unter Erstellen eines Datenpunkts.

Unterstützte Transformationen

Alle unterstützten Transformationen sind unten aufgeführt.
Klicken Sie auf den Namen einer Transformation, um vollständige Details dazu anzuzeigen.

Name	Beschreibung	Typ
Regulärer Ausdruck	Den Wert mit dem regulären Ausdruck abgleichen und durch die erforderliche Ausgabe ersetzen.	Text
Ersetzen	Die Suchzeichenfolge finden und durch eine andere Zeichenfolge (oder nichts) ersetzen.	
Trimmen	Angegebene Zeichen am Anfang und Ende des Werts entfernen.	
Rechts trimmen	Angegebene Zeichen am Ende des Werts entfernen.	
Links trimmen	Angegebene Zeichen am Anfang des Werts entfernen.	
XML XPath	Einen Wert oder ein Fragment aus XML-Daten mithilfe der XPath-Funktionalität extrahieren.	Strukturierte Daten
JSON Path	Einen Wert oder ein Fragment aus JSON-Daten mithilfe der JSONPath-Funktionalität extrahieren.	
CSV in JSON	Dateidaten im CSV-Format in das JSON-Format konvertieren.	
XML in JSON	Daten im XML-Format in JSON konvertieren.	
SNMP-Walk-Wert	Einen Wert anhand des angegebenen OID-/MIB-Namens extrahieren und Formatierungsoptionen anwenden.	SNMP
SNMP-Walk in JSON	SNMP-Werte in JSON konvertieren.	
SNMP-Get-Wert	Formatierungsoptionen auf den SNMP-Get-Wert anwenden.	
Benutzerdefinierter Multiplikator	Den Wert mit einem angegebenen ganzzahligen Wert oder Gleitkommawert multiplizieren.	Arithmetik
Einfache Änderung	Die Differenz zwischen dem aktuellen und dem vorherigen Wert berechnen.	Änderung
Änderung pro Sekunde	Die Geschwindigkeit der Wertänderung (Differenz zwischen dem aktuellen und dem vorherigen Wert) pro Sekunde berechnen.	
Boolesch in Dezimal	Den Wert vom booleschen Format in das Dezimalformat konvertieren.	Zahlensysteme

Name	Beschreibung	Typ
Oktal in Dezimal	Den Wert vom Oktalformat in das Dezimalformat konvertieren.	
Hexadezimal in Dezimal	Den Wert vom Hexadezimalformat in das Dezimalformat konvertieren.	
JavaScript	JavaScript-Code eingeben.	Benutzerdefinierte Skripte
Im Bereich	Einen Bereich definieren, in dem sich ein Wert befinden muss.	Validierung
Entspricht einem regulären Ausdruck	Einen regulären Ausdruck angeben, dem ein Wert entsprechen muss.	
Entspricht keinem regulären Ausdruck	Einen regulären Ausdruck angeben, dem ein Wert nicht entsprechen darf.	
Auf Fehler in JSON prüfen	Auf eine auf Anwendungsebene befindliche Fehlermeldung am JSONPath prüfen.	
Auf Fehler in XML prüfen	Auf eine auf Anwendungsebene befindliche Fehlermeldung am XPath prüfen.	
Mithilfe eines regulären Ausdrucks auf Fehler prüfen	Mithilfe eines regulären Ausdrucks auf eine auf Anwendungsebene befindliche Fehlermeldung prüfen.	
Auf nicht unterstützten Wert prüfen	Prüfen, ob kein Datenpunktwert abgerufen werden konnte.	
Unveränderte Werte verwerfen	Einen Wert verwerfen, wenn er sich nicht geändert hat.	Drosselung
Unveränderte Werte mit Heartbeat verwerfen	Einen Wert verwerfen, wenn er sich innerhalb des definierten Zeitraums nicht geändert hat.	
Prometheus-Muster	Die folgende Abfrage verwenden, um die erforderlichen Daten aus Prometheus-Metriken zu extrahieren.	Prometheus
Prometheus in JSON	Die erforderlichen Prometheus-Metriken in JSON konvertieren.	

Wenn das Kontrollkästchen *Benutzerdefiniert bei Fehler* für die ausgewählte Transformation aktiviert ist, können Sie es markieren und benutzerdefinierte Optionen zur Fehlerbehandlung angeben: *Wert verwerfen* (den Wert verwerfen), *Wert setzen auf* (einen angegebenen Wert festlegen) oder *Fehler setzen auf* (eine angegebene Fehlermeldung festlegen).
Wenn ein Vorverarbeitungsschritt fehlschlägt, wird der Datenpunkt nicht als nicht unterstützt markiert, wenn entweder die Option *Wert verwerfen* oder *Wert setzen auf* ausgewählt ist.

Beachten Sie, dass Zabbix für Vorverarbeitungsschritte vom Typ *Änderung* und *Drosselung* den letzten Wert speichern muss, um den neuen Wert wie erforderlich zu berechnen bzw. zu vergleichen.
Diese vorherigen Werte werden vom Vorverarbeitungsmanager verarbeitet.
Wenn der Zabbix-Server oder der Proxy neu gestartet wird oder Änderungen an den Vorverarbeitungsschritten vorgenommen werden, wird der letzte Wert des entsprechenden Datenpunkts zurückgesetzt. Dies führt zu Folgendem:

- Bei den Schritten *Einfache Änderung* und *Änderung pro Sekunde* wird der nächste Wert ignoriert, da kein vorheriger Wert vorhanden ist, anhand dessen die Änderung berechnet werden kann.
- Bei den Schritten *Unveränderte Werte verwerfen* und *Unveränderte Werte mit Heartbeat verwerfen* wird der nächste Wert niemals verworfen, selbst wenn er gemäß den Verwerfungsregeln verworfen werden sollte.

Regulärer Ausdruck

Den Wert mit dem regulären Ausdruck abgleichen und durch die erforderliche Ausgabe ersetzen.

Parameter:

- **pattern** – der reguläre Ausdruck;
- **output** – Vorlage für die Ausgabeformatierung. Eine Escape-Sequenz \N (wobei N=1…9) wird durch die N-te übereinstimmende Gruppe ersetzt. Eine Escape-Sequenz \0 wird durch den übereinstimmenden Text ersetzt.

Kommentare:

- Wenn für den Eingabewert keine Übereinstimmung gefunden wird, wird der Datenpunkt nicht unterstützt;
- Der reguläre Ausdruck unterstützt die Extraktion von maximal 10 erfassten Gruppen mit der Sequenz \N;
- Weitere vorhandene Beispiele finden Sie im Abschnitt Reguläre Ausdrücke.

Ersetzen

Sucht nach der Suchzeichenfolge und ersetzt sie durch eine andere Zeichenfolge (oder durch nichts).

Parameter:

- **Suchzeichenfolge** – die zu suchende und zu ersetzende Zeichenfolge, bei der die Groß-/Kleinschreibung berücksichtigt wird (erforderlich);
- **Ersetzung** – die Zeichenfolge, durch die die Suchzeichenfolge ersetzt werden soll. Die Ersetzungszeichenfolge kann auch leer sein, sodass die Suchzeichenfolge bei ihrem Auffinden gelöscht werden kann.

Kommentare:

- Alle Vorkommen der Suchzeichenfolge werden ersetzt;
- Es ist möglich, Escape-Sequenzen zu verwenden, um nach Zeilenumbrüchen, Wagenrückläufen, Tabulatoren und Leerzeichen zu suchen oder diese zu ersetzen: "\n \r \t \s"; der umgekehrte Schrägstrich kann als "\\" maskiert werden, und Escape-Sequenzen können als "\\n" maskiert werden;
- Das Maskieren von Zeilenumbrüchen, Wagenrückläufen und Tabulatoren wird während der Low-Level-Erkennung automatisch durchgeführt.

Trim

Entfernt angegebene Zeichen vom Anfang und Ende des Werts.

Rechts trimmen

Entfernt die angegebenen Zeichen vom Ende des Werts.

Linkes Trimmen

Entfernt angegebene Zeichen vom Anfang des Werts.

XML XPath

Extrahiert einen Wert oder ein Fragment aus XML-Daten mithilfe der XPath-Funktionalität.

Kommentare:

- Damit diese Option funktioniert, muss der Zabbix Server (oder Zabbix Proxy) mit Unterstützung für libxml kompiliert worden sein;
- Namespaces werden nicht unterstützt.

Beispiele:

```
number(/document/item/value) #extrahiert „10“ aus <document><item><value>10</value></item></document>
number(/document/item/@attribute) #extrahiert „10“ aus <document><item attribute="10"></item></document>
/document/item #extrahiert „<item><value>10</value></item>“ aus <document><item><value>10</value></item></
```

JSON-Pfad

Extrahieren Sie einen Wert oder ein Fragment aus JSON-Daten mithilfe der JSONPath-Funktionalität.

CSV in JSON

Konvertiert Daten aus CSV-Dateien in das JSON-Format.

Weitere Informationen finden Sie unter: CSV-in-JSON-Präprozessierung.

XML in JSON

Konvertieren Sie Daten im XML-Format in JSON.

Weitere Informationen finden Sie unter: Serialisierungsregeln.

SNMP-Walk-Wert

Extrahiert den Wert anhand der angegebenen OID-/MIB-Bezeichnung und wendet Formatierungsoptionen an:

- **Unverändert** – gibt die Hex-Zeichenfolge als nicht maskierte Hex-Zeichenfolge zurück (*beachten Sie*, dass Anzeigehinweise weiterhin angewendet werden);
- **UTF-8 aus HEX-STRING** – konvertiert die Hex-Zeichenfolge in eine UTF-8-Zeichenfolge;
- **MAC aus HEX-STRING** – validiert die Hex-Zeichenfolge als MAC-Adresse und gibt eine gültige MAC-Adresszeichenfolge zurück (wobei `' '` durch `':'` ersetzt wird);
- **Ganzzahl aus BITS** – konvertiert die ersten 8 Bytes einer Bit-Zeichenfolge, die als Folge von Hexadezimalzeichen dargestellt wird (z. B. „1A 2B 3C 4D“), in eine vorzeichenlose 64-Bit-Ganzzahl. Bei Bit-Zeichenfolgen mit mehr als 8 Bytes werden die nachfolgenden Bytes ignoriert.

SNMP walk zu JSON

Konvertiert SNMP-Werte in JSON.

Geben Sie einen Feldnamen im JSON und den entsprechenden SNMP-OID-Pfad an.
Die Feldwerte werden mit den Werten aus dem angegebenen SNMP-OID-Pfad befüllt.

Kommentare:

- Es stehen ähnliche Optionen zur Werteformatierung wie im Schritt *SNMP walk value* zur Verfügung;
- Sie können diesen Vorverarbeitungsschritt für die SNMP-OID-Ermittlung verwenden.

SNMP-Get-Wert

Wenden Sie Formatierungsoptionen auf den SNMP-Get-Wert an:

- **UTF-8 aus Hex-STRING** – konvertiert eine Hex-Zeichenfolge in eine UTF-8-Zeichenfolge;
- **MAC aus Hex-STRING** – validiert die Hex-Zeichenfolge als MAC-Adresse und gibt eine gültige MAC-Adresszeichenfolge zurück (wobei `' '` durch `':'` ersetzt wird);
- **Ganzzahl aus BITS** – konvertiert die ersten 8 Bytes einer Bitzeichenfolge, die als Folge von Hexadezimalzeichen dargestellt wird (z. B. „1A 2B 3C 4D“), in eine vorzeichenlose 64-Bit-Ganzzahl. Bei Bitzeichenfolgen mit mehr als 8 Bytes werden die nachfolgenden Bytes ignoriert.

Benutzerdefinierter Multiplikator

Multipliziert den Wert mit der angegebenen Ganzzahl oder Gleitkommazahl.

Kommentare:

- Verwenden Sie diese Option, um Werte, die in KB, MBps usw. empfangen werden, in B, Bps umzuwandeln. Andernfalls kann Zabbix Präfixe (K, M, G usw.) nicht korrekt festlegen.
- Beachten Sie, dass bei einem Datentyp *Numerisch (vorzeichenlos)* eingehende Werte mit einem Nachkommateil (z. B. „0.9“) vor der Anwendung des benutzerdefinierten Multiplikators gekürzt werden (d. h. „0.9“ wird zu „0“);
- Wenn Sie einen benutzerdefinierten Multiplikator verwenden oder den Wert für Datenpunkte mit dem Datentyp *Numerisch (vorzeichenlos)* als *Änderung pro Sekunde* speichern und der daraus resultierende berechnete Wert tatsächlich eine Gleitkommazahl ist, wird der berechnete Wert dennoch als korrekt akzeptiert, indem der Dezimalteil abgeschnitten und der Wert als Ganzzahl gespeichert wird;
- Unterstützt werden wissenschaftliche Schreibweisen, z. B. 1e+70, Benutzermakros und LLD-Makros sowie Zeichenfolgen, die Makros enthalten, z. B. `{#MACRO}e+10`, `{$MACRO1}e+{$MACRO2}`. Die Makros müssen zu einer Ganzzahl oder Gleitkommazahl aufgelöst werden.

Einfache Änderung

Berechnet die Differenz zwischen dem aktuellen und dem vorherigen Wert.

Kommentare:

- Dieser Schritt kann nützlich sein, um einen kontinuierlich wachsenden Wert zu messen;
- Wird als **value**-**prev_value** ausgewertet, wobei *value* der aktuelle Wert und *prev_value* der zuvor empfangene Wert ist;
- Pro Datenpunkt ist nur eine Änderungsoperation („Einfache Änderung“ oder „Änderung pro Sekunde“) zulässig;
- Wenn der aktuelle Wert kleiner als der vorherige Wert ist, verwirft Zabbix diese Differenz (speichert nichts) und wartet auf einen weiteren Wert.

Änderung pro Sekunde

Berechnet die Geschwindigkeit der Wertänderung (Differenz zwischen dem aktuellen und dem vorherigen Wert) pro Sekunde.

Kommentare:

- Dieser Schritt ist nützlich, um die Geschwindigkeit pro Sekunde eines kontinuierlich wachsenden Werts zu berechnen;
- Da diese Berechnung Gleitkommazahlen erzeugen kann, wird empfohlen, den „Informationstyp“ auf *Numerisch (Gleitkommazahl)* zu setzen, selbst wenn die eingehenden Rohwerte Ganzzahlen sind. Dies ist besonders bei kleinen Zahlen relevant, bei denen der Dezimalteil eine Rolle spielt. Wenn die Gleitkommawerte groß sind und die Feldlänge von „float“ überschreiten können, wodurch der gesamte Wert verloren gehen kann, wird tatsächlich empfohlen, *Numerisch (vorzeichenlos)* zu verwenden und dadurch nur den Dezimalteil abzuschneiden.
- Wird als (**value**-**prev_value**)/(**time**-**prev_time**) ausgewertet, wobei *value* - der aktuelle Wert; *prev_value* - der zuvor empfangene Wert; *time* - der aktuelle Zeitstempel; *prev_time* - der Zeitstempel des vorherigen Werts;
- Pro Datenpunkt ist nur eine Änderungsoperation („Einfache Änderung“ oder „Änderung pro Sekunde“) zulässig;
- Wenn der aktuelle Wert kleiner als der vorherige Wert ist, verwirft Zabbix diese Differenz (speichert nichts) und wartet auf einen weiteren Wert. Dies hilft beispielsweise beim korrekten Umgang mit einem Überlauf von 32-Bit-SNMP-Zählern.

Boolean in Dezimalzahl umwandeln

Wandelt den Wert vom booleschen Format in eine Dezimalzahl um.

Kommentare:

- Die textuelle Darstellung wird entweder in 0 oder 1 umgewandelt. Daher wird „TRUE“ als 1 und „FALSE“ als 0 gespeichert. Alle Werte werden ohne Berücksichtigung der Groß-/Kleinschreibung abgeglichen. Derzeit werden für *TRUE* folgende Werte

erkannt: true, t, yes, y, on, up, running, enabled, available, ok, master; für *FALSE*: false, f, no, n, off, down, unused, disabled, unavailable, err, slave. Außerdem wird jeder numerische Wert ungleich null als TRUE und null als FALSE betrachtet.

Oktal in Dezimal

Konvertiert den Wert vom Oktalformat in das Dezimalformat.

Hexadezimal in Dezimal

Konvertiert den Wert vom Hexadezimalformat in eine Dezimalzahl.

JavaScript

Geben Sie JavaScript-Code im modalen Editor ein, der sich öffnet, wenn Sie in das Parameterfeld oder auf das Stiftsymbol daneben klicken.

> **Attention:**
> Verwenden Sie in der Vorverarbeitungs-JavaScript keine nicht deklarierten Zuweisungen. Verwenden Sie `var`, um lokale Variablen zu deklarieren.

Kommentare:

- Die verfügbare JavaScript-Länge hängt von der verwendeten Datenbank ab;
- Weitere Informationen finden Sie unter: Javascript-Vorverarbeitung.

Im Bereich

Definieren Sie einen Bereich, in dem ein Wert liegen soll, indem Sie Minimal-/Maximalwerte (einschließlich der Grenzwerte) angeben.

Kommentare:

- Numerische Werte werden akzeptiert (einschließlich einer beliebigen Anzahl von Ziffern, eines optionalen Dezimalteils und eines optionalen Exponentialteils sowie negativer Werte);
- Der Minimalwert muss kleiner als der Maximalwert sein;
- Mindestens ein Wert muss vorhanden sein;
- Benutzermakros und Makros der Low-Level-Discovery können verwendet werden.

Entspricht dem regulären Ausdruck

Geben Sie einen regulären Ausdruck an, dem ein Wert entsprechen muss.

Entspricht nicht dem regulären Ausdruck

Geben Sie einen regulären Ausdruck an, mit dem ein Wert nicht übereinstimmen darf.

Auf einen Fehler in JSON prüfen

Prüft auf eine Fehlermeldung auf Anwendungsebene, die sich am JSONPath befindet.
Beendet die Verarbeitung, wenn die Prüfung erfolgreich war und die Meldung nicht leer ist; andernfalls wird die Verarbeitung mit dem Wert fortgesetzt, der vor diesem Vorverarbeitungsschritt vorhanden war.

Kommentare:

- Diese Fehler externer Dienste werden dem Benutzer unverändert gemeldet, ohne Informationen zum Vorverarbeitungsschritt hinzuzufügen;
- Falls ungültiges JSON nicht analysiert werden kann, wird kein Fehler gemeldet.

Auf Fehler in XML prüfen

Prüft auf eine Fehlermeldung auf Anwendungsebene, die sich am XPath befindet.
Beendet die Verarbeitung, wenn die Prüfung erfolgreich ist und die Meldung nicht leer ist; andernfalls wird die Verarbeitung mit dem Wert fortgesetzt, der vor diesem Vorverarbeitungsschritt vorhanden war.

Kommentare:

- Diese Fehler externer Dienste werden dem Benutzer unverändert gemeldet, ohne Informationen zum Vorverarbeitungsschritt hinzuzufügen;
- Beim Fehlschlagen der Analyse von ungültigem XML wird kein Fehler gemeldet.

Auf Fehler mithilfe eines regulären Ausdrucks prüfen

Mithilfe eines regulären Ausdrucks auf eine Fehlermeldung auf Anwendungsebene prüfen.
Die Verarbeitung beenden, wenn die Prüfung erfolgreich ist und die Meldung nicht leer ist; andernfalls die Verarbeitung mit dem Wert fortsetzen, der vor diesem Vorverarbeitungsschritt vorhanden war.

Parameter:

- **pattern** - der reguläre Ausdruck;
- **output** - die Vorlage für die Ausgabeformatierung.
  Eine Escape-Sequenz \N (wobei N=1…9) wird durch die N-te übereinstimmende Gruppe ersetzt.
  Eine Escape-Sequenz \0 wird durch den übereinstimmenden Text ersetzt.

Kommentare:

- Diese Fehler externer Dienste werden dem Benutzer unverändert gemeldet, ohne Informationen zum Vorverarbeitungsschritt hinzuzufügen.

Auf nicht unterstützten Wert prüfen

Prüfen, ob kein Datenpunktwert abgerufen werden konnte.
Geben Sie an, wie der Fehler verarbeitet werden soll, basierend auf der Prüfung der zurückgegebenen Fehlermeldung.

Parameter:

- **scope** – Wählen Sie den Fehlerverarbeitungsbereich aus:
any error – jeder Fehler;
error matches – nur der Fehler, der dem unter *pattern* angegebenen regulären Ausdruck entspricht;
error does not match – nur der Fehler, der nicht dem unter *pattern* angegebenen regulären Ausdruck entspricht
- **pattern** – der reguläre Ausdruck, mit dem der Fehler abgeglichen werden soll.
  Wenn im Parameter *scope any error* ausgewählt ist, wird dieses Feld nicht angezeigt.
  Wenn dieses Feld angezeigt wird, ist es obligatorisch.

Kommentare:

- Normalerweise würde das Fehlen eines Werts bzw. das Fehlschlagen seines Abrufs dazu führen, dass der Datenpunkt nicht unterstützt wird.
  Dieser Vorverarbeitungsschritt ermöglicht es Ihnen, dieses Verhalten zu ändern.
- Dieser Vorverarbeitungsschritt prüft nur, ob kein Datenpunktwert abgerufen werden konnte.
  Er prüft beispielsweise nicht, ob der Typ des abgerufenen Werts (z. B. eine Zeichenkette) mit dem Informationstyp des Datenpunkts (z. B. numerisch) übereinstimmt. Weitere Informationen finden Sie unter Beispiele für die Vorverarbeitung.
  Bei einer Typabweichung kann der Datenpunkt weiterhin nicht unterstützt werden, nachdem alle Vorverarbeitungsschritte ausgeführt wurden.
  Um eine Typabweichung zu prüfen, können Sie beispielsweise den Vorverarbeitungsschritt *Custom multiplier* verwenden. Siehe Beispiele für die Vorverarbeitung.
- Das Erfassen von Gruppen regulärer Ausdrücke wird in den Feldern *Set value to* oder *Set error to* unterstützt.
  Verwenden Sie \N (wobei N=1…9), um die N-te übereinstimmende Gruppe abzurufen; verwenden Sie \0, um den übereinstimmenden Text abzurufen;
- Diese Schritte werden immer als erste Vorverarbeitungsschritte ausgeführt und nach dem Speichern der Änderungen am Datenpunkt oberhalb aller anderen Schritte platziert;
- Mehrere Schritte vom Typ *Check for not supported value* werden in der angegebenen Reihenfolge unterstützt.
  Ein Schritt für *any error* wird automatisch als letzter Schritt in dieser Gruppe platziert.

Unveränderte Werte verwerfen

Verwirft einen Wert, wenn er sich nicht geändert hat.

Kommentare:

- Wenn ein Wert verworfen wird, wird er nicht in der Datenbank gespeichert und der Zabbix Server hat keine Kenntnis davon, dass dieser Wert empfangen wurde. Es werden keine Auslöser-Ausdrücke ausgewertet; infolgedessen werden für zugehörige Auslöser keine Probleme erstellt/gelöst. Funktionen arbeiten nur auf Basis der Daten, die tatsächlich in der Datenbank gespeichert sind. Da Trends auf Basis der Daten in der Datenbank erstellt werden, gibt es für eine Stunde auch keine Trenddaten, wenn für diese Stunde kein Wert gespeichert wurde.
- Pro Datenpunkt kann nur eine Drosselungsoption angegeben werden.

Unveränderte Werte mit Heartbeat verwerfen

Einen Wert verwerfen, wenn er innerhalb des definierten Zeitraums (in Sekunden) nicht geändert wurde.

Kommentare:

- Zur Angabe der Sekunden werden positive Ganzzahlen unterstützt (Minimum: 1 Sekunde);
- Zeitsuffixe können verwendet werden (z. B. `30s`, `1m`, `2h`, `1d`);
- Benutzermakros und Makros der Low-Level-Discovery können verwendet werden;
- Wenn ein Wert verworfen wird, wird er nicht in der Datenbank gespeichert, und der Zabbix Server weiß nicht, dass dieser Wert empfangen wurde. Es werden keine Ausdrücke von Auslösern ausgewertet. Daher werden für zugehörige Auslöser keine Probleme erstellt oder behoben. Funktionen arbeiten nur auf Grundlage der Daten, die tatsächlich in der Datenbank

gespeichert sind. Da Trends auf Grundlage der Daten in der Datenbank erstellt werden, gibt es für eine Stunde auch keine Trenddaten, wenn für diese Stunde kein Wert gespeichert wurde.
- Pro Datenpunkt kann nur eine Drosselungsoption angegeben werden.

Prometheus-Muster

Verwenden Sie die folgende Abfrage, um die erforderlichen Daten aus Prometheus-Metriken zu extrahieren.

Weitere Details finden Sie unter Prometheus-Prüfungen.

Prometheus nach JSON

Konvertiert die erforderlichen Prometheus-Metriken in JSON.

Weitere Details finden Sie unter Prometheus-Prüfungen.

Unterstützung von Makros

Benutzermakros und Benutzermakros mit Kontext werden unterstützt in:

- Parametern von Vorverarbeitungsschritten, einschließlich JavaScript-Code;
- benutzerdefinierten Parametern für die Fehlerbehandlung (Felder *Wert setzen auf* und *Fehler setzen auf*).

Note:
Der Makrokontext wird ignoriert, wenn ein Makro durch seinen Wert ersetzt wird. Der Makrowert wird unverändert in den Code eingefügt; es ist nicht möglich, vor dem Einfügen des Werts in den JavaScript-Code zusätzliches Escaping hinzuzufügen. Bitte beachten Sie, dass dies in einigen Fällen JavaScript-Fehler verursachen kann.

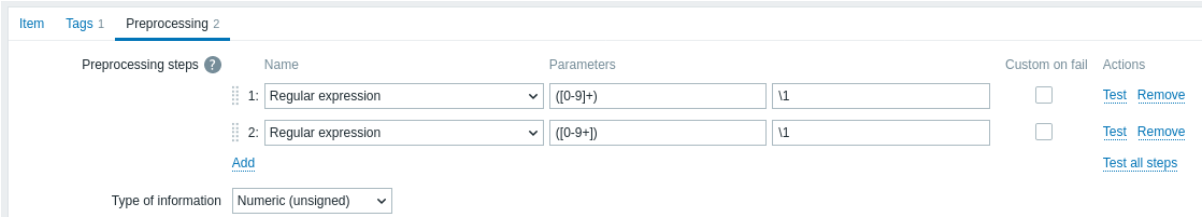
Testen

Siehe Testen der Vorverarbeitung.

Vorverarbeitungstests

Testen

Das Testen von Vorverarbeitungsschritten ist nützlich, um sicherzustellen, dass komplexe Vorverarbeitungspipelines die von ihnen erwarteten Ergebnisse liefern, ohne darauf zu warten, dass der Datenpunktwert empfangen und vorverarbeitet wird.

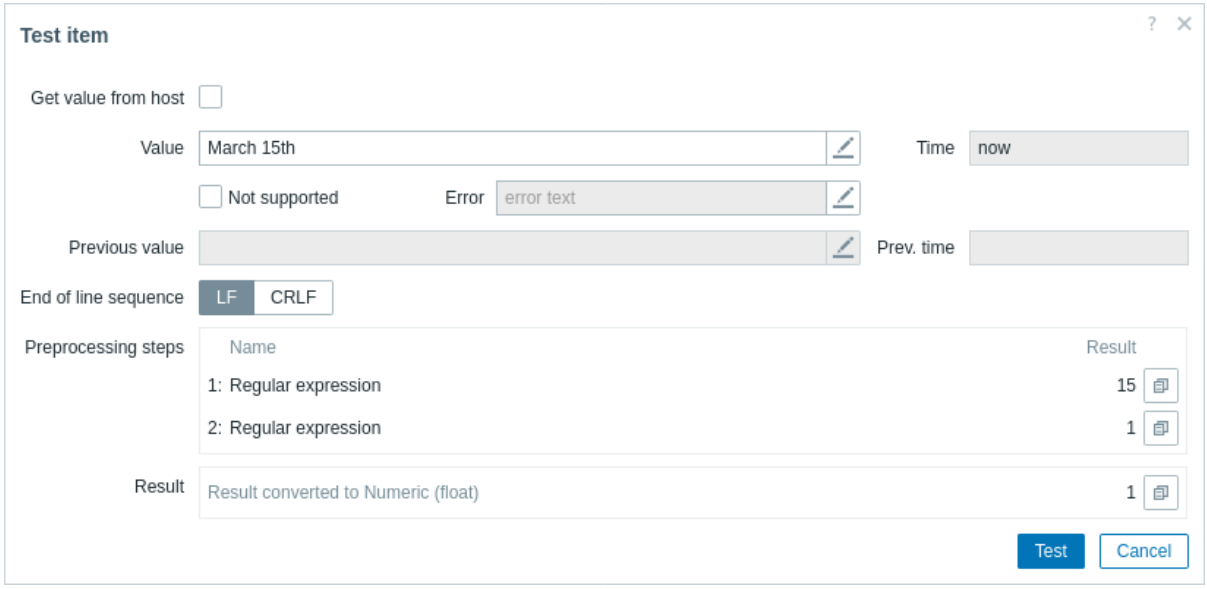


Es ist möglich zu testen:

- gegen einen hypothetischen Wert
- gegen einen realen Wert von einem Host

Jeder Vorverarbeitungsschritt kann einzeln getestet werden, ebenso können alle Schritte zusammen getestet werden. Wenn Sie im Aktionsblock auf die Schaltfläche *Test* bzw. *Alle Schritte testen* klicken, wird ein Testfenster geöffnet.

Testen eines hypothetischen Werts



Parameter	Beschreibung
Get value from host	Wenn Sie einen hypothetischen Wert testen möchten, lassen Sie dieses Kontrollkästchen deaktiviert. Siehe auch: Echten Wert testen.
Value	Geben Sie den Eingabewert zum Testen ein. Wenn Sie in das Parameterfeld oder auf die Schaltfläche zum Anzeigen/Bearbeiten klicken, wird ein Textbereich geöffnet, in den Sie den Wert oder einen Codeblock eingeben können.
Not supported	Aktivieren Sie dieses Kontrollkästchen, um einen nicht unterstützten Wert zu testen. Diese Option ist nützlich, um den Vorverarbeitungsschritt *Check for not supported value* zu testen.
Error	Geben Sie den Fehlertext ein. Dieses Feld ist aktiviert, wenn *Get value from host* deaktiviert ist, aber *Not supported* aktiviert ist. Wenn *Get value from host* aktiviert ist, wird dieses Feld mit der tatsächlichen Fehlermeldung (schreibgeschützt) vom Host gefüllt.
Time	Die Zeit des Eingabewerts wird angezeigt: `now` (schreibgeschützt).
Previous value	Geben Sie einen vorherigen Eingabewert zum Vergleichen ein. Nur für die Vorverarbeitungsschritte *Change* und *Throttling*.
Previous time	Geben Sie die Zeit des vorherigen Eingabewerts zum Vergleichen ein. Nur für die Vorverarbeitungsschritte *Change* und *Throttling*. Der Standardwert basiert auf dem Feldwert 'Update interval' des Datenpunkts (wenn '1m', dann wird dieses Feld mit `now-1m` gefüllt). Wenn nichts angegeben ist oder der Benutzer keinen Zugriff auf den Host hat, ist der Standardwert `now-30s`.
Macros	Wenn Makros verwendet werden, werden sie zusammen mit ihren Werten aufgelistet. Die Werte sind zu Testzwecken bearbeitbar, Änderungen werden jedoch nur im Testkontext gespeichert.
End of line sequence	Wählen Sie die Zeilenendesequenz für mehrzeilige Eingabewerte aus: **LF** - LF-(Line-Feed-)Sequenz **CRLF** - CRLF-(Carriage-Return-Line-Feed-)Sequenz.
Preprocessing steps	Die Vorverarbeitungsschritte werden aufgelistet; das Testergebnis wird für jeden Schritt angezeigt, nachdem die Schaltfläche *Test* geklickt wurde. Testergebnisse werden auf eine maximale Größe von 512 KB gekürzt, wenn sie an das Frontend gesendet werden. Testergebnisse können kopiert werden (jedoch nicht mehr als die gekürzten 512 KB). Wenn ein Ergebnis gekürzt wurde, wird ein Warnsymbol angezeigt. Die Warnbeschreibung wird beim Überfahren mit der Maus angezeigt. Beachten Sie, dass Daten, die größer als 512 KB sind, von Zabbix Server weiterhin vollständig verarbeitet werden. Wenn der Schritt beim Testen fehlgeschlagen ist, wird ein Fehlersymbol angezeigt. Die Fehlerbeschreibung wird beim Überfahren mit der Maus angezeigt. Falls für den Schritt "Custom on fail" angegeben ist und diese Aktion ausgeführt wird, erscheint direkt nach der Zeile des Vorverarbeitungsschritts eine neue Zeile, die zeigt, welche Aktion ausgeführt wurde und welches Ergebnis sie erzeugt hat (Fehler oder Wert).

Parameter	Beschreibung
Result	Das Endergebnis der Vorverarbeitungstests wird in allen Fällen angezeigt, in denen alle Schritte gemeinsam getestet werden (wenn Sie auf die Schaltfläche *Test all steps* klicken). Auch der Typ der Konvertierung in den Werttyp des Datenpunkts wird angezeigt, zum Beispiel `Result converted to Numeric (unsigned)`. Testergebnisse werden auf eine maximale Größe von 512 KB gekürzt, wenn sie an das Frontend gesendet werden. Das Endergebnis kann kopiert werden (jedoch nicht mehr als die gekürzten 512 KB). Wenn ein Ergebnis gekürzt wurde, wird ein Warnsymbol angezeigt. Die Warnbeschreibung wird beim Überfahren mit der Maus angezeigt. Beachten Sie, dass Daten, die größer als 512 KB sind, von Zabbix Server weiterhin vollständig verarbeitet werden.

Klicken Sie auf *Test*, um das Ergebnis nach jedem Vorverarbeitungsschritt anzuzeigen.

Testwerte werden zwischen Testsitzungen entweder für einzelne Schritte oder für alle Schritte gespeichert, sodass der Benutzer Vorverarbeitungsschritte oder die Datenpunktkonfiguration ändern und dann zum Testfenster zurückkehren kann, ohne Informationen erneut eingeben zu müssen. Die Werte gehen jedoch bei einem Seiten-Refresh verloren.

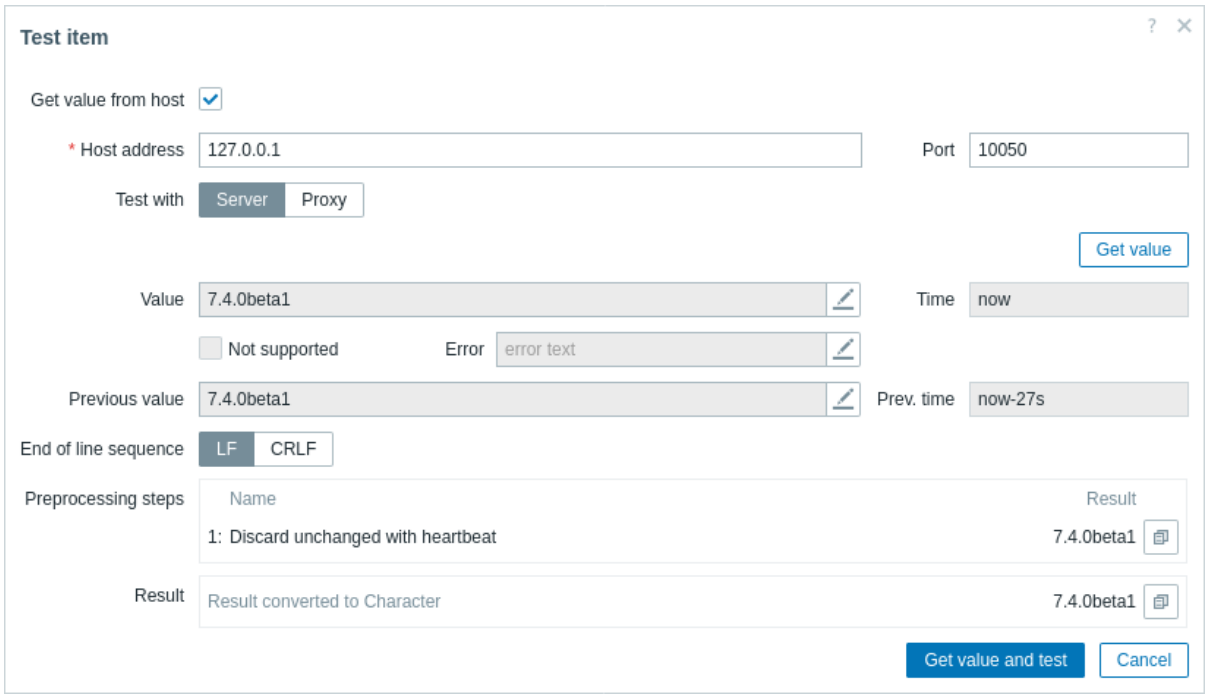
Der Test wird von Zabbix Server durchgeführt. Das Frontend sendet eine entsprechende Anfrage an den Server und wartet auf das Ergebnis. Die Anfrage enthält den Eingabewert und die Vorverarbeitungsschritte (mit erweiterten Benutzermakros). Für die Schritte *Change* und *Throttling* können ein optionaler vorheriger Wert und eine Zeit angegeben werden. Der Server antwortet mit Ergebnissen für jeden Vorverarbeitungsschritt.

Alle technischen Fehler oder Eingabevalidierungsfehler werden im Fehlerfeld oben im Testfenster angezeigt.

Testen eines realen Werts

So testen Sie die Vorverarbeitung anhand eines realen Werts:

- Aktivieren Sie das Kontrollkästchen *Wert vom Host abrufen*.
- Geben Sie die Host-Parameter (Host-Adresse, Port, Proxy-Name/kein Proxy) und die datentypspezifischen Details (z. B. SNMPv2-Community oder SNMPv3-Sicherheitsdaten) ein oder überprüfen Sie diese. Diese Felder sind kontextabhängig:
  - Die Werte werden, sofern möglich, vorausgefüllt, d. h. bei Datenpunkten, die einen Agent erfordern, werden die Informationen aus der ausgewählten Agent-Schnittstelle des Hosts übernommen.
  - Bei Datenpunkten von Vorlagen müssen die Werte manuell eingetragen werden.
  - Makrowerte im Klartext werden aufgelöst.
  - Wenn der Feldwert (oder ein Teil des Werts) ein Secret oder eine Vault-Makro ist, bleibt das Feld leer und muss manuell ausgefüllt werden. Wenn ein Datenpunktparameter einen Secret-Makrowert enthält, wird folgende Warnmeldung angezeigt: „Datenpunkt enthält benutzerdefinierte Makros mit geheimen Werten. Die Werte dieser Makros müssen manuell eingegeben werden.“
  - Die Felder sind deaktiviert, wenn sie im Kontext des Datenpunkttyps nicht benötigt werden (z. B. sind die Felder für Host-Adresse und Proxy bei berechneten Datenpunkten deaktiviert).
- Klicken Sie auf *Wert abrufen und testen*, um die Vorverarbeitung zu testen.



Wenn Sie in der Datenpunkt-Konfigurationsmaske eine Wertzuordnung angegeben haben (Feld *Wert anzeigen*), zeigt der Datenpunkttestdialog nach dem Endergebnis eine weitere Zeile mit dem Namen *Ergebnis mit angewendeter Wertzuordnung* an.

Parameter, die für das Abrufen eines realen Werts von einem Host spezifisch sind:

Parameter	Beschreibung
Wert vom Host abrufen	Aktivieren Sie dieses Kontrollkästchen, um einen realen Wert vom Host abzurufen.
Host-Adresse	Geben Sie die Host-Adresse ein. Dieses Feld wird automatisch mit der Adresse der Host-Schnittstelle des Datenpunkts ausgefüllt.
Port	Geben Sie den Host-Port ein. Dieses Feld wird automatisch mit dem Port der Host-Schnittstelle des Datenpunkts ausgefüllt.
Zusätzliche Felder für SNMP-Schnittstellen (SNMP-Version, SNMP-Community, Kontextname usw.)	Weitere Informationen zur Konfiguration einer SNMP-Schnittstelle (v1, v2 und v3) finden Sie unter Konfigurieren der SNMP-Überwachung. Diese Felder werden automatisch aus der Host-Schnittstelle des Datenpunkts ausgefüllt.
Proxy	Geben Sie den Proxy an, wenn der Host von einem Proxy überwacht wird. Dieses Feld wird automatisch mit dem Proxy des Hosts (falls vorhanden) ausgefüllt.
Wert	Vom Host abgerufener Wert. Durch Klicken in das Parameterfeld oder auf die Schaltfläche zum Anzeigen/Bearbeiten wird ein Textbereichfenster mit dem Wert oder Codeblock geöffnet. Werte werden auf eine maximale Größe von 512 KB gekürzt, und zwar nur im Frontend. Wenn ein Ergebnis gekürzt wird, wird ein Warnsymbol angezeigt. Die Warnungsbeschreibung wird beim Überfahren mit dem Mauszeiger angezeigt. Beachten Sie, dass Daten mit einer Größe von mehr als 512 KB weiterhin vollständig vom Zabbix Server verarbeitet werden.

Weitere Informationen zu den übrigen Parametern finden Sie oben unter Testen eines hypothetischen Werts.

Vorverarbeitungsdetails

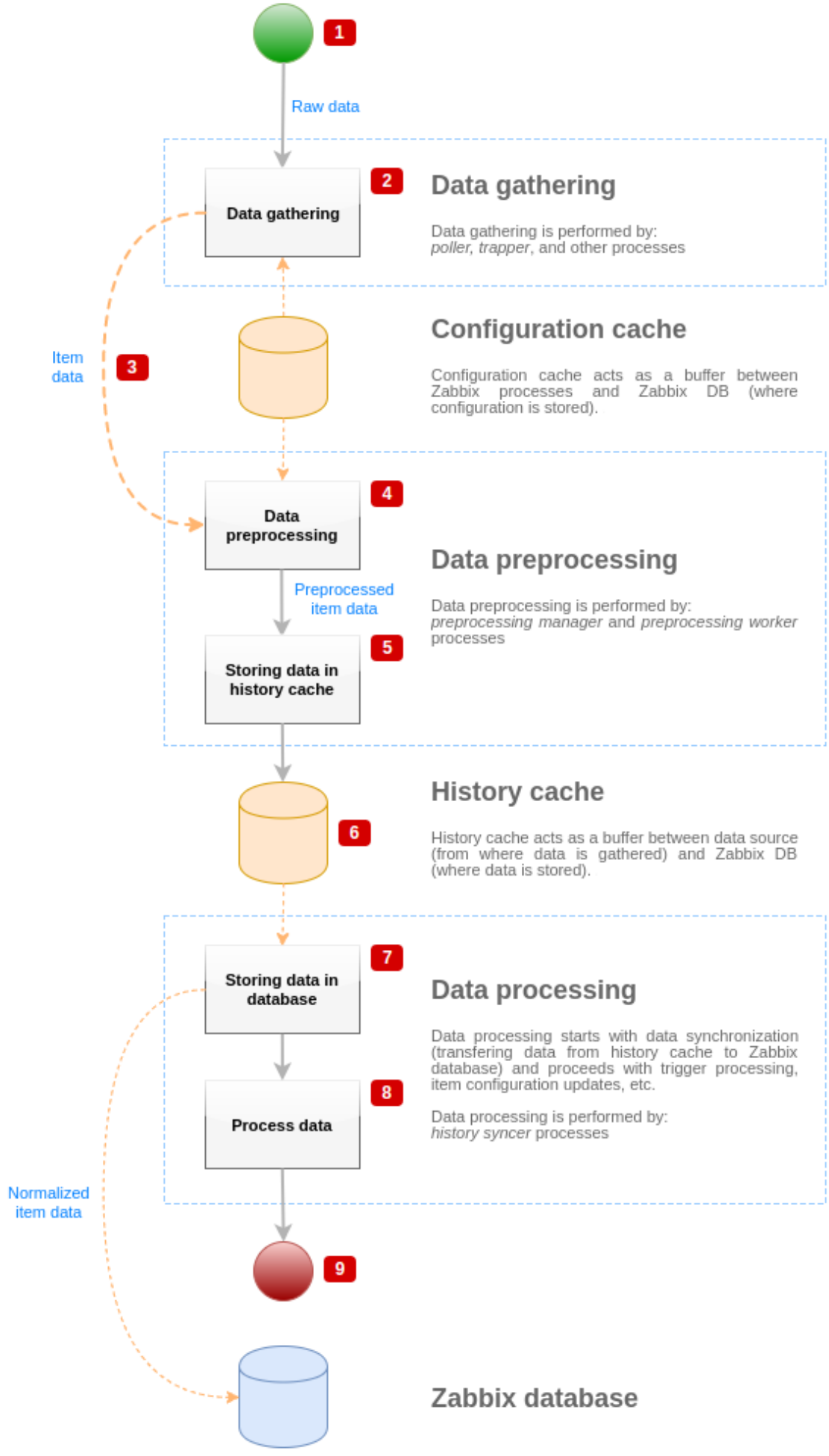
Übersicht

Dieser Abschnitt enthält Details zur Vorverarbeitung von Datenpunktwerten. Mit der Vorverarbeitung von Datenpunkten können Sie Transformationsregeln für die empfangenen Datenpunktwerte definieren und ausführen.

Die Vorverarbeitung wird vom Prozess des Vorverarbeitungsmanagers zusammen mit Vorverarbeitungs-Workern verwaltet, die die Vorverarbeitungsschritte ausführen. Alle vorverarbeiteten Werte, die von verschiedenen Datensammlern empfangen werden, durchlaufen den Vorverarbeitungsmanager, bevor sie zum Verlaufscache hinzugefügt werden. Zwischen den Datensammlern (Poller, Trapper usw.) und dem Vorverarbeitungsprozess wird eine socketbasierte IPC-Kommunikation verwendet. Entweder der Zabbix Server oder der Zabbix Proxy (für die vom Proxy überwachten Datenpunkte) führt die Vorverarbeitungsschritte aus.

Verarbeitung von Datenpunktwerten

Um den Datenfluss von der Datenquelle zur Zabbix-Datenbank zu visualisieren, können wir das folgende vereinfachte Diagramm verwenden:


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

Das obige Diagramm zeigt nur Prozesse, Objekte und Aktionen im Zusammenhang mit der Verarbeitung von Datenpunktwerten in **vereinfachter** Form. Das Diagramm zeigt weder bedingte Richtungsänderungen noch Fehlerbehandlung oder Schleifen. Der lokale Daten-Cache des Präprozessierungsmanagers wird ebenfalls nicht gezeigt, da er den Datenfluss nicht direkt beeinflusst. Ziel dieses Diagramms ist es, die an der Verarbeitung von Datenpunktwerten beteiligten Prozesse und ihre Interaktion darzustellen.

- Die Datenerfassung beginnt mit Rohdaten aus einer Datenquelle. Zu diesem Zeitpunkt enthalten die Daten nur ID, Zeitstempel und Wert (es können auch mehrere Werte sein).
- Unabhängig davon, welcher Typ von Datensammler verwendet wird, ist die Grundidee bei aktiven oder passiven Prüfungen, bei Trapper-Datenpunkten usw. dieselbe, da sich nur das Datenformat und der Kommunikationsinitiator ändern (entweder wartet der Datensammler auf eine Verbindung und Daten, oder der Datensammler initiiert die Kommunikation und fordert die Daten an). Die Rohdaten werden validiert, die Datenpunktkonfiguration wird aus dem Konfigurations-Cache abgerufen (die Daten werden mit den Konfigurationsdaten angereichert).
- Ein socket-basierter IPC-Mechanismus wird verwendet, um Daten von Datensammlern an den Präprozessierungsmanager zu übergeben. An diesem Punkt setzt der Datensammler die Datenerfassung fort, ohne auf die Antwort des Präprozessierungsmanagers zu warten.
- Die Datenpräprozessierung wird durchgeführt. Dazu gehören die Ausführung von Präprozessierungsschritten und die Verarbeitung abhängiger Datenpunkte.

Note:
Ein Datenpunkt kann während der Präprozessierung in den Zustand NOT SUPPORTED wechseln, wenn einer der Präprozessierungsschritte fehlschlägt.

- Die Verlaufsdaten aus dem lokalen Daten-Cache des Präprozessierungsmanagers werden in den Verlaufs-Cache geschrieben.
- An diesem Punkt stoppt der Datenfluss bis zur nächsten Synchronisierung des Verlaufs-Caches (wenn der History-Syncer-Prozess die Datensynchronisierung durchführt).
- Der Synchronisierungsprozess beginnt mit der Datennormalisierung, bevor die Daten in der Zabbix-Datenbank gespeichert werden. Die Datennormalisierung führt Konvertierungen in den gewünschten Datenpunkttyp durch (Typ wie in der Datenpunktkonfiguration definiert), einschließlich der Kürzung von Textdaten auf Basis der vordefinierten Größen, die für diese Typen zulässig sind (HISTORY_STR_VALUE_LEN für string, HISTORY_TEXT_VALUE_LEN für text und HISTORY_LOG_VALUE_LEN für log-Werte). Nach Abschluss der Normalisierung werden die Daten an die Zabbix-Datenbank gesendet.

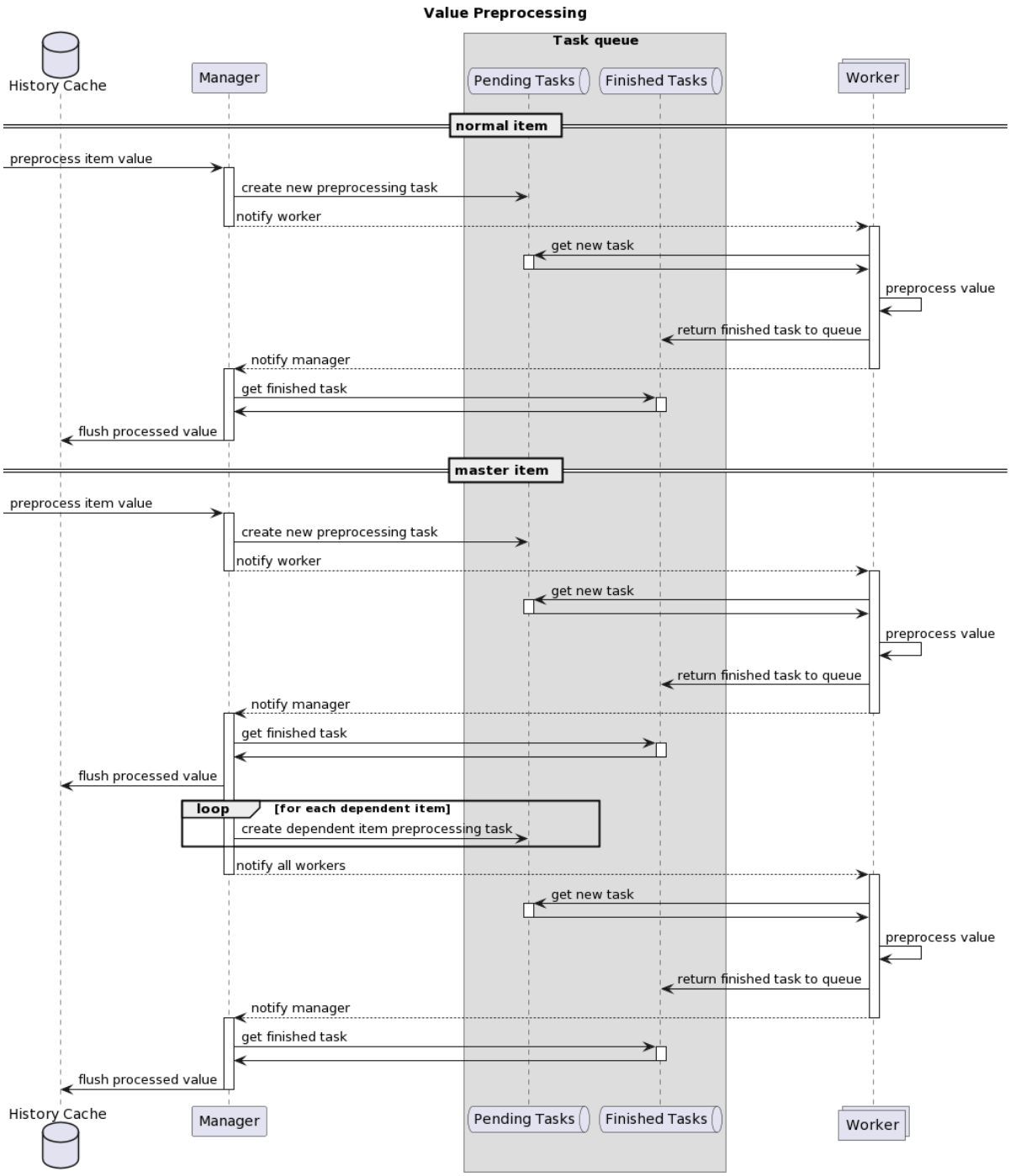
Note:
Ein Datenpunkt kann in den Zustand NOT SUPPORTED wechseln, wenn die Datennormalisierung fehlschlägt (zum Beispiel wenn ein Textwert nicht in eine Zahl konvertiert werden kann).

- Die erfassten Daten werden verarbeitet – Auslöser werden geprüft, die Datenpunktkonfiguration wird aktualisiert, wenn der Datenpunkt NOT SUPPORTED wird, usw.
- Dies gilt aus Sicht der Verarbeitung von Datenpunktwerten als Ende des Datenflusses.

Vorverarbeitung von Datenpunktwerten

Die Datenvorverarbeitung erfolgt in den folgenden Schritten:

- Wenn der Datenpunkt weder Vorverarbeitung noch abhängige Datenpunkte hat, wird sein Wert entweder zum Verlaufscache hinzugefügt oder an den LLD-Manager gesendet. Andernfalls wird der Datenpunktwert über einen UNIX-Socket-basierten IPC-Mechanismus an den Vorverarbeitungs-Manager übergeben.
- Eine Vorverarbeitungsaufgabe wird erstellt, zur Warteschlange hinzugefügt und die Vorverarbeitungs-Worker werden über die neue Aufgabe benachrichtigt.
- An diesem Punkt stoppt der Datenfluss, bis mindestens ein nicht belegter (d. h. aktuell keine Aufgaben ausführender) Vorverarbeitungs-Worker verfügbar ist.
- Wenn ein Vorverarbeitungs-Worker verfügbar ist, nimmt er die nächste Aufgabe aus der Warteschlange.
- Nach Abschluss der Vorverarbeitung (sowohl bei fehlgeschlagener als auch bei erfolgreicher Ausführung der Vorverarbeitungsschritte) wird der vorverarbeitete Wert zur Warteschlange der abgeschlossenen Aufgaben hinzugefügt und der Manager über eine neue abgeschlossene Aufgabe benachrichtigt.
- Der Vorverarbeitungs-Manager konvertiert das Ergebnis in das gewünschte Format (definiert durch den Typ des Datenpunktwerts) und fügt es entweder zum Verlaufscache hinzu oder sendet es an den LLD-Manager.
- Wenn es abhängige Datenpunkte für den verarbeiteten Datenpunkt gibt, werden die abhängigen Datenpunkte mit dem vorverarbeiteten Wert des Master-Datenpunkts zur Vorverarbeitungswarteschlange hinzugefügt. Abhängige Datenpunkte werden unter Umgehung der normalen Anfragen zur Wertvorverarbeitung in die Warteschlange eingereiht, jedoch nur für Master-Datenpunkte mit gesetztem Wert und nicht im Status NICHT UNTERSTÜTZT.



Beachten Sie, dass im Diagramm die Vorverarbeitung des Master-Datenpunkts leicht vereinfacht dargestellt ist, da das Caching der Vorverarbeitung ausgelassen wurde.

Vorverarbeitungswarteschlange

Die Vorverarbeitungswarteschlange ist wie folgt organisiert:

- die Liste der ausstehenden Aufgaben:
  - Aufgaben, die direkt aus Anfragen zur Wertvorverarbeitung in der Reihenfolge erstellt wurden, in der sie empfangen wurden
- die Liste der sofortigen Aufgaben (werden vor ausstehenden Aufgaben verarbeitet):
  - Testaufgaben (erstellt als Reaktion auf Anfragen zum Testen von Datenpunkt/Vorverarbeitung durch das Frontend)
  - Aufgaben abhängiger Datenpunkte
  - Sequenzaufgaben (Aufgaben, die in einer strikten Reihenfolge ausgeführt werden müssen):
    - mit Vorverarbeitungsschritten, die den letzten Wert verwenden:
      - Änderung
      - Drosselung
      - JavaScript (Bytecode-Caching)
    - Caching der Vorverarbeitung abhängiger Datenpunkte

- die Liste der abgeschlossenen Aufgaben

Caching für die Vorverarbeitung

Das Caching für die Vorverarbeitung wurde eingeführt, um die Leistung der Vorverarbeitung für mehrere abhängige Datenpunkte mit ähnlichen Vorverarbeitungsschritten zu verbessern (was ein häufiges Ergebnis von LLD ist).

Das Caching erfolgt, indem ein abhängiger Datenpunkt vorverarbeitet wird und einige der internen Vorverarbeitungsdaten für die übrigen abhängigen Datenpunkte wiederverwendet werden. Der Vorverarbeitungs-Cache wird nur für den ersten Vorverarbeitungsschritt der folgenden Typen unterstützt:

- Prometheus-Muster (indiziert die Eingabe nach Metriken)
- JSONPath (parst die Daten in einen Objektbaum und indiziert den ersten Ausdruck `[?(@.path == "value")]`)

Preprocessing-Worker

Die Konfigurationsdatei des Zabbix-Servers ermöglicht es Benutzern, die Anzahl der Threads für Preprocessing-Worker festzulegen. Der Konfigurationsparameter StartPreprocessors sollte verwendet werden, um die Anzahl der vorab gestarteten Instanzen von Preprocessing-Workern festzulegen; sie sollte mindestens der Anzahl der verfügbaren CPU-Kerne entsprechen.

Wenn Preprocessing-Aufgaben nicht CPU-gebunden sind und häufige Netzwerkanfragen beinhalten, wird die Konfiguration zusätzlicher Worker empfohlen. Die optimale Anzahl der Preprocessing-Worker kann von vielen Faktoren abhängen, darunter die Anzahl der ”preprocessable” Datenpunkte (Datenpunkte, die die Ausführung von Preprocessing-Schritten erfordern), die Anzahl der Datenerfassungsprozesse, die durchschnittliche Anzahl der Schritte für das Preprocessing von Datenpunkten usw. Zu wenige Worker können zu einer hohen Speicherauslastung führen. Hinweise zur Fehlerbehebung bei übermäßiger Speicherauslastung in Ihrer Zabbix-Installation finden Sie unter Profiling excessive memory usage with tcmalloc.

Wenn jedoch keine aufwendigen Preprocessing-Operationen wie das Parsen großer XML-/JSON-Blöcke anfallen, kann die Anzahl der Preprocessing-Worker der Gesamtzahl der Datenerfasser entsprechen. Auf diese Weise wird es meist (außer in Fällen, in denen Daten vom Datenerfasser in großen Mengen eintreffen) mindestens einen unbesetzten Preprocessing-Worker für erfasste Daten geben.

Warning:
Zu viele Datenerfassungsprozesse (Poller, unreachable pollers, ODBC pollers, HTTP pollers, Java pollers, pingers, trappers, proxypollers) zusammen mit dem IPMI manager, SNMP trapper und Preprocessing-Workern können das pro Prozess zulässige Limit für Dateideskriptoren des Preprocessing-Managers erschöpfen.

 Wenn das pro Prozess zulässige Limit für Dateideskriptoren erschöpft ist, wird der Zabbix-Server angehalten, typischerweise kurz nach dem Start, manchmal jedoch erst nach längerer Zeit. Um solche Probleme zu vermeiden, überprüfen Sie die Zabbix-Server-Konfigurationsdatei, um die Anzahl gleichzeitiger Prüfungen und Prozesse zu optimieren. Stellen Sie außerdem bei Bedarf sicher, dass das Limit für Dateideskriptoren ausreichend hoch gesetzt ist, indem Sie die Systemlimits prüfen und anpassen.

Pipeline zur Wertverarbeitung

Die Verarbeitung von Datenpunktwerten wird in mehreren Schritten (oder Phasen) von mehreren Prozessen ausgeführt. Dies kann dazu führen:

- Ein abhängiger Datenpunkt kann Werte empfangen, während der Master-Wert dies nicht kann. Dies lässt sich mit folgendem Anwendungsfall erreichen:
  - Der Master-Datenpunkt hat den Werttyp `UINT` (es kann ein Trapper-Datenpunkt verwendet werden), der abhängige Datenpunkt hat den Werttyp `TEXT`.
  - Für weder den Master- noch den abhängigen Datenpunkt sind Vorverarbeitungsschritte erforderlich.
  - Ein Textwert (zum Beispiel „abc“) soll an den Master-Datenpunkt übergeben werden.
  - Da keine Vorverarbeitungsschritte auszuführen sind, prüft der Vorverarbeitungsmanager, ob sich der Master-Datenpunkt nicht im Zustand NOT SUPPORTED befindet und ob ein Wert gesetzt ist (beides trifft zu), und reiht den abhängigen Datenpunkt mit demselben Wert wie den des Master-Datenpunkts in die Warteschlange ein (da es keine Vorverarbeitungsschritte gibt).
  - Wenn sowohl der Master- als auch der abhängige Datenpunkt die Phase der History-Synchronisierung erreichen, wird der Master-Datenpunkt aufgrund eines Fehlers bei der Wertkonvertierung zu NOT SUPPORTED (Textdaten können nicht in eine vorzeichenlose Ganzzahl konvertiert werden).

Als Ergebnis empfängt der abhängige Datenpunkt einen Wert, während der Master-Datenpunkt seinen Zustand in NOT SUPPORTED ändert.

- Ein abhängiger Datenpunkt empfängt einen Wert, der in der History des Master-Datenpunkts nicht vorhanden ist. Der Anwendungsfall ist dem vorherigen sehr ähnlich, mit Ausnahme des Typs des Master-Datenpunkts. Wenn zum Beispiel der Typ `CHAR` für den Master-Datenpunkt verwendet wird, dann wird der Wert des Master-Datenpunkts in der Phase der History-Synchronisierung abgeschnitten, während abhängige Datenpunkte ihre Werte aus dem ursprünglichen (nicht abgeschnittenen) Wert des Master-Datenpunkts erhalten.

Vorverarbeitungsbeispiele

Übersicht

Dieser Abschnitt enthält Beispiele für die Verwendung von Vorverarbeitungsschritten, um einige praktische Aufgaben zu erfüllen.

Filtern von VMware-Ereignisprotokolleinträgen

In diesem Beispiel wird der Vorverarbeitungsschritt Entspricht regulärem Ausdruck verwendet, um unnötige Ereignisse aus dem VMware-Ereignisprotokoll zu filtern.

1. Prüfen Sie auf einem funktionierenden VMware-Hypervisor-Host, dass der Datenpunkt vmware.eventlog vorhanden ist und ordnungsgemäß funktioniert. Beachten Sie, dass der Datenpunkt für das Ereignisprotokoll auf dem Hypervisor bereits vorhanden sein kann, wenn während der Host-Erstellung eine VMware-Vorlage verknüpft wurde.

2. Erstellen Sie auf dem VMware-Hypervisor-Host einen abhängigen Datenpunkt vom Typ *Log* und legen Sie den Datenpunkt des Ereignisprotokolls als Master-Datenpunkt fest.

3. Klicken Sie auf der Registerkarte *Vorverarbeitung* des abhängigen Datenpunkts auf *Hinzufügen*, um einen Vorverarbeitungsschritt zu erstellen, und wählen Sie *Entspricht regulärem Ausdruck* aus der Dropdown-Liste aus. Geben Sie dann eines der folgenden Muster an:

- Zum Filtern aller Protokollereignisse:

```
.* logged in .*
```

- Zum Filtern von Zeilen, die Benutzernamen nach „User“ enthalten:

```
\bUser\s+\K\S+
```

> **Attention:**
> Wenn der reguläre Ausdruck nicht übereinstimmt, wird der abhängige Datenpunkt mit einer entsprechenden Fehlermeldung nicht unterstützt. Um dies zu vermeiden, aktivieren Sie das Kontrollkästchen *Benutzerdefiniert bei Fehler* und wählen Sie eine Option wie das Verwerfen des Werts oder das Setzen eines benutzerdefinierten Werts aus. Bitte beachten Sie, dass verworfene Werte nicht in der Datenbank gespeichert werden; daher werden Auslöser nicht ausgewertet und es werden keine Trenddaten erzeugt.

Alternativ können Sie den Vorverarbeitungsschritt Regulärer Ausdruck verwenden, um übereinstimmende Gruppen zu extrahieren und die Ausgabe zu steuern:

- Um das gesamte Protokollereignis, das „logged in“ enthält, zu extrahieren und auszugeben, geben Sie die folgenden Parameter an:

```
Pattern: .*logged in.*
Output: \0
```

- Um Benutzernamen nach „User“ zu extrahieren und auszugeben:

```
Pattern: User (.*?)(?=\ )
Output: \1
```

Überprüfen des Typs des abgerufenen Werts

Dieses Beispiel verwendet den Vorverarbeitungsschritt Benutzerdefinierter Multiplikator, um zu prüfen, ob der Typ des abgerufenen Datenpunkt-Werts numerisch ist.

Wählen Sie auf der Registerkarte *Vorverarbeitung* eines Datenpunkts den Vorverarbeitungsschritt *Benutzerdefinierter Multiplikator* aus und geben Sie den folgenden Parameter an (multipliziert den abgerufenen Wert mit 1):

```
1
```

> **Attention:**
> Wenn die Vorverarbeitung fehlschlägt (z. B. wenn die Eingabe nicht numerisch ist), wird der Datenpunkt mit einer entsprechenden Fehlermeldung nicht unterstützt. Um dies zu vermeiden, aktivieren Sie das Kontrollkästchen *Benutzerdefiniert bei Fehler* und wählen Sie eine Option wie das Verwerfen des Werts oder das Setzen eines benutzerdefinierten Werts aus. Bitte beachten Sie, dass verworfene Werte nicht in der Datenbank gespeichert werden; infolgedessen werden Auslöser nicht ausgewertet und es werden keine Trenddaten erzeugt.

Prüfung auf nicht unterstützten Wert

Dieses Beispiel verwendet den Vorverarbeitungsschritt Prüfung auf nicht unterstützten Wert, um zu prüfen, ob der Datenpunkt-Wert nicht abgerufen werden konnte.

Wenn ein Poller-Prozess von Zabbix Server/Proxy versucht, einen Datenpunkt-Wert zu erfassen, kann er:

- Ein gültiges Ergebnis zurückgeben.
- Ein Ergebnis zurückgeben, das zunächst gültig erscheint, aber später nicht unterstützt werden kann (z. B. aufgrund eines Nichtübereinstimmens des Wertetyps nach der Vorverarbeitung).
- Einen Fehler bei der Erfassung des Werts zurückgeben, wodurch der Datenpunkt nicht unterstützt wird. Häufige Ursachen sind:
  - Unbekannter Datenpunkt-Schlüssel (für Zabbix Agent, einfache Prüfung oder interne Zabbix-Datenpunkte)
  - Unbekannte OID (SNMP-Agent), unbekannter Sensor (IPMI-Agent) oder keine JMX-Metrik (JMX-Agent)
  - Trap-Datei kann nicht gelesen werden (SNMP-Trap)
  - Skript nicht gefunden (externe Prüfung)
  - Keine solche URL (HTTP-Agent, Browser)
  - Anmeldung fehlgeschlagen (SSH-Agent, TELNET-Agent)
  - Ungültige Formelsyntax (berechnet), JavaScript-Syntaxfehler (Skript) oder ungültiges SQL (Datenbankmonitor)

Um Fehler bei der Erfassung von Datenpunkt-Werten zu erkennen und zu behandeln, können Sie den Vorverarbeitungsschritt *Prüfung auf nicht unterstützten Wert* verwenden. Beachten Sie, dass dieser Schritt immer zuerst ausgeführt wird und nur Fehler erkennt, die auftreten, bevor die Vorverarbeitung beginnt.

Wählen Sie auf der Registerkarte *Vorverarbeitung* eines Datenpunkts den Vorverarbeitungsschritt *Prüfung auf nicht unterstützten Wert* aus und geben Sie einen der folgenden Parameter an:

- Für beliebige Fehler:

```
Parameter: any error
```

- Für Fehler, die „cannot connect“ enthalten:

```
Parameter: error matches
Pattern: (?i)cannot connect
```

Verwenden Sie dann die Option *Benutzerdefiniert bei Fehler*, um den Wert (in diesem Fall den Fehler) zu verwerfen, einen benutzerdefinierten Wert festzulegen oder eine benutzerdefinierte Fehlermeldung zurückzugeben. Bitte beachten Sie, dass verworfene Werte nicht in der Datenbank gespeichert werden; infolgedessen werden Auslöser nicht ausgewertet und es werden keine Trenddaten erzeugt.

JSONPath-Funktionalität

Übersicht

Dieser Abschnitt beschreibt die unterstützte JSONPath-Funktionalität innerhalb der Vorverarbeitungsschritte von Datenpunkt-Werten.

JSONPath besteht aus Segmenten, die durch Punkte getrennt sind. Ein Segment kann die Form eines einfachen Wortes annehmen, das einen JSON-Wertnamen darstellt, des Platzhalterzeichens (*) oder eines komplexeren Konstrukts in eckigen Klammern. Der Punkt vor einem in Klammern gesetzten Segment ist optional und kann weggelassen werden.

JSONPath example	Description
`$.object.name`	Gibt den Inhalt von `object.name` zurück.
`$.object['name']`	Gibt den Inhalt von `object.name` zurück.
`$.object.['name']`	Gibt den Inhalt von `object.name` zurück.
`$["object"]['name']`	Gibt den Inhalt von `object.name` zurück.
`$.['object'].["name"]`	Gibt den Inhalt von `object.name` zurück.
`$.object.history.length()`	Gibt die Anzahl der Array-Elemente in `object.history` zurück.
`$[?(@.name == 'Object')].price.first()`	Gibt den Wert der Eigenschaft `price` aus dem ersten Objekt mit dem Namen „Object“ zurück.
`$[?(@.name == 'Object')].history.first().length()`	Gibt die Anzahl der History-Array-Elemente aus dem ersten Objekt mit dem Namen „Object“ zurück.
`$[?(@.price > 10)].length()`	Gibt die Anzahl der Objekte mit einem Preis größer als 10 zurück.

Siehe auch: Escaping special characters from LLD macro values in JSONPath.

Unterstützte Segmente

Segment	Beschreibung
`<name>`	Objekteigenschaft anhand des Namens abgleichen.
`*`	Alle Objekteigenschaften abgleichen.
`['<name>']`	Objekteigenschaft anhand des Namens abgleichen.
`['<name>', '<name>', ...]`	Objekteigenschaft anhand eines der aufgeführten Namen abgleichen.
`[<index>]`	Array-Element anhand des Index abgleichen.
`[<number>, <number>, ...]`	Array-Element anhand eines der aufgeführten Indizes abgleichen.
`[*]`	Alle Objekteigenschaften oder Array-Elemente abgleichen.
`[<start>:<end>]`	Array-Elemente anhand des definierten Bereichs abgleichen: **<start>** - der erste abzugleichende Index (einschließlich); wenn nicht angegeben, werden alle Array-Elemente ab dem Anfang abgeglichen; wenn negativ, wird der Startversatz vom Ende des Arrays angegeben; **<end>** - der letzte abzugleichende Index (ausschließlich); wenn nicht angegeben, werden alle Array-Elemente bis zum Ende abgeglichen; wenn negativ, wird der Startversatz vom Ende des Arrays angegeben.
`[?(<expression>)]`	Objekte/Array-Elemente durch Anwenden eines Filterausdrucks abgleichen.

Um ein passendes Segment unabhängig von seiner Herkunft zu finden (detached segment), muss ihm mit zwei Punkten (`..`) vorangestellt werden. Zum Beispiel geben `$..name` oder `$..['name']` die Werte aller `name`-Eigenschaften zurück.

Namen abgeglichener Elemente können extrahiert werden, indem dem JSONPath ein Tilde-Suffix (~) hinzugefügt wird. Dabei wird der Name des abgeglichenen Objekts oder ein Index des abgeglichenen Array-Elements im Zeichenfolgenformat zurückgegeben. Das Ausgabeformat folgt denselben Regeln wie bei anderen JSONPath-Abfragen – Ergebnisse eines definitiven Pfads werden unverändert zurückgegeben, und Ergebnisse eines nicht definitiven Pfads werden in einem Array zurückgegeben. Allerdings hat das Extrahieren des Namens eines Elements, das einem definitiven Pfad entspricht, nur einen geringen Nutzen, da dieser bereits bekannt ist.

Filterausdruck

Der Filterausdruck ist ein arithmetischer Ausdruck in Infix-Notation.

Unterstützte Operanden:

Operand	Beschreibung

`"<text>"`
`'<text>'`
	Textkonstante.
	
	Beispiel:
	`'value: \'1\''`
	`"value: '1'"`

`<number>`	Numerische Konstante mit Unterstützung für wissenschaftliche Notation. Beispiel: 123
`<jsonpath starting with $>`	Wert, auf den per JSONPath vom Wurzelknoten des Eingabedokuments verwiesen wird; es werden nur definite Pfade unterstützt. Beispiel: `$.object.name`
`<jsonpath starting with @>`	Wert, auf den per JSONPath vom aktuellen Objekt/Element verwiesen wird; es werden nur definite Pfade unterstützt. Beispiel: `@.name`

Unterstützte Operatoren: |Operator|Typ|Beschreibung|Ergebnis| |-|---|---|---| |−|Binär|Subtraktion|Zahl| |+|Binär|Addition|Zahl| |/|Binär|Division|Zahl| |*|Binär|Multiplikation|Zahl| |==|Binär|Gleichheit|Boolesch (1/0)| |!=|Binär|Ungleichheit|Boolesch (1/0)| |<|Binär|Kleiner als|Boolesch (1/0)| |<=|Binär|Kleiner als oder gleich|Boolesch (1/0)| |>|Binär|Größer als|Boolesch (1/0)| |>=|Binär|Größer als oder gleich|Boolesch (1/0)| |=~|Binär|Entspricht regulärem Ausdruck|Boolesch (1/0)| |!|Unary|Boolesches NICHT|Boolesch (1/0)| ||||Binär|Boolesches ODER|Boolesch (1/0)| |&&|Binär|Boolesches UND|Boolesch (1/0)|

Funktionen

Funktionen können am Ende von JSONPath verwendet werden. Mehrere Funktionen können verkettet werden, wenn die vorhergehende Funktion einen Wert zurückgibt, der von der nachfolgenden Funktion akzeptiert wird.

Unterstützte Funktionen:

Funktion	Beschreibung	Eingabe	Ausgabe
`avg`	Durchschnittswert der Zahlen in einem Eingabe-Array	Array von Zahlen	Zahl
`min`	Minimalwert der Zahlen in einem Eingabe-Array	Array von Zahlen	Zahl
`max`	Maximalwert der Zahlen in einem Eingabe-Array	Array von Zahlen	Zahl
`sum`	Summe der Zahlen in einem Eingabe-Array	Array von Zahlen	Zahl
`length`	Anzahl der Elemente in einem Eingabe-Array	Array	Zahl
`first`	Das erste Element eines Arrays	Array	Ein JSON-Konstrukt (Objekt, Array, Wert) abhängig vom Inhalt des Eingabe-Arrays

JSONPath-Aggregatfunktionen akzeptieren numerische Werte in Anführungszeichen. Diese Werte werden automatisch von Zeichenfolgen in numerische Typen umgewandelt, wenn eine Aggregation erforderlich ist. Nicht kompatible Eingaben führen dazu, dass die Funktion einen Fehler erzeugt.

Ausgabewert

JSONPaths können in bestimmte und unbestimmte Pfade unterteilt werden. Ein bestimmter Pfad kann nur null oder eine einzelne Übereinstimmung zurückgeben. Ein unbestimmter Pfad kann mehrere Übereinstimmungen zurückgeben: JSONPaths mit losgelösten Segmenten, mehreren Namens-/Indexlisten, Array-Slices oder Ausdruckssegmenten. Wird jedoch eine Funktion verwendet, wird der JSONPath zu einem bestimmten Pfad, da Funktionen immer einen einzelnen Wert ausgeben.

Ein bestimmter Pfad gibt das Objekt/Array/den Wert zurück, auf das/den er verweist. Im Gegensatz dazu gibt ein unbestimmter Pfad ein Array der übereinstimmenden Objekte/Arrays/Werte zurück.

> **Attention:**
> Die Reihenfolge der Eigenschaften in JSONPath-Abfrageergebnissen stimmt aufgrund interner Optimierungsmethoden möglicherweise nicht mit der ursprünglichen Reihenfolge der JSON-Eigenschaften überein. Beispielsweise kann der JSONPath `$.books[1]["author", "title"] ["title", "author"]` zurückgeben. Wenn die Beibehaltung der ursprünglichen Reihenfolge der Eigenschaften wesentlich ist, sollten alternative Methoden zur Nachbearbeitung nach der Abfrage in Betracht gezogen werden.

Regeln für die Pfadformatierung

Leerzeichen (Leerzeichen, Tabulatorzeichen) können in Segmenten und Ausdrücken der Klammernotation verwendet werden, zum Beispiel: `$[ 'a' ][ 0 ][ ?( $.b == 'c' ) ][ : -1 ].first( )`.

Zeichenfolgen sollten in einfache (') oder doppelte (") Anführungszeichen gesetzt werden. Innerhalb der Zeichenfolgen werden einfache oder doppelte Anführungszeichen (je nachdem, welche zum Einschließen verwendet werden) und Backslashes (\) mit dem Backslash-Zeichen (\) maskiert.

Beispiel

```
{
  "books": [
    {
      "category": "reference",
      "author": "Nigel Rees",
      "title": "Sprichwörter des Jahrhunderts",
      "price": 8.95,
      "id": 1
    },
    {
      "category": "fiction",
      "author": "Evelyn Waugh",
      "title": "Schwert der Ehre",
      "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": "Der Herr der Ringe",
      "isbn": "0-395-19395-8",
      "price": 22.99,
      "id": 4
    }
  ],
  "services": {
    "delivery": {
      "servicegroup": 1000,
      "description": "Lieferung am nächsten Tag in der örtlichen Stadt",
      "active": true,
      "price": 5
    },
    "bookbinding": {
      "servicegroup": 1001,
      "description": "Druck und Bindung eines Buches im A5-Format",
      "active": true,
      "price": 154.99
    },
    "restoration": {
      "servicegroup": 1002,
      "description": "Verschiedene Restaurierungsmethoden",
      "active": false,
      "methods": [
        {
          "description": "Chemische Reinigung",
          "price": 46
        },
        {
          "description": "Pressen von durch Feuchtigkeit beschädigten Seiten",
          "price": 24.5
        },
        {
          "description": "Neubinden eines zerrissenen Buches",
          "price": 99.49
        }
      ]
    }
  },
  "filters": {
    "price": 10,
    "category": "fiction",
    "no filters": "keine \"filters\""
  },
  "closed message": "Der Laden ist geschlossen",
  "tags": [
    "a",
    "b",
    "c",
    "d",
```

```
    "e"
  ]
}
```

JSONPath	Type	Result
`$.filters.price`	definite	10
`$.filters.category`	definite	fiction
`$.filters['no filters']`	definite	keine ”filters”
`$.filters`	definite	{ ”price”: 10, ”category”: ”fiction”, ”no filters”: ”keine \”filters\”” }
`$.books[1].title`	definite	Schwert der Ehre
`$.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
`$.books[0, 2].title`	indefinite	[”Moby Dick”, ”Sprichwörter des Jahrhunderts”]
`$.books[1]['author', "title"]`	indefinite	[”Schwert der Ehre”, ”Evelyn Waugh”]
`$..id`	indefinite	[1, 2, 3, 4]
`$.services..price`	indefinite	[154.99, 5, 46, 24.5, 99.49]
`$.books[?(@.id == 4 - 0.4 * 5)].title`	indefinite	[”Schwert der Ehre”] Hinweis: Diese Abfrage zeigt, dass arithmetische Operationen in Abfragen verwendet werden können; sie kann vereinfacht werden zu `$.books[?(@.id == 2)].title`
`$.books[?(@.id == 2 || @.id == 4)].title`	indefinite	[”Schwert der Ehre”, ”Der Herr der Ringe”]
`$.books[?(!(@.id == 2))].title`	indefinite	[”Sprichwörter des Jahrhunderts”, ”Moby Dick”, ”Der Herr der Ringe”]
`$.books[?(@.id != 2)].title`	indefinite	[”Sprichwörter des Jahrhunderts”, ”Moby Dick”, ”Der Herr der Ringe”]
`$.books[?(@.title =~ " of ")].title`	indefinite	[”Sprichwörter des Jahrhunderts”, ”Schwert der Ehre”, ”Der Herr der Ringe”]
`$.books[?(@.price > 12.99)].title`	indefinite	[”Der Herr der Ringe”]
`$.books[?(@.author > "Herman Melville")].title`	indefinite	[”Sprichwörter des Jahrhunderts”, ”Der Herr der Ringe”]
`$.books[?(@.price > $.filters.price)].title`	indefinite	[”Schwert der Ehre”, ”Der Herr der Ringe”]
`$.books[?(@.category == $.filters.category)].title`	indefinite	[”Schwert der Ehre”,”Moby Dick”,”Der Herr der Ringe”]
`$.books[?(@.category == "fiction" && @.price < 10)].title`	indefinite	[”Moby Dick”]

JSONPath	Type	Result

`$..[?(@.id)]`	indefinite	[
		{
		”price”: 8.95,
		”id”: 1,
		”category”: ”reference”,
		”author”: ”Nigel Rees”,
		”title”: ”Sprichwörter des Jahrhunderts”
		},
		{
		”price”: 12.99,
		”id”: 2,
		”category”: ”fiction”,
		”author”: ”Evelyn Waugh”,
		”title”: ”Schwert der Ehre”
		},
		{
		”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”: ”Der Herr der Ringe”,
		”isbn”: ”0-395-19395-8”
		}
		]

`$.services..[?(@.price > 50)].description`	indefinite	[”Druck und Bindung eines Buches im A5-Format”, ”Neubinden eines zerrissenen Buches”]
`$..id.length()`	definite	4
`$.books[?(@.id == 2)].title.first()`	definite	Schwert der Ehre
`$..tags.first().length()`	definite	5 Hinweis: `$..tags` ist ein unbestimmter Pfad, daher gibt er ein Array der übereinstimmenden Elemente zurück, also `[["a", "b", "c", "d", "e" ]]`; `first()` gibt das erste Element zurück, also `["a", "b", "c", "d", "e"]`; `length()` berechnet die Länge des Elements, also 5.
`$.books[*].price.min()`	definite	8.95
`$..price.max()`	definite	154.99
`$.books[?(@.category == "fiction")].price.avg()`	definite	14.99
`$.books[?(@.category == $.filters.xyz)].title`	indefinite	Hinweis: Eine Abfrage ohne Treffer gibt für bestimmte und unbestimmte Pfade NULL zurück.
`$.services[?(@.active=="true")].servicegroup`	indefinite	[1001,1000] Hinweis: Bei Vergleichen von booleschen Werten müssen Textkonstanten verwendet werden.
`$.services[?(@.active=="false")].servicegroup`	indefinite	[1002] Hinweis: Bei Vergleichen von booleschen Werten müssen Textkonstanten verwendet werden.
`$.services[?(@.servicegroup=="1002")].description`	definite	Testdienst

Sonderzeichen von LLD-Makrowerten in JSONPath maskieren

Wenn Low-Level-Discovery-Makros in der JSONPath-Vorverarbeitung verwendet werden und ihre Werte aufgelöst werden, gelten die folgenden Regeln zur Maskierung von Sonderzeichen:

- nur der Backslash (\) und das doppelte Anführungszeichen (") werden für die Maskierung berücksichtigt;
- wenn der aufgelöste Makrowert diese Zeichen enthält, wird jedes davon mit einem Backslash maskiert;
- wenn sie bereits mit einem Backslash maskiert sind, gilt dies nicht als Maskierung, und sowohl der Backslash als auch die folgenden Sonderzeichen werden erneut einmal maskiert.

Zum Beispiel:

JSONPath	LLD-Makrowert	Nach der Substitution
$.[?(@.value == "{#MACRO}")]	special "value"	$.[?(@.value == "special \"value\"")]
	c:\temp	$.[?(@.value == "c:\\temp")]
	a\\b	$.[?(@.value == "a\\\\b")]

Wenn das Makro im Ausdruck verwendet wird und Sonderzeichen enthalten kann, sollte es in doppelte Anführungszeichen eingeschlossen werden:

JSONPath	LLD-Makrowert	Nach der Substitution	Ergebnis
$.[?(@.value == "{#MACRO}")]	special "value"	$.[?(@.value == "special \"value\"")]	OK
$.[?(@.value == {#MACRO})]		$.[?(@.value == special \"value\")]	**Ungültiger JSONPath-Ausdruck**

Wenn das Makro im Pfad verwendet wird und Sonderzeichen enthalten kann, sollte es in eckige Klammern **und** doppelte Anführungszeichen eingeschlossen werden:

JSONPath	LLD-Makrowert	Nach der Substitution	Ergebnis
$.["{#MACRO}"].value	c:\temp	$.["c:\\temp"].value	OK
$.{#MACRO}.value		$.c:\\temp.value	**Ungültiger JSONPath-Ausdruck**

JavaScript-Vorverarbeitung

Übersicht

Dieser Abschnitt enthält Details zur Vorverarbeitung mit JavaScript.

> **Attention:**
> Verwenden Sie in der Vorverarbeitungs-JavaScript keine Zuweisungen an nicht deklarierte Variablen. Verwenden Sie `var`, um lokale Variablen zu deklarieren.

JavaScript-Vorverarbeitung

Die JavaScript-Vorverarbeitung erfolgt durch Aufruf einer JavaScript-Funktion mit einem einzelnen Parameter 'value' und einem vom Benutzer bereitgestellten Funktionsrumpf. Das Ergebnis des Vorverarbeitungsschritts ist der von dieser Funktion zurückgegebene Wert. Um beispielsweise eine Umrechnung von Fahrenheit in Celsius durchzuführen, geben Sie Folgendes ein:

```
return (value - 32)  * 5 / 9
```

in den JavaScript-Vorverarbeitungsparametern, die vom Server in eine JavaScript-Funktion eingebettet werden:

```
function (value)
{
   return (value - 32) * 5 / 9
}
```

Der Eingabeparameter 'value' wird immer als Zeichenfolge übergeben. Der Rückgabewert wird automatisch über die Methode toString() in eine Zeichenfolge umgewandelt (falls dies fehlschlägt, wird der Fehler als Zeichenfolgenwert zurückgegeben), mit einigen Ausnahmen:

- die Rückgabe des Werts undefined führt zu einem Fehler;
- die Rückgabe des Werts null bewirkt, dass der Eingabewert verworfen wird, ähnlich wie bei der Vorverarbeitung 'Discard value' bei der Aktion 'Custom on fail'.

Fehler können zurückgegeben werden, indem Werte/Objekte ausgelöst werden (normalerweise entweder Zeichenfolgen oder Error-Objekte).

Beispiel:

```
if (value == 0)
    throw "Zero input value"
return 1/value
```

Jedes Skript hat ein Ausführungs-Timeout von 10 Sekunden (je nach Skript kann es länger dauern, bis das Timeout ausgelöst wird); bei Überschreitung wird ein Fehler zurückgegeben. Es wird ein Heap-Limit von 512 Megabyte erzwungen.

Der Bytecode des JavaScript-Vorverarbeitungsschritts wird zwischengespeichert und wiederverwendet, wenn der Schritt das nächste Mal angewendet wird. Jede Änderung an den Vorverarbeitungsschritten des Datenpunkts führt dazu, dass das zwischengespeicherte Skript zurückgesetzt und später neu kompiliert wird.

Aufeinanderfolgende Laufzeitfehler (3 in Folge) führen dazu, dass die Engine neu initialisiert wird, um die Möglichkeit zu verringern, dass ein Skript die Ausführungsumgebung für die nächsten Skripte beeinträchtigt (diese Aktion wird mit DebugLevel 4 und höher protokolliert).

Die JavaScript-Vorverarbeitung wird mit der JavaScript-Engine Duktape implementiert.

> **Note:**
> Die Definitionen integrierter JavaScript-Objektmethoden können in Datenpunkt-Vorverarbeitungsskripten nicht neu definiert werden.

Siehe auch: Zusätzliche JavaScript-Objekte und globale Funktionen

Verwendung von Makros in Skripten

Es ist möglich, Benutzermakros (sowie LLD-Makros im Kontext der Low-Level-Discovery) in JavaScript-Code zu verwenden. Wenn ein Skript Benutzermakros enthält, werden diese Makros vor der Ausführung bestimmter Vorverarbeitungsschritte vom Server/Proxy aufgelöst. Beachten Sie, dass beim Testen von Vorverarbeitungsschritten im Frontend keine Makrowerte abgerufen werden und diese manuell eingegeben werden müssen.

> **Note:**
> Der Kontext wird ignoriert, wenn ein Makro durch seinen Wert ersetzt wird. Der Makrowert wird unverändert in den Code eingefügt; es ist nicht möglich, vor dem Einfügen des Werts in den JavaScript-Code zusätzliches Escaping hinzuzufügen. Bitte beachten Sie, dass dies in einigen Fällen zu JavaScript-Fehlern führen kann.

Im folgenden Beispiel wird, wenn der empfangene Wert den Wert des Makros {$THRESHOLD} überschreitet, stattdessen der Schwellenwert (falls vorhanden) zurückgegeben:

```
var threshold = '{$THRESHOLD}';
return (!isNaN(threshold) && value > threshold) ? threshold : value;
```

Beispiele

Die folgenden Beispiele veranschaulichen, wie Sie die JavaScript-Vorverarbeitung verwenden können.

Jedes Beispiel enthält eine kurze Beschreibung, einen Funktionsrumpf für JavaScript-Vorverarbeitungsparameter und das Ergebnis des Vorverarbeitungsschritts – den von der Funktion zurückgegebenen Wert.

Beispiel 1: Eine Zahl umwandeln (wissenschaftliche Notation in Ganzzahl)

Wandeln Sie die Zahl „2.62128e+07“ von wissenschaftlicher Notation in eine Ganzzahl um.

```
return (Number(value))
```

Von der Funktion zurückgegebener Wert: 26212800.

Beispiel 2: Eine Zahl umwandeln (binär in dezimal)

Wandeln Sie die Binärzahl "11010010" in eine Dezimalzahl um.

```
return(parseInt(value,2))
```

Von der Funktion zurückgegebener Wert: 210.

Beispiel 3: Eine Zahl runden

Runden Sie die Zahl „18.2345“ auf 2 Stellen.

```
return(Math.round(value* 100) / 100)
```

Von der Funktion zurückgegebener Wert: 18.23.

Beispiel 4: Buchstaben in einer Zeichenkette zählen

Zählen Sie die Anzahl der Buchstaben in der Zeichenkette „Zabbix“.

```
return (value.length)
```

Von der Funktion zurückgegebener Wert: 6.

Beispiel 5: Verbleibende Zeit abrufen

Rufen Sie die verbleibende Zeit (in Sekunden) bis zum Ablaufdatum eines Zertifikats ab (12. Feb. 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);
```

Von der Funktion zurückgegebener Wert: 44380233.

Beispiel 6: JSON-Eigenschaften entfernen

Ändern Sie die JSON-Datenstruktur, indem Sie alle Eigenschaften mit dem Schlüssel "data_size" oder "index_size" entfernen.

```
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)
```

Von der Funktion akzeptierter Wert:

```
[
    {
        "table_name":"history",
        "data_size":"326.05",
        "index_size":"174.34"
    },
    {
        "table_name":"history_log",
        "data_size":"6.02",
        "index_size":"3.45"
    }
]
```

Von der Funktion zurückgegebener Wert:

```
[
    {
        "table_name":"history"
    },
    {
        "table_name":"history_log"
    }
]
```

Beispiel 7: Apache-Status in JSON konvertieren

Konvertieren Sie den Wert, der von einem web.page.get-Zabbix-Agent-Datenpunkt empfangen wird (z. B. web.page.get[http://127.0.0.1:80/server-status?auto]), in ein JSON-Objekt.

```
// Apache-Status in JSON konvertieren

// Den Wert in Teilzeichenfolgen aufteilen und diese Teilzeichenfolgen in ein Array einfügen
var lines = value.split('\n');

// Ein leeres Objekt "output" erstellen
var output = {};

// Ein Objekt "workers" mit vordefinierten Eigenschaften erstellen
var workers = {
    '_': 0, 'S': 0, 'R': 0, 'W': 0,
    'K': 0, 'D': 0, 'C': 0, 'L': 0,
    'G': 0, 'I': 0, '.': 0
};

// Die Teilzeichenfolgen aus dem Array "lines" als Eigenschaften (Schlüssel-Wert-Paare) zum Objekt "output
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]);
    }
}

// Metriken für mehrere Versionen
output.ServerUptimeSeconds = output.ServerUptimeSeconds || output.Uptime;
output.ServerVersion = output.ServerVersion || output.Server;

// Die Eigenschaft "Scoreboard" parsen, um die Anzahl der Worker zu ermitteln
if (typeof output.Scoreboard === 'string') {
    for (var i = 0; i < output.Scoreboard.length; i++) {
        var char = output.Scoreboard[i];

        workers[char]++;
    }
}

// Worker-Daten zum Objekt "output" hinzufügen
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['.']
};

// JSON-Zeichenfolge zurückgeben
return JSON.stringify(output);
```

Von der Funktion akzeptierter Wert:

```
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__.....................................................................................
```

Von der Funktion zurückgegebener Wert:


```
{
    "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
    }
}
```

Zusätzliche JavaScript-Objekte

Übersicht

Dieser Abschnitt beschreibt die von Zabbix vorgenommenen Erweiterungen der mit Duktape implementierten JavaScript-Sprache sowie die unterstützten globalen JavaScript-Funktionen.

Attention:
Verwenden Sie in der Vorverarbeitungs-JavaScript keine Zuweisungen an nicht deklarierte Variablen. Verwenden Sie `var`, um lokale Variablen zu deklarieren.

Integrierte Objekte

Zabbix

Das Zabbix-Objekt ermöglicht die Interaktion mit der internen Zabbix-Funktionalität.

Method	Description
`log(loglevel, message)`	Schreibt <message> mit dem Protokollierungsgrad <loglevel> in das Zabbix-Protokoll (siehe Parameter DebugLevel in der Konfigurationsdatei).

Beispiel:

```
Zabbix.log(3, "this is a log entry written with 'Warning' log level")
```

Sie können die folgenden Aliase verwenden:

Alias	Alias to
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:
Die Gesamtgröße aller protokollierten Meldungen ist auf 8 MB pro Skriptausführung begrenzt.

Method	Description
`sleep(delay)`	Verzögert die JavaScript-Ausführung um `delay` Millisekunden.

Beispiel (Ausführung um 15 Sekunden verzögern):

```
Zabbix.sleep(15000)
```

HttpRequest

Dieses Objekt kapselt ein cURL-Handle und ermöglicht das Ausführen einfacher HTTP-Anfragen. Fehler werden als Ausnahmen ausgelöst.

> **Attention:**
> Die Initialisierung mehrerer `HttpRequest`-Objekte ist auf 10 pro Skriptausführung begrenzt.

Method	Description
`addHeader(value)`	Fügt ein HTTP-Headerfeld hinzu. Dieses Feld wird für alle folgenden Anfragen verwendet, bis es mit der Methode `clearHeader()` gelöscht wird. Die Gesamtlänge der Headerfelder, die einem einzelnen `HttpRequest`-Objekt hinzugefügt werden können, ist auf 128 KByte begrenzt (einschließlich Sonderzeichen und Headernamen).
`clearHeader()`	Löscht den HTTP-Header. Wenn keine Headerfelder festgelegt sind, setzt `HttpRequest` Content-Type auf application/json, wenn die zu sendenden Daten im JSON-Format vorliegen, andernfalls auf text/plain.
`connect(url)`	Sendet eine HTTP-CONNECT-Anfrage an die URL und gibt die Antwort zurück.
`customRequest(method, url, data)`	Ermöglicht die Angabe einer beliebigen HTTP-Methode im ersten Parameter. Sendet die Methodenanforderung mit einer optionalen *data*-Nutzlast an die URL und gibt die Antwort zurück.
`delete(url, data)`	Sendet eine HTTP-DELETE-Anfrage mit einer optionalen *data*-Nutzlast an die URL und gibt die Antwort zurück.
`getHeaders(<asArray>)`	Gibt das Objekt der empfangenen HTTP-Headerfelder zurück. Der Parameter `asArray` kann auf "true" (z. B. `getHeaders(true)`), "false" gesetzt werden oder nicht definiert sein. Bei "true" werden die Werte der empfangenen HTTP-Headerfelder als Arrays zurückgegeben. Dies sollte verwendet werden, um die Feldwerte mehrerer Header mit demselben Namen abzurufen. Wenn der Parameter nicht gesetzt oder auf "false" gesetzt ist, werden die Werte der empfangenen HTTP-Headerfelder als Zeichenfolgen zurückgegeben.
`get(url, data)`	Sendet eine HTTP-GET-Anfrage mit einer optionalen *data*-Nutzlast an die URL und gibt die Antwort zurück.
`head(url)`	Sendet eine HTTP-HEAD-Anfrage an die URL und gibt die Antwort zurück.
`options(url)`	Sendet eine HTTP-OPTIONS-Anfrage an die URL und gibt die Antwort zurück.
`patch(url, data)`	Sendet eine HTTP-PATCH-Anfrage mit einer optionalen *data*-Nutzlast an die URL und gibt die Antwort zurück.
`put(url, data)`	Sendet eine HTTP-PUT-Anfrage mit einer optionalen *data*-Nutzlast an die URL und gibt die Antwort zurück.
`post(url, data)`	Sendet eine HTTP-POST-Anfrage mit einer optionalen *data*-Nutzlast an die URL und gibt die Antwort zurück.
`getStatus()`	Gibt den Statuscode der letzten HTTP-Anfrage zurück.
`setProxy(proxy)`	Setzt den HTTP-Proxy auf den Wert "proxy". Wenn dieser Parameter leer ist, wird kein Proxy verwendet.

`setHttpAuth(bitmask, username, password)`	Legt die aktivierten HTTP-Authentifizierungsmethoden (HTTPAUTH_BASIC, HTTPAUTH_DIGEST, HTTPAUTH_NEGOTIATE, HTTPAUTH_NTLM, HTTPAUTH_NONE) im Parameter 'bitmask' fest.
	Mit dem Flag HTTPAUTH_NONE kann die HTTP-Authentifizierung deaktiviert werden.
	Beispiele:
	`request.setHttpAuth(HTTPAUTH_NTLM | HTTPAUTH_BASIC, username, password)`
	`request.setHttpAuth(HTTPAUTH_NONE)`

`trace(url, data)`	Sendet eine HTTP-TRACE-Anfrage mit einer optionalen *data*-Nutzlast an die URL und gibt die Antwort zurück.

Beispiel:

```
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

Das XML-Objekt ermöglicht die Verarbeitung von XML-Daten in der Datenpunkt- und Low-Level-Discovery-Präprozessierung sowie in webhooks.

> **Attention:**
> Um das XML-Objekt zu verwenden, müssen Server/Proxy mit Unterstützung für libxml2 kompiliert sein.

Method	Description
`XML.query(data, expression)`	Ruft den Inhalt eines Knotens mithilfe von XPath ab. Gibt null zurück, wenn der Knoten nicht gefunden wird. **expression** - ein XPath-Ausdruck; **data** - XML-Daten als Zeichenfolge.
`XML.toJson(data)`	Konvertiert Daten im XML-Format in JSON.
`XML.fromJson(object)`	Konvertiert Daten im JSON-Format in XML.

Beispiel:

Eingabe:

```
<menu>
    <food type = "breakfast">
        <name>Chocolate</name>
        <price>$5.95</price>
        <description></description>
        <calories>650</calories>
    </food>
</menu>
```

Ausgabe:

```
{
    "menu": {
```

```
        "food": {
            "@type": "breakfast",
            "name": "Chocolate",
            "price": "$5.95",
            "description": null,
            "calories": "650"
        }
    }
}
```

Serialisierungsregeln

Die Konvertierung von XML nach JSON wird gemäß den folgenden Regeln verarbeitet (für Konvertierungen von JSON nach XML werden die Regeln umgekehrt angewendet):

1. XML-Attribute werden in Schlüssel umgewandelt, deren Namen mit '@' vorangestellt werden.

Beispiel:

Eingabe:

```
<xml foo="FOO">
  <bar>
    <baz>BAZ</baz>
  </bar>
</xml>
```

Ausgabe:

```
{
  "xml": {
    "@foo": "FOO",
    "bar": {
      "baz": "BAZ"
    }
  }
}
```

2. Selbstschließende Elemente (<foo/>) werden so umgewandelt, als hätten sie den Wert 'null'.

Beispiel:

Eingabe:

```
<xml>
  <foo/>
</xml>
```

Ausgabe:

```
{
  "xml": {
    "foo": null
  }
}
```

3. Leere Attribute (mit dem Wert "") werden so umgewandelt, als hätten sie einen leeren String-Wert ("").

Beispiel:

Eingabe:

```
<xml>
  <foo bar="" />
</xml>
```

Ausgabe:

```
{
  "xml": {
    "foo": {
      "@bar": ""
```

```
    }
  }
}
```

4. Mehrere untergeordnete Knoten mit demselben Elementnamen werden in einen einzelnen Schlüssel umgewandelt, dessen Wert ein Array von Werten ist.

Beispiel:

Eingabe:

```
<xml>
  <foo>BAR</foo>
  <foo>BAZ</foo>
  <foo>QUX</foo>
</xml>
```

Ausgabe:

```
{
  "xml": {
    "foo": ["BAR", "BAZ", "QUX"]
  }
}
```

5. Wenn ein Textelement keine Attribute und keine untergeordneten Elemente hat, wird es als String umgewandelt.

Beispiel:

Eingabe:

```
<xml>
    <foo>BAZ</foo>
</xml>
```

Ausgabe:

```
{
  "xml": {
    "foo": "BAZ"
   }
}
```

6. Wenn ein Textelement keine untergeordneten Elemente, aber Attribute hat, wird der Textinhalt in ein Element mit dem Schlüssel '#text' und dem Inhalt als Wert umgewandelt; Attribute werden wie in Serialisierungsregel 1 beschrieben umgewandelt.

Beispiel:

Eingabe:

```
<xml>
  <foo bar="BAR">
    BAZ
  </foo>
</xml>
```

Ausgabe:

```
{
  "xml": {
    "foo": {
      "@bar": "BAR",
      "#text": "BAZ"
    }
  }
}
```

Globale JavaScript-Funktionen

Zusätzliche globale JavaScript-Funktionen wurden mit Duktape implementiert:

- `btoa(data)` - kodiert die Daten in einen base64-String.

- `atob(base64_string)` - dekodiert einen base64-String als Uint8Array-Puffer.

```
try {
    b64 = btoa("test string");
    buffer = atob(b64);

    // Note that decoding logic depends on the data format of the buffer.
    decoded = String.fromCharCode.apply(this, [].slice.call(buffer));
}
catch (error) {
    return {'error.name' : error.name, 'error.message' : error.message};
}
```

- `md5(data)` - berechnet den MD5-Hash der Daten.
- `sha256(data)` - berechnet den SHA256-Hash der Daten.
- `hmac('<hash type>',key,data)` - gibt einen HMAC-Hash als hexadezimal formatierten String zurück; `md5` und `sha256` werden als `hash type` unterstützt; die Parameter `key` und `data` unterstützen Binärdaten.

  Beispiele:

  - `hmac('md5',key,data)`
  - `hmac('sha256',key,data)`

- `sign(hash,key,data)` - gibt die berechnete Signatur (RSA-Signatur mit SHA-256) als String zurück, wobei:
 hash - nur `sha256` zulässig ist, andernfalls wird ein Fehler ausgelöst.
 key - der private Schlüssel ist. Er sollte dem Standard PKCS#1 oder PKCS#8 entsprechen. Der Schlüssel kann in verschiedenen Formen angegeben werden:

  - mit Leerzeichen anstelle von Zeilenumbrüchen
  - mit maskierten oder nicht maskierten `\n` anstelle von Zeilenumbrüchen
  - ohne Zeilenumbrüche als einzeiliger String
  - als JSON-formatierter String

  Der Schlüssel kann auch aus einem Benutzermakro/geheimen Makro/Vault geladen werden.

  data - die Daten, die signiert werden. Es kann sich um einen String handeln (Binärdaten werden ebenfalls unterstützt) oder um einen Puffer (Uint8Array/ArrayBuffer).

  Beispiel:

  - `sign('sha256',key,data)`

  OpenSSL oder GnuTLS wird zur Berechnung der Signaturen verwendet. Wenn Zabbix ohne eine dieser Verschlüsselungsbibliotheken erstellt wurde, wird ein Fehler ausgelöst ('missing OpenSSL or GnuTLS library').

Browser-Datenpunkt JavaScript-Objekte

Überblick

Dieser Abschnitt beschreibt Zabbix-Erweiterungen der mit Duktape implementierten JavaScript-Sprache zur Verwendung im Skript des Browser-Datenpunkts. Diese Erweiterungen ergänzen die auf der Seite Zusätzliche JavaScript-Objekte beschriebenen JavaScript-Objekte.

Browser

Das Objekt `Browser` verwaltet WebDriver-Sitzungen, initialisiert eine Sitzung bei der Erstellung und beendet sie bei der Zerstörung. Ein einzelnes Skript kann bis zu vier `Browser`-Objekte unterstützen.

Um ein `Browser`-Objekt zu erstellen, verwenden Sie die Syntax `new Browser(options)`. Der Parameter `options` (*JSON object*) gibt Browser-Optionen an, in der Regel das Ergebnis der WebDriver-Optionsmethode (zum Beispiel `Browser.chromeOptions()`).

Die folgenden Methoden werden vom Objekt `Browser` unterstützt.

Method	Description
`navigate(url)`	Zur angegebenen URL navigieren. Parameter: `url` - (string) URL, zu der navigiert werden soll.
`getUrl()`	Eine Zeichenfolge mit der URL der geöffneten Seite zurückgeben.

Method	Description
`getPageSource()`	Eine Zeichenfolge mit dem Quelltext der geöffneten Seite zurückgeben.
`findElement(strategy, selector)`	Ein `Element`-Objekt mit einem Element auf der geöffneten Seite zurückgeben (oder `null` zurückgeben, wenn keine Elemente mit `strategy` und `selector` übereinstimmen). Parameter: `strategy` - (string, css selector/link text/partial link text/tag name/xpath) Lokalisierungsstrategie; `selector` - (string) Elementselektor unter Verwendung der angegebenen Lokalisierungsstrategie.
`findElements(strategy, target)`	Ein Array von `Element`-Objekten mit mehreren Elementen auf der geöffneten Seite zurückgeben (oder ein leeres Array zurückgeben, wenn keine Elemente mit der Lokalisierungsstrategie und dem Ziel übereinstimmen). Parameter: `strategy` - (string, css selector/link text/partial link text/tag name/xpath) Lokalisierungsstrategie; `target` - (string) Elementselektor unter Verwendung der angegebenen Lokalisierungsstrategie.
`getCookies()`	Ein Array von `Cookie`-Objekten zurückgeben.
`addCookie(cookie)`	Cookies setzen. Parameter: `cookie` - (`Cookie` object) Zu setzendes Cookie.
`getScreenshot()`	Eine base64-kodierte Zeichenfolge zurückgeben, die ein Bild des Viewports des Browsers darstellt.
`setScreenSize(x,y)`	Die Größe des Browser-Viewports festlegen. Parameter: `x` - (string) Breite in Pixeln; `y` - (string) Höhe in Pixeln.
`setScriptTimeout(timeout)`	Das Timeout für das Laden von Skripten festlegen. Parameter: `timeout` - (integer) Timeout-Wert in Millisekunden.
`setSessionTimeout(timeout)`	Das Sitzungs-Timeout (Seitenlade-Timeout) festlegen. Parameter: `timeout` - (integer) Timeout-Wert in Millisekunden.
`setElementWaitTimeout(timeout)`	Das (implizite) Timeout für die Element-Lokalisierungsstrategie festlegen. Parameter: `timeout` - (integer) Timeout-Wert in Millisekunden.
`collectPerfEntries(mark)`	Performance-Einträge sammeln, die mit der Methode `getResult()` abgerufen werden können. Parameter: `mark` - (string, optional) Markierung für den Performance-Snapshot.
`getRawPerfEntries()`	Ein Array von Performance-Eintragsobjekten zurückgeben.
`getResult()`	Ein `Result`-Objekt mit Browser-Sitzungsstatistiken zurückgeben (Fehlerinformationen, Performance-Snapshots usw.).
`getError()`	Ein `BrowserError`-Objekt mit Browser-Fehlern zurückgeben (oder `null` zurückgeben, wenn keine Browser-Fehler vorhanden sind).
`setError(message)`	Eine benutzerdefinierte Fehlermeldung festlegen, die in das `Result`-Objekt aufgenommen wird. Parameter: `message` - (string) Fehlermeldung.
`discardError()`	Den Fehler verwerfen, der im `Result`-Objekt zurückgegeben werden soll.
`getAlert()`	Ein `Alert`-Objekt mit Browser-Warnmeldungen zurückgeben (oder `null` zurückgeben, wenn keine Browser-Warnmeldungen vorhanden sind).

Method	Description
`chromeOptions()`	Ein `chromeOptions`-Objekt mit vordefinierten Chrome-Browser-Optionen zurückgeben.
`firefoxOptions()`	Ein `firefoxOptions`-Objekt mit vordefinierten Firefox-Browser-Optionen zurückgeben.
`safariOptions()`	Ein `safariOptions`-Objekt mit vordefinierten Safari-Browser-Optionen zurückgeben.
`edgeOptions()`	Ein `edgeOptions`-Objekt mit vordefinierten Edge-Browser-Optionen zurückgeben.
`switchFrame(target)`	Zum angegebenen Frame wechseln. Parameter: `target` - (browser element or integer, optional) Ziel-Frame. Um einen Frame anhand eines Elements auszuwählen, übergeben Sie das Element. Um einen Frame anhand des Index auszuwählen, übergeben Sie die Zahl. Wenn leer gelassen, wird zum Browsing-Kontext der obersten Ebene gewechselt.

Alle Browser-Methoden können die folgenden Fehler auslösen:

- `BrowserError` - vom Objekt `Error` abgeleitet und wird ausgelöst, wenn der `Browser`-Konstruktor fehlschlägt; enthält zusätzlich eine Eigenschaft `browser` mit einem `Browser`-Objekt, das diesen `BrowserError` ausgelöst hat.
- `WebdriverError` - von `BrowserError` abgeleitet; enthält dieselben Eigenschaften wie das Objekt `BrowserError`, die anzeigen, ob der Fehler als Reaktion auf einen Fehler in der WebDriver-Antwort erzeugt wurde.

Element

Das `Element`-Objekt wird von den Methoden `findElement()`/`findElements()` des `Browser`-Objekts zurückgegeben und kann nicht direkt erstellt werden.

Das `Element`-Objekt repräsentiert ein Element in der Webseite und stellt Methoden bereit, um mit ihm zu interagieren.

Die folgenden Methoden werden mit dem `Element`-Objekt unterstützt.

Method	Description
`getAttribute(name)`	Gibt einen Attributwert als Zeichenfolge des Elementattributs zurück (oder `null`, wenn das angegebene Attribut nicht gefunden wurde). Parameters: `name` - (string) Attributname.
`getProperty(name)`	Gibt einen Eigenschaftswert als Zeichenfolge der Elementeigenschaft zurück (oder `null`, wenn die angegebene Eigenschaft nicht gefunden wurde). Parameters: `name` - (string) Eigenschaftsname.
`getText()`	Gibt einen Textwert als Zeichenfolge des Elementtexts zurück.
`click()`	Klickt auf ein Element.
`clear()`	Löscht den Inhalt eines bearbeitbaren Elements.
`sendKeys(keys)`	Sendet Tastenanschläge. Parameters: `keys` - (string) Zu sendende Tasten.

Cookie

Das `Cookie`-Objekt wird von der Methode `getCookies()` des `Browser`-Objekts zurückgegeben und an die Methode `addCookie()` übergeben.

Obwohl das `Cookie`-Objekt keine Methoden hat, kann es die folgenden Eigenschaften enthalten:

Property	Type	Description
name	string	Name des Cookies.
value	string	Wert des Cookies.

Property	Type	Description
path	string	Pfad, für den das Cookie gültig ist. Standardmäßig `"/"`, wenn es beim Hinzufügen eines Cookies weggelassen wird.
domain	string	Domain, für die das Cookie sichtbar ist. Standardmäßig die URL-Domain des aktiven Dokuments im aktuellen Browsing-Kontext der Sitzung, wenn sie beim Hinzufügen eines Cookies weggelassen wird.
secure	boolean	Gibt an, ob das Cookie sicher ist. Standardmäßig `false`, wenn es beim Hinzufügen eines Cookies weggelassen wird.
httpOnly	boolean	Gibt an, ob das Cookie nur über HTTP verfügbar ist. Standardmäßig `false`, wenn es beim Hinzufügen eines Cookies weggelassen wird.
expiry	integer	Ablaufzeit des Cookies (in Sekunden seit der Unix-Epoche). Darf nicht gesetzt werden, wenn sie beim Hinzufügen eines Cookies weggelassen wird.
sameSite	string	Das Attribut `sameSite` des Cookies, das steuert, ob das Cookie auf einen First-Party- oder Same-Site-Kontext beschränkt werden soll. Kann entweder auf `"Lax"` oder `"Strict"` gesetzt werden. Standardmäßig `"None"`, wenn es beim Hinzufügen eines Cookies weggelassen wird.

Warnung

Das Objekt `Alert` repräsentiert eine Warnung auf einer Webseite, wird von der Methode `getAlert()` des Objekts `Browser` zurückgegeben und kann nicht direkt erstellt werden.

Das Objekt `Alert` enthält die Eigenschaft `text` mit dem Text der Warnung (oder `null`, wenn keine Warnungen vorhanden sind).

Die folgenden Methoden werden vom Objekt `Alert` unterstützt.

Method	Description
`accept()`	Die Warnung akzeptieren.
`dismiss()`	Die Warnung verwerfen.

Ergebnis

Das Objekt `Result` enthält Sitzungsstatistiken und wird von der Methode `getResult()` des Objekts `Browser` zurückgegeben.

Typischerweise wird das Objekt `Result` in eine Zeichenkette umgewandelt und vom Skript zurückgegeben und anschließend durch Vorverarbeitung in Werte abhängiger Datenpunkte geparst.

Obwohl das Objekt `Result` keine Methoden hat, kann es die folgenden Eigenschaften enthalten.

Eigenschaft	Typ	Beschreibung
duration	string	Sitzungsdauer von der Erstellung der Sitzung bis zum Abruf des Ergebnisses.
error	object	Fehlerinformationen.
http_status	integer	Vom WebDriver zurückgegebener HTTP-Status (oder 0, wenn keine WebDriver-Fehler vorliegen).
error_code	string	Vom WebDriver zurückgegebener Fehler (oder eine leere Zeichenkette, wenn keine WebDriver-Fehler vorliegen).
message	string	WebDriver-Fehlermeldung (oder eine leere Zeichenkette, wenn keine WebDriver-Fehler vorliegen).
performance_data	object	Leistungsstatistiken.
summary	object	Leistungsübersicht.
navigation	object	Navigationsübersicht.
resource	object	Ressourcenübersicht.
details	array of objects	Leistungsstatistiken nach jeder Operation, die zu einer Navigation geführt haben könnte.
mark	string	*(optional)* Mit der Methode `collectPerfEntries()` angegebene Markierung der Leistungsaufnahme.

Eigenschaft		Typ	Beschreibung
	navigation	object	Navigationsstatistiken.
	resource	object	Ressourcenübersicht für diesen Schritt.
	user	array of objects	Array von Statistiken des Typs mark/measure.
marks		array of objects	Indizes markierter Leistungsaufnahmen.
	name	string	Name der Markierung der Leistungsaufnahme.
	index	integer	Index der Leistungsaufnahme im Array details.

CSV-zu-JSON-Vorverarbeitung

Übersicht

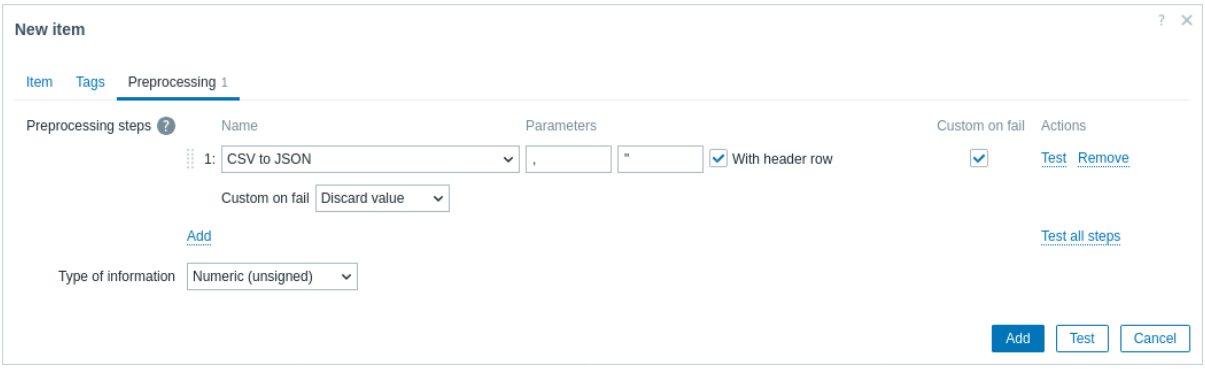
In diesem Vorverarbeitungsschritt können Daten aus CSV-Dateien in das JSON-Format konvertiert werden. Dies wird unterstützt in:

- Datenpunkten (Datenpunkt-Prototypen)
- Low-Level-Discovery-Regeln

Konfiguration

So konfigurieren Sie einen CSV-zu-JSON-Schritt der Vorverarbeitung:

1. Wechseln Sie in der Konfiguration des Datenpunkts/der Ermittlungsregel zur Registerkarte *Vorverarbeitung*.
2. Klicken Sie auf *Hinzufügen*.
3. Wählen Sie die Option *CSV zu JSON* aus.



Mit dem ersten Parameter können Sie ein benutzerdefiniertes Trennzeichen festlegen. Beachten Sie, dass das Zeichen als Trennzeichen verwendet wird, wenn die erste Zeile der CSV-Eingabe mit „Sep=“ beginnt und darauf ein einzelnes UTF-8-Zeichen folgt, sofern der erste Parameter nicht festgelegt ist. Wenn der erste Parameter nicht festgelegt ist und kein Trennzeichen aus der „Sep=“-Zeile abgerufen wird, wird ein Komma als Trennzeichen verwendet.

Mit dem zweiten optionalen Parameter können Sie ein Anführungszeichen festlegen.

Wenn das Kontrollkästchen *Mit Kopfzeile* aktiviert ist, werden die Werte der Kopfzeile als Spaltennamen interpretiert (weitere Informationen finden Sie unter Verarbeitung der CSV-Kopfzeile).

Wenn das Kontrollkästchen *Benutzerdefiniert bei Fehler* aktiviert ist, wird der Datenpunkt im Fall eines fehlgeschlagenen Vorverarbeitungsschritts nicht nicht unterstützt. Zusätzlich können benutzerdefinierte Optionen für die Fehlerbehandlung festgelegt werden: den Wert verwerfen, einen bestimmten Wert festlegen oder eine bestimmte Fehlermeldung festlegen.

Verarbeitung der Kopfzeile

Die Kopfzeile einer CSV-Datei kann auf zwei verschiedene Arten verarbeitet werden:

- Wenn das Kontrollkästchen *With header row* aktiviert ist, werden die Werte der Kopfzeile als Spaltennamen interpretiert. In diesem Fall müssen die Spaltennamen eindeutig sein und die Datenzeile darf nicht mehr Spalten enthalten als die Kopfzeile.
- Wenn das Kontrollkästchen *With header row* nicht aktiviert ist, wird die Kopfzeile als Daten interpretiert. Die Spaltennamen werden automatisch generiert (1,2,3,4...).

Beispiel für eine CSV-Datei:

```
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:
Ein Anführungszeichen innerhalb eines in Anführungszeichen gesetzten Feldes in der Eingabe muss maskiert werden, indem ein weiteres Anführungszeichen davor gesetzt wird.

Verarbeitung der Kopfzeile

JSON-Ausgabe, wenn eine Kopfzeile erwartet wird:


```
[
   {
      "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"
   }
]
```


Keine Verarbeitung der Kopfzeile

JSON-Ausgabe, wenn keine Kopfzeile erwartet wird:


```
[
   {
      "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"
   }
]
```


Datentypen

Übersicht

Datenpunkttypen decken verschiedene Methoden zur Erfassung von Daten aus Ihrem System ab. Jeder Datenpunkttyp verfügt über einen eigenen Satz unterstützter Datenpunktschlüssel und erforderlicher Parameter.

Die folgenden Datenpunkttypen werden derzeit von Zabbix angeboten:

- Zabbix Agent
  - Überwachung von Protokolldateien
- Einfache Prüfung
  - VMware-Überwachung
- SNMP Agent
- SNMP trap
- Zabbix intern
- Zabbix trapper
- Externe Prüfung
- Datenbankmonitor
- HTTP Agent
  - Prometheus-Prüfung
- IPMI Agent
- SSH Agent
- Telnet Agent
- JMX Agent
- Berechneter Datenpunkt
- Abhängiger Datenpunkt
- Skript-Datenpunkt
- Browser-Datenpunkt
- Telemetrieabfrage

Details zu allen Datenpunkttypen finden Sie auf den Unterseiten dieses Abschnitts. Obwohl Datenpunkttypen zahlreiche Optionen zur Datenerfassung bieten, gibt es weitere Möglichkeiten über Benutzerparameter oder ladbare Module.

Einige Prüfungen werden ausschließlich vom Zabbix Server durchgeführt (als agentenlose Überwachung), während andere den Zabbix Agent oder sogar das Zabbix Java gateway (bei der JMX-Überwachung) erfordern.

Attention:
Wenn ein bestimmter Datenpunkttyp eine bestimmte Schnittstelle erfordert (z. B. benötigt eine IPMI-Prüfung eine IPMI-Schnittstelle auf dem Host), muss diese Schnittstelle in der Hostdefinition vorhanden sein.

In der Hostdefinition können mehrere Schnittstellen festgelegt werden: Zabbix Agent, SNMP Agent, JMX und IPMI. Wenn ein Datenpunkt mehr als eine Schnittstelle verwenden kann, durchsucht er die verfügbaren Hostschnittstellen (in der Reihenfolge: Agent→SNMP→JMX→IPMI) nach der ersten geeigneten Schnittstelle, mit der er verknüpft werden kann.

Alle Datenpunkte, die Text zurückgeben (Informationstypen „Zeichen“, „Protokoll“ und „Text“), können gegebenenfalls auch ausschließlich Leerzeichen zurückgeben, wobei der Rückgabewert auf eine leere Zeichenfolge gesetzt wird (seit Version 2.0 unterstützt).

Zabbix Agent

Übersicht

Dieser Abschnitt enthält Details zu den Datenpunkt-Schlüsseln, die die Kommunikation mit dem Zabbix Agent zur Datenerfassung verwenden.

Es gibt passive und aktive Agent-Prüfungen. Beim Konfigurieren eines Datenpunkts können Sie den erforderlichen Typ auswählen:

- *Zabbix Agent* - für passive Prüfungen
- *Zabbix Agent (aktiv)* - für aktive Prüfungen

Beachten Sie, dass alle von Zabbix Agent unterstützten Datenpunkt-Schlüssel auch vom neuen Zabbix Agent 2 unterstützt werden. Siehe die zusätzlichen Datenpunkt-Schlüssel, die Sie nur mit Agent 2 verwenden können.

Unterstützte Datenpunktschlüssel

Die Datenpunktschlüssel, die Sie mit dem Zabbix Agent verwenden können, sind unten aufgeführt.

Die Datenpunktschlüssel werden ohne Parameter und zusätzliche Informationen aufgeführt.
Klicken Sie auf den Datenpunktschlüssel, um die vollständigen Details anzuzeigen.

Datenpunktschlüssel	Beschreibung	Datenpunktgruppe
kernel.maxfiles	Die maximale Anzahl geöffneter Dateien, die vom Betriebssystem unterstützt wird.	Kernel
kernel.maxproc	Die maximale Anzahl von Prozessen, die vom Betriebssystem unterstützt wird.	
kernel.openfiles	Die Anzahl der derzeit geöffneten Dateideskriptoren.	
log	Die Überwachung einer Protokolldatei.	Protokollüberwachung
log.count	Die Anzahl der übereinstimmenden Zeilen in einer überwachten Protokolldatei.	
logrt	Die Überwachung einer rotierten Protokolldatei.	
logrt.count	Die Anzahl der übereinstimmenden Zeilen in einer rotierten überwachten Protokolldatei.	
modbus.get	Liest Modbus-Daten.	Modbus
net.dns	Überprüft den Status eines DNS-Dienstes.	Netzwerk
net.dns.perf	Überprüft die Leistung eines DNS-Dienstes.	
net.dns.record	Führt eine DNS-Abfrage aus.	
net.if.collisions	Die Anzahl der Kollisionen außerhalb des Zeitfensters.	
net.if.discovery	Die Liste der Netzwerkschnittstellen.	
net.if.get	Die Liste der Netzwerkschnittstellen mit detaillierten Informationen.	
net.if.in	Die Statistik des eingehenden Datenverkehrs auf einer Netzwerkschnittstelle.	
net.if.out	Die Statistik des ausgehenden Datenverkehrs auf einer Netzwerkschnittstelle.	
net.if.total	Die Summe der Statistiken des eingehenden und ausgehenden Datenverkehrs auf einer Netzwerkschnittstelle.	
net.tcp.listen	Überprüft, ob sich dieser TCP-Port im Status LISTEN befindet.	
net.tcp.port	Überprüft, ob eine TCP-Verbindung zum angegebenen Port hergestellt werden kann.	
net.tcp.service	Überprüft, ob ein Dienst ausgeführt wird und TCP-Verbindungen akzeptiert.	
net.tcp.service.perf	Überprüft die Leistung eines TCP-Dienstes.	
net.tcp.socket.count	Gibt die Anzahl der TCP-Sockets zurück, die den Parametern entsprechen.	
net.udp.listen	Überprüft, ob sich dieser UDP-Port im Status LISTEN befindet.	
net.udp.service	Überprüft, ob ein Dienst ausgeführt wird und auf UDP-Anfragen antwortet.	
net.udp.service.perf	Überprüft die Leistung eines UDP-Dienstes.	
net.udp.socket.count	Gibt die Anzahl der UDP-Sockets zurück, die den Parametern entsprechen.	
proc.cpu.util	Der prozentuale CPU-Auslastungsgrad des Prozesses.	Prozesse
proc.get	Die Liste der Betriebssystemprozesse und ihrer Parameter.	
proc.mem	Der vom Prozess verwendete Speicher in Byte.	
proc.num	Die Anzahl der Prozesse.	
sensor	Messwert eines Hardwaresensors.	Sensoren
system.boottime	Die Systemstartzeit.	System
system.cpu.discovery	Die Liste der erkannten CPUs/CPU-Kerne.	
system.cpu.intr	Die Geräteinterrupts.	
system.cpu.load	Die CPU-Auslastung.	
system.cpu.num	Die Anzahl der CPUs.	
system.cpu.switches	Die Anzahl der Kontextwechsel.	
system.cpu.util	Der prozentuale CPU-Auslastungsgrad.	
system.hostname	Der Hostname des Systems.	
system.hw.chassis	Die Gehäuseinformationen.	
system.hw.cpu	Die CPU-Informationen.	
system.hw.devices	Die Auflistung der PCI- oder USB-Geräte.	
system.hw.macaddr	Die Auflistung der MAC-Adressen.	
system.localtime	Die Systemzeit.	
system.run	Führt den angegebenen Befehl auf dem Host aus.	
system.stat	Die Systemstatistiken.	
system.sw.arch	Die Informationen zur Softwarearchitektur.	
system.sw.os	Die Informationen zum Betriebssystem.	
system.sw.os.get	Detaillierte Informationen zum Betriebssystem (Version, Typ, Distributionsname, Neben- und Hauptversion usw.).	
system.sw.packages	Die Auflistung der installierten Pakete.	
system.sw.packages.get	Eine detaillierte Auflistung der installierten Pakete.	
system.swap.in	Die Statistik des Swap-In (vom Gerät in den Speicher).	
system.swap.out	Die Statistik des Swap-Out (vom Speicher auf das Gerät).	
system.swap.size	Die Größe des Swap-Speichers in Byte oder als Prozentsatz des Gesamtspeichers.	
system.uname	Die Identifikation des Systems.	
system.uptime	Die Systembetriebszeit in Sekunden.	
system.users.num	Die Anzahl der angemeldeten Benutzer.	

Datenpunktschlüssel	Beschreibung	Datenpunktgruppe
vfs.dev.discovery	Die Liste der Blockgeräte und deren Typ.	Virtuelle Dateisysteme
vfs.dev.get	Die Liste der Blockgeräte mit detaillierten Informationen.	
vfs.dev.read	Die Statistik der Datenträgerlesevorgänge.	
vfs.dev.write	Die Statistik der Datenträgerschreibvorgänge.	
vfs.dir.count	Die Anzahl der Verzeichniseinträge.	
vfs.dir.get	Die Liste der Verzeichniseinträge.	
vfs.dir.size	Die Verzeichnisgröße.	
vfs.file.cksum	Die Dateiprüfsumme, berechnet mit dem UNIX-cksum-Algorithmus.	
vfs.file.contents	Das Abrufen des Inhalts einer Datei.	
vfs.file.exists	Überprüft, ob die Datei vorhanden ist.	
vfs.file.get	Gibt Informationen zu einer Datei zurück.	
vfs.file.md5sum	Die MD5-Prüfsumme der Datei.	
vfs.file.owner	Ruft den Besitzer einer Datei ab.	
vfs.file.permissions	Gibt eine 4-stellige Zeichenfolge zurück, die die Oktalzahl mit UNIX-Berechtigungen enthält.	
vfs.file.regexp	Ruft eine Zeichenfolge aus der Datei ab.	
vfs.file.regmatch	Sucht eine Zeichenfolge in der Datei.	
vfs.file.size	Die Dateigröße.	
vfs.file.time	Die Zeitinformationen der Datei.	
vfs.fs.discovery	Die Liste der eingehängten Dateisysteme mit ihrem Typ und ihren Einhängeoptionen.	
vfs.fs.get	Die Liste der eingehängten Dateisysteme mit ihrem Typ und ihren Einhängeoptionen sowie im Modus *full* der verfügbare Speicherplatz und die Inode-Statistiken.	
vfs.fs.inode	Die Anzahl oder der Prozentsatz der Inodes.	
vfs.fs.size	Der Speicherplatz in Byte oder als Prozentsatz des Gesamtspeicherplatzes.	
vm.memory.size	Die Speichergröße in Byte oder als Prozentsatz des Gesamtspeichers.	Virtueller Speicher
web.page.get	Ruft den Inhalt einer Webseite ab.	Webüberwachung
web.page.perf	Die Ladezeit einer vollständigen Webseite.	
web.page.regexp	Sucht eine Zeichenfolge auf der Webseite.	
agent.hostmetadata	Die Hostmetadaten des Agents.	Zabbix
agent.hostname	Der Hostname des Agents.	
agent.ping	Die Verfügbarkeitsprüfung des Agents.	
agent.variant	Die Variante des Zabbix Agents (Zabbix Agent oder Zabbix Agent 2).	
agent.version	Die Version des Zabbix Agents.	
zabbix.stats	Gibt eine Reihe interner Metriken des Zabbix Servers oder Proxys zurück. Wenn `ip` und `port` angegeben sind, werden die Metriken von der entfernten Instanz abgerufen, andernfalls von der lokalen Instanz.	
zabbix.stats (queue)	Gibt die Anzahl der überwachten Datenpunkte in der Warteschlange zurück, deren Verarbeitung auf dem Zabbix Server oder Proxy verzögert ist. Wenn `ip` und `port` angegeben sind, werden die Metriken von der entfernten Instanz abgerufen, andernfalls von der lokalen Instanz.	

Unterstützte Plattformen

Sofern in den Details des Datenpunkts nicht anders angegeben, werden die Agent-Datenpunkte (und alle Parameter) unterstützt auf:

- **Linux**
- **FreeBSD**
- **Solaris**
- **HP-UX**
- **AIX**
- **MacOS X**
- **OpenBSD**
- **NetBSD**

Viele Agent-Datenpunkte werden auch unter **Windows** unterstützt. Siehe die Seite Windows-Agent-Datenpunkt für Details.

Details zum Datenpunktschlüssel

Parameter ohne spitze Klammern sind obligatorisch. Parameter, die mit spitzen Klammern **< >** gekennzeichnet sind, sind optional.

kernel.maxfiles

 Die maximale Anzahl geöffneter Dateien, die vom Betriebssystem unterstützt wird.
 Rückgabewert: *Integer*.
 Unterstützte Plattformen: Linux, FreeBSD, MacOS X, OpenBSD, NetBSD.

kernel.maxproc

 Die maximale Anzahl von Prozessen, die vom Betriebssystem unterstützt wird.
 Rückgabewert: *Integer*.
 Unterstützte Plattformen: Linux 2.6 und höher, FreeBSD, Solaris, MacOS X, OpenBSD, NetBSD.

kernel.openfiles

 Die Anzahl der derzeit offenen Dateideskriptoren.
 Rückgabewert: *Integer*.
 Unterstützte Plattformen: Linux (der Datenpunkt kann auch auf anderen UNIX-ähnlichen Plattformen funktionieren).

log[file,<regexp>,<encoding>,<maxlines>,<mode>,<output>,<maxdelay>,<options>,<persistent dir>]

 Die Überwachung einer Protokolldatei.
 Rückgabewert: *Log*.
 Siehe unterstützte Plattformen.

Parameter:

- **file** - der vollständige Pfad und Name einer Protokolldatei;
- **regexp** - ein regulärer Ausdruck, der das erforderliche Muster beschreibt;
- **encoding** - die Codepage-Kennung;
- **maxlines** - die maximale Anzahl neuer Zeilen pro Sekunde, die der Agent an den Server oder Proxy sendet. Dieser Parameter überschreibt den Wert von `MaxLinesPerSecond` in `zabbix_agentd.conf`;
- **mode** - mögliche Werte:
  - *all* (Standard);
  - *skip* - die Verarbeitung älterer Daten überspringen (wirkt sich nur auf neu erstellte Datenpunkte aus);
- **output** - eine optionale Ausgabeformatierungs-Vorlage. Die Escape-Sequenz **\0** wird durch den übereinstimmenden Teil des Textes ersetzt (vom ersten Zeichen, an dem die Übereinstimmung beginnt, bis zum Zeichen, an dem die Übereinstimmung endet), während **\N** (wobei N=1...9) durch die N-te übereinstimmende Gruppe ersetzt wird (oder durch eine leere Zeichenfolge, wenn N die Anzahl der erfassten Gruppen überschreitet);
- **maxdelay** - die maximale Verzögerung in Sekunden. Typ: float. Werte:
  - 0 - (Standard) Zeilen der Protokolldatei niemals ignorieren;
  - > 0.0 - ältere Zeilen ignorieren, um die aktuellsten Zeilen zu erhalten, die innerhalb von ”maxdelay” Sekunden ausgewertet werden. Lesen Sie vor der Verwendung die Hinweise zu maxdelay;
- **options** - zusätzliche Optionen:
mtime-noreread - nicht eindeutige Einträge, erneut lesen nur, wenn sich die Dateigröße ändert (Änderung der Änderungszeit ignorieren). (Dieser Parameter ist seit 5.0.2 veraltet, da mtime jetzt ignoriert wird.)
- **persistent dir** (nur in zabbix_agentd auf Unix-Systemen; nicht unterstützt in Zabbix agent 2) - der absolute Pfadname des Verzeichnisses, in dem persistente Dateien gespeichert werden. Siehe auch zusätzliche Hinweise zu persistenten Dateien.

Kommentare:

- Der Datenpunkt muss als aktiver Check konfiguriert sein.
- Wenn die Datei fehlt oder Berechtigungen den Zugriff nicht erlauben, wird der Datenpunkt nicht unterstützt.
- Wenn `output` leer gelassen wird, wird die gesamte Zeile zurückgegeben, die den übereinstimmenden Text enthält. Beachten Sie, dass alle globalen regulären Ausdruckstypen außer *Matches regular expression* immer die gesamte übereinstimmende Zeile zurückgeben und der Parameter `output` ignoriert wird.
- Die Inhaltsextraktion mithilfe des Parameters `output` erfolgt auf dem Agent.

Beispiele:

```
log[/var/log/syslog]
log[/var/log/syslog,error]
log[/home/zabbix/logs/logfile,,,100]
```

Beispiel für die Verwendung des Parameters `output` zum Extrahieren einer Zahl aus einem Protokolleintrag:

```
log[/app1/app.log,"task run [0-9.]+ sec, processed ([0-9]+) records, [0-9]+ errors",,,,\1] #this item will
```

Beispiel für die Verwendung des Parameters `output` zum Umschreiben eines Protokolleintrags vor dem Senden an den Server:

```
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>]

 Die Anzahl der übereinstimmenden Zeilen in einer überwachten Protokolldatei.
 Rückgabewert: *Integer*.
 Siehe unterstützte Plattformen.

Parameter:

- **file** - der vollständige Pfad und Dateiname der Protokolldatei;
- **regexp** - ein regulärer Ausdruck, der das erforderliche Muster beschreibt;
- **encoding** - die Codepage-Kennung;
- **maxproclines** - die maximale Anzahl neuer Zeilen pro Sekunde, die der Agent analysiert (darf 10000 nicht überschreiten). Der Standardwert ist 10*MaxLinesPerSecond in zabbix_agentd.conf.
- **mode** - mögliche Werte:
  - *all* (Standard);
  - *skip* - die Verarbeitung älterer Daten überspringen (wirkt sich nur auf neu erstellte Datenpunkte aus).
- **maxdelay** - die maximale Verzögerung in Sekunden. Typ: float. Werte:
  - 0 - (Standard) Protokolldateizeilen niemals ignorieren;
  - > 0.0 - ältere Zeilen ignorieren, um die neuesten Zeilen innerhalb von ”maxdelay” Sekunden analysieren zu lassen. Lesen Sie vor der Verwendung die Hinweise zu maxdelay!
- **options** - zusätzliche Optionen:
mtime-noreread - nicht eindeutige Datensätze, nur erneut lesen, wenn sich die Dateigröße ändert (Änderung der Änderungszeit ignorieren). (Dieser Parameter ist seit 5.0.2 veraltet, da mtime jetzt ignoriert wird.)
- **persistent dir** (nur in zabbix_agentd auf Unix-Systemen; in Zabbix agent 2 nicht unterstützt) - der absolute Pfad des Verzeichnisses, in dem persistente Dateien gespeichert werden. Siehe auch zusätzliche Hinweise zu persistenten Dateien.

Kommentare:

- Der Datenpunkt muss als aktiver Check konfiguriert sein.
- Übereinstimmende Zeilen werden in den neuen Zeilen seit der letzten Protokollprüfung durch den Agent gezählt und hängen daher vom Aktualisierungsintervall des Datenpunkts ab.
- Wenn die Datei fehlt oder Berechtigungen den Zugriff nicht zulassen, wird der Datenpunkt nicht unterstützt.

logrt[file regexp,<regexp>,<encoding>,<maxlines>,<mode>,<output>,<maxdelay>,<options>,<persistent dir>]

 Überwachung einer rotierten Protokolldatei.
 Rückgabewert: *Log*.
 Siehe unterstützte Plattformen.

Parameter:

- **file regexp** - der absolute Pfad zur Datei, wobei der Dateiname mithilfe eines regulären Ausdrucks angegeben wird. Beachten Sie, dass der reguläre Ausdruck nur auf den Dateinamen angewendet wird und nicht mit dem gesamten Namen übereinstimmen muss (z. B. stimmt /path/to/agent mit zabbix_agentd.log überein).
- **regexp** - ein regulärer Ausdruck, der das erforderliche Inhaltsmuster beschreibt.
- **encoding** - der Codepage-Bezeichner.
- **maxlines** - die maximale Anzahl neuer Zeilen pro Sekunde, die der Agent an Server oder Proxy sendet. Dieser Parameter überschreibt den Wert von MaxLinesPerSecond in zabbix_agentd.conf.
- **mode** - mögliche Werte:
  - *all* (Standard);
  - *skip* - die Verarbeitung älterer Daten überspringen (wirkt sich nur auf neu erstellte Datenpunkte aus).
- **output** - eine optionale Vorlage für die Ausgabeformatierung. Die Escape-Sequenz **\0** wird durch den übereinstimmenden Teil des Textes ersetzt (vom ersten Zeichen, an dem die Übereinstimmung beginnt, bis zu dem Zeichen, an dem die Übereinstimmung endet), während **\N** (wobei N=1...9) durch die N-te übereinstimmende Gruppe ersetzt wird (oder durch eine leere Zeichenfolge, wenn N größer als die Anzahl der erfassten Gruppen ist).
- **maxdelay** - die maximale Verzögerung in Sekunden. Typ: float. Werte:
  - 0 - (Standard) Zeilen der Protokolldatei niemals ignorieren;
  - > 0.0 - ältere Zeilen ignorieren, um die aktuellsten Zeilen innerhalb von ”maxdelay” Sekunden auszuwerten. Lesen Sie vor der Verwendung die Hinweise zu maxdelay.
- **options** - der Typ der Rotation der Protokolldatei und weitere Optionen. Mögliche Werte:
  - *rotate* (Standard).
  - *copytruncate* - beachten Sie, dass *copytruncate* nicht zusammen mit *maxdelay* verwendet werden kann. In diesem Fall muss *maxdelay* 0 sein oder darf nicht angegeben werden; siehe Hinweise zu copytruncate.
  - *mtime-reread* - nicht eindeutige Datensätze erneut lesen, wenn sich Änderungszeit oder Größe ändert (wird standardmäßig verwendet, wenn keine mtime-*-Option explizit gesetzt ist).
  - *mtime-noreread* - nicht eindeutige Datensätze erneut lesen, nur wenn sich die Größe ändert (Änderung der Änderungszeit ignorieren).
- **persistent dir** (nur in zabbix_agentd auf Unix-Systemen; in Zabbix agent 2 nicht unterstützt) - der absolute Pfad des Verzeichnisses, in dem persistente Dateien gespeichert werden. Siehe auch zusätzliche Hinweise zu persistenten Dateien.

Kommentare:

- Der Datenpunkt muss als aktiver Check konfiguriert sein.
- Die Rotation der Protokolldatei basiert auf dem Zeitpunkt der letzten Änderung der Dateien.
- Beachten Sie, dass logrt für die Arbeit mit einer aktuell aktiven Protokolldatei vorgesehen ist, während mehrere andere passende inaktive Dateien rotiert werden. Wenn ein Verzeichnis beispielsweise viele aktive Protokolldateien enthält, sollte

für jede einzelne ein separater `logrt`-Datenpunkt erstellt werden. Andernfalls kann es, wenn ein `logrt`-Datenpunkt zu viele Dateien erfasst, zu erschöpftem Speicher und einem Absturz der Überwachung kommen.
- Wenn `output` leer bleibt, wird die gesamte Zeile zurückgegeben, die den übereinstimmenden Text enthält. Beachten Sie, dass alle globalen regulären Ausdruckstypen außer *Matches regular expression* immer die gesamte übereinstimmende Zeile zurückgeben und der Parameter `output` ignoriert wird.
- Die Inhaltsextraktion mithilfe des Parameters `output` erfolgt auf dem Agent.
- Im Parameter `file regexp` müssen der Pfad zum Protokollverzeichnis und der reguläre Ausdruck für die Protokolldatei durch das korrekte Verzeichnistrennzeichen getrennt werden:
  - Unter Windows muss das Trennzeichen ein Backslash (\) sein; Schrägstriche können an anderen Stellen toleriert werden, außer an der Stelle, die den Pfad zum Protokollverzeichnis und den regulären Ausdruck für die Protokolldatei trennt (siehe Beispiele unten).
  - Auf anderen Systemen muss es ein Schrägstrich (/) sein.

Beispiele für Windows:

```
logrt["c:/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #dieser Datenpunkt erfasst Daten aus Protoko
logrt["//example.com/share/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #dieser Datenpunkt erfasst
```

Beispiele für andere Systeme:

```
logrt["/home/zabbix/logs/^logfile[0-9]{1,3}$",,,100] #dieser Datenpunkt stimmt mit einer Datei wie "logfil
logrt["/home/user/^logfile_.*_[0-9]{1,3}$","pattern_to_match","UTF-8",100] #dieser Datenpunkt erfasst Date
```

Beispiel für die Verwendung des Parameters `output` zum Extrahieren einer Zahl aus einem Protokolleintrag:

```
logrt[/app1/^test.*log$,"task run [0-9.]+ sec, processed ([0-9]+) records, [0-9]+ errors",,,,\1] #dieser D
```

Beispiel für die Verwendung des Parameters `output`, um einen Protokolleintrag vor dem Senden an den Server umzuschreiben:

```
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>]

 Die Anzahl der übereinstimmenden Zeilen in einer überwachten rotierenden Protokolldatei.
 Rückgabewert: *Integer*.
 Siehe unterstützte Plattformen.

Parameter:

- **file regexp** - der absolute Pfad zur Datei, wobei der Dateiname mithilfe eines regulären Ausdrucks angegeben wird. Beachten Sie, dass der reguläre Ausdruck nur auf den Dateinamen angewendet wird und nicht mit dem gesamten Namen übereinstimmen muss (z. B. stimmt `/path/to/agent` mit `zabbix_agentd.log` überein).
- **regexp** - ein regulärer Ausdruck, der das erforderliche Muster beschreibt.
- **encoding** - der Codepage-Bezeichner.
- **maxproclines** - die maximale Anzahl neuer Zeilen pro Sekunde, die der Agent analysiert (darf 10000 nicht überschreiten). Der Standardwert ist 10*`MaxLinesPerSecond` in `zabbix_agentd.conf`.
- **mode** - mögliche Werte:
  - *all* (Standard);
  - *skip* - die Verarbeitung älterer Daten überspringen (wirkt sich nur auf neu erstellte Datenpunkte aus).
- **maxdelay** - die maximale Verzögerung in Sekunden. Typ: float. Werte:
  - 0 - (Standard) Zeilen der Protokolldatei niemals ignorieren;
  - > 0.0 - ältere Zeilen ignorieren, um die neuesten Zeilen innerhalb von "maxdelay" Sekunden analysieren zu lassen. Lesen Sie vor der Verwendung die Hinweise zu `maxdelay`.
- **options** - der Typ der Rotation der Protokolldatei und weitere Optionen. Mögliche Werte:
  - *rotate* (Standard).
  - *copytruncate* - beachten Sie, dass `copytruncate` nicht zusammen mit `maxdelay` verwendet werden kann. In diesem Fall muss `maxdelay` 0 sein oder darf nicht angegeben werden; siehe Hinweise zu `copytruncate`.
  - *mtime-reread* - nicht eindeutige Datensätze, erneut lesen, wenn sich Änderungszeit oder Größe ändert (Standard).
  - *mtime-noreread* - nicht eindeutige Datensätze, erneut lesen, nur wenn sich die Größe ändert (Änderung der Änderungszeit ignorieren).
- **persistent dir** (nur in `zabbix_agentd` auf Unix-Systemen; wird in Zabbix Agent 2 nicht unterstützt) - der absolute Pfad des Verzeichnisses, in dem persistente Dateien gespeichert werden. Siehe auch zusätzliche Hinweise zu persistenten Dateien.

Kommentare:

- Der Datenpunkt muss als aktiver Check konfiguriert sein.
- Übereinstimmende Zeilen werden in den neuen Zeilen seit der letzten Protokollprüfung durch den Agent gezählt und hängen daher vom Aktualisierungsintervall des Datenpunkts ab.
- Die Protokollrotation basiert auf der letzten Änderungszeit der Dateien.

- Im Parameter `file regexp` müssen der Pfad des Protokollverzeichnisses und der reguläre Ausdruck der Protokolldatei durch das korrekte Verzeichnistrennzeichen getrennt werden:
  - Unter Windows muss das Trennzeichen ein Backslash (\) sein; Schrägstriche können an anderen Stellen toleriert werden, außer an der Stelle, die den Pfad des Protokollverzeichnisses und den regulären Ausdruck der Protokolldatei trennt (siehe Beispiele unten).
  - Auf anderen Systemen muss es ein Schrägstrich (/) sein.

Beispiele für Windows:

```
logrt.count["c:/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #dieser Datenpunkt zählt die Anzahl de
logrt.count["//example.com/share/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #dieser Datenpunkt zä
```

Beispiele für andere Systeme:

```
logrt.count["/home/zabbix/logs/.*","^logfile[0-9]{1,3}$",,100] #dieser Datenpunkt zählt die Anzahl der Übe
logrt.count["/home/user/^logfile_.*_[0-9]{1,3}$","pattern_to_match","UTF-8",100] #dieser Datenpunkt zählt
```

modbus.get[endpoint,<slave id>,<function>,<address>,<count>,<type>,<endianness>,<offset>]

 Liest Modbus-Daten.
 Rückgabewert: *JSON-String*.
 Unterstützte Plattformen: Linux, Windows.

Parameter:

- **endpoint** - der als `protocol://connection_string` definierte Endpunkt;
- **slave id** - die Slave-ID;
- **function** - die Modbus-Funktion;
- **address** - die Adresse des ersten Registers, Coils oder Inputs;
- **count** - die Anzahl der zu lesenden Datensätze;
- **type** - der Datentyp;
- **endianness** - die Endian-Konfiguration;
- **offset** - die Anzahl der Register, beginnend bei `address`, deren Ergebnisse verworfen werden.

Unter Windows folgt dieser Datenpunkt UNC-Pfaden im Parameter `endpoint`, was ein Sicherheitsrisiko darstellen kann.

Siehe eine detaillierte Beschreibung der Parameter.

net.dns[<ip>,name,<type>,<timeout>,<count>,<protocol>]

 Prüft den Status eines DNS-Dienstes.
 Rückgabewerte: - 0 - DNS-Auflösung fehlgeschlagen (DNS-Server hat nicht geantwortet oder einen Fehler zurückgegeben); - 1 - DNS-Auflösung erfolgreich.
 Siehe unterstützte Plattformen.

Parameter:

- **ip** (unter Windows ignoriert, sofern nicht Zabbix agent 2 verwendet wird) - die IP-Adresse des DNS-Servers (leer lassen für den Standard-DNS-Server);
- **name** - der abzufragende DNS-Name;
- **type** - der abzufragende Record-Typ (Standard ist `SOA`);
- **timeout** (unter Windows ignoriert, sofern nicht Zabbix agent 2 verwendet wird) - das Zeitlimit für die Anfrage in Sekunden (Standard ist 1 Sekunde);
- **count** (unter Windows ignoriert, sofern nicht Zabbix agent 2 verwendet wird) - die Anzahl der Versuche für die Anfrage (Standard ist 2);
- **protocol** - das für DNS-Abfragen verwendete Protokoll: *udp* (Standard) oder *tcp*.

Kommentare:

- Mögliche Werte für `type` sind: *ANY*, *A*, *NS*, *CNAME*, *MB*, *MG*, *MR*, *PTR*, *MD*, *MF*, *MX*, *SOA*, *NULL*, *WKS* (nicht unterstützt für Zabbix agent unter Windows, Zabbix agent 2 auf allen Betriebssystemen), *HINFO*, *MINFO*, *TXT*, *SRV*.
- Für Reverse-DNS-Abfragen (wenn `type` auf *PTR* gesetzt ist) können Sie den DNS-Namen sowohl im umgekehrten als auch im nicht umgekehrten Format angeben (siehe Beispiele unten). Beachten Sie, dass bei einer angeforderten PTR-Record der DNS-Name tatsächlich eine IP-Adresse ist.
- Internationalisierte Domainnamen werden nicht unterstützt, bitte verwenden Sie stattdessen IDNA-kodierte Namen.

Beispiele:

```
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.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR]
```

net.dns.perf[<ip>,name,<type>,<timeout>,<count>,<protocol>]

 Prüft die Leistung eines DNS-Dienstes.
 Rückgabewert: *Float* (0 - Dienst ist ausgefallen; Sekunden - die Anzahl der Sekunden, die auf eine Antwort des Dienstes gewartet wurde).
 Siehe unterstützte Plattformen.

Parameter:

- **ip** (unter Windows ignoriert, sofern nicht Zabbix agent 2 verwendet wird) - die IP-Adresse des DNS-Servers (leer lassen für den Standard-DNS-Server);
- **name** - der abzufragende DNS-Name;
- **type** - der abzufragende Record-Typ (Standard ist `SOA`);
- **timeout** (unter Windows ignoriert, sofern nicht Zabbix agent 2 verwendet wird) - das Timeout für die Anfrage in Sekunden (Standard ist 1 Sekunde);
- **count** (unter Windows ignoriert, sofern nicht Zabbix agent 2 verwendet wird) - die Anzahl der Versuche für die Anfrage (Standard ist 2);
- **protocol** - das für DNS-Abfragen verwendete Protokoll: *udp* (Standard) oder *tcp*.

Kommentare:

- Mögliche Werte für `type` sind:
ANY, *A*, *NS*, *CNAME*, *MB*, *MG*, *MR*, *PTR*, *MD*, *MF*, *MX*, *SOA*, *NULL*, *WKS* (nicht unterstützt für Zabbix agent unter Windows, Zabbix agent 2 auf allen Betriebssystemen), *HINFO*, *MINFO*, *TXT*, *SRV*.
- Für Reverse-DNS-Abfragen (wenn `type` auf *PTR* gesetzt ist) können Sie den DNS-Namen sowohl im umgekehrten als auch im nicht umgekehrten Format angeben (siehe Beispiele unten). Beachten Sie, dass bei einer PTR-Abfrage der DNS-Name tatsächlich eine IP-Adresse ist.
- Internationalisierte Domainnamen werden nicht unterstützt, bitte verwenden Sie stattdessen IDNA-kodierte Namen.
- Der Datenpunkt gibt eine Antwortzeit statt 0 zurück, wenn der DNS-Server mit einem Fehlercode antwortet (zum Beispiel `NXDOMAIN` oder `SERVFAIL`).

Beispiele:

```
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>]

 Führt eine DNS-Abfrage durch.

Zabbix agent 2 stellt außerdem net.dns.get bereit, das zusätzliche Funktionen wie weitere Record-Typen und mehr Kontrolle über das DNS-Monitoring bietet. Rückgabewert: eine Zeichenkette mit dem erforderlichen Informationstyp.
 Siehe unterstützte Plattformen.

Parameter:

- **ip** (unter Windows ignoriert, sofern nicht Zabbix agent 2 verwendet wird) - die IP-Adresse des DNS-Servers (leer lassen für den Standard-DNS-Server);
- **name** - der abzufragende DNS-Name;
- **type** - der abzufragende Record-Typ (Standard ist `SOA`);
- **timeout** (unter Windows ignoriert, sofern nicht Zabbix agent 2 verwendet wird) - das Timeout für die Anfrage in Sekunden (Standard ist 1 Sekunde);
- **count** (unter Windows ignoriert, sofern nicht Zabbix agent 2 verwendet wird) - die Anzahl der Versuche für die Anfrage (Standard ist 2);
- **protocol** - das für DNS-Abfragen verwendete Protokoll: *udp* (Standard) oder *tcp*.

Kommentare:

- Mögliche Werte für `type` sind:
ANY, *A*, *NS*, *CNAME*, *MB*, *MG*, *MR*, *PTR*, *MD*, *MF*, *MX*, *SOA*, *NULL*, *WKS* (unter Zabbix agent unter Windows nicht unterstützt, Zabbix agent 2 auf allen Betriebssystemen), *HINFO*, *MINFO*, *TXT*, *SRV*.
- Für Reverse-DNS-Abfragen (wenn `type` auf *PTR* gesetzt ist) können Sie den DNS-Namen im umgekehrten oder nicht umgekehrten Format angeben (siehe Beispiele unten). Beachten Sie, dass bei einer angeforderten PTR-Record der DNS-Name tatsächlich eine IP-Adresse ist.
- Internationalisierte Domainnamen werden nicht unterstützt, bitte verwenden Sie stattdessen IDNA-kodierte Namen.

Beispiele:

```
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]

 Die Anzahl der Kollisionen außerhalb des Fensters.
 Rückgabewert: *Integer*.
 Unterstützte Plattformen: Linux, FreeBSD, Solaris, AIX, MacOS X, OpenBSD, NetBSD. Auf NetBSD sind Root-Rechte erforderlich.

Parameter:

- **if** - Name der Netzwerkschnittstelle

net.if.discovery

 Die Liste der Netzwerkschnittstellen. Wird für Low-Level-Discovery verwendet.
 Rückgabewert: *JSON-String*.
 Unterstützte Plattformen: Linux, FreeBSD, Solaris, HP-UX, AIX, OpenBSD, NetBSD, Windows.

net.if.get[<regexp>]

 Die Liste der Netzwerk-Schnittstellen mit detaillierten Informationen.
 Wird für die Low-Level-Erkennung verwendet.
 Dieser Datenpunkt ist eine erweiterte Version des Zabbix-Agent-Datenpunkts `net.if.discovery`.
 Rückgabewert: *JSON-Objekt*.
 Unterstützte Plattformen: Linux.

Parameter:

- **regexp** - ein regulärer Ausdruck, der die Namen von Netzwerk-Schnittstellen abgleicht.

Kommentare:

- Gibt ein JSON-Objekt mit Netzwerk-Schnittstellen zurück, deren Namen dem angegebenen regulären Ausdruck entsprechen (siehe weitere Details).

net.if.in[if,<mode>]

 Die Statistik des eingehenden Datenverkehrs auf einer Netzwerkschnittstelle.
 Rückgabewert: *Integer*.
 Unterstützte Plattformen: Linux, FreeBSD, Solaris[5], HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows. Auf NetBSD sind Root-Rechte erforderlich.

Parameter:

- **if** - Name der Netzwerkschnittstelle (Unix); vollständige Beschreibung der Netzwerkschnittstelle oder IPv4-Adresse; oder, in geschweiften Klammern, GUID der Netzwerkschnittstelle (Windows);
- **mode** - mögliche Werte:
  - *bytes* - Anzahl der Bytes (Standard);
  - *packets* - Anzahl der Pakete;
  - *errors* - Anzahl der Fehler;
  - *dropped* - Anzahl der verworfenen Pakete;
  - *overruns (fifo)* - Anzahl der FIFO-Pufferfehler;
  - *frame* - Anzahl der Paket-Frame-Fehler;
  - *compressed* - Anzahl der vom Gerätetreiber empfangenen komprimierten Pakete;
  - *multicast* - Anzahl der vom Gerätetreiber empfangenen Multicast-Frames.

Kommentare:

- Sie können diesen Schlüssel mit dem Vorverarbeitungsschritt *Change per second* verwenden, um Statistiken pro Sekunde in Bytes zu erhalten.
- Der Modus *dropped* wird nur unter Linux, FreeBSD, HP-UX, MacOS X, OpenBSD und NetBSD unterstützt.
- Die Modi *overruns*, *frame*, *compressed* und *multicast* werden nur unter Linux unterstützt.
- Unter HP-UX liefert dieser Datenpunkt keine Details zu Loopback-Schnittstellen (z. B. lo0).

Beispiele:

```
net.if.in[eth0]
net.if.in[eth0,errors]
```

net.if.out[if,<mode>]

 Die Statistik des ausgehenden Datenverkehrs auf einer Netzwerkschnittstelle.
 Rückgabewert: *Integer*.
 Unterstützte Plattformen: Linux, FreeBSD, Solaris[5], HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows. Auf NetBSD sind Root-Rechte erforderlich.

Parameter:

- **if** - Name der Netzwerkschnittstelle (Unix); vollständige Beschreibung der Netzwerkschnittstelle oder IPv4-Adresse; oder, in geschweiften Klammern, GUID der Netzwerkschnittstelle (Windows);
- **mode** - mögliche Werte:
  - *bytes* - Anzahl der Bytes (Standard);
  - *packets* - Anzahl der Pakete;
  - *errors* - Anzahl der Fehler;
  - *dropped* - Anzahl der verworfenen Pakete;
  - *overruns (fifo)* - Anzahl der FIFO-Pufferfehler;
  - *collisions (colls)* - Anzahl der auf der Schnittstelle erkannten Kollisionen;
  - *carrier* - Anzahl der vom Gerätetreiber erkannten Trägerverluste;
  - *compressed* - Anzahl der vom Gerätetreiber übertragenen komprimierten Pakete.

Kommentare:

- Sie können diesen Schlüssel mit dem Vorverarbeitungsschritt *Change per second* verwenden, um die Statistik der Bytes pro Sekunde zu erhalten.
- Der Modus *dropped* wird nur unter Linux und HP-UX unterstützt.
- Die Modi *overruns*, *collision*, *carrier*, *compressed* werden nur unter Linux unterstützt.
- Unter HP-UX liefert dieser Datenpunkt keine Details zu Loopback-Schnittstellen (z. B. lo0).

Beispiele:

```
net.if.out[eth0]
net.if.out[eth0,errors]
```

net.if.total[if,<mode>]

 Die Summe der eingehenden und ausgehenden Verkehrsdaten auf einer Netzwerkschnittstelle.
 Rückgabewert: *Integer*.
 Unterstützte Plattformen: Linux, FreeBSD, Solaris[5], HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows. Auf NetBSD sind Root-Rechte erforderlich.

Parameter:

- **if** - Name der Netzwerkschnittstelle (Unix); vollständige Beschreibung der Netzwerkschnittstelle oder IPv4-Adresse; oder, wenn in geschweiften Klammern angegeben, GUID der Netzwerkschnittstelle (Windows);
- **mode** - mögliche Werte:
  - *bytes* - Anzahl der Bytes (Standard);
  - *packets* - Anzahl der Pakete;
  - *errors* - Anzahl der Fehler;
  - *dropped* - Anzahl der verworfenen Pakete;
  - *overruns (fifo)* - Anzahl der FIFO-Pufferfehler;
  - *collisions (colls)* - Anzahl der auf der Schnittstelle erkannten Kollisionen;
  - *compressed* - Anzahl der komprimierten Pakete, die vom Gerätetreiber gesendet oder empfangen wurden.

Kommentare:

- Sie können diesen Schlüssel mit dem Vorverarbeitungsschritt *Change per second* verwenden, um Statistiken zu Bytes pro Sekunde zu erhalten.
- Der Modus *dropped* wird nur unter Linux, HP-UX unterstützt. Verworfene Pakete werden nur unterstützt, wenn sowohl `net.if.in` als auch `net.if.out` auf Ihrer Plattform für verworfene Pakete funktionieren.
- Die Modi *overruns*, *collision*, *compressed* werden nur unter Linux unterstützt.
- Unter HP-UX liefert dieser Datenpunkt keine Details zu Loopback-Schnittstellen (z. B. lo0).

Beispiele:

```
net.if.total[eth0]
net.if.total[eth0,errors]
```

net.tcp.listen[port]

 Prüft, ob sich dieser TCP-Port im LISTEN-Zustand befindet.
 Rückgabewerte: 0 - er befindet sich nicht im LISTEN-Zustand; 1 - er befindet sich im LISTEN-Zustand.
 Unterstützte Plattformen: Linux, FreeBSD, Solaris, MacOS X, Windows.

Parameter:

- **port** - TCP-Portnummer

Bei Linux-Kernels ab Version 2.6.14 werden Informationen über lauschende TCP-Sockets, sofern möglich, über die NETLINK-Schnittstelle des Kernels ermittelt. Andernfalls werden die Informationen aus den Dateien `/proc/net/tcp` und `/roc/net/tcp6` abgerufen.

Beispiel:

```
net.tcp.listen[80]
```

net.tcp.port[<ip>,port]

 Prüft, ob eine TCP-Verbindung zum angegebenen Port hergestellt werden kann.
 Rückgabewerte: 0 - keine Verbindung möglich; 1 - Verbindung möglich.
 Siehe unterstützte Plattformen.

Parameter:

- **ip** - die IP-Adresse oder der DNS-Name (Standard ist `127.0.0.1`);
- **port** - die Portnummer.

Kommentare:

- Für einfache TCP-Leistungstests verwenden Sie `net.tcp.service.perf[tcp,<ip>,<port>]`.
- Diese Prüfungen können zu zusätzlichen Meldungen in den Protokolldateien des System-Daemons führen (SMTP- und SSH-Sitzungen werden normalerweise protokolliert).

Beispiel:

```
net.tcp.port[,80] #dieser Datenpunkt kann verwendet werden, um die Verfügbarkeit des auf Port 80 laufenden
```

net.tcp.service[service,<ip>,<port>]

 Prüft, ob ein Dienst läuft und TCP-Verbindungen annimmt.
 Rückgabewerte: 0 - Dienst ist nicht verfügbar; 1 - Dienst läuft.
 Siehe unterstützte Plattformen.

Parameter:

- **service** - *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https* oder *telnet* (siehe Details);
- **ip** - die IP-Adresse oder der DNS-Name (standardmäßig `127.0.0.1`);
- **port** - die Portnummer (standardmäßig wird die übliche Portnummer des Dienstes verwendet).

Kommentare:

- Diese Prüfungen können zu zusätzlichen Meldungen in den Protokolldateien des System-Daemons führen (SMTP- und SSH-Sitzungen werden normalerweise protokolliert).
- Die Prüfung verschlüsselter Protokolle (wie IMAP auf Port 993 oder POP auf Port 995) wird derzeit nicht unterstützt. Als Workaround verwenden Sie bitte `net.tcp.port[]` für solche Prüfungen.
- Die Prüfung von LDAP und HTTPS unter Windows wird nur von Zabbix Agent 2 unterstützt.
- Die Telnet-Prüfung sucht nach einer Anmeldeaufforderung (':' am Ende).

Beispiel:

```
net.tcp.service[ftp,,45] #dieser Datenpunkt kann verwendet werden, um die Verfügbarkeit des FTP-Servers au
```

net.tcp.service.perf[service,<ip>,<port>]

 Prüft die Leistung eines TCP-Dienstes.
 Rückgabewerte: *Float* (0 - der Dienst ist nicht verfügbar; Sekunden - die Anzahl der Sekunden, die auf eine Antwort des Dienstes gewartet wurde).
 Siehe unterstützte Plattformen.

Parameter:

- **service** - *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https* oder *telnet* (siehe Details);
- **ip** - die IP-Adresse oder der DNS-Name (standardmäßig `127.0.0.1`);
- **port** - die Portnummer (standardmäßig wird die Standard-Portnummer des Dienstes verwendet).

Hinweise:

- Die Prüfung verschlüsselter Protokolle (wie IMAP auf Port 993 oder POP auf Port 995) wird derzeit nicht unterstützt. Als Workaround verwenden Sie bitte `net.tcp.service.perf[tcp,<ip>,<port>]` für solche Prüfungen.
- Die Telnet-Prüfung sucht nach einer Login-Aufforderung (':' am Ende).

Beispiel:

```
net.tcp.service.perf[ssh] #dieser Datenpunkt kann verwendet werden, um die Geschwindigkeit der ersten Antw
```

net.tcp.socket.count[<laddr>,<lport>,<raddr>,<rport>,<state>]

 Gibt die Anzahl der TCP-Sockets zurück, die den Parametern entsprechen.
 Rückgabewert: *Integer*.
 Unterstützte Plattformen: Linux, Windows.

Parameter:

- **laddr** - die lokale IPv4/6-Adresse oder das CIDR-Subnetz;
- **lport** - die lokale Portnummer oder der Dienstname;
- **raddr** - die entfernte IPv4/6-Adresse oder das CIDR-Subnetz;
- **rport** - die entfernte Portnummer oder der Dienstname;
- **state** - der Verbindungsstatus (*established*, *syn_sent*, *syn_recv*, *fin_wait1*, *fin_wait2*, *time_wait*, *close*, *close_wait*, *last_ack*, *listen*, *closing*).

Beispiel:

```
net.tcp.socket.count[,80,,,established] #die Anzahl der Verbindungen zu lokalem TCP-Port 80 im Status esta
```

net.udp.listen[port]

 Prüft, ob sich dieser UDP-Port im Status LISTEN befindet.
 Rückgabewerte: 0 - er befindet sich nicht im Status LISTEN; 1 - er befindet sich im Status LISTEN.
 Unterstützte Plattformen: Linux, FreeBSD, Solaris, MacOS X.

Parameter:

- **port** - UDP-Portnummer

Beispiel:

```
net.udp.listen[68]
```

net.udp.service[service,<ip>,<port>]

 Prüft, ob ein Dienst ausgeführt wird und auf UDP-Anfragen antwortet.
 Rückgabewerte: 0 - Dienst ist nicht verfügbar; 1 - Dienst wird ausgeführt.
 Siehe unterstützte Plattformen.

Parameter:

- **service** - *ntp* (siehe Details);
- **ip** - die IP-Adresse oder der DNS-Name (standardmäßig `127.0.0.1`);
- **port** - die Portnummer (standardmäßig wird die Standard-Portnummer des Dienstes verwendet).

Beispiel:

```
net.udp.service[ntp,,45] #this item can be used to test the availability of NTP service on UDP port 45
```

net.udp.service.perf[service,<ip>,<port>]

 Prüft die Leistung eines UDP-Dienstes.
 Rückgabewerte: *Float* (0 - der Dienst ist ausgefallen; Sekunden - die Anzahl der Sekunden, die auf eine Antwort des Dienstes gewartet wurde).
 Siehe unterstützte Plattformen.

Parameter:

- **service** - *ntp* (siehe Details);
- **ip** - die IP-Adresse oder der DNS-Name (standardmäßig `127.0.0.1`);
- **port** - die Portnummer (standardmäßig wird die Standard-Portnummer des Dienstes verwendet).

Beispiel:

```
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>]

 Gibt die Anzahl der UDP-Sockets zurück, die den Parametern entsprechen.
 Rückgabewert: *Integer*.
 Unterstützte Plattformen: Linux, Windows.

Parameter:

- **laddr** - die lokale IPv4/6-Adresse oder das CIDR-Subnetz;
- **lport** - die lokale Portnummer oder der Dienstname;
- **raddr** - die entfernte IPv4/6-Adresse oder das CIDR-Subnetz;
- **rport** - die entfernte Portnummer oder der Dienstname;
- **state** - der Verbindungsstatus (*established*, *unconn*).

Beispiel:

```
net.udp.socket.count[,,,,established] #gibt die Anzahl der UDP-Sockets im verbundenen Zustand zurück
```

proc.cpu.util[<name>,<user>,<type>,<cmdline>,<mode>,<zone>]

 Der Prozentsatz der CPU-Auslastung des Prozesses.
 Rückgabewert: *Float*.
 Unterstützte Plattformen: Linux, Solaris[6].

Parameter:

- **name** - der Prozessname (Standard ist *alle Prozesse*);
- **user** - der Benutzername (Standard ist *alle Benutzer*);
- **type** - der Typ der CPU-Auslastung: *total* (Standard), *user* oder *system*;
- **cmdline** - Filter nach Befehlszeile (es handelt sich um einen regulären Ausdruck);
- **mode** - der Modus der Datenerfassung: *avg1* (Standard), *avg5* oder *avg15*;
- **zone** - die Zielzone: *current* (Standard) oder *all*. Dieser Parameter wird nur unter Solaris unterstützt.

Kommentare:

- Der zurückgegebene Wert basiert auf dem Prozentsatz der Auslastung eines einzelnen CPU-Kerns. Zum Beispiel beträgt die CPU-Auslastung eines Prozesses, der zwei Kerne vollständig nutzt, 200%.
- Die Daten zur CPU-Auslastung des Prozesses werden von einem Collector erfasst, der maximal 1024 eindeutige Abfragen (nach Name, Benutzer und Befehlszeile) unterstützt. Abfragen, auf die in den letzten 24 Stunden nicht zugegriffen wurde, werden aus dem Collector entfernt.
- Wenn der Parameter `zone` auf *current* (oder standardmäßig) gesetzt wird, obwohl der Agent auf einem Solaris ohne Zonenunterstützung kompiliert wurde, aber auf einem neueren Solaris mit Zonenunterstützung ausgeführt wird, gibt der Agent NOTSUPPORTED zurück (der Agent kann die Ergebnisse nicht auf die aktuelle Zone beschränken). Allerdings wird *all* in diesem Fall unterstützt.

Beispiele:

```
proc.cpu.util[,root] #CPU-Auslastung aller Prozesse, die unter dem Benutzer "root" ausgeführt werden
proc.cpu.util[zabbix_server,zabbix] #CPU-Auslastung aller zabbix_server-Prozesse, die unter dem Benutzer z
```

proc.get[<name>,<user>,<cmdline>,<mode>]

 Die Liste der Betriebssystemprozesse und ihrer Parameter. Kann für die Low-Level-Discovery verwendet werden.
 Rückgabewert: *JSON-String*.
 Unterstützte Plattformen: Linux, FreeBSD, Windows, OpenBSD, NetBSD.

Parameter:

- **name** - der Prozessname (Standard *alle Prozesse*).
- **user** - der Benutzername (Standard *alle Benutzer*).
- **cmdline** - Filterung nach Befehlszeile (es handelt sich um einen regulären Ausdruck). Dieser Parameter wird unter Windows nicht unterstützt; auf anderen Plattformen wird er nicht unterstützt, wenn mode auf 'summary' gesetzt ist.
- **mode** - mögliche Werte:
  - *process* (Standard);
  - *thread* (unter NetBSD nicht unterstützt);
  - *summary*. Siehe eine Liste der für jeden Modus und jedes Betriebssystem zurückgegebenen Prozessparameter.

Kommentare:

- Wenn ein Wert nicht abgerufen werden kann, zum Beispiel aufgrund eines Fehlers (Prozess bereits beendet, fehlende Berechtigungen, Fehler bei einem Systemaufruf), wird −1 zurückgegeben.
- Siehe Hinweise zur Auswahl von Prozessen mit den Parametern `name` und `cmdline` (nur für Linux).

Beispiele:

```
proc.get[zabbix_server,zabbix,,process] #Liste aller zabbix_server-Prozesse, die unter dem Benutzer zabbix
proc.get[java,,,thread] #Liste aller Java-Prozesse, gibt einen Eintrag pro Thread zurück
proc.get[,zabbix,,summary] #zusammengefasste Daten für Prozesse jedes Typs, die unter dem Benutzer zabbix
```

proc.mem[<name>,<user>,<mode>,<cmdline>,<memtype>]

 Der vom Prozess verwendete Speicher in Bytes.
 Rückgabewert: - *Integer* - bei `mode` als *max*, *min*, *sum*; - *Float* - bei `mode` als *avg*
 Unterstützte Plattformen: Linux, FreeBSD, Solaris, AIX, OpenBSD, NetBSD.

Parameter:

- **name** - der Prozessname (Standard ist *alle Prozesse*);
- **user** - der Benutzername (Standard ist *alle Benutzer*);
- **mode** - mögliche Werte: *avg*, *max*, *min* oder *sum* (Standard);
- **cmdline** - Filter nach Befehlszeile (es handelt sich um einen regulären Ausdruck);
- **memtype** - der von dem Prozess verwendete Speichertyp

Kommentare:

- Der Parameter `memtype` wird nur unter Linux, FreeBSD, Solaris[6], AIX unterstützt.
- Wenn mehrere Prozesse gemeinsam genutzten Speicher verwenden, kann die Summe des von Prozessen verwendeten Speichers zu großen, unrealistischen Werten führen.

Siehe Hinweise zur Auswahl von Prozessen mit den

Parametern `name` und `cmdline` (nur für Linux).

Wenn dieser Datenpunkt über die Befehlszeile aufgerufen wird und einen Befehlszeilenparameter enthält (z. B. bei Verwendung des Agent-Testmodus: `zabbix_agentd -t proc.mem[,,,apache2]`), wird ein zusätzlicher Prozess gezählt, da der Agent sich selbst mitzählt.

Beispiele:

```
proc.mem[,root] #der von allen unter dem Benutzer "root" laufenden Prozessen verwendete Speicher
proc.mem[zabbix_server,zabbix] #der von allen zabbix_server-Prozessen verwendete Speicher, die unter dem B
proc.mem[,oracle,max,oracleZABBIX] #der von dem speicherintensivsten Prozess verwendete Speicher, der unte
```

proc.num[<name>,<user>,<state>,<cmdline>,<zone>]

 Die Anzahl der Prozesse.
 Rückgabewert: *Integer*.
 Unterstützte Plattformen: Linux, FreeBSD, Solaris[6], HP-UX, AIX, OpenBSD, NetBSD, Windows.

Parameter:

- **name** - der Prozessname (Standard ist *alle Prozesse*).
- **user** - der Benutzername (Standard ist *alle Benutzer*).
- **state** - mögliche Werte:
  - *all* (Standard);
  - *disk* - nicht unterbrechbarer Schlaf;
  - *run* - läuft;
  - *sleep* - unterbrechbarer Schlaf;
  - *trace* - angehalten;
  - *zomb* - Zombie.
- **cmdline** - Filter nach Befehlszeile (es handelt sich um einen regulären Ausdruck).
- **zone** - die Zielzone: *current* (Standard) oder *all*. Dieser Parameter wird nur unter Solaris unterstützt.

Kommentare:

- Die Statusparameter *disk* und *trace* werden nur unter Linux, FreeBSD, OpenBSD und NetBSD unterstützt.
- Unter Windows werden nur die Parameter `name` und `user` unterstützt.
- Wenn dieser Datenpunkt über die Befehlszeile aufgerufen wird und einen Befehlszeilenparameter enthält (z. B. bei Verwendung des Agent-Testmodus: `zabbix_agentd -t proc.num[,,,apache2]`), wird ein zusätzlicher Prozess gezählt, da der Agent sich selbst mitzählt.
- Wenn der Parameter `zone` auf `current` gesetzt wird (oder standardmäßig verwendet wird) und der Agent auf einem Solaris ohne Zonenunterstützung kompiliert wurde, aber auf einem neueren Solaris mit Zonenunterstützung ausgeführt wird, gibt der Agent NOTSUPPORTED zurück (der Agent kann die Ergebnisse nicht auf die aktuelle Zone beschränken). *all* wird in diesem Fall jedoch unterstützt.
- Siehe Hinweise zur Auswahl von Prozessen mit den Parametern `name` und `cmdline` (nur für Linux).

Beispiele:

```
proc.num[,mysql] #die Anzahl der Prozesse, die unter dem Benutzer mysql ausgeführt werden
proc.num[apache2,www-data] #die Anzahl der apache2-Prozesse, die unter dem Benutzer www-data ausgeführt we
proc.num[,oracle,sleep,oracleZABBIX] #die Anzahl der Prozesse im sleep-Zustand, die unter Oracle ausgeführ
```

sensor[device,sensor,<mode>]

 Hardware-Sensorwert.
 Rückgabewert: *Float*.
 Unterstützte Plattformen: Linux, OpenBSD.

Parameter:

- **device** - der Gerätename;
- **sensor** - der Sensorname;
- **mode** - mögliche Werte: *avg*, *max* oder *min*.

Kommentare:

- Liest `/proc/sys/dev/sensors` unter Linux 2.4.
- Liest `/sys/class/hwmon` unter Linux 2.6+. Siehe eine ausführlichere Beschreibung des sensor-Datenpunkts unter Linux.
- Liest die `hw.sensors`-MIB unter OpenBSD.
- Unter Linux 2.4 werden device und sensor als reguläre Ausdrücke behandelt, wenn Sie mode angeben.
- Unter Linux 2.6+ und OpenBSD oder wenn mode weggelassen wird, werden device und sensor wörtlich behandelt.

Beispiele:

```
sensor[w83781d-i2c-0-2d,temp1]
sensor[cpu0,temp0] #die Temperatur einer CPU
sensor["cpu[0-2]$",temp,avg] #die durchschnittliche Temperatur der ersten drei CPUs
```

system.boottime

 Die Systemstartzeit.
 Rückgabewert: *Integer (Unix-Zeitstempel)*.
 Unterstützte Plattformen: Linux, FreeBSD, Solaris, MacOS X, OpenBSD, NetBSD.

system.cpu.discovery

 Die Liste der erkannten CPUs/CPU-Kerne. Wird für Low-Level-Discovery verwendet.
 Rückgabewert: *JSON-String*.
 Siehe unterstützte Plattformen.

system.cpu.intr

 Die Geräte-Interrupts.
 Rückgabewert: *Integer*.
 Unterstützte Plattformen: Linux, FreeBSD, Solaris, AIX, OpenBSD, NetBSD.

system.cpu.load[<cpu>,<mode>]

 Die CPU-Last.
 Rückgabewert: *Float*.
 Siehe unterstützte Plattformen.

Parameter:

- **cpu** – mögliche Werte: *all* (Standard) oder *percpu* (die Gesamtlast geteilt durch die Anzahl der online befindlichen CPUs);
- **mode** – mögliche Werte: *avg1* (Ein-Minuten-Durchschnitt, Standard), *avg5* oder *avg15*.

Beispiel:

```
system.cpu.load[,avg5]
```

system.cpu.num[<type>]

 Die Anzahl der CPUs.
 Rückgabewert: *Integer*.
 Unterstützte Plattformen: Linux, FreeBSD, Solaris, HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows.

Parameter:

- **type** - mögliche Werte: *online* (Standard) oder *max*

Der Parameter *max* wird nur unter Linux, FreeBSD, Solaris und MacOS X unterstützt.

Beispiel:

```
system.cpu.num
```

system.cpu.switches

 Die Anzahl der Kontextwechsel.
 Rückgabewert: *Integer*.
 Unterstützte Plattformen: Linux, FreeBSD, Solaris, AIX, OpenBSD, NetBSD.

system.cpu.util[<cpu>,<type>,<mode>,<logical or physical>]

 Die CPU-Auslastung in Prozent.
 Rückgabewert: *Float*.
 Unterstützte Plattformen: Linux, FreeBSD, Solaris, HP-UX, AIX, OpenBSD, NetBSD, Windows.

Parameter:

- **cpu** - *<CPU-Nummer>* oder *all* (Standard);
- **type** - mögliche Werte: *user* (Standard), *idle*, *nice*, *system*, *iowait*, *interrupt*, *softirq*, *steal*, *spin* (unter OpenBSD), *guest* (unter Linux-Kernels 2.6.24 und höher) oder *guest_nice* (unter Linux-Kernels 2.6.33 und höher);
- **mode** - mögliche Werte: *avg1* (Ein-Minuten-Durchschnitt, Standard), *avg5* oder *avg15*;
- **logical or physical** - mögliche Werte: *logical* (Standard) oder *physical*. Dieser Parameter wird nur unter AIX unterstützt.

Kommentare:

- Der Typparameter *nice* wird nur unter Linux, FreeBSD, HP-UX, OpenBSD und NetBSD unterstützt.
- Der Typparameter *iowait* wird nur unter Linux 2.6 und höher, Solaris und AIX unterstützt.
- Der Typparameter *interrupt* wird nur unter Linux 2.6 und höher, FreeBSD und OpenBSD unterstützt.
- Die Typparameter *softirq*, *steal*, *guest*, *guest_nice* werden nur unter Linux 2.6 und höher unterstützt.
- Die Modusparameter *avg5* und *avg15* werden unter Linux, FreeBSD, Solaris, HP-UX, AIX, OpenBSD und NetBSD unterstützt.

Beispiel:

```
system.cpu.util[0,user,avg5]
```

system.hostname[<type>,<transform>]

 Der System-Hostname.
 Rückgabewert: *String*.
 Siehe unterstützte Plattformen.

Parameter:

- **type** - mögliche Werte:
  - *netbios* (Standard unter Windows);
  - *host* (Standard unter Linux);
  - *shorthost* (gibt den Teil des Hostnamens vor dem ersten Punkt zurück, bei Namen ohne Punkte den vollständigen String);
  - *fqdn* (gibt den Fully Qualified Domain Name zurück).
- **transform** - mögliche Werte:
  - *none* (Standard);
  - *lower* (in Kleinbuchstaben umwandeln).

Der Wert wird ermittelt, indem `nodename` aus der System-API-Ausgabe von uname() verwendet wird.

Beispiele für zurückgegebene Werte:

```
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>]

 Die Gehäuseinformationen.
 Rückgabewert: *String*.
 Unterstützte Plattformen: Linux.

Parameter:

- **info** – mögliche Werte: *full* (Standard), *model*, *serial*, *type* oder *vendor*

Kommentare:

- Dieser Datenpunktschlüssel hängt von der Verfügbarkeit der SMBIOS-Tabelle ab.
- Es wird versucht, die DMI-Tabelle aus sysfs zu lesen; falls der Zugriff auf sysfs fehlschlägt, wird versucht, direkt aus dem Speicher zu lesen.
- **Root-Berechtigungen** sind erforderlich, da der Wert durch Lesen aus sysfs oder dem Speicher ermittelt wird.

Beispiel:

```
system.hw.chassis[full] → Hewlett-Packard HP Pro 3010 Small Form Factor PC CZXXXXXXXX Desktop
```

system.hw.cpu[<cpu>,<info>]

 Die CPU-Informationen.
 Rückgabewert: *String* oder *Integer*.
 Unterstützte Plattformen: Linux.

Parameter:

- **cpu** - *<CPU-Nummer>* oder *all* (Standard);
- **info** - mögliche Werte: *full* (Standard), *curfreq*, *maxfreq*, *model* oder *vendor*.

Kommentare:

- Sammelt Informationen aus `/proc/cpuinfo` und `/sys/devices/system/cpu/[cpunum]/cpufreq/cpuinfo_max_freq`.
- Wenn eine CPU-Nummer und *curfreq* oder *maxfreq* angegeben ist, wird ein numerischer Wert zurückgegeben (Hz).

Beispiel:

```
system.hw.cpu[0,vendor] → AuthenticAMD
```

system.hw.devices[<type>]

 Die Auflistung der PCI- oder USB-Geräte.
 Rückgabewert: *Text*.
 Unterstützte Plattformen: Linux.

Parameter:

- **type** - *pci* (Standard) oder *usb*

Gibt die Ausgabe des Dienstprogramms lspci bzw. lsusb zurück (jeweils ohne Parameter ausgeführt).

Beispiel:

```
system.hw.devices → 00:00.0 Host bridge: Advanced Micro Devices [AMD] RS780 Host Bridge
```

system.hw.macaddr[<interface>,<format>]

 Die Auflistung der MAC-Adressen.
 Rückgabewert: *String*.
 Unterstützte Plattformen: Linux.

Parameter:

- **interface** - *all* (Standard) oder ein regulärer Ausdruck;
- **format** - *full* (Standard) oder *short*.

Kommentare:

- Listet die MAC-Adressen der Schnittstellen auf, deren Name dem angegebenen regulären Ausdruck `interface` entspricht (*all* listet alle Schnittstellen auf).
- Wenn `format` als *short* angegeben ist, werden Schnittstellennamen und identische MAC-Adressen nicht aufgelistet.

Beispiel:

```
system.hw.macaddr["eth0$",full] → [eth0] 00:11:22:33:44:55
```

system.localtime[<type>]

 Die Systemzeit.
 Rückgabewert: *Integer* - bei `type` = *utc*; *String* - bei `type` = *local*.
 Siehe unterstützte Plattformen.

Parameter:

- **type** - mögliche Werte:
  - *utc* - (Standard) die Zeit seit der Unix-Epoche (00:00:00 UTC, 1. Januar 1970), gemessen in Sekunden;
  - *local* - die Zeit im Format 'yyyy-mm-dd,hh:mm:ss.nnn,+hh:mm'.

Muss für Zabbix Agent als passive Prüfung verwendet werden; bei Zabbix agent 2 kann sie als aktive Prüfung konfiguriert werden.

Beispiel:

```
system.localtime[local] #create an item using this key and then use it to display the host time in the *Cl
```

system.run[command,<mode>]

 Führt den angegebenen Befehl auf dem Host aus.
 Rückgabewert: *Text*-Ergebnis des Befehls oder 1 - bei `mode` als *nowait* (unabhängig vom Ergebnis des Befehls).
 Siehe unterstützte Plattformen.

Parameter:

- **command** - Befehl zur Ausführung;
- **mode** - mögliche Werte:
  - *wait* - bis zum Ende der Ausführung warten (Standard);
  - *nowait* - nicht warten.

Kommentare:

- Dieser Datenpunkt ist standardmäßig deaktiviert. Erfahren Sie, wie Sie ihn aktivieren.
- Der Rückgabewert des Datenpunkts ist die Standardausgabe zusammen mit der Standardfehlerausgabe, die vom Befehl erzeugt wird. Prüfung des Exit-Codes wird nicht durchgeführt.
- Damit die Verarbeitung korrekt erfolgt, muss der Rückgabewert des Befehls vom Datentyp `text` sein. Ein leeres Ergebnis ist ebenfalls zulässig.
- Der Rückgabewert ist auf 16 MB begrenzt (einschließlich nachfolgender Leerzeichen, die abgeschnitten werden); Datenbankgrenzen gelten ebenfalls.
- Siehe auch: Befehlsausführung.

Beispiel:

```
system.run[ls -l /] #gibt eine detaillierte Dateiliste des Stammverzeichnisses zurück
```

system.stat[resource,<type>]

 Die Systemstatistiken.
 Rückgabewert: *Integer* oder *float*.
 Unterstützte Plattformen: AIX.

Parameter:

- **ent** - die Anzahl der Prozessor-Einheiten, die diese Partition erhalten darf (float).
- **kthr,<type>** - Informationen über Kernel-Thread-Zustände:
  - *r* - durchschnittliche Anzahl ausführbarer Kernel-Threads (float);
  - *b* - durchschnittliche Anzahl von Kernel-Threads in der Wartewarteschlange des Virtual Memory Managers (float).
- **memory,<type>** - Informationen zur Nutzung von virtuellem und realem Speicher:
  - *avm* - aktive virtuelle Seiten (integer);
  - *fre* - Größe der freien Liste (integer).
- **page,<type>** - Informationen über Page Faults und Paging-Aktivität:
  - *fi* - Datei-Page-Ins pro Sekunde (float);
  - *fo* - Datei-Page-Outs pro Sekunde (float);

  - *pi* - Seiten, die aus dem Paging-Speicher eingelagert wurden (float);
  - *po* - Seiten, die in den Paging-Speicher ausgelagert wurden (float);
  - *fr* - freigegebene Seiten (Seitenersetzung) (float);
  - *sr* - von der Seitenersetzungs-Algorithmus gescannte Seiten (float).
- **faults,<type>** - Rate von Traps und Interrupts:
  - *in* - Geräteinterrupts (float);
  - *sy* - Systemaufrufe (float);
  - *cs* - Kontextwechsel von Kernel-Threads (float).
- **cpu,<type>** - Aufschlüsselung der prozentualen Nutzung der Prozessorzeit:
  - *us* - Benutzerzeit (float);
  - *sy* - Systemzeit (float);
  - *id* - Leerlaufzeit (float);
  - *wa* - Leerlaufzeit, während der das System ausstehende Festplatten-/NFS-I/O-Anforderungen hatte (float);
  - *pc* - Anzahl der verbrauchten physischen Prozessoren (float);
  - *ec* - der Prozentsatz der verbrauchten zugewiesenen Kapazität (float);
  - *lbusy* - gibt den Prozentsatz der Auslastung logischer Prozessor(en) an, der bei Ausführung auf Benutzer- und Systemebene auftrat (float);
  - *app* - gibt die verfügbaren physischen Prozessoren im gemeinsam genutzten Pool an (float).
- **disk,<type>** - Festplattenstatistiken:
  - *bps* - gibt die Menge der an das Laufwerk übertragenen Daten (gelesen oder geschrieben) in Bytes pro Sekunde an (integer);
  - *tps* - gibt die Anzahl der pro Sekunde an das physische Festplatten-/Bandlaufwerk ausgegebenen Übertragungen an (float).

Kommentare:

- Beachten Sie die folgenden Einschränkungen bei diesen Datenpunkten:
  - `system.stat[cpu,app]` - nur auf AIX LPAR vom Typ ”Shared” unterstützt;
  - `system.stat[cpu,ec]` - auf AIX LPAR vom Typ ”Shared” und ”Dedicated” unterstützt (”Dedicated” gibt immer 100 (Prozent) zurück);
  - `system.stat[cpu,lbusy]` - nur auf AIX LPAR vom Typ ”Shared” unterstützt;
  - `system.stat[cpu,pc]` - auf AIX LPAR vom Typ ”Shared” und ”Dedicated” unterstützt;
  - `system.stat[ent]` - auf AIX LPAR vom Typ ”Shared” und ”Dedicated” unterstützt.

system.sw.arch

 Die Informationen zur Softwarearchitektur.
 Rückgabewert: *String*.
 Siehe unterstützte Plattformen.

Die Informationen werden über die Funktion `uname()` ermittelt.

Beispiel:

```
system.sw.arch → i686
```

system.sw.os[<info>]

 Informationen zum Betriebssystem.
 Rückgabewert: *String*.
 Unterstützte Plattformen: Linux, Windows.

Parameter:

- **info** - mögliche Werte: *full* (Standard), *short* oder *name*

Die info wird aus folgenden Quellen ermittelt (beachten Sie, dass nicht alle Dateien und Optionen in allen Distributionen vorhanden sind):

- `/proc/version` (*full*) unter Linux;
- `/proc/version_signature` (*short*) unter Linux;
- der Parameter PRETTY_NAME aus `/etc/os-release` auf Linux-Systemen, die ihn unterstützen, oder `/etc/issue.net` (*name*);
- der Registrierungsschlüssel `HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion` unter Windows.

Beispiele:

```
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

 Detaillierte Informationen über das Betriebssystem (Version, Typ, Name der Distribution, Neben- und Hauptversion usw.).
 Rückgabewert: *JSON-String*.
 Unterstützte Plattformen: Linux, Windows.

system.sw.packages[<regexp>,<manager>,<format>]

 Die Auflistung der installierten Pakete.
 Rückgabewert: *Text*.
 Unterstützte Plattformen: Linux.

Parameter:

- **regexp** - *all* (Standard) oder ein regulärer Ausdruck;
- **manager** - *all* (Standard) oder ein Paketmanager;
- **format** - *full* (Standard) oder *short*.

Kommentare:

- Listet die installierten Pakete (alphabetisch) auf, deren Name dem angegebenen regulären Ausdruck entspricht (*all* listet alle auf).
- Unterstützte Paketmanager (ausgeführter Befehl):
  - pkg (`dpkg --get-selections`);
  - pkgtool (`ls /var/log/packages`);
  - rpm (`rpm -qa`);
  - pacman (`pacman -Q`);
  - portage.
- Wenn `format` als *full* angegeben ist, werden die Pakete nach Paketmanagern gruppiert (jeder Manager in einer eigenen Zeile, beginnend mit seinem Namen in eckigen Klammern).
- Wenn `format` als *short* angegeben ist, werden die Pakete nicht gruppiert und in einer einzigen Zeile aufgelistet.

Beispiel:

```
system.sw.packages[mini,dpkg,short] → python-minimal, python2.6-minimal, ubuntu-minimal
```

system.sw.packages.get[<regexp>,<manager>]

 Eine detaillierte Auflistung der installierten Pakete.
 Rückgabewert: *JSON string*.
 Unterstützte Plattformen: Linux.

Parameter:

- **regexp** - *all* (Standard) oder ein regulärer Ausdruck;
- **manager** - *all* (Standard) oder ein Paketmanager (mögliche Werte: *rpm*, *dpkg*, *pkgtool*, *pacman* oder *portage*).

Kommentare:

- Gibt unformatiertes JSON mit den installierten Paketen zurück, deren Name dem angegebenen regulären Ausdruck entspricht.
- Die Ausgabe ist ein Array von Objekten, die jeweils die folgenden Schlüssel enthalten: name, manager, version, size, architecture, buildtime und installtime (siehe weitere Details).

system.swap.in[<device>,<type>]

 Die Swap-in-Statistiken (vom Gerät in den Speicher).
 Rückgabewert: *Integer*.
 Unterstützte Plattformen: Linux, FreeBSD, OpenBSD.

Parameter:

- **device** - gibt das für das Swapping verwendete Gerät an (nur Linux) oder *all* (Standard);
- **type** - mögliche Werte:
  - *count* (Anzahl der Swap-ins, Standard auf Nicht-Linux-Plattformen);
  - *sectors* (eingetauschte Sektoren);
  - *pages* (eingetauschte Seiten, Standard unter Linux).

Kommentare:

- Die Quelle dieser Informationen ist:
  - `/proc/swaps`, `/proc/partitions`, `/proc/stat` (Linux 2.4);
  - `/proc/swaps`, `/proc/diskstats`, `/proc/vmstat` (Linux 2.6).
- Beachten Sie, dass *pages* nur funktioniert, wenn device nicht angegeben wurde.
- Der Parametertyp *sectors* wird nur unter Linux unterstützt.

Beispiel:

```
system.swap.in[,pages]
```

system.swap.out[<device>,<type>]

 Die Swap-Out-Statistiken (aus dem Speicher auf das Gerät).
 Rückgabewert: *Integer*.
 Unterstützte Plattformen: Linux, FreeBSD, OpenBSD.

Parameter:

- **device** - gibt das für das Swapping verwendete Gerät an (nur Linux) oder *all* (Standard);
- **type** - mögliche Werte:
  - *count* (Anzahl der Swap-Out-Vorgänge, Standard auf Nicht-Linux-Plattformen);
  - *sectors* (ausgelagerte Sektoren);
  - *pages* (ausgelagerte Seiten, Standard auf Linux).

Kommentare:

- Die Quelle dieser Informationen ist:
  - `/proc/swaps`, `/proc/partitions`, `/proc/stat` (Linux 2.4);
  - `/proc/swaps`, `/proc/diskstats`, `/proc/vmstat` (Linux 2.6).
- Beachten Sie, dass *pages* nur funktioniert, wenn device nicht angegeben wurde.
- Der Parametertyp *sectors* wird nur unter Linux unterstützt.

Beispiel:

```
system.swap.out[,pages]
```

system.swap.size[<device>,<type>]

 Die Größe des Swap-Speichers in Bytes oder als Prozentsatz des Gesamtwerts.
 Rückgabewert: *Integer* - für Bytes; *Float* - für Prozentsatz.
 Unterstützte Plattformen: Linux, FreeBSD, Solaris, AIX, OpenBSD, Windows.

Parameter:

- **device** - gibt das für den Swap verwendete Gerät an (nur FreeBSD) oder *all* (Standard);
- **type** - mögliche Werte:
  - *free* (freier Swap-Speicher, Standard);
  - *pfree* (freier Swap-Speicher, in Prozent);
  - *pused* (genutzter Swap-Speicher, in Prozent);
  - *total* (gesamter Swap-Speicher);
  - *used* (genutzter Swap-Speicher).

Kommentare:

- Beachten Sie, dass *pfree* und *pused* unter Windows nicht unterstützt werden, wenn die Swap-Größe 0 ist.
- Wenn device nicht angegeben ist, berücksichtigt der Zabbix Agent nur Swap-Geräte (Dateien); der physische Speicher wird ignoriert. Zum Beispiel enthält der Befehl `swap -s` auf Solaris-Systemen einen Teil des physischen Speichers und der Swap-Geräte (im Gegensatz zu `swap -l`).

Beispiel:

```
system.swap.size[,pfree] → prozentualer freier Swap-Speicher
```

system.uname

 Identifikation des Systems.
 Rückgabewert: *String*.
 Siehe unterstützte Plattformen.

Kommentare:

- Unter UNIX wird der Wert für diesen Datenpunkt mit dem Systemaufruf uname() ermittelt.
- Unter Windows gibt der Datenpunkt die Betriebssystemarchitektur zurück, während er unter UNIX die CPU-Architektur zurückgibt.

Beispiele:

```
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

 Die Systemlaufzeit in Sekunden.
 Rückgabewert: *Integer*.
 Unterstützte Plattformen: Linux, FreeBSD, Solaris, AIX, MacOS X, OpenBSD, NetBSD, Windows.

Verwenden Sie in der Datenpunkt-Konfiguration die Einheiten **s** oder **uptime**, um lesbare Werte zu erhalten.

system.users.num

 Die Anzahl der angemeldeten Benutzer.
 Rückgabewert: *Integer*.
 Siehe unterstützte Plattformen.

Der Befehl **who** wird auf der Agent-Seite verwendet, um den Wert zu ermitteln.

vfs.dev.discovery

 Die Liste der Blockgeräte und ihres Typs. Wird für Low-Level-Discovery verwendet.
 Rückgabewert: *JSON-String*.
 Unterstützte Plattformen: Linux.

vfs.dev.get[<mode>,<regexp>]

 Die Liste der Blockgeräte mit detaillierten Informationen. Wird für Low-Level-Discovery verwendet.
 Rückgabewert: *JSON-String*.
 Unterstützte Plattformen: Linux, AIX.

Parameter:

- **mode** - mögliche Werte:
  - *disks* (Standard) - gibt die Liste der Festplatten zurück;
  - *disk_stats* - gibt Statistiken für jede Festplatte zurück;
  - *devices* (nur Linux) - gibt die Liste der Blockgeräte zurück (Festplatten und Partitionen);
  - *device_stats* (nur Linux) - gibt Statistiken für jedes Gerät zurück (Festplatten und Partitionen).
- **regexp** - ein regulärer Ausdruck, der verwendet wird, um Blockgeräte anhand ihres Namens zu filtern.

vfs.dev.read[<device>,<type>,<mode>]

 Die Statistik der Festplattenlesevorgänge.
 Rückgabewert: - *Integer* - bei `type` in *sectors*, *operations*, *bytes*; - *Float* - bei `type` in *sps*, *ops*, *bps*.
 Unterstützte Plattformen: Linux, FreeBSD, Solaris, AIX, OpenBSD.

Parameter:

- **device** - Festplattengerät (Standard ist *all* [3]);
- **type** - mögliche Werte: *sectors*, *operations*, *bytes*, *sps*, *ops* oder *bps* (*sps*, *ops*, *bps* stehen jeweils für: sectors, operations, bytes per second);
- **mode** - mögliche Werte: *avg1* (Ein-Minuten-Durchschnitt, Standard), *avg5* oder *avg15*. Dieser Parameter wird nur mit `type` in: sps, ops, bps unterstützt.

Kommentare:

- Wenn ein Aktualisierungsintervall von drei Stunden oder mehr[2] verwendet wird, gibt dieser Datenpunkt immer 0 zurück.
- Die Parameter *sectors* und *sps* werden nur unter Linux unterstützt.
- Der Parameter *ops* wird nur unter Linux und FreeBSD unterstützt.
- Der Parameter *bps* wird nur unter FreeBSD unterstützt.
- Der Parameter *bytes* wird nur unter FreeBSD, Solaris, AIX, OpenBSD unterstützt.
- Der Parameter `mode` wird nur unter Linux, FreeBSD unterstützt.
- Sie können relative Gerätenamen (zum Beispiel `sda`) sowie optional ein `/dev/`-Präfix (zum Beispiel `/dev/sda`) verwenden.
- LVM-Logische Volumes werden unterstützt.
- Die Standardwerte des Parameters 'type' für verschiedene Betriebssysteme:
  - *AIX* - operations;
  - *FreeBSD* - bps;
  - *Linux* - sps;
  - *OpenBSD* - operations;
  - *Solaris* - bytes.
- *sps*, *ops* und *bps* sind auf unterstützten Plattformen auf 1024 Geräte begrenzt (1023 einzelne und eines für *all*).

Beispiel:

```
vfs.dev.read[,operations]
```

vfs.dev.write[<device>,<type>,<mode>]

 Die Statistiken für Schreibzugriffe auf Datenträger.
 Rückgabewert: *Integer* - bei `type` in *sectors*, *operations*, *bytes*; *Float* - bei `type` in *sps*, *ops*, *bps*.
 Unterstützte Plattformen: Linux, FreeBSD, Solaris, AIX, OpenBSD.

Parameter:

- **device** - Datenträgergerät (Standard ist *all* [3]);
- **type** - mögliche Werte: *sectors*, *operations*, *bytes*, *sps*, *ops* oder *bps* (*sps*, *ops*, *bps* stehen jeweils für: Sektoren, Operationen, Bytes pro Sekunde);
- **mode** - mögliche Werte: *avg1* (Ein-Minuten-Durchschnitt, Standard), *avg5* oder *avg15*. Dieser Parameter wird nur mit `type` in: sps, ops, bps unterstützt.

Kommentare:

- Wenn ein Aktualisierungsintervall von drei Stunden oder mehr[2] verwendet wird, gibt dieser Datenpunkt immer 0 zurück.
- Die Typ-Parameter *sectors* und *sps* werden nur unter Linux unterstützt.
- Der Typ-Parameter *ops* wird nur unter Linux und FreeBSD unterstützt.
- Der Typ-Parameter *bps* wird nur unter FreeBSD unterstützt.
- Der Typ-Parameter *bytes* wird nur unter FreeBSD, Solaris, AIX und OpenBSD unterstützt.

- Der Parameter `mode` wird nur unter Linux und FreeBSD unterstützt.
- Sie können relative Gerätenamen (zum Beispiel `sda`) sowie optional ein Präfix `/dev/` (zum Beispiel `/dev/sda`) verwenden.
- LVM-Logische Volumes werden unterstützt.
- Die Standardwerte des Parameters `type` für verschiedene Betriebssysteme:
  - *AIX* - operations
  - *FreeBSD* - bps
  - *Linux* - sps
  - *OpenBSD* - operations
  - *Solaris* - bytes
- *sps*, *ops* und *bps* sind auf unterstützten Plattformen auf 1024 Geräte begrenzt (1023 einzelne und eines für *all*).

Beispiel:

```
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>]

 Die Anzahl der Verzeichniseinträge.
 Rückgabewert: *Integer*.
 Siehe unterstützte Plattformen.

Parameter:

- **dir** - der absolute Pfad zum Verzeichnis;
- **regex incl** - ein regulärer Ausdruck, der das Namensmuster der einzuschließenden Entität (Datei, Verzeichnis, symbolischer Link) beschreibt; bei leerem Wert werden alle eingeschlossen (Standardwert);
- **regex excl** - ein regulärer Ausdruck, der das Namensmuster der auszuschließenden Entität (Datei, Verzeichnis, symbolischer Link) beschreibt; bei leerem Wert wird nichts ausgeschlossen (Standardwert);
- **types incl** - zu zählende Typen von Verzeichniseinträgen, mögliche Werte:
  - *file* - reguläre Datei;
  - *dir* - Unterverzeichnis;
  - *sym* - symbolischer Link;
  - *sock* - Socket;
  - *bdev* - Blockgerät;
  - *cdev* - Zeichengerät;
  - *fifo* - FIFO;
  - *dev* - synonym für „bdev,cdev“;
  - *all* - alle Typen (Standard), d. h. „file,dir,sym,sock,bdev,cdev,fifo“. Mehrere Typen müssen durch Kommas getrennt und in Anführungszeichen gesetzt werden.
- **types excl** - Typen von Verzeichniseinträgen (siehe `types incl`), die NICHT gezählt werden sollen. Wenn ein Eintragstyp sowohl in `types incl` als auch in `types excl` enthalten ist, werden Verzeichniseinträge dieses Typs NICHT gezählt.
- **max depth** - die maximale Tiefe der zu durchlaufenden Unterverzeichnisse:
  - **-1** (Standard) - unbegrenzt;
  - **0** - keine Unterverzeichnisse durchsuchen.
- **min size** - die Mindestgröße (in Bytes) der zu zählenden Datei. Kleinere Dateien werden nicht gezählt. Speicher-Suffixe können verwendet werden.
- **max size** - die maximale Größe (in Bytes) der zu zählenden Datei. Größere Dateien werden nicht gezählt. Speicher-Suffixe können verwendet werden.
- **min age** - das Mindestalter (in Sekunden) des zu zählenden Verzeichniseintrags. Neuere Einträge werden nicht gezählt. Zeit-Suffixe können verwendet werden.
- **max age** - das Höchstalter (in Sekunden) des zu zählenden Verzeichniseintrags. Einträge, die so alt oder älter sind, werden nicht gezählt (Änderungszeit). Zeit-Suffixe können verwendet werden.
- **regex excl dir** - ein regulärer Ausdruck, der das Namensmuster des auszuschließenden Verzeichnisses beschreibt. Der gesamte Inhalt des Verzeichnisses wird ausgeschlossen (im Gegensatz zu `regex_excl`).

Hinweise:

- Unter Windows folgt dieser Datenpunkt UNC-Pfaden, was ein Sicherheitsrisiko darstellen kann.
- Umgebungsvariablen wie `%APP_HOME%`, `$HOME` und `%TEMP%` werden nicht unterstützt.
- Die Pseudoverzeichnisse „.“ und „..“ werden niemals gezählt.
- Symbolischen Links wird bei der Verzeichnisdurchsuchung niemals gefolgt.
- Sowohl `regex incl` als auch `regex excl` werden bei der Berechnung der Anzahl der Einträge auf Dateien und Verzeichnisse angewendet, beim Auswählen der zu durchsuchenden Unterverzeichnisse jedoch ignoriert (wenn `regex incl (?i)^.+\.zip$` lautet und `max depth` nicht festgelegt ist, werden alle Unterverzeichnisse durchsucht, aber nur Dateien des Typs zip gezählt).
- Die Ausführungszeit ist durch den Timeout-Wert in der Agent-Konfiguration begrenzt (3 Sekunden). Da die Durchsuchung großer Verzeichnisse länger dauern kann, werden keine Daten zurückgegeben und der Datenpunkt wird nicht unterstützt. Eine teilweise Anzahl wird nicht zurückgegeben.

- Beim Filtern nach Größe haben nur reguläre Dateien aussagekräftige Größen. Unter Linux und BSD haben auch Verzeichnisse eine Größe ungleich null (typischerweise einige KB). Geräte haben eine Größe von null; beispielsweise entspricht die Größe von `/dev/sda1` nicht der Größe der jeweiligen Partition. Daher wird bei Verwendung von `<min_size>` und `<max_size>` empfohlen, `<types_incl>` als „*file*“ anzugeben, um unerwartete Ergebnisse zu vermeiden.

Beispiele:

```
vfs.dir.count[/dev] #überwacht die Anzahl der Geräte in /dev (Linux)
vfs.dir.count["C:\Users\ADMINI~1\AppData\Local\Temp"] #überwacht die Anzahl der Dateien in einem temporäre
```

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>]

 Die Liste der Verzeichniseinträge.
 Rückgabewert: *JSON-Zeichenfolge*.
 Siehe unterstützte Plattformen.

Parameter:

- **dir** - der absolute Pfad zum Verzeichnis;
- **regex incl** - ein regulärer Ausdruck, der das Namensmuster der einzuschließenden Entität (Datei, Verzeichnis, symbolischer Link) beschreibt; alle einschließen, wenn leer (Standardwert);
- **regex excl** - ein regulärer Ausdruck, der das Namensmuster der auszuschließenden Entität (Datei, Verzeichnis, symbolischer Link) beschreibt; nichts ausschließen, wenn leer (Standardwert);
- **types incl** - aufzulistende Typen von Verzeichniseinträgen, mögliche Werte:
  - *file* - reguläre Datei;
  - *dir* - Unterverzeichnis;
  - *sym* - symbolischer Link;
  - *sock* - Socket;
  - *bdev* - Blockgerät;
  - *cdev* - Zeichengerät;
  - *fifo* - FIFO;
  - *dev* - synonym für ”bdev,cdev”;
  - *all* - alle Typen (Standard), d. h. ”file,dir,sym,sock,bdev,cdev,fifo”. Mehrere Typen müssen durch Kommas getrennt und in Anführungszeichen gesetzt werden.
- **types excl** - Typen von Verzeichniseinträgen (siehe `types incl`), die NICHT aufgelistet werden sollen. Wenn ein Eintragstyp sowohl in `types incl` als auch in `types excl` enthalten ist, werden Verzeichniseinträge dieses Typs NICHT aufgelistet.
- **max depth** - die maximale Tiefe der zu durchlaufenden Unterverzeichnisse:
  - **-1** (Standard) - unbegrenzt;
  - **0** - keine Unterverzeichnisse durchsuchen.
- **min size** - die Mindestgröße (in Byte), damit eine Datei aufgelistet wird. Kleinere Dateien werden nicht aufgelistet. Speichergrößen-Suffixe können verwendet werden.
- **max size** - die maximale Größe (in Byte), damit eine Datei aufgelistet wird. Größere Dateien werden nicht aufgelistet. Speichergrößen-Suffixe können verwendet werden.
- **min age** - das Mindestalter (in Sekunden) eines aufzulistenden Verzeichniseintrags. Neuere Einträge werden nicht aufgelistet. Zeitsuffixe können verwendet werden.
- **max age** - das Höchstalter (in Sekunden) eines aufzulistenden Verzeichniseintrags. Einträge, die so alt oder älter sind, werden nicht aufgelistet (Änderungszeit). Zeitsuffixe können verwendet werden.
- **regex excl dir** - ein regulärer Ausdruck, der das Namensmuster des auszuschließenden Verzeichnisses beschreibt. Der gesamte Inhalt des Verzeichnisses wird ausgeschlossen (im Gegensatz zu `regex excl`).

Hinweise:

- Unter Windows folgt dieser Datenpunkt UNC-Pfaden, was ein Sicherheitsrisiko darstellen kann.
- Umgebungsvariablen wie `%APP_HOME%`, `$HOME` und `%TEMP%` werden nicht unterstützt.
- Die Pseudoverzeichnisse ”.” und ”..” werden niemals aufgelistet.
- Symbolischen Links wird beim Durchlaufen von Verzeichnissen niemals gefolgt.
- Sowohl `regex incl` als auch `regex excl` werden beim Erstellen der Liste der Einträge auf Dateien und Verzeichnisse angewendet, beim Auswählen der zu durchlaufenden Unterverzeichnisse jedoch ignoriert (wenn `regex incl` `(?i)^.+\.zip$` lautet und `max depth` nicht festgelegt ist, werden alle Unterverzeichnisse durchsucht, aber nur Dateien des Typs zip gezählt).
- Die Ausführungszeit wird durch den Timeout-Wert in der Agent-Konfiguration begrenzt. Da das Durchlaufen großer Verzeichnisse länger dauern kann, werden keine Daten zurückgegeben und der Datenpunkt wird nicht unterstützt. Eine Teilliste wird nicht zurückgegeben.
- Beim Filtern nach Größe haben nur reguläre Dateien aussagekräftige Größen. Unter Linux und BSD haben auch Verzeichnisse eine Größe ungleich null (typischerweise einige KB). Geräte haben die Größe null; beispielsweise entspricht die Größe von `/dev/sda1` nicht der Größe der jeweiligen Partition. Daher wird empfohlen, bei Verwendung von `min size` und `max size` `types incl` als ”*file*” anzugeben, um Überraschungen zu vermeiden.

Beispiele:

```
vfs.dir.get[/dev] #ruft die Geräteliste in /dev ab (Linux)
vfs.dir.get["C:\Users\ADMINI~1\AppData\Local\Temp"] #ruft die Dateiliste in einem temporären Verzeichnis a
```

vfs.dir.size[dir,<regex incl>,<regex excl>,<mode>,<max depth>,<regex excl dir>]

 Die Verzeichnisgröße (in Bytes).
 Rückgabewert: *Integer*.
 Unterstützte Plattformen: Linux, Windows. Das item kann auch auf anderen UNIX-ähnlichen Plattformen funktionieren.

Parameter:

- **dir** - der absolute Pfad zum Verzeichnis;
- **regex incl** - ein regulärer Ausdruck, der das Namensmuster des einzuschließenden Objekts (Datei, Verzeichnis, symbolischer Link) beschreibt; wenn leer, werden alle eingeschlossen (Standardwert);
- **regex excl** - ein regulärer Ausdruck, der das Namensmuster des auszuschließenden Objekts (Datei, Verzeichnis, symbolischer Link) beschreibt; wenn leer, wird nichts ausgeschlossen (Standardwert);
- **mode** - mögliche Werte:
  - *apparent* (Standard) - ermittelt die scheinbaren Dateigrößen statt der belegten Festplattennutzung (wirkt wie `du -sb dir`);
  - *disk* - ermittelt die belegte Festplattennutzung (wirkt wie `du -s -B1 dir`). Im Gegensatz zum du-Befehl berücksichtigt das item vfs.dir.size bei der Berechnung der Verzeichnisgröße auch versteckte Dateien (wirkt innerhalb von dir wie `du -sb .[^.]* *`).
- **max depth** - die maximale Tiefe der zu durchlaufenden Unterverzeichnisse:
  - **-1** (Standard) - unbegrenzt;
  - **0** - kein Abstieg in Unterverzeichnisse.
- **regex excl dir** - ein regulärer Ausdruck, der das Namensmuster des auszuschließenden Verzeichnisses beschreibt. Der gesamte Inhalt des Verzeichnisses wird ausgeschlossen (im Gegensatz zu `regex excl`)

Kommentare:

- Unter Windows folgt dieses item UNC-Pfaden, was ein Sicherheitsrisiko darstellen kann.
- Es werden nur Verzeichnisse mit mindestens Leseberechtigung für den Benutzer *zabbix* berechnet. Bei Verzeichnissen mit nur Leseberechtigung wird nur die Größe des Verzeichnisses selbst berechnet. Verzeichnisse mit Lese- und Ausführungsberechtigung werden einschließlich ihres Inhalts berechnet.
- Bei großen Verzeichnissen oder langsamen Laufwerken kann dieses item aufgrund der Timeout-Einstellung in den Konfigurationsdateien von Agent und Server/Proxy ein Timeout erreichen. Erhöhen Sie die Timeout-Werte bei Bedarf.
- Die Dateigrößenbegrenzung hängt von der Unterstützung großer Dateien ab.

Beispiele:

```
vfs.dir.size[/tmp,log] #berechnet die Größe aller Dateien in /tmp, deren Namen 'log' enthalten
vfs.dir.size[/tmp,log,^.+\.old$] #berechnet die Größe aller Dateien in /tmp, deren Namen 'log' enthalten,
```

vfs.file.cksum[file,<mode>]

 Die Dateiprüfsumme, berechnet mit dem UNIX-Algorithmus cksum.
 Rückgabewert: *Integer* - bei `mode` als *crc32*, *String* - bei `mode` als *md5*, *sha256*.
 Siehe unterstützte Plattformen.

Parameter:

- **file** - der vollständige Pfad zur Datei;
- **mode** - *crc32* (Standard), *md5* oder *sha256*.

Kommentare:

- Unter Windows folgt dieser Datenpunkt UNC-Pfaden, was ein Sicherheitsrisiko darstellen kann.
- Die Dateigrößenbegrenzung hängt von der Unterstützung großer Dateien ab.

Beispiel:

```
vfs.file.cksum[/etc/passwd]
```

Beispiele für zurückgegebene Werte (jeweils crc32/md5/sha256):

```
675436101
9845acf68b73991eb7fd7ee0ded23c44
ae67546e4aac995e5c921042d0cf0f1f7147703aa42bfbfb65404b30f238f2dc
```

vfs.file.contents[file,<encoding>]

 Abrufen des Inhalts einer Datei[7].
 Rückgabewert: *Text*.
 Siehe unterstützte Plattformen.

Parameter:

- **file** - der vollständige Pfad zur Datei;
- **encoding** - der Codepage-Bezeichner.

Kommentare:

- Unter Windows folgt dieser Datenpunkt UNC-Pfaden, was ein Sicherheitsrisiko darstellen kann.
- Der Rückgabewert ist auf 16 MB begrenzt (einschließlich nachfolgender Leerzeichen, die abgeschnitten werden); Datenbankgrenzen gelten ebenfalls.
- Ein leerer String wird zurückgegeben, wenn die Datei leer ist oder nur LF-/CR-Zeichen enthält.
- Die Byte-Reihenfolge-Markierung (BOM) wird von der Ausgabe ausgeschlossen.

Beispiel:

```
vfs.file.contents[/etc/passwd]
```

vfs.file.exists[file,<types incl>,<types excl>]

 Prüft, ob die Datei vorhanden ist.
 Rückgabewert: 0 - nicht gefunden; 1 - Datei des angegebenen Typs vorhanden.
 Siehe unterstützte Plattformen.

Parameter:

- **file** - der vollständige Pfad zur Datei;
- **types incl** - die Liste der einzuschließenden Dateitypen, mögliche Werte:
  - *file* (normale Datei, Standard (wenn `types_excl` nicht gesetzt ist));
  - *dir* (Verzeichnis);
  - *sym* (symbolischer Link);
  - *sock* (Socket);
  - *bdev* (Blockgerät);
  - *cdev* (Zeichengerät);
  - *fifo* (FIFO);
  - *dev* (synonym für "bdev,cdev");
  - *all* (alle genannten Typen, Standard, wenn `types_excl` gesetzt ist).
- **types excl** - die Liste der auszuschließenden Dateitypen, siehe `types_incl` für mögliche Werte (standardmäßig werden keine Typen ausgeschlossen)

Kommentare:

- Unter Windows folgt dieser Datenpunkt UNC-Pfaden, was ein Sicherheitsrisiko darstellen kann.
- Mehrere Typen müssen durch ein Komma getrennt und die gesamte Menge in Anführungszeichen "" eingeschlossen werden.
- Wenn derselbe Typ sowohl in <types_incl> als auch in <types_excl> enthalten ist, werden Dateien dieses Typs ausgeschlossen.
- Die Dateigrößenbegrenzung hängt von der Unterstützung großer Dateien ab.

Beispiele:

```
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]

 Gibt Informationen über eine Datei zurück.
 Rückgabewert: *JSON-String*.
 Siehe unterstützte Plattformen.

Parameter:

- **file** - der vollständige Pfad zur Datei

Kommentare:

- Unter Windows folgt dieser Datenpunkt UNC-Pfaden, was ein Sicherheitsrisiko darstellen kann.
- Unterstützte Dateitypen auf UNIX-ähnlichen Systemen: reguläre Datei, Verzeichnis, symbolischer Link, Socket, Blockgerät, Zeichengerät, FIFO.
- Die Dateigrößenbeschränkung hängt von der Unterstützung großer Dateien ab.

Beispiel:

```
vfs.file.get[/etc/passwd] #return a JSON with information about the /etc/passwd file (type, user, permissi
```

vfs.file.md5sum[file]

 Die MD5-Prüfsumme der Datei.
 Rückgabewert: Zeichenkette (MD5-Hash der Datei).
 Siehe unterstützte Plattformen.

Parameter:

- **file** - der vollständige Pfad zur Datei

Kommentare:

- Unter Windows folgt dieses Item UNC-Pfaden, was ein Sicherheitsrisiko darstellen kann.
- Die Dateigrößenbegrenzung hängt von der Unterstützung großer Dateien ab.

Beispiel:

```
vfs.file.md5sum[/usr/local/etc/zabbix_agentd.conf]
```

Beispiel für den zurückgegebenen Wert:

```
b5052decb577e0fffd622d6ddc017e82
```

vfs.file.owner[file,<ownertype>,<resulttype>]

 Ruft den Besitzer einer Datei ab.
 Rückgabewert: *String*.
 Siehe unterstützte Plattformen.

Parameter:

- **file** - der vollständige Pfad zur Datei;
- **ownertype** - *user* (Standard) oder *group* (nur Unix);
- **resulttype** - *name* (Standard) oder *id*; bei id - gibt unter Unix uid/gid und unter Windows SID zurück.

Kommentare:

- Unter Windows folgt dieser Datenpunkt UNC-Pfaden, was ein Sicherheitsrisiko darstellen kann.
- Die Dateigrößenbeschränkung hängt von der Unterstützung großer Dateien ab.

Beispiele:

```
vfs.file.owner[/tmp/zabbix_server.log] #gibt den Dateibesitzer von /tmp/zabbix_server.log zurück
vfs.file.owner[/tmp/zabbix_server.log,,id] #gibt die Besitzer-ID von /tmp/zabbix_server.log zurück
```

vfs.file.permissions[file]

 Gibt eine 4-stellige Zeichenfolge zurück, die die Oktalzahl mit UNIX-Berechtigungen enthält.
 Rückgabewert: *String*.
 Unterstützte Plattformen: Linux. Der Datenpunkt kann auch auf anderen UNIX-ähnlichen Plattformen funktionieren.

Parameter:

- **file** - der vollständige Pfad zur Datei

Die Dateigrößenbeschränkung hängt von der Unterstützung großer Dateien ab.

Beispiel:

```
vfs.file.permissions[/etc/passwd] #gibt die Berechtigungen von /etc/passwd zurück, zum Beispiel '0644'
```

vfs.file.regexp[file,regexp,<encoding>,<start line>,<end line>,<output>]

 Eine Zeichenkette in der Datei abrufen[7].
 Rückgabewert: Die Zeile, die die übereinstimmende Zeichenkette enthält, oder wie durch den optionalen Parameter `output` angegeben.
 Siehe unterstützte Plattformen.

Parameter:

- **file** - der vollständige Pfad zur Datei;
- **regexp** - ein regulärer Ausdruck, der das erforderliche Muster beschreibt;
- **encoding** - die Codepage-Kennung;
- **start line** - die Nummer der ersten zu durchsuchenden Zeile (standardmäßig die erste Zeile der Datei);
- **end line** - die Nummer der letzten zu durchsuchenden Zeile (standardmäßig die letzte Zeile der Datei);
- **output** - eine optionale Vorlage für die Ausgabeformatierung. Die Escape-Sequenz **\0** wird durch den übereinstimmenden Teil des Textes ersetzt (vom ersten Zeichen, an dem die Übereinstimmung beginnt, bis zu dem Zeichen, an dem die Übereinstimmung endet), während eine **\N**-Escape-Sequenz (wobei N=1...9) durch die N-te übereinstimmende Gruppe ersetzt wird (oder durch eine leere Zeichenkette, wenn N die Anzahl der erfassten Gruppen überschreitet).

Kommentare:

- Unter Windows folgt dieses Item UNC-Pfaden, was ein Sicherheitsrisiko darstellen kann.
- Die Dateigrößenbeschränkung hängt von der Unterstützung großer Dateien ab.
- Es wird nur die erste übereinstimmende Zeile zurückgegeben.

- Wenn keine Zeile mit dem Ausdruck übereinstimmt, wird eine leere Zeichenkette zurückgegeben.
- Die Byte Order Mark (BOM) wird von der Ausgabe ausgeschlossen.
- Die Inhaltsextraktion mithilfe des Parameters `output` erfolgt auf dem Agent.

Beispiele:

```
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] → Abrufen der ID des Benutzers *zabbix*
```

vfs.file.regmatch[file,regexp,<encoding>,<start line>,<end line>]

 Findet eine Zeichenfolge in der Datei[7].
 Rückgabewerte: 0 - Übereinstimmung nicht gefunden; 1 - gefunden.
 Siehe unterstützte Plattformen.

Parameter:

- **file** - der vollständige Pfad zur Datei;
- **regexp** - ein regulärer Ausdruck, der das erforderliche Muster beschreibt;
- **encoding** - die Codepage-Kennung;
- **start line** - die Nummer der ersten zu durchsuchenden Zeile (standardmäßig die erste Zeile der Datei);
- **end line** - die Nummer der letzten zu durchsuchenden Zeile (standardmäßig die letzte Zeile der Datei).

Kommentare:

- Unter Windows folgt dieser Datenpunkt UNC-Pfaden, was ein Sicherheitsrisiko darstellen kann.
- Die Dateigrößenbegrenzung hängt von der Unterstützung großer Dateien ab.
- Die Byte Order Mark (BOM) wird ignoriert.

Beispiel:

```
vfs.file.regmatch[/var/log/app.log,error]
```

vfs.file.size[file,<mode>]

 Die Dateigröße (in Bytes).
 Rückgabewert: *Integer*.
 Siehe unterstützte Plattformen.

Parameter:

- **file** - der vollständige Pfad zur Datei;
- **mode** - mögliche Werte:
  - *bytes* (Standard);
  - *lines* (leere Zeilen werden ebenfalls gezählt).

Kommentare:

- Unter Windows folgt dieser Datenpunkt UNC-Pfaden, was ein Sicherheitsrisiko darstellen kann.
- Die Datei muss für den Benutzer *zabbix* Leserechte haben.
- Die Dateigrößenbegrenzung hängt von der Unterstützung großer Dateien ab.

Beispiel:

```
vfs.file.size[/var/log/syslog]
```

vfs.file.time[file,<mode>]

 Informationen zur Dateizeit.
 Rückgabewert: *Integer* (Unix-Zeitstempel).
 Siehe unterstützte Plattformen.

Parameter:

- **file** - der vollständige Pfad zur Datei;
- **mode** - mögliche Werte:
  - *modify* (Standard) - der Zeitpunkt der letzten Änderung des Dateiinhalts (Standard);
  - *access* - der Zeitpunkt des letzten Zugriffs auf die Datei;
  - *change* - der Zeitpunkt der letzten Änderung der Dateieigenschaften.

Kommentare:

- Unter Windows folgt dieser Datenpunkt UNC-Pfaden, was ein Sicherheitsrisiko darstellen kann.
- Die Dateigrößenbegrenzung hängt von der Unterstützung großer Dateien ab.

Beispiel:

```
vfs.file.time[/etc/passwd,modify]
```

vfs.fs.discovery

 Die Liste der eingehängten Dateisysteme mit ihrem Typ und den Einhängeoptionen. Wird für Low-Level-Discovery verwendet.
 Rückgabewert: *JSON-String*.
 Unterstützte Plattformen: Linux, FreeBSD, Solaris, HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows.

vfs.fs.get[<mode>,<mountpoint>]

 Die Liste der eingehängten Dateisysteme mit ihrem Typ und ihren Einhängeoptionen sowie im Modus *full* der verfügbare Speicherplatz und die Inode-Statistiken. Kann für die Low-Level-Discovery verwendet werden.
 Rückgabewert: *JSON-Zeichenfolge*.
 Unterstützte Plattformen: Linux, FreeBSD, Solaris, HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows.

Parameter:

- **mode** - mögliche Werte:
  - *full* (Standard) - Dateisystemeigenschaften mit Speicherplatz- und Inode-Statistiken;
  - *short* - nur Dateisystemeigenschaften, ohne Speicherplatz- und Inode-Statistiken;
- **mountpoint** - der Einhängepunkt, für den Daten zurückgegeben werden sollen (zum Beispiel `/var` oder `C:`). Wenn nicht angegeben, werden alle eingehängten Dateisysteme zurückgegeben.

Kommentare:

- Im Modus *short* werden nur der Dateisystemname, der Typ und die Einhängeoptionen zurückgegeben (unter Windows: Name, Typ, Bezeichnung und Laufwerkstyp). Die Dateisysteme werden nicht auf Statistiken abgefragt; auf Unix-ähnlichen Systemen wird nur die Liste der eingehängten Dateisysteme gelesen (zum Beispiel `/proc/mounts` unter Linux), sodass die Prüfung kein Automounting auslöst und nicht durch nicht reagierende Netzwerkdateisysteme beeinträchtigt wird. Dadurch eignet sich der Modus *short* für die Low-Level-Discovery.
- Der Wert **mountpoint** muss exakt mit der Eigenschaft `fsname` übereinstimmen. Unter Windows wird die Groß- und Kleinschreibung nicht berücksichtigt; geben Sie den Einhängepunkt ohne den abschließenden umgekehrten Schrägstrich an (zum Beispiel `C:`).
- Wenn **mountpoint** angegeben ist, werden nur für diesen Einhängepunkt Daten zurückgegeben. Wenn kein eingehängtes Dateisystem übereinstimmt, wird ein leeres Array (`[]`) zurückgegeben.
- Die Ausgabe kann mehrere Einträge mit demselben `fsname` enthalten, wenn mehr als ein Dateisystem im selben Verzeichnis eingehängt ist.
- Im Modus *full* werden verfügbarer Speicherplatz und Inode-Statistiken nur zurückgegeben, wenn der Benutzer des Zabbix Agent über die Berechtigung verfügt, jedes Verzeichnis im Pfad des Einhängepunkts zu durchqueren; andernfalls wird das Dateisystem aus der Ausgabe ausgelassen. Dies gilt nicht für den Modus *short*, der daher Dateisysteme auflisten kann, die in der Ausgabe des Modus *full* fehlen.
- Die Eigenschaften `fslabel` und `fsdrivetype` werden nur unter Windows zurückgegeben; die Eigenschaften `inodes` und `options` werden unter Windows nicht zurückgegeben.
- Dateisysteme mit einer Inode-Anzahl von null, was bei Dateisystemen mit dynamischen Inodes (z. B. btrfs) der Fall sein kann, werden ebenfalls gemeldet.
- Siehe auch: Erkennung eingehängter Dateisysteme.

Beispiele:

```
vfs.fs.get #alle eingehängten Dateisysteme mit Statistiken; entspricht vfs.fs.get[full]
vfs.fs.get[short] #alle eingehängten Dateisysteme ohne Statistiken
vfs.fs.get[full,/var] #Statistiken nur für den Einhängepunkt /var
vfs.fs.get[short,C:] #Eigenschaften nur des C:-Volumes (Windows)
```

vfs.fs.inode[fs,<mode>]

 Die Anzahl oder der Prozentsatz der Inodes.
 Rückgabewert: *Integer* - für die Anzahl; *Float* - für den Prozentsatz.
 Siehe unterstützte Plattformen.

Parameter:

- **fs** - das Dateisystem;
- **mode** - mögliche Werte: *total* (Standard), *free*, *used*, *pfree* (frei, in Prozent) oder *pused* (belegt, in Prozent).

Wenn die Anzahl der Inodes gleich null ist, was bei Dateisystemen mit dynamischen Inodes der Fall sein kann (z. B. btrfs), werden die Werte für pfree/pused jeweils als „100“ und „0“ gemeldet.

Beispiel:

```
vfs.fs.inode[/,pfree]
```

vfs.fs.size[fs,<mode>]

 Der Festplattenspeicher in Byte oder als Prozentsatz der Gesamtkapazität.
 Rückgabewert: *Integer* - für Byte; *Float* - für Prozentangaben.
 Siehe unterstützte Plattformen.

Parameter:

- **fs** - das Dateisystem;
- **mode** - mögliche Werte: *total* (Standard), *free*, *used*, *pfree* (frei, in Prozent) oder *pused* (belegt, in Prozent).

Kommentare:

- Wenn das Dateisystem nicht eingehängt ist, wird die Größe eines lokalen Dateisystems zurückgegeben, auf dem sich der Einhängepunkt befindet.
- Der reservierte Speicherplatz eines Dateisystems wird berücksichtigt und bei Verwendung des Modus *free* nicht einbezogen.

Beispiel:

```
vfs.fs.size[/tmp,free]
```

vm.memory.size[<mode>]

 Die Speichergröße in Byte oder als Prozentsatz der Gesamtmenge.
 Rückgabewert: *Integer* - für Byte; *Float* - für Prozentangaben.
 Siehe unterstützte Plattformen.

Parameter:

- **mode** - mögliche Werte: *total* (Standard), *active*, *anon*, *buffers*, *cached*, *exec*, *file*, *free*, *inactive*, *pinned*, *shared*, *slab*, *wired*, *used*, *pused* (verwendet, prozentual), *available* oder *pavailable* (verfügbar, prozentual).

Kommentare:

- Dieser Datenpunkt akzeptiert drei Kategorien von Parametern:
  - *total* - Gesamtspeichermenge;
  - plattformspezifische Speichertypen: *active*, *anon*, *buffers*, *cached*, *exec*, *file*, *free*, *inactive*, *pinned*, *shared*, *slab*, *wired*;
  - Schätzwerte auf Benutzerebene dafür, wie viel Speicher verwendet wird und verfügbar ist: *used*, *pused*, *available*, *pavailable*.
- Der Modusparameter *active* wird nur unter FreeBSD, HP-UX, MacOS X, OpenBSD und NetBSD unterstützt.
- Die Modusparameter *anon*, *exec*, *file* werden nur unter NetBSD unterstützt.
- Der Modusparameter *buffers* wird nur unter Linux, FreeBSD, OpenBSD und NetBSD unterstützt.
- Der Modusparameter *cached* wird nur unter Linux, FreeBSD, AIX, OpenBSD und NetBSD unterstützt.
- Die Modusparameter *inactive*, *wired* werden nur unter FreeBSD, MacOS X, OpenBSD und NetBSD unterstützt.
- Der Modusparameter *pinned* wird nur unter AIX unterstützt.
- Der Modusparameter *shared* wird nur unter Linux 2.4, FreeBSD, OpenBSD und NetBSD unterstützt.
- Siehe auch zusätzliche Details zu diesem Datenpunkt.

Beispiel:

```
vm.memory.size[pavailable]
```

web.page.get[host,<path>,<port>,<redirect_limit>]

 Ruft den Inhalt einer Webseite ab.
 Rückgabewert: Quelltext der Webseite als Text (einschließlich Headern).
 Siehe unterstützte Plattformen.

Parameter:

- **host** - der Hostname oder die URL (als `scheme://host:port/path`, wobei nur *host* erforderlich ist). Zulässige URL-Schemata: *http*, *https*[4]. Ein fehlendes Schema wird als *http* behandelt. Wenn eine URL angegeben wird, müssen `path` und `port` leer sein. Die Angabe eines Benutzernamens und Passworts für die Verbindung zu Servern, die eine Authentifizierung erfordern, beispielsweise `http://user:password@www.example.com`, ist nur mit cURL-Unterstützung möglich [4]. Punycode wird in Hostnamen unterstützt.
- **path** - der Pfad zu einem HTML-Dokument (Standard ist /);
- **port** - die Portnummer (Standard ist 80 für HTTP)
- **redirect_limit** - die maximale Anzahl der Weiterleitungen, denen gefolgt werden soll [4], von 0 bis 1000 (Standard ist 0 - Weiterleitungen werden nicht verfolgt).

Hinweise:

- Dieser Datenpunkt wird nicht unterstützt, wenn die in `host` angegebene Ressource nicht existiert oder nicht verfügbar ist oder wenn `redirect_limit` größer als 0 ist und die Anzahl der Weiterleitungen diesen Wert überschreitet.
- `host` kann ein Hostname, ein Domänenname, eine IPv4- oder IPv6-Adresse sein. Für eine IPv6-Adresse muss der Zabbix Agent jedoch mit aktivierter IPv6-Unterstützung kompiliert worden sein.

- Wenn Weiterleitungen verfolgt werden, enthält der zurückgegebene Wert die Header jeder Weiterleitungsantwort, gefolgt von den Headern und dem Inhalt der endgültigen Antwort.

Beispiele:

```
web.page.get[www.example.com,index.php,80]
web.page.get[https://www.example.com]
web.page.get[https://blog.example.com/?s=zabbix]
web.page.get[localhost:80]
web.page.get["[::1]/server-status"]
web.page.get[http://www.example.com,,,5]
```

web.page.perf[host,<path>,<port>,<redirect_limit>]

 Die Ladezeit einer vollständigen Webseite (in Sekunden).
 Rückgabewert: *Float*.
 Siehe unterstützte Plattformen.

Parameter:

- **host** - der Hostname oder die URL (als `scheme://host:port/path`, wobei nur *host* erforderlich ist). Zulässige URL-Schemata: *http*, *https*[4]. Ein fehlendes Schema wird als *http* behandelt. Wenn eine URL angegeben wird, müssen `path` und `port` leer sein. Die Angabe eines Benutzernamens und Kennworts für die Verbindung zu Servern, die eine Authentifizierung erfordern, beispielsweise `http://user:password@www.example.com`, ist nur mit cURL-Unterstützung möglich [4]. Punycode wird in Hostnamen unterstützt.
- **path** - der Pfad zu einem HTML-Dokument (Standard ist /);
- **port** - die Portnummer (Standard ist 80 für HTTP)
- **redirect_limit** - die maximale Anzahl der Weiterleitungen, denen gefolgt werden soll [4], von 0 bis 1000 (Standard ist 0 - Weiterleitungen werden nicht verfolgt).

Kommentare:

- Dieser Datenpunkt wird nicht unterstützt, wenn die in `host` angegebene Ressource nicht existiert oder nicht verfügbar ist oder wenn `redirect_limit` größer als 0 ist und die Anzahl der Weiterleitungen diesen Wert überschreitet.
- `host` kann ein Hostname, ein Domänenname, eine IPv4- oder IPv6-Adresse sein. Für eine IPv6-Adresse muss der Zabbix Agent jedoch mit aktivierter IPv6-Unterstützung kompiliert worden sein.
- Wenn Weiterleitungen verfolgt werden, umfasst die Ladezeit die für alle Weiterleitungen aufgewendete Zeit.

Beispiele:

```
web.page.perf[www.example.com,index.php,80]
web.page.perf[https://www.example.com]
web.page.perf[http://www.example.com,,,5]
```

web.page.regexp[host,<path>,<port>,regexp,<length>,<output>,<redirect_limit>]

 Sucht eine Zeichenfolge auf der Webseite.
 Rückgabewert: Die gefundene Zeichenfolge oder, wie durch den optionalen Parameter `output` angegeben.
 Siehe unterstützte Plattformen.

Parameter:

- **host** - der Hostname oder die URL (als `scheme://host:port/path`, wobei nur *host* erforderlich ist). Zulässige URL-Schemata: *http*, *https*[4]. Ein fehlendes Schema wird als *http* behandelt. Wenn eine URL angegeben wird, müssen `path` und `port` leer sein. Die Angabe von Benutzername und Passwort beim Verbinden mit Servern, die eine Authentifizierung erfordern, zum Beispiel: `http://user:password@www.example.com`, ist nur mit cURL-Unterstützung möglich [4]. Punycode wird in Hostnamen unterstützt.
- **path** - der Pfad zu einem HTML-Dokument (Standard ist /).
- **port** - die Portnummer (Standard ist 80 für HTTP).
- **regexp** - ein regulärer Ausdruck, der das erforderliche Muster beschreibt.
- **length** - die maximale Anzahl der zurückzugebenden Zeichen;
- **output** - eine optionale Vorlage zur Ausgabeformatierung. Die Escape-Sequenz **\0** wird durch den gefundenen Textabschnitt ersetzt (vom ersten Zeichen, an dem die Übereinstimmung beginnt, bis zu dem Zeichen, an dem sie endet), während eine Escape-Sequenz **\N** (wobei N=1...9) durch die N-te gefundene Gruppe ersetzt wird (oder durch eine leere Zeichenfolge, wenn N die Anzahl der erfassten Gruppen überschreitet).
- **redirect_limit** - die maximale Anzahl der Weiterleitungen, denen gefolgt werden soll [4], von 0 bis 1000 (Standard ist 0 - Weiterleitungen werden nicht verfolgt).

Hinweise:

- Dieser Datenpunkt wird nicht unterstützt, wenn die in `host` angegebene Ressource nicht existiert oder nicht verfügbar ist oder wenn `redirect_limit` größer als 0 ist und die Anzahl der Weiterleitungen diesen Wert überschreitet.

- `host` kann ein Hostname, ein Domänenname, eine IPv4- oder IPv6-Adresse sein. Für eine IPv6-Adresse muss der Zabbix Agent jedoch mit aktivierter IPv6-Unterstützung kompiliert worden sein.
- Die Inhaltsextraktion mithilfe des Parameters `output` erfolgt auf dem Agent.
- Wenn Weiterleitungen verfolgt werden, werden auch die Header jeder Weiterleitungsantwort durchsucht.

Beispiele:

```
web.page.regexp[www.example.com,index.php,80,OK,2]
web.page.regexp[https://www.example.com,,,OK,2]
web.page.regexp[http://www.example.com,,,OK,2,,5]
```

agent.hostmetadata

 Die Host-Metadaten des Agenten.
 Rückgabewert: *String*.
 Siehe unterstützte Plattformen.

Gibt den Wert der Parameter `HostMetadata` oder `HostMetadataItem` zurück, oder einen leeren String, wenn keine definiert sind.

Agent.hostname

 Der Hostname des Agent.
 Rückgabewert: *String*.
 Siehe unterstützte Plattformen.

Gibt zurück:

- Als passiver Check - der Name des ersten Hosts, der im Parameter Hostname der Konfigurationsdatei des Agent aufgeführt ist;
- Als aktiver Check - der Name des aktuellen Hostnamens.

agent.ping

 Die Verfügbarkeitsprüfung des Agent.
 Rückgabewert: Nichts - nicht verfügbar; 1 - verfügbar.
 Siehe unterstützte Plattformen.

Verwenden Sie die `nodata()`-Auslöserfunktion, um die Nichtverfügbarkeit des Host zu prüfen.

agent.variant

 Die Variante des Zabbix Agent (Zabbix Agent oder Zabbix Agent 2).
 Rückgabewert: 1 - Zabbix Agent; 2 - Zabbix Agent 2.
 Siehe unterstützte Plattformen.

agent.version

 Die Version des Zabbix Agent.
 Rückgabewert: *String*.
 Siehe unterstützte Plattformen.

Beispiel für einen Rückgabewert:

```
6.0.3
```

zabbix.stats[<ip>,<port>]

 Gibt eine Menge interner Metriken des Zabbix-Servers oder Proxys zurück. Wenn `ip` und `port` angegeben sind, werden die Metriken von der entfernten Instanz abgerufen; andernfalls von der lokalen Instanz.
 Rückgabewert: *JSON-String*.
 Siehe unterstützte Plattformen.

Parameter:

- **ip** - die IP-/DNS-/Netzmaskenliste der Server/Proxys, die remote abgefragt werden sollen (Standard ist `127.0.0.1`);
- **port** - der Port des Servers/Proxys, der remote abgefragt werden soll (Standard ist 10051)

Kommentare:

- Eine ausgewählte Menge interner Metriken wird von diesem Datenpunkt zurückgegeben. Für Details siehe Remote monitoring of Zabbix stats.
- Beachten Sie, dass die Stats-Anfrage nur von den Adressen akzeptiert wird, die im Parameter `StatsAllowedIP` der Zielinstanz Server/Proxy aufgeführt sind.

zabbix.stats[<ip>,<port>,queue,<from>,<to>]

 Gibt die Anzahl der überwachten Datenpunkte in der Warteschlange zurück, die auf dem Zabbix Server oder Proxy verzögert sind. Wenn `ip` und `port` angegeben sind, werden die Metriken von der entfernten Instanz abgerufen; andernfalls von der lokalen Instanz.
 Rückgabewert: *JSON-Zeichenfolge*.
 Siehe unterstützte Plattformen.

Parameter:

- **ip** - die IP-/DNS-/Netzmaskenliste der Server/Proxys, die remote abgefragt werden sollen (Standard ist `127.0.0.1`);
- **port** - der Port des Servers/Proxys, der remote abgefragt werden soll (Standard ist 10051)
- **queue** - Konstante (unverändert zu verwenden)

- **from** - mindestens verzögert um (Standard ist 6 Sekunden)
- **to** - höchstens verzögert um (Standard ist unendlich)

Beachten Sie, dass die Stats-Anfrage nur von den Adressen akzeptiert wird, die im Parameter 'StatsAllowedIP' von Server/Proxy auf der Zielinstanz aufgeführt sind.

Fußnoten

[1] Eine Linux-spezifische Anmerkung.
Der Zabbix Agent muss schreibgeschützten Zugriff auf das Dateisystem `/proc` haben.
Kernel-Patches von www.grsecurity.org schränken die Zugriffsrechte nicht privilegierter Benutzer ein.

[2] `vfs.dev.read[]`, `vfs.dev.write[]`: Der Zabbix Agent beendet „veraltete" Geräteverbindungen, wenn auf die Datenpunktwerte länger als 3 Stunden nicht zugegriffen wurde.
Dies kann vorkommen, wenn ein System Geräte mit sich dynamisch ändernden Pfaden enthält oder ein Gerät manuell entfernt wird.
Beachten Sie außerdem, dass diese Datenpunkte bei Verwendung eines Aktualisierungsintervalls von 3 Stunden oder mehr immer „0" zurückgeben.

[3] `vfs.dev.read[]`, `vfs.dev.write[]`: Wenn für den ersten Parameter der Standardwert `all` verwendet wird, gibt der Schlüssel zusammengefasste Statistiken zurück, einschließlich aller Blockgeräte wie sda und sdb sowie ihrer Partitionen (sda1, sda2, sdb3 ...) und mehrerer Geräte (MD-RAID), die auf diesen Blockgeräten/Partitionen basieren, sowie logischer Volumes (LVM), die auf diesen Blockgeräten/Partitionen basieren.
In solchen Fällen sollten die zurückgegebenen Werte nur als relative Werte (zeitlich dynamisch), nicht jedoch als absolute Werte betrachtet werden.

[4] HTTPS und das Folgen von Weiterleitungen (`redirect_limit` größer als 0) werden nur unterstützt, wenn der Zabbix Agent mit cURL-Unterstützung kompiliert wurde. Andernfalls wird der Datenpunkt nicht unterstützt. Für Zabbix Agent 2 ist cURL in beiden Fällen nicht erforderlich.

[5] Die Werte `bytes` und `errors` werden für Loopback-Schnittstellen auf Solaris-Systemen bis einschließlich Solaris 10 6/06 nicht unterstützt, da Byte-, Fehler- und Auslastungsstatistiken vom Kernel nicht gespeichert und/oder gemeldet werden.
Wenn Sie ein Solaris-System jedoch über net-snmp überwachen, werden möglicherweise Werte zurückgegeben, da net-snmp Legacy-Code aus cmu-snmp enthält, der bereits aus dem Jahr 1997 stammen kann. Wenn das Lesen von Bytewerten aus den Schnittstellenstatistiken fehlschlägt, gibt dieser Code den Paket-Zähler (der auf Loopback-Schnittstellen vorhanden ist), multipliziert mit einem willkürlichen Wert von 308, zurück.
Dabei wird angenommen, dass die durchschnittliche Länge eines Pakets 308 Oktette beträgt. Dies ist eine sehr grobe Schätzung, da die MTU-Grenze für Loopback-Schnittstellen auf Solaris-Systemen 8892 Byte beträgt.
Diese Werte sollten weder als korrekt noch als annähernd genau betrachtet werden.
Es handelt sich um Schätzwerte.
Der Zabbix Agent führt keine Schätzungen durch, net-snmp gibt jedoch einen Wert für diese Felder zurück.

[6] Die Befehlszeile unter Solaris, die aus `/proc/pid/psinfo` abgerufen wird, ist auf 80 Byte begrenzt und enthält die Befehlszeile so, wie sie beim Start des Prozesses war.

[7] Mit den Datenpunkten `vfs.file.contents[]`, `vfs.file.regexp[]`, `vfs.file.regmatch[]` können Dateiinhalte abgerufen werden.
Wenn Sie den Zugriff auf bestimmte Dateien mit vertraulichen Informationen beschränken möchten, führen Sie den Zabbix Agent unter einem Benutzer aus, der keine Zugriffsberechtigungen zum Anzeigen dieser Dateien besitzt.

Verwendung mit Befehlszeilenprogrammen

Beachten Sie, dass Sie beim Testen oder Verwenden von Datenpunktschlüsseln mit zabbix_agentd oder zabbix_get über die Befehlszeile auch die Shell-Syntax berücksichtigen sollten.

Wenn beispielsweise ein bestimmter Parameter des Schlüssels in doppelte Anführungszeichen eingeschlossen werden muss, müssen Sie die doppelten Anführungszeichen explizit maskieren. Andernfalls werden sie von der Shell als Sonderzeichen entfernt und nicht an das Zabbix-Dienstprogramm übergeben.

Beispiele:

```
zabbix_agentd -t 'vfs.dir.count[/var/log,,,"file,dir",,0]'

zabbix_agentd -t vfs.dir.count[/var/log,,,\"file,dir\",,0]
```

Kodierungseinstellungen

Um sicherzustellen, dass die erfassten Daten nicht beschädigt werden, können Sie im Parameter `encoding` die korrekte Kodierung für die Verarbeitung der Prüfung (z. B. `vfs.file.contents`) angeben. Die Liste der unterstützten Kodierungen (Codepage-Bezeichner) finden Sie in der Dokumentation zu libiconv (GNU-Projekt) oder in der Dokumentation des Microsoft Windows SDK zu "Code Page Identifiers". Beachten Sie, dass Microsoft einige Codepages gelegentlich als „nur für verwaltete Anwendungen

verfügbar“ kennzeichnet. Das bedeutet, dass sie nur über die .NET-Laufzeitumgebung bereitgestellt werden und nicht über die nativen Win32-APIs verfügbar sind. Der Zabbix Agent implementiert eine eigene Logik zur Kodierungskonvertierung. Daher werden diese Codepages vom Zabbix Agent unterstützt, auch wenn die nativen Windows-Funktionen sie nicht bereitstellen.

Wenn im Parameter `encoding` keine Kodierung angegeben ist, werden die folgenden Auflösungsstrategien angewendet:

- Wenn keine Kodierung angegeben ist (oder eine leere Zeichenfolge verwendet wird), wird UTF-8 angenommen und die Daten werden „unverändert“ verarbeitet;
- BOM-Analyse – anwendbar für die Datenpunkte `vfs.file.contents`, `vfs.file.regexp`, `vfs.file.regmatch`. Es wird versucht, die korrekte Kodierung anhand der Byte-Reihenfolgemarkierung (BOM) am Anfang der Datei zu bestimmen. Wenn keine BOM vorhanden ist, wird stattdessen die standardmäßige Auflösung (siehe oben) angewendet.

Fehlerbehebung bei Agent-Datenpunkten

Bei passiven Prüfungen sollte Folgendes beachtet werden, um zu verhindern, dass der Datenpunkt keinen Wert erhält, weil die Anfrage des Servers an den Agent zuerst eine Zeitüberschreitung erreicht:

- Wenn die Agent-Version älter als die Server-Version ist, muss der Wert `Timeout` in der Datenpunktkonfiguration (oder das globale Timeout) möglicherweise höher sein als der Wert `Timeout` in der Agent-Konfigurationsdatei.
- Wenn die Agent-Version neuer als die Server-Version ist, muss der Wert `Timeout` in der Server-Konfigurationsdatei möglicherweise höher sein als der Wert `Timeout` in der Agent-Konfigurationsdatei.

Zabbix Agent 2

Zabbix Agent 2 unterstützt alle Datenpunkt-Schlüssel, die für den Zabbix Agent unter Unix und Windows unterstützt werden. Diese Seite enthält Details zu den zusätzlichen Datenpunkt-Schlüsseln, die Sie nur mit Zabbix Agent 2 verwenden können, gruppiert nach dem Plugin, zu dem sie gehören.

Die Datenpunkt-Schlüssel werden ohne Parameter und zusätzliche Informationen aufgeführt. Klicken Sie auf den Datenpunkt-Schlüssel, um die vollständigen Details anzuzeigen.

Item key	Description	Plugin
ceph.df.details	Die Datennutzung des Clusters und die Verteilung auf die Pools.	Ceph (loadable)
ceph.osd.stats	Aggregierte Statistiken und Statistiken pro OSD.	
ceph.osd.discovery	Die Liste der erkannten OSDs.	
ceph.osd.dump	Die Nutzungsschwellen und Status der OSDs.	
ceph.ping	Prüft, ob eine Verbindung zu Ceph hergestellt werden kann.	
ceph.pool.discovery	Die Liste der erkannten Pools.	
ceph.status	Der Gesamtstatus des Clusters.	
docker.container_info	Niedrigstufige Informationen zu einem Container.	Docker
docker.container_stats	Die Ressourcenverbrauchsstatistiken des Containers.	
docker.containers	Gibt die Liste der Container zurück.	
docker.containers.discovery	Gibt die Liste der Container zurück. Wird für die Low-Level-Erkennung verwendet.	
docker.data.usage	Informationen zur aktuellen Datennutzung.	
docker.images	Gibt die Liste der Images zurück.	
docker.images.discovery	Gibt die Liste der Images zurück. Wird für die Low-Level-Erkennung verwendet.	
docker.info	Die Systeminformationen.	
docker.ping	Prüft, ob der Docker-Daemon aktiv ist.	
ember.get	Gibt das Ergebnis des angeforderten Geräts zurück.	Ember+ (loadable)
memcached.ping	Prüft, ob eine Verbindung aktiv ist.	Memcached
memcached.stats	Liest die Ausgabe des Befehls STATS aus.	
mongodb.cfg.discovery	Gibt eine Liste der erkannten Konfigurationsserver zurück.	MongoDB (loadable)
mongodb.collection.stats	Gibt verschiedene Speicherstatistiken für eine angegebene Collection zurück.	
mongodb.collections.discovery	Gibt eine Liste der erkannten Collections zurück.	
mongodb.collections.usage	Gibt die Nutzungsstatistiken für Collections zurück.	
mongodb.connpool.stats	Gibt Informationen zu den offenen ausgehenden Verbindungen der aktuellen Datenbankinstanz zu anderen Mitgliedern des Sharded Clusters oder Replikatsatzes zurück.	
mongodb.db.stats	Gibt die Statistiken zurück, die einen bestimmten Datenbankzustand widerspiegeln.	
mongodb.db.discovery	Gibt eine Liste der erkannten Datenbanken zurück.	

Item key	Description	Plugin
mongodb.jumbo_chunks.count	Gibt die Anzahl der Jumbo-Chunks zurück.	
mongodb.oplog.stats	Gibt den Status des Replikatsatzes zurück, basierend auf Daten, die aus dem Oplog abgefragt wurden.	
mongodb.ping	Prüft, ob eine Verbindung aktiv ist.	
mongodb.rs.config	Gibt die aktuelle Konfiguration des Replikatsatzes zurück.	
mongodb.rs.status	Gibt den Status des Replikatsatzes aus Sicht des Mitglieds zurück, auf dem die Methode ausgeführt wird.	
mongodb.server.status	Gibt den Datenbankzustand zurück.	
mongodb.sh.discovery	Gibt die Liste der im Cluster vorhandenen erkannten Shards zurück.	
mongodb.version	Gibt die Version des Datenbankservers zurück.	
mqtt.get	Abonniert ein bestimmtes Topic oder mehrere Topics (mit Platzhaltern) des angegebenen Brokers und wartet auf Veröffentlichungen.	MQTT
mssql.availability.group.get	Gibt Verfügbarkeitsgruppen zurück.	MSSQL (loadable)
mssql.custom.query	Gibt das Ergebnis einer benutzerdefinierten Abfrage zurück.	
mssql.db.get	Gibt alle verfügbaren MSSQL-Datenbanken zurück.	
mssql.job.status.get	Gibt den Status von Jobs zurück.	
mssql.last.backup.get	Gibt die Zeit der letzten Sicherung für alle Datenbanken zurück.	
mssql.local.db.get	Gibt Datenbanken zurück, die an einer Always On-Verfügbarkeitsgruppe und Replikat (primär oder sekundär) teilnehmen und sich auf dem Server befinden, zu dem die Verbindung hergestellt wurde.	
mssql.mirroring.get	Gibt Informationen zum Mirroring zurück.	
mssql.nonlocal.db.get	Gibt Datenbanken zurück, die an einer Always On-Verfügbarkeitsgruppe und Replikat (primär oder sekundär) teilnehmen und sich auf anderen Servern befinden (die Datenbank ist nicht lokal zur SQL-Server-Instanz, zu der die Verbindung hergestellt wurde).	
mssql.perfcounter.get	Gibt die Leistungsindikatoren zurück.	
mssql.ping	Prüft, ob eine Verbindung aktiv ist.	
mssql.quorum.get	Gibt Informationen zum Quorum zurück.	
mssql.quorum.member.get	Gibt die Quorum-Mitglieder zurück.	
mssql.replica.get	Gibt die Replikate zurück.	
mssql.version	Gibt die MSSQL-Version zurück.	
mysql.custom.query	Gibt das Ergebnis einer benutzerdefinierten Abfrage zurück.	MySQL
mysql.db.discovery	Gibt die Liste der MySQL-Datenbanken zurück.	
mysql.db.size	Die Datenbankgröße in Bytes.	
mysql.get_status_variables	Werte der globalen Statusvariablen.	
mysql.ping	Prüft, ob eine Verbindung aktiv ist.	
mysql.replication.discovery	Gibt die Liste der MySQL-Replikationen zurück.	
mysql.replication.get_slave_status	Der Replikationsstatus.	
mysql.version	Die MySQL-Version.	
net.dns.get	Führt eine DNS-Abfrage aus und gibt detaillierte Informationen zu DNS-Einträgen zurück.	Network
nvml.device.count	Die Anzahl der GPU-Geräte.	NVIDIA GPU (loadable)
nvml.device.decoder.utilization	Decoder-Auslastung des GPU-Geräts.	
nvml.device.ecc.mode	ECC-Modus des GPU-Geräts.	
nvml.device.encoder.stats.get	Encoder-Statistiken des GPU-Geräts.	
nvml.device.encoder.utilization	Encoder-Auslastung des GPU-Geräts.	
nvml.device.energy.consumption	Energieverbrauch des GPU-Geräts.	
nvml.device.errors.memory	Informationen zu ECC-Speicherfehlern des GPU-Geräts.	
nvml.device.errors.register	Informationen zu ECC-Registerfehlern des GPU-Geräts.	
nvml.device.fan.speed.avg	Durchschnittliche Lüftergeschwindigkeit des GPU-Geräts.	
nvml.device.get	Erkannte GPUs mit UUID und Name.	
nvml.device.graphics.frequency	Grafiktakt des GPU-Geräts.	
nvml.device.memory.bar1.get	BAR1-Speicherinformationen des GPU-Geräts.	
nvml.device.memory.fb.get	Framebuffer-Speicherinformationen des GPU-Geräts.	
nvml.device.memory.frequency	Speichertakt des GPU-Geräts.	
nvml.device.pci.utilization	PCI-Auslastung des GPU-Geräts.	
nvml.device.performance.state	Leistungszustand des GPU-Geräts.	
nvml.device.power.limit	Leistungsgrenze des GPU-Geräts.	

Item key	Description	Plugin
nvml.device.power.usage	Leistungsaufnahme des GPU-Geräts.	
nvml.device.serial	Seriennummer des GPU-Geräts.	
nvml.device.sm.frequency	Takt des Streaming-Multiprozessors des GPU-Geräts.	
nvml.device.temperature	Temperatur des GPU-Geräts.	
nvml.device.utilization	Auslastungsstatistiken des GPU-Geräts.	
nvml.device.video.frequency	Videotakt des GPU-Geräts.	
nvml.system.driver.version	Die NVIDIA-Treiberversion.	
nvml.version	Die Version der NVML-Bibliothek.	
oracle.diskgroups.stats	Gibt die Statistiken der Automatic Storage Management-(ASM-)Diskgruppen zurück.	Oracle
oracle.diskgroups.discovery	Gibt die Liste der ASM-Diskgruppen zurück.	
oracle.archive.info	Die Statistiken der Archivprotokolle.	
oracle.cdb.info	Die Informationen zu Container-Datenbanken (CDBs).	
oracle.custom.query	Das Ergebnis einer benutzerdefinierten Abfrage.	
oracle.datafiles.stats	Gibt die Statistiken der Datendateien zurück.	
oracle.db.discovery	Gibt die Liste der Datenbanken zurück.	
oracle.fra.stats	Gibt die Statistiken des Fast Recovery Area (FRA) zurück.	
oracle.instance.info	Die Instanzstatistiken.	
oracle.pdb.info	Die Informationen zu Pluggable Databases (PDBs).	
oracle.pdb.discovery	Gibt die Liste der PDBs zurück.	
oracle.pga.stats	Gibt die Statistiken des Program Global Area (PGA) zurück.	
oracle.ping	Prüft, ob eine Verbindung zu Oracle hergestellt werden kann.	
oracle.proc.stats	Gibt die Prozessstatistiken zurück.	
oracle.redolog.info	Die Protokolldateiinformationen aus der Steuerdatei.	
oracle.sga.stats	Gibt die Statistiken des System Global Area (SGA) zurück.	
oracle.sessions.stats	Gibt die Sitzungsstatistiken zurück.	
oracle.sys.metrics	Gibt eine Reihe von Systemmetriken zurück.	
oracle.sys.params	Gibt eine Reihe von Systemparameterwerten zurück.	
oracle.ts.stats	Gibt die Statistiken der Tablespaces zurück.	
oracle.ts.discovery	Gibt eine Liste der Tablespaces zurück.	
oracle.user.info	Gibt Informationen zu Oracle-Benutzern zurück.	
oracle.version	Gibt die Version des Datenbankservers zurück.	
pgsql.autovacuum.count	Die Anzahl der Autovacuum-Worker.	PostgreSQL (loadable)
pgsql.archive	Informationen zu archivierten Dateien.	
pgsql.bgwriter	Die kombinierte Anzahl der Checkpoints für den Datenbankcluster, aufgeschlüsselt nach Checkpoint-Typ.	
pgsql.cache.hit	Die Trefferquote des PostgreSQL-Buffer-Cache.	
pgsql.connections	Gibt Verbindungen nach Typ zurück.	
pgsql.custom.query	Gibt das Ergebnis einer benutzerdefinierten Abfrage zurück.	
pgsql.db.age	Das Alter des ältesten FrozenXID der Datenbank.	
pgsql.db.bloating_tables	Die Anzahl der aufgeblähten Tabellen pro Datenbank.	
pgsql.db.discovery	Die Liste der PostgreSQL-Datenbanken.	
pgsql.db.size	Die Datenbankgröße in Bytes.	
pgsql.dbstat	Sammelt die Statistiken pro Datenbank.	
pgsql.dbstat.sum	Die zusammengefassten Daten für alle Datenbanken in einem Cluster.	
pgsql.locks	Informationen zu gewährten Sperren pro Datenbank.	
pgsql.oldest.xid	Das Alter der ältesten XID.	
pgsql.ping	Prüft, ob eine Verbindung aktiv ist.	
pgsql.queries	Abfragemetriken nach Ausführungszeit.	
pgsql.replication.count	Die Anzahl der Standby-Server.	
pgsql.replication.process	Der Flush-Lag, Write-Lag und Replay-Lag pro Senderprozess.	
pgsql.replication.process.discovery	Die Erkennung des Replikationsprozessnamens.	
pgsql.replication.recovery_role	Der Wiederherstellungsstatus.	
pgsql.replication.status	Der Status der Replikation.	
pgsql.replication_lag.b	Die Replikationsverzögerung in Bytes.	
pgsql.replication_lag.sec	Die Replikationsverzögerung in Sekunden.	
pgsql.uptime	Die PostgreSQL-Laufzeit in Millisekunden.	
pgsql.version	Gibt die PostgreSQL-Version zurück.	
pgsql.wal.stat	Die WAL-Statistiken.	
redis.config	Liest die Konfigurationsparameter einer Redis-Instanz, die dem Muster entsprechen.	Redis

Item key	Description	Plugin
redis.info	Liest die Ausgabe des Befehls INFO aus.	
redis.ping	Prüft, ob eine Verbindung aktiv ist.	
redis.slowlog.count	Die Anzahl der Slowlog-Einträge seit dem Start von Redis.	
smart.attribute.discovery	Gibt eine Liste der S.M.A.R.T.-Geräteattribute zurück.	S.M.A.R.T.
smart.disk.discovery	Gibt eine Liste der S.M.A.R.T.-Geräte zurück.	
smart.disk.get	Gibt alle verfügbaren Eigenschaften der S.M.A.R.T.-Geräte zurück.	
systemd.unit.get	Gibt alle Eigenschaften einer systemd-Einheit zurück.	Systemd
systemd.unit.info	Informationen zur systemd-Einheit.	
systemd.unit.discovery	Die Liste der systemd-Einheiten und ihre Details.	
web.certificate.get	Validiert die Zertifikate und gibt Zertifikatsdetails zurück.	Web certificates

Details zum Datenpunktschlüssel

Parameter ohne spitze Klammern sind obligatorisch. Parameter, die mit spitzen Klammern **< >** gekennzeichnet sind, sind optional.

ceph.df.details[connString,<user>,<apikey>]

 Die Datennutzung des Clusters und die Verteilung auf die Pools.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - die URI oder der Sitzungsname;
- **user, apikey** - der Benutzername und der API-Schlüssel für die Verbindung zu Ceph.

ceph.osd.stats[connString,<user>,<apikey>]

 Aggregierte und OSD-spezifische Statistiken.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - die URI oder der Sitzungsname;
- **user, apikey** - der Benutzername und der API-Schlüssel für die Verbindung zu Ceph.

ceph.osd.discovery[connString,<user>,<apikey>]

 Die Liste der erkannten OSDs. Wird für die Low-Level-Discovery verwendet.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - die URI oder der Sitzungsname;
- **user, apikey** - der Benutzername und der API-Schlüssel für die Verbindung zu Ceph.

ceph.osd.dump[connString,<user>,<apikey>]

 Die Nutzungsschwellenwerte und Status von OSDs.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - die URI oder der Sitzungsname;
- **user, apikey** - der Benutzername und der API-Schlüssel für die Verbindung zu Ceph.

ceph.ping[connString,<user>,<apikey>]

 Prüft, ob eine Verbindung zu Ceph hergestellt werden kann.
 Rückgabewert: *0* - die Verbindung ist unterbrochen (wenn ein Fehler auftritt, einschließlich AUTH- und Konfigurationsproblemen); *1* - die Verbindung ist erfolgreich.

Parameter:

- **connString** - die URI oder der Sitzungsname;
- **user, apikey** - der Benutzername und der API-Schlüssel für die Verbindung zu Ceph.

ceph.pool.discovery[connString,<user>,<apikey>]

 Die Liste der erkannten Pools. Wird für Low-Level-Discovery verwendet.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - die URI oder der Sitzungsname;
- **user, apikey** - der Benutzername und der API-Schlüssel für die Verbindung zu Ceph.

ceph.status[connString,<user>,<apikey>]

 Der Gesamtstatus des Clusters.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - die URI oder der Sitzungsname;
- **user, apikey** - der Benutzername und der API-Schlüssel für die Verbindung zu Ceph.

docker.container_info[<ID>,<info>]

 Detaillierte Informationen über einen Container.
 Rückgabewert: Die Ausgabe des API-Aufrufs ContainerInspect, als JSON serialisiert.

Parameter:

- **ID** - die ID oder der Name des Containers;
- **info** - der Umfang der zurückgegebenen Informationen. Unterstützte Werte: *short* (Standard) oder *full*.

Der Benutzer des Agent 2 (’zabbix’) muss der group ’docker’ hinzugefügt werden, damit ausreichende Berechtigungen vorhanden sind. Andernfalls schlägt die Prüfung fehl.

docker.container_stats[<ID>]

 Die Statistiken zur Ressourcennutzung des Containers.
 Rückgabewert: Die Ausgabe des API-Aufrufs ContainerStats und der CPU-Auslastungsprozentsatz, als JSON serialisiert.

Parameter:

- **ID** - die ID oder der Name des Containers.

Der Agent 2-Benutzer (’zabbix’) muss der ’docker’-group hinzugefügt werden, damit ausreichende Berechtigungen vorhanden sind. Andernfalls schlägt die Prüfung fehl.

docker.containers

 Die Liste der Container.
 Rückgabewert: Die Ausgabe des API-Aufrufs ContainerList, serialisiert als JSON.

Der Agent 2-Benutzer (’zabbix’) muss der group ’docker’ hinzugefügt werden, damit ausreichende Berechtigungen vorhanden sind. Andernfalls schlägt die Prüfung fehl.

docker.containers.discovery[<options>]

 Gibt die Liste der Container zurück. Wird für die Low-Level-Discovery verwendet.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **options** - gibt an, ob alle oder nur laufende Container erkannt werden sollen. Unterstützte Werte: *true* - alle Container zurückgeben; *false* - nur laufende Container zurückgeben (Standard).

Der Agent-2-Benutzer (’zabbix’) muss der Gruppe ’docker’ hinzugefügt werden, damit ausreichende Berechtigungen vorhanden sind. Andernfalls schlägt die Prüfung fehl.

docker.data.usage

 Informationen über die aktuelle Datennutzung.
 Rückgabewert: Die Ausgabe des API-Aufrufs SystemDataUsage, serialisiert als JSON.

Der Agent-2-Benutzer (’zabbix’) muss der group ’docker’ hinzugefügt werden, um ausreichende Berechtigungen zu haben. Andernfalls schlägt die Prüfung fehl.

docker.images

 Gibt die Liste der Images zurück.
 Rückgabewert: Die Ausgabe des API-Aufrufs ImageList, als JSON serialisiert.

Der Agent 2-Benutzer (’zabbix’) muss der ’docker’-group hinzugefügt werden, um über ausreichende Berechtigungen zu verfügen. Andernfalls schlägt die Prüfung fehl.

docker.images.discovery

 Gibt die Liste der Images zurück. Wird für die Low-Level-Discovery verwendet.
 Rückgabewert: *JSON-Objekt*.

Der Agent-2-Benutzer (’zabbix’) muss der Gruppe ’docker’ hinzugefügt werden, damit ausreichende Berechtigungen vorhanden sind. Andernfalls schlägt die Prüfung fehl.

docker.info

 Die Systeminformationen.
 Rückgabewert: Die Ausgabe des API-Aufrufs SystemInfo, serialisiert als JSON.

Der Agent-2-Benutzer (’zabbix’) muss der group ’docker’ hinzugefügt werden, damit ausreichende Berechtigungen vorhanden sind. Andernfalls schlägt die Prüfung fehl.

docker.ping

 Prüft, ob der Docker-Daemon aktiv ist oder nicht.
 Rückgabewert: *1* - die Verbindung ist aktiv; *0* - die Verbindung ist unterbrochen.

Der Agent 2-Benutzer (’zabbix’) muss der ’docker’-group hinzugefügt werden, um über ausreichende Berechtigungen zu verfügen. Andernfalls schlägt die Prüfung fehl.

ember.get[<uri>,<path>]

 Gibt das Ergebnis des angeforderten Geräts zurück.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **uri** - URI des Ember+-Geräts. Standard: 127.0.0.1:9998;
- **path** - OID-Pfad zum Gerät. Standardmäßig leer; gibt Daten der Root-Collection zurück.

memcached.ping[connString,<user>,<password>]

 Prüft, ob eine Verbindung aktiv ist oder nicht.
 Rückgabewert: *1* - die Verbindung ist aktiv; *0* - die Verbindung ist unterbrochen (wenn ein beliebiger Fehler vorliegt, einschließlich AUTH- und Konfigurationsproblemen).

Parameter:

- **connString** - die URI oder der Sitzungsname;
- **user, password** - die Memcached-Anmeldedaten.

memcached.stats[connString,<user>,<password>,<type>]

 Ruft die Ausgabe des Befehls STATS ab.
 Rückgabewert: *JSON* - die Ausgabe wird als JSON serialisiert.

Parameter:

- **connString** - die URI oder der Sitzungsname;
- **user, password** - die Memcached-Anmeldeinformationen;
- **type** - der zurückzugebende Statistiktyp: *items*, *sizes*, *slabs* oder *settings* (standardmäßig leer, gibt allgemeine Statistiken zurück).

mongodb.collection.stats[connString,<user>,<password>,<database>,collection]

 Gibt verschiedene Speicherstatistiken für eine angegebene Collection zurück.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - der URI- oder Sitzungsname;
- **user, password** - die MongoDB-Anmeldeinformationen;
- **database** - der Datenbankname (Standard: admin);
- **collection** - der Name der Collection.

mongodb.cfg.discovery[connString,<user>,<password>]

 Gibt eine Liste der erkannten Konfigurations-Server zurück.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - die URI oder der Sitzungsname;
- **user, password** - die MongoDB-Anmeldeinformationen.

mongodb.collections.discovery[connString,<user>,<password>]

 Gibt eine Liste der erkannten Collections zurück. Wird für Low-Level-Discovery verwendet.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - die URI oder der Sitzungsname;
- **user, password** - die MongoDB-Anmeldedaten.

mongodb.collections.usage[connString,<user>,<password>]

 Gibt die Nutzungsstatistiken für Collections zurück.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - die URI oder der Sitzungsname;
- **user, password** - die MongoDB-Anmeldedaten.

mongodb.connpool.stats[connString,<user>,<password>]

 Gibt Informationen zu den offenen ausgehenden Verbindungen von der aktuellen Datenbankinstanz zu anderen Mitgliedern des Sharded-Clusters oder Replica-Sets zurück.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - die URI oder der Sitzungsname;
- **user, password** - die MongoDB-Anmeldedaten.

mongodb.db.stats[connString,<user>,<password>,<database>]

 Gibt die Statistiken zurück, die den Zustand eines bestimmten Datenbanksystems widerspiegeln.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - die URI oder der Sitzungsname;
- **user, password** - die MongoDB-Anmeldeinformationen;
- **database** - der Datenbankname (Standard: admin).

mongodb.db.discovery[connString,<user>,<password>]

 Gibt eine Liste der erkannten Datenbanken zurück. Wird für die Low-Level-Discovery verwendet.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - der URI- oder Sitzungsname;
- **user, password** - die MongoDB-Anmeldeinformationen.

mongodb.jumbo_chunks.count[connString,<user>,<password>]

 Gibt die Anzahl der Jumbo-Chunks zurück.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - die URI oder der Sitzungsname;
- **user, password** - die MongoDB-Anmeldedaten.

mongodb.oplog.stats[connString,<user>,<password>]

 Gibt den Status des Replica-Sets zurück, wobei Daten verwendet werden, die aus dem Oplog abgefragt werden.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - die URI oder der Sitzungsname;
- **user, password** - die MongoDB-Anmeldeinformationen.

mongodb.ping[connString,<user>,<password>]

 Prüft, ob eine Verbindung aktiv ist oder nicht.
 Rückgabewert: *1* - die Verbindung ist aktiv; *0* - die Verbindung ist unterbrochen (wenn ein Fehler vorliegt, einschließlich AUTH- und Konfigurationsproblemen).

Parameter:

- **connString** - die URI oder der Sitzungsname;
- **user, password** - die MongoDB-Anmeldedaten.

mongodb.rs.config[connString,<user>,<password>]

 Gibt die aktuelle Konfiguration des Replica-Sets zurück.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - die URI oder der Sitzungsname;
- **user, password** - die MongoDB-Anmeldedaten.

mongodb.rs.status[connString,<user>,<password>]

 Gibt den Status des Replikatsatzes aus der Sicht des Mitglieds zurück, auf dem die Methode ausgeführt wird.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - die URI oder der Sitzungsname;
- **user, password** - die MongoDB-Anmeldedaten.

mongodb.server.status[connString,<user>,<password>]

 Gibt den Datenbankstatus zurück.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - die URI oder der Sitzungsname;
- **user, password** - die MongoDB-Anmeldedaten.

mongodb.sh.discovery[connString,<user>,<password>]

 Gibt die Liste der im Cluster vorhandenen erkannten Shards zurück.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - die URI oder der Sitzungsname;
- **user, password** - die MongoDB-Anmeldedaten.

mongodb.version[connString,<user>,<password>]

 Gibt die Version des Datenbank-Servers zurück.
 Rückgabewert: *String*.

Parameter:

- **connString** - der URI- oder Sitzungsname;
- **user, password** - die MongoDB-Anmeldeinformationen.

mqtt.get[<broker url>,topic,<user>,<password>]

 Abonniert ein bestimmtes Topic oder mehrere Topics (mit Platzhaltern) des angegebenen Brokers und wartet auf Veröffentlichungen.
 Rückgabewert: Abhängig vom Inhalt des Topics. Wenn Platzhalter verwendet werden, wird der Topic-Inhalt als JSON zurückgegeben.

Parameter:

- **broker url** - die MQTT-Broker-URL im Format `protocol://host:port` ohne Abfrageparameter (unterstützte Protokolle: `tcp`, `ssl`, `ws`). Wenn kein Wert angegeben ist, verwendet der Agent `tcp://localhost:1883`. Wenn ein Protokoll oder Port weggelassen wird, werden das Standardprotokoll (`tcp`) bzw. der Standardport (1883) verwendet;
- **topic** - das MQTT-Topic (erforderlich). Platzhalter (+,#) werden unterstützt;
- **user, password** - die Authentifizierungsdaten (falls erforderlich).

Kommentare:

- Der Datenpunkt muss als active check (Datenpunkt-Typ „Zabbix agent (active)“) konfiguriert werden;
- TLS-Verschlüsselungszertifikate können verwendet werden, indem sie an einem Standardort gespeichert werden (z. B. im Verzeichnis `/etc/ssl/certs/` unter Ubuntu). Für TLS verwenden Sie das Schema `tls://`.

mssql.availability.group.get[URI,<user>,<password>]

 Gibt Verfügbarkeitsgruppen zurück.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **URI** - URI des MSSQL-Servers (das einzige unterstützte Schema ist `sqlserver://`). Eingebettete Zugangsdaten werden ignoriert. Es ist möglich, einen Instanznamen als Teil der URI anzugeben, z. B.: `sqlserver://localhost/InstanceName` (kein Port). Wenn ein Port angegeben ist, wird der Instanzname ignoriert;
- **user, password** - Benutzername, Passwort, die an den geschützten MSSQL-Server gesendet werden.

Weitere Informationen finden Sie in der README des MSSQL-Plugins.

mssql.custom.query[URI,<user>,<password>,queryName,<args...>]

 Gibt das Ergebnis einer benutzerdefinierten Abfrage zurück.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **URI** - URI des MSSQL-Servers (das einzige unterstützte Schema ist `sqlserver://`). Eingebettete Zugangsdaten werden ignoriert. Es ist möglich, einen Instanznamen als Teil der URI anzugeben, z. B.: `sqlserver://localhost/InstanceName` (kein Port). Wenn ein Port angegeben ist, wird der Instanzname ignoriert;
- **user, password** - Benutzername, Passwort, die an den geschützten MSSQL-Server gesendet werden;
- **queryName** - Name einer benutzerdefinierten Abfrage, die in `Plugins.MSSQL.CustomQueriesDir` ohne die Erweiterung .sql konfiguriert ist;
- **args** - ein oder mehrere durch Kommas getrennte Argumente, die an eine Abfrage übergeben werden.

Weitere Informationen finden Sie in der README des MSSQL-Plugins.

mssql.db.get

 Gibt alle verfügbaren MSSQL-Datenbanken zurück.
 Rückgabewert: *JSON-Objekt*.

Weitere Informationen finden Sie in der Readme des MSSQL-Plugins.

mssql.job.status.get

 Gibt den Status von Jobs zurück.
 Rückgabewert: *JSON-Objekt*.

Weitere Informationen finden Sie in der README zum MSSQL-Plugin.

mssql.last.backup.get

 Gibt die Zeit der letzten Sicherung für alle Datenbanken zurück.
 Rückgabewert: *JSON-Objekt*.

Weitere Informationen finden Sie in der Readme zum MSSQL plugin.

mssql.local.db.get

 Gibt Datenbanken zurück, die an einer Always On-Verfügbarkeitsgruppe und einem Replikat (primär oder sekundär) teilnehmen und sich auf dem Server befinden, zu dem die Verbindung hergestellt wurde.
 Rückgabewert: *JSON-Objekt*.

Weitere Informationen finden Sie in der Readme des MSSQL-Plugins.

mssql.mirroring.get

 Gibt Informationen zur Spiegelung zurück.
 Rückgabewert: *JSON-Objekt*.

Weitere Informationen finden Sie in der Readme des MSSQL-Plugins.

mssql.nonlocal.db.get

 Gibt Datenbanken zurück, die an einer Always On-Verfügbarkeitsgruppe und einem Replikat (primär oder sekundär) beteiligt sind, die sich auf anderen Servern befinden (die Datenbank ist nicht lokal für die SQL Server-Instanz, zu der die Verbindung hergestellt wurde).
 Rückgabewert: *JSON-Objekt*.

Weitere Informationen finden Sie in der Readme des MSSQL-Plugins.

mssql.perfcounter.get

 Gibt die Leistungsindikatoren zurück.
 Rückgabewert: *JSON-Objekt*.

Weitere Informationen finden Sie in der README zum MSSQL-Plugin.

mssql.ping

 Die Datenbank anpingen. Prüfen, ob die Verbindung korrekt konfiguriert ist.
 Rückgabewert: *1* - erreichbar, *0* - nicht erreichbar.

Weitere Informationen finden Sie in der README des MSSQL-Plugins.

mssql.quorum.get

 Gibt die Quorumsinformationen zurück.
 Rückgabewert: *JSON-Objekt*.

Weitere Informationen finden Sie in der README des MSSQL-Plugins.

mssql.quorum.member.get

 Gibt die Quorum-Mitglieder zurück.
 Rückgabewert: *JSON-Objekt*.

Weitere Informationen finden Sie in der README des MSSQL-Plugins.

mssql.replica.get

 Gibt die Replikate zurück.
 Rückgabewert: *JSON-Objekt*.

Weitere Informationen finden Sie in der Readme des MSSQL plugin.

mssql.version

 Gibt die MSSQL-Version zurück.
 Rückgabewert: *String*.

Weitere Informationen finden Sie in der README zum MSSQL plugin.

mysql.custom.query[connString,<user>,<password>,queryName,<args...>]

 Gibt das Ergebnis einer benutzerdefinierten Abfrage zurück.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - URI oder Sitzungsname;
- **user, password** - MySQL-Anmeldedaten;

- **queryName** - Name einer benutzerdefinierten Abfrage; muss mit dem SQL-Dateinamen ohne Erweiterung übereinstimmen;
- **args** - ein oder mehrere durch Kommas getrennte Argumente, die an eine Abfrage übergeben werden.

Weitere Informationen finden Sie in der README des MySQL plugin, Abschnitt *Benutzerdefinierte Abfragen*.

mysql.db.discovery[connString,<user>,<password>]

 Gibt die Liste der MySQL-Datenbanken zurück. Wird für die Low-Level-Discovery verwendet.
 Rückgabewert: Das Ergebnis der SQL-Abfrage „show databases“ im LLD-JSON-Format.

Parameter:

- **connString** - die URI oder der Sitzungsname;
- **user, password** - die MySQL-Anmeldedaten.

mysql.db.size[connString,<user>,<password>,<database name>]

 Die Datenbankgröße in Byte.
 Rückgabewert: Ergebnis der SQL-Abfrage „select coalesce(sum(data_length + index_length),0) as size from information_schema.tables where table_schema=?“ für die angegebene Datenbank in Byte.

Parameter:

- **connString** - der URI- oder Sitzungsname;
- **user, password** - die MySQL-Anmeldeinformationen;
- **database name** - der Name der Datenbank.

mysql.get_status_variables[connString,<user>,<password>]

 Werte der globalen Statusvariablen.
 Rückgabewert: Ergebnis der SQL-Abfrage ”show global status” im JSON-Format.

Parameter:

- **connString** - der URI- oder Sitzungsname;
- **user, password** - die MySQL-Anmeldedaten.

mysql.ping[connString,<user>,<password>]

 Prüft, ob eine Verbindung aktiv ist oder nicht.
 Rückgabewert: *1* - die Verbindung ist aktiv; *0* - die Verbindung ist unterbrochen (wenn ein Fehler auftritt, einschließlich AUTH- und Konfigurationsproblemen).

Parameter:

- **connString** - die URI oder der Sitzungsname;
- **user, password** - die MySQL-Anmeldedaten.

mysql.replication.discovery[connString,<user>,<password>]

 Gibt die Liste der MySQL-Replikationen zurück. Wird für Low-Level-Discovery verwendet.
 Rückgabewert: das Ergebnis der SQL-Abfrage `SHOW SLAVE STATUS` oder `SHOW REPLICA STATUS` im LLD-JSON-Format.

Parameter:

- **connString** - die URI oder der Sitzungsname;
- **user, password** - die MySQL-Anmeldeinformationen.

mysql.replication.get_slave_status[connString,<user>,<password>,<master host>]

 Der Replikationsstatus.
 Rückgabewert: Ergebnis der SQL-Abfrage `SHOW SLAVE STATUS` oder `SHOW REPLICA STATUS` im JSON-Format.

Parameter:

- **connString** - der URI- oder Sitzungsname;
- **user, password** - die MySQL-Anmeldeinformationen;
- **master host** - der Hostname des Replikations-Masters. Wenn keiner gefunden wird, wird ein Fehler zurückgegeben. Wenn dieser Parameter nicht angegeben wird, werden alle Hosts zurückgegeben.

mysql.version[connString,<user>,<password>]

 Die MySQL-Version.
 Rückgabewert: *String* (mit der Version der MySQL-Instanz).

Parameter:

- **connString** - die URI oder der Sitzungsname;
- **user, password** - die MySQL-Anmeldeinformationen.

net.dns.get[<ip>,name,<type>,<timeout>,<count>,<protocol>,"<flags>"]

Führt eine DNS-Abfrage aus und gibt detaillierte Informationen zu DNS-Einträgen zurück.
 Dieser Datenpunkt ist eine erweiterte Version des Zabbix-Agent-Datenpunkts `net.dns.record` mit mehr unterstützten Eintragstypen und anpassbaren Flags.
 Rückgabewerte: *JSON object*

Parameter:

- **ip** - die IP-Adresse des DNS-Servers (leer lassen für den Standard-DNS-Server);
- **name** - der abzufragende DNS-Name;
- **type** - der abzufragende Eintragstyp (Standard ist *SOA*);
- **timeout** - die Zeitüberschreitung für die Anfrage in Sekunden (Standard ist 1 Sekunde);
- **count** - die Anzahl der Versuche für die Anfrage (Standard ist 2);
- **protocol** - das für DNS-Abfragen verwendete Protokoll: *udp* (Standard) oder *tcp*;
- **flags** - ein oder mehrere durch Kommas getrennte Argumente, die an eine Abfrage übergeben werden.

Kommentare:

- Die möglichen Werte für `type` sind: *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*. Beachten Sie, dass Werte nur in Großbuchstaben angegeben werden dürfen; Kleinbuchstaben oder gemischte Schreibweise werden nicht unterstützt.
- Für Reverse-DNS-Abfragen (wenn `type` auf *PTR* gesetzt ist) können Sie den DNS-Namen sowohl im umgekehrten als auch im nicht umgekehrten Format angeben (siehe Beispiele unten). Beachten Sie, dass bei einer PTR-Eintrag-Abfrage der DNS-Name tatsächlich eine IP-Adresse ist.
- Die möglichen Werte für `flags` sind:
  - *cdflag* oder *nocdflag* (Standard);
  - *rdflag* (Standard) oder *nordflag*;
  - *dnssec* oder *nodnssec* (Standard);
  - *nsid* oder *nonsid* (Standard);
  - *edns0* (Standard) oder *noedns0*;
  - *aaflag* oder *noaaflag* (Standard);
  - *adflag* oder *noadflag* (Standard). Die `flags` *dnssec* und *nsid* können nicht zusammen mit *noedns0* verwendet werden, da beide *edns0* erfordern. Beachten Sie, dass Werte nur in Kleinbuchstaben angegeben werden dürfen; Großbuchstaben oder gemischte Schreibweise werden nicht unterstützt.
- Internationalisierte Domainnamen werden nicht unterstützt, bitte verwenden Sie stattdessen IDNA-kodierte Namen.
- Die Ausgabe ist ein Objekt, das auf Grundlage der angegebenen Parameter Informationen zu DNS-Einträgen enthält (siehe weitere Details).

Beispiele:

```
net.dns.get[192.0.2.1,example.com,DNSKEY,3,3,tcp,"cdflag,rdflag,nsid"]

net.dns.get[,198.51.100.1,PTR,,,,"cdflag,rdflag,nsid"]
net.dns.get[,1.100.51.198.in-addr.arpa,PTR,,,,"cdflag,rdflag,nsid"]

net.dns.get[,2a00:1450:400f:800::200e,PTR,,,,"cdflag,rdflag,nsid"]
net.dns.get[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR,,,,"cdflag,rdfla
```

nvml.device.count

 Die Anzahl der GPU-Geräte.
 Rückgabewert: *Integer*.

Weitere Informationen finden Sie in der README des NVIDIA GPU plugin.

nvml.device.decoder.utilization[<deviceUUID>]

 GPU-Geräte-Decoder-Auslastung in Prozent.
 Rückgabewert: *Integer*.

Parameter:

- **deviceUUID** - UUID des GPU-Geräts.

Weitere Informationen finden Sie in der README des NVIDIA GPU plugin.

nvml.device.ecc.mode[<deviceUUID>]

 Informationen zum ECC-Modus des GPU-Geräts (aktuell, ausstehend).
 Rückgabewert: *JSON*.

Parameter:

- **deviceUUID** - UUID des GPU-Geräts.

Weitere Informationen finden Sie in der README zum NVIDIA GPU plugin.

nvml.device.encoder.stats.get[<deviceUUID>]

 Encoder-Statistiken des GPU-Geräts.
 Rückgabewert: *JSON*.

Parameter:

- **deviceUUID** - UUID des GPU-Geräts.

Weitere Informationen finden Sie in der README des NVIDIA GPU plugin.

nvml.device.encoder.utilization[<deviceUUID>]

 Auslastung des GPU-Geräte-Encoders in Prozent.
 Rückgabewert: *Integer*.

Parameter:

- **deviceUUID** - UUID des GPU-Geräts.

Weitere Informationen finden Sie in der Readme des NVIDIA GPU plugin.

nvml.device.energy.consumption[<deviceUUID>]

 Gesamtenergieverbrauch des GPU-Geräts in Millijoule (mJ), seit der Treiber zuletzt neu geladen wurde.
 Rückgabewert: *Integer*.

Parameter:

- **deviceUUID** - UUID des GPU-Geräts.

Weitere Informationen finden Sie in der README des NVIDIA GPU plugin.

nvml.device.errors.memory[<deviceUUID>]

 GPU-Geräteinformationen zu ECC-Speicherfehlern (korrigiert, nicht korrigiert).
 Rückgabewert: *JSON*.

Parameter:

- **deviceUUID** - UUID des GPU-Geräts.

Weitere Informationen finden Sie in der Readme des NVIDIA GPU plugin.

nvml.device.errors.register[<deviceUUID>]

 GPU-Geräte-ECC-Registerfehlerinformationen (korrigiert, unkorrigiert).
 Rückgabewert: *JSON*.

Parameter:

- **deviceUUID** - UUID des GPU-Geräts.

Weitere Informationen finden Sie in der README zum NVIDIA GPU plugin.

nvml.device.fan.speed.avg[<deviceUUID>]

 Durchschnittliche Lüftergeschwindigkeit des GPU-Geräts als Prozentsatz der maximalen Geschwindigkeit.
 Rückgabewert: *Integer*.

Parameter:

- **deviceUUID** - UUID des GPU-Geräts.

Weitere Informationen finden Sie in der README des NVIDIA GPU plugin.

nvml.device.get

 Erkannte GPUs mit UUID und Namen.
 Rückgabewert: *JSON*.

Weitere Informationen finden Sie in der README des NVIDIA GPU plugin.

nvml.device.graphics.frequency[<deviceUUID>]

 Grafiktaktfrequenz des GPU-Geräts in MHz.
 Rückgabewert: *Integer*.

Parameter:

- **deviceUUID** - UUID des GPU-Geräts.

Weitere Informationen finden Sie in der README zum NVIDIA GPU plugin.

nvml.device.memory.fb.get[<deviceUUID>]

 Statistiken zum Framebuffer-Speicher des GPU-Geräts (gesamt, reserviert, frei, verwendet).
 Rückgabewert: *JSON*.

Parameter:

- **deviceUUID** - UUID des GPU-Geräts.

Weitere Informationen finden Sie in der Readme des NVIDIA GPU plugin.

nvml.device.memory.bar1.get[<deviceUUID>]

 GPU-Geräte-BAR1-Speicherstatistiken (gesamt, frei, verwendet).
 Rückgabewert: *JSON*.

Parameter:

- **deviceUUID** - UUID des GPU-Geräts.

Weitere Informationen finden Sie in der README zum NVIDIA GPU plugin.

nvml.device.memory.frequency[<deviceUUID>]

 GPU-Gerätespeichertakt in MHz.
 Rückgabewert: *Integer*.

Parameter:

- **deviceUUID** - GPU-Geräte-UUID.

Weitere Informationen finden Sie in der Readme zum NVIDIA GPU plugin.

nvml.device.pci.utilization[<deviceUUID>]

 PCI-Auslastung des GPU-Geräts (Sende-/Empfangsdurchsatz in KBps).
 Rückgabewert: *JSON*.

Parameter:

- **deviceUUID** - UUID des GPU-Geräts.

Weitere Informationen finden Sie in der README des NVIDIA GPU plugin.

nvml.device.performance.state[<deviceUUID>]

 Leistungszustand des GPU-Geräts.
 Rückgabewert: *Integer*.

Parameter:

- **deviceUUID** - UUID des GPU-Geräts.

Weitere Informationen finden Sie in der Readme des NVIDIA GPU plugin.

nvml.device.power.limit[<deviceUUID>]

 Leistungsgrenze des GPU-Geräts in Milliwatt.
 Rückgabewert: *Integer*.

Parameter:

- **deviceUUID** - UUID des GPU-Geräts.

Weitere Informationen finden Sie in der Readme zum NVIDIA GPU plugin.

nvml.device.power.usage[<deviceUUID>]

 Aktuelle Leistungsaufnahme des GPU-Geräts in Milliwatt.
 Rückgabewert: *Integer*.

Parameter:

- **deviceUUID** - UUID des GPU-Geräts.

Weitere Informationen finden Sie in der README zum NVIDIA GPU plugin.

nvml.device.serial[<deviceUUID>]

 Seriennummer des GPU-Geräts.
 Rückgabewert: *String*.

Parameter:

- **deviceUUID** - UUID des GPU-Geräts.

Weitere Informationen finden Sie in der README des NVIDIA GPU plugin.

nvml.device.sm.frequency[<deviceUUID>]

 Taktfrequenz des Streaming-Multiprozessors des GPU-Geräts in MHz.
 Rückgabewert: *Integer*.

Parameter:

- **deviceUUID** - UUID des GPU-Geräts.

Weitere Informationen finden Sie in der README des NVIDIA GPU plugin.

nvml.device.temperature[<deviceUUID>]

 Temperatur des GPU-Geräts in Grad Celsius.
 Rückgabewert: *Integer*.

Parameter:

- **deviceUUID** - UUID des GPU-Geräts.

Weitere Informationen finden Sie in der README des NVIDIA GPU plugin.

nvml.device.utilization[<deviceUUID>]

 Statistiken zur GPU-Geräteauslastung (GPU-/Speicherauslastung in Prozent).
 Rückgabewert: *JSON*.

Parameter:

- **deviceUUID** - GPU-Geräte-UUID.

Weitere Informationen finden Sie in der README des NVIDIA GPU plugin.

nvml.device.video.frequency[<deviceUUID>]

 GPU-Geräte-Videotaktfrequenz in MHz.
 Rückgabewert: *Integer*.

Parameter:

- **deviceUUID** - UUID des GPU-Geräts.

Weitere Informationen finden Sie in der Readme zum NVIDIA GPU plugin.

nvml.system.driver.version

 Die NVIDIA-Treiberversion.
 Rückgabewert: *String*.

Weitere Informationen finden Sie in der README zum NVIDIA GPU plugin.

nvml.version

 Die Version der NVML-Bibliothek.
 Rückgabewert: *String*.

Weitere Informationen finden Sie in der README zum NVIDIA GPU plugin.

oracle.diskgroups.stats[connString,<user>,<password>,<service>,<diskgroup>]

 Gibt die Statistiken der Automatic Storage Management (ASM)-Datenträgergruppen zurück.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - kann Folgendes sein:
  - die URI - wenn keine Sitzung mit dem angegebenen Namen gefunden wird und `Plugins.Oracle.ResolveTNS` auf false gesetzt ist. Wenn sie ein Schema (z. B. "tcp://"), einen Port (z. B. 1521) oder beides enthält, wird die Option ResolveTNS nicht berücksichtigt und sie wird in jedem Fall als URI behandelt;
  - Sitzungsname - wenn ein solcher Name in der Plugin-Konfigurationsdatei gefunden wird;
  - TNS-Schlüssel - wenn die Option ResolveTNS auf true gesetzt ist und keine der oben genannten Bedingungen zutrifft;
  - TNS-Wert - wenn er mit der öffnenden Klammer "(" beginnt (führende Leerzeichen werden ignoriert). In diesem Fall wird die Option ResolveTNS nicht berücksichtigt.
- **user** - der Oracle-Benutzername; unterstützt das Anhängen einer der Anmeldeoptionen `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oder `as sysrac` im Format `user as sysdba` (bei einer Anmeldeoption wird die Groß-/Kleinschreibung nicht beachtet und sie darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Name des Oracle-Service;
- **diskgroup** - der Name der ASM-Datenträgergruppe, die abgefragt werden soll.

oracle.diskgroups.discovery[connString,<user>,<password>,<service>]

 Gibt die Liste der ASM-Datenträgergruppen zurück. Wird für Low-Level-Discovery verwendet.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - kann Folgendes sein:
  - die URI - wenn keine Sitzung mit dem angegebenen Namen gefunden wird und `Plugins.Oracle.ResolveTNS` auf false gesetzt ist. Wenn sie ein Schema (z. B. "tcp://"), einen Port (z. B. 1521) oder beides enthält, wird die Option ResolveTNS nicht berücksichtigt und sie wird dennoch als URI behandelt;
  - Sitzungsname - wenn ein solcher Name in der Plugin-Konfigurationsdatei gefunden wird;
  - TNS-Schlüssel - wenn die Option ResolveTNS auf true gesetzt ist und keine der oben genannten Bedingungen zutrifft;
  - TNS-Wert - wenn er mit der öffnenden Klammer "(" beginnt (führende Leerzeichen werden ignoriert). In diesem Fall wird die Option ResolveTNS nicht berücksichtigt.
- **user** - der Oracle-Benutzername; unterstützt das Anhängen einer der Anmeldeoptionen `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oder `as sysrac` im Format `user as sysdba` (eine Anmeldeoption unterscheidet nicht zwischen Groß- und Kleinschreibung und darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Oracle-Service-Name.

oracle.archive.info[connString,<user>,<password>,<service>,<destination>]

 Die Statistik der Archivprotokolle.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - kann Folgendes sein:
  - die URI - wenn keine Sitzung mit dem angegebenen Namen gefunden wird und `Plugins.Oracle.ResolveTNS` auf false gesetzt ist. Wenn sie ein Schema (z. B. "tcp://"), einen Port (z. B. 1521) oder beides enthält, wird die Option ResolveTNS nicht berücksichtigt und sie wird dennoch als URI behandelt;
  - Sitzungsname - wenn ein solcher Name in der Plugin-Konfigurationsdatei gefunden wird;
  - TNS-Schlüssel - wenn die Option ResolveTNS auf true gesetzt ist und keine der oben genannten Bedingungen zutrifft;
  - TNS-Wert - wenn er mit der öffnenden Klammer "(" beginnt (führende Leerzeichen werden ignoriert). In diesem Fall wird die Option ResolveTNS nicht berücksichtigt.
- **user** - der Oracle-Benutzername; unterstützt das Anhängen einer der Anmeldeoptionen `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oder `as sysrac` im Format `user as sysdba` (bei einer Anmeldeoption wird die Groß-/Kleinschreibung nicht beachtet und sie darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Name des Oracle-Service;
- **destination** - der Name des Ziels, das abgefragt werden soll.

oracle.cdb.info[connString,<user>,<password>,<service>,<database>]

 Die Informationen zu Container-Datenbanken (CDBs).
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - kann Folgendes sein:
  - die URI - wenn keine Sitzung mit dem angegebenen Namen gefunden wird und `Plugins.Oracle.ResolveTNS` auf false gesetzt ist. Wenn sie ein Schema (z. B. "tcp://"), einen Port (z. B. 1521) oder beides enthält, wird die Option ResolveTNS nicht berücksichtigt und sie wird in jedem Fall als URI behandelt;
  - Sitzungsname - wenn ein solcher Name in der Plugin-Konfigurationsdatei gefunden wird;
  - TNS-Schlüssel - wenn die Option ResolveTNS auf true gesetzt ist und keine der oben genannten Bedingungen zutrifft;
  - TNS-Wert - wenn er mit der öffnenden Klammer "(" beginnt (führende Leerzeichen werden ignoriert). In diesem Fall wird die Option ResolveTNS nicht berücksichtigt.
- **user** - der Oracle-Benutzername; unterstützt das Anhängen einer der Anmeldeoptionen `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oder `as sysrac` im Format `user as sysdba` (bei einer Anmeldeoption wird die Groß-/Kleinschreibung nicht beachtet und sie darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Name des Oracle-Service;
- **database** - der Name der Datenbank, die abgefragt werden soll.

oracle.custom.query[connString,<user>,<password>,<service>,queryName,<args...>]

 Das Ergebnis einer benutzerdefinierten Abfrage.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - kann Folgendes sein:
  - die URI - wenn keine Sitzung mit dem angegebenen Namen gefunden wird und `Plugins.Oracle.ResolveTNS` auf false gesetzt ist. Wenn sie ein Schema (z. B. "tcp://"), einen Port (z. B. 1521) oder beides enthält, wird die Option ResolveTNS nicht berücksichtigt und sie wird dennoch als URI behandelt;
  - Sitzungsname - wenn ein solcher Name in der Plugin-Konfigurationsdatei gefunden wird;
  - TNS-Schlüssel - wenn die Option ResolveTNS auf true gesetzt ist und keine der oben genannten Bedingungen zutrifft;
  - TNS-Wert - wenn er mit der öffnenden Klammer "(" beginnt (führende Leerzeichen werden ignoriert). In diesem Fall wird die Option ResolveTNS nicht berücksichtigt.
- **user** - der Oracle-Benutzername; unterstützt das Anhängen einer der Anmeldeoptionen `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oder `as sysrac` im Format `user as sysdba` (bei einer Anmeldeoption wird die Groß-/Kleinschreibung nicht beachtet; sie darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Oracle-Servicename;
- **queryName** - der Name einer benutzerdefinierten Abfrage; muss mit dem SQL-Dateinamen ohne Erweiterung übereinstimmen;
- **args** - ein oder mehrere durch Kommas getrennte Argumente, die an die Abfrage übergeben werden.

Kommentare: - Zurückgegebene Daten werden automatisch in JSON konvertiert. - Vermeiden Sie es, JSON direkt aus Abfragen zurückzugeben, da es beschädigt wird, wenn das Plugin versucht, es erneut in JSON zu konvertieren.

oracle.datafiles.stats[connString,<user>,<password>,<service>]

 Gibt die Statistik der Datendateien zurück.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - kann Folgendes sein:
  - die URI - wenn keine Sitzung mit dem angegebenen Namen gefunden wird und `Plugins.Oracle.ResolveTNS` auf false gesetzt ist. Wenn sie ein Schema (z. B. "tcp://"), einen Port (z. B. 1521) oder beides enthält, wird die Option ResolveTNS nicht berücksichtigt und sie wird dennoch als URI behandelt;
  - Sitzungsname - wenn ein solcher Name in der Plugin-Konfigurationsdatei gefunden wird;
  - TNS-Schlüssel - wenn die Option ResolveTNS auf true gesetzt ist und keine der oben genannten Bedingungen zutrifft;
  - TNS-Wert - wenn er mit der öffnenden Klammer "(" beginnt (führende Leerzeichen werden ignoriert). In diesem Fall wird die Option ResolveTNS nicht berücksichtigt.
- **user** - der Oracle-Benutzername; unterstützt das Anhängen einer der Anmeldeoptionen `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oder `as sysrac` im Format `user as sysdba` (bei einer Anmeldeoption wird die Groß-/Kleinschreibung nicht berücksichtigt; sie darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Oracle-Service-Name.

oracle.db.discovery[connString,<user>,<password>,<service>]

 Gibt die Liste der Datenbanken zurück. Wird für Low-Level-Discovery verwendet.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - kann Folgendes sein:
  - die URI - wenn keine Sitzung mit dem angegebenen Namen gefunden wird und `Plugins.Oracle.ResolveTNS` auf false gesetzt ist. Wenn sie ein Schema (z. B. "tcp://"), einen Port (z. B. 1521) oder beides enthält, wird die Option ResolveTNS nicht berücksichtigt und sie wird dennoch als URI behandelt;
  - Sitzungsname - wenn ein solcher Name in der Plugin-Konfigurationsdatei gefunden wird;
  - TNS-Schlüssel - wenn die Option ResolveTNS auf true gesetzt ist und keine der oben genannten Bedingungen zutrifft;
  - TNS-Wert - wenn er mit der öffnenden Klammer "(" beginnt (führende Leerzeichen werden ignoriert). In diesem Fall wird die Option ResolveTNS nicht berücksichtigt.
- **user** - der Oracle-Benutzername; unterstützt das Anhängen einer der Anmeldeoptionen `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oder `as sysrac` im Format `user as sysdba` (bei einer Anmeldeoption wird die Groß-/Kleinschreibung nicht beachtet und sie darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Oracle-Servicename.

oracle.fra.stats[connString,<user>,<password>,<service>]

 Gibt die Statistiken des Fast Recovery Area (FRA) zurück.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - kann Folgendes sein:
  - die URI - wenn keine Sitzung mit dem angegebenen Namen gefunden wird und `Plugins.Oracle.ResolveTNS` auf false gesetzt ist. Wenn sie ein Schema (z. B. "tcp://"), einen Port (z. B. 1521) oder beides enthält, wird die Option ResolveTNS nicht berücksichtigt und sie wird in jedem Fall als URI behandelt;
  - Sitzungsname - wenn ein solcher Name in der Konfigurationsdatei des Plugins gefunden wird;
  - TNS-Schlüssel - wenn die Option ResolveTNS auf true gesetzt ist und keine der oben genannten Bedingungen zutrifft;
  - TNS-Wert - wenn er mit der öffnenden Klammer "(" beginnt (führende Leerzeichen werden ignoriert). In diesem Fall wird die Option ResolveTNS nicht berücksichtigt.
- **user** - der Oracle-Benutzername; unterstützt das Anhängen einer der Anmeldeoptionen `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oder `as sysrac` im Format `user as sysdba` (bei einer Anmeldeoption wird die Groß-/Kleinschreibung nicht beachtet; sie darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Oracle-Servicename.

oracle.instance.info[connString,<user>,<password>,<service>]

 Die Instanzstatistiken.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - kann Folgendes sein:
  - die URI - wenn keine Sitzung mit dem angegebenen Namen gefunden wird und `Plugins.Oracle.ResolveTNS` auf false gesetzt ist. Wenn sie ein Schema (z. B. "tcp://"), einen Port (z. B. 1521) oder beides enthält, wird die Option ResolveTNS nicht berücksichtigt und sie wird dennoch als URI behandelt;
  - Sitzungsname - wenn ein solcher Name in der Plugin-Konfigurationsdatei gefunden wird;
  - TNS-Schlüssel - wenn die Option ResolveTNS auf true gesetzt ist und keine der oben genannten Bedingungen zutrifft;
  - TNS-Wert - wenn er mit der öffnenden Klammer "(" beginnt (führende Leerzeichen werden ignoriert). In diesem Fall wird die Option ResolveTNS nicht berücksichtigt.
- **user** - der Oracle-Benutzername; unterstützt das Anhängen einer der Anmeldeoptionen `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oder `as sysrac` im Format `user as sysdba` (bei einer Anmeldeoption wird die Groß-/Kleinschreibung nicht beachtet und sie darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Name des Oracle-Service.

oracle.pdb.info[connString,<user>,<password>,<service>,<database>]

 Die Informationen zu Pluggable Databases (PDBs).
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - kann Folgendes sein:
  - die URI - wenn keine Sitzung mit dem angegebenen Namen gefunden wird und `Plugins.Oracle.ResolveTNS` auf false gesetzt ist. Wenn sie ein Schema (z. B. "tcp://"), einen Port (z. B. 1521) oder beides enthält, wird die Option ResolveTNS nicht berücksichtigt und sie wird dennoch als URI behandelt;
  - Sitzungsname - wenn ein solcher Name in der Plugin-Konfigurationsdatei gefunden wird;
  - TNS-Schlüssel - wenn die Option ResolveTNS auf true gesetzt ist und keine der oben genannten Bedingungen zutrifft;
  - TNS-Wert - wenn er mit der öffnenden Klammer "(" beginnt (führende Leerzeichen werden ignoriert). In diesem Fall wird die Option ResolveTNS nicht berücksichtigt.
- **user** - der Oracle-Benutzername; unterstützt das Anhängen einer der Anmeldeoptionen `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oder `as sysrac` im Format `user as sysdba` (bei einer Anmeldeoption wird die Groß-/Kleinschreibung nicht beachtet; sie darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Oracle-Service-Name;
- **database** - der Name der Datenbank, die abgefragt werden soll.

oracle.pdb.discovery[connString,<user>,<password>,<service>]

 Gibt die Liste der PDBs zurück. Wird für Low-Level-Discovery verwendet.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - kann Folgendes sein:
  - die URI - wenn keine Sitzung mit dem angegebenen Namen gefunden wird und `Plugins.Oracle.ResolveTNS` auf false gesetzt ist. Wenn sie ein Schema (z. B. "tcp://"), einen Port (z. B. 1521) oder beides enthält, wird die Option

ResolveTNS nicht berücksichtigt und sie wird dennoch als URI behandelt;
    - Sitzungsname - wenn ein solcher Name in der Plugin-Konfigurationsdatei gefunden wird;
    - TNS-Schlüssel - wenn die Option ResolveTNS auf true gesetzt ist und keine der oben genannten Bedingungen zutrifft;
    - TNS-Wert - wenn er mit der öffnenden Klammer "(" beginnt (führende Leerzeichen werden ignoriert). In diesem Fall wird die Option ResolveTNS nicht berücksichtigt.
- **user** - der Oracle-Benutzername; unterstützt das Anhängen einer der Anmeldeoptionen `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oder `as sysrac` im Format `user as sysdba` (bei einer Anmeldeoption wird die Groß-/Kleinschreibung nicht beachtet; sie darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Oracle-Service-Name.

oracle.pga.stats[connString,<user>,<password>,<service>]

 Gibt die Statistiken des Program Global Area (PGA) zurück.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - kann Folgendes sein:
    - die URI - wenn keine Sitzung mit dem angegebenen Namen gefunden wird und `Plugins.Oracle.ResolveTNS` auf false gesetzt ist. Wenn sie ein Schema (z. B. "tcp://"), einen Port (z. B. 1521) oder beides enthält, wird die Option ResolveTNS nicht berücksichtigt und sie wird dennoch als URI behandelt;
    - Sitzungsname - wenn ein solcher Name in der Plugin-Konfigurationsdatei gefunden wird;
    - TNS-Schlüssel - wenn die Option ResolveTNS auf true gesetzt ist und keine der oben genannten Bedingungen zutrifft;
    - TNS-Wert - wenn er mit der öffnenden Klammer "(" beginnt (führende Leerzeichen werden ignoriert). In diesem Fall wird die Option ResolveTNS nicht berücksichtigt.
- **user** - der Oracle-Benutzername; unterstützt das Anhängen einer der Anmeldeoptionen `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oder `as sysrac` im Format `user as sysdba` (bei einer Anmeldeoption wird die Groß-/Kleinschreibung nicht beachtet und sie darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Name des Oracle-Dienstes.

oracle.ping[connString,<user>,<password>,<service>]

 Prüft, ob eine Verbindung zu Oracle hergestellt werden kann.
 Rückgabewert: *1* - die Verbindung ist erfolgreich; *0* - die Verbindung ist unterbrochen (wenn ein beliebiger Fehler auftritt, einschließlich AUTH- und Konfigurationsproblemen).

Parameter:

- **connString** - kann Folgendes sein:
    - die URI - wenn keine Sitzung mit dem angegebenen Namen gefunden wird und `Plugins.Oracle.ResolveTNS` auf false gesetzt ist. Wenn sie ein Schema (z. B. "tcp://"), einen Port (z. B. 1521) oder beides enthält, wird die Option ResolveTNS nicht berücksichtigt und sie wird in jedem Fall als URI behandelt;
    - Sitzungsname - wenn ein solcher Name in der Plugin-Konfigurationsdatei gefunden wird;
    - TNS-Schlüssel - wenn die Option ResolveTNS auf true gesetzt ist und keine der oben genannten Bedingungen zutrifft;
    - TNS-Wert - wenn er mit der öffnenden Klammer "(" beginnt (führende Leerzeichen werden ignoriert). In diesem Fall wird die Option ResolveTNS nicht berücksichtigt.
- **user** - der Oracle-Benutzername; unterstützt das Anhängen einer der Anmeldeoptionen `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oder `as sysrac` im Format `user as sysdba` (bei einer Anmeldeoption wird die Groß-/Kleinschreibung nicht beachtet; sie darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Oracle-Service-Name.

Beispiele:

```
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>]

 Gibt die Prozessstatistiken zurück.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - kann Folgendes sein:

  - die URI - wenn keine Sitzung mit dem angegebenen Namen gefunden wird und `Plugins.Oracle.ResolveTNS` auf false gesetzt ist. Wenn sie ein Schema (z. B. "tcp://"), einen Port (z. B. 1521) oder beides enthält, wird die Option ResolveTNS nicht berücksichtigt und sie wird dennoch als URI behandelt;
  - Sitzungsname - wenn ein solcher Name in der Plugin-Konfigurationsdatei gefunden wird;
  - TNS-Schlüssel - wenn die Option ResolveTNS auf true gesetzt ist und keine der oben genannten Bedingungen zutrifft;
  - TNS-Wert - wenn er mit der öffnenden Klammer "(" beginnt (führende Leerzeichen werden ignoriert). In diesem Fall wird die Option ResolveTNS nicht berücksichtigt.
- **user** - der Oracle-Benutzername; unterstützt das Anhängen einer der Anmeldeoptionen `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oder `as sysrac` im Format `user as sysdba` (bei einer Anmeldeoption wird die Groß-/Kleinschreibung nicht beachtet und sie darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Name des Oracle-Service.

oracle.redolog.info[connString,<user>,<password>,<service>]

 Die Logdateiinformationen aus der Kontrolldatei.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - kann Folgendes sein:
  - die URI - wenn keine Sitzung mit dem angegebenen Namen gefunden wird und `Plugins.Oracle.ResolveTNS` auf false gesetzt ist. Wenn sie ein Schema (z. B. "tcp://"), einen Port (z. B. 1521) oder beides enthält, wird die Option ResolveTNS nicht berücksichtigt und sie wird dennoch als URI behandelt;
  - Sitzungsname - wenn ein solcher Name in der Plugin-Konfigurationsdatei gefunden wird;
  - TNS-Schlüssel - wenn die Option ResolveTNS auf true gesetzt ist und keine der oben genannten Bedingungen zutrifft;
  - TNS-Wert - wenn er mit der öffnenden Klammer "(" beginnt (führende Leerzeichen werden ignoriert). In diesem Fall wird die Option ResolveTNS nicht berücksichtigt.
- **user** - der Oracle-Benutzername; unterstützt das Anhängen einer der Anmeldeoptionen `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oder `as sysrac` im Format `user as sysdba` (bei einer Anmeldeoption wird die Groß-/Kleinschreibung nicht beachtet; sie darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Name des Oracle-Dienstes.

oracle.sga.stats[connString,<user>,<password>,<service>]

 Gibt die Statistiken der System Global Area (SGA) zurück.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - kann Folgendes sein:
  - die URI - wenn keine Sitzung mit dem angegebenen Namen gefunden wird und `Plugins.Oracle.ResolveTNS` auf false gesetzt ist. Wenn sie ein Schema (z. B. "tcp://"), einen Port (z. B. 1521) oder beides enthält, wird die Option ResolveTNS nicht berücksichtigt und sie wird dennoch als URI behandelt;
  - Sitzungsname - wenn ein solcher Name in der Konfigurationsdatei des Plugins gefunden wird;
  - TNS-Schlüssel - wenn die Option ResolveTNS auf true gesetzt ist und keine der oben genannten Bedingungen zutrifft;
  - TNS-Wert - wenn er mit der öffnenden Klammer "(" beginnt (führende Leerzeichen werden ignoriert). In diesem Fall wird die Option ResolveTNS nicht berücksichtigt.
- **user** - der Oracle-Benutzername; unterstützt das Anhängen einer der Anmeldeoptionen `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oder `as sysrac` im Format `user as sysdba` (bei einer Anmeldeoption wird die Groß-/Kleinschreibung nicht beachtet; sie darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Oracle-Service-Name.

oracle.sessions.stats[connString,<user>,<password>,<service>,<lockMaxTime>]

 Gibt die Sitzungsstatistiken zurück.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - kann Folgendes sein:
  - die URI - wenn keine Sitzung mit dem angegebenen Namen gefunden wird und `Plugins.Oracle.ResolveTNS` auf false gesetzt ist. Wenn sie ein Schema (z. B. "tcp://"), einen Port (z. B. 1521) oder beides enthält, wird die Option ResolveTNS nicht berücksichtigt und sie wird dennoch als URI behandelt;
  - Sitzungsname - wenn ein solcher Name in der Plugin-Konfigurationsdatei gefunden wird;
  - TNS-Schlüssel - wenn die Option ResolveTNS auf true gesetzt ist und keine der oben genannten Bedingungen zutrifft;

    - TNS-Wert - wenn er mit der öffnenden Klammer "(" beginnt (führende Leerzeichen werden ignoriert). In diesem Fall wird die Option ResolveTNS nicht berücksichtigt.
- **user** - der Oracle-Benutzername; unterstützt das Anhängen einer der Anmeldeoptionen `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oder `as sysrac` im Format `user as sysdba` (eine Anmeldeoption unterscheidet nicht zwischen Groß- und Kleinschreibung und darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Oracle-Servicename;
- **lockMaxTime** - die maximale Dauer einer Sitzungssperre in Sekunden, damit die Sitzung als längerfristig gesperrt gezählt wird. Standard: 600 Sekunden.

oracle.sys.metrics[connString,<user>,<password>,<service>,<duration>]

 Gibt eine Reihe von Systemmetrikwerten zurück.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - kann Folgendes sein:
    - die URI - wenn keine Sitzung mit dem angegebenen Namen gefunden wird und `Plugins.Oracle.ResolveTNS` auf false gesetzt ist. Wenn sie ein Schema (z. B. "tcp://"), einen Port (z. B. 1521) oder beides enthält, wird die ResolveTNS-Option nicht berücksichtigt und sie wird in jedem Fall als URI behandelt;
    - der Sitzungsname - wenn ein solcher Name in der Plugin-Konfigurationsdatei gefunden wird;
    - der TNS-Schlüssel - wenn die ResolveTNS-Option auf true gesetzt ist und keine der oben genannten Bedingungen zutrifft;
    - der TNS-Wert - wenn er mit der öffnenden Klammer "(" beginnt (führende Leerzeichen werden ignoriert). In diesem Fall wird die ResolveTNS-Option nicht berücksichtigt.
- **user** - der Oracle-Benutzername; unterstützt das Anhängen einer der Anmeldeoptionen `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oder `as sysrac` im Format `user as sysdba` (eine Anmeldeoption unterscheidet nicht zwischen Groß- und Kleinschreibung und darf kein nachfolgendes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Oracle-Service-Name;
- **duration** - das Erfassungsintervall (in Sekunden) der Systemmetrikw erte. Mögliche Werte: *60* - lange Dauer (Standard), *15* - kurze Dauer.

oracle.sys.params[connString,<user>,<password>,<service>]

 Gibt eine Menge von Systemparameterwerten zurück.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - kann Folgendes sein:
    - die URI - wenn keine Sitzung mit dem angegebenen Namen gefunden wird und `Plugins.Oracle.ResolveTNS` auf false gesetzt ist. Wenn sie ein Schema (z. B. "tcp://"), einen Port (z. B. 1521) oder beides enthält, wird die Option ResolveTNS nicht berücksichtigt und sie wird dennoch als URI behandelt;
    - Sitzungsname - wenn ein solcher Name in der Plugin-Konfigurationsdatei gefunden wird;
    - TNS-Schlüssel - wenn die Option ResolveTNS auf true gesetzt ist und keine der oben genannten Bedingungen zutrifft;
    - TNS-Wert - wenn er mit der öffnenden Klammer "(" beginnt (führende Leerzeichen werden ignoriert). In diesem Fall wird die Option ResolveTNS nicht berücksichtigt.
- **user** - der Oracle-Benutzername; unterstützt das Anhängen einer der Anmeldeoptionen `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oder `as sysrac` im Format `user as sysdba` (bei einer Anmeldeoption wird die Groß-/Kleinschreibung nicht beachtet; sie darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Oracle-Service-Name.

oracle.ts.stats[connString,<user>,<password>,<service>,<tablespace>,<type>,<conname>]

 Gibt die Tablespace-Statistiken zurück.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - kann Folgendes sein:
    - die URI - wenn keine Sitzung mit dem angegebenen Namen gefunden wird und `Plugins.Oracle.ResolveTNS` auf false gesetzt ist. Wenn sie ein Schema (z. B. "tcp://"), einen Port (z. B. 1521) oder beides enthält, wird die Option ResolveTNS nicht berücksichtigt und sie wird in jedem Fall als URI behandelt;
    - Sitzungsname - wenn ein solcher Name in der Plugin-Konfigurationsdatei gefunden wird;
    - TNS-Schlüssel - wenn die Option ResolveTNS auf true gesetzt ist und keine der oben genannten Bedingungen zutrifft;
    - TNS-Wert - wenn er mit der öffnenden Klammer "(" beginnt (führende Leerzeichen werden ignoriert). In diesem Fall wird die Option ResolveTNS nicht berücksichtigt.

- **user** - der Oracle-Benutzername; unterstützt das Anhängen einer der Anmeldeoptionen `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oder `as sysrac` im Format `user as sysdba` (eine Anmeldeoption unterscheidet nicht zwischen Groß- und Kleinschreibung und darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Oracle-Service-Name;
- **tablespace** - Name des abzufragenden Tablespace. Standardwert (wenn leer gelassen und `type` gesetzt ist):
  - "TEMP" (wenn `type` auf "TEMPORARY" gesetzt ist);
  - "USERS" (wenn `type` auf "PERMANENT" gesetzt ist).
- **type** - der Typ des abzufragenden Tablespace. Standardwert (wenn `tablespace` gesetzt ist): "PERMANENT".
- **conname** - Name des Containers, für den die Informationen benötigt werden.

Wenn `tablespace`, `type` oder `conname` weggelassen wird, gibt der Datenpunkt Tablespace-Statistiken für alle übereinstimmenden Container zurück (einschließlich PDBs und CDB).

oracle.ts.discovery[connString,<user>,<password>,<service>]

 Gibt eine Liste von Tablespaces zurück. Wird für Low-Level-Discovery verwendet.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - kann Folgendes sein:
  - die URI - wenn keine Sitzung mit dem angegebenen Namen gefunden wird und `Plugins.Oracle.ResolveTNS` auf false gesetzt ist. Wenn sie ein Schema (z. B. "tcp://"), einen Port (z. B. 1521) oder beides enthält, wird die Option ResolveTNS nicht berücksichtigt und sie wird dennoch als URI behandelt;
  - Sitzungsname - wenn ein solcher Name in der Plugin-Konfigurationsdatei gefunden wird;
  - TNS-Schlüssel - wenn die Option ResolveTNS auf true gesetzt ist und keine der oben genannten Bedingungen zutrifft;
  - TNS-Wert - wenn er mit der öffnenden Klammer "(" beginnt (führende Leerzeichen werden ignoriert). In diesem Fall wird die Option ResolveTNS nicht berücksichtigt.
- **user** - der Oracle-Benutzername; unterstützt das Anhängen einer der Anmeldeoptionen `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oder `as sysrac` im Format `user as sysdba` (bei einer Anmeldeoption wird die Groß-/Kleinschreibung nicht beachtet und sie darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Oracle-Service-Name.

oracle.user.info[connString,<user>,<password>,<service>,<username>]

 Gibt Informationen über Oracle-Benutzer zurück.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **connString** - kann Folgendes sein:
  - die URI - wenn keine Sitzung mit dem angegebenen Namen gefunden wird und `Plugins.Oracle.ResolveTNS` auf false gesetzt ist. Wenn sie ein Schema (z. B. "tcp://"), einen Port (z. B. 1521) oder beides enthält, wird die Option ResolveTNS nicht berücksichtigt und sie wird dennoch als URI behandelt;
  - Sitzungsname - wenn ein solcher Name in der Plugin-Konfigurationsdatei gefunden wird;
  - TNS-Schlüssel - wenn die Option ResolveTNS auf true gesetzt ist und keine der oben genannten Bedingungen zutrifft;
  - TNS-Wert - wenn er mit der öffnenden Klammer "(" beginnt (führende Leerzeichen werden ignoriert). In diesem Fall wird die Option ResolveTNS nicht berücksichtigt.
- **user** - der Oracle-Benutzername; unterstützt das Anhängen einer der Anmeldeoptionen `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oder `as sysrac` im Format `user as sysdba` (bei einer Anmeldeoption wird die Groß-/Kleinschreibung nicht beachtet; sie darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Oracle-Service-Name;
- **username** - der Benutzername, für den die Informationen benötigt werden. Benutzernamen in Kleinbuchstaben werden nicht unterstützt. Standard: aktueller Benutzer.

oracle.version[connString,<user>,<password>,<service>]

 Gibt die Version des Datenbank-Servers zurück.
 Rückgabewert: *String*.

Parameter:

- **connString** - kann Folgendes sein:
  - die URI - wenn keine Sitzung mit dem angegebenen Namen gefunden wird und `Plugins.Oracle.ResolveTNS` auf false gesetzt ist. Wenn sie ein Schema (z. B. "tcp://"), einen Port (z. B. 1521) oder beides enthält, wird die Option ResolveTNS nicht berücksichtigt und sie wird in jedem Fall als URI behandelt;
  - Sitzungsname - wenn ein solcher Name in der Plugin-Konfigurationsdatei gefunden wird;

  - TNS-Schlüssel - wenn die Option ResolveTNS auf true gesetzt ist und keine der oben genannten Bedingungen zutrifft;
  - TNS-Wert - wenn er mit der öffnenden Klammer "(" beginnt (führende Leerzeichen werden ignoriert). In diesem Fall wird die Option ResolveTNS nicht berücksichtigt.
- **user** - der Oracle-Benutzername; unterstützt das Anhängen einer der Anmeldeoptionen `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` oder `as sysrac` im Format `user as sysdba` (bei einer Anmeldeoption wird die Groß-/Kleinschreibung nicht beachtet; sie darf kein nachgestelltes Leerzeichen enthalten).
- **password** - das Oracle-Passwort;
- **service** - der Oracle-Service-Name.

pgsql.autovacuum.count[uri,<username>,<password>,<database name>]

 Die Anzahl der Autovacuum-Worker.
 Rückgabewert: *Integer*.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Anmeldedaten;
- **database name** - der Datenbankname (falls weggelassen, wird der Wert von `Plugins.PostgreSQL.Default.Database` aus `postgresql.conf` verwendet).

pgsql.archive[uri,<username>,<password>,<database name>]

 Informationen zu archivierten Dateien.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Anmeldedaten;
- **database name** - der Datenbankname (falls weggelassen, wird der Wert von `Plugins.PostgreSQL.Default.Database` aus `postgresql.conf` verwendet).

pgsql.bgwriter[uri,<username>,<password>,<database name>]

 Die Gesamtzahl der Checkpoints für den Datenbankcluster, aufgeschlüsselt nach Checkpoint-Typ.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Zugangsdaten;
- **database name** - der Name der Datenbank (falls weggelassen, wird der Wert von `Plugins.PostgreSQL.Default.Database` aus `postgresql.conf` verwendet).

pgsql.cache.hit[uri,<username>,<password>,<database name>]

 Die Trefferquote des PostgreSQL-Puffer-Cache.
 Rückgabewert: *Float*.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Zugangsdaten;
- **database name** - der Datenbankname (falls weggelassen, wird der Wert von `Plugins.PostgreSQL.Default.Database` aus `postgresql.conf` verwendet).

pgsql.connections[uri,<username>,<password>,<database name>]

 Gibt Verbindungen nach Typ zurück.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Zugangsdaten;
- **database name** - der Datenbankname (falls weggelassen, wird der Wert von `Plugins.PostgreSQL.Default.Database` aus `postgresql.conf` verwendet).

pgsql.custom.query[uri,<username>,<password>,<database name>,queryName,<args...>]

 Gibt das Ergebnis einer benutzerdefinierten Abfrage zurück.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Zugangsdaten;

- **database name** - der Datenbankname (falls weggelassen, wird der Wert von Plugins.PostgreSQL.Default.Database aus postgresql.conf verwendet);
- **queryName** - der Name einer benutzerdefinierten Abfrage, muss mit dem SQL-Dateinamen ohne Erweiterung übereinstimmen;
- **args** - ein oder mehrere durch Kommas getrennte Argumente, die an eine Abfrage übergeben werden.

pgsql.db.age[uri,<username>,<password>,<database name>]

 Das Alter des ältesten FrozenXID der Datenbank (Anzahl der Transaktionen).
 Rückgabewert: *Integer*.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Anmeldedaten;
- **database name** - der Datenbankname (falls weggelassen, wird der Wert von `Plugins.PostgreSQL.Default.Database` aus `postgresql.conf` verwendet).

pgsql.db.bloating_tables[uri,<username>,<password>,<database name>]

 Die Anzahl der aufgeblähten Tabellen pro Datenbank.
 Rückgabewert: *Integer*.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Anmeldedaten;
- **database name** - der Datenbankname (falls weggelassen, wird der Wert von `Plugins.PostgreSQL.Default.Database` aus `postgresql.conf` verwendet).

pgsql.db.discovery[uri,<username>,<password>,<database name>]

 Die Liste der PostgreSQL-Datenbanken. Wird für Low-Level-Discovery verwendet.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Zugangsdaten;
- **database name** - der Datenbankname (falls weggelassen, wird der Wert von `Plugins.PostgreSQL.Default.Database` aus `postgresql.conf` verwendet).

pgsql.db.size[uri,<username>,<password>,<database name>]

 Die Größe der Datenbank in Bytes.
 Rückgabewert: *Integer*.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Zugangsdaten;
- **database name** - der Datenbankname (falls weggelassen, wird der Wert von `Plugins.PostgreSQL.Default.Database` aus `postgresql.conf` verwendet).

pgsql.dbstat[uri,<username>,<password>,<database name>]

 Sammelt die Statistiken pro Datenbank. Wird für Low-Level-Discovery verwendet.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Zugangsdaten;
- **database name** - der Datenbankname (falls weggelassen, wird der Wert von `Plugins.PostgreSQL.Default.Database` aus `postgresql.conf` verwendet).

pgsql.dbstat.sum[uri,<username>,<password>,<database name>]

 Die zusammengefassten Daten für alle Datenbanken in einem Cluster.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Anmeldedaten;
- **database name** - der Datenbankname (falls weggelassen, wird der Wert von `Plugins.PostgreSQL.Default.Database` aus `postgresql.conf` verwendet).

pgsql.locks[uri,<username>,<password>,<database name>]

 Informationen über gewährte Sperren pro Datenbank. Verwendet für Low-Level-Discovery.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Anmeldedaten;
- **database name** - der Datenbankname (falls weggelassen, wird der Wert von `Plugins.PostgreSQL.Default.Database` aus `postgresql.conf` verwendet).

pgsql.oldest.xid[uri,<username>,<password>,<database name>]

 Das Alter der ältesten XID (Anzahl der Transaktionen).
 Rückgabewert: *Integer*.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Anmeldedaten;
- **database name** - der Datenbankname (falls weggelassen, wird der Wert von `Plugins.PostgreSQL.Default.Database` aus `postgresql.conf` verwendet).

pgsql.ping[uri,<username>,<password>,<database name>]

 Prüft, ob eine Verbindung aktiv ist oder nicht.
 Rückgabewert: *1* - die Verbindung ist aktiv; *0* - die Verbindung ist unterbrochen (wenn ein Fehler auftritt, einschließlich AUTH- und Konfigurationsproblemen).

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Zugangsdaten;
- **database name** - der Datenbankname (falls weggelassen, wird der Wert von `Plugins.PostgreSQL.Default.Database` aus `postgresql.conf` verwendet).

pgsql.queries[uri,<username>,<password>,<database name>,time period]

 Ermittelt Metriken für Abfragen nach Ausführungszeit.
 Rückgabewert: *JSON object*.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Anmeldedaten;
- **database name** - der Datenbankname (falls weggelassen, wird der Wert von `Plugins.PostgreSQL.Default.Database` aus `postgresql.conf` verwendet);
- **time period** - das Zeitlimit für die Ausführungszeit zur Zählung langsamer Abfragen (muss eine positive ganze Zahl sein).

pgsql.replication.count[uri,<username>,<password>]

 Die Anzahl der Standby-Server.
 Rückgabewert: *Integer*.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Zugangsdaten.

pgsql.replication.process[uri,<username>,<password>]

 Die Flush-Verzögerung, Schreibverzögerung und Replay-Verzögerung für jeden Senderprozess.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **uri** - der URI- oder Sitzungsname;
- **username, password** - die PostgreSQL-Anmeldedaten.

pgsql.replication.process.discovery[uri,<username>,<password>]

 Die Erkennung des Namens des Replikationsprozesses.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Zugangsdaten.

pgsql.replication.recovery_role[uri,<username>,<password>]

 Der Wiederherstellungsstatus.
 Rückgabewert: *0* - Master-Modus; *1* - die Wiederherstellung ist noch im Gange (Standby-Modus).

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Zugangsdaten.

pgsql.replication.status[uri,<username>,<password>]

 Der Status der Replikation.
 Rückgabewert: *0* - Streaming ist ausgefallen; *1* - Streaming ist aktiv; *2* - Master-Modus.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Zugangsdaten.

pgsql.replication_lag.b[uri,<username>,<password>]

 Die Replikationsverzögerung in Byte.
 Rückgabewert: *Integer*.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Anmeldedaten.

pgsql.replication_lag.sec[uri,<username>,<password>]

 Die Replikationsverzögerung in Sekunden.
 Rückgabewert: *Integer*.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Zugangsdaten.

pgsql.uptime[uri,<username>,<password>,<database name>]

 Die PostgreSQL-Uptime in Millisekunden.
 Rückgabewert: *Float*.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Anmeldedaten;
- **database name** - der Datenbankname (falls weggelassen, wird der Wert von `Plugins.PostgreSQL.Default.Database` aus `postgresql.conf` verwendet).

pgsql.version[uri,<username>,<password>,<database name>]

 Gibt die PostgreSQL-Version zurück.
 Rückgabewert: *String*.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Zugangsdaten;
- **database name** - der Datenbankname (falls weggelassen, wird der Wert von `Plugins.PostgreSQL.Default.Database` aus `postgresql.conf` verwendet).

pgsql.wal.stat[uri,<username>,<password>,<database name>]

 Die WAL-Statistiken.
 Rückgabewert: *JSON object*.

Parameter:

- **uri** - die URI oder der Sitzungsname;
- **username, password** - die PostgreSQL-Anmeldedaten;
- **database name** - der Datenbankname (falls weggelassen, wird der Wert von `Plugins.PostgreSQL.Default.Database` aus `postgresql.conf` verwendet).

redis.config[connString,<password>,<pattern>]

 Ruft die Konfigurationsparameter einer Redis-Instanz ab, die dem Muster entsprechen.
 Rückgabewert: *JSON* - wenn ein Muster im Glob-Stil verwendet wurde; Einzelwert - wenn ein Muster kein Platzhalterzeichen enthielt.

Parameter:

- **connString** - der URI- oder Sitzungsname;
- **password** - das Redis-Passwort;
- **pattern** - ein Muster im Glob-Stil (* standardmäßig).

Kommentar:

- Sicherheitswarnung: `redis.config` führt den Redis-Befehl `CONFIG GET` aus und kann sensible Konfigurationsparameter zurückgeben (zum Beispiel `requirepass`, `masterauth`, TLS-bezogene Einstellungen und andere Geheimnisse). Bei Verwendung des Standardmusters (`*`) oder weit gefasster Muster können diese Geheimnisse für jeden offengelegt werden, der berechtigt ist, Datenpunkt-Werte in Zabbix anzuzeigen.

redis.info[connString,<password>,<section>]

 Ruft die Ausgabe des INFO-Befehls ab.
 Rückgabewert: *JSON* - die Ausgabe wird als JSON serialisiert.

Parameter:

- **connString** - die URI oder der Sitzungsname;
- **password** - das Redis-Passwort;
- **section** - der Abschnitt der Informationen (*default* standardmäßig).

redis.ping[connString,<password>]

 Prüft, ob eine Verbindung aktiv ist oder nicht.
 Rückgabewert: *1* - die Verbindung ist aktiv; *0* - die Verbindung ist unterbrochen (wenn ein Fehler auftritt, einschließlich AUTH- und Konfigurationsproblemen).

Parameter:

- **connString** - die URI oder der Sitzungsname;
- **password** - das Redis-Passwort.

redis.slowlog.count[connString,<password>]

 Die Anzahl der Slow-Log-Einträge seit dem Start von Redis.
 Rückgabewert: *Integer*.

Parameter:

- **connString** - die URI oder der Sitzungsname;
- **password** - das Redis-Passwort.

smart.attribute.discovery

 Gibt eine Liste der S.M.A.R.T.-Geräteattribute zurück.
 Rückgabewert: *JSON object*.

Kommentare:

- Die folgenden Makros und ihre Werte werden zurückgegeben: `{#NAME}`, `{#DISKTYPE}`, `{#ID}`, `{#ATTRNAME}`, `{#THRESH}`;
- HDD-, SSD- und NVME-Laufwerkstypen werden unterstützt. Laufwerke können einzeln oder in einem RAID kombiniert sein. `{#NAME}` erhält im Fall eines RAID einen Zusatz, z. B.: `{"{#NAME}": "/dev/sda cciss,2"}`.

smart.disk.discovery[<type>]

 Gibt eine Liste von S.M.A.R.T.-Geräten zurück.
 Rückgabewert: *JSON object*.

Parameter:

- **type** - gibt einen Wert an, nach dem die Datenträger durchsucht werden sollen. Mögliche Werte: *id*, *name* (Standard). *id* wird unter Windows nicht unterstützt und führt bei Auswahl zu einem Fehler

Kommentare:

- Die folgenden Makros und ihre Werte werden zurückgegeben: `{#NAME}`, `{#DISKTYPE}`, `{#MODEL}`, `{#SN}`, `{#PATH}`, `{#ATTRIBUTES}`, `{#RAIDTYPE}`;
- HDD-, SSD- und NVME-Laufwerkstypen werden unterstützt. Wenn ein Laufwerk nicht zu einem RAID gehört, ist `{#RAIDTYPE}` leer. `{#NAME}` erhält im Fall eines RAID einen Zusatz, z. B.: `{"{#NAME}": "/dev/sda cciss,2"}`.

smart.disk.get[<path>,<raid type>]

 Gibt alle verfügbaren Eigenschaften von S.M.A.R.T.-Geräten zurück.
 Rückgabewert: *JSON object*.

Parameter:

- **path** - der Pfad der Festplatte, das Makro `{#PATH}` kann als Wert verwendet werden;
- **raid_type** - der RAID-Typ, das Makro `{#RAID}` kann als Wert verwendet werden

Kommentare:

- HDD-, SSD- und NVME-Laufwerkstypen werden unterstützt. Laufwerke können einzeln oder in einem RAID kombiniert sein;
- Die Daten enthalten die smartctl-Version und Aufrufargumente sowie zusätzliche Felder:
disk_name - enthält den Namen mit den erforderlichen Ergänzungen für die RAID-Erkennung, z. B.: `{"disk_name": "/dev/sda cciss,2"}`
disk_type - enthält den Laufwerkstyp HDD, SSD oder NVME, z. B.: `{"disk_type": "ssd"}`;
- Wenn keine Parameter angegeben werden, gibt das Datenpunkt Informationen zu allen Laufwerken zurück.

systemd.unit.get[unit name,<interface>]

 Gibt alle Eigenschaften einer systemd-Einheit zurück.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **unit name** - der Name der Einheit (Sie können das Makro `{#UNIT.NAME}` im Datenpunkt-Prototyp verwenden, um den Namen zu ermitteln);
- **interface** - der Typ der Einheiten-Schnittstelle, mögliche Werte: *Unit* (Standard), *Service*, *Socket*, *Device*, *Mount*, *Automount*, *Swap*, *Target*, *Path*.

Kommentare:

- Dieser Datenpunkt wird nur auf der Linux-Plattform unterstützt;
- LoadState, ActiveState und UnitFileState für die Unit-Schnittstelle werden als Text und Ganzzahl zurückgegeben: `"ActiveState":{"state":1,"text":"active"}`.

systemd.unit.info[unit name,<property>,<interface>]

 Informationen zu einer systemd-Unit.
 Rückgabewert: *String*.

Parameter:

- **unit name** – der Name der Unit (möglicherweise möchten Sie das Makro `{#UNIT.NAME}` im Datenpunkt-Prototyp verwenden, um den Namen zu ermitteln);
- **property** – die Eigenschaft der Unit (z. B. ActiveState (Standard), LoadState, Description);
- **interface** – der Schnittstellentyp der Unit (z. B. Unit (Standard), Socket, Service).

Kommentare:

- Dieser Datenpunkt wird nur auf der Linux-Plattform unterstützt;
- Mit diesem Datenpunkt können Sie eine bestimmte Eigenschaft eines bestimmten Schnittstellentyps abrufen, wie in der dbus-API beschrieben.

Beispiele:

```
systemd.unit.info["{#UNIT.NAME}"] #Informationen zum aktiven Status (active, reloading, inactive, failed,
systemd.unit.info["{#UNIT.NAME}",LoadState] #Informationen zum Ladezustand der ermittelten systemd-Units e
systemd.unit.info[mysqld.service,Id] #Den technischen Namen des Dienstes abrufen (mysqld.service)
systemd.unit.info[mysqld.service,Description] #Die Beschreibung des Dienstes abrufen (MySQL Server)
systemd.unit.info[mysqld.service,ActiveEnterTimestamp] #Den Zeitpunkt des letzten Wechsels des Dienstes in
systemd.unit.info[dbus.socket,NConnections,Socket] #Die Anzahl der Verbindungen dieser Socket-Unit erfasse
```

systemd.unit.discovery[<type>]

 Liste der systemd-Units und ihrer Details. Wird für die Low-Level-Discovery verwendet.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **type** – mögliche Werte: *all*, *automount*, *device*, *mount*, *path*, *service* (Standard), *socket*, *swap*, *target*.

Dieser Datenpunkt wird nur auf der Linux-Plattform unterstützt.

web.certificate.get[hostname,<port>,<address>]

 Validiert die Zertifikate und gibt Zertifikatsdetails zurück.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **hostname** - kann entweder eine IP oder DNS sein.
Kann das URL-Schema (*nur https*), den Pfad (wird ignoriert) und den Port enthalten.
Wenn ein Port sowohl im ersten als auch im zweiten Parameter angegeben ist, müssen die Werte übereinstimmen.
Wenn address (der 3. Parameter) angegeben ist, wird der hostname nur für SNI und die Hostnamenprüfung verwendet;
- **port** - die Portnummer (Standard ist 443 für HTTPS);
- **address** - kann entweder eine IP oder DNS sein. Wenn angegeben, wird sie für die Verbindung verwendet, und hostname (der 1. Parameter) wird für SNI und die Hostprüfung verwendet. Falls der 1. Parameter eine IP und der 3. Parameter DNS ist, wird der 1. Parameter für die Verbindung verwendet und der 3. Parameter für SNI und die Hostprüfung.

Kommentare:

- Dieser Datenpunkt wird nicht unterstützt, wenn das im Host-Konfigurationsobjekt angegebene Ziel nicht existiert, nicht verfügbar ist oder wenn der TLS-Handshake mit einem beliebigen Fehler fehlschlägt, außer einem ungültigen Zertifikat;
- Derzeit werden die X.509-Erweiterung AIA (Authority Information Access), CRLs und OCSP (einschließlich OCSP Stapling) sowie Certificate Transparency nicht unterstützt;
- JSON-Antwortfelder:

  - *x509*: enthält die Details des X.509-Zertifikats.
    * *version*: die X.509-Version (z. B. 3).
    * *serial_number*: die Seriennummer des Zertifikats.
    * *signature_algorithm*: der zum Signieren des Zertifikats verwendete Algorithmus (z. B. SHA256-RSA).
    * *issuer*: der Aussteller des Zertifikats.
    * *not_before*: das Startdatum der Gültigkeit des Zertifikats.
    * *not_after*: das Ablaufdatum des Zertifikats.
    * *subject*: das Subjekt des Zertifikats.
    * *public_key_algorithm*: der für den öffentlichen Schlüssel verwendete Algorithmus (z. B. RSA).
    * *alternative_names*: alternative Subjektnamen (falls vorhanden), andernfalls `null`.
  - result: enthält das Validierungsergebnis.
    * *value*: der Validierungsstatus (siehe mögliche Werte unten).
    * *message*: detaillierte Validierungsnachricht (z. B. "`certificate verified successfully`").
  - *sha1_fingerprint*: der SHA-1-Fingerabdruck des Zertifikats.
  - *sha256_fingerprint*: der SHA-256-Fingerabdruck des Zertifikats.
- Das Feld `$.result.value` gibt das Ergebnis der Zertifikatsvalidierung an. Mögliche Werte sind:
  - *valid* - das Zertifikat ist gültig und vertrauenswürdig.
  - *valid-but-self-signed* - das Zertifikat ist gültig, aber selbstsigniert, d. h. sein Subjekt entspricht seinem Aussteller.
  - *invalid* - das Zertifikat ist aufgrund eines Problems wie Ablauf, falschem Hostnamen oder einer unbekannten Signaturstelle ungültig.

Beispiel:

```
web.certificate.get[example.com,443]
```

JSON-Antwort:


```
{
  "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"
}
```


Windows Zabbix Agent

Übersicht

Die Windows-Zabbix-Agent-Datenpunkte werden in zwei Listen dargestellt:

- Gemeinsame Datenpunkte - die Datenpunktschlüssel, die mit dem UNIX-Zabbix-Agent gemeinsam genutzt werden;
- Windows-spezifische Datenpunkte - die Datenpunktschlüssel, die **nur** unter Windows unterstützt werden.

Beachten Sie, dass alle vom Zabbix-Agent unter Windows unterstützten Datenpunktschlüssel auch vom Zabbix-Agent 2 der neuen Generation unterstützt werden. Siehe die zusätzlichen Datenpunktschlüssel, die Sie nur mit Agent 2 verwenden können.

Siehe auch: Mindestberechtigungen für Windows-Datenpunkte

Gemeinsame Datenpunkte

Die folgende Tabelle listet Zabbix-Agent-Datenpunkte auf, die unter Windows unterstützt werden und mit dem UNIX-Zabbix-Agent gemeinsam sind:

- Der Datenpunkt-Schlüssel ist ein Link zu den vollständigen Details des UNIX-Zabbix-Agent-Datenpunkts
- Windows-relevante Kommentare zu Datenpunkten sind enthalten

Item key	Description	Item group
log	Die Überwachung einer Protokolldatei. Dieser Datenpunkt wird für das Windows-Ereignisprotokoll nicht unterstützt. Der Parameter `persistent_dir` wird unter Windows nicht unterstützt.	Log monitoring
log.count	Die Anzahl der übereinstimmenden Zeilen in einer überwachten Protokolldatei. Dieser Datenpunkt wird für das Windows-Ereignisprotokoll nicht unterstützt. Der Parameter `persistent_dir` wird unter Windows nicht unterstützt.	
logrt	Die Überwachung einer rotierenden Protokolldatei. Dieser Datenpunkt wird für das Windows-Ereignisprotokoll nicht unterstützt. Der Parameter `persistent_dir` wird unter Windows nicht unterstützt.	
logrt.count	Die Anzahl der übereinstimmenden Zeilen in einer überwachten rotierenden Protokolldatei. Dieser Datenpunkt wird für das Windows-Ereignisprotokoll nicht unterstützt. Der Parameter `persistent_dir` wird unter Windows nicht unterstützt.	
modbus.get	Liest Modbus-Daten.	Modbus
net.dns	Prüft, ob der DNS-Dienst aktiv ist. Die Parameter `ip`, `timeout` und `count` werden unter Windows ignoriert, sofern nicht Zabbix agent 2 verwendet wird.	Network
net.dns.perf	Prüft die Leistung eines DNS-Dienstes. Die Parameter `ip`, `timeout` und `count` werden unter Windows ignoriert, sofern nicht Zabbix agent 2 verwendet wird.	
net.dns.record	Führt eine DNS-Abfrage aus. Die Parameter `ip`, `timeout` und `count` werden unter Windows ignoriert, sofern nicht Zabbix agent 2 verwendet wird.	
net.if.discovery	Die Liste der Netzwerkinterfaces. Einige Windows-Versionen (z. B. Server 2008) benötigen möglicherweise die neuesten Updates, um Nicht-ASCII-Zeichen in Interfacenamen zu unterstützen.	
net.if.in	Die Statistiken des eingehenden Datenverkehrs auf einem Netzwerkinterface. Unter Windows bezieht der Datenpunkt Werte aus 64-Bit-Zählern, sofern verfügbar. 64-Bit-Zähler für Interface-Statistiken wurden in Windows Vista und Windows Server 2008 eingeführt. Wenn keine 64-Bit-Zähler verfügbar sind, verwendet der Agent 32-Bit-Zähler. Mehrbyte-Interfacenamen werden unter Windows unterstützt. Sie können Beschreibungen von Netzwerkinterfaces unter Windows mit den Datenpunkten net.if.discovery oder net.if.list abrufen.	
net.if.out	Die Statistiken des ausgehenden Datenverkehrs auf einem Netzwerkinterface. Unter Windows bezieht der Datenpunkt Werte aus 64-Bit-Zählern, sofern verfügbar. 64-Bit-Zähler für Interface-Statistiken wurden in Windows Vista und Windows Server 2008 eingeführt. Wenn keine 64-Bit-Zähler verfügbar sind, verwendet der Agent 32-Bit-Zähler. Mehrbyte-Interfacenamen werden unter Windows unterstützt. Sie können Beschreibungen von Netzwerkinterfaces unter Windows mit den Datenpunkten net.if.discovery oder net.if.list abrufen.	
net.if.total	Die Summe der Statistiken des eingehenden und ausgehenden Datenverkehrs auf einem Netzwerkinterface. Unter Windows bezieht der Datenpunkt Werte aus 64-Bit-Zählern, sofern verfügbar. 64-Bit-Zähler für Interface-Statistiken wurden in Windows Vista und Windows Server 2008 eingeführt. Wenn keine 64-Bit-Zähler verfügbar sind, verwendet der Agent 32-Bit-Zähler. Mehrbyte-Interfacenamen werden unter Windows unterstützt. Sie können Beschreibungen von Netzwerkinterfaces unter Windows mit den Datenpunkten net.if.discovery oder net.if.list abrufen.	

Item key	Description	Item group
net.tcp.listen	Prüft, ob sich dieser TCP-Port im Zustand LISTEN befindet.	
net.tcp.port	Prüft, ob eine TCP-Verbindung zum angegebenen Port hergestellt werden kann.	
net.tcp.service	Prüft, ob ein Dienst ausgeführt wird und TCP-Verbindungen annimmt. Die Prüfung von LDAP und HTTPS unter Windows wird nur von Zabbix agent 2 unterstützt.	
net.tcp.service.perf	Prüft die Leistung eines TCP-Dienstes. Die Prüfung von LDAP und HTTPS unter Windows wird nur von Zabbix agent 2 unterstützt.	
net.tcp.socket.count	Gibt die Anzahl der TCP-Sockets zurück, die den Parametern entsprechen. Dieser Datenpunkt wird unter Linux von Zabbix agent unterstützt, unter Windows jedoch nur von Zabbix agent 2 auf 64-Bit-Windows.	
net.udp.service	Prüft, ob ein Dienst ausgeführt wird und auf UDP-Anfragen antwortet.	
net.udp.service.perf	Prüft die Leistung eines UDP-Dienstes.	
net.udp.socket.count	Gibt die Anzahl der UDP-Sockets zurück, die den Parametern entsprechen. Dieser Datenpunkt wird unter Linux von Zabbix agent unterstützt, unter Windows jedoch nur von Zabbix agent 2 auf 64-Bit-Windows.	
proc.get	Die Liste der Betriebssystemprozesse und ihrer Parameter. Der Parameter `cmdline` wird unter Windows nicht unterstützt.	Processes
proc.num	Die Anzahl der Prozesse. Unter Windows werden nur die Parameter `name` und `user` unterstützt.	
system.cpu.discovery	Die Liste der erkannten CPUs/CPU-Kerne.	System
system.cpu.load	Die CPU-Auslastung. Wenn auf Zabbix agent ein Collector-Prozess gestartet wird, werden die folgenden Leistungsindikatoren initialisiert und später für diesen Datenpunkt verwendet: `\System\Processor Queue Length`	
system.cpu.num	Die Anzahl der CPUs.	
system.cpu.util	Der Prozentsatz der CPU-Auslastung. Der Wert wird mithilfe des Leistungsindikators *Processor Time* ermittelt. Beachten Sie, dass der Task-Manager seit Windows 8 die CPU-Auslastung auf Basis des Leistungsindikators *Processor Utility* anzeigt, während in früheren Versionen der Zähler *Processor Time* verwendet wurde (siehe weitere Details). *system* ist der einzige unter Windows unterstützte Parameter `type`.	
system.hostname	Der Hostname des Systems. Der Wert wird unter Windows entweder über die Funktionen `GetComputerName()` (für **netbios**), `GetComputerNameExA()` (für **fqdn**) oder `gethostname()` (für **host**) ermittelt. Siehe auch eine detailliertere Beschreibung.	
system.localtime	Die Systemzeit.	
system.run	Führt den angegebenen Befehl auf dem Host aus.	
system.sw.arch	Informationen zur Softwarearchitektur.	
system.sw.os	Informationen zum Betriebssystem.	
system.sw.os.get	Detaillierte Informationen zum Betriebssystem (Version, Typ, Distributionsname, Neben- und Hauptversion usw.).	
system.swap.size	Die Größe des Swap-Speichers in Bytes oder als Prozentsatz des Gesamtspeichers. Der Typparameter `pused` wird unter Linux von Zabbix agent unterstützt, unter Windows jedoch nur von Zabbix agent 2. Beachten Sie, dass dieser Schlüssel auf virtualisierten Windows-Plattformen (VMware ESXi, VirtualBox) möglicherweise eine falsche Swap-Größe bzw. einen falschen Prozentsatz meldet. In diesem Fall können Sie den Schlüssel `perf_counter[\700(_Total)\702]` verwenden, um den korrekten Swap-Prozentsatz zu ermitteln.	
system.uname	Identifikation des Systems. Unter Windows wird der Wert für diesen Datenpunkt aus den WMI-Klassen `Win32_OperatingSystem` und `Win32_Processor` ermittelt. Der OS-Name (einschließlich Edition) kann in die Anzeigesprache des Benutzers übersetzt sein. In einigen Windows-Versionen enthält er Markensymbole und zusätzliche Leerzeichen.	
system.uptime	Die Systemlaufzeit in Sekunden.	
vfs.dir.count	Die Anzahl der Verzeichniseinträge. Unter Windows werden Verzeichnis-Symlinks übersprungen und Hardlinks nur einmal gezählt.	Virtual file systems

Item key	Description	Item group
vfs.dir.get	Die Liste der Verzeichniseinträge. Unter Windows werden Verzeichnis-Symlinks übersprungen und Hardlinks nur einmal gezählt.	
vfs.dir.size	Die Größe des Verzeichnisses. Unter Windows werden alle Symlinks übersprungen und Hardlinks nur einmal berücksichtigt.	
vfs.file.cksum	Die Dateiprüfsumme, berechnet mit dem UNIX-Algorithmus cksum.	
vfs.file.contents	Abrufen des Inhalts einer Datei.	
vfs.file.exists	Prüft, ob die Datei vorhanden ist. Unter Windows müssen doppelte Anführungszeichen mit einem Backslash \\ maskiert werden, und der gesamte Datenpunkt-Schlüssel muss in doppelte Anführungszeichen gesetzt werden, wenn das Befehlszeilenprogramm zum Aufruf von `zabbix_get.exe` oder `agent2` verwendet wird. Beachten Sie, dass der Datenpunkt unter Windows nicht unterstützt werden kann, wenn innerhalb eines nicht vorhandenen Verzeichnisses nach einem Verzeichnis gesucht wird, z. B. `vfs.file.exists[C:\no\dir,dir]` (wobei 'no' nicht existiert).	
vfs.file.get	Gibt Informationen zu einer Datei zurück. Unter Windows unterstützte Dateitypen: reguläre Datei, Verzeichnis, symbolischer Link	
vfs.file.md5sum	Die MD5-Prüfsumme der Datei.	
vfs.file.owner	Ruft den Besitzer einer Datei ab.	
vfs.file.regexp	Ruft eine Zeichenfolge in der Datei ab.	
vfs.file.regmatch	Findet eine Zeichenfolge in der Datei.	
vfs.file.size	Die Dateigröße.	
vfs.file.time	Die Dateizeitinformationen. Unter Windows XP kann `vfs.file.time[file,change]` gleich `vfs.file.time[file,access]` sein.	
vfs.fs.discovery	Die Liste der eingehängten Dateisysteme mit ihrem Typ und den Einhängeoptionen. Das Makro `{#FSLABEL}` wird unter Windows unterstützt.	
vfs.fs.get	Die Liste der eingehängten Dateisysteme mit ihrem Typ, verfügbarem Speicherplatz, Inode-Statistiken und Einhängeoptionen. Das Makro {#FSLABEL} wird unter Windows unterstützt.	
vfs.fs.size	Der Speicherplatz in Bytes oder als Prozentsatz des Gesamtspeichers.	
vm.memory.size	Die Speichergröße in Bytes oder als Prozentsatz des Gesamtspeichers.	Virtual memory
web.page.get	Ruft den Inhalt einer Webseite ab.	Web monitoring
web.page.perf	Die Ladezeit einer vollständigen Webseite.	
web.page.regexp	Findet eine Zeichenfolge auf der Webseite.	
agent.hostmetadata	Die Host-Metadaten des Agent.	Zabbix
agent.hostname	Der Hostname des Agent.	
agent.ping	Die Verfügbarkeitsprüfung des Agent.	
agent.variant	Die Variante des Zabbix agent (Zabbix agent oder Zabbix agent 2).	
agent.version	Die Version des Zabbix agent.	
zabbix.stats	Gibt eine Reihe interner Metriken von Zabbix Server oder Proxy aus der Ferne zurück.	
zabbix.stats (queue)	Gibt die Anzahl der überwachten Datenpunkte in der Warteschlange zurück, die auf Zabbix Server oder Proxy aus der Ferne verzögert sind.	

Windows-spezifische Datenpunkte

Die Tabelle enthält Details zu den Datenpunktschlüsseln, die **nur** vom Windows-Zabbix-Agent unterstützt werden.

Windows-spezifische Datenpunkte sind manchmal ein ungefähres Gegenstück zu einem ähnlichen Agent-Datenpunkt; zum Beispiel entspricht `proc_info`, das unter Windows unterstützt wird, in etwa dem Datenpunkt `proc.mem`, der unter Windows nicht unterstützt wird.

Der Datenpunktschlüssel ist ein Link zu den vollständigen Details des Datenpunktschlüssels.

Datenpunktschlüssel	Beschreibung	Datenpunktgruppe
eventlog	Die Überwachung des Windows-Ereignisprotokolls.	Log-Überwachung
eventlog.count	Die Anzahl der Zeilen im Windows-Ereignisprotokoll.	
net.if.list	Die Liste der Netzwerkschnittstellen (einschließlich Schnittstellentyp, Status, IPv4-Adresse, Beschreibung).	Netzwerk
perf_counter	Der Wert eines beliebigen Windows-Leistungsindikators.	Leistungsindikatoren
perf_counter_en	Der Wert eines beliebigen Windows-Leistungsindikators auf Englisch.	
perf_instance.discovery	Die Liste der Objektinstanzen von Windows-Leistungsindikatoren.	
perf_instance_en.discovery	Die Liste der Objektinstanzen von Windows-Leistungsindikatoren, ermittelt unter Verwendung der Objektnamen auf Englisch.	
proc_info	Verschiedene Informationen über bestimmte Prozesse.	Prozesse
registry.data	Gibt Daten für den angegebenen Wertnamen im Windows-Registrierungsschlüssel zurück.	Registrierung
registry.get	Die Liste der Windows-Registrierungswerte oder -schlüssel am angegebenen Schlüssel.	
service.discovery	Die Liste der Windows-Dienste.	Dienste
service.info	Informationen über einen Dienst.	
services	Die Auflistung der Dienste.	
vm.vmemory.size	Die Größe des virtuellen Speichers in Byte oder als Prozentsatz der Gesamtsumme.	Virtueller Speicher
wmi.get	Führt eine WMI-Abfrage aus und gibt das erste ausgewählte Objekt zurück.	WMI
wmi.getall	Führt eine WMI-Abfrage aus und gibt die gesamte Antwort zurück.	

Details zum Datenpunktschlüssel

Parameter ohne spitze Klammern sind obligatorisch. Parameter, die mit spitzen Klammern **< >** gekennzeichnet sind, sind optional.

eventlog[name,<regexp>,<severity>,<source>,<eventid>,<maxlines>,<mode>]

 Die Überwachung des Ereignisprotokolls.
 Rückgabewert: *Log*.

Parameter:

- **name** - der Name des Ereignisprotokoll-Kanals (*Log Name* in der GUI des Event Viewers);
- **regexp** - ein regulärer Ausdruck, der das erforderliche Muster beschreibt (Groß-/Kleinschreibung wird beachtet);
- **severity** - ein regulärer Ausdruck, der den Schweregrad beschreibt (Groß-/Kleinschreibung wird nicht beachtet). Dieser Parameter akzeptiert einen regulären Ausdruck auf Basis der folgenden Werte: ”Information”, ”Warning”, ”Error”, ”Critical”, ”Verbose” (unter Windows Vista oder neuer).
- **source** - ein regulärer Ausdruck, der die Quellkennung beschreibt (Groß-/Kleinschreibung wird nicht beachtet);
- **eventid** - ein regulärer Ausdruck, der die Ereigniskennung(en) beschreibt (Groß-/Kleinschreibung wird beachtet);
- **maxlines** - die maximale Anzahl neuer Zeilen pro Sekunde, die der Agent an den Zabbix Server oder Proxy sendet. Dieser Parameter überschreibt den Wert von `MaxLinesPerSecond` in `zabbix_agentd.conf`.
- **mode** - mögliche Werte:
  - *all* (Standard);
  - *skip* - die Verarbeitung älterer Daten überspringen (wirkt sich nur auf neu erstellte Datenpunkte aus).

Kommentare:

- Der Datenpunkt muss als aktiver Check konfiguriert sein;
- Der Agent kann keine Ereignisse aus dem Protokoll ”Forwarded events” senden;
- Windows Eventing 6.0 wird unterstützt;
- Wenn für diesen Datenpunkt ein anderer Informationstyp als Log ausgewählt wird, gehen der lokale Zeitstempel sowie die Informationen zu Protokollschweregrad und Quelle verloren;
- Siehe auch zusätzliche Informationen zur Protokollüberwachung.

Beispiele:

```
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] #hier wird ein benutzerdefinierter regulärer Ausdruck mit dem Namen `TWOSHOR
```

eventlog.count[name,<regexp>,<severity>,<source>,<eventid>,<maxproclines>,<mode>]

 Die Anzahl der Zeilen im Windows-Ereignisprotokoll.
 Rückgabewert: *Integer*.

Parameter:

- **name** - der Name des Ereignisprotokoll-Kanals (*Log Name* in der GUI des Event Viewers);
- **regexp** - ein regulärer Ausdruck, der das erforderliche Muster beschreibt (Groß-/Kleinschreibung wird beachtet);
- **severity** - ein regulärer Ausdruck, der den Schweregrad beschreibt (Groß-/Kleinschreibung wird nicht beachtet). Dieser Parameter akzeptiert einen regulären Ausdruck auf Basis der folgenden Werte: ”Information”, ”Warning”, ”Error”, ”Critical”, ”Verbose” (unter Windows Vista oder neuer ausgeführt).
- **source** - ein regulärer Ausdruck, der die Quellkennung beschreibt (Groß-/Kleinschreibung wird nicht beachtet);
- **eventid** - ein regulärer Ausdruck, der die Ereigniskennung(en) beschreibt (Groß-/Kleinschreibung wird beachtet);
- **maxproclines** - die maximale Anzahl neuer Zeilen pro Sekunde, die der Agent analysiert (darf 10000 nicht überschreiten). Der Standardwert ist 10*`MaxLinesPerSecond` in `zabbix_agentd.conf`.
- **mode** - mögliche Werte:
  - *all* (Standard);
  - *skip* - die Verarbeitung älterer Daten überspringen (wirkt sich nur auf neu erstellte Datenpunkte aus).

Kommentare:

- Der Datenpunkt muss als aktiver Check konfiguriert sein;
- Der Agent kann keine Ereignisse aus dem Protokoll ”Forwarded events” senden;
- Windows Eventing 6.0 wird unterstützt;
- Die Auswahl eines anderen Informationstyps als Log für diesen Datenpunkt führt zum Verlust des lokalen Zeitstempels sowie der Informationen zu Protokollschweregrad und Quelle;
- Siehe auch zusätzliche Informationen zur Protokollüberwachung.

Beispiele:

```
eventlog.count[System,,"Warning|Error"]
eventlog.count[Windows PowerShell,,,,,,skip]
```

net.if.list

 Die Liste der Netzwerkschnittstellen (einschließlich Schnittstellentyp, Status, IPv4-Adresse, Beschreibung).
 Rückgabewert: *Text*.

Kommentare:

- Schnittstellennamen mit mehreren Bytes werden unterstützt;
- Deaktivierte Schnittstellen werden nicht aufgelistet;
- Das Aktivieren/Deaktivieren einiger Komponenten kann ihre Reihenfolge im Windows-Schnittstellennamen ändern;
- Einige Windows-Versionen (zum Beispiel Server 2008) erfordern möglicherweise die Installation der neuesten Updates, um Nicht-ASCII-Zeichen in Schnittstellennamen zu unterstützen.

perf_counter[counter,<interval>]

 Der Wert eines beliebigen Windows-Leistungsindikators.
 Rückgabewert: *Integer*, *float*, *string* oder *text* (je nach Anfrage).

Parameter:

- **counter** - der Pfad zum Leistungsindikator;
- **interval** - die letzten N Sekunden zur Speicherung des Durchschnittswerts. Der Wert von `interval` muss zwischen 1 und 900 Sekunden liegen (einschließlich), und der Standardwert ist 1.

Kommentare:

- Dieser Datenpunkt folgt UNC-Pfaden, was ein Sicherheitsrisiko darstellen kann;
- `interval` wird für Leistungsindikatoren verwendet, die mehr als eine Messprobe erfordern (z. B. CPU-Auslastung), sodass die Prüfung jedes Mal einen Durchschnittswert für die letzten ”interval” Sekunden zurückgibt;
- Der Performance Monitor kann verwendet werden, um die Liste der verfügbaren Leistungsindikatoren zu erhalten.
- Siehe auch: Windows performance counters.

perf_counter_en[counter,<interval>]

 Der Wert eines beliebigen Windows-Leistungsindikators in englischer Sprache.
 Rückgabewert: *Integer*, *float*, *string* oder *text* (je nach Anfrage).

Parameter:

- **counter** - der Pfad zum Leistungsindikator in englischer Sprache;

- **interval** - die letzten N Sekunden zur Speicherung des Durchschnittswerts. Das `interval` muss zwischen 1 und 900 Sekunden (einschließlich) liegen, und der Standardwert ist 1.

Kommentare:

- Dieser Datenpunkt folgt UNC-Pfaden, was ein Sicherheitsrisiko darstellen kann;
- `interval` wird für Leistungsindikatoren verwendet, die mehr als eine Messprobe erfordern (wie die CPU-Auslastung), sodass die Prüfung jedes Mal einen Durchschnittswert der letzten "interval" Sekunden zurückgibt;
- Dieser Datenpunkt wird nur unter **Windows Server 2008/Vista** und höher unterstützt;
- Die Liste der englischen Zeichenfolgen finden Sie, indem Sie den folgenden Registrierungsschlüssel anzeigen: `HKEY_LOCAL_MACHINE\S NT\CurrentVersion\Perflib\009`.

perf_instance.discovery[object]

 Die Liste der Objektinstanzen von Windows-Leistungsindikatoren. Wird für Low-Level-Discovery verwendet.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **object** - der Objektname (lokalisiert).

perf_instance_en.discovery[object]

 Die Liste der Objektinstanzen von Windows-Leistungsindikatoren, die unter Verwendung der Objektnamen auf Englisch erkannt werden. Wird für Low-Level-Discovery verwendet.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **object** - der Objektname (auf Englisch).

proc_info[process,<attribute>,<type>]

 Verschiedene Informationen über bestimmte Prozesse.
 Rückgabewert: *Float*.

Parameter:

- **process** - der Prozessname;
- **attribute** - das angeforderte Prozessattribut;
- **type** - der Darstellungstyp (sinnvoll, wenn mehr als ein Prozess mit demselben Namen vorhanden ist)

Kommentare:

- Die folgenden `attributes` werden unterstützt:
  - *vmsize* (Standard) - Größe des virtuellen Speichers des Prozesses in KByte;
  - *wkset* - Größe des Working Sets des Prozesses (Menge des vom Prozess verwendeten physischen Speichers) in KByte;
  - *pf* - Anzahl der Page Faults;
  - *ktime* - Kernelzeit des Prozesses in Millisekunden;
  - *utime* - Benutzerzeit des Prozesses in Millisekunden;
  - *io_read_b* - Anzahl der vom Prozess während I/O-Operationen gelesenen Bytes;
  - *io_read_op* - Anzahl der vom Prozess ausgeführten Leseoperationen;
  - *io_write_b* - Anzahl der vom Prozess während I/O-Operationen geschriebenen Bytes;
  - *io_write_op* - Anzahl der vom Prozess ausgeführten Schreiboperationen;
  - *io_other_b* - Anzahl der vom Prozess während anderer Operationen als Lese- und Schreiboperationen übertragenen Bytes;
  - *io_other_op* - Anzahl der vom Prozess ausgeführten I/O-Operationen, die keine Lese- oder Schreiboperationen sind;
  - *gdiobj* - Anzahl der vom Prozess verwendeten GDI-Objekte;
  - *userobj* - Anzahl der vom Prozess verwendeten USER-Objekte.
- Gültige `types` sind:
  - *avg* (Standard) - Durchschnittswert für alle Prozesse mit dem Namen <process>;
  - *min* - Minimalwert unter allen Prozessen mit dem Namen <process>;
  - *max* - Maximalwert unter allen Prozessen mit dem Namen <process>;
  - *sum* - Summe der Werte aller Prozesse mit dem Namen <process>.
- Auf einem 64-Bit-System ist für die korrekte Funktion dieses Datenpunkts ein 64-Bit-Zabbix-Agent erforderlich.

Beispiele:

```
proc_info[iexplore.exe,wkset,sum] #ermittelt die Menge des physischen Speichers, die von allen Internet-Ex
proc_info[iexplore.exe,pf,avg] #ermittelt die durchschnittliche Anzahl der Page Faults für Internet-Explor
```

registry.data[key,<value name>]

 Gibt Daten für den angegebenen Wertnamen im Windows-Registrierungsschlüssel zurück.
 Rückgabewert: *Integer*, *string* oder *text* (abhängig vom Werttyp)

Parameter:

- **key** - der Registrierungsschlüssel einschließlich des Stamm-Schlüssels; Stammabkürzungen (z. B. HKLM) sind zulässig;
- **value name** - der Name des Registrierungswerts im Schlüssel (standardmäßig leerer String ””). Der Standardwert wird zurückgegeben, wenn der Wertname nicht angegeben ist.

Kommentare:

- Unterstützte Stammabkürzungen:
  - 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
- Schlüssel mit Leerzeichen müssen in doppelte Anführungszeichen gesetzt werden.

Beispiele:

```
registry.data["HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Windows Error Reporting"] #gibt die Daten des
registry.data["HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting","EnableZip"] #gibt die Daten des W
```

registry.get[key,<mode>,<name regexp>]

 Die Liste der Windows-Registrierungswerte oder -Schlüssel, die sich unter dem angegebenen Schlüssel befinden.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **key** - der Registrierungs-Schlüssel einschließlich des Stamm-Schlüssels; Stammabkürzungen (z. B. HKLM) sind zulässig (siehe Kommentare zu `registry.data[]` für die vollständige Liste der Abkürzungen);
- **mode** - mögliche Werte:
values (Standard) oder *keys*;
- **name regexp** - ermittelt nur Werte, deren Namen mit dem regulären Ausdruck übereinstimmen (Standard - ermittelt alle Werte). Nur mit *values* als *mode* zulässig.

Schlüssel mit Leerzeichen müssen in doppelte Anführungszeichen gesetzt werden.

Beispiele:

```
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,values,"^DisplayName|DisplayVersion$
Das JSON enthält Details zum Schlüssel, zum letzten Unterschlüssel, zum Wertnamen, zum Werttyp und zu den
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,values] #gibt die Daten aller Werte
Das JSON enthält Details zum Schlüssel, zum letzten Unterschlüssel, zum Wertnamen, zum Werttyp und zu den
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,keys] #gibt alle Unterschlüssel dies
Das JSON enthält Details zum Schlüssel und zum letzten Unterschlüssel.
```

service.discovery

 Die Liste der Windows-Dienste. Wird für Low-Level-Discovery verwendet.
 Rückgabewert: *JSON-Objekt*.

service.info[service,<param>]

 Informationen zu einem Dienst.
 Rückgabewert: - *Integer* - mit `param` als *state*, *startup*;

 Speziell für *state*: - 0 - läuft - 1 - angehalten - 2 - Start ausstehend - 3 - Pause ausstehend - 4 - Fortsetzen ausstehend - 5 - Stopp ausstehend - 6 - gestoppt - 7 - unbekannt - 255 - kein solcher Dienst

 Speziell für *startup*: - 0 - automatisch - 1 - automatisch verzögert - 2 - manuell - 3 - deaktiviert - 4 - unbekannt - 5 - automatischer Trigger-Start - 6 - automatischer verzögerter Trigger-Start - 7 - manueller Trigger-Start - *String* - mit `param` als *displayname*, *path*, *user*; - *Text* - mit `param` als *description*.

Parameter:

- **service** - ein tatsächlicher Dienstname oder sein Anzeigename, wie er im MMC-Snap-In ”Dienste” angezeigt wird;
- **param** - *state* (Standard), *displayname*, *path*, *user*, *startup* oder *description*.

Kommentare:

- Elemente wie `service.info[service,state]` und `service.info[service]` geben dieselben Informationen zurück;
- Nur mit `param` als *state* gibt dieses Element einen Wert für nicht vorhandene Dienste zurück (255).

Beispiele:

```
service.info[SNMPTRAP] - Zustand des Dienstes SNMPTRAP;
service.info[SNMP Trap] - Zustand desselben Dienstes, aber mit angegebenem Anzeigenamen;
service.info[EventLog,startup] - der Starttyp des Dienstes EventLog
```

services[<type>,<state>,<exclude>]

 Die Auflistung der Dienste.
 Rückgabewert: *0* - wenn leer; *Text* - die durch einen Zeilenumbruch getrennte Liste der Dienste.

Parameter:

- **type** - *all* (Standard), *automatic*, *manual* oder *disabled*;
- **state** - *all* (Standard), *stopped*, *started*, *start_pending*, *stop_pending*, *running*, *continue_pending*, *pause_pending* oder *paused*;
- **exclude** - die Dienste, die vom Ergebnis ausgeschlossen werden sollen. Ausgeschlossene Dienste müssen in doppelten Anführungszeichen und durch Kommas ohne Leerzeichen getrennt aufgeführt werden.

Beispiele:

```
services[,started] #gibt die Liste der gestarteten Dienste zurück;
services[automatic, stopped] #gibt die Liste der gestoppten Dienste zurück, die ausgeführt werden sollten;
services[automatic, stopped, "service1,service2,service3"] #gibt die Liste der gestoppten Dienste zurück,
```

vm.vmemory.size[<type>]

 Die Größe des virtuellen Speichers in Byte oder als Prozentsatz der Gesamtsumme.
 Rückgabewert: *Integer* - für Byte; *float* - für Prozentwerte.

Parameter:

- **type** - mögliche Werte: *available* (verfügbarer virtueller Speicher), *pavailable* (verfügbarer virtueller Speicher in Prozent), *pused* (genutzter virtueller Speicher in Prozent), *total* (gesamter virtueller Speicher, Standard) oder *used* (genutzter virtueller Speicher)

Kommentare:

- Die Überwachung der Statistiken zum virtuellen Speicher basiert auf:
  - dem gesamten virtuellen Speicher unter Windows (gesamter physischer Speicher + Größe der Auslagerungsdatei);
  - der maximalen Speichermenge, die der Zabbix Agent reservieren kann;
  - dem aktuellen Limit für reservierten Speicher des Systems oder des Zabbix Agent, je nachdem, welcher Wert kleiner ist.

Beispiel:

```
vm.vmemory.size[pavailable] #gibt den verfügbaren virtuellen Speicher in Prozent zurück
```

wmi.get[<namespace>,<query>]

 Führen Sie eine WMI-Abfrage aus und geben Sie das erste ausgewählte Objekt zurück.
 Rückgabewert: *Integer*, *float*, *string* oder *text* (abhängig von der Anfrage).

Parameter:

- **namespace** - der WMI-Namespace;
- **query** - die WMI-Abfrage, die ein einzelnes Objekt zurückgibt.

WMI-Abfragen werden mit WQL ausgeführt.

Beispiel:

```
wmi.get[root\cimv2,select status from Win32_DiskDrive where Name like '%PHYSICALDRIVE0%'] #gibt den Status
```

wmi.getall[<namespace>,<query>]

 Führt eine WMI-Abfrage aus und gibt die gesamte Antwort zurück. Kann für die Low-Level-Discovery verwendet werden.
 Rückgabewert: *JSON-Objekt*

Parameter:

- **namespace** - der WMI-Namespace;
- **query** - die WMI-Abfrage.

Kommentare:

- WMI-Abfragen werden mit WQL ausgeführt.
- Die JSONPath-Vorverarbeitung kann verwendet werden, um auf spezifischere Werte im zurückgegebenen JSON zu verweisen.

Beispiel:

```
wmi.getall[root\cimv2,select * from Win32_DiskDrive where Name like '%PHYSICALDRIVE%'] #gibt Statusinforma
```

Überwachung von Windows-Diensten

Dieses Tutorial enthält eine Schritt-für-Schritt-Anleitung zum Einrichten der Überwachung von Windows-Diensten. Es wird davon ausgegangen, dass Zabbix Server und Agent konfiguriert und betriebsbereit sind.

Schritt 1

Ermitteln Sie den Dienstnamen.

Sie können den Dienstnamen ermitteln, indem Sie das MMC-Snap-In „Dienste“ öffnen und die Eigenschaften des Dienstes aufrufen. Auf der Registerkarte *Allgemein* sollte ein Feld mit der Bezeichnung *Dienstname* angezeigt werden. Der folgende Wert ist der Name, den Sie beim Einrichten eines Datenpunkts für die Überwachung verwenden. Wenn Sie beispielsweise den Dienst „workstation“ überwachen möchten, könnte Ihr Dienstname **lanmanworkstation** lauten.

Schritt 2

Konfigurieren Sie einen Datenpunkt zur Überwachung des Dienstes.

Der Datenpunkt `service.info[service,<param>]` ruft Informationen über einen bestimmten Dienst ab. Geben Sie je nach den benötigten Informationen die Option `param` an, die die folgenden Werte akzeptiert: *displayname*, *state*, *path*, *user*, *startup* oder *description*. Der Standardwert ist *state*, wenn `param` nicht angegeben ist (`service.info[service]`).

Der Typ des Rückgabewerts hängt vom gewählten `param` ab: integer für *state* und *startup*; Zeichenkette für *displayname*, *path* und *user*; Text für *description*.

Beispiel:

- *Schlüssel:* `service.info[lanmanworkstation]`
- *Informationstyp:* Numerisch (ohne Vorzeichen)

Der Datenpunkt `service.info[lanmanworkstation]` ruft Informationen über den Status des Dienstes als numerischen Wert ab. Um einen numerischen Wert im Frontend einer Textdarstellung zuzuordnen („0“ als „Running“, „1“ als „Paused“ usw.), können Sie auf dem Host, auf dem der Datenpunkt konfiguriert ist, eine Wertzuordnung konfigurieren. Dazu verknüpfen Sie entweder die Vorlage *Windows services by Zabbix agent* oder *Windows services by Zabbix agent active* mit dem Host oder konfigurieren auf dem Host eine neue Wertzuordnung, die auf der Wertzuordnung *Windows service state* basiert, die in den genannten Vorlagen konfiguriert ist.

Beachten Sie, dass in beiden genannten Vorlagen eine Discovery-Regel konfiguriert ist, die Dienste automatisch erkennt. Wenn Sie dies nicht möchten, können Sie die Discovery-Regel deaktivieren auf Host-Ebene, nachdem die Vorlage mit dem Host verknüpft wurde.

Discovery von Windows-Diensten

Low-level discovery bietet eine Möglichkeit, automatisch Datenpunkte, Auslöser und Diagramme für verschiedene Entitäten auf einem Computer zu erstellen. Zabbix kann automatisch mit der Überwachung von Windows-Diensten auf Ihrem Rechner beginnen, ohne dass der genaue Name eines Dienstes bekannt sein muss oder für jeden Dienst manuell Datenpunkte erstellt werden müssen. Ein Filter kann verwendet werden, um echte Datenpunkte, Auslöser und Diagramme nur für Dienste von Interesse zu erzeugen.

Protokolldateiüberwachung

Übersicht

Zabbix kann für die zentrale Überwachung und Analyse von Protokolldateien mit oder ohne Unterstützung für Log-Rotation verwendet werden.

Benachrichtigungen können verwendet werden, um Benutzer zu warnen, wenn eine Protokolldatei bestimmte Zeichenfolgen oder Zeichenfolgenmuster enthält.

Um eine Protokolldatei zu überwachen, müssen Sie Folgendes haben:

- Zabbix Agent, der auf dem Host ausgeführt wird.
- Ein Log-Überwachungs-Datenpunkt eingerichtet.

Attention:
Die Größenbeschränkung einer überwachten Protokolldatei hängt von der Unterstützung großer Dateien ab.

Konfiguration

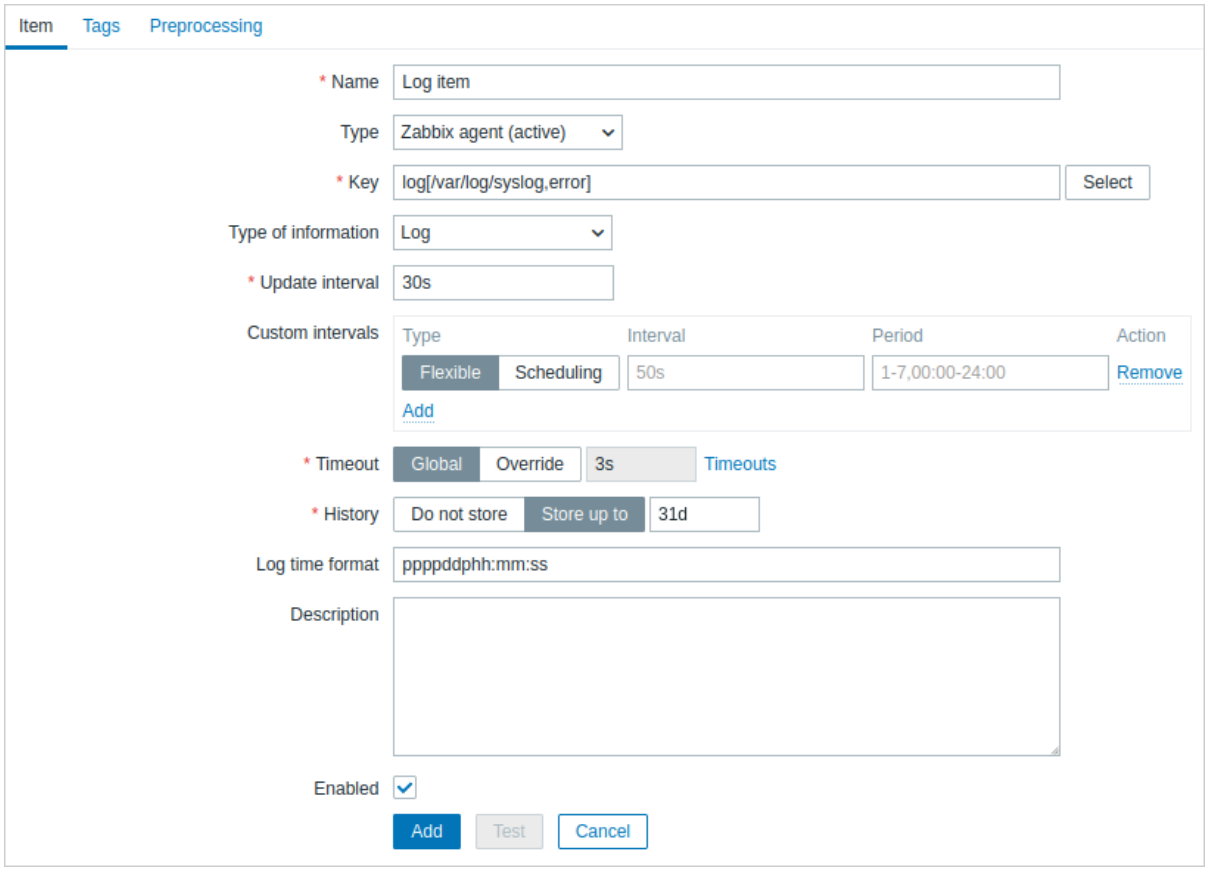
Agent-Parameter überprüfen

Stellen Sie sicher, dass in der Agent-Konfigurationsdatei:

- Der Parameter `Hostname` mit dem Hostnamen im Frontend übereinstimmt.
- Die im Parameter `ServerActive` angegebenen Server für die Verarbeitung aktiver Prüfungen festgelegt sind.

Datenpunktkonfiguration

Konfigurieren Sie einen Datenpunkt zur Protokollüberwachung.



Alle Pflichtfelder sind mit einem roten Sternchen gekennzeichnet.

Für Datenpunkte zur Protokollüberwachung geben Sie insbesondere Folgendes ein:

Typ	Wählen Sie hier **Zabbix agent (active)** aus.
Schlüssel	Verwenden Sie einen der folgenden Datenpunktschlüssel: `log[]` oder `logrt[]`: Mit diesen beiden Datenpunktschlüsseln können Sie Protokolle überwachen und Protokolleinträge anhand des Inhalts-regexp filtern, sofern vorhanden. Beispiel: `log[/var/log/syslog,error]`. Stellen Sie sicher, dass der Benutzer „zabbix“ Leseberechtigungen für die Datei besitzt. Andernfalls wird der Status des Datenpunkts auf „unsupported“ gesetzt. **log.count[]** oder **logrt.count[]**: Mit diesen beiden Datenpunktschlüsseln können Sie ausschließlich die Anzahl der übereinstimmenden Zeilen zurückgeben. Weitere Informationen zur Verwendung dieser Datenpunktschlüssel und ihrer Parameter finden Sie im Abschnitt zu den unterstützten Zabbix agent-Datenpunktschlüsseln.

Informationstyp	Wird automatisch vorausgefüllt: Für Datenpunkte mit `log[]` oder `logrt[]` - *Log*; Für Datenpunkte mit `log.count[]` oder `logrt.count[]` - *Numeric (unsigned)*. Wenn Sie optional den Parameter `output` verwenden, können Sie manuell einen geeigneten Informationstyp auswählen, der nicht *Log* ist. Beachten Sie, dass die Auswahl eines anderen Informationstyps als *Log* zum Verlust des lokalen Zeitstempels führt.
Aktualisierungsintervall (in Sek.)	Dieser Parameter legt fest, wie oft der Zabbix agent die Protokolldatei auf Änderungen überprüft. Wenn Sie den Wert auf 1 Sekunde setzen, erhalten Sie neue Datensätze so schnell wie möglich.
Zeitformat des Protokolls	In diesem Feld können Sie optional das Muster zum Parsen des Zeitstempels der Protokollzeile angeben. Unterstützte Platzhalter: * **y**: *Jahr (1970-2038)* * **M**: *Monat (01-12)* * **d**: *Tag (01-31)* * **h**: *Stunde (00-23)* * **m**: *Minute (00-59)* * **s**: *Sekunde (00-59)* Wenn das Feld leer bleibt, wird der Zeitstempel in der Unix-Zeit auf 0 gesetzt, was den `1. Januar` 1970 darstellt. Betrachten Sie beispielsweise die folgende Zeile aus der Protokolldatei des Zabbix agent: `23480:20100328:154718.045 Zabbix agent started. Zabbix 1.8.2 (revision 11211).` Sie beginnt mit sechs Zeichenpositionen für die PID, gefolgt von Datum, Uhrzeit und dem Rest der Meldung. Das Zeitformat des Protokolls für diese Zeile wäre „pppppp:yyyyMMdd:hhmmss“. Beachten Sie, dass die Zeichen „p“ und „:“ Platzhalter sind und jedes Zeichen außer „yMdhms“ sein können.

Wichtige Hinweise

- Der Server und der Agent verfolgen die Größe und den Zeitpunkt der letzten Änderung eines überwachten Protokolls (für `logrt`) in zwei Zählern. Zusätzlich:
  - Der Agent verwendet intern auch Inode-Nummern (unter UNIX/GNU/Linux), Dateiindizes (unter Microsoft Windows) und MD5-Summen der ersten 512 Bytes der Protokolldatei, um Entscheidungen zu verbessern, wenn Protokolldateien gekürzt und rotiert werden.
  - Auf UNIX/GNU/Linux-Systemen wird davon ausgegangen, dass die Dateisysteme, auf denen Protokolldateien gespeichert sind, Inode-Nummern melden, die zum Verfolgen von Dateien verwendet werden können.
  - Unter Microsoft Windows ermittelt der Zabbix Agent den Dateisystemtyp, auf dem sich die Protokolldateien befinden, und verwendet:
    * Auf NTFS-Dateisystemen 64-Bit-Dateiindizes.
    * Auf ReFS-Dateisystemen (nur ab Microsoft Windows Server 2012) 128-Bit-Datei-IDs.
    * Auf Dateisystemen, bei denen sich Dateiindizes ändern (z. B. FAT32, exFAT), wird ein Fallback-Algorithmus verwendet, um in unsicheren Situationen einen sinnvollen Ansatz zu wählen, wenn die Rotation von Protokolldateien zu mehreren Protokolldateien mit demselben Zeitpunkt der letzten Änderung führt.
  - Die Inode-Nummern, Dateiindizes und MD5-Summen werden intern vom Zabbix Agent erfasst. Sie werden nicht an den Zabbix Server übertragen und gehen verloren, wenn der Zabbix Agent beendet wird.
  - Ändern Sie nicht den Zeitpunkt der letzten Änderung einer Protokolldatei (zum Beispiel mit `touch`), und ersetzen Sie eine überwachte Protokolldatei nicht, indem Sie eine Datei unter ihrem ursprünglichen Namen zurückkopieren (dadurch wird ein neuer Inode erstellt). In beiden Fällen kann Zabbix die Datei als andere Datei behandeln und sie von Anfang an erneut einlesen, was doppelte Alarme verursachen kann.
  - Wenn es mehrere passende Protokolldateien für ein `logrt[]`-Element gibt und der Zabbix Agent die aktuellste davon verfolgt und diese aktuellste Protokolldatei gelöscht wird, wird die Warnmeldung `"there are no files matching "<regexp mask>" in "<directory>"` protokolliert. Der Zabbix Agent ignoriert Protokolldateien mit einem Änderungszeitpunkt, der kleiner ist als der vom Agenten zuletzt für das geprüfte `logrt[]`-Element gesehene Änderungszeitpunkt.
- Der Agent beginnt mit dem Lesen der Protokolldatei an der Stelle, an der er beim vorherigen Mal aufgehört hat.
- Die Anzahl der bereits analysierten Bytes (der Größenzähler) und der Zeitpunkt der letzten Änderung (der Zeitzähler) werden in der Zabbix-Datenbank gespeichert und an den Agenten gesendet, damit sichergestellt ist, dass der Agent das Lesen der Protokolldatei ab diesem Punkt beginnt, wenn der Agent gerade gestartet wurde oder Elemente erhalten hat, die zuvor deaktiviert oder nicht unterstützt waren. Wenn der Agent jedoch einen Größenzähler ungleich null vom Server erhalten hat, das `logrt[]`- oder `logrt.count[]`-Element aber keine passenden Dateien finden kann, wird der Größenzähler auf 0 zurückgesetzt, damit ab Beginn analysiert wird, falls die Dateien später erscheinen.

- Immer wenn die Protokolldatei kleiner wird als der dem Agenten bekannte Protokollgrößenzähler, wird der Zähler auf null zurückgesetzt und der Agent beginnt mit dem Lesen der Protokolldatei von Anfang an, wobei der Zeitzähler berücksichtigt wird.
- Wenn es im Verzeichnis mehrere passende Dateien mit demselben Zeitpunkt der letzten Änderung gibt, versucht der Agent, alle Protokolldateien mit demselben Änderungszeitpunkt korrekt zu analysieren und das Überspringen von Daten oder eine doppelte Analyse derselben Daten zu vermeiden, auch wenn dies nicht in allen Situationen garantiert werden kann. Der Agent geht von keinem bestimmten Rotationsschema für Protokolldateien aus und bestimmt auch keines. Wenn mehrere Protokolldateien mit demselben Zeitpunkt der letzten Änderung vorhanden sind, verarbeitet der Agent sie in lexikografisch absteigender Reihenfolge. Daher werden die Protokolldateien bei einigen Rotationsschemata in ihrer ursprünglichen Reihenfolge analysiert und gemeldet. Bei anderen Rotationsschemata wird die ursprüngliche Reihenfolge der Protokolldateien nicht eingehalten, was dazu führen kann, dass übereinstimmende Protokolldateieinträge in veränderter Reihenfolge gemeldet werden (das Problem tritt nicht auf, wenn die Protokolldateien unterschiedliche Zeitpunkte der letzten Änderung haben).
- Der Zabbix Agent verarbeitet neue Einträge einer Protokolldatei einmal pro *Update interval* Sekunden.
- Der Zabbix Agent sendet pro Sekunde nicht mehr als `maxlines` einer Protokolldatei. Die Begrenzung verhindert eine Überlastung von Netzwerk- und CPU-Ressourcen und überschreibt den Standardwert, der durch den Parameter `MaxLinesPerSecond` in der Agent-Konfigurationsdatei vorgegeben wird.
- Um die erforderliche Zeichenfolge zu finden, verarbeitet Zabbix 10-mal mehr neue Zeilen als in `MaxLinesPerSecond` festgelegt. Wenn also beispielsweise ein `log[]`- oder `logrt[]`-Element ein *Update interval* von 1 Sekunde hat, analysiert der Agent standardmäßig nicht mehr als 200 Protokolldateieinträge und sendet nicht mehr als 20 übereinstimmende Einträge in einem Prüfvorgang an den Zabbix Server. Durch Erhöhen von `MaxLinesPerSecond` in der Agent-Konfigurationsdatei oder durch Setzen des Parameters `maxlines` im Elementschlüssel kann die Begrenzung auf bis zu 10000 analysierte Protokolldateieinträge und 1000 an den Zabbix Server gesendete übereinstimmende Einträge pro Prüfvorgang erhöht werden. Wenn das *Update interval* auf 2 Sekunden gesetzt ist, wären die Grenzen für einen Prüfvorgang doppelt so hoch wie bei einem *Update interval* von 1 Sekunde.
- Zusätzlich sind `log`- und `log.count`-Werte immer auf 50% der Größe des Sendepuffers des Agenten begrenzt, selbst wenn sich darin keine Nicht-Protokoll-Werte befinden. Damit die `maxlines`-Werte in einer Verbindung (und nicht in mehreren Verbindungen) gesendet werden können, muss der Parameter `BufferSize` des Agenten mindestens `maxlines` x 2 betragen. Der Zabbix Agent kann während der Protokollerfassung Daten hochladen und dadurch den Puffer freigeben, während Zabbix Agent 2 die Protokollerfassung stoppt, bis die Daten hochgeladen und der Puffer freigegeben wurden; dies erfolgt asynchron.
- Wenn keine Protokollelemente vorhanden sind, wird die gesamte Puffergröße des Agenten für Nicht-Protokoll-Werte verwendet. Wenn Protokollwerte eintreffen, ersetzen sie bei Bedarf die älteren Nicht-Protokoll-Werte, bis zu den vorgesehenen 50%.
- Bei Protokolldateieinträgen, die länger als 256 kB sind, werden nur die ersten 256 kB mit dem regulären Ausdruck abgeglichen, und der Rest des Eintrags wird ignoriert. Wenn der Zabbix Agent jedoch beendet wird, während er einen langen Eintrag verarbeitet, geht der interne Zustand des Agenten verloren, und der lange Eintrag kann nach dem erneuten Start des Agenten erneut und anders analysiert werden.
- Besondere Hinweise zu \ als Pfadtrennzeichen: Wenn `file_format file\.log` ist, sollte es kein Verzeichnis `file` geben, da nicht eindeutig festgelegt werden kann, ob `.` maskiert ist oder das erste Zeichen des Dateinamens darstellt.
- Reguläre Ausdrücke für `logrt` werden nur im Dateinamen unterstützt, ein Abgleich regulärer Ausdrücke für Verzeichnisse wird nicht unterstützt.
- Auf UNIX-Plattformen wird ein `logrt[]`-Element zu NOTSUPPORTED, wenn ein Verzeichnis, in dem die Protokolldateien erwartet werden, nicht existiert.
- Unter Microsoft Windows wird das Element nicht zu NOTSUPPORTED, wenn ein Verzeichnis nicht existiert (zum Beispiel, wenn das Verzeichnis im Elementschlüssel falsch geschrieben ist).
- Das Fehlen von Protokolldateien für ein `logrt[]`-Element führt nicht dazu, dass es NOTSUPPORTED wird. Fehler beim Lesen von Protokolldateien für ein `logrt[]`-Element werden als Warnungen in die Protokolldatei des Zabbix Agenten geschrieben, führen aber nicht dazu, dass das Element NOTSUPPORTED wird.
- Die Protokolldatei des Zabbix Agenten kann hilfreich sein, um herauszufinden, warum ein `log[]`- oder `logrt[]`-Element NOTSUPPORTED geworden ist. Zabbix kann die Protokolldatei seines Agenten überwachen, außer bei `DebugLevel=4` oder `DebugLevel=5`.
- Die Suche nach einem Fragezeichen mit einem regulären Ausdruck, z. B. `\?`, kann zu Fehlalarmen führen, wenn die Textdatei NUL-Symbole enthält, da diese von Zabbix durch ? ersetzt werden, um die Verarbeitung der Zeile bis zum Zeilenumbruch fortzusetzen.

Extrahieren des übereinstimmenden Teils eines regulären Ausdrucks

Manchmal soll aus einer Zieldatei nur der relevante Wert extrahiert werden, anstatt bei einer Übereinstimmung mit einem regulären Ausdruck die gesamte Zeile zurückzugeben.

Log-Datenpunkte können gewünschte Werte aus übereinstimmenden Zeilen extrahieren. Dies wird durch den zusätzlichen Parameter **output** in `log`- und `logrt`-Datenpunkten ermöglicht.

Mit dem Parameter „output“ können Sie die „Erfassungsgruppe“ der Übereinstimmung angeben, die von Interesse ist.

Zum Beispiel:

```
log[/path/to/the/file,"large result buffer allocation.*Entries: ([0-9]+)",,,,\1]
```

sollte die Anzahl der Einträge zurückgeben, die im folgenden Inhalt gefunden wurde:

```
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
```

Es wird nur die Zahl zurückgegeben, da sich `\1` auf die erste und einzige Erfassungsgruppe `([0-9]+)` bezieht.

Durch die Möglichkeit, eine Zahl zu extrahieren und zurückzugeben, kann der Wert zum Definieren von Auslösern verwendet werden.

Verwendung des Parameters maxdelay

Der Parameter `maxdelay` in Log-Datenpunkten ermöglicht es, einige ältere Zeilen aus Logdateien zu ignorieren, damit die neuesten Zeilen innerhalb der `maxdelay` Sekunden analysiert werden.

> **Warning:**
> Das Festlegen von `maxdelay` > 0 kann dazu führen, dass **wichtige Einträge in Logdateien ignoriert und Alarme verpasst werden**. Verwenden Sie dies nur bei Bedarf und auf eigenes Risiko.

Standardmäßig folgen Datenpunkte für die Logüberwachung allen neuen Zeilen, die in den Logdateien erscheinen. Es gibt jedoch Anwendungen, die in manchen Situationen eine enorme Anzahl von Meldungen in ihre Logdateien schreiben. Wenn beispielsweise eine Datenbank oder ein DNS-Server nicht verfügbar ist, überfluten solche Anwendungen die Logdateien mit Tausenden nahezu identischer Fehlermeldungen, bis der Normalbetrieb wiederhergestellt ist. Standardmäßig werden all diese Meldungen gewissenhaft analysiert, und passende Zeilen werden wie in den `log`- und `logrt`-Datenpunkten konfiguriert an den Server gesendet.

Der integrierte Schutz vor Überlastung besteht aus einem konfigurierbaren `maxlines`-Parameter (schützt den Server vor zu vielen eingehenden passenden Logzeilen) und einem Limit von 10*`maxlines` (schützt CPU und I/O des Host vor Überlastung durch den Agent in einer Prüfung). Dennoch gibt es 2 Probleme mit diesem integrierten Schutz. Erstens wird eine große Anzahl potenziell wenig informativer Meldungen an den Server gemeldet und belegt Speicherplatz in der Datenbank. Zweitens kann der Agent aufgrund der begrenzten Anzahl pro Sekunde analysierter Zeilen den neuesten Logeinträgen stundenlang hinterherhinken. Wahrscheinlich möchten Sie lieber früher über den aktuellen Zustand in den Logdateien informiert werden, anstatt stundenlang alte Einträge durchzugehen.

Die Lösung für beide Probleme ist die Verwendung des Parameters `maxdelay`. Wenn `maxdelay` > 0 angegeben ist, werden bei jeder Prüfung die Anzahl der verarbeiteten Bytes, die Anzahl der verbleibenden Bytes und die Verarbeitungszeit gemessen. Aus diesen Werten berechnet der Agent eine geschätzte Verzögerung - also wie viele Sekunden es dauern würde, alle verbleibenden Einträge in einer Logdatei zu analysieren.

Wenn die Verzögerung `maxdelay` nicht überschreitet, fährt der Agent wie gewohnt mit der Analyse der Logdatei fort.

Wenn die Verzögerung größer als `maxdelay` ist, dann **ignoriert der Agent einen Abschnitt einer Logdatei, indem er darüber "springt"** und zu einer neuen geschätzten Position wechselt, sodass die verbleibenden Zeilen innerhalb von `maxdelay` Sekunden analysiert werden können.

Beachten Sie, dass der Agent die ignorierten Zeilen nicht einmal in den Puffer einliest, sondern eine ungefähre Position berechnet, zu der in der Datei gesprungen werden soll.

Das Überspringen von Logdateizeilen wird im Agent-Logfile wie folgt protokolliert:

```
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
```

Die Zahl bei "to byte" ist ungefähr, da der Agent nach dem "Sprung" die Position in der Datei an den Anfang einer Logzeile anpasst, die weiter in der Datei oder auch früher liegen kann.

Je nachdem, wie schnell die Datei wächst im Vergleich zur Geschwindigkeit der Analyse der Logdatei, sehen Sie möglicherweise keine "Sprünge", seltene oder häufige "Sprünge", große oder kleine "Sprünge" oder sogar bei jeder Prüfung einen kleinen "Sprung". Schwankungen in der Systemauslastung und der Netzwerklatenz beeinflussen ebenfalls die Berechnung der Verzögerung und damit das "Springen" nach vorne, um mit dem Parameter `maxdelay` Schritt zu halten.

Es wird nicht empfohlen, `maxdelay` < `update interval` zu setzen (dies kann zu häufigen kleinen "Sprüngen" führen).

Hinweise zum Umgang mit der Protokolldateirotation mit 'copytruncate'

logrt mit der Option copytruncate geht davon aus, dass unterschiedliche Protokolldateien unterschiedliche Datensätze enthalten (zumindest unterscheiden sich ihre Zeitstempel), daher werden die MD5-Summen der Anfangsblöcke (bis zu den ersten 512 Bytes) unterschiedlich sein. Zwei Dateien mit denselben MD5-Summen der Anfangsblöcke bedeuten, dass eine davon das Original und die andere eine Kopie ist.

logrt mit der Option copytruncate unternimmt den Versuch, Kopien von Protokolldateien korrekt zu verarbeiten, ohne Duplikate zu melden. Allerdings werden Dinge wie das Erzeugen mehrerer Kopien von Protokolldateien mit demselben Zeitstempel, eine Protokolldateirotation häufiger als das Aktualisierungsintervall des logrt[]-Datenpunkts sowie häufige Neustarts des Agent nicht empfohlen. Der Agent versucht, all diese Situationen möglichst sinnvoll zu behandeln, aber gute Ergebnisse können unter allen Umständen nicht garantiert werden.

Hinweise zu persistenten Dateien für log*[]-Datenpunkte

Zweck persistenter Dateien

Wenn der Zabbix Agent gestartet wird, erhält er eine Liste aktiver Prüfungen vom Zabbix Server oder Proxy. Für log*[]-Metriken erhält er die verarbeitete Protokollgröße und die Änderungszeit, um zu bestimmen, wo die Überwachung der Protokolldatei beginnen soll. Abhängig von der tatsächlichen Protokolldateigröße und der vom Dateisystem gemeldeten Änderungszeit entscheidet der Agent, ob die Überwachung der Protokolldatei ab der verarbeiteten Protokollgröße fortgesetzt oder die Protokolldatei von Anfang an erneut analysiert werden soll.

Ein laufender Agent verwaltet einen größeren Satz von Attributen, um alle überwachten Protokolldateien zwischen den Prüfungen nachzuverfolgen. Dieser Zustand im Arbeitsspeicher geht verloren, wenn der Agent gestoppt wird.

Der neue optionale Parameter persistent_dir gibt ein Verzeichnis an, in dem dieser Zustand des log[], log.count[], logrt[] oder logrt.count[]-Datenpunkts in einer Datei gespeichert wird. Der Zustand des Protokolldatenpunkts wird nach einem Neustart des Zabbix Agent aus der persistenten Datei wiederhergestellt.

Der primäre Anwendungsfall ist die Überwachung einer Protokolldatei auf einem gespiegelten Dateisystem. Bis zu einem bestimmten Zeitpunkt wird die Protokolldatei auf beide Spiegel geschrieben. Danach werden die Spiegel getrennt. Auf der aktiven Kopie wächst die Protokolldatei weiter und erhält neue Einträge. Der Zabbix Agent analysiert sie und sendet die verarbeitete Protokollgröße und Änderungszeit an den Server. Auf der passiven Kopie bleibt die Protokolldatei unverändert und liegt deutlich hinter der aktiven Kopie zurück. Später werden das Betriebssystem und der Zabbix Agent von der passiven Kopie aus neu gestartet. Die verarbeitete Protokollgröße und Änderungszeit, die der Zabbix Agent vom Server erhält, sind möglicherweise für die Situation auf der passiven Kopie nicht gültig. Um die Überwachung der Protokolldatei an der Stelle fortzusetzen, an der der Agent zum Zeitpunkt der Trennung des Dateisystemspiegels aufgehört hat, stellt der Agent seinen Zustand aus der persistenten Datei wieder her.

Agent-Betrieb mit persistenter Datei

Beim Start weiß der Zabbix Agent nichts über persistente Dateien. Erst nachdem er von Zabbix Server (Proxy) eine Liste aktiver Prüfungen erhalten hat, erkennt der Agent, dass einige Log-Datenpunkte durch persistente Dateien in den angegebenen Verzeichnissen unterstützt werden sollen.

Während des Agent-Betriebs werden die persistenten Dateien zum Schreiben geöffnet (mit fopen(filename, ”w”)) und mit den neuesten Daten überschrieben. Die Wahrscheinlichkeit, dass Daten in der persistenten Datei verloren gehen, wenn das Überschreiben und ein Split des Dateisystem-Mirrors gleichzeitig auftreten, ist sehr gering; dafür ist keine besondere Behandlung vorgesehen. Das Schreiben in die persistente Datei wird NICHT durch eine erzwungene Synchronisierung auf das Speichermedium abgeschlossen (fsync() wird nicht aufgerufen).

Das Überschreiben mit den neuesten Daten erfolgt nach erfolgreicher Meldung des passenden Logdatei-Eintrags oder der Metadaten (verarbeitete Loggröße und Änderungszeit) an Zabbix Server. Das kann so oft wie bei jeder Datenpunkt-Prüfung geschehen, wenn sich die Logdatei weiter ändert.

Beim Herunterfahren des Agenten sind keine besonderen Aktionen erforderlich.

Nach dem Empfang einer Liste aktiver Prüfungen markiert der Agent veraltete persistente Dateien zur Entfernung.

Eine persistente Datei wird veraltet, wenn:

1. Der entsprechende Log-Datenpunkt nicht mehr überwacht wird.
2. Ein Log-Datenpunkt mit einem anderen persistent_dir-Speicherort als zuvor neu konfiguriert wird.

Das Entfernen erfolgt mit einer Verzögerung von 24 Stunden, da Logdateien im Zustand NOTSUPPORTED nicht in der Liste der aktiven Prüfungen enthalten sind, später jedoch wieder SUPPORTED werden können und ihre persistenten Dateien dann nützlich sind.

Wenn der Agent vor Ablauf von 24 Stunden gestoppt wird, werden die veralteten Dateien nicht gelöscht, da der Zabbix Agent keine Informationen mehr über ihren Speicherort vom Zabbix Server erhält.

Warning:
Wenn `persistent_dir` eines Log-Datenpunkts bei gestopptem Agenten wieder auf den alten `persistent_dir`-Speicherort zurückkonfiguriert wird, ohne dass die alte persistente Datei vom Benutzer gelöscht wird, wird der Agent-Zustand aus der alten persistenten Datei wiederhergestellt. Dies kann zu fehlenden Meldungen oder falschen Alarmen führen.

Benennung und Speicherort persistenter Dateien

Der Zabbix Agent unterscheidet aktive Prüfungen anhand ihrer Schlüssel. Beispielsweise sind `logrt[/home/zabbix/test.log]` und `logrt[/home/zabbix/test.log,]` unterschiedliche Datenpunkte. Wenn der Datenpunkt `logrt[/home/zabbix/test.log,,,10]` im Frontend in `logrt[/home/zabbix/test.log,,,20]` geändert wird, wird der Datenpunkt `logrt[/home/zabbix/test.log,,,10]` aus der Liste der aktiven Prüfungen des Agents gelöscht und der Datenpunkt `logrt[/home/zabbix/test.log,,,20]` erstellt (einige Attribute werden bei der Änderung im Frontend/Server übernommen, jedoch nicht im Agent).

Der Dateiname setzt sich aus der MD5-Prüfsumme des Datenpunktschlüssels zusammen, an die die Länge des Datenpunktschlüssels angehängt wird, um die Wahrscheinlichkeit von Kollisionen zu verringern. Beispielsweise wird der Status des Datenpunkts `logrt[/home/zabbix50/test.log,,,,,,,,/home/zabbix50/agent_private]` in der persistenten Datei c963ade4008054813bbc0a650bb8e09266 gespeichert.

Mehrere Datenpunkte für Protokolldateien können denselben Wert für `persistent_dir` verwenden.

`persistent_dir` wird unter Berücksichtigung der spezifischen Dateisystemlayouts, Einhängepunkte und Einhängeoptionen sowie der Konfiguration der Speicherspiegelung angegeben - die persistente Datei sollte sich auf demselben gespiegelten Dateisystem wie die überwachte Protokolldatei befinden.

Wenn das Verzeichnis `persistent_dir` nicht erstellt werden kann oder nicht vorhanden ist oder der Zabbix Agent nicht über ausreichende Berechtigungen zum Erstellen, Schreiben, Lesen oder Löschen von Dateien verfügt, wird der Datenpunkt für die Protokolldatei zu `NOTSUPPORTED`.

Wenn während des Agent-Betriebs die Zugriffsrechte auf Dateien im persistenten Speicher entfernt werden oder andere Fehler auftreten (z. B. ein voller Datenträger), werden die Fehler in die Agent-Protokolldatei geschrieben, aber der Datenpunkt wird nicht zu `NOTSUPPORTED`.

Last auf I/O

Die persistente Datei des Datenpunkts wird nach dem erfolgreichen Senden jedes Datenpakets (das die Daten des Datenpunkts enthält) an den Server aktualisiert. Zum Beispiel beträgt `BufferSize` standardmäßig 100. Wenn ein Log-Datenpunkt 70 passende Datensätze gefunden hat, werden die ersten 50 Datensätze in einem Paket gesendet, die persistente Datei wird aktualisiert, dann werden die verbleibenden 20 Datensätze in einem zweiten Paket gesendet (möglicherweise mit einer Verzögerung, wenn mehr Daten angesammelt werden), und die persistente Datei wird erneut aktualisiert.

Aktionen bei Kommunikationsausfall zwischen Agent und Server

Jede übereinstimmende Zeile aus einem `log[]`- und `logrt[]`-Datenpunkt sowie das Ergebnis jeder Prüfung eines `log.count[]`- und `logrt.count[]`-Datenpunkts benötigt einen freien Slot im dafür vorgesehenen 50%-Bereich im Sende-Puffer des Agenten. Die Pufferelemente werden regelmäßig an den Server (oder Proxy) gesendet, und die Puffer-Slots werden anschließend wieder frei.

Solange im vorgesehenen Log-Bereich des Sende-Puffers des Agenten freie Slots vorhanden sind und die Kommunikation zwischen Agent und Server (oder Proxy) ausfällt, werden die Ergebnisse der Log-Überwachung im Sende-Puffer gesammelt. Dies hilft, kurze Kommunikationsausfälle abzufangen.

Bei längeren Kommunikationsausfällen werden alle Log-Slots belegt, und die folgenden Aktionen werden ausgeführt:

- Prüfungen von `log[]`- und `logrt[]`-Datenpunkten werden gestoppt. Wenn die Kommunikation wiederhergestellt ist und freie Slots im Puffer verfügbar sind, werden die Prüfungen an der vorherigen Position fortgesetzt. Es gehen keine übereinstimmenden Zeilen verloren, sie werden lediglich später gemeldet.
- Prüfungen von `log.count[]`- und `logrt.count[]`-Datenpunkten werden gestoppt, wenn `maxdelay = 0` (Standard). Das Verhalten ist ähnlich wie bei den oben beschriebenen `log[]`- und `logrt[]`-Datenpunkten. Beachten Sie, dass dies die Ergebnisse von `log.count[]`- und `logrt.count[]` beeinflussen kann: Beispielsweise zählt eine Prüfung 100 übereinstimmende Zeilen in einer Protokolldatei, aber da keine freien Slots im Puffer vorhanden sind, wird die Prüfung gestoppt. Wenn die Kommunikation wiederhergestellt ist, zählt der Agent dieselben 100 übereinstimmenden Zeilen sowie zusätzlich 70 neue übereinstimmende Zeilen. Der Agent sendet nun count = 170, als wären sie in einer einzigen Prüfung gefunden worden.
- Prüfungen von `log.count[]`- und `logrt.count[]`-Datenpunkten mit `maxdelay > 0`: Wenn während der Prüfung kein "Sprung" aufgetreten ist, ist das Verhalten ähnlich wie oben beschrieben. Wenn ein "Sprung" über Zeilen der Protokolldatei stattgefunden hat, wird die Position nach dem "Sprung" beibehalten und das gezählte Ergebnis verworfen. So versucht der Agent, auch bei Kommunikationsausfällen mit einer wachsenden Protokolldatei Schritt zu halten.

Behandlung von Kompilierungs- und Laufzeitfehlern regulärer Ausdrücke

Wenn ein regulärer Ausdruck, der in einem `log[]`-, `logrt[]`-, `log.count[]`- oder `logrt.count[]`-Datenpunkt verwendet wird, von der PCRE- oder PCRE2-Bibliothek nicht kompiliert werden kann, wechselt der Datenpunkt mit einer Fehlermeldung in den Zustand NICHT UNTERSTÜTZT. Um die Überwachung des Log-Datenpunkts fortzusetzen, muss der reguläre Ausdruck korrigiert werden.

Wenn der reguläre Ausdruck erfolgreich kompiliert wird, aber zur Laufzeit fehlschlägt (bei einigen oder allen Log-Datensätzen), bleibt der Log-Datenpunkt unterstützt und die Überwachung wird fortgesetzt. Der Laufzeitfehler wird in der Logdatei des Zabbix Agent protokolliert (ohne den Logdatei-Datensatz).

Die Protokollierungsrate ist auf einen Laufzeitfehler pro Prüfung begrenzt, damit der Zabbix Agent seine eigene Logdatei überwachen kann. Wenn beispielsweise 10 Datensätze analysiert werden und bei 3 Datensätzen ein Laufzeitfehler des regulären Ausdrucks auftritt, wird ein Datensatz im Agent-Log erzeugt.

Ausnahme: Wenn `MaxLinesPerSecond=1` und `update interval=1` festgelegt sind (sodass pro Prüfung nur ein Datensatz analysiert werden kann), werden Laufzeitfehler regulärer Ausdrücke nicht protokolliert.

`zabbix_agentd` protokolliert bei einem Laufzeitfehler den Schlüssel des Datenpunkts, während `zabbix_agent2` die ID des Datenpunkts protokolliert, um die Identifizierung des Log-Datenpunkts mit Laufzeitfehlern zu erleichtern. Bei Laufzeitfehlern wird empfohlen, den regulären Ausdruck neu zu entwerfen.

Einfache Prüfung

Übersicht

Einfache Prüfungen werden normalerweise für entfernte Agent-lose Prüfungen von Diensten verwendet.

Beachten Sie, dass der Zabbix Agent für einfache Prüfungen nicht erforderlich ist. Der Zabbix Server/Proxy ist für die Verarbeitung einfacher Prüfungen verantwortlich (Herstellen externer Verbindungen usw.).

Beispiele für die Verwendung einfacher Prüfungen:

```
net.tcp.service[ftp,,155]
net.tcp.service[http]
net.tcp.service.perf[http,,8080]
net.udp.service.perf[ntp]
```

Note:
Die Felder *Benutzername* und *Passwort* (auf 255 Zeichen begrenzt) in der Konfiguration von Datenpunkten für einfache Prüfungen werden für VMware-Überwachungsdatenpunkte verwendet; andernfalls werden sie ignoriert.

Unterstützte Prüfungen

Die Datenpunkt-Schlüssel werden ohne optionale Parameter und zusätzliche Informationen aufgeführt. Klicken Sie auf den Datenpunkt-Schlüssel, um die vollständigen Details anzuzeigen.

Siehe auch VMware-Monitoring-Datenpunkt-Schlüssel.

Datenpunkt-Schlüssel	Beschreibung
icmpping	Die Erreichbarkeit des Hosts per ICMP-Ping.
icmppingloss	Der Prozentsatz der verlorenen Pakete.
icmppingretry	Die Erreichbarkeit des Hosts per ICMP-Ping mit Wiederholungsversuchen.
icmppingsec	Die Antwortzeit des ICMP-Pings.
net.tcp.service	Prüft, ob ein Dienst ausgeführt wird und TCP-Verbindungen annimmt.
net.tcp.service.perf	Prüft die Leistung eines TCP-Dienstes.
net.udp.service	Prüft, ob ein Dienst ausgeführt wird und auf UDP-Anfragen antwortet.
net.udp.service.perf	Prüft die Leistung eines UDP-Dienstes.

Details zum Datenpunkt-Schlüssel

Parameter ohne spitze Klammern sind obligatorisch. Parameter, die mit spitzen Klammern **< >** gekennzeichnet sind, sind optional.

icmpping[<target>,<packets>,<interval>,<size>,<timeout>,<options>]

 Die Erreichbarkeit des Hosts per ICMP-Ping.
 Rückgabewert: *0* - ICMP-Ping fehlgeschlagen; *1* - ICMP-Ping erfolgreich.

Parameter:

- **target** - die IP-Adresse oder der DNS-Name des Hosts;

- **packets** - die Anzahl der Pakete;
- **interval** - die Zeit zwischen aufeinanderfolgenden Paketen in Millisekunden;
- **size** - die Paketgröße in Bytes;
- **timeout** - das Zeitlimit in Millisekunden;
- **options** - wird verwendet, um Weiterleitungen zuzulassen: Wenn leer (Standardwert), werden umgeleitete Antworten als Host ausgefallen behandelt; wenn auf *allow_redirect* gesetzt, werden umgeleitete Antworten als Host erreichbar behandelt.

Siehe auch die Tabelle der Standardwerte.

Kommentar:

- Wenn mindestens eines der durch den Parameter `packets` definierten Pakete zurückgegeben wird, gibt der Datenpunkt 1 zurück. Wenn keines der Pakete zurückgegeben wird, gibt der Datenpunkt 0 zurück.

Beispiel:

```
icmpping[,4] #Wenn mindestens eines der vier Pakete zurückgegeben wird, gibt der Datenpunkt 1 zurück.
```

icmppingloss[<target>,<packets>,<interval>,<size>,<timeout>,<options>]

 Der Prozentsatz verlorener Pakete.
 Rückgabewert: *Float*.

Parameter:

- **target** - die IP-Adresse oder der DNS-Name des Hosts;
- **packets** - die Anzahl der Pakete;
- **interval** - die Zeit zwischen aufeinanderfolgenden Paketen in Millisekunden;
- **size** - die Paketgröße in Byte;
- **timeout** - das Timeout in Millisekunden;
- **options** - wird verwendet, um Umleitungen zuzulassen: wenn leer (Standardwert), werden umgeleitete Antworten so behandelt, als wäre der Ziel-Host nicht erreichbar; wenn auf *allow_redirect* gesetzt, werden umgeleitete Antworten so behandelt, als wäre der Ziel-Host erreichbar.

Siehe auch die Tabelle der Standardwerte.

icmppingretry[<target>,<retries>,<backoff>,<size>,<timeout>,<options>]

 Die Erreichbarkeit des Hosts per ICMP-Ping mit Wiederholungsversuchen. Wenn das erste Paket erfolgreich ist, wird gestoppt; wenn das Paket fehlschlägt, wird der Versuch wiederholt, bis die im Parameter `retries` definierte maximale Anzahl von Versuchen erreicht ist. Dieser Datenpunkt ist nützlich, um die Anzahl der über das Netzwerk gesendeten Pakete zu reduzieren.
 Rückgabewert: *0* - ICMP-Ping fehlgeschlagen; *1* - ICMP-Ping erfolgreich.

Parameter:

- **target** - die IP-Adresse oder der DNS-Name des Hosts;
- **retries** - die Anzahl der Versuche, ein Ziel per Ping zu erreichen, ohne den ersten Versuch mit einzubeziehen (0 oder größer; Standardwert 1);
- **backoff** - der Wert, mit dem die Wartezeit bei jeder nachfolgenden Anfrage multipliziert wird (Bereich 1.0-5.0; Standardwert 1.0);
- **size** - die Paketgröße in Byte;
- **timeout** - das Timeout in Millisekunden;
- **options** - wird verwendet, um Umleitungen zuzulassen: wenn leer (Standardwert), werden umgeleitete Antworten so behandelt, als sei der Ziel-Host nicht erreichbar; wenn auf *allow_redirect* gesetzt, werden umgeleitete Antworten so behandelt, als sei der Ziel-Host erreichbar.

Siehe auch die Tabelle der Standardwerte.

icmppingsec[<target>,<packets>,<interval>,<size>,<timeout>,<mode>,<options>]

 Die Antwortzeit des ICMP-Pings (in Sekunden).
 Rückgabewert: *Float*.

Parameter:

- **target** - die IP-Adresse oder der DNS-Name des Hosts;
- **packets** - die Anzahl der Pakete;
- **interval** - die Zeit zwischen aufeinanderfolgenden Paketen in Millisekunden;
- **size** - die Paketgröße in Byte;
- **timeout** - das Timeout in Millisekunden;
- **mode** - mögliche Werte: *min*, *max* oder *avg* (Standard);
- **options** - wird verwendet, um Umleitungen zuzulassen: wenn leer (Standardwert), werden umgeleitete Antworten so behandelt, als wäre der Ziel-Host nicht erreichbar; wenn auf *allow_redirect* gesetzt, werden umgeleitete Antworten so behandelt, als wäre der Ziel-Host erreichbar.

Kommentare:

- Pakete, die verloren gehen oder bei denen ein Timeout auftritt, werden bei der Berechnung nicht berücksichtigt;
- Wenn der Host nicht verfügbar ist (Timeout erreicht), gibt der Datenpunkt 0 zurück;
- Wenn der Rückgabewert kleiner als 0.0001 Sekunden ist, wird der Wert auf 0.0001 Sekunden gesetzt;
- Siehe auch die Tabelle der Standardwerte.

net.tcp.service[service,<ip>,<port>]

 Prüft, ob ein Dienst läuft und TCP-Verbindungen akzeptiert.
 Rückgabewert: *0* - der Dienst ist nicht verfügbar; *1* - der Dienst läuft.

Parameter:

- **service** - mögliche Werte: *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https*, *telnet* (siehe Details);
- **ip** - die IP-Adresse oder der DNS-Name (standardmäßig wird die Host-IP/der Host-DNS verwendet);
- **port** - die Portnummer (standardmäßig wird die Standard-Portnummer des Dienstes verwendet).

Kommentare:

- Beachten Sie, dass beim Dienst *tcp* die Angabe des Ports obligatorisch ist;
- Diese Prüfungen können zu zusätzlichen Meldungen in den System-Daemon-Logdateien führen (SMTP- und SSH-Sitzungen werden in der Regel protokolliert);
- Die Prüfung verschlüsselter Protokolle (wie IMAP auf Port 993 oder POP auf Port 995) wird derzeit nicht unterstützt. Als Behelfslösung verwenden Sie bitte für solche Prüfungen `net.tcp.service[tcp,<ip>,port]`.

Beispiel:

```
net.tcp.service[ftp,,45] #Dieser Datenpunkt kann verwendet werden, um die Verfügbarkeit des FTP-Servers au
```

Attention:
Wenn SELinux im Enforcing-Modus läuft, können benutzerdefinierte einfache TCP/UDP-Prüfungen durch die Richtlinie blockiert werden. Um die neue ausgehende Verbindung zu überprüfen und zuzulassen, kontrollieren Sie die Audit-Ablehnungen mit `grep denied /var/log/audit/audit.log`

net.tcp.service.perf[service,<ip>,<port>]

 Prüft die Performance eines TCP-Dienstes.
 Rückgabewert: *Float*: *0.000000* - der Dienst ist nicht verfügbar; *Sekunden* - die Anzahl der Sekunden, die für die Verbindung mit dem Dienst benötigt wurden.

Parameter:

- **service** - mögliche Werte: *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https*, *telnet* (siehe Details);
- **ip** - die IP-Adresse oder der DNS-Name (standardmäßig wird die IP/DNS des Hosts verwendet);
- **port** - die Portnummer (standardmäßig wird die Standard-Portnummer des Dienstes verwendet).

Kommentare:

- Beachten Sie, dass beim Dienst *tcp* die Angabe des Ports zwingend erforderlich ist;
- Die Prüfung verschlüsselter Protokolle (wie IMAP auf Port 993 oder POP auf Port 995) wird derzeit nicht unterstützt. Als Workaround verwenden Sie bitte `net.tcp.service[tcp,<ip>,port]` für solche Prüfungen.

Beispiel:

```
net.tcp.service.perf[ssh] #Dieser Datenpunkt kann verwendet werden, um die Geschwindigkeit der ersten Antw
```

net.udp.service[service,<ip>,<port>]

 Prüft, ob ein Dienst läuft und auf UDP-Anfragen antwortet.
 Rückgabewert: *0* - der Dienst ist nicht verfügbar; *1* - der Dienst läuft.

Parameter:

- **service** - mögliche Werte: *ntp* (siehe Details);
- **ip** - die IP-Adresse oder der DNS-Name (standardmäßig wird die IP/DNS des Hosts verwendet);
- **port** - die Portnummer (standardmäßig wird die Standard-Portnummer des Dienstes verwendet).

Beispiel:

```
net.udp.service[ntp,,45] #Dieser Datenpunkt kann verwendet werden, um die Verfügbarkeit des NTP-Dienstes a
```

net.udp.service.perf[service,<ip>,<port>]

 Prüft die Performance eines UDP-Dienstes.
 Rückgabewert: *Float*: *0.000000* - der Dienst ist nicht verfügbar; *Sekunden* - die Anzahl der Sekunden, die auf eine Antwort des Dienstes gewartet wurde.

Parameter:

- **service** - mögliche Werte: *ntp* (siehe Details);
- **ip** - die IP-Adresse oder der DNS-Name (standardmäßig wird die Host-IP/der Host-DNS verwendet);
- **port** - die Portnummer (standardmäßig wird die Standard-Portnummer des Dienstes verwendet).

Beispiel:

```
net.udp.service.perf[ntp] #Dieser Datenpunkt kann verwendet werden, um die Antwortzeit des NTP-Dienstes zu
```

Attention:
Für die Unterstützung von SourceIP bei einfachen LDAP-Prüfungen (z. B. `net.tcp.service[ldap]`) ist OpenLDAP Version 2.6.1 oder höher erforderlich.

Timeout-Verarbeitung

Flexible Datenpunkt-Timeouts werden zwar für einfache Prüfungen unterstützt, gelten jedoch nicht für `icmpping*`- und VMware-Datenpunkte. Siehe Unterstützung für flexible Timeouts.

ICMP-Pings

Zabbix verwendet das externe Dienstprogramm **fping**, um ICMP-Pings (**icmpping**, **icmppingloss**, **icmppingretry**, **icmppingsec**) zu verarbeiten.

Installation

fping ist nicht in Zabbix enthalten und muss separat installiert werden:

- Verschiedene Unix-basierte Plattformen haben das Paket fping in ihren Standard-Repositories, es ist jedoch nicht vorinstalliert. In diesem Fall können Sie den Paketmanager verwenden, um fping zu installieren.
- Zabbix stellt fping-Pakete für RHEL und seine Derivate bereit. Bitte beachten Sie, dass diese Pakete ohne offiziellen Support bereitgestellt werden.
- fping kann auch aus dem Quellcode kompiliert werden.

Konfiguration

Geben Sie den Speicherort von fping im Parameter *FpingLocation* der Zabbix Server/Proxy-Konfigurationsdatei an (oder den Parameter *Fping6Location* für die Verwendung von IPv6-Adressen).

fping sollte für den Benutzer ausführbar sein, unter dem der Zabbix Server/Proxy ausgeführt wird, und dieser Benutzer sollte über ausreichende Rechte verfügen.

Siehe auch: Bekannte Probleme bei der Verarbeitung einfacher Prüfungen mit fping-Versionen unter 3.10.

Standardwerte

Standardwerte, Grenzwerte und Beschreibung der Werte für ICMP-Prüfparameter:

Parameter	Einheit	Beschreibung	Fping-Flag	Standardwerte festgelegt durch		Zulässige Grenzwerte durch Zabbix	
				fping	**Zabbix**	**min**	**max**
packets	Anzahl	Anzahl der an ein Ziel gesendeten Anfragepakete	-C		3	1	10000
interval	Millisekunden	Wartezeit zwischen aufeinanderfolgenden Paketen an ein einzelnes Ziel	-p	1000		20	unbegrenzt
size	Bytes	Paketgröße in Bytes 56 Bytes auf x86, 68 Bytes auf x86_64	-b	56 oder 68		24	65507

Parameter	Einheit	Beschreibung	Fping-Flag	Standardwerte festgelegt durch		Zulässige Grenzwerte durch Zabbix	
timeout	Millisekunden	**fping v3.x** – Zeitüberschreitung, die nach dem Senden des letzten Pakets abgewartet wird; beeinflusst durch das Flag `-C` **fping v4.x** – individuelle Zeitüberschreitung für jedes Paket	-t	**fping v3.x** – 500 **fping v4.x** und neuer – vom Flag `-p` übernommen, jedoch nicht mehr als 2000		50	unbegrenzt
retries	Anzahl	Anzahl der Versuche, ein Ziel anzupingen, ohne den ersten Versuch mitzuzählen	-r	3	1	0	unbegrenzt
backoff factor	Anzahl	Faktor, mit dem die Wartezeit bei jeder nachfolgenden Anfrage multipliziert wird	-B	1.5	1.0	1.0	5.0

Die Standardwerte können je nach Plattform und Version geringfügig abweichen.

Zusätzlich verwendet Zabbix die fping-Optionen `-i interval ms` (nicht mit dem im obigen Abschnitt genannten Datenpunktparameter `interval` verwechseln, der der fping-Option `-p` entspricht) und `-S source IP address` (oder `-I` in älteren fping-Versionen). Diese Optionen werden automatisch erkannt, indem Prüfungen mit verschiedenen Optionskombinationen ausgeführt werden. Zabbix versucht, den kleinsten von fping für die Option `-i` zugelassenen Wert in Millisekunden zu ermitteln, indem drei Werte getestet werden: 0, 1 und 10. Der erste erfolgreiche Wert wird anschließend für weitere ICMP-Prüfungen verwendet. Dieser Vorgang wird von jedem ICMP-Pinger-Prozess einzeln durchgeführt.

Automatisch erkannte fping-Optionen werden stündlich ungültig gemacht und beim nächsten Versuch, eine ICMP-Prüfung durchzuführen, erneut erkannt. Setzen Sie DebugLevel>=4, um Details zu diesem Vorgang in der Protokolldatei des Servers oder Proxys anzuzeigen.

Zabbix schreibt die von beliebigen `icmpping*`-Schlüsseln zu prüfenden IP-Adressen in eine temporäre Datei, die anschließend an fping übergeben wird. Wenn Datenpunkte unterschiedliche Schlüsselparameter haben, werden nur diejenigen mit identischen Schlüsselparametern in eine gemeinsame Datei geschrieben. Alle in die gemeinsame Datei geschriebenen IP-Adressen werden von fping parallel geprüft. Daher benötigt der Zabbix-ICMP-Pinger-Prozess unabhängig von der Anzahl der IP-Adressen in der Datei eine feste Zeit.

VMware-Überwachungs-Datenpunkt-Schlüssel

Die Liste der VMware-Monitoring-Datenpunktschlüssel wurde in den Abschnitt VMware-Monitoring verschoben.

SNMP-Agent

Übersicht

Möglicherweise möchten Sie SNMP-Monitoring auf Geräten wie Druckern, Netzwerk-Switches, Routern oder USV-Anlagen verwenden, die in der Regel SNMP-fähig sind und auf denen es unpraktisch wäre, komplette Betriebssysteme und Zabbix Agenten einzurichten.

Damit die von SNMP-Agenten auf diesen Geräten bereitgestellten Daten abgerufen werden können, muss der Zabbix Server zunächst mit SNMP-Unterstützung konfiguriert werden, indem das Flag `--with-net-snmp` angegeben wird. Es wird empfohlen, zusätzlich MIB-Dateien zu installieren, um sicherzustellen, dass die Werte der Datenpunkte im richtigen Format angezeigt werden. Ohne die MIB-Dateien können Formatierungsprobleme auftreten, z. B. die Anzeige von Werten in HEX statt UTF-8 oder umgekehrt.

SNMP-Prüfungen werden ausschließlich über das UDP-Protokoll durchgeführt.

Zabbix Server- und Proxy-Daemons protokollieren Zeilen ähnlich der folgenden, wenn sie eine fehlerhafte SNMP-Antwort erhalten:

```
SNMP response from host "gateway" does not contain all of the requested variable bindings
```

Auch wenn sie nicht alle problematischen Fälle abdecken, sind sie nützlich, um einzelne SNMP-Geräte zu identifizieren, für die kombinierte Anfragen deaktiviert werden sollten.

Zabbix Server/Proxy versucht bei SNMP-`walk`- und `get`-Datenpunkten bis zu 5 Mal erneut. Der Wiederholungsmechanismus gilt nicht für DNS-Auflösungsfehler.

Bei älteren SNMP-Prüfungen (eine einzelne OID-Nummer oder ein String) versucht Zabbix Server/Proxy nach einem erfolglosen Abfrageversuch mindestens einmal erneut: entweder über den Wiederholungsmechanismus der SNMP-Bibliothek oder über den internen Mechanismus der kombinierten Verarbeitung.

Warning:
Wenn Sie SNMPv3-Geräte überwachen, stellen Sie sicher, dass msgAuthoritativeEngineID (auch bekannt als snmpEngineID oder "Engine ID") niemals von zwei Geräten gemeinsam verwendet wird. Gemäß RFC 2571 (Abschnitt 3.1.1.1) muss sie für jedes Gerät eindeutig sein.

Warning:
RFC3414 verlangt, dass SNMPv3-Geräte ihre engineBoots persistent speichern. Einige Geräte tun dies nicht, was dazu führt, dass ihre SNMP-Nachrichten nach einem Neustart als veraltet verworfen werden. In einer solchen Situation muss der SNMP-Cache auf einem Server/Proxy manuell geleert werden (durch Verwendung von -R snmp_cache_reload) oder der Server/Proxy muss neu gestartet werden.

Note:
Zabbix speichert SNMPv3 EngineID → IP-Zuordnungen im Cache und verwendet zwischengespeicherte EngineIDs für nachfolgende Prüfungen erneut, anstatt jedes Mal einen Probeversuch zu senden, wodurch der Netzwerkverkehr reduziert wird. Wenn eine EngineID nicht erneut verwendet werden kann, wird ein erneuter Versuch mit einem Probeversuch durchgeführt, um die neue EngineID zu ermitteln.

Konfiguration der SNMP-Überwachung

Um die Überwachung eines Geräts über SNMP zu starten, müssen die folgenden Schritte durchgeführt werden:

Schritt 1

Ermitteln Sie die SNMP-Zeichenfolge (oder OID) des Datenpunkts, den Sie überwachen möchten.

Um eine Liste von SNMP-Zeichenfolgen zu erhalten, verwenden Sie den Befehl `snmpwalk` (Teil der net-snmp Software, die Sie im Rahmen der Zabbix-Installation installiert haben sollten) oder ein entsprechendes Tool:

```
snmpwalk -v 2c -c public <host IP> .
```

Da 2c hier für die SNMP-Version steht, können Sie es auch durch 1 ersetzen, um auf dem Gerät SNMP Version 1 anzugeben.

Dies sollte Ihnen eine Liste von SNMP-Zeichenfolgen und deren letztem Wert liefern. Falls nicht, ist es möglich, dass die SNMP-„Community" von der Standard-Community `public` abweicht; in diesem Fall müssen Sie diese ermitteln.

Sie können dann die Liste durchgehen, bis Sie die Zeichenfolge finden, die Sie überwachen möchten. Wenn Sie z. B. die eingehenden Bytes an Ihrem Switch auf Port 3 überwachen möchten, würden Sie die Zeichenfolge `IF-MIB::ifHCInOctets.3` aus dieser Zeile verwenden:

```
IF-MIB::ifHCInOctets.3 = Counter64: 3409739121
```

Sie können nun den Befehl `snmpget` verwenden, um die numerische OID für `IF-MIB::ifHCInOctets.3` zu ermitteln:

```
snmpget -v 2c -c public -On <host IP> IF-MIB::ifHCInOctets.3
```

Beachten Sie, dass die letzte Zahl in der Zeichenfolge die Portnummer ist, die Sie überwachen möchten. Siehe auch: Dynamische Indizes.

Dies sollte Ihnen etwa Folgendes liefern:

```
.1.3.6.1.2.1.31.1.1.1.6.3 = Counter64: 3472126941
```

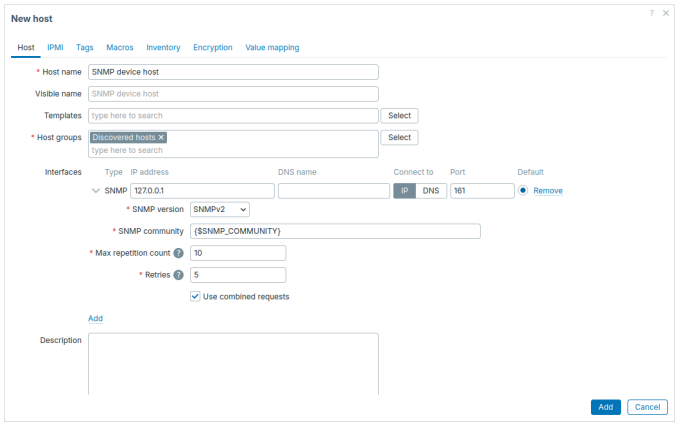
Auch hier ist die letzte Zahl in der OID die Portnummer.

Note:
Einige der am häufigsten verwendeten SNMP-OIDs werden von Zabbix automatisch in eine numerische Darstellung übersetzt.

Im letzten Beispiel oben ist der Werttyp ”Counter64”, was intern dem ASN_COUNTER64-Typ entspricht. Die vollständige Liste der unterstützten Typen ist 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 und ASN_OBJECT_ID. Diese Typen entsprechen ungefähr ”Counter32”, ”Counter64”, ”UInteger32”, ”INTEGER”, ”Float”, ”Double”, ”Timeticks”, ”Gauge32”, ”IpAddress”, ”OCTET STRING”, ”OBJECT IDENTIFIER” in der Ausgabe von **snmpget**, können jedoch je nach Vorhandensein eines Display-Hinweises auch als ”STRING”, ”Hex-STRING”, ”OID” und andere angezeigt werden.

Schritt 2

Erstellen Sie einen Host, der einem Gerät entspricht.



Fügen Sie dem Host eine SNMP-Schnittstelle hinzu:

1. Geben Sie die IP-Adresse bzw. den DNS-Namen und die Portnummer ein.
2. Wählen Sie die *SNMP-Version* aus der Dropdown-Liste aus.
3. Fügen Sie je nach ausgewählter SNMP-Version die Schnittstellenanmeldedaten hinzu:
    - SNMPv1, v2 erfordern nur die Community (in der Regel ‘public’).
    - SNMPv3 erfordert spezifischere Optionen; siehe SNMPv3.
4. Geben Sie den Maximalwert für Wiederholungen (Standard: 10) für native SNMP-Bulk-Anfragen (GetBulkRequest-PDUs) an; nur für `discovery[]`- und `walk[]`-Datenpunkte in SNMPv2 und v3. Beachten Sie, dass ein zu hoher Wert dazu führen kann, dass das Timeout der SNMP-Agent-Prüfung ausgelöst wird.
5. Aktivieren Sie das Kontrollkästchen *Kombinierte Anfragen verwenden*, um die kombinierte Verarbeitung von SNMP-Anfragen zu ermöglichen (nicht im Zusammenhang mit nativen SNMP-Bulk-Anfragen ”walk” und ”get”).

Sie können eine der bereitgestellten SNMP-Vorlagen verwenden, die automatisch eine Reihe von Datenpunkten hinzufügen. Bevor Sie eine Vorlage verwenden, prüfen Sie, ob sie mit dem Host kompatibel ist.

Klicken Sie auf *Hinzufügen*, um den Host zu speichern.

SNMPv3

Die folgenden Parameter sind für SNMPv3 erforderlich:

- *Context name* - Geben Sie den Context-Namen ein, um den Datenpunkt im SNMP-Subnetz zu identifizieren.
Benutzer-Makros werden in diesem Feld aufgelöst.
- *Security name* - Geben Sie den Security-Namen ein.
Benutzer-Makros werden in diesem Feld aufgelöst.
- *Security level* - Wählen Sie den Security-Level:
    - *noAuthNoPriv* - es werden weder Authentifizierungs- noch Datenschutzprotokolle verwendet;
    - *AuthNoPriv* - das Authentifizierungsprotokoll wird verwendet, das Datenschutzprotokoll nicht;
    - *AuthPriv* - sowohl Authentifizierungs- als auch Datenschutzprotokolle werden verwendet.
- *Authentication protocol* - Wählen Sie das Authentifizierungsprotokoll: *MD5*, *SHA1*; mit net-snmp 5.8 und neuer auch *SHA224*, *SHA256*, *SHA384* oder *SHA512*.
- *Authentication passphrase* - Geben Sie die Authentifizierungs-Passphrase ein.
Benutzer-Makros werden in diesem Feld aufgelöst.
- *Privacy protocol* - Wählen Sie das Datenschutzprotokoll: *DES*, *AES128*, *AES192*, *AES256*, *AES192C* (Cisco) oder *AES256C* (Cisco).
Siehe Hinweise zur Unterstützung von Datenschutzprotokollen.
- *Privacy passphrase* - Geben Sie die Datenschutz-Passphrase ein.
Benutzer-Makros werden in diesem Feld aufgelöst.

Bei falschen SNMPv3-Zugangsdaten (Security name, Authentication protocol/Passphrase, Privacy protocol):

- Zabbix erhält von net-snmp einen ERROR, außer bei einer falschen *Privacy passphrase*; in diesem Fall erhält Zabbix von net-snmp einen TIMEOUT-Fehler.
- Die Verfügbarkeit der SNMP-Schnittstelle wechselt zu rot (nicht verfügbar).

Note:
Änderungen an *Authentication protocol*, *Authentication passphrase*, *Privacy protocol* oder *Privacy passphrase*, die ohne Änderung des *Security name* vorgenommen werden, werden normalerweise automatisch angewendet, wenn die entsprechende SNMPv3-Schnittstelle in Zabbix aktualisiert wird. Wenn auch der *Security name* geändert wird, werden alle Parameter sofort aktualisiert.

Unterstützung für Privacy-Protokolle

Abhängig von Ihrem Betriebssystem und der net-snmp-Konfiguration sind einige Privacy-Protokolle möglicherweise nicht verfügbar:

- Auf einigen neueren Betriebssystemen (zum Beispiel RHEL9) wird die Unterstützung für DES für das Paket `net-snmp` entfernt.
- Verschlüsselungsprotokolle AES192 und stärker werden auf Betriebssystemen, die älter sind als RHEL 8, CentOS 8, Oracle Linux 8, Debian 12, Ubuntu LTS 22.04, openSUSE Leap 15.5, nicht standardmäßig unterstützt.

Um zu prüfen, ob die net-snmp-Bibliothek AES192+ unterstützt, verwenden Sie eine der folgenden Optionen:

1. `net-snmp-config`:

```
net-snmp-config --configure-options
```

Wenn die Ausgabe `--enable-blumenthal-aes` enthält, wird AES192+ unterstützt.

Beachten Sie, dass `net-snmp-config` Teil des Entwicklungspakets für SNMP ist (`libsnmp-dev` für Debian/Ubuntu, `net-snmp-devel` für CentOS/RHEL/OL/SUSE) und möglicherweise nicht standardmäßig installiert ist.

2. `snmpget`:

```
snmpget -v 3 -x AES-256
```

Wenn die Ausgabe `Invalid privacy protocol specified after -3x flag: AES-256` enthält, wird AES192+ nicht unterstützt. Wenn die Ausgabe `No hostname specified.` enthält, wird AES192+ nicht unterstützt.

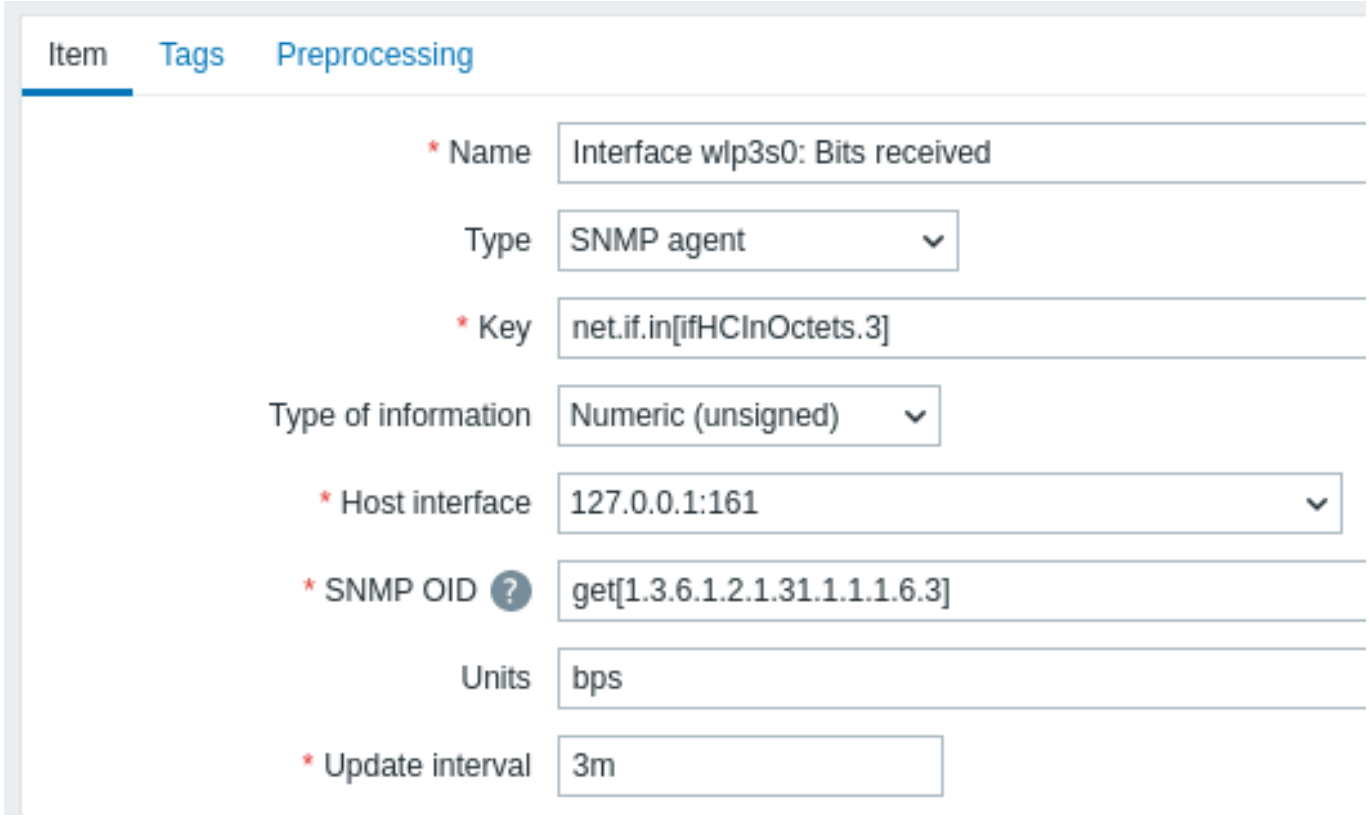
Wenn Ihre net-snmp-Bibliothek AES192 und höhere Protokolle nicht unterstützt, kompilieren Sie net-snmp mit der Option `--enable-blumenthal-aes` neu und kompilieren Sie anschließend den Zabbix-Server mit der Option `--with-net-snmp=/home/user/yo` neu.

Schritt 3

Erstellen Sie einen Datenpunkt zur Überwachung.

Kehren Sie nun zu Zabbix zurück und klicken Sie beim zuvor erstellten SNMP-Host auf *Datenpunkte*. Je nachdem, ob Sie beim Erstellen Ihres Hosts eine Vorlage verwendet haben oder nicht, sehen Sie entweder eine Liste der mit Ihrem Host verknüpften SNMP-Datenpunkte oder eine leere Liste. Wir gehen davon aus, dass Sie den Datenpunkt selbst anhand der Informationen erstellen, die Sie gerade mit snmpwalk und snmpget gesammelt haben. Klicken Sie daher auf *Datenpunkt erstellen*.

Füllen Sie die erforderlichen Parameter im Formular für den neuen Datenpunkt aus:



Parameter	Beschreibung
Name	Geben Sie den Namen des Datenpunkts ein.
Typ	Wählen Sie hier **SNMP agent** aus.
Schlüssel	Geben Sie einen aussagekräftigen Schlüssel ein.
Host-Schnittstelle	Stellen Sie sicher, dass Sie die SNMP-Schnittstelle auswählen, z. B. die Ihres Switches/Routers.

Parameter	Beschreibung
SNMP-OID	Verwenden Sie eines der unterstützten Formate, um OID-Werte einzugeben: **walk[OID1,OID2,...]** – ruft einen Teilbaum von Werten ab. Beispiel: `walk[1.3.6.1.2.1.2.2.1.2,1.3.6.1.2.1.2.2.1.3]`. Diese Option verwendet native SNMP-Bulk-Anfragen (GetBulkRequest-PDUs) asynchron. Die Timeout-Einstellungen für diesen Datenpunkt können im Formular Datenpunktkonfiguration festgelegt werden. Erwägen Sie, einen niedrigen Timeout-Wert festzulegen, um lange Verzögerungen zu vermeiden, wenn das Gerät nicht erreichbar ist, da bis zu fünf Wiederholungsversuche durchgeführt werden, wenn vorherige Versuche wegen eines Timeouts oder Fehlers fehlschlagen (ein Timeout von beispielsweise 3 Sekunden kann zu einer Wartezeit von 15 Sekunden führen). Sie können diesen Datenpunkt als Master-Datenpunkt verwenden und abhängige Datenpunkte erstellen, die mithilfe der Vorverarbeitung Daten aus dem Master-Datenpunkt extrahieren. Es ist möglich, mehrere OIDs in einem einzelnen SNMP-Walk anzugeben, z. B. `walk[OID1,OID2,...]`, um jeweils eine OID asynchron zu verarbeiten. Wenn die Bulk-Anfrage keine Ergebnisse liefert, wird versucht, einen einzelnen Datensatz ohne Bulk-Anfrage abzurufen. MIB-Namen werden als Parameter unterstützt. Daher liefern `walk[1.3.6.1.2.1.2.2.1.2]` und `walk[ifDescr]` dieselbe Ausgabe. Wenn mehrere OIDs/MIBs angegeben werden, z. B. `walk[ifDescr,ifType,ifPhysAddress]`, wird die Ausgabe als verkettete Liste zurückgegeben. GetBulk-Anfragen werden mit SNMPv2- und SNMPv3-Schnittstellen verwendet, GetNext mit SNMPv1-Schnittstellen. Die maximale Anzahl von Wiederholungen für Bulk-Anfragen wird auf Schnittstellenebene konfiguriert. Der Parameter für die maximale Anzahl von Wiederholungen beeinflusst Bulk-Anfragen, indem er die maximale Anzahl der OIDs festlegt, die in einer einzelnen Bulk-Antwort zurückgegeben werden. Ein höherer Wert führt zu größeren Bulk-Antworten und reduziert die Anzahl der erforderlichen Übertragungen. Allerdings unterstützen möglicherweise nicht alle Geräte sehr hohe Werte, was zu Problemen führen kann. Dieser Datenpunkt gibt die Ausgabe des Dienstprogramms snmpwalk mit den Parametern -Oe -Ot -On zurück. Sie können diesen Datenpunkt als Master-Datenpunkt in der SNMP-Erkennung verwenden. **get[OID]** – ruft einen *einzelnen* Wert asynchron ab. Beispiel: `get[1.3.6.1.2.1.31.1.1.1.6.3]` Die Timeout-Einstellungen für diesen Datenpunkt können im Formular Datenpunktkonfiguration festgelegt werden. Erwägen Sie, einen niedrigen Timeout-Wert festzulegen, um lange Verzögerungen zu vermeiden, wenn das Gerät nicht erreichbar ist, da bis zu fünf Wiederholungsversuche durchgeführt werden, wenn vorherige Versuche wegen eines Timeouts oder Fehlers fehlschlagen (ein Timeout von beispielsweise 3 Sekunden kann zu einer Wartezeit von 15 Sekunden führen). **OID** – (veraltet) geben Sie eine einzelne textuelle oder numerische OID ein, um synchron einen einzelnen Wert abzurufen, optional kombiniert mit anderen Werten. Beispiel: `1.3.6.1.2.1.31.1.1.1.6.3`. Bei dieser Option entspricht das Timeout der Datenpunktprüfung dem im Server-Konfigurationsdatei festgelegten Wert. Es wird **empfohlen**, für eine bessere Leistung Datenpunkte mit `walk[OID]` und `get[OID]` zu verwenden. Alle Datenpunkte mit `walk[OID]` und `get[OID]` werden asynchron ausgeführt – es ist nicht erforderlich, die Antwort auf eine Anfrage abzuwarten, bevor andere Prüfungen gestartet werden. Auch die DNS-Auflösung erfolgt asynchron. Die maximale Parallelität asynchroner Prüfungen beträgt 1000 (definiert durch MaxConcurrentChecksPerPoller). Die Anzahl der asynchronen SNMP-Poller wird durch den Parameter StartSNMPPollers festgelegt. Beachten Sie, dass für Netzwerkverkehrsstatistiken, die von einer der Methoden zurückgegeben werden, im Tab *Vorverarbeitung* ein Schritt *Änderung pro Sekunde* hinzugefügt werden muss. Andernfalls erhalten Sie den kumulativen Wert vom SNMP-Gerät anstelle der letzten Änderung.

Alle Pflichtfelder sind mit einem roten Sternchen gekennzeichnet.

Speichern Sie nun den Datenpunkt und gehen Sie zu *Überwachung* > *Neueste Daten*, um Ihre SNMP-Daten anzuzeigen.

Beispiel 1

Allgemeines Beispiel:

Parameter	Beschreibung
OID	1.2.3.45.6.7.8.0 (oder .1.2.3.45.6.7.8.0)
Key	<Eindeutige Zeichenfolge, die als Referenz für Auslöser verwendet wird> Zum Beispiel „my_param“.

Beachten Sie, dass OID entweder in numerischer oder in Zeichenfolgenform angegeben werden kann. In einigen Fällen muss eine Zeichenfolgen-OID jedoch in eine numerische Darstellung umgewandelt werden. Das Dienstprogramm snmpget kann zu diesem Zweck verwendet werden:

```
snmpget -On localhost public enterprises.ucdavis.memory.memTotalSwap.0
```

Beispiel 2

Überwachung der Uptime:

Parameter	Beschreibung
OID	MIB::sysUpTime.0
Key	router.uptime
Value type	Float
Units	uptime
Preprocessing step: Custom multiplier	0.01

Native SNMP-Bulk-Anfragen

Der **walk[OID1,OID2,...]**-Datenpunkt ermöglicht die Verwendung nativer SNMP-Funktionen für Bulk-Anfragen (GetBulkRequest-PDUs), die in den SNMP-Versionen 2/3 verfügbar sind.

Eine GetBulk-Anfrage in SNMP führt mehrere GetNext-Anfragen aus und gibt das Ergebnis in einer einzigen Antwort zurück.
Dies kann sowohl für reguläre SNMP-Datenpunkte als auch für die SNMP-Erkennung verwendet werden, um Netzwerk-Roundtrips zu minimieren.

Der SNMP-Datenpunkt **walk[OID1,OID2,...]** kann als Master-Datenpunkt verwendet werden, der Daten in einer Anfrage erfasst. Abhängige Datenpunkte können die Antwort anschließend mithilfe der Vorverarbeitung nach Bedarf analysieren.

Beachten Sie, dass die Verwendung nativer SNMP-Bulk-Anfragen nicht mit der Option zum Kombinieren von SNMP-Anfragen zusammenhängt. Dabei handelt es sich um eine eigene Methode von Zabbix zum Kombinieren mehrerer SNMP-Anfragen (siehe nächsten Abschnitt).

Für SNMP-Bulk-Datenpunkte werden bis zu fünf Wiederholungsversuche durchgeführt, um einen Fehler zu vermeiden, falls eines der Pakete verloren geht.
Das Timeout für SNMP-Datenpunkte mit `get` und `walk` (festgelegt im Formular zur Datenpunktkonfiguration) gilt für eine gesamte Sitzung. Das Timeout wird unabhängig davon angewendet, ob die Daten vollständig abgerufen wurden. Wenn Daten nur teilweise empfangen werden (beispielsweise wenn Daten erfolgreich nur für eine von mehreren OIDs erfasst wurden), wird der Datenpunkt mit der Meldung „Only partial data received“ nicht unterstützt.
Wenn das Timeout erreicht wird, erfolgt ein Wiederholungsversuch, das Timeout wird zurückgesetzt und die letzte Anfrage wird erneut gesendet. Dadurch kann die Sitzung mit der letzten Anfrage fortgesetzt werden, falls ein einzelnes Paket verloren geht oder zu spät eintrifft.
Erwägen Sie, einen niedrigen Timeout-Wert festzulegen, um lange Verzögerungen zu vermeiden, wenn das Gerät nicht erreichbar ist, da bis zu fünf Wiederholungsversuche durchgeführt werden, wenn vorherige Versuche wegen eines Timeouts oder aus anderen Gründen fehlschlagen (beispielsweise kann ein Timeout von 3 Sekunden zu einer Wartezeit von 15 Sekunden führen).

Interne Funktionsweise der kombinierten Verarbeitung

Der Zabbix Server und Proxy können SNMP-Geräte in einer einzigen Anfrage nach mehreren Werten abfragen. Dies betrifft mehrere Typen von SNMP-Datenpunkten:

- reguläre SNMP-Datenpunkte
- SNMP-Datenpunkte mit dynamischen Indizes
- SNMP-Low-Level-Discovery-Regeln

Alle SNMP-Datenpunkte auf einer einzelnen Schnittstelle mit identischen Parametern werden zur gleichen Zeit für die Abfrage eingeplant. Die ersten beiden Typen von Datenpunkten werden von Pollern in Stapeln von höchstens 128 Datenpunkten verarbeitet, während Low-Level-Discovery-Regeln wie bisher einzeln verarbeitet werden.

Auf der unteren Ebene gibt es zwei Arten von Vorgängen zum Abfragen von Werten: das Abrufen mehrerer angegebener Objekte und das Durchlaufen eines OID-Baums.

Für das „Abrufen“ wird eine GetRequest-PDU mit höchstens 128 Variablenbindungen verwendet. Für das „Durchlaufen“ wird für SNMPv1 eine GetNextRequest-PDU verwendet, während für SNMPv2 und SNMPv3 eine GetBulkRequest mit einem Feld „max-repetitions“ von höchstens 128 verwendet wird.

Daraus ergeben sich für die einzelnen SNMP-Datentypen folgende Vorteile der kombinierten Verarbeitung:

- reguläre SNMP-Datenpunkte profitieren von den Verbesserungen beim „Abrufen“;
- SNMP-Datenpunkte mit dynamischen Indizes profitieren sowohl von den Verbesserungen beim „Abrufen“ als auch beim „Durchlaufen“: „Abrufen“ wird zur Indexüberprüfung und „Durchlaufen“ zum Erstellen des Caches verwendet;
- SNMP-Low-Level-Discovery-Regeln profitieren von den Verbesserungen beim „Durchlaufen“.

Es gibt jedoch ein technisches Problem: Nicht alle Geräte können 128 Werte pro Anfrage zurückgeben. Einige geben immer eine ordnungsgemäße Antwort zurück, andere antworten entweder mit einem „tooBig(1)“-Fehler oder überhaupt nicht mehr, sobald die potenzielle Antwort eine bestimmte Größe überschreitet.

Um die optimale Anzahl von Objekten zu ermitteln, die für ein bestimmtes Gerät abgefragt werden sollen, verwendet Zabbix die folgende Strategie. Zunächst wird vorsichtig mit der Abfrage von 1 Wert pro Anfrage begonnen. Wenn dies erfolgreich ist, werden 2 Werte pro Anfrage abgefragt. Wenn dies erneut erfolgreich ist, werden 3 Werte pro Anfrage abgefragt. Anschließend wird die Anzahl der abgefragten Objekte jeweils mit 1,5 multipliziert. Daraus ergibt sich folgende Sequenz von Anfragegrößen: 1, 2, 3, 4, 6, 9, 13, 19, 28, 42, 63, 94, 128.

Sobald ein Gerät jedoch keine ordnungsgemäße Antwort mehr liefert, beispielsweise bei 42 Variablen, führt Zabbix zwei Aktionen aus.

Erstens halbiert Zabbix für den aktuellen Datenpunktstapel die Anzahl der Objekte in einer einzelnen Anfrage und fragt 21 Variablen ab. Wenn das Gerät erreichbar ist, sollte die Abfrage in der überwiegenden Mehrheit der Fälle funktionieren, da bekannt ist, dass 28 Variablen funktionieren, und 21 deutlich weniger sind. Wenn dies jedoch ebenfalls fehlschlägt, greift Zabbix auf die Abfrage einzelner Werte zurück. Wenn dies zu diesem Zeitpunkt weiterhin fehlschlägt, antwortet das Gerät definitiv nicht, und die Anfragegröße ist nicht die Ursache.

Zweitens beginnt Zabbix für nachfolgende Datenpunktstapel mit der letzten erfolgreichen Anzahl von Variablen, in unserem Beispiel 28, und erhöht die Anfragegröße anschließend jeweils um 1, bis das Limit erreicht ist. Angenommen, die größte Antwortgröße beträgt 32 Variablen, haben die nachfolgenden Anfragen die Größen 29, 30, 31, 32 und 33. Die letzte Anfrage schlägt fehl, und Zabbix wird nie wieder eine Anfrage der Größe 33 senden. Ab diesem Zeitpunkt fragt Zabbix für dieses Gerät höchstens 32 Variablen ab.

Wenn große Abfragen mit dieser Anzahl von Variablen fehlschlagen, kann dies eine von zwei Ursachen haben. Die genauen Kriterien, anhand derer ein Gerät die Antwortgröße begrenzt, sind nicht bekannt. Wir versuchen jedoch, dies anhand der Anzahl der Variablen anzunähern. Die erste Möglichkeit besteht darin, dass diese Anzahl von Variablen ungefähr dem tatsächlichen Antwortgrößenlimit des Geräts im Allgemeinen entspricht: Manchmal liegt die Antwort unterhalb des Limits, manchmal darüber. Die zweite Möglichkeit besteht darin, dass ein UDP-Paket in eine der beiden Richtungen einfach verloren gegangen ist. Aus diesen Gründen reduziert Zabbix bei einer fehlgeschlagenen Abfrage die maximale Anzahl der Variablen, um in den komfortablen Bereich des Geräts vorzudringen, jedoch höchstens zweimal.

Wenn im obigen Beispiel eine Abfrage mit 32 Variablen zufällig fehlschlägt, reduziert Zabbix die Anzahl auf 31. Wenn auch diese Abfrage fehlschlägt, reduziert Zabbix die Anzahl auf 30. Zabbix reduziert die Anzahl jedoch nicht unter 30, da davon ausgegangen wird, dass weitere Fehler durch verlorene UDP-Pakete und nicht durch das Limit des Geräts verursacht werden.

Wenn ein Gerät kombinierte Anfragen aus anderen Gründen nicht ordnungsgemäß verarbeiten kann und die oben beschriebene Heuristik nicht funktioniert, gibt es für jede Schnittstelle die Einstellung „Use combined requests“, mit der Sie kombinierte Anfragen für dieses Gerät deaktivieren können.

Wenn kombinierte Anfragen zu unvollständigen oder fehlerhaften Antworten führen, die falsche Berechnungen pro Sekunde (Delta) verursachen, beispielsweise scheinbare Spitzen in Schnittstellenzählern, deaktivieren Sie *Use combined requests* für die betroffene Schnittstelle, um separate Abfragen pro Datenpunkt zu erzwingen. Dadurch werden falsche Spitzen häufig verhindert. Alternativ können Sie asynchrone `get[]`- oder `walk[]`-Datenpunkte verwenden. Diese werden asynchron ausgeführt und unterliegen nicht der stapelweisen Verarbeitung durch *Use combined requests* pro Schnittstelle. Sie können anstelle älterer synchroner OID-Prüfungen verwendet werden, um Probleme im Zusammenhang mit kombinierten Anfragen zu vermeiden. Suchen Sie in den Server-/Proxy-Protokolleinträgen nach ähnlichen Einträgen wie dem im Abschnitt Übersicht gezeigten, um betroffene Geräte zu identifizieren.

Wenn die Schnittstelle außerdem häufig nicht verfügbar wird, kann es erforderlich sein, den Parameter `UnavailableDelay` in den Konfigurationsdateien des Zabbix Servers oder des Zabbix Proxys zu erhöhen, um die Häufigkeit der Anfragen zu reduzieren.

Datenpunkte können nicht unterstützt werden, wenn während der Erkennung oder beim Durchlaufen von OIDs unvollständige Daten empfangen werden.

Dynamische Indizes

Übersicht

Auch wenn Sie die erforderliche Indexnummer (zum Beispiel die einer Netzwerkschnittstelle) unter den SNMP-OIDs finden können, können Sie sich manchmal nicht vollständig darauf verlassen, dass die Indexnummer immer gleich bleibt.

Indexnummern können dynamisch sein – sie können sich im Laufe der Zeit ändern, wodurch Ihr Datenpunkt möglicherweise nicht mehr funktioniert.

Um dieses Szenario zu vermeiden, kann eine OID definiert werden, die die Möglichkeit berücksichtigt, dass sich eine Indexnummer ändert.

Wenn Sie beispielsweise den Indexwert abrufen müssen, der an **ifInOctets** angehängt werden soll und der der Schnittstelle **GigabitEthernet0/1** auf einem Cisco-Gerät entspricht, verwenden Sie die folgende OID:

```
ifInOctets["index","ifDescr","GigabitEthernet0/1"]
```

Die Syntax

Es wird eine spezielle Syntax für OID verwendet:

<OID der Daten>[”index”,”<Basis-OID des Index>”,”<zu suchende Zeichenfolge>”]

Parameter	Beschreibung
OID der Daten	Haupt-OID, die zum Abrufen von Daten für den Datenpunkt verwendet wird.
index	Verarbeitungsmethode. Derzeit wird eine Methode unterstützt: **index** - nach dem Index suchen und ihn an die Daten-OID anhängen
Basis-OID des Index	Diese OID wird nachgeschlagen, um den Indexwert zu erhalten, der der Zeichenfolge entspricht.
zu suchende Zeichenfolge	Die Zeichenfolge, die für eine exakte Übereinstimmung mit einem Wert bei der Suche verwendet wird. Groß-/Kleinschreibung wird beachtet.

Beispiel

Abrufen der Speicherauslastung des *apache*-Prozesses.

Bei Verwendung dieser OID-Syntax:

```
HOST-RESOURCES-MIB::hrSWRunPerfMem["index","HOST-RESOURCES-MIB::hrSWRunPath", "/usr/sbin/apache2"]
```

wird die Indexnummer hier nachgeschlagen:

```
...
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"
...
```

Nun haben wir den Index 5388. Der Index wird an die Daten-OID angehängt, um den Wert zu erhalten, an dem wir interessiert sind:

```
HOST-RESOURCES-MIB::hrSWRunPerfMem.5388 = INTEGER: 31468 KBytes
```

Zwischenspeicherung der Indexsuche

Wenn ein Datenpunkt mit dynamischem Index angefordert wird, ruft Zabbix die gesamte SNMP-Tabelle unter der Basis-OID für den Index ab und speichert sie im Cache, auch wenn eine Übereinstimmung früher gefunden würde. Dies geschieht für den Fall, dass sich später ein anderer Datenpunkt auf dieselbe Basis-OID bezieht - Zabbix sucht den Index dann im Cache nach, anstatt den überwachten Host erneut abzufragen. Beachten Sie, dass jeder Poller-Prozess einen separaten Cache verwendet.

Bei allen nachfolgenden Wertermittlungen wird nur der gefundene Index überprüft. Wenn er sich nicht geändert hat, wird der Wert angefordert. Wenn er sich geändert hat, wird der Cache neu aufgebaut - jeder Poller, der auf einen geänderten Index trifft, durchläuft die SNMP-Indextabelle erneut.

Speciale OIDs

Einige der am häufigsten verwendeten SNMP-OIDs werden von Zabbix automatisch in eine numerische Darstellung übersetzt. Zum Beispiel wird **ifIndex** in **1.3.6.1.2.1.2.2.1.1** übersetzt, **ifIndex.0** wird in **1.3.6.1.2.1.2.2.1.1.0** übersetzt.

Die Tabelle enthält eine Liste der speziellen OIDs.

Special OID	Identifier	Description
ifIndex	1.3.6.1.2.1.2.2.1.1	Ein eindeutiger Wert für jede Schnittstelle.
ifDescr	1.3.6.1.2.1.2.2.1.2	Eine Textzeichenfolge mit Informationen über die Schnittstelle. Diese Zeichenfolge sollte den Namen des Herstellers, den Produktnamen und die Version der Hardware-Schnittstelle enthalten.
ifType	1.3.6.1.2.1.2.2.1.3	Der Typ der Schnittstelle, unterschieden nach den physischen/Link-Protokollen, die sich unmittelbar 'unterhalb' der Netzwerkschicht im Protokollstapel befinden.
ifMtu	1.3.6.1.2.1.2.2.1.4	Die Größe des größten Datagramms, das auf der Schnittstelle gesendet/empfangen werden kann, angegeben in Oktetten.
ifSpeed	1.3.6.1.2.1.2.2.1.5	Eine Schätzung der aktuellen Bandbreite der Schnittstelle in Bits pro Sekunde.
ifPhysAddress	1.3.6.1.2.1.2.2.1.6	Die Adresse der Schnittstelle auf der Protokollschicht unmittelbar 'unterhalb' der Netzwerkschicht im Protokollstapel.
ifAdminStatus	1.3.6.1.2.1.2.2.1.7	Der aktuelle administrative Status der Schnittstelle.
ifOperStatus	1.3.6.1.2.1.2.2.1.8	Der aktuelle Betriebsstatus der Schnittstelle.
ifInOctets	1.3.6.1.2.1.2.2.1.10	Die Gesamtzahl der auf der Schnittstelle empfangenen Oktette, einschließlich Framing-Zeichen.
ifInUcastPkts	1.3.6.1.2.1.2.2.1.11	Die Anzahl der an ein höheres Protokoll gelieferten Unicast-Pakete des Subnetzes.
ifInNUcastPkts	1.3.6.1.2.1.2.2.1.12	Die Anzahl der Nicht-Unicast-Pakete (d. h. Subnetz-Broadcast- oder Subnetz-Multicast-Pakete), die an ein höheres Protokoll geliefert wurden.
ifInDiscards	1.3.6.1.2.1.2.2.1.13	Die Anzahl der eingehenden Pakete, die verworfen wurden, obwohl keine Fehler erkannt wurden, um zu verhindern, dass sie an ein höheres Protokoll weitergeleitet werden. Ein möglicher Grund für das Verwerfen eines solchen Pakets könnte das Freigeben von Pufferplatz sein.
ifInErrors	1.3.6.1.2.1.2.2.1.14	Die Anzahl der eingehenden Pakete, die Fehler enthielten, wodurch sie nicht an ein höheres Protokoll weitergeleitet werden konnten.
ifInUnknownProtos	1.3.6.1.2.1.2.2.1.15	Die Anzahl der über die Schnittstelle empfangenen Pakete, die aufgrund eines unbekannten oder nicht unterstützten Protokolls verworfen wurden.
ifOutOctets	1.3.6.1.2.1.2.2.1.16	Die Gesamtzahl der von der Schnittstelle gesendeten Oktette, einschließlich Framing-Zeichen.
ifOutUcastPkts	1.3.6.1.2.1.2.2.1.17	Die Gesamtzahl der Pakete, deren Übertragung von höherstufigen Protokollen angefordert wurde und die auf dieser Unterschicht nicht an eine Multicast- oder Broadcast-Adresse gerichtet waren, einschließlich derer, die verworfen oder nicht gesendet wurden.
ifOutNUcastPkts	1.3.6.1.2.1.2.2.1.18	Die Gesamtzahl der Pakete, deren Übertragung von höherstufigen Protokollen angefordert wurde und die auf dieser Unterschicht an eine Multicast- oder Broadcast-Adresse gerichtet waren, einschließlich derer, die verworfen oder nicht gesendet wurden.
ifOutDiscards	1.3.6.1.2.1.2.2.1.19	Die Anzahl der ausgehenden Pakete, die verworfen wurden, obwohl keine Fehler erkannt wurden, um zu verhindern, dass sie übertragen werden. Ein möglicher Grund für das Verwerfen eines solchen Pakets könnte das Freigeben von Pufferplatz sein.
ifOutErrors	1.3.6.1.2.1.2.2.1.20	Die Anzahl der ausgehenden Pakete, die aufgrund von Fehlern nicht übertragen werden konnten.
ifOutQLen	1.3.6.1.2.1.2.2.1.21	Die Länge der Ausgabepaketwarteschlange (in Paketen).

MIB-Dateien

Einführung

MIB steht für Management Information Base. MIB-Dateien ermöglichen die Verwendung textueller Darstellungen von OIDs (Objektbezeichnern). Bei der Überwachung von SNMP-Geräten mit Zabbix können Sie rohe OIDs verwenden. Wenn Sie jedoch lieber die textuelle Darstellung verwenden, müssen Sie MIB-Dateien installieren.

Zum Beispiel ist

`ifHCOutOctets`

die textuelle Darstellung der OID

`1.3.6.1.2.1.31.1.1.1.10`

Installieren von MIB-Dateien

Auf Debian-basierten Systemen:

```
apt install snmp-mibs-downloader
download-mibs
```

Auf RedHat-basierten Systemen:

```
dnf install net-snmp-libs
```

MIB-Dateien aktivieren

Auf RedHat-basierten Systemen sollten MIB-Dateien standardmäßig aktiviert sein. Auf Debian-basierten Systemen müssen Sie die Datei `/etc/snmp/snmp.conf` bearbeiten und die Zeile `mibs :` auskommentieren:

```
# Da die snmp-Pakete aus Lizenzgründen ohne MIB-Dateien ausgeliefert werden, ist das Laden
# von MIBs standardmäßig deaktiviert. Wenn Sie die MIBs hinzugefügt haben, können Sie das
# Laden wieder aktivieren, indem Sie die folgende Zeile auskommentieren.
mibs :
```

Testen von MIB-Dateien

Das Testen von SNMP-MIBs kann mit dem Dienstprogramm `snmpwalk` durchgeführt werden. Wenn es nicht installiert ist, verwenden Sie die folgenden Anweisungen.

Auf Debian-basierten Systemen:

```
apt install snmp
```

Auf RedHat-basierten Systemen:

```
dnf install net-snmp-utils
```

Danach darf der folgende Befehl bei der Abfrage eines Netzwerkgeräts keinen Fehler ausgeben:

```
$ 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
[...]
```

Verwendung von MIBs in Zabbix

Das Wichtigste, das Sie beachten sollten, ist, dass Zabbix-Prozesse nicht über Änderungen an MIB-Dateien informiert werden. Daher müssen Sie nach jeder Änderung den Zabbix Server oder Proxy neu starten, z. B.:

```
systemctl restart zabbix-server
```

Danach werden die an den MIB-Dateien vorgenommenen Änderungen wirksam.

Benutzerdefinierte MIB-Dateien verwenden

Es gibt Standard-MIB-Dateien, die mit jeder GNU/Linux-Distribution mitgeliefert werden. Einige Gerätehersteller stellen jedoch ihre eigenen bereit.

Angenommen, Sie möchten die MIB-Datei CISCO-SMI verwenden. Mit den folgenden Anweisungen wird sie heruntergeladen und installiert:

```
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
```

Nun sollten Sie sie verwenden können. Versuchen Sie, den Namen des Objekts *ciscoProducts* aus der MIB-Datei in eine OID zu übersetzen:

```
snmptranslate -IR -On CISCO-SMI::ciscoProducts
.1.3.6.1.4.1.9.1
```

Wenn Sie statt der OID Fehler erhalten, stellen Sie sicher, dass alle vorherigen Befehle keine Fehler zurückgegeben haben.

Die Übersetzung des Objektnamens hat funktioniert, Sie können nun die benutzerdefinierte MIB-Datei verwenden. Beachten Sie das in der Abfrage verwendete MIB-Namenspräfix (*CISCO-SMI::*). Sie werden dieses sowohl bei der Verwendung von Kommandozeilenwerkzeugen als auch in Zabbix benötigen.

Vergessen Sie nicht, den Zabbix Server/Proxy neu zu starten, bevor Sie diese MIB-Datei in Zabbix verwenden.

Attention:
Beachten Sie, dass MIB-Dateien Abhängigkeiten haben können. Das heißt, eine MIB kann eine andere erfordern. Um diese Abhängigkeiten zu erfüllen, müssen Sie alle betroffenen MIB- Dateien installieren.

SNMP-Trap

Übersicht

Das Empfangen von SNMP-Traps ist das Gegenstück zum Abfragen von SNMP-fähigen Geräten.

In diesem Fall werden die Informationen von einem SNMP-fähigen Gerät an `snmptrapd` gesendet und von Zabbix Server oder Zabbix Proxy aus einer Datei gesammelt oder "gefangen".

Üblicherweise werden Traps bei einer Zustandsänderung gesendet, und der Agent verbindet sich auf Port 162 mit dem Server (im Gegensatz zu Port 161 auf der Agent-Seite, der für Abfragen verwendet wird). Die Verwendung von Traps kann einige kurzzeitige Probleme erkennen, die zwischen den Abfrageintervallen auftreten und von den Abfragedaten möglicherweise übersehen werden.

Das Empfangen von SNMP-Traps in Zabbix ist für die Zusammenarbeit mit `snmptrapd` und einem der Mechanismen zum Weiterleiten der Traps an Zabbix ausgelegt - entweder ein Bash- oder Perl-Skript oder SNMPTT.

Note:
Die einfachste Möglichkeit, die Trap-Überwachung nach der Konfiguration von Zabbix einzurichten, ist die Bash-Skript-Lösung, da Perl und SNMPTT in modernen Distributionen oft fehlen und eine komplexere Konfiguration erfordern. Diese Lösung verwendet jedoch ein Skript, das als `traphandle` konfiguriert ist. Für eine bessere Leistung in Produktionssystemen verwenden Sie die eingebettete Perl-Lösung (entweder ein Skript mit der Option `do perl` oder SNMPTT).

Der Ablauf beim Empfangen eines Traps:

1. `snmptrapd` empfängt einen Trap.
2. `snmptrapd` leitet den Trap an das Empfangsskript (Bash, Perl) oder SNMPTT weiter.
3. Der Empfänger parst, formatiert und schreibt den Trap in eine Datei.
4. Der Zabbix SNMP-Trapper liest und parst die Trap-Datei.
5. Für jeden Trap findet Zabbix alle SNMP-Trapper-Datenpunkte mit Host-Schnittstellen, die zur empfangenen Trap-Adresse passen. Beachten Sie, dass beim Abgleich nur die ausgewählte *IP* oder *DNS* in der Host-Schnittstelle verwendet wird.
6. Für jeden gefundenen Datenpunkt wird der Trap mit dem regulären Ausdruck in `snmptrap[regexp]` verglichen. Der Trap wird als Wert **aller** übereinstimmenden Datenpunkte gesetzt. Wenn kein passender Datenpunkt gefunden wird und ein `snmptrap.fallback`-Datenpunkt vorhanden ist, wird der Trap als dessen Wert gesetzt.
7. Wenn der Trap nicht als Wert eines Datenpunkts gesetzt wurde, protokolliert Zabbix standardmäßig den nicht zugeordneten Trap. (Dies wird über *Nicht zugeordnete SNMP-Traps protokollieren* unter *Administration* > *Allgemein* > *Sonstiges* konfiguriert.)

Hinweise zum HA-Failover

Während des Wechsels eines Hochverfügbarkeits-(HA-)Knotens setzt Zabbix die Verarbeitung nach dem letzten Datensatz innerhalb des letzten ISO-8601-Zeitstempels fort; wird derselbe Datensatz nicht gefunden, wird nur der Zeitstempel verwendet, um die letzte Position zu bestimmen.

Konfigurieren von SNMP-Traps

Dieser Datentyp erfordert die folgende Frontend-Konfiguration:

Schritt 1: Eine SNMP-Schnittstelle für den Host erstellen

1. Gehen Sie in *Datenerfassung* > *Hosts* und erstellen Sie einen Host oder bearbeiten Sie den vorhandenen.
2. Wählen Sie im Feld *Schnittstellen* den Schnittstellentyp *SNMP* aus.
3. Geben Sie die IP-Adresse/DNS-Namen und die Portnummer ein.
4. Wählen Sie die *SNMP-Version* aus der Dropdown-Liste: (*SNMPv1*, *SNMPv2* oder *SNMPv3*) und fügen Sie die Schnittstellenanmeldedaten entsprechend der ausgewählten SNMP-Version hinzu.

Die Adresse jedes empfangenen Traps wird mit den IP- und DNS-Adressen aller SNMP-Schnittstellen verglichen, um die entsprechenden Hosts zu finden.

Schritt 2: Den Datenpunkt konfigurieren

1. Erstellen Sie für den Host einen Datenpunkt oder bearbeiten Sie den vorhandenen.
2. Verwenden Sie im Feld *Key* einen der SNMP-Trap-Keys:
   - `snmptrap[regexp]` - erfasst alle SNMP-Traps, die dem im Parameter `regexp` angegebenen regulären Ausdruck entsprechen; wenn `regexp` nicht angegeben ist, wird jeder Trap erfasst.
 Benutzer-Makros und globale reguläre Ausdrücke werden im Parameter unterstützt.
 Rückgabewert: SNMP-Trap.
 Dieser Datenpunkt kann nur für SNMP-Schnittstellen festgelegt werden.
   - `snmptrap.fallback` - erfasst alle SNMP-Traps, die von keinem der `snmptrap[]`-Datenpunkte für diese Schnittstelle erfasst wurden.
 Rückgabewert: SNMP-Trap.
 Dieser Datenpunkt kann nur für SNMP-Schnittstellen festgelegt werden.

> **Note:**
> Die Übereinstimmung mit regulären Ausdrücken über mehrere Zeilen wird derzeit nicht unterstützt.

3. Setzen Sie *Typ der Information* auf *Log*, damit Zeitstempel geparst werden. Andere Formate wie *Numerisch* sind ebenfalls zulässig, erfordern jedoch möglicherweise einen benutzerdefinierten Trap-Handler.

Einrichten der SNMP-Trap-Überwachung

Konfigurieren von Zabbix Server/Proxy

Um die Traps zu lesen, muss der Zabbix Server oder Proxy so konfiguriert werden, dass der SNMP-Trapper-Prozess gestartet wird und auf die Trap-Datei verwiesen wird, die von SNMPTT oder einem Bash/Perl-Trap-Receiver geschrieben wird. Dazu bearbeiten Sie die Konfigurationsdatei (`zabbix_server.conf` oder `zabbix_proxy.conf`):

```
StartSNMPTrapper=1
SNMPTrapperFile=[TRAP FILE]
```

> **Warning:**
> Wenn der systemd-Parameter `PrivateTmp` verwendet wird, funktioniert diese Datei in `/tmp` wahrscheinlich nicht.

Konfigurieren des Bash-Trap-Receivers

Anforderungen: nur snmptrapd.

Ein Bash-Trap-Receiver-Skript kann verwendet werden, um Traps von snmptrapd über die Trapper-Datei an den Zabbix Server zu übergeben. Um es zu konfigurieren, fügen Sie die Option `traphandle` zur Konfigurationsdatei von snmptrapd (`snmptrapd.conf`) hinzu, siehe Beispiel.

> **Note:**
> snmptrapd muss möglicherweise neu gestartet werden, damit Änderungen an seiner Konfiguration übernommen werden.

Konfiguration des Perl-Trap-Receivers

Anforderungen: Perl, Net-SNMP kompiliert mit `--enable-embedded-perl` (seit Net-SNMP 5.4 standardmäßig aktiviert)

Ein Perl-Trap-Receiver (siehe `misc/snmptrap/zabbix_trap_receiver.pl`) kann verwendet werden, um Traps direkt von snmptrapd an den Server zu übergeben. Zur Konfiguration:

- Fügen Sie das Perl-Skript zur Konfigurationsdatei von snmptrapd (`snmptrapd.conf`) hinzu, z. B.:

```
perl do "[FULL PATH TO PERL RECEIVER SCRIPT]";
```

- Konfigurieren Sie den Receiver, z. B.:

```
$SNMPTrapperFile = '[TRAP FILE]';
$DateTimeFormat = '[DATE TIME FORMAT]';
```

Note:
snmptrapd muss möglicherweise neu gestartet werden, damit Änderungen an seiner Konfiguration übernommen werden.

Note:
Wenn der Skriptname nicht in Anführungszeichen gesetzt ist, verweigert snmptrapd den Start mit Meldungen, die diesen ähneln:


```
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
```

SNMPTT konfigurieren

Zunächst sollte `snmptrapd` so konfiguriert werden, dass es SNMPTT verwendet.

Note:
Für die beste Leistung sollte SNMPTT als Daemon mit `snmptthandler-embedded` konfiguriert werden, um die Traps an ihn weiterzuleiten. Siehe die Anweisungen zum Konfigurieren von SNMPTT.

Wenn SNMPTT so konfiguriert ist, dass es die Traps empfängt, konfigurieren Sie `snmptt.ini`:

1. Aktivieren Sie die Verwendung des Perl-Moduls aus dem `NET-SNMP`-Paket:

```
net_snmp_perl_enable = 1
```

2. Protokollieren Sie Traps in die Trap-Datei, die von Zabbix gelesen wird:

```
log_enable = 1
log_file = [TRAP FILE]
```

3. Legen Sie das Datums- und Zeitformat fest:

```
date_time_format = %Y-%m-%dT%H:%M:%S%z
```

Warning:
Das Paket `net-snmp-perl` wurde in RHEL 8.0-8.2 entfernt und in RHEL 8.3 wieder hinzugefügt. Weitere Informationen finden Sie unter bekannte Probleme.

Formatieren Sie nun die Traps so, dass Zabbix sie erkennt (bearbeiten Sie `snmptt.conf`):

1. Jede `FORMAT`-Anweisung sollte mit `ZBXTRAP [address]` beginnen, wobei `[address]` mit IP- und DNS-Adressen der SNMP-Schnittstellen auf Zabbix verglichen wird. Z. B.:

```
EVENT coldStart .1.3.6.1.6.3.1.1.5.1 "Status Events" Normal
FORMAT ZBXTRAP $aA Device reinitialized (coldStart)
```

2. Weitere Informationen zum SNMP-Trap-Format finden Sie unten.

Attention:
Verwenden Sie keine unbekannten Traps - Zabbix kann sie nicht erkennen. Unbekannte Traps können durch die Definition eines allgemeinen Ereignisses in `snmptt.conf` behandelt werden:


```
EVENT general .* "General event" Normal
```

SNMP-Trap-Format

Alle benutzerdefinierten Perl-Trap-Empfänger und die SNMPTT-Trap-Konfiguration müssen den Trap auf folgende Weise formatieren:

```
[timestamp] [the trap, part 1] ZBXTRAP [address] [the trap, part 2]
```

wobei

- `[timestamp]` - der Zeitstempel im Format `%Y-%m-%dT%H:%M:%S%z`.
- `ZBXTRAP` - Kopfzeile, die angibt, dass in dieser Zeile ein neuer Trap beginnt.
- `[address]` - IP-Adresse, die verwendet wird, um den Host für diesen Trap zu ermitteln.

Beachten Sie, dass `ZBXTRAP` und `[address]` während der Verarbeitung aus der Nachricht entfernt werden. Wenn der Trap anders formatiert ist, kann Zabbix die Traps möglicherweise unerwartet parsen.

Beispiel-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
```

Dies führt zu folgendem Trap für die SNMP-Schnittstelle mit 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
```

Systemanforderungen

Note:
Es wird empfohlen, MIB-Dateien zu installieren, um sicherzustellen, dass Datenpunktwerte im korrekten Format angezeigt werden. Ohne die MIB-Dateien können Formatierungsprobleme auftreten, z. B. die Anzeige von Werten in HEX statt in UTF-8 oder umgekehrt.

Unterstützung großer Dateien

Zabbix unterstützt große Dateien für SNMP-Trapper-Dateien. Die maximale Dateigröße, die Zabbix lesen kann, beträgt 2^63 (8 EiB). Beachten Sie, dass das Dateisystem möglicherweise eine niedrigere Grenze für die Dateigröße festlegt.

Log-Rotation

Zabbix stellt kein eigenes System zur Log-Rotation bereit - dies muss vom Benutzer übernommen werden. Die Log-Rotation sollte zuerst die alte Datei umbenennen und sie erst später löschen, damit keine Traps verloren gehen:

1. Zabbix öffnet die Trap-Datei am zuletzt bekannten Speicherort und geht zu Schritt 3.
2. Zabbix prüft, ob die aktuell geöffnete Datei rotiert wurde, indem die Inode-Nummer mit der Inode-Nummer der definierten Trap-Datei verglichen wird. Wenn keine geöffnete Datei vorhanden ist, setzt Zabbix den letzten Speicherort zurück und geht zu Schritt 1.
3. Zabbix liest die Daten aus der aktuell geöffneten Datei und setzt den neuen Speicherort.
4. Die neuen Daten werden geparst. Wenn dies die rotierte Datei war, wird die Datei geschlossen und es geht zurück zu Schritt 2.
5. Wenn keine neuen Daten vorhanden waren, schläft Zabbix 1 Sekunde und geht zurück zu Schritt 2.

Dateisystem

Aufgrund der Implementierung der Trap-Datei benötigt Zabbix ein Dateisystem mit Unterstützung für Inodes, um Dateien zu unterscheiden (die Informationen werden durch einen `stat()`-Aufruf ermittelt).

Einrichtungsbeispiele mit verschiedenen SNMP-Protokollversionen

Dieses Beispiel verwendet snmptrapd und ein Bash-Empfängerskript, um Traps an den Zabbix-Server weiterzuleiten.

Einrichtung:

1. Konfigurieren Sie Zabbix so, dass der SNMP-Trapper gestartet und die Trap-Datei festgelegt wird. Fügen Sie Folgendes zu `zabbix_server.conf` hinzu:

```
StartSNMPTrapper=1
SNMPTrapperFile=/var/lib/zabbix/snmptraps/snmptraps.log
```

2. Laden Sie das Bash-Skript nach `/usr/sbin/zabbix_trap_handler.sh` herunter:

```
curl -o /usr/sbin/zabbix_trap_handler.sh https://raw.githubusercontent.com/zabbix/zabbix-docker/trunk/temp
```

Passen Sie bei Bedarf die Variable `ZABBIX_TRAPS_FILE` im Skript an. Um den Standardwert zu verwenden, erstellen Sie zunächst das übergeordnete Verzeichnis:

```
mkdir -p /var/lib/zabbix/snmptraps
```

3. Fügen Sie Folgendes zu `snmtrapd.conf` hinzu (siehe das funktionierende Beispiel):

```
traphandle default /bin/bash /usr/sbin/zabbix_trap_handler.sh
```

Note:
snmptrapd muss möglicherweise neu gestartet werden, damit die Änderungen an der Konfiguration übernommen werden.

4. Erstellen Sie einen SNMP-Datenpunkt TEST (beachten Sie die anfänglichen Konfigurationsanforderungen):

   Typ: *SNMP trap*
 Informationstyp: *Log*
 Host-Schnittstelle: *SNMP 127.0.0.1*
 Schlüssel: `snmptrap["linkUp"]`
 Format der Log-Zeit: `yyyyMMdd.hhmmss`

Beachten Sie, dass das ISO-8601-Format für Datum und Uhrzeit verwendet wird.

5. Als Nächstes konfigurieren wir `snmptrapd` für die ausgewählte SNMP-Protokollversion und senden Test-Traps mit dem Dienstprogramm `snmptrap`.

SNMPv1, SNMPv2

Die Protokolle SNMPv1 und SNMPv2 verwenden zur Authentifizierung eine „Community-Zeichenfolge“. Im folgenden Beispiel verwenden wir „secret“ als Community-Zeichenfolge. Dieser Wert muss auf den SNMP-Trap-Sendern identisch festgelegt werden.

Beachten Sie, dass SNMPv2 zwar in Produktionsumgebungen weiterhin weit verbreitet ist, jedoch keine Verschlüsselung und keine echte Authentifizierung des Senders bietet. Die Daten werden im Klartext übertragen. Daher sollten diese Protokollversionen nur in sicheren Umgebungen wie privaten Netzwerken verwendet werden und niemals über öffentliche Netzwerke oder Netzwerke von Drittanbietern.

SNMP Version 1 wird heutzutage kaum noch verwendet, da sie keine 64-Bit-Zähler unterstützt und als veraltetes Protokoll gilt.

Um den Empfang von SNMPv1- oder SNMPv2-Traps zu aktivieren, fügen Sie die folgende Zeile zu `snmptrapd.conf` hinzu. Ersetzen Sie `secret` durch die auf den SNMP-Trap-Sendern konfigurierte SNMP-Community-Zeichenfolge:

```
authCommunity log,execute,net secret
```

Anschließend können wir mit `snmptrap` einen Test-Trap senden. In diesem Beispiel verwenden wir die verbreitete „link up“-OID:

```
snmptrap -v 2c -c secret localhost '' linkUp.0
```

SNMPv3

SNMPv3 behebt die Sicherheitsprobleme von SNMPv1/v2 und bietet Authentifizierung und Verschlüsselung. Sie können MD5 oder verschiedene SHA-Authentifizierungsmethoden sowie DES oder verschiedene AES-Varianten als Verschlüsselung verwenden.

Konfigurieren Sie `snmptrapd` für den Empfang von SNMPv3-Traps, indem Sie die folgenden Zeilen zu `snmptrapd.conf` hinzufügen:

```
##### Einen SNMPv3-USM-Benutzer mit Authentifizierung und Datenschutz erstellen
createUser -e <engineID> <securityName> SHA <authKey> AES <privKey>
##### Dem Benutzer erlauben, Traps zu protokollieren und Trap-Handler-Skripte auszuführen
authUser log,execute <securityName>
```

Beispiel:

```
createUser -e 0x8000000001020304 traptest SHA myAuthPassword AES myPrivPassword
authUser log,execute traptest
```

Attention:
Beachten Sie das Schlüsselwort „execute“, mit dem Sie Skripte für dieses Benutzersicherheitsmodell ausführen können.

Um die SNMPv3-Trap-Konfiguration zu testen, senden Sie mit dem folgenden Befehl einen Test-Trap:

```
snmptrap -v 3 -n "" -a SHA -A myAuthPassword -x AES -X myPrivPassword -l authPriv -u traptest -e 0x8000000
```

Warning:
Wenn Sie starke Verschlüsselungsmethoden wie AES192 oder AES256 verwenden möchten, verwenden Sie bitte net-snmp ab Version 5.8. Möglicherweise müssen Sie es mit der `configure`-Option `--enable-blumenthal-aes` neu kompilieren. Ältere Versionen von net-snmp unterstützen AES192/AES256 nicht. Siehe auch: Strong Authentication or Encryption.

Überprüfung

In beiden Beispielen sehen Sie ähnliche Zeilen in Ihrer Datei `/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
```

Der Wert des Datenpunkts in Zabbix lautet:

```
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
```

Beispiel mit 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
```

Siehe auch

- Zabbix-Blogartikel zu SNMP-Traps
- Konfiguration von snmptrapd (offizielle net-snmp-Dokumentation)
- Konfiguration von snmptrapd für den Empfang von SNMPv3-Benachrichtigungen (offizielle net-snmp-Dokumentation)

Zabbix intern

Übersicht

Mit internen Prüfungen können Sie die internen Prozesse von Zabbix überwachen. Mit anderen Worten: Sie können überwachen, was auf dem Zabbix Server oder Zabbix Proxy geschieht.

Interne Prüfungen werden berechnet:

- auf dem Zabbix Server, wenn der Host vom Server überwacht wird
- auf dem Zabbix Proxy, wenn der Host vom Proxy überwacht wird

Interne Prüfungen werden vom Server oder Proxy unabhängig vom Wartungsstatus des Hosts verarbeitet.

Um diesen Datenpunkt zu verwenden, wählen Sie den Datenpunkttyp **Zabbix intern** aus.

> **Note:**
> Interne Prüfungen werden von Zabbix Pollern verarbeitet.

Leistung

Die Verwendung einiger interner Datenpunkte kann sich negativ auf die Leistung auswirken. Diese Datenpunkte sind:

- `zabbix[host,,items]`
- `zabbix[host,,items_unsupported]`
- `zabbix[hosts]`
- `zabbix[items]`
- `zabbix[items_unsupported]`
- `zabbix[queue,,]`
- `zabbix[requiredperformance]`
- `zabbix[stats,,,queue,,]`
- `zabbix[triggers]`

Die Frontend-Bereiche Systeminformationen und Warteschlange sind ebenfalls betroffen.

Unterstützte Prüfungen

Die Datenpunktschlüssel werden ohne anpassbare Parameter und zusätzliche Informationen aufgeführt.
Klicken Sie auf den Datenpunktschlüssel, um die vollständigen Details anzuzeigen.

Datenpunktschlüssel	Beschreibung
zabbix[apm,]	Die Statistiken der vom APM-Collector empfangenen Telemetriedaten.
zabbix[boottime]	Die Startzeit des Zabbix-Server- oder Zabbix-Proxy-Prozesses in Sekunden.
zabbix[cluster,discovery,nodes]	Ermittelt die Knoten des Hochverfügbarkeitsclusters.
zabbix[connector_queue]	Die Anzahl der in die Connector-Warteschlange eingereihten Werte.

Datenpunktschlüssel	Beschreibung
zabbix[discovery_queue]	Die Anzahl der in die Erkennungswarteschlange eingereihten Netzwerkprüfungen.
zabbix[host,,items]	Die Anzahl der aktivierten Datenpunkte (unterstützt und nicht unterstützt) auf dem Host.
zabbix[host,,items_unsupported]	Die Anzahl der aktivierten nicht unterstützten Datenpunkte auf dem Host.
zabbix[host,maintenance]	Der aktuelle Wartungsstatus des Hosts.
zabbix[host,active_agent,available]	Die Verfügbarkeit aktiver Agent-Prüfungen auf dem Host.
zabbix[host,discovery,interfaces]	Die Details aller im Zabbix Frontend konfigurierten Schnittstellen des Hosts.
zabbix[host,,available]	Die Verfügbarkeit der Hauptschnittstelle für einen bestimmten Prüfungstyp auf dem Host.
zabbix[hosts]	Die Anzahl der überwachten Hosts.
zabbix[items]	Die Anzahl der aktivierten Datenpunkte (unterstützt und nicht unterstützt).
zabbix[items_unsupported]	Die Anzahl der nicht unterstützten Datenpunkte.
zabbix[java,,]	Die Informationen über das Zabbix Java gateway.
zabbix[lld_queue]	Die Anzahl der in die Warteschlange für die Verarbeitung der Low-Level-Erkennung eingereihten Werte.
zabbix[preprocessing]	Statistiken der vom Vorverarbeitungsmanager empfangenen Werte.
zabbix[preprocessing_queue]	Die Anzahl der in die Vorverarbeitungswarteschlange eingereihten Werte.
zabbix[process,,,]	Der Prozentsatz der Zeit, die ein bestimmter Zabbix-Prozess oder eine Gruppe von Prozessen (identifiziert durch <type> und <mode>) im <state> verbracht hat.
zabbix[proxy,,]	Die Informationen über den Zabbix Proxy.
zabbix[proxy,discovery]	Die Liste der Zabbix Proxys.
zabbix[proxy group,,available]	Die Anzahl der Online-Proxys in einer Proxygruppe.
zabbix[proxy group,,pavailable]	Der Prozentsatz der Online-Proxys in einer Proxygruppe.
zabbix[proxy group,,proxies]	Die Liste der Zabbix Proxys in einer Proxygruppe.
zabbix[proxy group,,state]	Der Status einer Proxygruppe.
zabbix[proxy group,discovery]	Gibt eine Liste von Proxygruppen mit Konfigurationsdaten und Echtzeitdaten zurück.
zabbix[proxy_buffer,buffer,]	Gibt die Statistiken zur Speichernutzung des Proxy-Speicherpuffers zurück.
zabbix[proxy_buffer,state,changes]	Gibt die Anzahl der Statusänderungen zwischen Festplatten- und Speicherpuffermodus seit dem Start zurück.
zabbix[proxy_buffer,state,current]	Gibt den aktuellen Betriebsstatus zurück, in dem die neuen Daten gespeichert werden.
zabbix[proxy_history]	Die Anzahl der Werte in der Proxy-Verlaufstabelle, die darauf warten, an den Server gesendet zu werden.
zabbix[queue,,]	Die Anzahl der überwachten Datenpunkte in der Warteschlange, deren Verarbeitung sich um mindestens <from> Sekunden, aber weniger als <to> Sekunden verzögert.
zabbix[rcache,,]	Die Verfügbarkeitsstatistiken des Zabbix-Konfigurationscaches.
zabbix[requiredperformance]	Die erforderliche Leistung des Zabbix-Servers oder Zabbix-Proxys, angegeben als erwartete Anzahl neuer Werte pro Sekunde.
zabbix[stats,,]	Gibt die internen Metriken eines Zabbix-Servers oder Proxys zurück. Wenn <ip> und <port> angegeben sind, werden die Metriken von der Remote-Instanz abgerufen, andernfalls von der lokalen Instanz.
zabbix[stats,,,queue,,]	Gibt die internen Warteschlangenmetriken eines Zabbix-Servers oder Proxys zurück. Wenn <ip> und <port> angegeben sind, werden die Metriken von der Remote-Instanz abgerufen, andernfalls von der lokalen Instanz.
zabbix[tcache,,]	Die Effektivitätsstatistiken des Zabbix-Trendfunktionscaches.
zabbix[triggers]	Die Anzahl der aktivierten Auslöser in der Zabbix-Datenbank, bei denen alle Datenpunkte auf aktivierten Hosts aktiviert sind.
zabbix[uptime]	Die Betriebszeit des Zabbix-Server- oder Zabbix-Proxy-Prozesses in Sekunden.
zabbix[vcache,buffer,]	Die Verfügbarkeitsstatistiken des Zabbix-Wertecaches.
zabbix[vcache,cache,]	Die Effektivitätsstatistiken des Zabbix-Wertecaches.
zabbix[version]	Die Version des Zabbix-Servers oder Proxys.
zabbix[vmware,buffer,]	Die Verfügbarkeitsstatistiken des Zabbix-VMware-Caches.
zabbix[vps,written]	Die Gesamtzahl der in die Datenbank geschriebenen Verlaufswerte.
zabbix[wcache,,]	Die Statistiken und Verfügbarkeit des Zabbix-Schreibcaches.

Details zum Datenpunktschlüssel

- Parameter ohne spitze Klammern sind obligatorisch und müssen *unverändert* verwendet werden (zum Beispiel „host“ und „available“ in `zabbix[host,<type>,available]`).

- Parameter mit spitzen Klammern **< >** müssen durch einen gültigen Wert ersetzt werden. Wenn ein Parameter einen Standardwert hat, kann er weggelassen werden.
- Werte für Datenpunkte und Datenpunktparameter mit der Kennzeichnung „not supported on proxy“ können nur abgerufen werden, wenn der Host vom Server überwacht wird. Umgekehrt können Werte mit der Kennzeichnung „not supported on server“ nur abgerufen werden, wenn der Host vom Proxy überwacht wird.

zabbix[apm,<type>]

 Die Statistiken der vom APM collector empfangenen Telemetriedaten:

- *requests* – die Anzahl der vom APM collector empfangenen Anfragen (Zähler):
  - *accepted* – die Anzahl der zur Verarbeitung angenommenen Anfragen;
  - *dropped* – die Anzahl der verworfenen Anfragen;
- *rows* – die Anzahl der in die Datenquelle geschriebenen Telemetriedatenzeilen, nach Datentyp (Zähler).

Rückgabewert: *JSON-Objekt*.

Parameter:

- **type** – *traces*, *metrics* oder *logs*. Wenn nicht angegeben, werden Statistiken für alle Typen zurückgegeben.

Kommentare:

- Dieser Datenpunkt wird auf dem Zabbix Server nicht unterstützt.

Beispiel für Rückgabewerte von `zabbix[apm]`:

```
{
  "requests": {
    "accepted": 15230,
    "dropped": 12
  },
  "rows": {
    "traces": 48710,
    "metrics_gauge": 31250,
    "metrics_sum": 18400,
    "metrics_histogram": 5120,
    "metrics_exponential_histogram": 640,
    "metrics_summary": 210,
    "logs": 22870
  }
}
```

zabbix[boottime]

 Die Startzeit des Zabbix-Server- oder Zabbix-Proxy-Prozesses in Sekunden.
 Rückgabewert: *Integer*.

zabbix[cep]

 Complex event processing statistics.
 Return value: *JSON*.

zabbix[cluster,discovery,nodes]

 Ermittelt die Knoten des Hochverfügbarkeitsclusters.
 Rückgabewert: *JSON-Objekt*.

Kommentare:

- Dieser Datenpunkt kann für die Low-Level-Ermittlung verwendet werden.
- Die zurückgegebenen Felder entsprechen größtenteils den Eigenschaften des Hochverfügbarkeitsknotenobjekts.
- Zusätzliche Felder:
  - `db_timestamp` – aktuelle Serverzeit (Unix-Zeitstempel), die in der Antwort enthalten ist
  - `lastaccess_age` – Sekunden seit dem letzten Heartbeat des Knotens (db_timestamp - lastaccess)
  - `status` – Knotenstatus:
    - 0 – Standby
    - 1 – manuell gestoppt
    - 2 – nicht verfügbar
    - 3 – aktiv

Beispiel für Rückgabewerte:

```
[
  {
    "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]

 Die Anzahl der in die connector queue eingereihten Werte.
 Rückgabewert: *Integer*.

zabbix[discovery_queue]

 Die Anzahl der in die Discovery-Warteschlange eingereihten Netzwerkprüfungen.
 Rückgabewert: *Integer*.

zabbix[host,„items]

 Die Anzahl der aktivierten Datenpunkte (unterstützt und nicht unterstützt) auf dem Host.
 Rückgabewert: *Integer*.

zabbix[host,„items_unsupported]

 Die Anzahl der aktivierten nicht unterstützten Datenpunkte auf dem Host.
 Rückgabewert: *Integer*.

zabbix[host,„maintenance]

 Der aktuelle Wartungsstatus des Hosts.
 Rückgabewerte: *0* - normaler Zustand; *1* - Wartung mit Datenerfassung; *2* - Wartung ohne Datenerfassung.

Kommentare:

- Dieser Datenpunkt wird immer vom Zabbix Server verarbeitet, unabhängig vom Standort des Hosts (auf dem Server oder Proxy). Der Proxy erhält diesen Datenpunkt nicht mit den Konfigurationsdaten.
- Der zweite Parameter muss leer sein und ist für die zukünftige Verwendung reserviert.

zabbix[host,active_agent,available]

 Die Verfügbarkeit aktiver Agent-Prüfungen auf dem Host.
 Rückgabewerte: *0* - unbekannt; *1* - verfügbar; *2* - nicht verfügbar.

zabbix[host,discovery,interfaces]

 Die Details aller konfigurierten Schnittstellen des Hosts im Zabbix Frontend.
 Rückgabewert: *JSON-Objekt*.

Kommentare:

- Dieser Datenpunkt kann in der Low-Level-Discovery verwendet werden.
- Dieser Datenpunkt wird auf dem Zabbix Proxy nicht unterstützt.

zabbix[host,<type>,available]

 Die Verfügbarkeit der Hauptschnittstelle eines bestimmten Prüftyps auf dem Host.
 Rückgabewerte: *0* - nicht verfügbar; *1* - verfügbar; *2* - unbekannt.

Parameter:

- **type** - *agent*, *snmp*, *ipmi* oder *jmx*.

Kommentare:

- Der Datenpunktwert wird gemäß den Konfigurationsparametern zur Nicht-Erreichbarkeit/Nicht-Verfügbarkeit von Hosts berechnet.

zabbixhosts

 Die Anzahl der überwachten Hosts.
 Rückgabewert: *Integer*.

zabbix[items]

 Die Anzahl der aktivierten Datenpunkte (unterstützt und nicht unterstützt).
 Rückgabewert: *Integer*.

zabbix[items_unsupported]

 Die Anzahl der nicht unterstützten Datenpunkte.
 Rückgabewert: *Integer*.

zabbix[java,„<param>]

 Informationen über das Zabbix Java gateway.
 Rückgabewerte: *1* - wenn `<param>` *ping* ist; *Java gateway version* - wenn `<param>` *version* ist (zum Beispiel: ”8.0.0”).

Parameter:

- **param** - *ping* oder *version*.

Kommentare:

- Dieser Datenpunkt kann verwendet werden, um die Verfügbarkeit des Java gateway mit der `nodata()`-Triggerfunktion zu überprüfen.
- Der zweite Parameter muss leer sein und ist für die zukünftige Verwendung reserviert.

zabbix[lld_queue]

 Die Anzahl der Werte, die in die Verarbeitungswarteschlange für Low-Level-Discovery eingereiht sind.
 Rückgabewert: *Integer*.

Kommentare:

- Dieser Datenpunkt kann verwendet werden, um die Länge der Verarbeitungswarteschlange für Low-Level-Discovery zu überwachen.

zabbix[preprocessing]

 Statistiken der vom Preprocessing-Manager empfangenen Werte:

- *queued* - die Anzahl und Größe der in der Warteschlange befindlichen Werte, die Preprocessing erfordern (Zähler)
- *direct* - die Anzahl und Größe der in der Warteschlange befindlichen Werte, die kein Preprocessing erfordern (Zähler)
- *queue* - die Anzahl der in die Preprocessing-Warteschlange eingereihten Werte (entspricht `zabbix[preprocessing_queue]`)

Rückgabewert: *JSON*.

Beispiel für Rückgabewerte:

```
{"data":
    {
    "queued": {
        "count": 106,
        "size": 58620
    },
    "direct": {
        "count": 395,
        "size": 33843
    },
    "queue": 0
    }
}
```

zabbix[preprocessing_queue]

 Die Anzahl der in die Vorverarbeitungswarteschlange eingereihten Werte.
 Rückgabewert: *Integer*.

Kommentare:

- Dieser Datenpunkt kann verwendet werden, um die Länge der Vorverarbeitungswarteschlange zu überwachen.

zabbix[process,<type>,<mode>,<state>]

 Der prozentuale Anteil der Zeit, die ein bestimmter Zabbix-Prozess oder eine Gruppe von Prozessen (identifiziert durch `<type>` und `<mode>`) im Zustand `<state>` verbracht hat. Er wird ausschließlich für die letzte Minute berechnet.
 Rückgabewert: *Float*.

Parameter:

- **type**:
  - für Server-Prozesse: *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*;
  - für Proxy-Prozesse: *Agent poller*, *APM manager*, *APM worker*, *Availability manager*, *Browser poller*, *Configuration syncer*, *Data sender*, *Discovery manager*, *Discovery worker*, *History syncer*, *Housekeeper*, *HTTP Agent poller*, *HTTP poller*, *ICMP pinger*, *Internal poller*, *IPMI manager*, *IPMI poller*, *Java poller*, *ODBC poller*, *Poller*, *Preprocessing manager*, *Preprocessing worker*, *Self-monitoring*, *SNMP poller*, *SNMP trapper*, *Task manager*, *Trapper*, *Unreachable poller*, *VMware collector*;
- **mode**:
  - *avg* – Durchschnittswert für alle Prozesse eines bestimmten Typs (Standard);
  - *count* – gibt die Anzahl der Forks für einen bestimmten Prozesstyp zurück; `<state>` darf nicht angegeben werden;
  - *max* – Maximalwert;
  - *min* – Minimalwert;
  - *<process number>* – Prozessnummer (zwischen 1 und der Anzahl der vorab gestarteten Instanzen; wenn beispielsweise 4 Trapper ausgeführt werden, liegt der Wert zwischen 1 und 4);
- **state**:
  - *busy* – der Prozess befindet sich im ausgelasteten Zustand, beispielsweise bei der Verarbeitung einer Anfrage (Standard);
  - *idle* – der Prozess befindet sich im Leerlauf und führt keine Aktion aus.

Kommentare:

- Wenn `<mode>` eine Zabbix-Prozessnummer ist, die nicht ausgeführt wird (wenn beispielsweise 5 Poller ausgeführt werden und für `<mode>` 6 angegeben ist), wird ein solcher Datenpunkt nicht unterstützt.
- Minimum und Maximum beziehen sich auf den prozentualen Anteil der Nutzung eines einzelnen Prozesses. Wenn die Nutzungsanteile in einer Gruppe von 3 Pollern beispielsweise 2, 18 und 66 betragen, gibt „min“ 2 und „max“ 66 zurück.
- Prozesse melden im gemeinsam genutzten Speicher, was sie gerade tun, und der Self-Monitoring-Prozess fasst diese Daten jede Sekunde zusammen. Zustandsänderungen (busy/idle) werden bei einer Änderung registriert – ein Prozess, der ausgelastet wird, registriert dies entsprechend und ändert oder aktualisiert den Zustand erst wieder, wenn er in den Leerlauf wechselt. Dadurch wird sichergestellt, dass auch vollständig hängende Prozesse korrekt als zu 100 % ausgelastet registriert werden.
- Derzeit bedeutet „busy“ „nicht im Ruhezustand“, aber künftig könnten zusätzliche Zustände eingeführt werden – etwa das Warten auf Sperren oder das Ausführen von Datenbankabfragen. Beachten Sie, dass asynchrone Poller als ausgelastet gelten, wenn sie das durch den Konfigurationsparameter MaxConcurrentChecksPerPoller für Server/Proxy festgelegte Limit erreicht haben.
- Unter Linux und den meisten anderen Systemen beträgt die Auflösung 1/100 Sekunde.

Beispiele:

```
zabbix[process,poller,avg,busy] #der durchschnittliche Zeitanteil, den Poller-Prozesse während der letzten
zabbix[process,"icmp pinger",max,busy] #der maximale Zeitanteil, den ein beliebiger ICMP-Pinger-Prozess wä
zabbix[process,"history syncer",2,busy] #der Zeitanteil, den History Syncer Nummer 2 während der letzten M
zabbix[process,trapper,count] #die Anzahl der derzeit ausgeführten Trapper-Prozesse
```

zabbix[proxy,<name>,<param>]

 Die Informationen über den Zabbix-Proxy.
 Rückgabewert: *Integer*.

Parameter:

- **name** – der Name des Proxys;
- **param**:
  - *lastaccess* – der Zeitstempel der letzten vom Proxy empfangenen Heartbeat-Nachricht;
  - *delay* – wie lange die erfassten Werte noch nicht gesendet wurden; berechnet als „Proxy-Verzögerung“ + („aktuelle Serverzeit“ – „letzter Proxy-Zugriff“), wobei die „Proxy-Verzögerung“ die Differenz zwischen der aktuellen Proxy-Zeit und dem Zeitstempel des ältesten nicht gesendeten Werts auf dem Proxy ist.

Kommentare:

- Dieser Datenpunkt wird auf Hosts, die von einem Proxy oder einer Proxy-Gruppe überwacht werden, nicht unterstützt.
- Mit der Funktion `fuzzytime()` kann die Verfügbarkeit des Proxys überprüft werden.

Beispiel:

```
zabbix[proxy,"Germany",lastaccess] #der Zeitstempel der letzten vom Proxy „Germany“ empfangenen Heartbeat-
```

zabbix[proxy,discovery]

 Die Liste der Zabbix-Proxys mit Name, Modus, Verschlüsselung, Komprimierung, Version, zuletzt gesehen, Host-Anzahl, Datenpunkt-Anzahl, erforderlichen Werten pro Sekunde (vps), Versionsstatus (aktuell/veraltet/nicht unterstützt), Zeitüberschreitungen nach Datenpunkttyp, Proxy-Gruppenname (falls der Proxy zu einer Gruppe gehört) und Status (unbekannt/offline/online).
 Rückgabewert: *JSON-Objekt*.

Kommentar:

- Dieser Datenpunkt wird auf Hosts, die von einem Proxy oder einer Proxy-Gruppe überwacht werden, nicht unterstützt.

Beispiel für Rückgabewerte:

```
[
  {
    "name":"Zabbix proxy",
    "passive":false,
    "unencrypted":true,
    "psk":false,
    "cert":false,
    "items":22,
    "compression":true,
    "version":"8.0.0beta2",
    "compatibility":1,
    "last_seen":1,
    "hosts":1,
    "requiredperformance":0.333333,
    "timeouts": {
        "zabbix_agent":3,
        "simple_check":3,
        "snmp_check":3,
        "external_check":3,
        "db_monitor":3,
        "http_agent":3,
        "ssh_agent":3,
        "telnet_agent":3,
        "script":3,
        "browser":60
    },
    "state":"online"
  }
]
```

zabbix[proxy group,<name>,available]

 Die Anzahl der verfügbaren Proxys in einer Proxygruppe.
 Rückgabewert: *Integer*.

Parameter:

- **name** - der Name der Proxygruppe.

Kommentar:

- Dieser Datenpunkt wird für Hosts, die von einem Proxy oder einer Proxygruppe überwacht werden, nicht unterstützt.

zabbix[proxy group,<name>,pavailable]

 Der Prozentsatz der Online-Proxys in einer Proxygruppe.
 Rückgabewert: *Float*.

Parameter:

- **name** - der Name der Proxygruppe.

Kommentar:

- Dieser Datenpunkt wird auf Hosts, die von einem Proxy oder einer Proxygruppe überwacht werden, nicht unterstützt.

zabbix[proxy group,<name>,proxies]

 Die Liste der Zabbix-Proxys in einer Proxy-Gruppe mit Name, Modus, Verschlüsselung, Komprimierung, Version, zuletzt gesehen, Host-Anzahl, Datenpunktanzahl, erforderlichen Werten pro Sekunde (vps), Versionsstatus (aktuell/veraltet/nicht unterstützt), Timeouts, Proxy-Gruppenname und Status (unbekannt/offline/online).
 Rückgabewert: *JSON*.

Parameter:

- **name** - der Name der Proxy-Gruppe.

Kommentar:

- Dieser Datenpunkt wird auf Hosts, die von einem Proxy oder einer Proxy-Gruppe überwacht werden, nicht unterstützt.

zabbix[proxy group,<name>,state]

 Der Status einer Proxy-Gruppe.
 Rückgabewert: *0* - unbekannt; *1* - offline; *2* - Wiederherstellung; *3* - online; *4* - Verschlechterung.

Parameter:

- **name** - der Name der Proxy-Gruppe.

Kommentar:

- Dieser Datenpunkt wird auf Hosts, die von einem Proxy oder einer Proxy-Gruppe überwacht werden, nicht unterstützt.

zabbix[proxy group,discovery]

 Gibt eine Liste von Proxy-Gruppen mit Konfigurationsdaten und Echtzeitdaten zurück. Zu den Konfigurationsdaten gehören der Name der Proxy-Gruppe, die Failover-Verzögerung und die erforderliche Mindestanzahl an verfügbaren Proxys. Zu den Echtzeitdaten gehören der Status der Proxy-Gruppe (siehe Kommentare für Details), die Anzahl der verfügbaren Proxys und der Prozentsatz der verfügbaren Proxys.
 Rückgabewert: *JSON*.

Kommentare:

- Dieser Datenpunkt wird auf Hosts, die von einem Proxy oder einer Proxy-Gruppe überwacht werden, nicht unterstützt.
- Dieser Datenpunkt gibt Proxys ohne Gruppe nicht zurück.
- Wenn ein ungültiger Wert für `failover_delay` oder `min_online` vorliegt, wird ein spezieller Wert −1 gemeldet, um dies anzuzeigen. Ungültige Werte können auftreten, wenn Makros für die Konfiguration verwendet werden und die Makros nicht in einen gültigen Wert aufgelöst werden können.
- Der Status der Proxy-Gruppe wird als Ganzzahl gemeldet: *0* – unbekannt; *1* – offline; *2* – Wiederherstellung; *3* – online; *4* – beeinträchtigt.

Beispiel für Rückgabewerte:

```
{
    "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>]

 Die Nutzungsstatistiken des Proxy-Speicherpuffers.
 Rückgabewerte: *Integer* (für die Größe); *Float* (für den Prozentsatz).

Parameter:

- **mode**:
  - *total* – die Gesamtgröße des Puffers (kann verwendet werden, um zu prüfen, ob der Speicherpuffer aktiviert ist);
  - *free* – die Größe des freien Puffers;
  - *pfree* – der Prozentsatz des freien Puffers;
  - *used* – die Größe des verwendeten Puffers;
  - *pused* – der Prozentsatz des verwendeten Puffers.

Kommentare:

- Gibt den Fehler „Proxy memory buffer is disabled“ zurück, wenn der Speicherpuffer deaktiviert ist;
- Dieser Datenpunkt wird auf dem Zabbix-Server nicht unterstützt.

zabbix[proxy_buffer,state,changes]

 Gibt die Anzahl der Zustandsänderungen zwischen den Festplatten-/Speicherpuffer-Modi seit dem Start zurück.
 Rückgabewerte: *Integer*; *0* - der Speicherpuffer ist deaktiviert.

Kommentare:

- Häufige Zustandsänderungen weisen darauf hin, dass entweder die Größe oder das Alter des Speicherpuffers erhöht werden muss.
- Wenn der Zustand des Speicherpuffers nur selten überwacht wird (zum Beispiel einmal pro Minute), kann der Puffer seinen Zustand ändern, ohne dass dies registriert wird.

zabbix[proxy_buffer,state,current]

 Gibt den aktuellen Betriebsstatus zurück, in dem die neuen Daten gespeichert werden.
 Rückgabewerte: *0* – Festplatte; *1* – Arbeitsspeicher.

Kommentar:

- 0 wird auch zurückgegeben, wenn der Speicherpuffer deaktiviert ist.

zabbix[proxy_history]

 Die Anzahl der Werte in der Proxy-History-Tabelle, die darauf warten, an den Server gesendet zu werden.
 Rückgabewerte: *Integer*.

Kommentar:

- Dieser Datenpunkt wird auf dem Zabbix-Server nicht unterstützt.

zabbix[queue,<from>,<to>]

 Die Anzahl der überwachten Datenpunkte in der Warteschlange, die um mindestens `<from>` Sekunden, aber weniger als `<to>` Sekunden verzögert sind.
 Rückgabewert: *Integer*.

Parameter:

- **from** - um mindestens verzögert (Standard sind 6 Sekunden);
- **to** - um höchstens verzögert (Standard ist unendlich).

Kommentare:

- Zeitsuffixe (`s,m,h,d,w`) werden in den Parametern unterstützt.

> **Attention:**
> Der Zabbix Proxy ist sich Wartungszeiträumen nicht bewusst; siehe Berechnung von Warteschlangen während der Wartung für Details.

zabbix[rcache,<cache>,<mode>]

 Die Verfügbarkeitsstatistiken des Zabbix-Konfigurationscaches.
 Rückgabewerte: *Integer* (für die Größe); *Float* (für den Prozentsatz).

Parameter:

- **cache** - *buffer*;
- **mode**:
  - *total* - die Gesamtgröße des Buffers;
  - *free* - die Größe des freien Buffers;
  - *pfree* - der Prozentsatz des freien Buffers;
  - *used* - die Größe des verwendeten Buffers;
  - *pused* - der Prozentsatz des verwendeten Buffers.

zabbix[requiredperformance]

 Die erforderliche Leistung des Zabbix-Servers oder Zabbix-Proxy, erwartet in neuen Werten pro Sekunde.
 Rückgabewert: *Float*.

Kommentare:

- Entspricht ungefähr "Required server performance, new values per second" unter *Reports* > *System information*.

zabbix[stats,<ip>,<port>]

 Gibt die internen Metriken eines Zabbix-Servers oder -Proxys zurück. Wenn `<ip>` und `<port>` angegeben sind, werden die Metriken von der entfernten Instanz abgerufen; andernfalls von der lokalen Instanz.
 Rückgabewerte: *JSON-Objekt*.

Parameter:

- **ip** - die IP-/DNS-/Netzmaskenliste der Server/Proxys, die remote abgefragt werden sollen (Standard ist 127.0.0.1);
- **port** - der Port des Servers/Proxys, der remote abgefragt werden soll (Standard ist 10051).

Kommentare:

- Die Stats-Anfrage wird nur von den Adressen akzeptiert, die im Parameter `StatsAllowedIP` Server/Proxy der Zielinstanz aufgeführt sind.
- Eine ausgewählte Menge interner Metriken wird von diesem Datenpunkt zurückgegeben. Weitere Details finden Sie unter Remote monitoring of Zabbix stats.

zabbix[stats,<ip>,<port>,queue,<from>,<to>]

 Gibt die internen Warteschlangenmetriken zurück (siehe `zabbix[queue,<from>,<to>]`) eines Zabbix-Servers oder Proxys. Wenn `<ip>` und `<port>` angegeben sind, werden die Metriken von der entfernten Instanz abgerufen; andernfalls von der lokalen Instanz.
 Rückgabewerte: *JSON-Objekt*.

Parameter:

- **ip** - die IP-/DNS-/Netzmaskenliste der Server/Proxys, die remote abgefragt werden sollen (Standard ist `127.0.0.1`);
- **port** - der Port des Servers/Proxys, der remote abgefragt werden soll (Standard ist 10051);
- **from** - mindestens verzögert um (Standard ist 6 Sekunden);
- **to** - höchstens verzögert um (Standard ist unendlich).

Kommentare:

- Die stats-Anfrage wird nur von den Adressen akzeptiert, die im Parameter `StatsAllowedIP` des Zielsystems Server/Proxy aufgeführt sind.
- Eine ausgewählte Menge interner Metriken wird von diesem Datenpunkt zurückgegeben. Für Details siehe Remote monitoring of Zabbix stats.

> **Attention:**
> Der Zabbix-Proxy ist sich Wartungszeiträumen nicht bewusst; siehe Calculation of queues during maintenance für Details.

zabbix[tcache,cache,<parameter>]

 Die Wirksamkeitsstatistiken des Zabbix-Trendfunktions-Cache.
 Rückgabewerte: *Integer* (für die Größe); *Float* (für den Prozentsatz).

Parameter:

- **parameter**:
  - *all* - gesamte Cache-Anfragen (Standard);
  - *hits* - Cache-Treffer;
  - *phits* - Prozentsatz der Cache-Treffer;
  - *misses* - Cache-Fehlzugriffe;
  - *pmisses* - Prozentsatz der Cache-Fehlzugriffe;
  - *items* - die Anzahl der zwischengespeicherten Elemente;
  - *requests* - die Anzahl der zwischengespeicherten Anfragen;
  - *pitems* - Prozentsatz der zwischengespeicherten Elemente aus zwischengespeicherten Elementen + Anfragen. Eine niedrige Prozentzahl bedeutet höchstwahrscheinlich, dass die Cache-Größe reduziert werden kann.
  - *paccessed* - Prozentsatz der in den letzten 24 Stunden aufgerufenen Einträge.

Kommentare:

- Dieses Item wird auf dem Zabbix Proxy nicht unterstützt.

zabbix[triggers]

 Die Anzahl der aktivierten Auslöser in der Zabbix-Datenbank, bei denen alle Datenpunkte auf aktivierten Hosts aktiviert sind.
 Rückgabewert: *Integer*.

Kommentare:

- Dieser Datenpunkt wird auf dem Zabbix Proxy nicht unterstützt.

zabbix[uptime]

 Die Laufzeit des Zabbix-Server- oder Proxy-Prozesses in Sekunden.
 Rückgabewert: *Integer*.

zabbix[vcache,buffer,<mode>]

 Die Verfügbarkeitsstatistiken des Zabbix-Wertcaches.
 Rückgabewerte: *Integer* (für die Größe); *Float* (für den Prozentsatz).

Parameter:

- **mode**:
  - *total* - die Gesamtgröße des Puffers;
  - *free* - die Größe des freien Puffers;
  - *pfree* - der Prozentsatz des freien Puffers;
  - *used* - die Größe des verwendeten Puffers;
  - *pused* - der Prozentsatz des verwendeten Puffers.

Kommentare:

- Dieser Datenpunkt wird auf einem Zabbix Proxy nicht unterstützt.

zabbix[vcache,cache,<parameter>]

 Die Effektivitätsstatistiken des Zabbix-Wertcaches.
 Rückgabewert: *Integer*. Wenn <parameter> auf *mode* gesetzt ist, lautet der Rückgabewert: *0* - normaler Modus; *1* - Modus mit geringem Speicher.

Parameter:

- **parameter**:
  - *requests* - die Gesamtzahl der Anfragen;
  - *hits* - die Anzahl der Cache-Treffer (Verlaufswerte, die aus dem Cache übernommen wurden);
  - *misses* - die Anzahl der Cache-Fehlschläge (Verlaufswerte, die aus der Datenbank übernommen wurden);
  - *mode* - der Betriebsmodus des Wertcaches.

Kommentare:

- Sobald der Modus mit geringem Speicher aktiviert wurde, bleibt der Wertcache 24 Stunden lang in diesem Zustand, selbst wenn das Problem, das diesen Modus ausgelöst hat, früher behoben wird.
- Sie können diesen Schlüssel mit dem Vorverarbeitungsschritt *Change per second* verwenden, um Werte-pro-Sekunde-Statistiken zu erhalten.
- Dieses Datenpunkt wird auf dem Zabbix Proxy nicht unterstützt.

zabbixversion

 Die Version des Zabbix-Servers oder -Proxys.
 Rückgabewert: *String*. Zum Beispiel: `8.0.0`.

zabbix[vmware,buffer,<mode>]

 Die Verfügbarkeitsstatistik des Zabbix-vmware-Caches.
 Rückgabewerte: *Integer* (für die Größe); *Float* (für den Prozentsatz).

Parameter:

- **mode**:
  - *total* - die Gesamtgröße des Puffers;
  - *free* - die Größe des freien Puffers;
  - *pfree* - der Prozentsatz des freien Puffers;
  - *used* - die Größe des verwendeten Puffers;
  - *pused* - der Prozentsatz des verwendeten Puffers.

zabbix[vps,written]

 Die Gesamtzahl der in die Datenbank geschriebenen Verlaufswerte.
 Rückgabewert: *Integer*.

zabbix[wcache,<cache>,<mode>]

 Die Statistiken und die Verfügbarkeit des Zabbix-Schreibcaches.
 Rückgabewerte: *Integer* (für Anzahl/Größe); *Float* (für Prozent).

Parameter:

- **cache** - *values*, *history*, *index* oder *trend*;
- **mode**:
  - mit dem Cache-Parameter *values*:
    - *all* - die Gesamtzahl der von Server/Proxy verarbeiteten Werte, außer nicht unterstützten Datenpunkten (Zähler) (Standard);
    - *float* - die Anzahl der verarbeiteten Float-Werte (Zähler);
    - *uint* - die Anzahl der verarbeiteten vorzeichenlosen Ganzzahlwerte (Zähler);
    - *str* - die Anzahl der verarbeiteten Zeichen-/String-Werte (Zähler);
    - *log* - die Anzahl der verarbeiteten Log-Werte (Zähler);
    - *text* - die Anzahl der verarbeiteten Text-Werte (Zähler);
    - *bin* - die Anzahl der verarbeiteten Binärwerte (Zähler);
    - *json* - die Anzahl der verarbeiteten JSON-Werte (Zähler);
    - *not supported* - die Anzahl der Fälle, in denen die Verarbeitung eines Datenpunkts dazu führte, dass der Datenpunkt nicht unterstützt wurde oder in diesem Zustand blieb (Zähler);
  - mit den Cache-Parametern *history*, *index*, *trend*:
    - *pfree* - der prozentuale Anteil des freien Puffers (Standard);
    - *total* - die Gesamtgröße des Puffers;
    - *free* - die Größe des freien Puffers;
    - *used* - die Größe des belegten Puffers;
    - *pused* - der prozentuale Anteil des belegten Puffers.

Kommentare:

- Der Cache-Parameter `trend` wird mit Zabbix Proxy nicht unterstützt.
- Der History-Cache wird zum Speichern von Datenpunktwerten verwendet. Eine niedrige Zahl weist auf Leistungsprobleme auf der Datenbankseite hin.
- Der History-Index-Cache wird verwendet, um die im History-Cache gespeicherten Werte zu indizieren.
- Nachdem der History-Cache gefüllt und anschließend geleert wurde, behält der History-Index-Cache weiterhin einige Daten. Dieses Verhalten ist erwartungsgemäß und trägt dazu bei, dass das System effizienter arbeitet, da der zusätzliche Aufwand vermieden wird, der durch das ständige Vergrößern des Speichers entsteht.
- Der Trend-Cache speichert die Aggregation für die aktuelle Stunde für alle Datenpunkte, die Daten empfangen.
- Sie können den Schlüssel `zabbix[wcache,values]` zusammen mit dem Vorverarbeitungsschritt *Change per second* verwenden, um Werte-pro-Sekunde-Statistiken zu erhalten.

Zabbix-Trapper

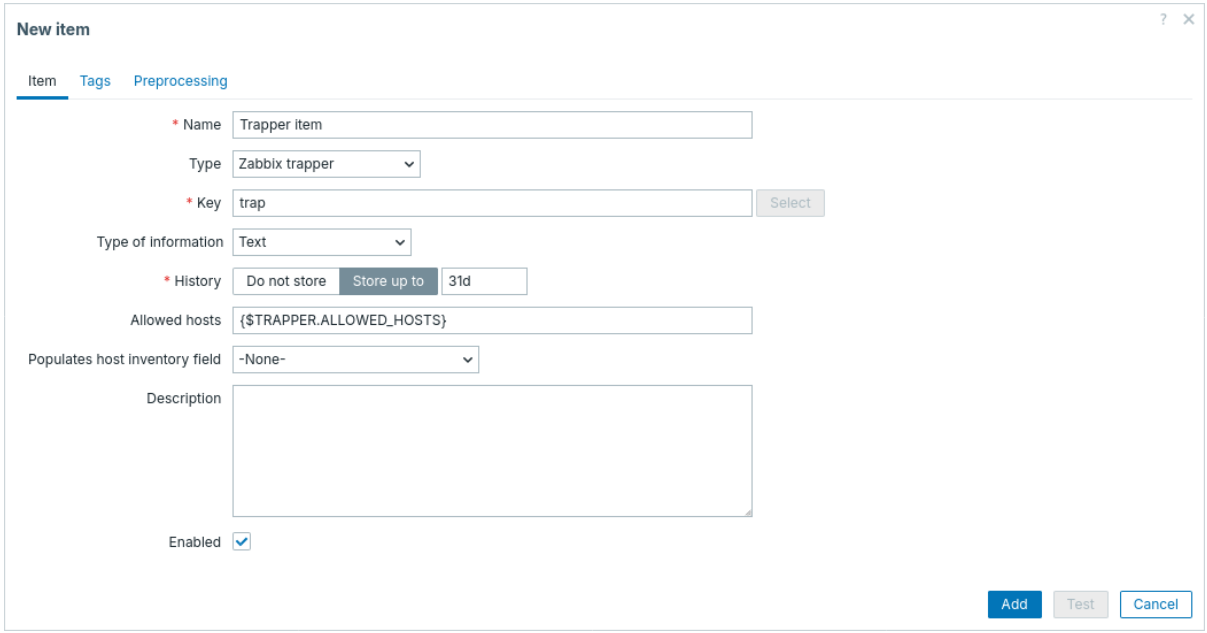
Übersicht

Trapper-Datenpunkte akzeptieren eingehende Daten, anstatt sie abzufragen. Dies ist für alle Daten nützlich, die Sie an Zabbix senden möchten.

Konfiguration

So konfigurieren Sie einen Trapper-Datenpunkt:

1. Gehen Sie zu *Daten sammeln* > *Hosts*.
2. Klicken Sie in der Zeile des Hosts auf *Datenpunkte*.
3. Klicken Sie auf *Datenpunkt erstellen*.
4. Geben Sie die Parameter des Datenpunkts im Konfigurationsformular ein.



Alle obligatorischen Eingabefelder sind mit einem roten Sternchen markiert.

Die Felder, die für Trapper-Datenpunkte spezifische Informationen erfordern, sind:

Typ	Wählen Sie ”Zabbix trapper”.
Schlüssel	Geben Sie einen Schlüssel ein, der verwendet wird, um den Datenpunkt beim Senden von Daten an den Zabbix server zu erkennen.
Informationstyp	Wählen Sie den Informationstyp aus, der dem Format der zu sendenden Daten entspricht.
Zugelassene Hosts	Liste von durch Kommas getrennten IP-Adressen (optional in CIDR-Notation) oder DNS-Namen. Der Standardwert ist das Benutzermakro `{$TRAPPER.ALLOWED_HOSTS}`. Standardmäßig ist `{$TRAPPER.ALLOWED_HOSTS}` `127.0.0.1,::1`. Sie können die Adressen entweder direkt in das Feld eingeben oder den Makrowert ändern. Wenn das Feld leer bleibt, werden eingehende Verbindungen von allen Hosts abgelehnt. Wenn ein Wert angegeben ist, werden eingehende Verbindungen nur von den hier aufgeführten Hosts akzeptiert. Wenn IPv6-Unterstützung aktiviert ist, werden ’127.0.0.1’, ’::127.0.0.1’ und ’::ffff:127.0.0.1’ gleich behandelt, und ’::/0’ erlaubt jede IPv4- oder IPv6-Adresse. ’0.0.0.0/0’ kann verwendet werden, um jede IPv4-Adresse zu erlauben. Beachten Sie, dass ”IPv4-kompatible IPv6-Adressen” (Präfix 0000::/96) unterstützt werden, aber gemäß RFC4291 veraltet sind. Beispiel: 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} Leerzeichen, Benutzermakros und Host-Makros {HOST.HOST}, {HOST.NAME}, {HOST.IP}, {HOST.DNS}, {HOST.CONN} werden unterstützt.

Note:
Bevor Werte gesendet werden, müssen Sie nach dem Speichern des Datenpunkts möglicherweise bis zu 60 Sekunden warten, bis der Zabbix server die Änderungen aus einer Aktualisierung des Konfigurationscaches übernimmt.

Senden von Daten

Das Senden von Daten an den Zabbix Server oder Proxy ist mit dem Dienstprogramm Zabbix sender oder dem Zabbix sender-Protokoll möglich. Das Senden von Daten an den Zabbix Server ist außerdem mit der API-Methode `history.push` möglich.

Zabbix sender

Zum Senden von Daten an den Zabbix Server oder Proxy mit dem Zabbix sender-Dienstprogramm können Sie den folgenden Befehl ausführen, um den Wert ”test value” zu senden:

```
zabbix_sender -z <server IP address> -p 10051 -s "New host" -k trap -o "test value"
```

Um den Wert "test value" zu senden, werden die folgenden Befehlsoptionen verwendet:

- `-z`, um die IP-Adresse des Zabbix Server anzugeben
- `-p`, um die Portnummer des Zabbix Server anzugeben (standardmäßig 10051)
- `-s`, um den Host anzugeben (achten Sie darauf, den technischen statt des sichtbaren Hostnamens zu verwenden)
- `-k`, um den Schlüssel des konfigurierten Datenpunkts im Trapper-Datenpunkt anzugeben
- `-o`, um den zu sendenden Wert anzugeben

Attention:
Der Zabbix Trapper-Prozess erweitert keine Makros, die im Datenpunktschlüssel verwendet werden, um die Existenz des entsprechenden Datenpunktschlüssels für den Ziel-Host zu prüfen.

Weitere Informationen zur Kommunikation zwischen Zabbix sender und Zabbix Server oder Proxy finden Sie unter *Zabbix sender protocol*.

history.push

Um Daten mit der API-Methode `history.push` an den Zabbix-Server zu senden, können Sie die folgende HTTP-POST-Anfrage mit einigen Testwerten ausführen:

```
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
```

Wenn die Anfrage korrekt ist, könnte die von der API zurückgegebene Antwort wie folgt aussehen:

```
{
    "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
}
```

Fehler in den Antwortdaten weisen darauf hin, dass das Senden von Daten für bestimmte Datenpunkte die Validierung durch den Zabbix-Server nicht bestanden hat. Dies kann aus folgenden Gründen passieren:

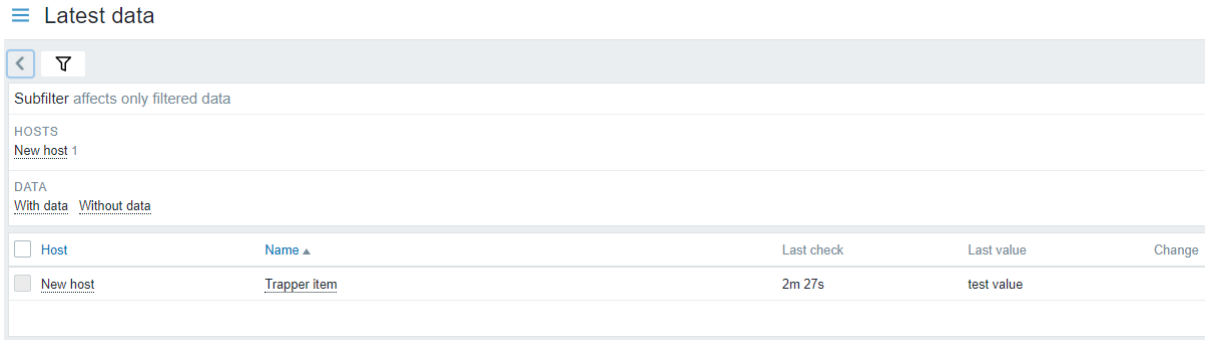
- der Benutzer, der die Daten sendet, hat keine *Leseberechtigung* für den Host des Datenpunkts;
- der Host ist deaktiviert oder befindet sich in der Wartung ohne Datenerfassung;
- der Datenpunkt existiert nicht oder ist noch nicht im Konfigurations-Cache des Servers enthalten;
- der Datenpunkt ist deaktiviert oder sein Typ ist weder Zabbix-Trapper noch HTTP Agent (mit aktiviertem Trapping);
- die IP-Adresse oder der DNS-Name des Benutzers ist nicht in der Liste *Allowed hosts* des Datenpunkts eingetragen;
- ein anderer Datenpunkt hat einen Wert mit einem doppelten Zeitstempel auf Nanosekundenebene.

Das Fehlen von Fehlern zeigt an, dass die gesendeten Werte zur Verarbeitung akzeptiert wurden, einschließlich Vorverarbeitung (falls vorhanden), Auslöser-Verarbeitung und Speicherung in der Datenbank. Beachten Sie, dass die Verarbeitung eines akzeptierten Werts ebenfalls fehlschlagen kann (zum Beispiel während der Vorverarbeitung), wodurch der Wert verworfen wird.

Weitere Informationen zur Arbeit mit der Zabbix-API finden Sie unter API.

Anzeigen von Daten

Sobald Daten gesendet wurden, können Sie zu *Monitoring* > *Neueste Daten* navigieren, um das Ergebnis zu sehen:



Note:
Wenn ein einzelner numerischer Wert gesendet wird, zeigt der Daten-Graph links und rechts vom Zeitpunkt des Werts eine horizontale Linie an.

Externe Prüfung

Übersicht

Ein externer Check ist ein Check, der vom Zabbix Server durch Ausführen eines Shell-Skripts oder eines Binärprogramms durchgeführt wird. Wenn Hosts jedoch von einem Zabbix Proxy überwacht werden, werden die externen Checks vom Proxy ausgeführt.

Externe Checks erfordern keinen laufenden Agent auf dem überwachten Host.

Die Syntax des Datenpunkt-Schlüssels lautet:

```
script[<parameter1>,<parameter2>,...]
```

Dabei gilt:

ARGUMENT	DEFINITION
script	Name eines Shell-Skripts oder eines Binärprogramms.
parameter(s)	Optionale Befehlszeilenparameter.

Wenn Sie keine Parameter an das Skript übergeben möchten, können Sie Folgendes verwenden:

```
script[] oder
script
```

Der Zabbix Server oder Proxy durchsucht das für externe Skripte angegebene Verzeichnis und führt den Befehl aus (siehe Parameter `ExternalScripts` in der Zabbix Server/Proxy-Konfigurationsdatei). Der Befehl wird unter demselben Benutzer ausgeführt wie der Zabbix Server/Proxy, daher sollten Zugriffsberechtigungen oder Umgebungsvariablen bei Bedarf in einem Wrapper-Skript behandelt werden. Die Berechtigungen für den Befehl müssen diesem Benutzer ebenfalls die Ausführung erlauben. Nur Befehle im angegebenen Verzeichnis stehen zur Ausführung zur Verfügung.

Warning:
Verwenden Sie externe Checks nicht übermäßig, da jedes Skript das Starten eines Fork-Prozesses durch den Zabbix Server/Proxy erfordert und die Ausführung vieler Skripte die Zabbix-Performance erheblich beeinträchtigen kann.

Verwendungsbeispiel

Ausführen des Skripts `check_oracle.sh` mit dem ersten Parameter `'-h'`. Der zweite Parameter wird durch die IP-Adresse oder den DNS-Namen ersetzt, abhängig von der Auswahl in den Host-Eigenschaften.

```
check_oracle.sh["-h","{HOST.CONN}"]
```

Wenn angenommen wird, dass der Host so konfiguriert ist, dass er die IP-Adresse verwendet, führt der Zabbix Server/Proxy Folgendes aus:

```
check_oracle.sh '-h' '192.168.1.4'
```

Ergebnis der externen Prüfung

Der Rückgabewert einer externen Prüfung ist die Standardausgabe zusammen mit der Standardfehlerausgabe, die von der Prüfung erzeugt werden.

Attention:
Ein Datenpunkt, der Text zurückgibt (Zeichen-, Log- oder Texttyp von Informationen), wird im Fall einer Standardfehlerausgabe nicht zu „nicht unterstützt".

Der Rückgabewert ist auf 16 MB begrenzt (einschließlich nachgestellter Leerzeichen, die abgeschnitten werden); Datenbankgrenzen gelten ebenfalls.

Wenn das angeforderte Skript nicht gefunden wird oder der Zabbix Server/Proxy keine Berechtigungen hat, es auszuführen, wird der Datenpunkt zu „nicht unterstützt" und eine entsprechende Fehlermeldung wird angezeigt.

Im Fall einer Zeitüberschreitung wird der Datenpunkt zu „nicht unterstützt", eine entsprechende Fehlermeldung wird angezeigt, und der für das Skript erzeugte Prozess wird beendet.

Datenbankmonitor

Übersicht

Der *Database monitor*-Datenpunkttyp im Zabbix-Frontend wird für die ODBC-Überwachung (ODBC-Prüfungen) verwendet.

ODBC ist eine Middleware-API in der Programmiersprache C für den Zugriff auf Datenbankmanagementsysteme (DBMS). Das ODBC-Konzept wurde von Microsoft entwickelt und später auf andere Plattformen portiert.

Zabbix kann jede Datenbank abfragen, die von ODBC unterstützt wird. Dazu stellt Zabbix keine direkte Verbindung zu den Datenbanken her, sondern verwendet die ODBC-Schnittstelle und die in ODBC eingerichteten Treiber. Dies ermöglicht eine effizientere Überwachung verschiedener Datenbanken für mehrere Zwecke (zum Beispiel das Prüfen bestimmter Datenbankwarteschlangen, Nutzungsstatistiken usw.).

Zabbix unterstützt unixODBC, eine der am häufigsten verwendeten Open-Source-Implementierungen der ODBC-API.

Attention:
Siehe auch: bekannte Probleme für ODBC-Prüfungen.

Installieren von unixODBC

Die empfohlene Methode zur Installation von unixODBC ist die Verwendung der standardmäßigen Paketquellen des Linux-Betriebssystems. In den meisten gängigen Linux-Distributionen ist unixODBC standardmäßig im Paket-Repository enthalten. Wenn keine Pakete verfügbar sind, können die Quelldateien auf der unixODBC-Homepage bezogen werden: http://www.unixodbc.org/download.html.

Um unixODBC zu installieren, verwenden Sie den Paketmanager des Systems Ihrer Wahl:

```
##### Für Ubuntu/Debian-Systeme:
apt install unixodbc unixodbc-dev

##### Für RedHat/Fedora-basierte Systeme:
dnf install unixODBC unixODBC-devel

##### Für SUSE-basierte Systeme:
zypper in unixODBC-devel
```

Attention:
Das Paket `unixodbc-dev` oder `unixODBC-devel` ist erforderlich, um Zabbix mit unixODBC-Unterstützung zu kompilieren. Um ODBC-Unterstützung zu aktivieren, sollte Zabbix mit der folgenden Konfigurationsoption kompiliert werden:

`--with-unixodbc[=ARG] # ODBC-Treiber gegen das unixODBC-Paket verwenden.`

Installation von unixODBC-Treibern

Der unixODBC-Datenbanktreiber sollte für die Datenbank installiert werden, die überwacht wird. Eine Liste der unterstützten Datenbanken und Treiber finden Sie auf der unixODBC-Homepage: http://www.unixodbc.org/drivers.html.

Note:
In einigen Linux-Distributionen sind Datenbanktreiber in Paket-Repositories enthalten.

MySQL

Um den MySQL-unixODBC-Datenbanktreiber zu installieren, verwenden Sie den Paketmanager des Systems Ihrer Wahl:

```
##### Für Ubuntu-/Debian-Systeme:
apt install odbc-mariadb

##### Für RedHat-/Fedora-basierte Systeme:
dnf install mariadb-connector-odbc

##### Für SUSE-basierte Systeme:
zypper install mariadb-connector-odbc
```

Um den Datenbanktreiber ohne Paketmanager zu installieren, lesen Sie bitte die MySQL-Dokumentation für `mysql-connector-odbc` oder die MariaDB-Dokumentation für `mariadb-connector-odbc`.

PostgreSQL

Um den PostgreSQL-unixODBC-Datenbanktreiber zu installieren, verwenden Sie den Paketmanager des Systems Ihrer Wahl:

```
##### Für Ubuntu/Debian-Systeme:
apt install odbc-postgresql

##### Für RedHat/Fedora-basierte Systeme:
dnf install postgresql-odbc

##### Für SUSE-basierte Systeme:
zypper install psqlODBC
```

Um den Datenbanktreiber ohne Paketmanager zu installieren, lesen Sie bitte die PostgreSQL-Dokumentation.

Oracle

Um den unixODBC-Datenbanktreiber zu installieren, lesen Sie bitte die Oracle-Dokumentation.

MSSQL

Um den MSSQL-unixODBC-Datenbanktreiber zu installieren, verwenden Sie den Paketmanager des Systems Ihrer Wahl:

```
##### Für Ubuntu-/Debian-Systeme:
apt install tdsodbc

##### Für RedHat-/Fedora-basierte Systeme (EPEL-Pakete: https://docs.fedoraproject.org/en-US/epel/):
dnf install epel-release
dnf install freetds

##### Für SUSE-basierte Systeme:
zypper install libtdsodbc0
```

Um den Datenbanktreiber ohne Paketmanager zu installieren, lesen Sie bitte den FreeTDS user guide.

Konfiguration von unixODBC

Um unixODBC zu konfigurieren, müssen Sie die Dateien `odbcinst.ini` und `odbc.ini` bearbeiten. Sie können den Speicherort dieser Dateien überprüfen, indem Sie den folgenden Befehl ausführen:

```
odbcinst -j
```

Das Befehlsergebnis sollte Informationen enthalten, die in etwa den folgenden entsprechen:

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

odbcinst.ini

Die Datei `odbcinst.ini` listet die installierten ODBC-Datenbanktreiber auf. Falls `odbcinst.ini` fehlt, muss sie manuell erstellt werden.

```
[TEST_MYSQL]
Description=ODBC for MySQL
```

```
Driver=/usr/lib/libmyodbc5.so
FileUsage=1
```

Parameter	Beschreibung
TEST_MYSQL	Name des Datenbanktreibers.
Description	Beschreibung des Datenbanktreibers.
Driver	Speicherort der Bibliothek des Datenbanktreibers.
FileUsage	Legt fest, ob der Datenbanktreiber die Verbindung zu einem Datenbank-Server ohne Unterstützung für den Zugriff auf lokale Dateien unterstützt (0); das Lesen von Daten aus Dateien unterstützt (1); das Schreiben von Daten in Dateien unterstützt (2).
Threading	Grad der Thread-Serialisierung. Unterstützt für PostgreSQL. Seit 1.6 können Sie, wenn der Treibermanager mit Thread-Unterstützung erstellt wurde, einen weiteren Treibereintrag hinzufügen.

odbc.ini

Die Datei `odbc.ini` wird verwendet, um Datenquellen zu konfigurieren. Beachten Sie, dass die Liste der unterstützten Parameter vom Datenbanktreiber abhängt (zum Beispiel können Oracle-Datenbanken statt Server den Parameter ServerName verwenden usw.).

```
[TEST_MYSQL]
Description=MySQL Test Database
Driver=mysql
Server=127.0.0.1
User=root
Password=
Port=3306
Socket=
Database=zabbix
```

Parameter	Beschreibung
TEST_MYSQL	Name der Datenquelle (DSN).
Description	Beschreibung der Datenquelle.
Driver	Name des Datenbanktreibers (wie in `odbcinst.ini` angegeben).
Server	IP/DNS des Datenbank-Servers.
User	Datenbankbenutzer für die Verbindung.
Password	Passwort des Datenbankbenutzers.
Port	Port der Datenbankverbindung.
Socket	Socket der Datenbankverbindung.
Database	Name der Datenbank.

Weitere mögliche Optionen für Konfigurationsparameter finden Sie in der MySQL-Dokumentation.

Die Datei `odbc.ini` für PostgreSQL kann zusätzliche Parameter enthalten:

```
[TEST_PSQL]
Description=PostgreSQL Test Database
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=
```

Parameter	Beschreibung
ReadOnly	Gibt an, ob die Datenbankverbindung nur Leseoperationen (SELECT-Abfragen) zulässt und Änderungen (INSERT-, UPDATE- und DELETE-Anweisungen) einschränkt; nützlich für Szenarien, in denen Daten unverändert bleiben sollen.
Protocol	Version des PostgreSQL-Backend-Protokolls (wird bei Verwendung von SSL-Verbindungen ignoriert).
ShowOidColumn	Gibt an, ob die Object ID (OID) in SQLColumns eingeschlossen werden soll.
FakeOidIndex	Gibt an, ob ein gefälschter eindeutiger Index für OID erstellt werden soll.
RowVersioning	Gibt an, ob Anwendungen erkennen können, ob Daten von anderen Benutzern geändert wurden, während Sie versuchen, eine Zeile zu aktualisieren. Beachten Sie, dass dieser Parameter den Aktualisierungsvorgang beschleunigen kann, da zum Aktualisieren einer Zeile nicht jede einzelne Spalte in der WHERE-Klausel angegeben werden muss.
ShowSystemTables	Gibt an, ob der Datenbanktreiber Systemtabellen in SQLTables als reguläre Tabellen behandeln soll; nützlich für die Zugänglichkeit, da dadurch Systemtabellen sichtbar werden.
Fetch	Gibt an, ob der Treiber automatisch declare cursor/fetch verwenden soll, um SELECT-Anweisungen zu verarbeiten und einen Cache von 100 Zeilen beizubehalten.
BoolsAsChar	Steuert die Zuordnung von booleschen Typen. Wenn auf "Yes" gesetzt, werden boolesche Werte SQL_CHAR zugeordnet; andernfalls werden sie SQL_BIT zugeordnet.
SSLmode	Gibt den SSL-Modus für die Verbindung an.
ConnSettings	Zusätzliche Einstellungen, die beim Verbindungsaufbau an das Backend gesendet werden.

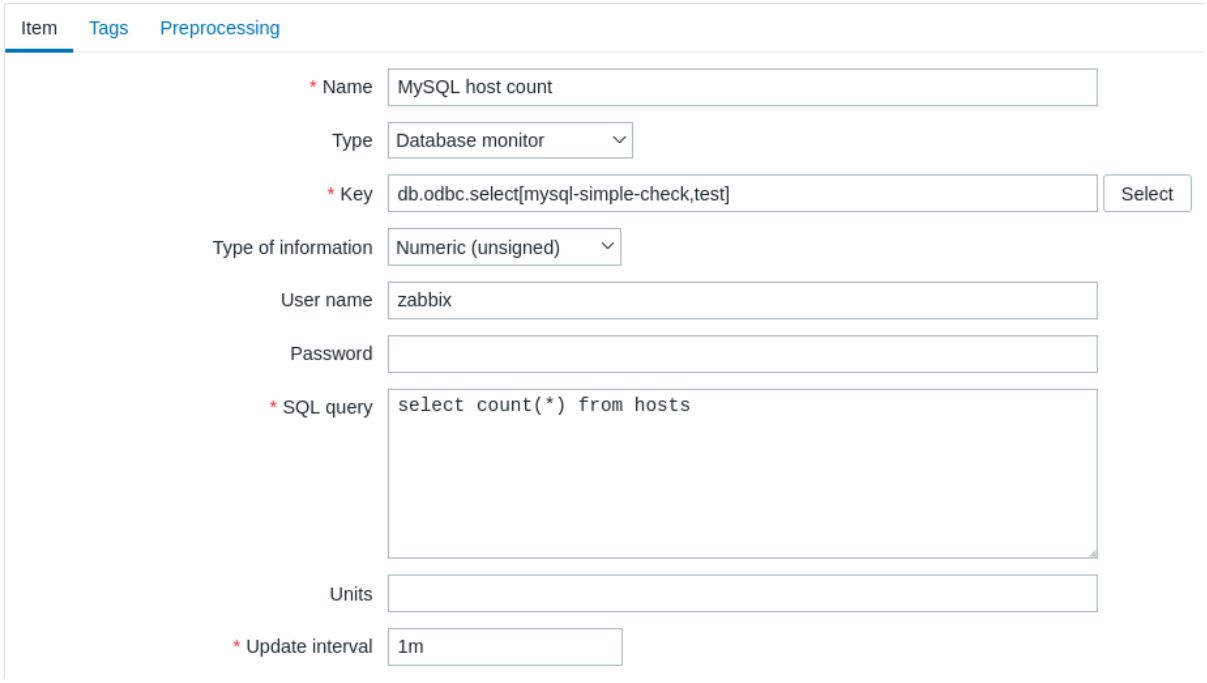
ODBC-Verbindung testen

Um zu testen, ob die ODBC-Verbindung erfolgreich funktioniert, können Sie das Dienstprogramm `isql` verwenden (im Paket `unixODBC` enthalten):

```
isql test
+---------------------------------------+
| Connected!                            |
|                                       |
| sql-statement                         |
| help [tablename]                      |
| quit                                  |
|                                       |
+---------------------------------------+
```

Item-Konfiguration im Zabbix Frontend

Konfigurieren Sie einen Datenpunkt für die **Datenbanküberwachung**.



Alle obligatorischen Eingabefelder sind mit einem roten Sternchen markiert.

Für Datenbanküberwachungs-Datenpunkte müssen Sie Folgendes angeben:

Type	Wählen Sie hier "Database monitor" aus.
Key	Geben Sie einen der unterstützten Datenpunkt-Schlüssel ein: **db.odbc.select[]** - dieser Datenpunkt gibt einen Wert zurück (die erste Spalte der ersten Zeile des SQL-Abfrageergebnisses); **db.odbc.get[]** - dieser Datenpunkt gibt mehrere Zeilen/Spalten im JSON-Format zurück; **db.odbc.discovery[]** - dieser Datenpunkt gibt Daten für die Low-Level-Erkennung zurück.
User name	Geben Sie den Benutzernamen der Datenbank ein (bis zu 255 Zeichen). Dieser Parameter ist optional, wenn der Datenbank-Benutzername in der Datei `odbc.ini` angegeben ist. Wenn eine Verbindungszeichenfolge verwendet wird und das Feld *User name* nicht leer ist, wird es als `UID=<user>` an die Verbindungszeichenfolge angehängt.
Password	Geben Sie das Datenbank-Benutzerpasswort ein (bis zu 255 Zeichen). Dieser Parameter ist optional, wenn das Passwort in der Datei `odbc.ini` angegeben ist. Wenn eine Verbindungszeichenfolge verwendet wird und das Feld *Password* nicht leer ist, wird es als `PWD=<password>` an die Verbindungszeichenfolge angehängt. Sonderzeichen werden in diesem Feld unterstützt. Für alle Datenbanken außer Oracle müssen Passwörter, die `%` enthalten, in geschweifte Klammern `{}` eingeschlossen werden. Das Passwort wird nach dem Benutzernamen an die Verbindungszeichenfolge angehängt, zum Beispiel `UID=<username>;PWD=P?;)*word`. Um die resultierende Zeichenfolge zu testen, können Sie den folgenden Befehl ausführen: `isql -v -k 'Driver=libmaodbc.so;Database=zabbix;UID=zabbix;PWD=P?;)*word'`
SQL query	Geben Sie die SQL-Abfrage ein. Beachten Sie, dass bei `db.odbc.select[]` die Abfrage nur einen Wert zurückgeben darf.
Type of information	Wählen Sie hier den Typ der Informationen aus, die von der Abfrage zurückgegeben werden. Wenn der Informationstyp falsch ausgewählt ist, wird der Datenpunkt nicht unterstützt.

Wichtige Hinweise

- Datenbanküberwachungs-Datenpunkte werden nicht mehr unterstützt, wenn in der Server- oder Proxy-Konfiguration keine *odbc poller*-Prozesse gestartet werden. Um ODBC-Poller zu aktivieren, setzen Sie den Parameter `StartODBCPollers` in der Zabbix-Server-Konfigurationsdatei oder, für Prüfungen, die vom Proxy ausgeführt werden, in der Zabbix-Proxy-Konfigurationsdatei.
- Der Wert des Parameters *Timeout* im Formular Datenpunktkonfiguration wird als ODBC-Anmelde-Timeout und als Timeout für die Ausführung der Abfrage verwendet. Beachten Sie, dass diese Timeout-Einstellungen möglicherweise ignoriert werden, wenn der installierte ODBC-Treiber sie nicht unterstützt.
- Der SQL-Befehl muss einen Resultset zurückgeben, wie jede Abfrage mit der Anweisung `select`. Die Abfragesyntax hängt vom RDBMS ab, das sie verarbeitet. Die Syntax einer Anfrage an eine gespeicherte Prozedur muss mit dem Schlüsselwort `call` beginnen.

Details zum Datenpunktschlüssel

Parameter ohne spitze Klammern sind obligatorisch. Parameter, die mit spitzen Klammern **< >** gekennzeichnet sind, sind optional.

db.odbc.select[<eindeutige kurze Beschreibung>,<dsn>,<Verbindungszeichenfolge>]

 Gibt einen Wert zurück, nämlich die erste Spalte der ersten Zeile des SQL-Abfrageergebnisses.
 Rückgabewert: Abhängig von der SQL-Abfrage.

Parameter:

- **eindeutige kurze Beschreibung** - eine eindeutige kurze Beschreibung zur Identifizierung des Datenpunkts (zur Verwendung in Auslösern usw.);
- **dsn** - der Name der Datenquelle (wie in `odbc.ini` angegeben);
- **Verbindungszeichenfolge** - die Verbindungszeichenfolge (kann treiberspezifische Argumente enthalten).

Kommentare:

- Obwohl `dsn` und `Verbindungszeichenfolge` optionale Parameter sind, ist mindestens einer von ihnen erforderlich; wenn beide definiert sind, wird `dsn` ignoriert.
- Wenn eine Abfrage mehr als eine Spalte zurückgibt, wird nur die erste Spalte gelesen. Wenn eine Abfrage mehr als eine Zeile zurückgibt, wird nur die erste Zeile gelesen.

db.odbc.get[<unique short description>,<dsn>,<connection string>]

 Transformiert das Ergebnis der SQL-Abfrage in ein JSON-Array.
 Rückgabewert: *JSON object*.

Parameter:

- **unique short description** - eine eindeutige kurze Beschreibung zur Identifizierung des Datenpunkts (zur Verwendung in Auslösern usw.);
- **dsn** - der Name der Datenquelle (wie in `odbc.ini` angegeben);
- **connection string** - die Verbindungszeichenfolge (kann treiberspezifische Argumente enthalten).

Kommentare:

- Obwohl `dsn` und `connection string` optionale Parameter sind, ist mindestens einer von ihnen erforderlich; wenn beide definiert sind, wird `dsn` ignoriert.
- Mehrere Zeilen/Spalten im JSON-Format können zurückgegeben werden. Dieser Datenpunkt kann als Master-Datenpunkt verwendet werden, der alle Daten in einem Systemaufruf sammelt, während die JSONPath-Vorverarbeitung in abhängigen Datenpunkten verwendet werden kann, um einzelne Werte zu extrahieren. Weitere Informationen finden Sie in einem Beispiel des zurückgegebenen Formats, das in der Low-Level-Discovery verwendet wird.

Beispiel:

```
# Verbindung für MySQL ODBC-Treiber 5:
db.odbc.get[MySQL example,,"Driver=/usr/local/lib/libmyodbc5a.so;Database=master;Server=127.0.0.1;Port=330
```

db.odbc.discovery[<eindeutige Kurzbeschreibung>,<dsn>,<Verbindungszeichenfolge>]

 Wandelt das Ergebnis der SQL-Abfrage in ein JSON-Array um, das für Low-Level-Discovery verwendet wird. Die Spaltennamen aus dem Abfrageergebnis werden in Makronamen für die Low-Level-Discovery umgewandelt und mit den ermittelten Feldwerten verknüpft. Diese Makros können beim Erstellen von Datenpunkt-, Auslöser- usw. Prototypen verwendet werden.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **eindeutige Kurzbeschreibung** - eine eindeutige Kurzbeschreibung zur Identifizierung des Datenpunkts (zur Verwendung in Auslösern usw.);
- **dsn** - der Name der Datenquelle (wie in `odbc.ini` angegeben);
- **Verbindungszeichenfolge** - die Verbindungszeichenfolge (kann treiberspezifische Argumente enthalten).

Kommentare:

- Obwohl `dsn` und `Verbindungszeichenfolge` optionale Parameter sind, ist mindestens einer von ihnen erforderlich; wenn beide definiert sind, wird `dsn` ignoriert.

Fehlermeldungen

ODBC-Fehlermeldungen sind in Felder strukturiert, um detaillierte Informationen bereitzustellen. Zum Beispiel könnte eine Fehlermeldung wie folgt aussehen:

```
ODBC-Abfrage kann nicht ausgeführt werden: [SQL_ERROR]:[42601][7][ERROR: syntax error at or near ";"; Erro
```

- "ODBC-Abfrage kann nicht ausgeführt werden" - Zabbix-Meldung

- "[SQL_ERROR]" - ODBC-Rückgabecode
- "[42601]" - SQLState
- "[7]" - nativer Fehlercode
- "[ERROR: syntax error at or near ";"; Error while executing the query]" - native Fehlermeldung

Beachten Sie, dass die Länge der Fehlermeldung auf 2048 Byte begrenzt ist, daher kann die Meldung abgeschnitten werden. Wenn es mehr als einen ODBC-Diagnosedatensatz gibt, versucht Zabbix, diese zu verketten (getrennt durch |), soweit es die Längenbegrenzung zulässt.

HTTP-Agent

Übersicht

Dieser Datentyp ermöglicht das Abrufen von Daten über das HTTP/HTTPS-Protokoll. Das Trapping ist auch mit dem Dienstprogramm Zabbix sender oder dem Protokoll von Zabbix sender möglich (zum Senden von Daten an den Zabbix Server oder Proxy) oder mithilfe der API-Methode `history.push` (zum Senden von Daten an den Zabbix Server).

HTTP-Datenpunktprüfungen werden vom Zabbix Server ausgeführt. Wenn Hosts jedoch von einem Zabbix Proxy überwacht werden, werden HTTP-Datenpunktprüfungen vom Proxy ausgeführt.

HTTP-Datenpunktprüfungen erfordern keinen auf dem überwachten Host laufenden Agent.

HTTP agent unterstützt sowohl HTTP als auch HTTPS. Zabbix folgt optional Weiterleitungen (siehe die Option *Weiterleitungen folgen* unten). Die maximale Anzahl von Weiterleitungen ist fest auf 10 codiert (unter Verwendung der cURL-Option `CURLOPT_MAXREDIRS`).

> **Attention:**
> Der Zabbix Server/Proxy muss zunächst mit Unterstützung für cURL (libcurl) konfiguriert werden.

HTTP-Prüfungen werden asynchron ausgeführt - es ist nicht erforderlich, die Antwort auf eine Anfrage zu erhalten, bevor andere Prüfungen gestartet werden. Auch die DNS-Auflösung erfolgt asynchron.

Die maximale Parallelität asynchroner Prüfungen beträgt 1000 (definiert durch MaxConcurrentChecksPerPoller).

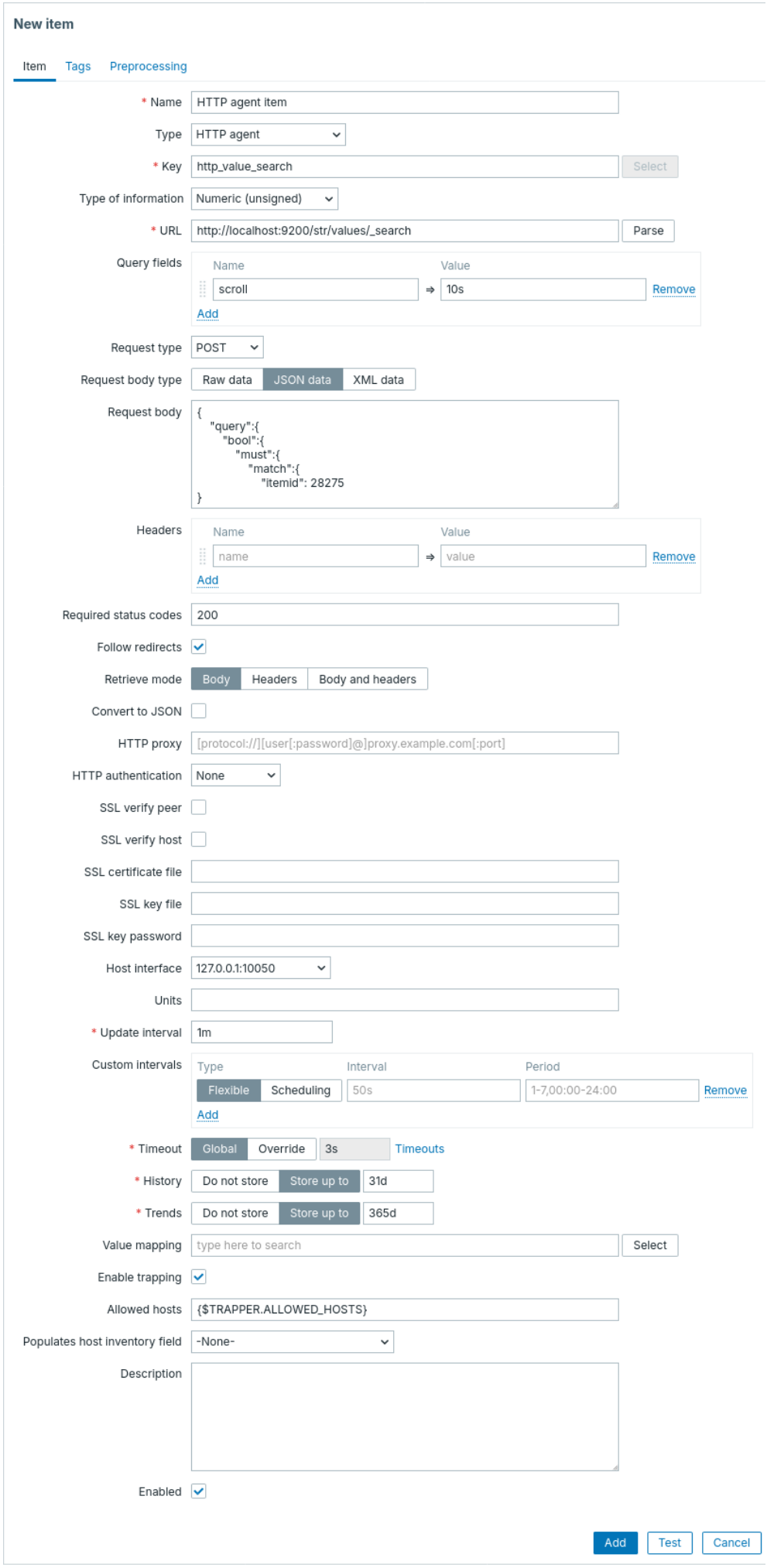
Die Anzahl der asynchronen HTTP agent Poller wird durch den Parameter StartHTTPAgentPollers definiert.

Die cURL-Funktion für persistente Verbindungen wurde seit Zabbix 7.0 zu HTTP agent-Prüfungen hinzugefügt.

Konfiguration

So konfigurieren Sie einen HTTP-Datenpunkt:

1. Gehen Sie zu *Datenerfassung* > *Hosts*.
2. Klicken Sie in der Zeile des Hosts auf *Datenpunkte*.
3. Klicken Sie auf *Datenpunkt erstellen*.
4. Geben Sie die Parameter des Datenpunkts im Formular ein.


New item
Item
Tags
Preprocessing
* Name
HTTP agent item
Type
HTTP agent
* Key
http_value_search
Select
Type of information
Numeric (unsigned)
* URL
http://localhost:9200/str/values/_search
Parse
Query fields
Name
Value
scroll
10s
Remove
Add
Request type
POST
Request body type
Raw data
JSON data
XML data
Request body
{
"query":{
"bool":{
"must":{
"match":{
"itemid": 28275
}
Headers
Name
Value
name
value
Remove
Add
Required status codes
200
Follow redirects
Retrieve mode
Body
Headers
Body and headers
Convert to JSON
HTTP proxy
[protocol://][user[:password]@]proxy.example.com[:port]
HTTP authentication
None
SSL verify peer
SSL verify host
SSL certificate file
SSL key file
SSL key password
Host interface
127.0.0.1:10050
Units
* Update interval
1m
Custom intervals
Type
Interval
Period
Flexible
Scheduling
50s
1-7,00:00-24:00
Remove
Add
* Timeout
Global
Override
3s
Timeouts
* History
Do not store
Store up to
31d
* Trends
Do not store
Store up to
365d
Value mapping
type here to search
Select
Enable trapping
Allowed hosts
{$TRAPPER.ALLOWED_HOSTS}
Populates host inventory field
-None-
Description
Enabled
Add
Test
Cancel

Alle Pflichtfelder sind mit einem roten Sternchen gekennzeichnet.

Die Felder, die für HTTP-Datenpunkte spezifische Informationen erfordern, sind:

Parameter	Beschreibung
Typ	Wählen Sie hier **HTTP agent** aus.
Schlüssel	Geben Sie einen eindeutigen Datenpunktschlüssel ein.

URL	URL, zu der eine Verbindung hergestellt und von der Daten abgerufen werden sollen. Zum Beispiel:
	`https://www.example.com`
	`http://www.example.com/download`
	Domänennamen können in Unicode-Zeichen angegeben werden. Bei der Ausführung der HTTP-Prüfung werden sie automatisch in Punycode und anschließend in ASCII konvertiert.
	Mit der Schaltfläche *Parsen* können optionale Abfragefelder (wie `?key=value&limit=10`) von der URL getrennt werden. Dabei werden die Attribute und Werte zur automatischen URL-Kodierung in *Abfragefelder* verschoben.
	Geben Sie aus Sicherheitsgründen keine tatsächlichen Benutzernamen oder Passwörter in URLs an. Verwenden Sie stattdessen Benutzermakros, um die Offenlegung von Zugangsdaten zu vermeiden.
	Auf 2048 Zeichen begrenzt.
	Unterstützte Makros: `{HOST.IP}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.PORT}`, `{ITEM.ID}`, `{ITEM.KEY}`, `{ITEM.KEY.ORIG}`, Benutzermakros, Makros der Low-Level-Discovery.
	Damit wird die cURL-Option `CURLOPT_URL` gesetzt.

Abfragefelder	Variablen für die URL (siehe oben). Werden als Attribut-Wert-Paare angegeben. Werte werden automatisch URL-kodiert. Werte aus Makros werden aufgelöst und anschließend automatisch URL-kodiert. Unterstützte Makros: `{HOST.IP}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.PORT}`, `{ITEM.ID}`, `{ITEM.KEY}`, `{ITEM.KEY.ORIG}`, Benutzermakros, Makros der Low-Level-Discovery. Damit wird die cURL-Option `CURLOPT_URL` gesetzt.
Anforderungstyp	Wählen Sie den Typ der Anforderungsmethode aus: *GET*, *POST*, *PUT* oder *HEAD*.
Typ des Anforderungstexts	Wählen Sie den Typ des Anforderungstexts aus: **Rohdaten** – benutzerdefinierter HTTP-Anforderungstext; Makros werden ersetzt, aber es wird keine Kodierung durchgeführt. **JSON-Daten** – HTTP-Anforderungstext im JSON-Format. Makros können als Zeichenkette, Zahl, true und false verwendet werden; als Zeichenketten verwendete Makros müssen in doppelte Anführungszeichen eingeschlossen werden. Werte aus Makros werden aufgelöst und anschließend automatisch maskiert. Wenn in den Headern kein „Content-Type“ angegeben ist, wird standardmäßig „Content-Type: application/json“ verwendet. **XML-Daten** – HTTP-Anforderungstext im XML-Format. Makros können als Textknoten, Attribut oder CDATA-Abschnitt verwendet werden. Werte aus Makros werden in einem Textknoten und Attribut aufgelöst und anschließend automatisch maskiert. Wenn in den Headern kein „Content-Type“ angegeben ist, wird standardmäßig „Content-Type: application/xml“ verwendet.
Anforderungstext	Geben Sie den Anforderungstext ein. Unterstützte Makros: `{HOST.IP}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.PORT}`, `{ITEM.ID}`, `{ITEM.KEY}`, `{ITEM.KEY.ORIG}`, Benutzermakros, Makros der Low-Level-Discovery.
Header	Benutzerdefinierte HTTP-Header, die bei der Ausführung einer Anforderung gesendet werden. Werden als Attribut-Wert-Paare angegeben. Unterstützte Makros: `{HOST.IP}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.PORT}`, `{ITEM.ID}`, `{ITEM.KEY}`, `{ITEM.KEY.ORIG}`, Benutzermakros, Makros der Low-Level-Discovery. Damit wird die cURL-Option `CURLOPT_HTTPHEADER` gesetzt.
Erforderliche Statuscodes	Liste der erwarteten HTTP-Statuscodes. Wenn Zabbix einen Code erhält, der nicht in der Liste enthalten ist, wird der Datenpunkt nicht unterstützt. Wenn das Feld leer ist, wird keine Prüfung durchgeführt. Zum Beispiel: 200,201,210−299. Unterstützte Makros in der Liste: Benutzermakros, Makros der Low-Level-Discovery. Hierfür wird die cURL-Option `CURLINFO_RESPONSE_CODE` verwendet.
Weiterleitungen folgen	Aktivieren Sie das Kontrollkästchen, um HTTP-Weiterleitungen zu folgen. Damit wird die cURL-Option `CURLOPT_FOLLOWLOCATION` gesetzt.

Parameter	Beschreibung
Abrufmodus	Wählen Sie den Teil der Antwort aus, der abgerufen werden soll: **Text** – nur Text **Header** – nur Header **Text und Header** – Text und Header
In JSON konvertieren	Header werden als Attribut-Wert-Paare unter dem Schlüssel „header“ gespeichert, wenn *Abrufmodus* nicht auf *Text* gesetzt ist. Wenn „Content-Type: application/json“ erkannt wird, wird der Text als Objekt gespeichert; andernfalls wird er als Zeichenkette gespeichert, zum Beispiel: `{ "header": { "<key>": "<value>", "<key2>": "<value>" }, "body": <body> }`
HTTP-Proxy	Sie können einen zu verwendenden HTTP-Proxy im Format `[protocol://][username[:password]@]proxy.example.com[:port]` angeben. Das optionale Präfix `protocol://` kann verwendet werden, um alternative Proxy-Protokolle anzugeben (z. B. `https`, `socks4`, `socks5`; siehe Dokumentation; die Unterstützung des Protokollpräfixes wurde in cURL 7.21.7 hinzugefügt). Wenn kein Protokoll angegeben ist, wird der Proxy als HTTP-Proxy behandelt. Wenn Sie das falsche Protokoll angeben, schlägt die Verbindung fehl und der Datenpunkt wird nicht unterstützt. Standardmäßig wird der Port 1080 verwendet. Wenn angegeben, überschreibt der Proxy proxybezogene Umgebungsvariablen wie `http_proxy` und `HTTPS_PROXY`. Wenn kein Proxy angegeben ist, werden proxybezogene Umgebungsvariablen nicht überschrieben. Der eingegebene Wert wird unverändert übernommen; es findet keine Plausibilitätsprüfung statt. Beachten Sie, dass mit einem HTTP-Proxy nur die einfache Authentifizierung unterstützt wird. Unterstützte Makros: `{HOST.IP}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.PORT}`, `{ITEM.ID}`, `{ITEM.KEY}`, `{ITEM.KEY.ORIG}`, Benutzermakros, Makros der Low-Level-Discovery. Damit wird die cURL-Option `CURLOPT_PROXY` gesetzt.
HTTP-Authentifizierung	Wählen Sie die Authentifizierungsoption aus: **Keine** – keine Authentifizierung wird verwendet; **Basic** – die Basic-Authentifizierung wird verwendet; **NTLM** – die NTLM-Authentifizierung (Windows NT LAN Manager) wird verwendet; **Kerberos** – die Kerberos-Authentifizierung wird verwendet (siehe auch: Kerberos mit Zabbix konfigurieren); **Digest** – die Digest-Authentifizierung wird verwendet. Damit wird die cURL-Option `CURLOPT_HTTPAUTH` gesetzt.
Benutzername	Geben Sie den Benutzernamen ein (bis zu 255 Zeichen). Dieses Feld ist verfügbar, wenn *HTTP-Authentifizierung* auf *Basic*, *NTLM*, *Kerberos* oder *Digest* gesetzt ist. Benutzermakros und Makros der Low-Level-Discovery werden unterstützt.
Passwort	Geben Sie das Benutzerpasswort ein (bis zu 255 Zeichen). Dieses Feld ist verfügbar, wenn *HTTP-Authentifizierung* auf *Basic*, *NTLM*, *Kerberos* oder *Digest* gesetzt ist. Benutzermakros und Makros der Low-Level-Discovery werden unterstützt.
SSL-Peer überprüfen	Aktivieren Sie das Kontrollkästchen, um das SSL-Zertifikat des Webservers zu überprüfen. Das Serverzertifikat wird anhand der vertrauenswürdigen Zertifikate der Zertifizierungsstellen (CA) aus dem systemweiten CA-Speicher überprüft. Sie können den Speicherort der CA-Dateien mit dem Zabbix-Server- oder Proxy-Konfigurationsparameter SSLCALocation überschreiben. Damit wird die cURL-Option `CURLOPT_SSL_VERIFYPEER` gesetzt.
SSL-Host überprüfen	Aktivieren Sie das Kontrollkästchen, um zu überprüfen, ob das Feld *Common Name* oder das Feld *Subject Alternate Name* des Webserverzertifikats übereinstimmt. Damit wird die cURL-Option `CURLOPT_SSL_VERIFYHOST` gesetzt.

Parameter	Beschreibung
SSL-Zertifikatsdatei	Name der für die Clientauthentifizierung verwendeten SSL-Zertifikatsdatei. Die Zertifikatsdatei muss im PEM[1]-Format vorliegen. Wenn die Zertifikatsdatei auch den privaten Schlüssel enthält, lassen Sie das Feld *SSL-Schlüsseldatei* leer. Wenn der Schlüssel verschlüsselt ist, geben Sie das Passwort im Feld *SSL-Schlüsselpasswort* an. Das Verzeichnis, das diese Datei enthält, wird durch den Zabbix-Server- oder Proxy-Konfigurationsparameter `SSLCertLocation` angegeben. Unterstützte Makros: `{HOST.IP}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.PORT}`, `{ITEM.ID}`, `{ITEM.KEY}`, `{ITEM.KEY.ORIG}`, Benutzermakros, Makros der Low-Level-Discovery. Damit wird die cURL-Option `CURLOPT_SSLCERT` gesetzt.
SSL-Schlüsseldatei	Name der für die Clientauthentifizierung verwendeten privaten SSL-Schlüsseldatei. Die Datei mit dem privaten Schlüssel muss im PEM[1]-Format vorliegen. Das Verzeichnis, das diese Datei enthält, wird durch den Zabbix-Server- oder Proxy-Konfigurationsparameter `SSLKeyLocation` angegeben. Unterstützte Makros: `{HOST.IP}`, `{HOST.CONN}`, `{HOST.DNS}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.PORT}`, `{ITEM.ID}`, `{ITEM.KEY}`, `{ITEM.KEY.ORIG}`, Benutzermakros, Makros der Low-Level-Discovery. Damit wird die cURL-Option `CURLOPT_SSLKEY` gesetzt.
SSL-Schlüsselpasswort	Passwort der privaten SSL-Schlüsseldatei. Unterstützte Makros: Benutzermakros, Makros der Low-Level-Discovery. Damit wird die cURL-Option `CURLOPT_KEYPASSWD` gesetzt.
Zeitüberschreitung	Zabbix verwendet für die Verarbeitung der URL nicht mehr als die festgelegte Zeit (1–600 Sekunden). Tatsächlich definiert dieser Parameter die maximale Zeit für den Verbindungsaufbau zur URL und die maximale Zeit für die Ausführung einer HTTP-Anforderung. Daher verwendet Zabbix für eine Prüfung nicht mehr als 2 x *Zeitüberschreitung* Sekunden. Damit wird die cURL-Option `CURLOPT_TIMEOUT` gesetzt. Weitere Informationen zum Parameter *Zeitüberschreitung* finden Sie unter Allgemeine Datenpunktattribute.
Trapping aktivieren	Wenn dieses Kontrollkästchen aktiviert ist, funktioniert der Datenpunkt auch als Trapper-Datenpunkt und akzeptiert Daten, die mit dem Dienstprogramm Zabbix sender oder dem Protokoll von Zabbix sender an den Zabbix-Server oder Proxy gesendet werden, oder Daten, die mit der API-Methode `history.push` an den Zabbix-Server gesendet werden. Weitere Informationen zum Senden von Daten finden Sie unter: Trapper-Datenpunkte.
Zulässige Hosts	Nur sichtbar, wenn das Kontrollkästchen *Trapping aktivieren* aktiviert ist. Liste durch Kommas getrennter IP-Adressen (optional in CIDR-Notation) oder DNS-Namen. Der Standardwert ist das Benutzermakro `{$TRAPPER.ALLOWED_HOSTS}`. Standardmäßig lautet `{$TRAPPER.ALLOWED_HOSTS}` `127.0.0.1,::1`. Sie können Adressen direkt in das Feld eingeben oder den Makrowert ändern. Wenn das Feld leer bleibt, werden eingehende Verbindungen von allen Hosts abgelehnt. Wenn ein Wert angegeben ist, werden eingehende Verbindungen nur von den hier aufgeführten Hosts akzeptiert. Wenn die IPv6-Unterstützung aktiviert ist, werden `127.0.0.1`, `::127.0.0.1` und `::ffff:127.0.0.1` gleich behandelt, und `::/0` erlaubt jede IPv4- oder IPv6-Adresse. `0.0.0.0/0` kann verwendet werden, um jede IPv4-Adresse zuzulassen. Beachten Sie, dass „IPv4-kompatible IPv6-Adressen“ (Präfix `0000::/96`) unterstützt, aber durch RFC4291 als veraltet eingestuft werden. Beispiel: `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}` Leerzeichen und Benutzermakros sind in diesem Feld zulässig. Hostmakros: `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.IP}`, `{HOST.DNS}`, `{HOST.CONN}` sind in diesem Feld zulässig.

Note:
Wenn das Feld *HTTP-Proxy* leer bleibt, können Sie einen HTTP-Proxy auch über proxybezogene Umgebungsvariablen verwenden.
Für HTTP - setzen Sie die Umgebungsvariable `http_proxy` für den Zabbix-Serverbenutzer. Zum Beispiel:
 `http_proxy=http://proxy_ip:proxy_port`.
Für HTTPS - setzen Sie die Umgebungsvariable `HTTPS_PROXY`. Zum Beispiel:
 `HTTPS_PROXY=http://proxy_ip:proxy_port`.
Weitere Informationen erhalten Sie durch Ausführen des Shell-Befehls: *# man curl*.

Attention:
[1] Zabbix unterstützt Zertifikats- und private Schlüsseldateien ausschließlich im PEM-Format. Wenn Ihre Zertifikats- und privaten Schlüsseldaten in einer Datei im PKCS-#12-Format vorliegen (normalerweise mit der Erweiterung `*.p12` oder `*.pfx`), können Sie daraus mit den folgenden Befehlen eine PEM-Datei erzeugen:

```
openssl pkcs12 -in ssl-cert.p12 -clcerts -nokeys -out ssl-cert.pem
openssl pkcs12 -in ssl-cert.p12 -nocerts -nodes  -out ssl-cert.key
```

Beispiele

Beispiel 1

Senden Sie einfache GET-Anfragen, um Daten von Diensten wie Elasticsearch abzurufen:

- Erstellen Sie einen GET-Datenpunkt mit der URL: `localhost:9200/?pretty`
- Beachten Sie die Antwort:

```
{
  "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"
}
```

- Extrahieren Sie nun die Versionsnummer mithilfe eines JSONPath-Preprocessing-Schritts: `$.version.number`

Beispiel 2

Senden Sie einfache POST-Anfragen, um Daten von Diensten wie Elasticsearch abzurufen:

- Erstellen Sie einen POST-Datenpunkt mit der URL: `http://localhost:9200/str/values/_search?scroll=10s`
- Konfigurieren Sie den folgenden POST-Body, um die Prozessorlast zu ermitteln (1-Minuten-Durchschnitt pro Kern)

```
{
    "query": {
        "bool": {
            "must": [{
                "match": {
                    "itemid": 28275
                }
            }],
            "filter": [{
                "range": {
                    "clock": {
                        "gt": 1517565836,
                        "lte": 1517566137
                    }
                }
            }]
        }
    }
}
```

- Empfangen:

```
{
    "_scroll_id": "DnF1ZXJ5VGhlbkZldGNoBQAAAAAAAAAkFllRMlZBWS1UU1pxTmdEeGVwQjRBTFEAAAAAAAAAJRZZUTJWQVk
    "took": 18,
    "timed_out": false,
```

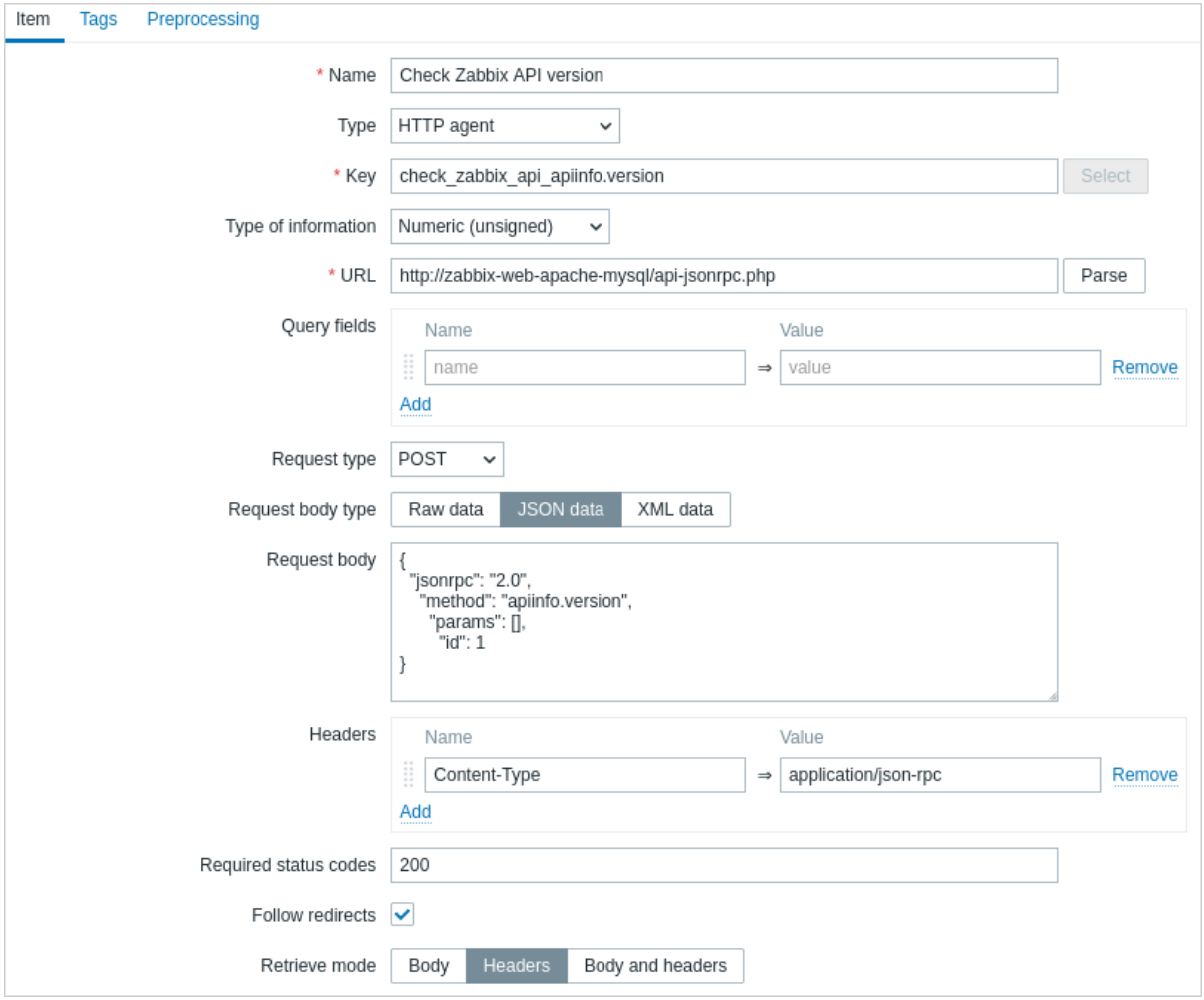
```
        "_shards": {
            "total": 5,
            "successful": 5,
            "skipped": 0,
            "failed": 0
        },
        "hits": {
            "total": 1,
            "max_score": 1.0,
            "hits": [{
                "_index": "dbl",
                "_type": "values",
                "_id": "dqX9VWEBV6sEKSMyk6sw",
                "_score": 1.0,
                "_source": {
                    "itemid": 28275,
                    "value": "0.138750",
                    "clock": 1517566136,
                    "ns": 25388713,
                    "ttl": 604800
                }
            }]
        }
    }
```


- Verwenden Sie nun einen JSONPath-Vorverarbeitungsschritt, um den Datenpunktwert zu ermitteln: `$.hits.hits[0]._source.value`

Beispiel 3

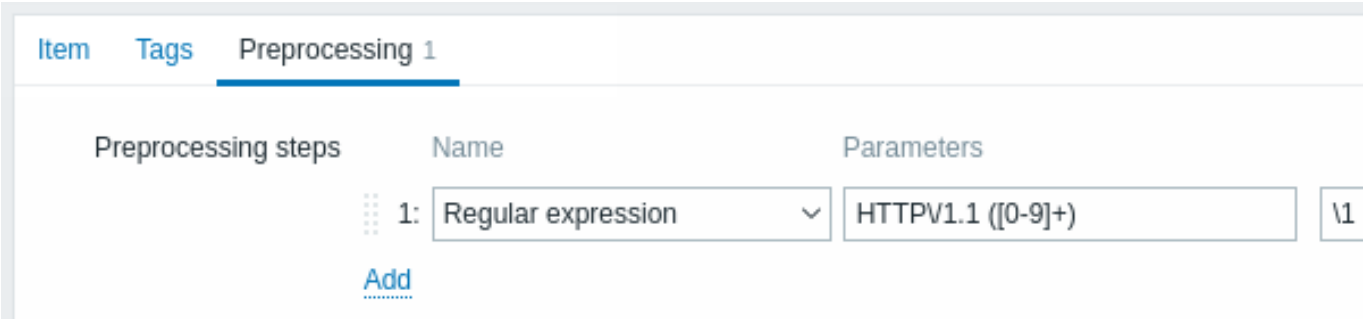
Prüfen, ob die Zabbix API aktiv ist, unter Verwendung von apiinfo.version.

- Datenpunktkonfiguration:

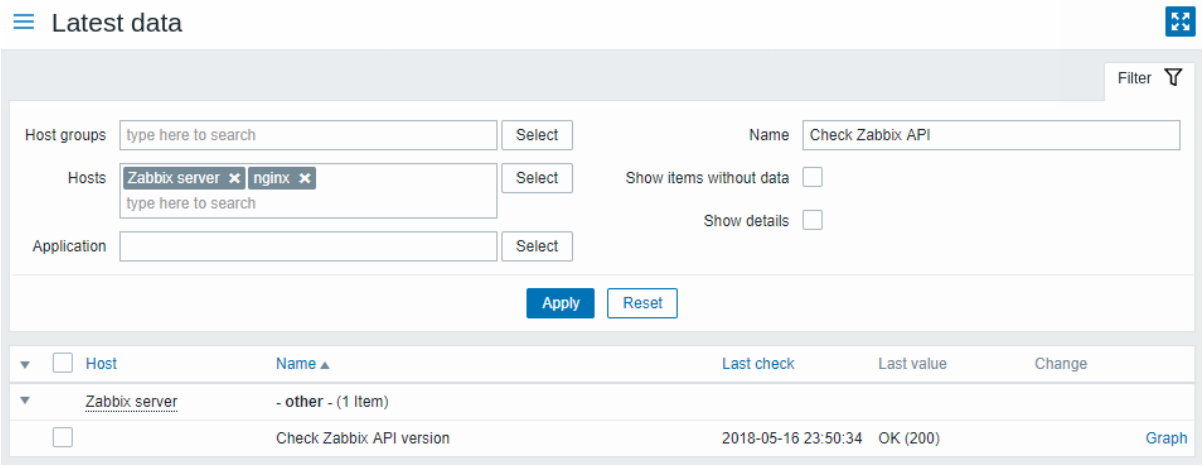


Beachten Sie die Verwendung der POST-Methode mit JSON-Daten, das Setzen von Request-Headern und die Anforderung, nur Header zurückzugeben:

- Datenpunkt-Vorverarbeitung mit regulärem Ausdruck, um den HTTP-Code zu ermitteln:



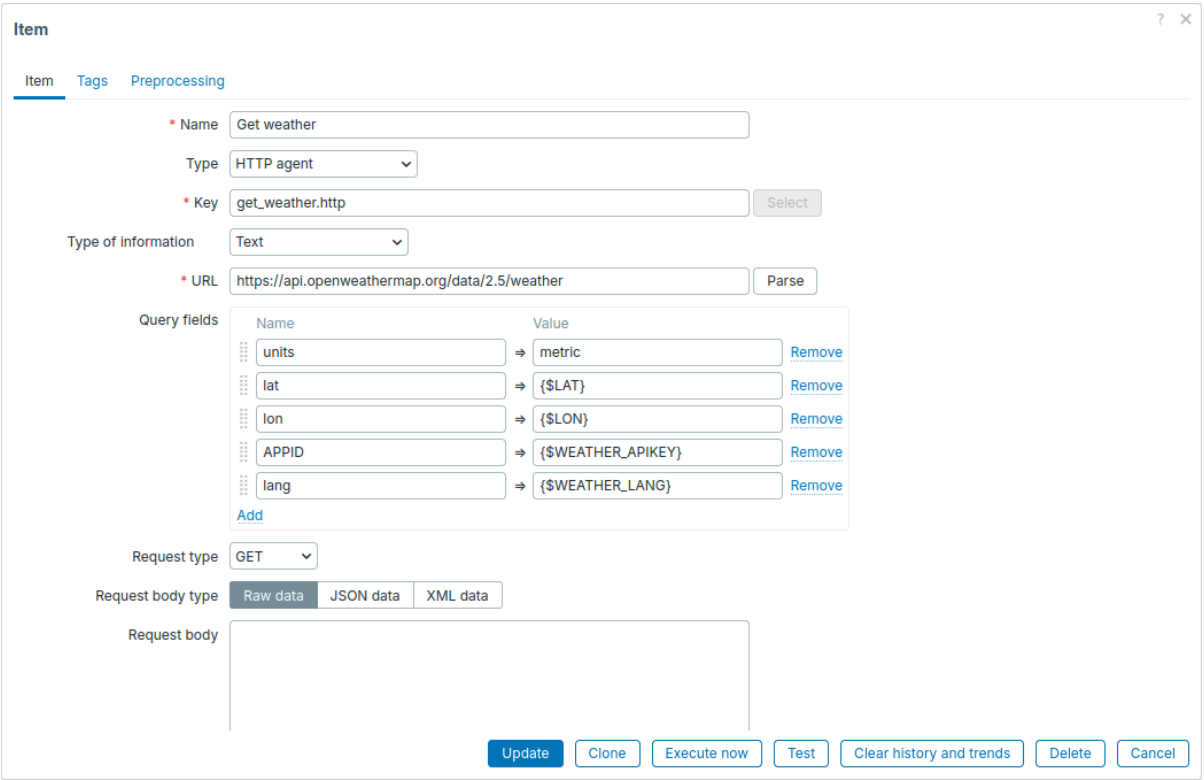
- Prüfen des Ergebnisses in *Latest data*:



Beispiel 4

Abrufen von Wetterinformationen durch Verbindung mit dem öffentlichen Openweathermap-Dienst.

- Konfigurieren Sie einen Master-Datenpunkt für die Massendatenerfassung in einem einzelnen JSON:



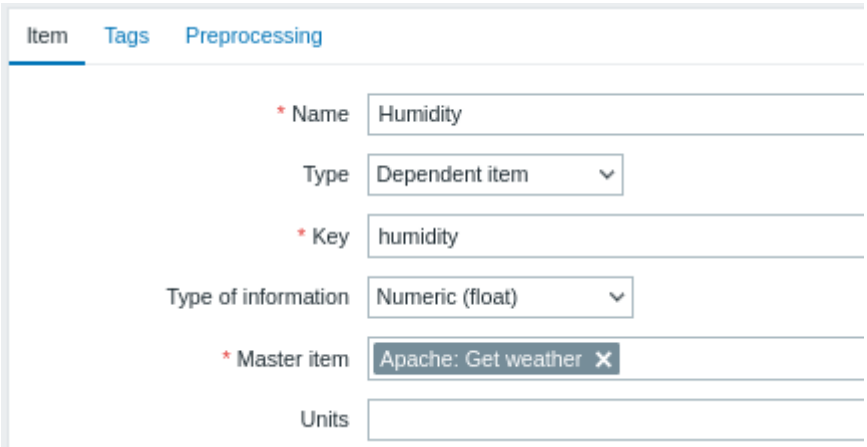
Beachten Sie die Verwendung von Makros in den Abfragefeldern. Informationen zum Ausfüllen finden Sie in der Openweathermap API.

Beispiel-JSON, das als Antwort vom HTTP-Agent zurückgegeben wird:


```
{
    "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
    }
}
```

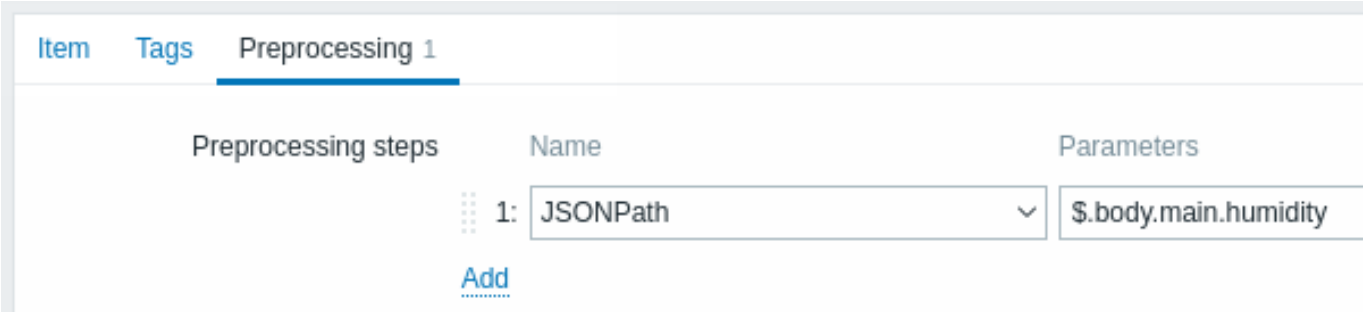

Die nächste Aufgabe besteht darin, abhängige Datenpunkte zu konfigurieren, die Daten aus dem JSON extrahieren.

- Konfigurieren Sie einen Beispiel-abhängigen Datenpunkt für die Luftfeuchtigkeit:

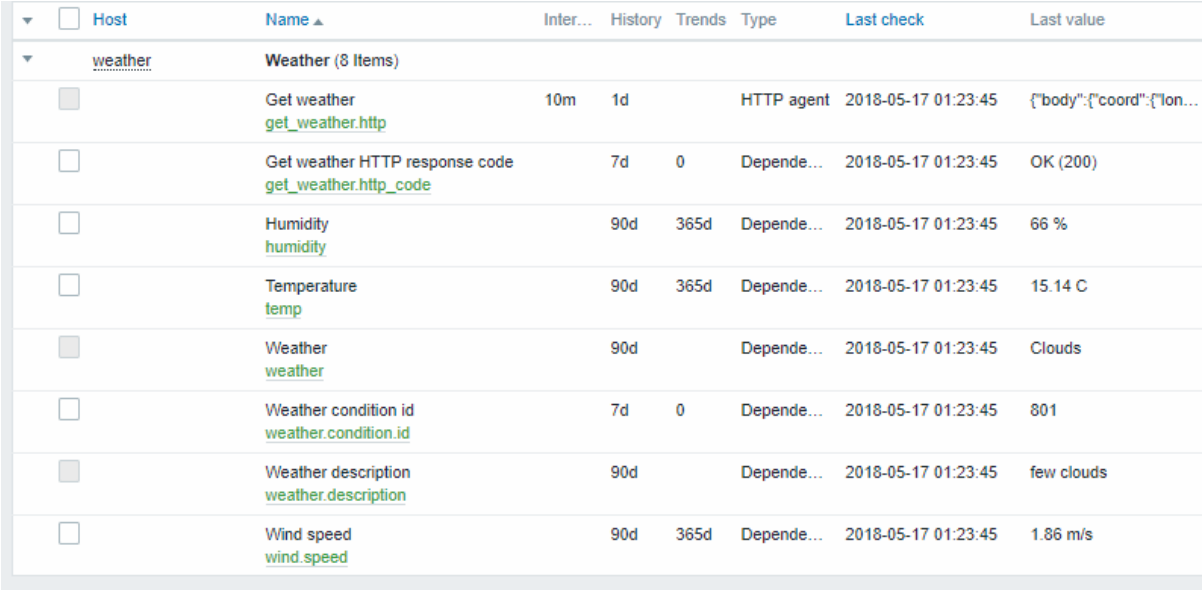


Andere Wettermetriken wie ”Temperatur” werden auf die gleiche Weise hinzugefügt.

- Beispiel für die Wertvorverarbeitung eines abhängigen Datenpunkts mit JSONPath:



- Prüfen Sie das Ergebnis der Wetterdaten unter *Neueste Daten*:

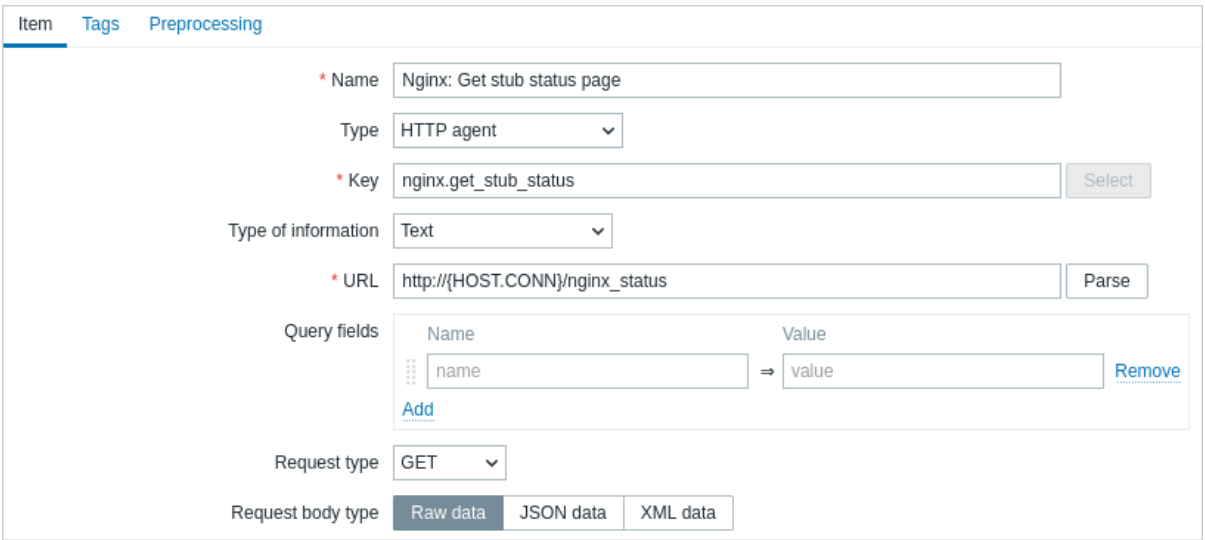


Host	Name ▲	Inter...	History	Trends	Type	Last check	Last value
weather	**Weather** (8 Items)						
	Get weather get_weather.http	10m	1d		HTTP agent	2018-05-17 01:23:45	{"body":{"coord":{"lon...
	Get weather HTTP response code get_weather.http_code		7d	0	Depende...	2018-05-17 01:23:45	OK (200)
	Humidity 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

Beispiel 5

Verbindung mit der NGINX-Statusseite und Abrufen ihrer Metriken in großen Mengen.

- Konfigurieren Sie NGINX gemäß der offiziellen Anleitung.
- Konfigurieren Sie einen Master-Datenpunkt für die Erfassung von Daten in großen Mengen:

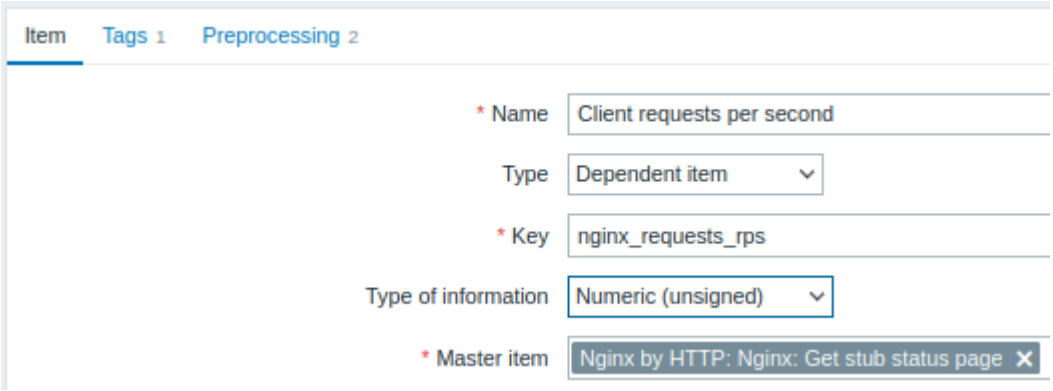


Beispielausgabe des NGINX-Stub-Status:

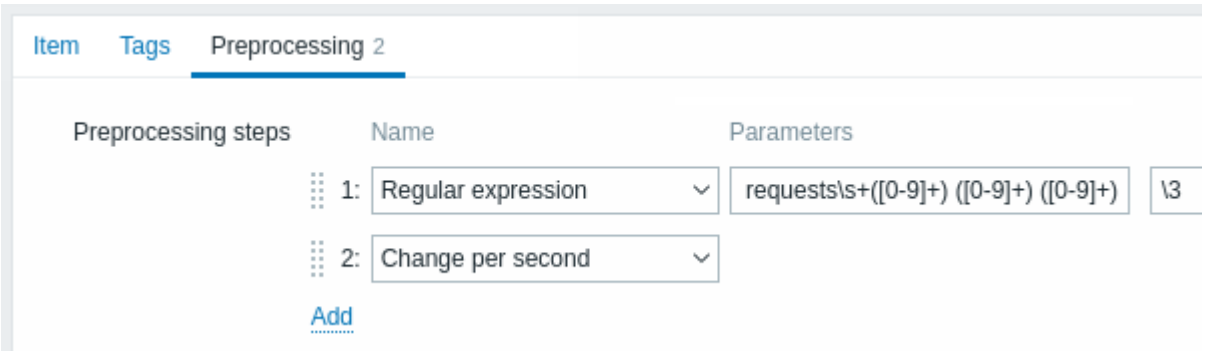
```
Active connections: 1 Active connections:
server accepts handled requests
 52 52 52
Reading: 0 Writing: 1 Waiting: 0
```

Als Nächstes müssen Sie abhängige Datenpunkte konfigurieren, die Daten extrahieren.

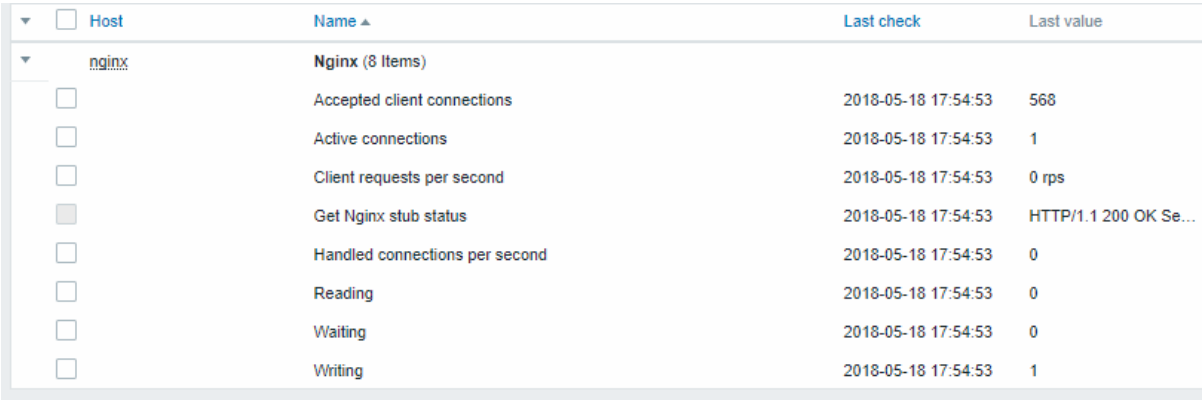
- Konfigurieren Sie einen beispielhaften abhängigen Datenpunkt für Anfragen pro Sekunde:



- Beispiel für die Vorverarbeitung des Werts eines abhängigen Datenpunkts mit dem regulären Ausdruck `server accepts handled requests\s+([0-9]+) ([0-9]+) ([0-9]+)`:



- Überprüfen Sie das vollständige Ergebnis des Stub-Moduls unter *Latest data*:



Prometheus-Prüfung

Übersicht

Zabbix kann Metriken abfragen, die im Prometheus-Zeilenformat bereitgestellt werden.

Zum Starten der Erfassung von Prometheus-Daten sind zwei Schritte erforderlich:

- ein HTTP-Master-Datenpunkt, der auf den entsprechenden Datenendpunkt verweist, z. B. `https://<prometheus host>/metrics`
- abhängige Datenpunkte, die eine Prometheus-Vorverarbeitungsoption verwenden, um die erforderlichen Daten aus den vom Master-Datenpunkt erfassten Metriken abzufragen

Es gibt zwei Prometheus-Datenvorverarbeitungsoptionen:

- *Prometheus pattern* - wird in normalen Datenpunkten verwendet, um Prometheus-Daten abzufragen
- *Prometheus to JSON* - wird in normalen Datenpunkten und für die Low-Level-Erkennung verwendet. In diesem Fall werden die abgefragten Prometheus-Daten im JSON-Format zurückgegeben.

Stapelverarbeitung

Die Stapelverarbeitung wird für abhängige Datenpunkte unterstützt. Um Caching und Indexierung zu aktivieren, muss die Vorverarbeitung *Prometheus pattern* der **erste** Vorverarbeitungsschritt sein. Wenn *Prometheus pattern* der erste Vorverarbeitungss-

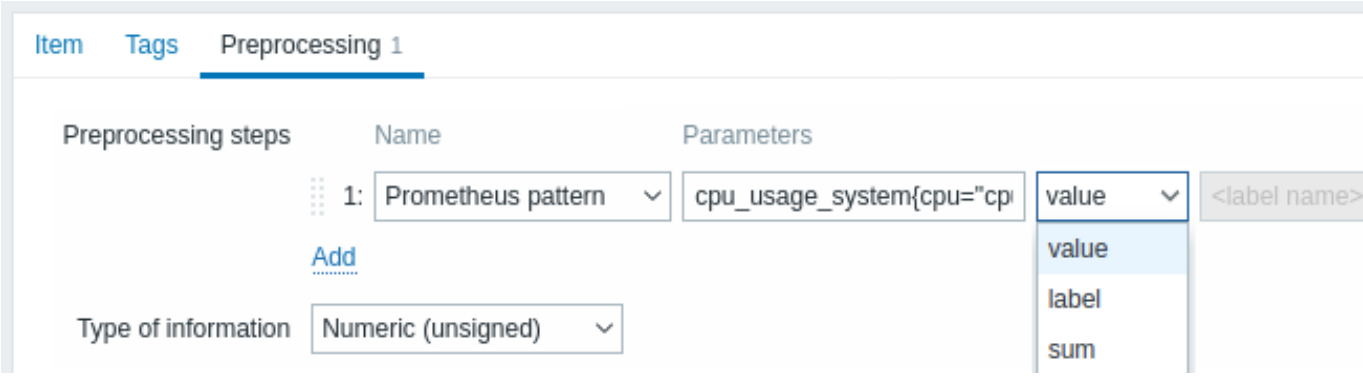
chritt ist, werden die geparsten Prometheus-Daten durch die erste `<label>==<value>`-Bedingung im Vorverarbeitungsschritt *Prometheus pattern* zwischengespeichert und indexiert. Dieser Cache wird bei der Verarbeitung anderer abhängiger Datenpunkte in diesem Batch wiederverwendet. Für eine optimale Leistung sollte das erste Label dasjenige mit den meisten unterschiedlichen Werten sein.

Wenn vor dem ersten Schritt noch weitere Vorverarbeitung durchgeführt werden muss, sollte sie entweder auf den Master-Datenpunkt oder auf einen neuen abhängigen Datenpunkt verschoben werden, der dann als Master-Datenpunkt für die abhängigen Datenpunkte verwendet wird.

Konfiguration

Vorausgesetzt, Sie haben den HTTP-Master-Datenpunkt konfiguriert, müssen Sie einen abhängigen Datenpunkt erstellen, der einen Prometheus-Vorverarbeitungsschritt verwendet:

1. Geben Sie die allgemeinen Parameter des abhängigen Datenpunkts im Konfigurationsformular ein.
2. Wechseln Sie zur Registerkarte *Vorverarbeitung*.
3. Wählen Sie eine Prometheus-Vorverarbeitungsoption (*Prometheus pattern* oder *Prometheus to JSON*).



Die folgenden Parameter gelten speziell für die Vorverarbeitungsoption *Prometheus pattern*:

Parameter	Beschreibung	Beispiele
Pattern	Um das erforderliche Datenmuster zu definieren, können Sie eine Abfragesprache verwenden, die der Prometheus-Abfragesprache ähnelt (siehe Vergleichstabelle), z. B.: <metric name> - Auswahl nach Metriknamen {__name__="<metric name>"} - Auswahl nach Metriknamen {__name__=~"<regex>"} - Auswahl nach Metriknamen, die einem regulären Ausdruck entsprechen {<label name>="<label value>",...} - Auswahl nach Labelnamen {<label name>=~"<regex>",...} - Auswahl nach Labelnamen, die einem regulären Ausdruck entsprechen {__name__=~".*"}==<value> - Auswahl nach Metrikwert Oder eine Kombination der oben genannten Möglichkeiten: <metric name>{<label1 name>="<label1 value>",<label2 name>=~"<regex>",...}==<value> Der Labelwert kann aus einer beliebigen Folge von UTF-8-Zeichen bestehen. Die Zeichen für umgekehrten Schrägstrich, doppelte Anführungszeichen und Zeilenumbruch müssen jedoch als \\, \" bzw. \n maskiert werden; andere Zeichen dürfen nicht maskiert werden.	*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*

Parameter	Beschreibung	Beispiele
Result processing	Geben Sie an, ob der Wert oder das Label zurückgegeben oder die entsprechende Funktion angewendet werden soll (wenn das Muster mehreren Zeilen entspricht und das Ergebnis aggregiert werden muss): **value** - Metrikwert zurückgeben (Fehler, wenn mehrere Zeilen übereinstimmen) **label** - Wert des im Feld *Label* angegebenen Labels zurückgeben (Fehler, wenn mehrere Metriken übereinstimmen) **sum** - Summe der Werte zurückgeben **min** - Minimalwert zurückgeben **max** - Maximalwert zurückgeben **avg** - Durchschnittswert zurückgeben **count** - Anzahl der Werte zurückgeben Dieses Feld ist nur für die Option *Prometheus pattern* verfügbar.	Siehe auch die folgenden Beispiele für die Verwendung der Parameter.
Output	Geben Sie den Labelnamen an (optional). In diesem Fall wird der dem Labelnamen entsprechende Wert zurückgegeben. Dieses Feld ist nur für die Option *Prometheus pattern* verfügbar, wenn im Feld *Result processing* die Option 'Label' ausgewählt ist.	

Beispiele für die Verwendung der Parameter

1. Der häufigste Anwendungsfall ist die Rückgabe des **Werts**. Um den Wert von `/var/db` aus folgenden Daten zurückzugeben:

   `node_disk_usage_bytes{path="/var/cache"} 2.1766144e+09<br> node_disk_usage_bytes{path="/var/db"} 20480<br> node_disk_usage_bytes{path="/var/dpkg"} 8192<br> node_disk_usage_bytes{path="/var/empty"} 4096`

verwenden Sie die folgenden Parameter:

- *Pattern* - `node_disk_usage_bytes{path="/var/db"}`
- *Result processing* - 'value' auswählen

2. Möglicherweise interessieren Sie sich auch für den **Durchschnittswert** aller `node_disk_usage_bytes`-Parameter:

- *Pattern* - `node_disk_usage_bytes`
- *Result processing* - 'avg' auswählen

3. Obwohl Prometheus nur numerische Daten unterstützt, wird häufig eine Behelfslösung verwendet, um zusätzlich die relevante Textbeschreibung zurückzugeben. Dies kann durch einen Filter und die Angabe des Labels erreicht werden. Um den Wert des Labels 'color' aus folgenden Daten zurückzugeben:

   `elasticsearch_cluster_health_status{cluster="elasticsearch",color="green"} 1<br> elasticsearch_cluster 0<br> elasticsearch_cluster_health_status{cluster="elasticsearch",color="yellow"} 0`

verwenden Sie die folgenden Parameter:

- *Pattern* - `elasticsearch_cluster_health_status {cluster="elasticsearch"} == 1`
- *Result processing* - 'label' auswählen
- *Label* - 'color' angeben

Der Filter (basierend auf dem numerischen Wert '1') findet die entsprechende Zeile, während das Label die Beschreibung des Systemzustands zurückgibt (derzeit 'green', möglicherweise aber auch 'red' oder 'yellow').

Prometheus zu JSON

Daten aus Prometheus können für Low-Level-Discovery verwendet werden. In diesem Fall werden Daten im JSON-Format benötigt, und die Vorverarbeitungsoption *Prometheus to JSON* liefert genau das.

Weitere Details finden Sie unter Discovery using Prometheus data.

Vergleich der Abfragesprache

Die folgende Tabelle listet Unterschiede und Gemeinsamkeiten zwischen PromQL und der Zabbix-Preprocessing-Abfragesprache für Prometheus auf.

PromQL instant vector selector		Zabbix Prometheus preprocessing
Unterschiede		
Abfrageziel	Prometheus Server	Klartext im Prometheus-Expositionsformat
Rückgabewert	Instant vector	Metrik- oder Labelwert (Prometheus-Muster) Array von Metriken für einen einzelnen Wert in JSON (Prometheus to JSON)
Label-Matching-Operatoren	**=**, **!=**, **=~**, **!~**	**=**, **!=**, **=~**, **!~**
Für den Abgleich von Label- oder Metriknamen verwendeter regulärer Ausdruck	RE2	PCRE
Vergleichsoperatoren	Siehe Liste	Für die Wertfilterung wird nur **==** (gleich) unterstützt
Gemeinsamkeiten		
Auswahl nach Metriknamen, der einer Zeichenfolge entspricht	<metric name> oder {__name__=”<metric name>”}	<metric name> oder {__name__=”<metric name>”}
Auswahl nach Metriknamen, der einem regulären Ausdruck entspricht	{__name__=~”<regex>”}	{__name__=~”<regex>”}
Auswahl nach Wert von <label name>, der einer Zeichenfolge entspricht	{<label name>=”<label value>”,...}	{<label name>=”<label value>”,...}
Auswahl nach Wert von <label name>, der einem regulären Ausdruck entspricht	{<label name>=~”<regex>”,...}	{<label name>=~”<regex>”,...}
Auswahl nach Wert, der einer Zeichenfolge entspricht	{__name__=~”.*”} == <value>	{__name__=~”.*”} == <value>

IPMI Agent

Übersicht

Sie können den Zustand und die Verfügbarkeit von Geräten mit Intelligent Platform Management Interface (IPMI) in Zabbix überwachen. Für die Durchführung von IPMI-Prüfungen muss der Zabbix Server zunächst mit IPMI-Unterstützung konfiguriert werden.

IPMI ist eine standardisierte Schnittstelle für die Remoteverwaltung von Computersystemen im sogenannten „Lights-out“- oder „Out-of-Band“-Modus. Sie ermöglicht die direkte Überwachung des Hardwarestatus über sogenannte „Out-of-Band“-Managementkarten, unabhängig vom Betriebssystem und davon, ob das Gerät überhaupt eingeschaltet ist.

Die Zabbix-IPMI-Überwachung funktioniert nur für Geräte mit IPMI-Unterstützung (HP iLO, DELL DRAC, IBM RSA, Sun SSP usw.).

Ein IPMI-Manager-Prozess plant die IPMI-Prüfungen über IPMI-Poller. Ein Host wird immer nur von einem IPMI-Poller gleichzeitig abgefragt, wodurch die Anzahl der offenen Verbindungen zu BMC-Controllern reduziert wird. Daher ist es sicher, die Anzahl der IPMI-Poller zu erhöhen, ohne eine Überlastung des BMC-Controllers befürchten zu müssen. Der IPMI-Manager-Prozess wird automatisch gestartet, sobald mindestens ein IPMI-Poller gestartet wurde.

Weitere Informationen zu IPMI-Prüfungen finden Sie unter Bekannte Probleme.

Konfiguration

Host-Konfiguration

Ein Host muss konfiguriert werden, um IPMI-Prüfungen zu verarbeiten. Eine IPMI-Schnittstelle muss hinzugefügt werden, mit den entsprechenden IP- und Portnummern, und die IPMI-Authentifizierungsparameter müssen definiert werden.

Weitere Einzelheiten finden Sie unter Konfiguration von Hosts.

Server-Konfiguration

Standardmäßig ist der Zabbix-Server nicht so konfiguriert, dass er IPMI-Poller startet. Daher funktionieren alle hinzugefügten IPMI-Datenpunkte nicht. Um dies zu ändern, öffnen Sie die Konfigurationsdatei des Zabbix-Servers (`zabbix_server.conf`) als root und suchen Sie nach der folgenden Zeile:

```
# StartIPMIPollers=0
```

Entfernen Sie das Kommentarzeichen und setzen Sie die Anzahl der Poller beispielsweise auf 3, sodass die Zeile wie folgt aussieht:

```
StartIPMIPollers=3
```

Speichern Sie die Datei und starten Sie zabbix_server anschließend neu.

Datenpunkt-Konfiguration

Beim Konfigurieren eines Datenpunkts auf Host-Ebene:

- Wählen Sie „IPMI agent“ als *Typ* aus.
- Geben Sie einen Schlüssel für den Datenpunkt ein, der innerhalb des Hosts eindeutig ist (zum Beispiel `ipmi.fan.rpm`).
- Wählen Sie für *Host-Schnittstelle* die entsprechende IPMI-Schnittstelle (IP und Port) aus. Beachten Sie, dass auf dem Host eine IPMI-Schnittstelle vorhanden sein muss.
- Geben Sie den *IPMI-Sensor* an (zum Beispiel „FAN MOD 1A RPM“ auf Dell Poweredge), von dem der Messwert abgerufen werden soll. Standardmäßig sollte die Sensor-ID angegeben werden. Es ist auch möglich, Präfixe vor dem Wert zu verwenden:
  - `id:` - zur Angabe der Sensor-ID;
  - `name:` - zur Angabe des vollständigen Sensornamens. Dies kann in Situationen nützlich sein, in denen Sensoren nur durch Angabe des vollständigen Namens unterschieden werden können.
- Wählen Sie den entsprechenden Informationstyp aus („Numerisch (Gleitkommazahl)“ in diesem Fall; bei diskreten Sensoren „Numerisch (Ganzzahl ohne Vorzeichen)“), die Einheiten (höchstwahrscheinlich `rpm`) sowie alle anderen erforderlichen Datenpunkt-Attribute aus.

Unterstützte Prüfungen

Der IPMI-Agent unterstützt den integrierten Datenpunkt **ipmi.get**, der Informationen zu IPMI-Sensoren zurückgibt und für die Discovery von IPMI-Sensoren verwendet werden kann.
 Rückgabewert: *JSON-Objekt*

Timeout und Sitzungsbeendigung

Zeitüberschreitungen für IPMI-Nachrichten und die Anzahl der Wiederholungsversuche sind in der OpenIPMI-Bibliothek definiert. Aufgrund des aktuellen Designs von OpenIPMI ist es nicht möglich, diese Werte in Zabbix zu konfigurieren, weder auf Schnittstellen- noch auf Datenpunkt-Ebene.

Das Inaktivitäts-Timeout einer IPMI-Sitzung für LAN beträgt 60 +/-3 Sekunden. Derzeit ist es nicht möglich, mit OpenIPMI das periodische Senden des Befehls „Activate Session“ zu implementieren. Wenn es von Zabbix an einen bestimmten BMC länger als das im BMC konfigurierte Sitzungs-Timeout keine Prüfungen von IPMI-Datenpunkten gibt, dann wird die nächste IPMI-Prüfung nach Ablauf des Timeouts aufgrund individueller Nachrichten-Timeouts, Wiederholungsversuche oder Empfangsfehler mit einer Zeitüberschreitung fehlschlagen. Danach wird eine neue Sitzung geöffnet und ein vollständiger erneuter Scan des BMC wird gestartet. Es kann ein neuer UDP- Port geöffnet werden, um die neue Sitzung zu verwalten. Inaktivität ist definiert als das Ausbleiben sowohl ausgehender Anfragen als auch eingehender Antworten. Wenn Sie unnötige erneute Scans des BMC vermeiden möchten, wird empfohlen, das Abfrageintervall des IPMI- Datenpunkts unterhalb des im BMC konfigurierten Inaktivitäts-Timeouts der IPMI-Sitzung festzulegen.

Hinweise zu diskreten IPMI-Sensoren

Um Sensoren auf einem Host zu finden, starten Sie den Zabbix Server mit aktiviertem **DebugLevel=4**. Warten Sie einige Minuten und suchen Sie dann im Logfile des Zabbix Server nach Einträgen zur Sensorerkennung:

```
$ 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
```

Zum Dekodieren von IPMI-Sensortypen und -zuständen ist eine Kopie der IPMI-2.0-Spezifikation verfügbar (bitte beachten Sie, dass keine weiteren Aktualisierungen der IPMI-Spezifikation geplant sind).

Der erste Parameter, mit dem Sie beginnen sollten, ist ”reading_type”. Verwenden Sie zum Dekodieren des Codes ”reading_type” die ”Table 42-1, Event/Reading Type Code Ranges” aus der Spezifikation. Die meisten Sensoren in unserem Beispiel haben ”reading_type:0x1”, was einen ”threshold”-Sensor bedeutet. ”Table 42-3, Sensor Type Codes” zeigt, dass ”type:0x1” einen Temperatursensor bedeutet, ”type:0x2” einen Spannungssensor, ”type:0x4” einen Lüftersensor usw. Threshold-Sensoren werden manchmal auch als ”analoge” Sensoren bezeichnet, da sie kontinuierliche Parameter wie Temperatur, Spannung oder Umdrehungen pro Minute messen.

Ein weiteres Beispiel ist ein Sensor mit ”reading_type:0x3”. ”Table 42-1, Event/Reading Type Code Ranges” besagt, dass Reading-Type-Codes von 02h bis 0Ch einen ”Generic Discrete”-Sensor bedeuten. Diskrete Sensoren haben bis zu 15 mögliche Zustände (mit anderen Worten: bis zu 15 bedeutungsvolle Bits). Für den Sensor ’CATERR’ mit ”type:0x7” zeigt ”Table 42-3, Sensor Type Codes” beispielsweise, dass dieser Typ ”Processor” bedeutet und die einzelnen Bits folgende Bedeutung haben: 00h (das niederwertigste Bit) - IERR, 01h - Thermal Trip usw.

In unserem Beispiel gibt es einige Sensoren mit ”reading_type:0x6f”. Für diese Sensoren empfiehlt ”Table 42-1, Event/Reading Type Code Ranges”, zum Dekodieren der Bitbedeutungen ”Table 42-3, Sensor Type Codes” zu verwenden. Beispielsweise hat der Sensor ’Power Unit Stat’ den Typ ”type:0x9”, was ”Power Unit” bedeutet. Offset 00h bedeutet ”PowerOff/Power Down”. Mit anderen Worten: Wenn das niederwertigste Bit 1 ist, dann ist der Server ausgeschaltet. Um dieses Bit zu prüfen, kann die Funktion `bitand` mit der Maske ’1’ verwendet werden. Der Auslöser-Ausdruck könnte wie folgt aussehen:

```
bitand(last(/www.example.com/Power Unit Stat,#1),1)=1
```

um vor einem ausgeschalteten Server zu warnen.

Hinweise zu Namen diskreter Sensoren in OpenIPMI-2.0.16, 2.0.17, 2.0.18 und 2.0.19

Namen diskreter Sensoren in OpenIPMI-2.0.16, 2.0.17 und 2.0.18 haben am Ende oft zusätzlich eine „0“ (oder eine andere Ziffer bzw. einen anderen Buchstaben) angehängt. Während beispielsweise `ipmitool` und OpenIPMI-2.0.19 die Sensornamen als „PhysicalSecurity“ oder „CATERR“ anzeigen, lauten die Namen in OpenIPMI-2.0.16, 2.0.17 und 2.0.18 entsprechend „PhysicalSecurity0“ bzw. „CATERR0“.

Wenn Sie einen IPMI-Datenpunkt mit dem Zabbix Server unter Verwendung von OpenIPMI-2.0.16, 2.0.17 und 2.0.18 konfigurieren, verwenden Sie diese auf „0“ endenden Namen im Feld *IPMI sensor* von IPMI-Agent-Datenpunkten. Wenn Ihr Zabbix Server auf eine neue Linux-Distribution aktualisiert wird, die OpenIPMI-2.0.19 (oder neuer) verwendet, werden Datenpunkte mit diesen diskreten IPMI-Sensoren zu „NOT SUPPORTED“. Sie müssen deren Namen im Feld *IPMI sensor* ändern (die „0“ am Ende entfernen) und некоторое время warten, bevor sie wieder zu „Enabled“ werden.

Hinweise zur gleichzeitigen Verfügbarkeit von Schwellenwert- und diskreten Sensoren

Einige IPMI-Agenten stellen sowohl einen Schwellenwertsensor als auch einen diskreten Sensor unter demselben Namen bereit. Dem Schwellenwertsensor wird immer der Vorzug gegeben.

Hinweise zur Beendigung der Verbindung

Wenn keine IPMI-Prüfungen durchgeführt werden (aus beliebigen Gründen: alle IPMI-Datenpunkte des Hosts deaktiviert/nicht unterstützt, Host deaktiviert/gelöscht, Host in Wartung usw.), wird die IPMI-Verbindung vom Zabbix Server oder Proxy innerhalb von 3 bis 4 Stunden beendet, abhängig davon, wann der Zabbix Server/Proxy gestartet wurde.

SSH Agent

Übersicht

SSH-Prüfungen werden als agentenlose Überwachung durchgeführt. Der Zabbix Agent wird für SSH-Prüfungen nicht benötigt.

Um SSH-Prüfungen durchzuführen, muss der Zabbix Server zunächst mit SSH2-Unterstützung (libssh oder libssh2) konfiguriert werden. Siehe auch: Anforderungen.

> **Attention:**
> Ab RHEL 8 wird nur libssh unterstützt. Für andere Distributionen wird libssh gegenüber libssh2 empfohlen.

Konfiguration

Authentifizierung mit Passphrase

SSH-Prüfungen bieten zwei Authentifizierungsmethoden – ein Benutzer/Passwort-Paar und eine auf Schlüsseldateien basierende Methode.

Wenn Sie keine Schlüssel verwenden möchten, ist keine zusätzliche Konfiguration erforderlich, abgesehen davon, libssh oder libssh2 mit Zabbix zu verknüpfen, falls Sie Zabbix aus dem Quellcode erstellen.

Authentifizierung mit Schlüsseldatei

Um eine schlüsselbasierte Authentifizierung für SSH-Datenpunkte zu verwenden, sind bestimmte Änderungen an der Serverkonfiguration erforderlich.

Öffnen Sie die Konfigurationsdatei des Zabbix-Servers (`zabbix_server.conf`) als `root` und suchen Sie nach der folgenden Zeile:

```
##### SSHKeyLocation=
```

Kommentieren Sie sie aus und setzen Sie den vollständigen Pfad zu dem Ordner, in dem sich die öffentlichen und privaten Schlüssel befinden werden:

```
SSHKeyLocation=/home/zabbix/.ssh
```

Speichern Sie die Datei und starten Sie anschließend den Zabbix-Server neu.

Der Pfad `/home/zabbix` ist hier das Home-Verzeichnis des Benutzerkontos *zabbix*, und `.ssh` ist ein Verzeichnis, in dem standardmäßig öffentliche und private Schlüssel durch den Befehl ssh-keygen innerhalb des Home-Verzeichnisses erzeugt werden.

Üblicherweise legen die Installationspakete des Zabbix-Servers aus verschiedenen OS-Distributionen das Benutzerkonto *zabbix* mit einem Home-Verzeichnis an einem anderen Ort an, zum Beispiel `/var/lib/zabbix` (wie bei Systemkonten).

Vor dem Generieren der Schlüssel könnten Sie das Home-Verzeichnis nach `/home/zabbix` verlegen, damit es mit dem oben genannten Konfigurationsparameter `SSHKeyLocation` des Zabbix-Servers übereinstimmt.

Note:
Die folgenden Schritte können übersprungen werden, wenn das Konto *zabbix* manuell gemäß dem Installationsabschnitt hinzugefügt wurde. In diesem Fall ist das Home-Verzeichnis des Kontos *zabbix* höchstwahrscheinlich bereits `/home/zabbix`.

Um das Home-Verzeichnis des Benutzerkontos *zabbix* zu ändern, müssen alle laufenden Prozesse, die es verwenden, beendet werden:

```
systemctl stop zabbix-agent
systemctl stop zabbix-server
```

Um den Speicherort des Home-Verzeichnisses zu ändern und dabei zu versuchen, es zu verschieben (falls es existiert), sollte der folgende Befehl ausgeführt werden:

```
usermod -m -d /home/zabbix zabbix
```

Es ist auch möglich, dass im alten Speicherort kein Home-Verzeichnis vorhanden war; in diesem Fall sollte es am neuen Speicherort erstellt werden. Ein sicherer Versuch dafür ist:

```
test -d /home/zabbix || mkdir /home/zabbix
```

Um sicherzustellen, dass alles abgesichert ist, können zusätzliche Befehle ausgeführt werden, um die Berechtigungen für das Home-Verzeichnis festzulegen:

```
chown zabbix:zabbix /home/zabbix
chmod 700 /home/zabbix
```

Die zuvor gestoppten Prozesse können nun wieder gestartet werden:

```
systemctl start zabbix-agent
systemctl start zabbix-server
```

Nun können die Schritte zum Generieren der öffentlichen und privaten Schlüssel mit den folgenden Befehlen ausgeführt werden (zur besseren Lesbarkeit sind die Eingabeaufforderungen auskommentiert):

```
sudo -u zabbix ssh-keygen -t rsa
##### Generating public/private rsa key pair.
##### Enter file in which to save the key (/home/zabbix/.ssh/id_rsa):
/home/zabbix/.ssh/id_rsa
##### Enter passphrase (empty for no passphrase):
<Leave empty>
##### Enter same passphrase again:
<Leave empty>
##### Your identification has been saved in /home/zabbix/.ssh/id_rsa.
##### Your public key has been saved in /home/zabbix/.ssh/id_rsa.pub.
##### The key fingerprint is:
##### 90:af:e4:c7:e3:f0:2e:5a:8d:ab:48:a2:0c:92:30:b9 zabbix@it0
##### The key's randomart image is:
```

```
##### +--[ RSA 2048]----+
##### |                 |
##### |       .         |
##### |      o          |
##### | .     o         |
##### |+     . S        |
##### |.+   o =         |
##### |E .   * =        |
##### |=o . ..* .       |
##### |... oo.o+        |
##### +-----------------+
```

Note:
Die öffentlichen und privaten Schlüssel (`id_rsa.pub` und `id_rsa`) wurden standardmäßig im Verzeichnis `/home/zabbix/.ssh` generiert, das dem Konfigurationsparameter `SSHKeyLocation` des Zabbix-Servers entspricht.

Attention:
Schlüsseltypen außer "rsa" werden möglicherweise vom ssh-keygen-Tool und von SSH-Servern unterstützt, jedoch eventuell nicht von libssh2, das von Zabbix verwendet wird.

Shell-Konfigurationsformular

Dieser Schritt sollte nur einmal für jeden Host durchgeführt werden, der durch SSH-Prüfungen überwacht wird.

Mit den folgenden Befehlen kann die **öffentliche** Schlüsseldatei auf einem entfernten Host *10.10.10.10* installiert werden, sodass die SSH-Prüfungen mit einem *root*-Konto durchgeführt werden können (zur besseren Lesbarkeit sind die Befehlsaufforderungen auskommentiert):

```
sudo -u zabbix ssh-copy-id root@10.10.10.10
##### The authenticity of host '10.10.10.10 (10.10.10.10)' can't be established.
##### RSA key fingerprint is 38:ba:f2:a4:b5:d9:8f:52:00:09:f7:1f:75:cc:0b:46.
##### Are you sure you want to continue connecting (yes/no)?
yes
##### Warning: Permanently added '10.10.10.10' (RSA) to the list of known hosts.
##### root@10.10.10.10's password:
<Enter root password>
##### Now try logging into the machine, with "ssh 'root@10.10.10.10'",
##### and check to make sure that only the key(s) you wanted were added.
```

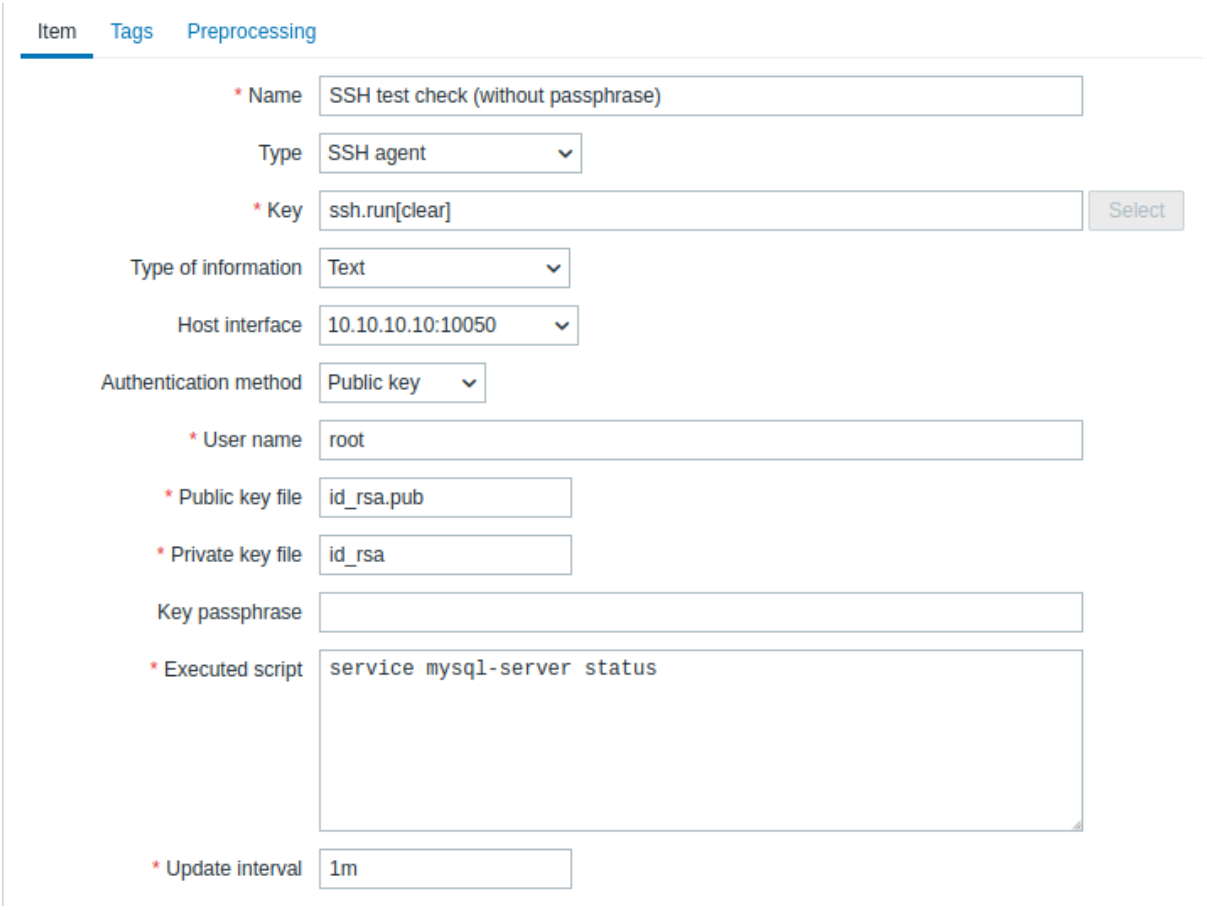
Nun ist es möglich, die SSH-Anmeldung mit dem standardmäßigen privaten Schlüssel (`/home/zabbix/.ssh/id_rsa`) für das *zabbix*-Benutzerkonto zu prüfen:

```
sudo -u zabbix ssh root@10.10.10.10
```

Wenn die Anmeldung erfolgreich ist, ist der Konfigurationsabschnitt in der Shell abgeschlossen und die entfernte SSH-Sitzung kann beendet werden.

Datenpunkt-Konfiguration

Die tatsächlich auszuführenden Befehle müssen im Feld *Ausgeführtes Skript* in der Datenpunkt-Konfiguration angegeben werden. Mehrere Befehle können nacheinander ausgeführt werden, indem sie in jeweils eine neue Zeile geschrieben werden. In diesem Fall werden auch die zurückgegebenen Werte mehrzeilig formatiert.



Alle Pflichtfelder sind mit einem roten Sternchen gekennzeichnet.

Die Felder, die spezifische Informationen für SSH-Datenpunkte erfordern, sind:

Parameter	Beschreibung	Kommentare
Typ	Wählen Sie hier **SSH agent** aus.	

Parameter	Beschreibung	Kommentare

Schlüssel	Eindeutiger (pro Host) Datenpunktschlüssel im Format `ssh.run[unique short description,<ip>,<port>,<encoding>,<ssh options>,<subsystem>]`	**unique short description** ist erforderlich und sollte für jeden SSH-Datenpunkt pro Host eindeutig sein.
		
		Der Standardport ist 22, nicht der Port, der in der Schnittstelle angegeben ist, der dieser Datenpunkt zugewiesen ist.
		
		Mit **ssh options** können zusätzliche SSH-Optionen im Format `key1=value1;key2=value2,value3` übergeben werden. Mehrere Werte für einen Schlüssel können durch Kommas getrennt übergeben werden (in diesem Fall muss der Parameter in Anführungszeichen gesetzt werden); mehrere Optionsschlüssel können durch Semikolons getrennt übergeben werden.
		
		Die folgenden Optionsschlüssel werden unterstützt: `KexAlgorithms`, `HostkeyAlgorithms`, `Ciphers`, `MACs`, `PubkeyAcceptedKeyTypes`. Die Unterstützung von Optionsschlüsseln und -werten hängt von der SSH-Bibliothek ab (beispielsweise wird `PubkeyAcceptedKeyTypes` nur mit libssh unterstützt); wenn eine Option nicht unterstützt wird, wird ein Fehler zurückgegeben und der Datenpunkt wird nicht unterstützt.
		
		Beachten Sie, dass das Zeichen "+" zum Anhängen von Chiffreeinstellungen und "!" zum Deaktivieren bestimmter Chiffreeinstellungen (wie in GnuTLS und OpenSSL) nicht unterstützt werden.
		
		Beispiele:
		=>
		`ssh.run[KexAlgorithms,127.0.0.1,,,Ciphers=aes128-`
		=>
		`ssh.run[KexAlgorithms,,,,"KexAlgorithms=diffie-he`
		=>
		`ssh.run[PubkeyAcceptedKeyTypes,127.0.0.1,,,Pubkey`
		
		Mit **subsystem** kann ein SSH-Subsystem übergeben werden, wodurch die SSH-Verbindung auf die vom Subsystem erlaubten spezifischen Vorgänge beschränkt wird (z. B. Dateiübertragungen mit SFTP oder die Verwaltung von Netzwerkgeräten mit NETCONF). Beachten Sie, dass die Verwendung eines Subsystems möglicherweise auch eine bestimmte Skriptsyntax im Parameter *Ausgeführtes Skript* erfordert.
		
		Beispiele:
		=>
		`ssh.run[SFTPBackup,192.0.2.18,,,,sftp]`
		=>
		`ssh.run[Cisco1234,192.0.2.18,,,,netconf]`

Authentifizierungsmethode	Eine der Optionen "Passwort" oder "Öffentlicher Schlüssel".	
Benutzername	Benutzername (bis zu 255 Zeichen) zur Authentifizierung auf dem Remote-Host. Erforderlich.	

Parameter	Beschreibung	Kommentare
Datei des öffentlichen Schlüssels	Dateiname des öffentlichen Schlüssels, wenn als *Authentifizierungsmethode* "Öffentlicher Schlüssel" ausgewählt ist. Erforderlich.	Beispiel: *id_rsa.pub* - Standarddateiname des öffentlichen Schlüssels, der mit dem Befehl ssh-keygen generiert wird.
Datei des privaten Schlüssels	Dateiname des privaten Schlüssels, wenn als *Authentifizierungsmethode* "Öffentlicher Schlüssel" ausgewählt ist. Erforderlich.	Beispiel: *id_rsa* - Standarddateiname des privaten Schlüssels.
Passwort oder *Passphrase des Schlüssels*	Passwort (bis zu 255 Zeichen) zur Authentifizierung oder Passphrase, **falls** sie für den privaten Schlüssel verwendet wurde.	Lassen Sie das Feld *Passphrase des Schlüssels* leer, wenn keine Passphrase verwendet wurde. Siehe auch bekannte Probleme bezüglich der Verwendung von Passphrasen.

Ausgeführtes Skript	Ausgeführte Shell-Befehle unter Verwendung einer entfernten SSH-Sitzung.	Der Rückgabewert der ausgeführten Shell-Befehle ist auf 16 MB begrenzt (einschließlich nachfolgender Leerzeichen, die abgeschnitten werden); Datenbanklimits gelten ebenfalls.
		
		Beachten Sie, dass die libssh2-Bibliothek ausführbare Skripte möglicherweise auf etwa 32 kB kürzt.
		
		Beispiele:
		`date +%s`
		`systemctl status mysql-server`
		`ps auxww | grep httpd | wc -l`
		
		Beispiel (für das NETCONF-Subsystem):
		`<rpc>`
		`<get-software-information/>`
		`</rpc>`
		`]]>]]>`
		`<rpc>`
		`<close-session/>`
		`</rpc>`
		`]]>]]>`

Telnet Agent

Übersicht

Telnet-Prüfungen werden als Agent-loses Monitoring durchgeführt. Der Zabbix Agent wird für Telnet-Prüfungen nicht benötigt.

Konfigurierbare Felder

Die tatsächlich auszuführenden Befehle müssen im Feld **Executed script** in der Datenpunkt-Konfiguration eingetragen werden.
 Mehrere Befehle können nacheinander ausgeführt werden, indem sie jeweils in eine neue Zeile geschrieben werden. In diesem Fall wird auch der zurückgegebene Wert mehrzeilig formatiert.

Unterstützte Zeichen, mit denen die Shell-Eingabeaufforderung enden kann:

- $
- #
- >
- %

Note:
Eine Telnet-Eingabeaufforderungszeile, die mit einem dieser Zeichen endet, wird aus dem zurückgegebenen Wert entfernt, jedoch nur für den ersten Befehl in der Befehlsliste, d. h. nur zu Beginn der Telnet-Sitzung.

Key	Description
telnet.run[<unique short description>,<ip>,<port>,<encoding>]	Einen Befehl auf einem entfernten Gerät über eine Telnet-Verbindung ausführen

Attention:
Wenn eine Telnet-Prüfung einen Wert mit Nicht-ASCII-Zeichen und in einer Nicht-UTF8-Kodierung zurückgibt, dann sollte der Parameter <*encoding*> des Schlüssels korrekt angegeben werden. Weitere Details finden Sie auf der Seite encoding of returned values.

JMX Agent

Übersicht

JMX-Monitoring kann verwendet werden, um JMX-Zähler einer Java-Anwendung zu überwachen.

JMX-Monitoring wird in Zabbix nativ unterstützt, und zwar in Form eines Zabbix-Daemons namens „Zabbix Java gateway“.

Um den Wert eines bestimmten JMX-Zählers auf einem Host abzurufen, fragt der Zabbix Server das Zabbix **Java gateway** ab, das wiederum die JMX-Management-API verwendet, um die betreffende Anwendung remote abzufragen.

Weitere Details und Informationen zur Einrichtung finden Sie im Abschnitt Zabbix Java gateway.

Warning:
Die Kommunikation zwischen dem Java gateway und der überwachten JMX-Anwendung sollte nicht durch eine Firewall blockiert werden.

Aktivierung der Remote-JMX-Überwachung für eine Java-Anwendung

Für eine Java-Anwendung muss keine zusätzliche Software installiert werden, sie muss jedoch mit den unten angegebenen Befehlszeilenoptionen gestartet werden, damit die Remote-JMX-Überwachung unterstützt wird.

Als absolutes Minimum können Sie, wenn Sie einfach mit der Überwachung einer einfachen Java-Anwendung auf einem lokalen Host ohne erzwungene Sicherheitsmaßnahmen beginnen möchten, sie mit diesen Optionen starten:

```
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
```

Dadurch lauscht Java auf eingehende JMX-Verbindungen an Port 12345, nur vom lokalen Host aus, und es wird festgelegt, dass weder Authentifizierung noch SSL erforderlich sind.

Wenn Sie Verbindungen über eine andere Schnittstelle zulassen möchten, setzen Sie den Parameter -Djava.rmi.server.hostname auf die IP-Adresse dieser Schnittstelle.

Wenn Sie strengere Sicherheitsanforderungen umsetzen möchten, stehen Ihnen viele weitere Java-Optionen zur Verfügung. Im nächsten Beispiel wird die Anwendung beispielsweise mit einem vielseitigeren Satz von Optionen gestartet und für ein größeres Netzwerk geöffnet, nicht nur für den lokalen Host.

```
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
```

Die meisten (wenn nicht alle) dieser Einstellungen können in `$JRE/lib/management/management.properties` angegeben werden (oder dort, wo sich diese Datei auf Ihrem System befindet).

Beachten Sie, dass Sie, wenn Sie SSL verwenden möchten, das Skript startup.sh ändern müssen, indem Sie die Optionen `-Djavax.net.ssl.*` zum Java gateway hinzufügen, damit es weiß, wo sich Schlüssel- und Truststores befinden.

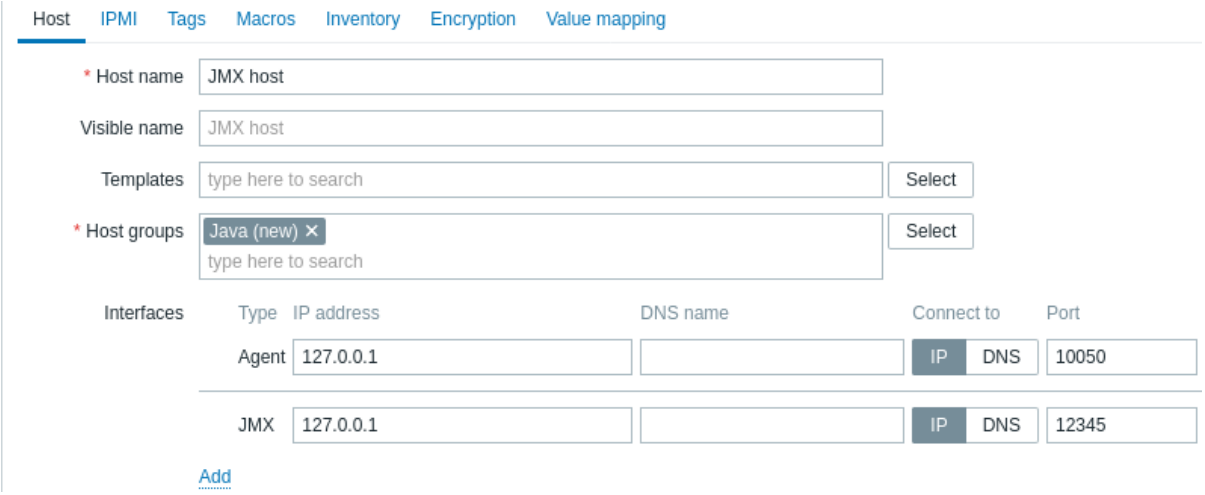
Eine ausführliche Beschreibung finden Sie unter Monitoring and Management Using JMX.

Konfiguration von JMX-Schnittstellen und Datenpunkten im Zabbix Frontend

Wenn der Java gateway läuft, der Server weiß, wo er ihn finden kann, und eine Java-Anwendung mit Unterstützung für die entfernte JMX-Überwachung gestartet wurde, ist es an der Zeit, die Schnittstellen und Datenpunkte in der Zabbix-GUI zu konfigurieren.

Konfigurieren der JMX-Schnittstelle

Sie beginnen damit, auf dem gewünschten Host eine Schnittstelle vom Typ JMX zu erstellen.

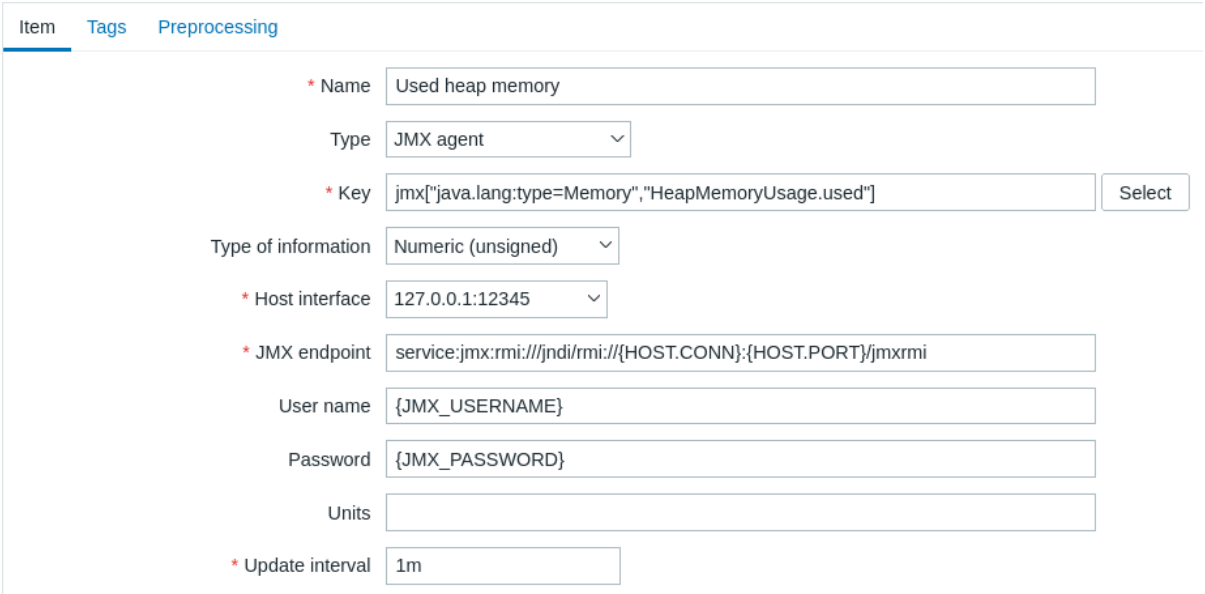


Alle obligatorischen Eingabefelder sind mit einem roten Sternchen markiert.

Hinzufügen eines JMX Agent-Datenpunkts

Für jeden JMX-Zähler, an dem Sie interessiert sind, fügen Sie einen **JMX Agent**-Datenpunkt hinzu, der an diese Schnittstelle angehängt ist.

Der Schlüssel im folgenden Screenshot lautet `jmx["java.lang:type=Memory","HeapMemoryUsage.used"]`.



Alle obligatorischen Eingabefelder sind mit einem roten Sternchen markiert.

Die Felder, die für JMX-Datenpunkte spezifische Informationen erfordern, sind:

Type	Setzen Sie hier **JMX Agent**.

Key	Der `jmx[]`-Datenpunktschlüssel enthält drei Parameter: **object name** - der Objektname eines MBeans **attribute name** - ein MBean-Attributname mit optionalen zusammengesetzten Datenfeldnamen, die durch Punkte getrennt sind **unique short description** - eine eindeutige Beschreibung, die mehrere JMX-Datenpunkte mit demselben Objektnamen und Attributnamen auf dem Host ermöglicht (optional) Weitere Details zu JMX-Datenpunktschlüsseln finden Sie unten. Sie können MBeans und MBean-Attribute mithilfe eines `jmx.discovery[]`-Low-Level-Discovery-Datenpunkts ermitteln.
JMX endpoint	Sie können einen benutzerdefinierten JMX-Endpunkt angeben. Stellen Sie sicher, dass die Verbindungsparameter des JMX-Endpunkts mit der JMX-Schnittstelle übereinstimmen. Dies kann durch die Verwendung von `{HOST.*}`-Makros erreicht werden, wie beim standardmäßigen JMX-Endpunkt. `{HOST.*}`-Makros und benutzerdefinierte Makros werden unterstützt.
User name	Geben Sie den Benutzernamen an (bis zu 255 Zeichen), wenn Sie die Authentifizierung für Ihre Java-Anwendung konfiguriert haben. Benutzer-Makros werden unterstützt.
Password	Geben Sie das Passwort an (bis zu 255 Zeichen), wenn Sie die Authentifizierung für Ihre Java-Anwendung konfiguriert haben. Benutzer-Makros werden unterstützt.

Wenn Sie einen Boolean-Zähler überwachen möchten, der entweder ”true” oder ”false” ist, geben Sie den Informationstyp als ”Numeric (unsigned)” an und wählen Sie auf der Registerkarte Preprocessing den Verarbeitungsschritt ”Boolean to decimal” aus. Der Server speichert Boolean-Werte dann entsprechend als 1 oder 0.

JMX-Datenpunktschlüssel im Detail

Einfache Attribute

Ein MBean-Objektname ist nichts weiter als eine Zeichenkette, die Sie in Ihrer Java-Anwendung definieren. Ein Attributname hingegen kann komplexer sein. Falls ein Attribut einen primitiven Datentyp zurückgibt (eine Ganzzahl, eine Zeichenkette usw.), gibt es nichts, worüber man sich Sorgen machen müsste; der Schlüssel sieht dann wie folgt aus:

```
jmx[com.example:Type=Hello,weight]
```

In diesem Beispiel ist der Objektname „com.example:Type=Hello“, der Attributname „weight“ und der Typ des zurückgegebenen Werts sollte wahrscheinlich „Numerisch (Gleitkommazahl)“ sein.

Attribute, die zusammengesetzte Daten zurückgeben

Es wird komplizierter, wenn Ihr Attribut zusammengesetzte Daten zurückgibt. Zum Beispiel: Ihr Attributname ist „apple“ und es gibt einen Hash zurück, der seine Parameter wie „weight“, „color“ usw. darstellt. Ihr Schlüssel kann dann wie folgt aussehen:

```
jmx[com.example:Type=Hello,apple.weight]
```

So werden ein Attributname und ein Hash-Schlüssel voneinander getrennt, nämlich durch die Verwendung eines Punkts. Genauso werden, wenn ein Attribut verschachtelte zusammengesetzte Daten zurückgibt, die Teile durch einen Punkt getrennt:

```
jmx[com.example:Type=Hello,fruits.apple.weight]
```

Attribute, die tabellarische Daten zurückgeben

Attribute für tabellarische Daten bestehen aus einem oder mehreren zusammengesetzten Attributen. Wenn ein solches Attribut im Parameter für den Attributnamen angegeben wird, gibt dieser Datenpunkt den vollständigen Aufbau des Attributs im JSON-Format zurück. Die Werte der einzelnen Elemente innerhalb des Attributs mit tabellarischen Daten können mithilfe der Vorverarbeitung abgerufen werden.

Beispiel für ein Attribut mit tabellarischen Daten:

```
jmx[com.example:type=Hello,foodinfo]
```

Datenpunktwert:

```
[
  {
    "a": "apple",
    "b": "banana",
    "c": "cherry"
  },
```

```
  {
    "a": "potato",
    "b": "lettuce",
    "c": "onion"
  }
]
```

Problem mit Punkten

So weit, so gut. Aber was ist, wenn ein Attributname oder ein Hash-Schlüssel ein Punktsymbol enthält? Hier ist ein Beispiel:

`jmx[com.example:Type=Hello,all.fruits.apple.weight]`

Das ist ein Problem. Wie teilt man Zabbix mit, dass der Attributname „all.fruits“ und nicht nur „all“ ist? Wie unterscheidet man einen Punkt, der Teil des Namens ist, von dem Punkt, der einen Attributnamen und Hash-Schlüssel trennt?

Das ist möglich; alles, was Sie tun müssen, ist, die Punkte, die Teil des Namens sind, mit einem Backslash zu maskieren:

`jmx[com.example:Type=Hello,all\.fruits.apple.weight]`

Auf die gleiche Weise maskieren Sie einen Punkt in Ihrem Hash-Schlüssel:

`jmx[com.example:Type=Hello,all\.fruits.apple.total\.weight]`

Weitere Probleme

Ein Backslash-Zeichen in einem Attributnamen sollte maskiert werden:

`jmx[com.example:type=Hello,c:\\documents]`

Informationen zur Behandlung aller anderen Sonderzeichen im JMX-Datenpunktschlüssel finden Sie im Abschnitt zum Datenpunktschlüsselformat section.

Das ist tatsächlich schon alles. Viel Erfolg beim JMX-Monitoring!

Nicht-primitive Datentypen

Es ist möglich, mit benutzerdefinierten MBeans zu arbeiten, die nicht-primitive Datentypen zurückgeben und die Methode **toString()** überschreiben.

Verwendung eines benutzerdefinierten Endpunkts mit JBoss EAP 6.4

Benutzerdefinierte Endpunkte ermöglichen die Arbeit mit verschiedenen Transportprotokollen anstelle des standardmäßigen RMI.

Um diese Möglichkeit zu veranschaulichen, konfigurieren wir beispielhaft die Überwachung von JBoss EAP 6.4. Zunächst treffen wir einige Annahmen:

- Sie haben den Zabbix Java gateway bereits installiert. Falls nicht, können Sie dies gemäß der Dokumentation tun.
- Zabbix Server und Java gateway sind mit dem Präfix `/usr/local/` installiert
- JBoss ist bereits in `/opt/jboss-eap-6.4/` installiert und läuft im Standalone-Modus
- Wir gehen davon aus, dass alle diese Komponenten auf demselben Host ausgeführt werden
- Firewall und SELinux sind deaktiviert (oder entsprechend konfiguriert)

Nehmen wir einige einfache Einstellungen in `zabbix_server.conf` vor:

```
JavaGateway=127.0.0.1
StartJavaPollers=5
```

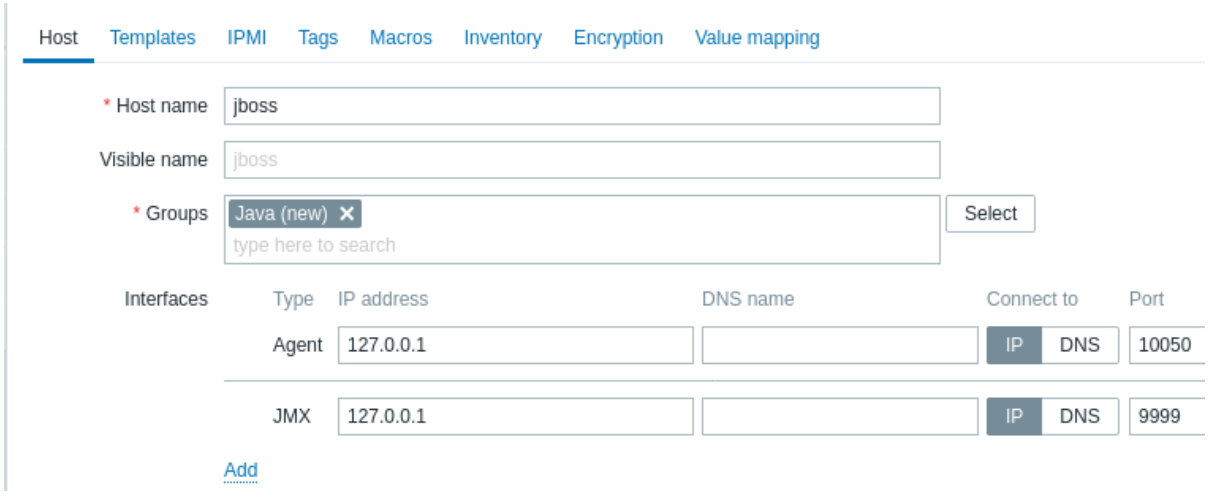
Und in der Konfigurationsdatei `zabbix_java/settings.sh` (oder `zabbix_java_gateway.conf`):

```
START_POLLERS=5
```

Prüfen Sie, ob JBoss an seinem standardmäßigen Verwaltungsport lauscht:

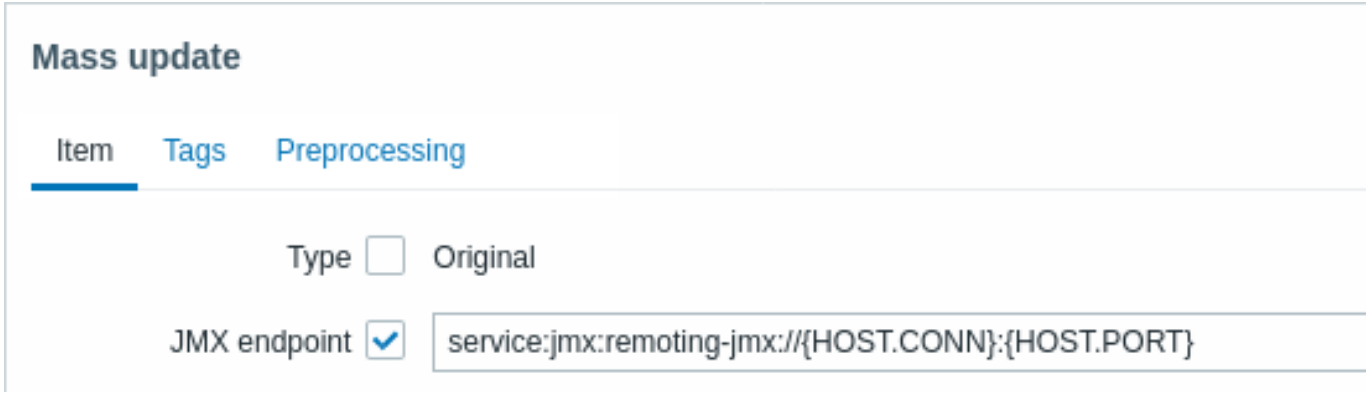
```
$ netstat -natp | grep 9999
tcp        0      0 127.0.0.1:9999          0.0.0.0:*               LISTEN      10148/java
```

Erstellen wir nun in Zabbix einen Host mit der JMX-Schnittstelle 127.0.0.1:9999.



Da wir wissen, dass diese Version von JBoss das JBoss-Remoting-Protokoll anstelle von RMI verwendet, können wir den Parameter des JMX-Endpunkts für Datenpunkte in unserer JMX-Vorlage entsprechend per Massenaktualisierung anpassen:

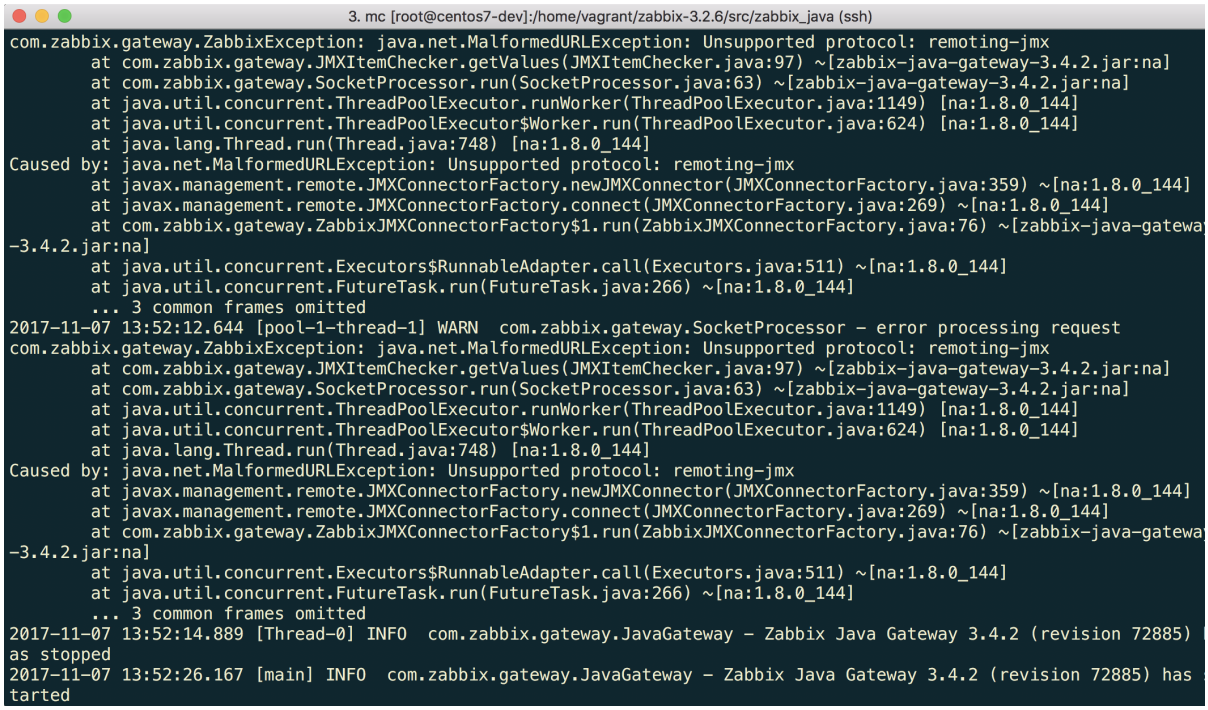
```
service:jmx:remoting-jmx://{HOST.CONN}:{HOST.PORT}
```



Aktualisieren wir den Konfigurationscache:

```
/usr/local/sbin/zabbix_server -R config_cache_reload
```

Beachten Sie, dass zunächst ein Fehler auftreten kann.



”Unsupported protocol: remoting-jmx” bedeutet, dass der Java gateway nicht weiß, wie mit dem angegebenen Protokoll umzugehen ist. Dies lässt sich beheben, indem Sie eine Datei `~/needed_modules.txt` mit folgendem Inhalt erstellen:

```
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
```

und anschließend den folgenden Befehl ausführen:

```
for i in $(cat ~/needed_modules.txt); do find /opt/jboss-eap-6.4 -iname "${i}*.jar" -exec cp '{}' /usr/loc
```

Damit verfügt der Java gateway über alle erforderlichen Module für die Arbeit mit jmx-remoting. Nun müssen Sie nur noch den Java gateway neu starten, kurz warten und wenn alles korrekt eingerichtet ist, sehen Sie, dass JMX-Überwachungsdaten in Zabbix eintreffen (siehe auch: Neueste Daten).

Berechneter Datenpunkt

Übersicht

Dieser Datentyp ermöglicht Berechnungen für bereits vorhandene Metriken. Ein berechneter Datenpunkt erfasst selbst keine Daten.

Beispielsweise können Sie:

- Zwei Werte zählen.
- Den Durchschnitt mehrerer Datenpunktwerte berechnen.
- Den Gesamtwert einer Gruppe von Datenpunkten berechnen.

Der resultierende Wert wird wie bei jedem anderen Datenpunkt in der Zabbix-Datenbank gespeichert; sowohl Verlaufs- als auch Trendwerte werden gespeichert, und es können Diagramme erstellt werden.

Die Berechnung basiert auf einer Formel. Die Syntax der Formel wird mit Auslöserausdrücken gemeinsam verwendet.

Tatsächlich ist es möglich, Auslöser einfach zu halten, indem komplexe Logik zur Datenaufbereitung in einem berechneten Datenpunkt implementiert wird. Anschließend kann der berechnete Datenpunkt in einem Auslöser über den Datenpunktschlüssel referenziert werden.

Für die Angabe der Datenquelle für Berechnungen gibt es zwei Möglichkeiten:

1. Datenpunkte in der Formel angeben.
2. Eine Datenaggregation anhand eines Filters durchführen, der Datenpunkte auswählt.

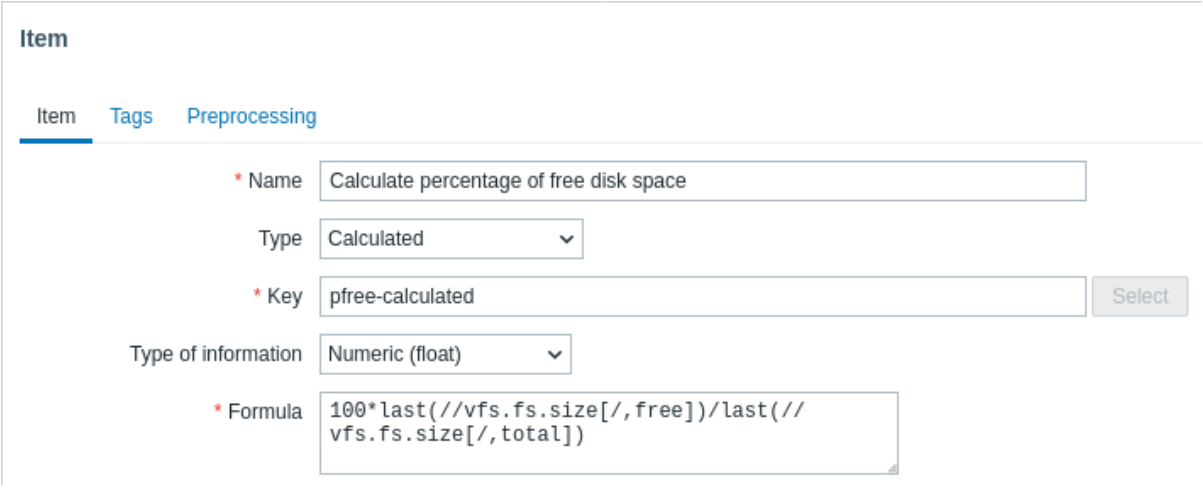
Mit der zweiten Option können Sie Gruppenmittelwerte, Summen usw. berechnen.

Alle Berechnungen werden vom Zabbix-Server durchgeführt und von den `history poller`-Prozessen verarbeitet; Proxy oder Agents sind daran überhaupt nicht beteiligt. Beachten Sie, dass eine umfangreiche Verwendung berechneter Datenpunkte mit langen Zeiträumen die Leistung des Servers beeinträchtigen kann.

Bitte beachten Sie, dass berechnete Datenpunkte auf den Verlauf anderer Datenpunkte zugreifen können, einschließlich Datenpunkten, auf die der Benutzer, der den berechneten Datenpunkt erstellt, keinen Zugriff hat. Wenn dieses Verhalten nicht gewünscht ist, können Sie berechnete Datenpunkte deaktivieren, indem Sie den Konfigurationsparameter `StartHistoryPollers` auf 0 setzen.

Konfigurierbare Felder

Um berechnete Datenpunkte zu verwenden, wählen Sie den Datenpunkttyp **Calculated**.



Der **Schlüssel** ist eine eindeutige Datenpunktkennung (pro Host). Sie können unter Verwendung der unterstützten Symbole einen beliebigen Schlüsselnamen erstellen. Mit dem Schlüssel können Sie in Auslösern auf den berechneten Datenpunkt verweisen.

Die Berechnungsdefinition muss im Feld **Formula** eingegeben werden.

Es besteht keine Verbindung zwischen der Formel und dem Schlüssel. Die Schlüsselparameter werden in keiner Weise in der Formel verwendet.

Im Gegensatz zu Auslöserausdrücken verarbeitet Zabbix berechnete Datenpunkte gemäß dem Datenpunkt-Aktualisierungsintervall und nicht beim Empfang eines neuen Werts.

Beachten Sie, dass das Berechnungsergebnis auf eine Ganzzahl gekürzt wird, wenn Sie *Numeric (unsigned)* als Informationstyp auswählen und das Ergebnis ein Gleitkommawert ist.

Formel für einfache Berechnungen

Die Syntax einer einfachen Formel lautet:

`function(/host/key,<parameter1>,<parameter2>,...)`

Dabei gilt:

Element	Beschreibung
`function`	Eine der unterstützten Funktionen: last, min, max, avg, count usw.
`host`	Host des Datenpunkts, der für die Berechnung verwendet wird. Der aktuelle Host kann weggelassen werden (d. h. wie in `function(//key,parameter,...)`).
`key`	Schlüssel des Datenpunkts, der für die Berechnung verwendet wird. Datenpunkte, die Werte mit dem Datentyp Binär oder JSON zurückgeben, werden nicht unterstützt. Wenn Sie den Datenpunktschlüssel eines referenzierten Datenpunkts ändern, muss der Schlüssel in der Formel manuell aktualisiert werden. Alle Datenpunkte, auf die von Verlaufsfunktionen verwiesen wird, müssen vorhanden sein und Daten erfassen.
`parameter(s)`	Parameter der Funktion, falls erforderlich. Zeitsuffixe und Speichergrößensuffixe werden unterstützt.

Eine komplexere Formel kann eine Kombination aus Funktionen, Operatoren und Klammern verwenden.

Siehe Beispiele für Formeln berechneter Datenpunkte:

- (1) Gesamter Traffic auf einer Schnittstelle
- (2) Prozentsatz des freien Festplattenspeichers
- (3) Prozentsatz des eingehenden Traffics
- (4) Durchschnittliche Anzahl verarbeiteter Werte

Einige Variablen können mit Benutzermakros an die Formel übergeben werden. Beachten Sie, dass Benutzermakros:

- aufgelöst werden - wenn sie verwendet werden, um auf einen Funktionsparameter, einen Datenpunktfilterparameter oder eine Konstante zu verweisen.
- **nicht** aufgelöst werden - wenn sie auf eine Funktion, einen Hostnamen, einen Datenpunktschlüssel, einen Datenpunktschlüsselparameter oder einen Operator verweisen.

Der Vergleich mit Zeichenfolgen ist zulässig.

Wenn sich keine aktuellen Daten im Cache befinden und in der Funktion kein Abfragezeitraum definiert ist, geht Zabbix standardmäßig bis zu eine Woche in die Vergangenheit zurück, um die Datenbank nach historischen Werten abzufragen.

Ein berechneter Datenpunkt kann in mehreren Fällen nicht unterstützt werden:

- referenzierter Datenpunkt
  - wird nicht gefunden
  - ist deaktiviert oder gehört zu einem deaktivierten Host
  - wird nicht unterstützt (außer mit der Funktion nodata() und Operatoren mit unbekannten Werten)
- keine Daten zur Berechnung einer Funktion
- Division durch null
- falsche Syntax verwendet

Datenaggregationen

Datenaggregationen basieren auf einem Filter, der mehrere Datenpunkte auswählt. Datenaggregationen sind ebenfalls ein Typ berechneter Datenpunkte, allerdings verwenden Sie in der Formel eine leicht andere Syntax.

Um Aggregatwerte abzurufen, verwenden Sie eine der unterstützten Aggregatfunktionen: `avg`, `max`, `min`, `sum` usw. Fügen Sie dann die Funktion **foreach** als einzigen Parameter sowie ihren Datenpunkt-Filter hinzu, um die erforderlichen Datenpunkte auszuwählen:

```
aggregate_function(function_foreach(/host/key?[group="host group"],timeperiod))
```

Eine **foreach**-Funktion (z. B. *avg_foreach*, *count_foreach* usw.) gibt für jeden ausgewählten Datenpunkt einen Aggregatwert zurück. Datenpunkte werden mithilfe des Datenpunkt-Filters (`/host/key?[group="host group"]`) aus der Datenpunkthistorie ausgewählt. Weitere Details finden Sie unter foreach functions.

Wenn einige der Datenpunkte für den angeforderten Zeitraum keine Daten haben, werden sie bei der Berechnung ignoriert. Wenn keine Datenpunkte Daten haben, gibt die Funktion einen Fehler zurück.

Alternativ können Sie mehrere Datenpunkte als Parameter für die Aggregation auflisten:

```
aggregate_function(function(/host/key,parameter),function(/host2/key2,parameter),...)
```

Beachten Sie, dass `function` hier eine Verlaufs-/Trendfunktion sein muss.

Siehe Beispiele für Aggregatberechnungen:

- (5) Gesamter eingehender Traffic für Host
- (6) Gesamter Festplattenspeicher für Gruppe
- (7) Durchschnittliche Prozessorlast für Gruppe
- (8) Durchschnittliche Prozessorlast für Gruppe und Tag
- (9) Durchschnittliche Anzahl verarbeiteter Werte
- (10) Durchschnittliche Anzahl von Abfragen für Gruppe
- (11) Gesamtzahl erfolgreicher DNS-Prüfungen
- (12) Gesamtzahl nicht unterstützter Datenpunkte für Gruppe
- (13) Schnittstellenstatistiken für Gruppe

Benutzermakros und Low-Level-Discovery-Makros werden unterstützt in:

- Datenpunktschlüsselparametern
- Funktionsparametern
- Filterbedingungen (Name der Host-Gruppe und Tag-Name)
- Ausdruckskonstanten

Eine Aggregatberechnung kann nicht unterstützt werden, wenn:

- keiner der referenzierten Datenpunkte gefunden wird (was passieren kann, wenn der Datenpunktschlüssel falsch ist, keiner der Datenpunkte existiert oder alle einbezogenen Gruppen falsch sind)
- keine Daten zur Berechnung einer Funktion vorhanden sind

Beispiele für berechnete Datenpunkte

(1) Gesamter Datenverkehr auf der Schnittstelle

```
last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes])
```

Berechnet die gesamte Bandbreite auf eth0.

(2) Prozentsatz des freien Festplattenspeichers

```
100*last(//vfs.fs.size[/,free])/last(//vfs.fs.size[/,total])
```

Zabbix verwendet die neuesten Werte für freien und gesamten Festplattenspeicher (auf /) und berechnet den Prozentsatz gemäß der angegebenen Formel.

(3) Prozentsatz des eingehenden Datenverkehrs

```
100*last(//net.if.in[eth0,bytes])/(last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes]))
```

Berechnet den Prozentsatz des eingehenden Datenverkehrs am gesamten Datenverkehr.

(4) Durchschnittliche Anzahl verarbeiteter Werte

```
avg(/Zabbix server/zabbix[wcache,values],10m)
```

Berechnet einen 10-Minuten-Durchschnitt der Anzahl der von Zabbix verarbeiteten Werte.

(5) Gesamter eingehender Datenverkehr für den Host

```
sum(last_foreach(/host/net.if.in[*]))
```

Berechnet die Summe aller Datenpunkte, die `net.if.in[*]` auf dem Host entsprechen.

(6) Gesamter Festplattenspeicher für Gruppe

```
sum(last_foreach(/*/vfs.fs.size[/,total]?[group="MySQL Servers"]))
```

Berechnet den gesamten Festplattenspeicher für alle `MySQL Servers`.

(7) Durchschnittliche Prozessorlast für Gruppe

```
avg(last_foreach(/*/system.cpu.load[,avg1]?[group="MySQL Servers"]))
```

Berechnet die durchschnittliche Prozessorlast für alle `MySQL Servers`.

(8) Durchschnittliche Prozessorlast für Gruppe und Tag

```
avg(last_foreach(/*/system.cpu.load?[(group="Servers A" or group="Servers B" or group="Servers C") and (ta
```

Berechnet die durchschnittliche CPU-Last auf allen Hosts in mehreren Host-Gruppen, die bestimmte Tags haben.

(9) Durchschnittliche Anzahl von Abfragen für Gruppe

```
avg(avg_foreach(/*/mysql.qps?[group="MySQL Servers"],5m))
```

Berechnet den 5-Minuten-Durchschnitt der Anzahl von Abfragen pro Sekunde für `MySQL Servers`.

(10) Gesamtzahl der erfolgreichen DNS-Prüfungen

```
sum(last_foreach(/*/net.dns[*,*,*]))
```

Berechnen Sie die Gesamtzahl der erfolgreichen DNS-Prüfungen über alle Hosts hinweg. Das angezeigte Datenpunkt-Formular ist `net.dns[192.0.2.1,example.com,A]` als Beispiel für einen möglichen Schlüssel.

Beachten Sie, dass Platzhalter mit der Anzahl der Parameter im Schlüssel übereinstimmen müssen (hier hat `net.dns` drei Parameter: ip, name, type).

(11) Gesamtzahl der nicht unterstützten Datenpunkte für die Gruppe

```
sum(last_foreach(/*/zabbix[host,,items_unsupported]?[group="Zabbix servers"]))
```

Berechnet die Gesamtzahl der nicht unterstützten Datenpunkte für `Zabbix servers`.

(12) Schnittstellenstatistiken für Gruppe

```
sum(last_foreach(/*/net.if.out[eth0,bytes]?[group="video"])) / sum(last_foreach(/*/nginx_stat.sh[active]?[
```

Berechnet die Schnittstellenstatistiken für die Gruppe `video`.

Beispiele für korrekte/inkorrekte Syntax

Ausdrücke (einschließlich Funktionsaufrufen) können nicht als Parameter für Verlaufs-, Trend- oder foreach-Funktionen verwendet werden. Diese Funktionen selbst können jedoch in anderen (nicht verlaufsbezogenen) Funktionsparametern verwendet werden.

Ausdruck	Beispiel
Gültig	`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)))`
Ungültig	`sum(/host/key,10+2)`
	`sum(/host/key, avg(10,2))`
	`sum(/host/key,last(/host/key2))`

Beachten Sie, dass bei einem Ausdruck wie:

```
sum(sum_foreach(//resptime[*],5m))/sum(count_foreach(//resptime[*],5m))
```

(wobei `resptime[*]` ein Beispiel für ein Datenpunktschlüsselmuster ist, das den Schlüsseln `resptime[1]`, `resptime[2]` usw. entspricht und zum Abrufen der Antwortzeit von mehreren aktiven Clusterknoten verwendet wird) nicht garantiert werden kann, dass beide Teile der Gleichung immer dieselbe Wertemenge enthalten. Während ein Teil des Ausdrucks ausgewertet wird, kann ein neuer Wert für den angeforderten Zeitraum eintreffen. Dadurch enthält der andere Teil des Ausdrucks eine andere Wertemenge.

Abhängiger Datenpunkt

Übersicht

Es gibt Situationen, in denen ein Datenpunkt mehrere Metriken gleichzeitig erfasst oder es sogar sinnvoller ist, zusammengehörige Metriken gleichzeitig zu sammeln, zum Beispiel:

- CPU-Auslastung einzelner Kerne
- Eingehender/ausgehender/gesamter Netzwerkverkehr

Um die Erfassung mehrerer Metriken auf einmal und die gleichzeitige Verwendung in mehreren zusammengehörigen Datenpunkten zu ermöglichen, unterstützt Zabbix abhängige Datenpunkte. Abhängige Datenpunkte hängen von dem Master-Datenpunkt ab, der ihre Daten gleichzeitig in einer Abfrage erfasst. Ein neuer Wert für den Master-Datenpunkt füllt die Werte der abhängigen Datenpunkte automatisch aus. Abhängige Datenpunkte können kein anderes Aktualisierungsintervall als der Master-Datenpunkt haben.

Die Zabbix-Optionen zur Vorverarbeitung können verwendet werden, um aus den Daten des Master-Datenpunkts den Teil zu extrahieren, der für den abhängigen Datenpunkt benötigt wird.

Die Vorverarbeitung wird von einem Prozess `preprocessing manager` verwaltet, zusammen mit Worker-Threads, die die Vorverarbeitungsschritte ausführen. Alle Werte mit Vorverarbeitung, die von verschiedenen Datensammlern empfangen werden, durchlaufen den preprocessing manager, bevor sie dem Verlaufscache hinzugefügt werden. Für die IPC-Kommunikation auf Socket-Basis zwischen Datensammlern (Pollern, Trappern usw.) und dem Vorverarbeitungsprozess wird Socket-basierte IPC-Kommunikation verwendet.

Zabbix Server oder Zabbix Proxy (wenn der Host durch einen Proxy überwacht wird) führen die Vorverarbeitungsschritte aus und verarbeiten abhängige Datenpunkte.

Datenpunkte jedes Typs, auch abhängige Datenpunkte, können als Master-Datenpunkte festgelegt werden. Zusätzliche Ebenen abhängiger Datenpunkte können verwendet werden, um kleinere Teile aus dem Wert eines vorhandenen abhängigen Datenpunkts zu extrahieren.

Einschränkungen

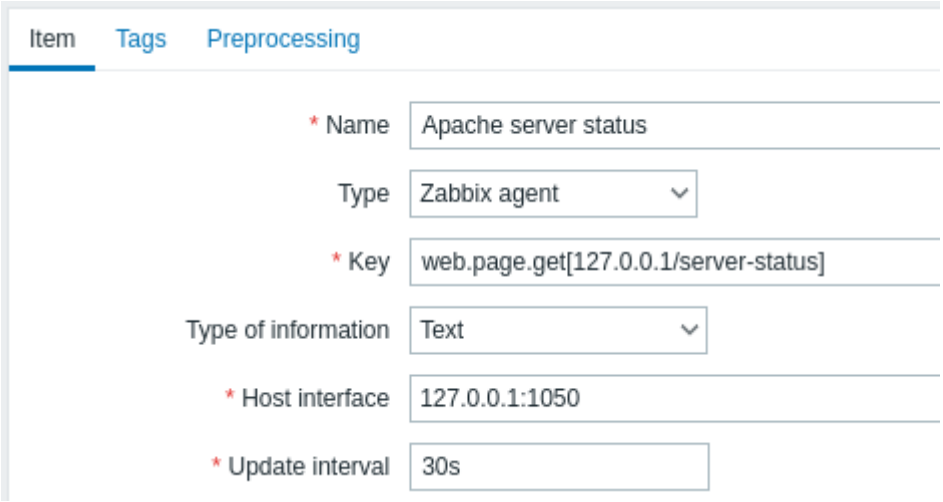
- Es sind nur Abhängigkeiten innerhalb desselben Hosts (derselben Vorlage) zulässig
- Ein Datenpunkt-Prototyp kann von einem anderen Datenpunkt-Prototyp oder einem regulären Datenpunkt desselben Hosts abhängen
- Ein abhängiger Datenpunkt auf einem Host mit einem Master-Datenpunkt aus einer Vorlage wird nicht nach XML exportiert

Item-Konfiguration

Ein abhängiger Datenpunkt ist für seine Daten von seinem Master-Datenpunkt abhängig. Deshalb muss der **Master-Datenpunkt** zuerst konfiguriert werden (oder bereits vorhanden sein):

1. Gehen Sie zu *Datenerfassung* > *Hosts*.
2. Klicken Sie in der Zeile des Hosts auf *Datenpunkte*.
3. Klicken Sie auf *Datenpunkt erstellen*.

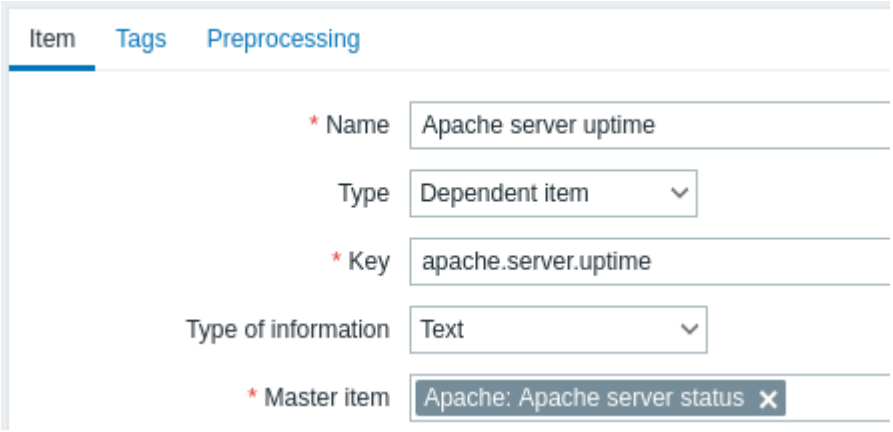
4. Geben Sie die Parameter des Datenpunkts im Formular ein.



Alle Pflichtfelder sind mit einem roten Sternchen markiert.

Klicken Sie auf *Hinzufügen*, um den Master-Datenpunkt zu speichern.

Danach können Sie einen **abhängigen Datenpunkt** konfigurieren.

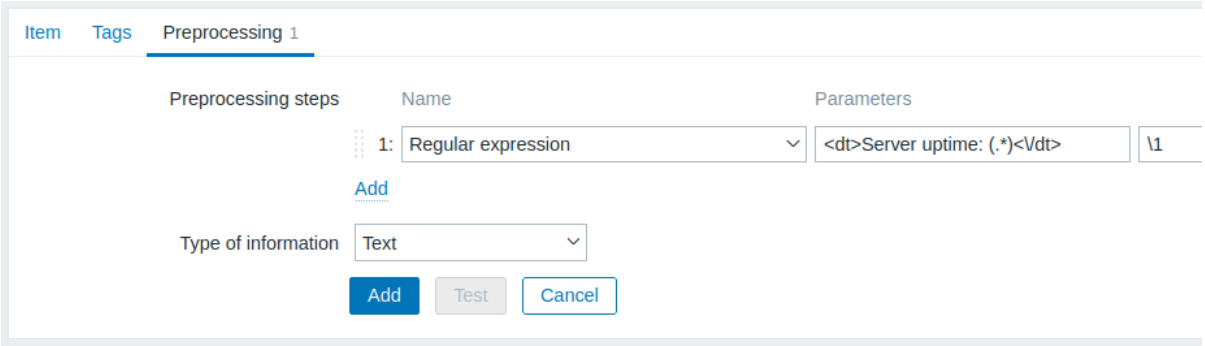


Alle Pflichtfelder sind mit einem roten Sternchen markiert.

Die Felder, die für abhängige Datenpunkte spezifische Informationen erfordern, sind:

Typ	Wählen Sie hier **Abhängiger Datenpunkt** aus.
Schlüssel	Geben Sie einen Schlüssel ein, der zur Identifizierung des Datenpunkts verwendet wird.
Master item	Wählen Sie den Master-Datenpunkt aus. Der Wert des Master-Datenpunkts wird verwendet, um den Wert des abhängigen Datenpunkts zu füllen.
Typ der Information	Wählen Sie den Informationstyp aus, der dem Format der zu speichernden Daten entspricht.

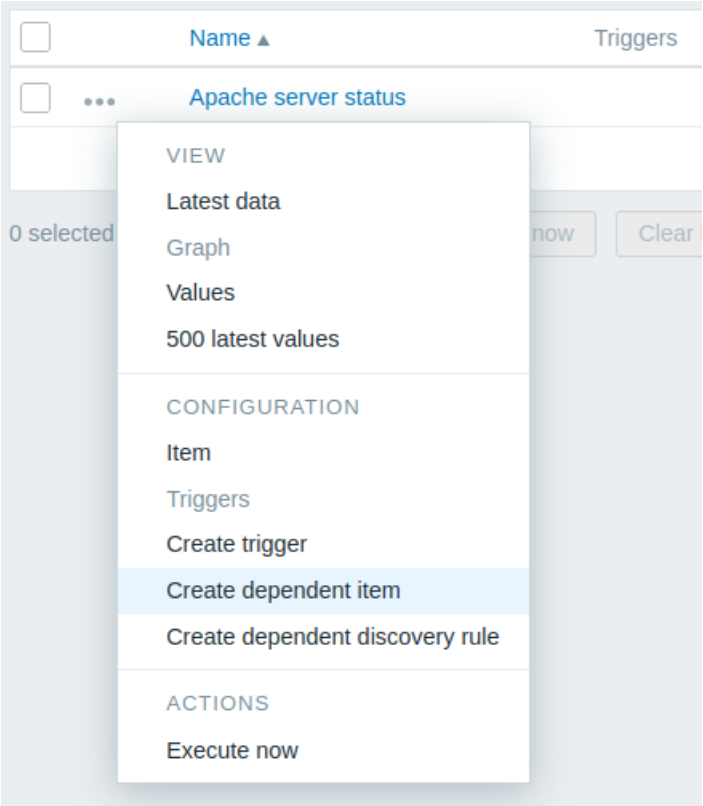
Sie können die Vorverarbeitung des Datenpunktwerts verwenden, um den benötigten Teil des Werts des Master-Datenpunkts zu extrahieren.



Ohne Vorverarbeitung ist der Wert des abhängigen Datenpunkts genau derselbe wie der Wert des Master-Datenpunkts.

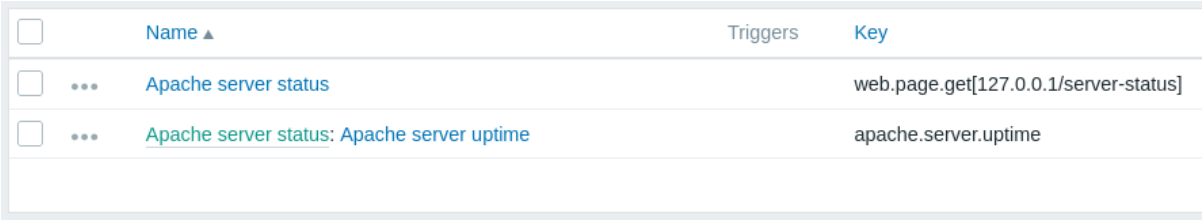
Klicken Sie auf *Hinzufügen*, um den abhängigen Datenpunkt zu speichern.

Eine schnellere Möglichkeit zum Erstellen eines abhängigen Datenpunkts erhalten Sie, indem Sie auf die Schaltfläche ••• in der Datenpunktliste klicken und *Abhängigen Datenpunkt erstellen* auswählen.



Anzeige

In der Datenpunktliste werden abhängige Datenpunkte mit dem Namen ihres Master-Datenpunkts als Präfix angezeigt.



Wenn ein Master-Datenpunkt gelöscht wird, werden auch alle von ihm abhängigen Datenpunkte gelöscht.

Skript-Datenpunkt

Übersicht

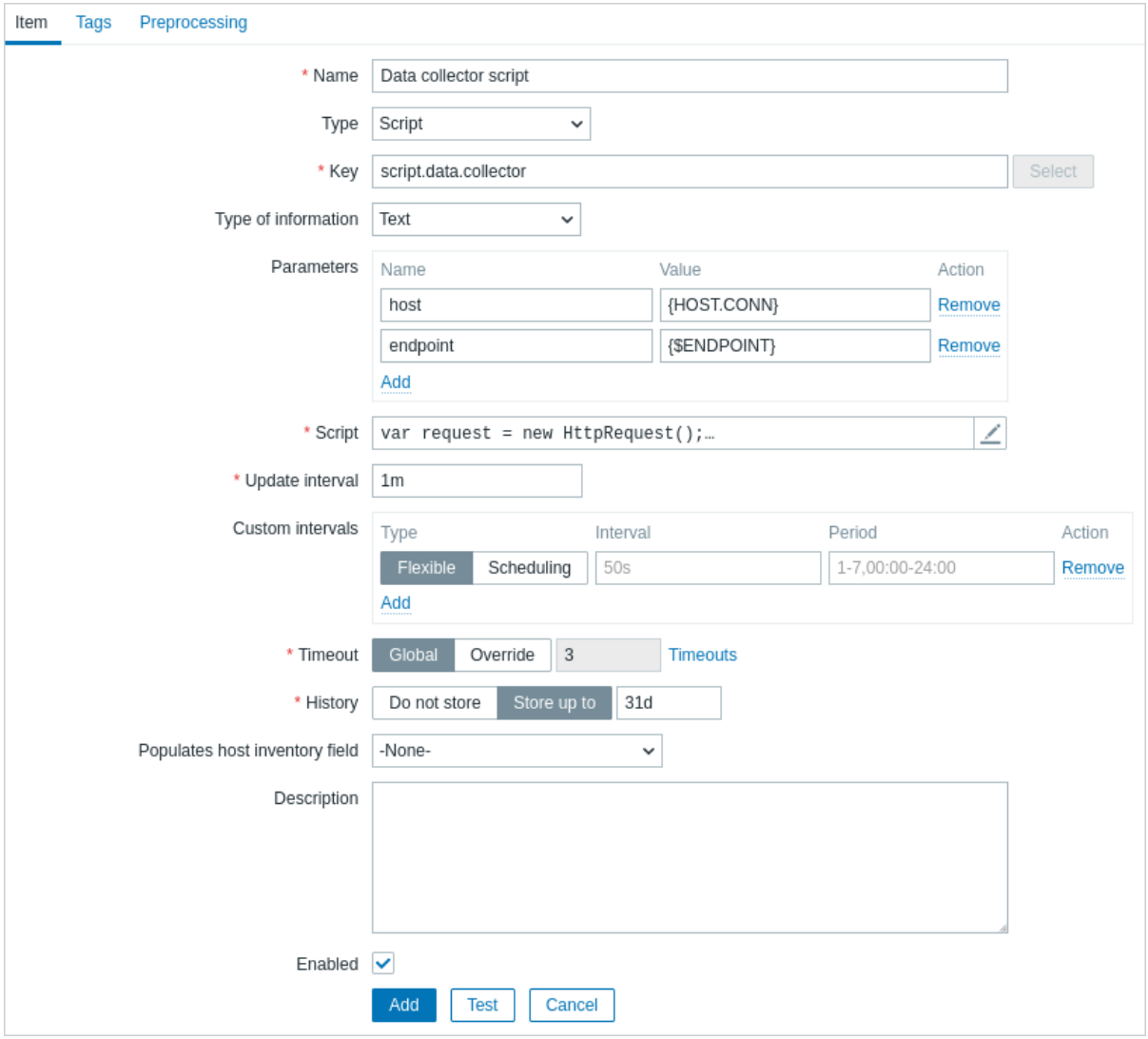
Script-Datenpunkte können verwendet werden, um Daten zu erfassen, indem benutzerdefinierter JavaScript-Code ausgeführt wird, mit der Möglichkeit, Daten über HTTP/HTTPS abzurufen. Zusätzlich zum Script können eine optionale Liste von Parametern (Name-Wert-Paare) sowie ein Timeout angegeben werden.

Dieser Datenpunkttyp kann in Datenerfassungsszenarien nützlich sein, die mehrere Schritte oder eine komplexe Logik erfordern. Beispielsweise kann ein Script-Datenpunkt so konfiguriert werden, dass er einen HTTP-Aufruf ausführt, dann die im ersten Schritt empfangenen Daten auf irgendeine Weise verarbeitet und den transformierten Wert an den zweiten HTTP-Aufruf übergibt.

Script-Datenpunkte werden von Pollern des Zabbix Server oder Proxy verarbeitet.

Konfiguration

Wählen Sie im Feld *Typ* des Datenpunkt-Konfigurationsformulars die Option Skript und füllen Sie dann die erforderlichen Felder aus.



Alle obligatorischen Eingabefelder sind mit einem roten Sternchen markiert.

Die Felder, für die bei Skript-Datenpunkten spezifische Informationen erforderlich sind, sind:

Feld	Beschreibung
Schlüssel	Geben Sie einen eindeutigen Schlüssel ein, der zur Identifizierung des Datenpunkts verwendet wird.
Parameter	Geben Sie die Variablen an, die als Attribut-Wert-Paare an das Skript übergeben werden sollen. Benutzermakros werden unterstützt. Um zu sehen, welche integrierten Makros unterstützt werden, suchen Sie in der Tabelle unterstützte Makros nach „Script-type item“.
Skript	Geben Sie JavaScript-Code im modalen Editor ein, der geöffnet wird, wenn Sie in das Parameterfeld klicken oder auf das Stiftsymbol daneben. Dieser Code muss die Logik zur Rückgabe des Metrikwerts bereitstellen. Der Code hat Zugriff auf alle Parameter und alle von Zabbix hinzugefügten zusätzlichen JavaScript-Objekte. Siehe auch: JavaScript Guide.
Zeitüberschreitung	Zeitlimit für die JavaScript-Ausführung (1–600 s; bei Überschreitung wird ein Fehler zurückgegeben). Beachten Sie, dass es je nach Skript länger dauern kann, bis die Zeitüberschreitung ausgelöst wird. Weitere Informationen zum Parameter *Timeout* finden Sie unter allgemeine Datenpunktattribute.

Beispiele

Einfache Datenerfassung

Erfassen Sie den Inhalt von *https://www.example.com/release_notes*:

- Erstellen Sie einen Datenpunkt vom Typ „Script“.

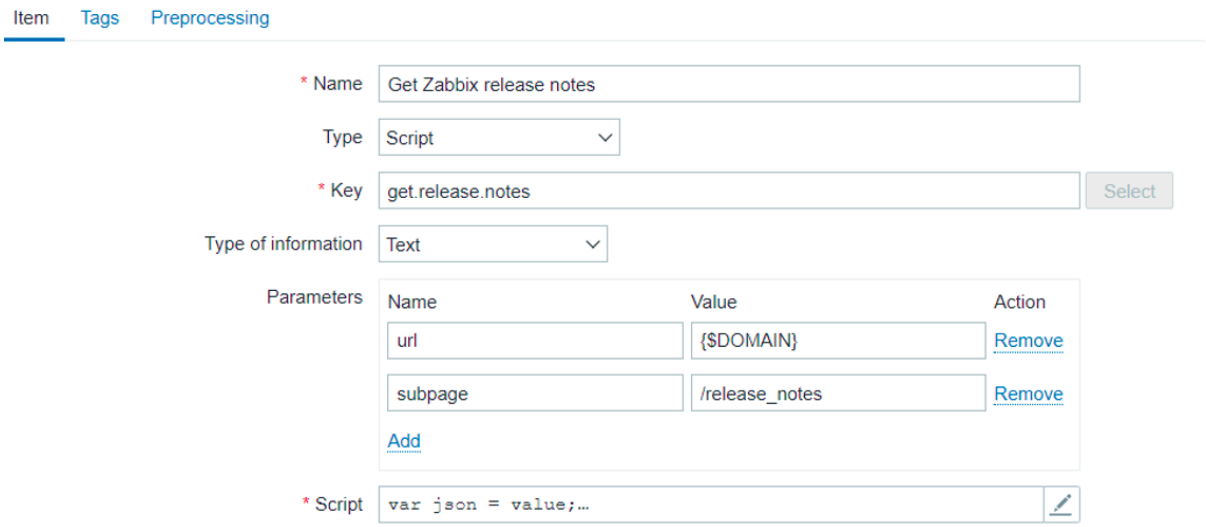
- Geben Sie im Feld *Script* Folgendes ein:

```
var request = new HttpRequest();
return request.get("https://www.example.com/release_notes");
```

Datenerfassung mit Parametern

Erfassen Sie den Inhalt einer bestimmten Seite und verwenden Sie dabei Parameter:

- Erstellen Sie einen Datenpunkt vom Typ „Script“ mit zwei Parametern:
  - **url : {$DOMAIN}** (das Benutzermakro {$DOMAIN} sollte definiert sein, vorzugsweise auf Host-Ebene)
  - **subpage : /release_notes**



- Geben Sie im Feld *Script* Folgendes ein:

```
var obj = JSON.parse(value);
var url = obj.url;
var subpage = obj.subpage;
var request = new HttpRequest();
return request.get(url + subpage);
```

Mehrere HTTP-Anfragen

Erfassen Sie den Inhalt von *https://www.example.com* und *https://www.example.com/release_notes*:

- Erstellen Sie einen Datenpunkt vom Typ „Script“.
- Geben Sie im Feld *Script* Folgendes ein:

```
var request = new HttpRequest();
return request.get("https://www.example.com") + request.get("https://www.example.com/release_notes");
```

Protokollierung

Fügen Sie den Eintrag „Log test“ zum Zabbix-Serverprotokoll hinzu und erhalten Sie den Datenpunktwert „1“ als Rückgabewert:

- Erstellen Sie einen Datenpunkt vom Typ „Script“.
- Geben Sie im Feld *Script* Folgendes ein:

```
Zabbix.log(3, 'Log test');
return 1;
```

Browser-Datenpunkt

Übersicht

Browser-Datenpunkte ermöglichen die Überwachung komplexer Websites und Webanwendungen mithilfe eines Browsers.

> **Attention:**
> Die Unterstützung von Browser-Datenpunkten ist derzeit experimentell.

Browser-Datenpunkte erfassen Daten, indem sie benutzerdefinierten JavaScript-Code ausführen und Daten über HTTP/HTTPS abrufen. Dieser Datenpunkt kann browserbezogene Aktionen wie Klicken, Texteingabe, Navigieren durch Webseiten und andere Benutzerinteraktionen mit Websites oder Webanwendungen simulieren.

Zusätzlich zum Skript können eine optionale Liste von Parametern (Paare aus Name und Wert) sowie ein Timeout angegeben werden.

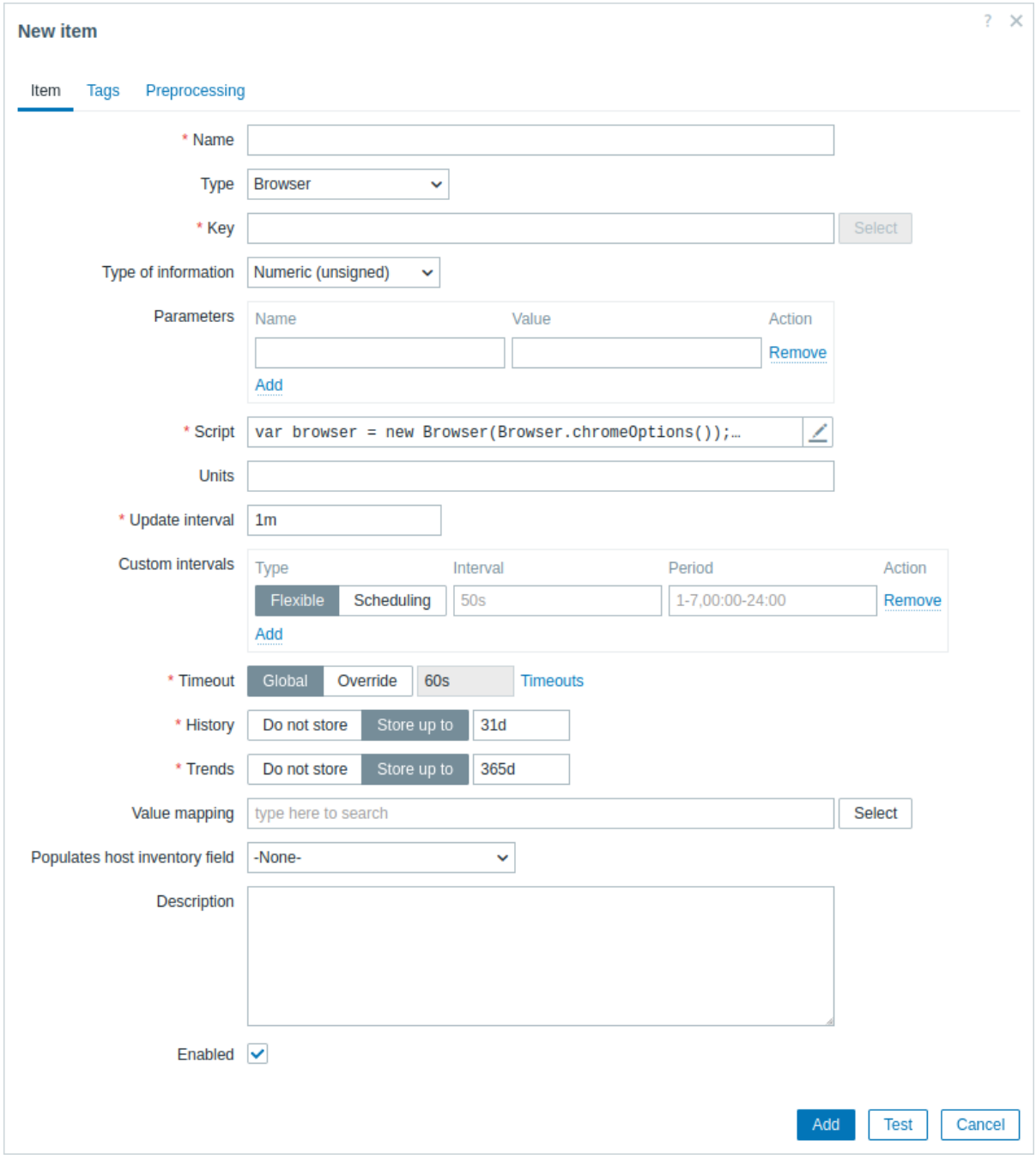
Attention:
Der Datenpunkt implementiert teilweise den W3C WebDriver-Standard mit entweder Selenium Server oder einem einfachen WebDriver (zum Beispiel ChromeDriver) als Webtest-Endpunkt. Damit der Datenpunkt funktioniert, setzen Sie den Parameter `WebDriverURL` in der Zabbix-Server/Proxy-Konfigurationsdatei (bei Verwendung von ChromeDriver siehe Security Considerations). Für eine bessere Leistung sollten Sie einen dedizierten Server für die Webtest-Umgebung in Betracht ziehen.

Browser-Datenpunktprüfungen werden von den Browser-Pollern des Zabbix-Servers oder -Proxys ausgeführt und verarbeitet. Falls erforderlich, können Sie die Anzahl der vorab erzeugten Instanzen der Browser-Poller im Parameter `StartBrowserPollers` der Zabbix-Server/Proxy-Konfigurationsdatei anpassen.

Für die Überwachung komplexer Websites und Webanwendungen steht die Vorlage Website by Browser als out-of-the-box-Vorlage zur Verfügung.

Konfiguration

Wählen Sie im Feld *Typ* des Datenpunkt-Konfigurationsformulars Browser aus und füllen Sie dann die erforderlichen Felder aus.



Alle obligatorischen Eingabefelder sind mit einem roten Sternchen markiert.

Die Felder, die für Browser-Datenpunkte spezifische Informationen erfordern, sind:

Feld	Beschreibung
Schlüssel	Geben Sie einen eindeutigen Schlüssel ein, der zur Identifizierung des Datenpunkts verwendet wird.
Parameter	Geben Sie die Variablen an, die als Attribut-Wert-Paare an das Skript übergeben werden sollen. Benutzermakros werden unterstützt. Um zu sehen, welche integrierten Makros unterstützt werden, suchen Sie in der Tabelle unterstützte Makros nach „Browser-Typ-Datenpunkt“.
Skript	Geben Sie JavaScript-Code im modalen Editor ein, der geöffnet wird, wenn Sie in das Parameterfeld klicken oder auf das Stiftsymbol daneben. Dieser Code muss die Logik zur Rückgabe des Metrikwerts bereitstellen. Der Code hat Zugriff auf alle Parameter, alle von Zabbix hinzugefügten zusätzlichen JavaScript-Objekte und JavaScript-Objekte für Browser-Datenpunkte. Siehe auch: JavaScript Guide.

Feld	Beschreibung
Timeout	Zeitüberschreitung für die JavaScript-Ausführung (1–600 s; bei Überschreitung wird ein Fehler zurückgegeben). Beachten Sie, dass es je nach Skript länger dauern kann, bis die Zeitüberschreitung ausgelöst wird. Weitere Informationen zum Parameter *Timeout* finden Sie unter allgemeine Datenpunktattribute.

Beispiele

Ein Beispiel dafür, wie Sie Zabbix für die Überwachung von Websites mit der Vorlage *Website by Browser* einrichten, finden Sie unter Websites mit Browser-Datenpunkten überwachen.

Standardskript

Das folgende Skript:

1. Initialisiert eine Browser-Sitzung.
2. Navigiert zu einer angegebenen URL.
3. Erfasst Performance-Einträge und Sitzungsstatistiken und gibt sie als JSON-Zeichenfolge zurück.

Geben Sie im Feld *Skript* Folgendes ein:

```
var browser = new Browser(Browser.chromeOptions());

try {
    browser.navigate("http://example.com");
    browser.collectPerfEntries();
}
finally {
    return JSON.stringify(browser.getResult());
}
```

Browser mit benutzerdefinierten Fähigkeiten initialisieren

Das folgende Skript:

1. Initialisiert eine Browser-Sitzung für den verfügbaren Browser basierend auf dem ersten übereinstimmenden Browser in der im Skript angegebenen Reihenfolge.
2. Definiert Browser-Fähigkeiten, einschließlich der Strategie zum Laden der Seite und browserspezifischer Optionen, wie z. B. den Headless-Modus für die Browser Chrome, Firefox und Microsoft Edge.

Geben Sie im Feld *Skript* Folgendes ein:

```
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"
            }
        ]
    }
});
```

Browser mit GUI initialisieren

Standardmäßig werden Browser-Sitzungen (außer Safari) im Headless-Modus initialisiert, d. h. die grafische Benutzeroberfläche (GUI) des Browsers wird nicht angezeigt.

Das folgende Skript initialisiert eine Browser-Sitzung mit aktivierter GUI.

Beachten Sie, dass Sie den Pfad manuell angeben können, wenn der WebDriver die Browser-Binärdatei nicht finden kann.

```
var opts = Browser.chromeOptions();
opts.capabilities.alwaysMatch['goog:chromeOptions'].args = [];

// To initialize a Firefox session with GUI, uncomment the following lines:
// var opts = Browser.firefoxOptions();
// opts.capabilities.alwaysMatch['moz:firefoxOptions'].binary = 'usr/bin/firefox';
// opts.capabilities.alwaysMatch['moz:firefoxOptions'].args = [];

// To initialize a Microsoft Edge session with GUI, uncomment the following lines:
// 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:**
> Wenn Ihre Tests auf einem Remote-Server oder in einem Container ausgeführt werden, können Sie einen Virtual Network Computing (VNC)-Client verwenden, um sich mit dem VNC-Server des Systems zu verbinden. So können Sie die GUI des Browsers aus der Ferne anzeigen und mit ihr interagieren.

Screenshots erstellen

Das folgende Skript:

1. Initialisiert eine Browser-Sitzung.
2. Legt die Größe des Browser-Viewports fest, um die Screenshot-Größe zu bestimmen (als Parameter angegeben, siehe unten).
3. Ruft eine URL auf (als Parameter angegeben, siehe unten).
4. Erfasst Sitzungsstatistiken, erstellt einen Screenshot und fügt ihn den erfassten Statistiken hinzu.
5. Behandelt Fehler, indem Fehlermeldungen und ein Screenshot erfasst werden.
6. Gibt die erfassten Ergebnisse als JSON-Zeichenfolge zurück.

Das Skript verwendet außerdem Parameter aus dem Datenpunkt-Konfigurationsformular:

- webURL - http://example.com
- width - 1920
- height - 1080

Geben Sie im Feld *Skript* Folgendes ein:

```
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);
}
```

Zabbix-Anmeldung prüfen

Das folgende Skript:

1. Initialisiert eine Browser-Sitzung.
2. Navigiert zu einer Seite (als Parameter angegeben, siehe unten).
3. Gibt den Benutzernamen und das Passwort ein (als Parameter angegeben, siehe unten).
4. Findet die Anmeldeschaltfläche und klickt darauf.
5. Findet die Abmeldeschaltfläche und klickt darauf.
6. Erfasst Performancedaten vor und nach der Anmeldung sowie nach der Abmeldung.
7. Behandelt Fehler, indem Fehlermeldungen und ein Screenshot erfasst werden.
8. Gibt die erfassten Ergebnisse als JSON-Zeichenfolge zurück.

Das Skript verwendet außerdem Parameter aus dem Datenpunkt-Konfigurationsformular:

- webURL - http://{HOST.CONN}/index.php
- username - {$USERNAME}
- password - {$PASSWORD}

Geben Sie im Feld *Script* Folgendes ein:

```
var browser, result;

browser = new Browser(Browser.chromeOptions());

try {
    var params = JSON.parse(value); // Parse the JSON string containing parameters passed from 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);
}
```

Links finden

Das folgende Skript:

1. Initialisiert eine Browser-Sitzung.
2. Definiert eine Funktion zum Entfernen doppelter Elemente aus einem Array (siehe Schritt 5).
3. Navigiert zu einer Seite (über Parameter angegeben, siehe unten).
4. Findet Links auf der Seite.
5. Entfernt doppelte Links, um sicherzustellen, dass sie eindeutig sind.
6. Extrahiert nur die Links, die mit "http" beginnen.
7. Formatiert die extrahierten Links in eine bestimmte Struktur.
8. Behandelt Fehler, indem Fehlermeldungen und ein Screenshot erfasst werden.
9. Gibt die gesammelten Ergebnisse als JSON-String zurück.

Das Skript verwendet außerdem Parameter aus dem Datenpunkt-Konfigurationsformular:

- scheme - {$WEBSITE.SCHEME}
- domain - {$WEBSITE.DOMAIN}
- path - {$WEBSITE.PATH}

Geben Sie im Feld *Script* Folgendes ein:

```
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);
}
```

Telemetrieabfrage

Mit diesem Datentyp können Sie Telemetriedaten abrufen oder aggregieren. Telemetriedaten werden für das Application Performance Monitoring (APM) verwendet. Damit können Sie nachvollziehen, wie Anforderungen durch Ihre Anwendungen und Dienste fließen, deren Leistung messen und die Ursache für langsame oder fehlgeschlagene Vorgänge ermitteln.

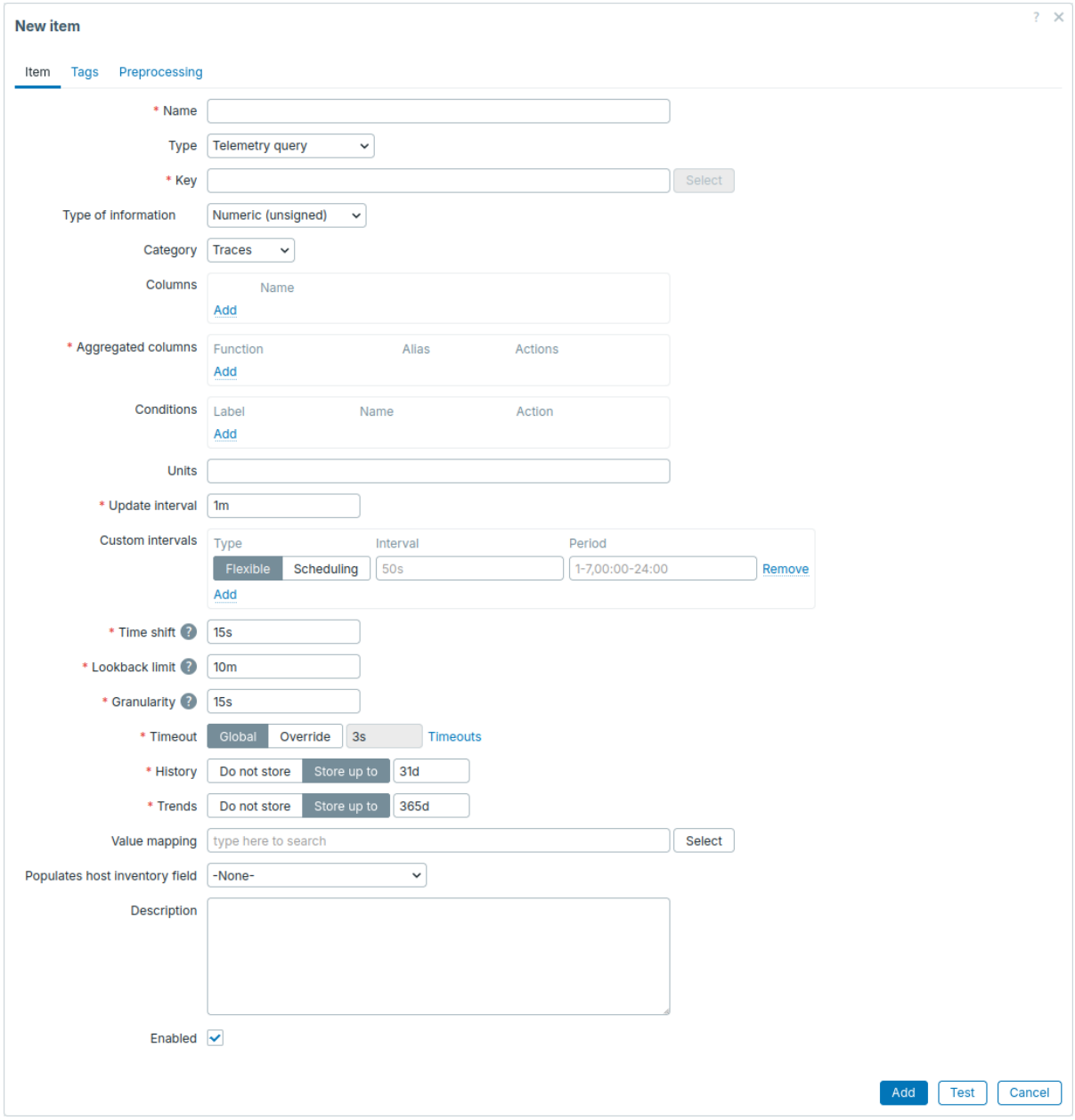
Telemetriedaten gibt es in drei Typen: Traces (der Pfad, den eine Anforderung durch Ihre Anwendungen und Dienste nimmt), Metriken (Messwerte wie die Antwortzeit) und Logs (Textaufzeichnungen von Ereignissen). Mit dem Datenpunkt für Telemetrieabfragen können Sie beispielsweise alle Traces finden, bei denen ein Dienst einen Fehler zurückgegeben hat, die durchschnittliche Antwortzeit aus Ihren Metriken ermitteln oder Ihre Logs nach einer bestimmten Fehlermeldung durchsuchen.

Wenn Sie einen Datenpunkt für Telemetrieabfragen konfigurieren, definieren Sie eine Reihe von Anweisungen (eine Abfrage), die Zabbix mitteilen, welche Telemetriedaten abgerufen, wie sie gefiltert und wie sie aggregiert werden sollen. Bei der Ausführung der Datenpunktprüfung sendet Zabbix diese Abfrage an die Telemetriedatenquelle und ruft das Ergebnis auf Grundlage der Abfrage ab.

Um diesen Datenpunkt zu verwenden, geben Sie die Telemetriedatenquelle an und setzen Sie den Parameter `StartTelemetryQueryPollers` in der Konfiguration des Zabbix-Servers oder des Zabbix-Proxys, je nachdem, welcher von beiden den Host des Datenpunkts überwacht.

Konfiguration

Beim Erstellen eines Datenpunkts legen Sie dessen Typ auf *Telemetry query* fest:



Parameter	Beschreibung
Name	Siehe Parameter, die für alle Datenpunkte gelten.
Type	Wählen Sie *Telemetry query*.
Key	Siehe Parameter, die für alle Datenpunkte gelten.
Type of information	
Category	Wählen Sie den Typ des Signals, den Sie abfragen möchten: *Traces*, *Metrics* oder *Logs*.
Metric points	Wählen Sie den Typ des Metrikpunkts, den Sie abfragen möchten: *Sum*, *Gauge*, *Histogram* oder *Exponential histogram*. Dieser Parameter ist verfügbar, wenn *Category* auf *Metrics* gesetzt ist.
Columns	Fügen Sie Spalten mit Telemetriedaten hinzu, die Sie abfragen möchten. Geben Sie für Spalten mit Schlüssel-Wert-Paaren zusätzlich den Schlüsselnamen ein. Die Datensätze aus diesen Spalten werden unverändert in den Wert des Datenpunkts aufgenommen.

Parameter	Beschreibung
Aggregated columns	Fügen Sie Spalten mit Telemetriedaten hinzu, die Sie abfragen möchten, und legen Sie fest, wie aus ihren Datensätzen ein Wert berechnet werden soll: **Function** – Wählen Sie die auf die Spaltendatensätze anzuwendende Funktion: *min*, *max*, *avg*, *count*, *sum* oder *percentile*. **Percentage** – Geben Sie das aus den Spaltendatensätzen zu berechnende Perzentil ein (wenn *Function* auf *percentile* gesetzt ist). **Column** – Wählen Sie die Spalte aus, aus der die Berechnung erfolgen soll (wenn *Function* nicht auf *count* gesetzt ist). **Alias** – Geben Sie einen eindeutigen Namen für den berechneten Wert ein, der zur Identifizierung im Wert des Datenpunkts verwendet wird.
Type of calculation	Wählen Sie aus, wie die im folgenden Parameter *Conditions* festgelegten Bedingungen ausgewertet werden sollen: **And/Or** – Alle Bedingungen müssen erfüllt sein; Bedingungen mit demselben *Type* und *Attribute key* werden mit *Or* verknüpft. **And** – Alle Bedingungen müssen erfüllt sein. **Or** – Mindestens eine Bedingung muss erfüllt sein. **Custom expression** – Die Bedingungen müssen wie in der Formel angegeben erfüllt sein, zum Beispiel `A and (B or C)`.
Conditions	Dieser Parameter ist verfügbar, wenn Sie zwei oder mehr Bedingungen festgelegt haben. Fügen Sie Bedingungen zum Filtern von Datensätzen aus Spalten mit Telemetriedaten hinzu: **Type** – Wählen Sie die Spalte mit Telemetriedaten aus, auf die die Bedingung angewendet werden soll. **Attribute key** – Geben Sie den Schlüssel ein, auf den die Bedingung angewendet werden soll (wenn *Type* auf eine Spalte mit Schlüssel-Wert-Paaren gesetzt ist). **Operator** – Wählen Sie den zu verwendenden Vergleich (siehe unten) zwischen dem Wert des Datensatzes und *Attribute value* aus. **Attribute value** – Geben Sie den Wert ein, mit dem der Wert des Datensatzes verglichen werden soll (wenn *Operator* nicht auf *Exists* gesetzt ist). Wenn *Type* auf eine Spalte mit einem einzelnen Wert gesetzt ist, wird die Bedingung auf den Datensatz der Spalte angewendet. Wenn die Spalte ein Array von Werten enthält, stimmt der Datensatz mit der Bedingung überein, wenn mindestens ein Wert im Array die Bedingung erfüllt. Wenn *Type* auf eine Spalte mit Schlüssel-Wert-Paaren gesetzt ist, wird die Bedingung auf den Wert des unter *Attribute key* angegebenen Schlüssels angewendet. Wenn die Spalte ein Array von Schlüssel-Wert-Paaren enthält, stimmt der Datensatz mit der Bedingung überein, wenn mindestens ein Paar mit dem angegebenen *Attribute key* die Bedingung erfüllt. Für Spalten mit einzelnen Werten können Sie folgende Operatoren verwenden: **Equals** – Datensätze auswählen, die *Attribute value* entsprechen (Groß-/Kleinschreibung wird berücksichtigt). **Does not equal** – Datensätze auswählen, die nicht *Attribute value* entsprechen (Groß-/Kleinschreibung wird berücksichtigt). **Contains** – Datensätze auswählen, die *Attribute value* enthalten (Groß-/Kleinschreibung wird berücksichtigt). **Does not contain** – Datensätze auswählen, die *Attribute value* nicht enthalten (Groß-/Kleinschreibung wird berücksichtigt). Für Spalten mit Schlüssel-Wert-Paaren können Sie folgende Operatoren verwenden: **Equals** – Datensätze auswählen, bei denen *Attribute key* vorhanden ist und dessen Wert *Attribute value* entspricht (Groß-/Kleinschreibung wird berücksichtigt). **Does not equal** – Datensätze auswählen, bei denen *Attribute key* vorhanden ist und dessen Wert nicht *Attribute value* entspricht (Groß-/Kleinschreibung wird berücksichtigt). **Exists** – Datensätze auswählen, bei denen *Attribute key* vorhanden ist (Groß-/Kleinschreibung wird berücksichtigt).
Units *Update interval* *Custom intervals*	Siehe Parameter, die für alle Datenpunkte gelten.

Parameter	Beschreibung
Time shift	Geben Sie an, wie weit vor der aktuellen Zeit der Zeitraum enden soll, für den der Datenpunkt Daten abfragt. Unterstützt eine Anzahl von Sekunden (z. B. 1, 60, 300), einen Wert mit einem Zeitsuffix (z. B. 30s, 1m, 2h, 1d) oder Benutzermakros. Jede Abfrage fordert Daten für einen Zeitraum an. Der Zeitraum beginnt normalerweise dort, wo der Zeitraum der vorherigen Abfrage endete, und endet an dem durch *Time shift* festgelegten Punkt. Wenn der Zeitraum der vorherigen Abfrage später als die aktuelle Zeit minus *Granularity* endete, beginnt der Zeitraum der aktuellen Abfrage bei der aktuellen Zeit minus *Granularity* und überschneidet sich somit mit dem Zeitraum der vorherigen Abfrage. Wenn keine vorherige Abfrage vorhanden ist, an die angeknüpft werden kann, beginnt der Zeitraum an dem durch *Lookback limit* festgelegten Punkt. Wenn der Host dieses Datenpunkts von einer anderen Entität überwacht wird (z. B. von einem anderen Proxy aus einer Proxy-Gruppe), weiß der Datenpunkt möglicherweise nicht mehr, wann die vorherige Abfrage endete. Dies kann zu doppelten Werten führen.
Lookback limit	Geben Sie an, wie weit vor der aktuellen Zeit der Datenpunkt Daten abfragt, wenn keine vorherige Abfrage vorhanden ist, an die angeknüpft werden kann (z. B. beim ersten Lauf des Datenpunkts), oder wenn die vorherige Abfrage älter als *Update interval* ist (z. B. weil der Zabbix Server nicht verfügbar war). Unterstützt eine Anzahl von Sekunden (z. B. 1, 60, 300), einen Wert mit einem Zeitsuffix (z. B. 30s, 1m, 2h, 1d) oder Benutzermakros. Wenn die Telemetriedatenquelle eine große Datenmenge enthält (z. B. mehr als eine Million Datensätze pro Stunde), kann ein großer Wert für *Lookback limit* die Leistung bei Verwendung dieses Parameters durch den Datenpunkt erheblich beeinträchtigen.
Granularity	Geben Sie die Länge eines Abschnitts an, in den der gesamte Zeitraum der Abfrage aufgeteilt wird. Unterstützt eine Anzahl von Sekunden (z. B. 1, 60, 300), einen Wert mit einem Zeitsuffix (z. B. 30s, 1m, 2h, 1d) oder Benutzermakros.
Timeout *History* *Trends* *Value mapping* *Populates host inventory field* *Description* *Enabled*	Siehe Parameter, die für alle Datenpunkte gelten.

Unterstützte Spalten für Telemetriedaten

In diesem Abschnitt werden die Spalten aufgeführt, die in den folgenden Parametern des Datenpunkts unterstützt werden: *Columns*, *Aggregated columns* und *Conditions*. Dabei handelt es sich um die Spalten, die von ClickHouse zum Speichern von Telemetriedaten gemäß dem Schema des OpenTelemetry-ClickHouse-Exporters verwendet werden.

Wählen Sie die *Category*:

- Traces
- Metrics: Sum
- Metrics: Gauge
- Metrics: Histogram
- Metrics: Exponential histogram
- Logs

Traces

Spalte	*Spalten*	*Aggregierte Spalten*	*Bedingungen*
Dauer	✓	✓	–
Events.Attributes	–	–	✓
Events.Name	–	–	✓
ParentSpanId	✓	–	✓
ResourceAttributes	✓	–	✓
ScopeName	✓	–	✓
ScopeVersion	✓	–	✓

Spalte	*Spalten*	*Aggregierte Spalten*	*Bedingungen*
ServiceName	✓	–	✓
SpanAttributes	✓	–	✓
SpanId	✓	–	✓
SpanKind	✓	–	✓
SpanName	✓	–	✓
StatusCode	✓	–	✓
StatusMessage	✓	–	✓
Zeitstempel	✓	✓	–
TraceId	✓	–	✓
TraceState	✓	–	✓

Metriken: Summe

Spalte	*Spalten*	*Aggregierte Spalten*	*Bedingungen*
AggregationTemporality	✓	–	–
Attributes	✓	–	✓
Exemplars.FilteredAttributes	–	–	✓
Flags	✓	–	–
IsMonotonic	✓	–	–
MetricDescription	✓	–	✓
MetricName	✓	–	✓
MetricUnit	✓	–	✓
ResourceAttributes	✓	–	✓
ResourceSchemaUrl	✓	–	✓
ScopeAttributes	✓	–	✓
ScopeDroppedAttrCount	✓	✓	–
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
ServiceName	✓	–	✓
StartTimeUnix	✓	✓	–
TimeUnix	✓	✓	–
Value	✓	✓	–

Metriken: Gauge

Spalte	*Spalten*	*Aggregierte Spalten*	*Bedingungen*
Attribute	✓	–	✓
Exemplars.FilteredAttributes	–	–	✓
Flags	✓	–	–
MetricDescription	✓	–	✓
MetricName	✓	–	✓
MetricUnit	✓	–	✓
ResourceAttributes	✓	–	✓
ResourceSchemaUrl	✓	–	✓
ScopeAttributes	✓	–	✓
ScopeDroppedAttrCount	✓	✓	–
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
ServiceName	✓	–	✓
StartTimeUnix	✓	✓	–
TimeUnix	✓	✓	–
Value	✓	✓	–

Metriken: Histogramm

Spalte	*Spalten*	*Aggregierte Spalten*	*Bedingungen*
AggregationTemporality	✓	–	–
Attributes	✓	–	✓
Count	✓	✓	–
Exemplars.FilteredAttributes	–	–	✓
Flags	✓	–	–
Max	✓	✓	–
MetricDescription	✓	–	✓
MetricName	✓	–	✓
MetricUnit	✓	–	✓
Min	✓	✓	–
ResourceAttributes	✓	–	✓
ResourceSchemaUrl	✓	–	✓
ScopeAttributes	✓	–	✓
ScopeDroppedAttrCount	✓	✓	–
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
ServiceName	✓	–	✓
StartTimeUnix	✓	✓	–
Sum	✓	✓	–
TimeUnix	✓	✓	–

Metriken: Exponentielles Histogramm

Spalte	*Spalten*	*Aggregierte Spalten*	*Bedingungen*
AggregationTemporality	✓	–	–
Attributes	✓	–	✓
Count	✓	✓	–
Exemplars.FilteredAttributes	–	–	✓
Flags	✓	–	–
Max	✓	✓	–
MetricDescription	✓	–	✓
MetricName	✓	–	✓
MetricUnit	✓	–	✓
Min	✓	✓	–
NegativeOffset	✓	✓	–
PositiveOffset	✓	✓	–
ResourceAttributes	✓	–	✓
ResourceSchemaUrl	✓	–	✓
Scale	✓	✓	–
ScopeAttributes	✓	–	✓
ScopeDroppedAttrCount	✓	✓	–
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
ServiceName	✓	–	✓
StartTimeUnix	✓	✓	–
Sum	✓	✓	–
TimeUnix	✓	✓	–
ZeroCount	✓	✓	–

Protokolle

Spalte	*Spalten*	*Aggregierte Spalten*	*Bedingungen*
Body	✓	–	✓
EventName	✓	–	✓
LogAttributes	✓	–	✓
ResourceAttributes	✓	–	✓

Spalte	*Spalten*	*Aggregierte Spalten*	*Bedingungen*
ResourceSchemaUrl	✓	–	✓
ScopeAttributes	✓	–	✓
ScopeName	✓	–	✓
ScopeSchemaUrl	✓	–	✓
ScopeVersion	✓	–	✓
SpanId	✓	–	✓
ServiceName	✓	–	✓
SeverityNumber	✓	✓	–
SeverityText	✓	–	✓
Timestamp	✓	✓	–
TraceFlags	✓	–	–
TraceId	✓	–	✓

Verlauf und Trends

Übersicht

Verlauf und Trends sind die zwei Arten, gesammelte Daten in Zabbix zu speichern.

Während der Verlauf jeden erfassten Wert speichert, enthalten Trends stündlich gemittelte Informationen und benötigen daher weniger Ressourcen.

Verlaufsspeicherung

Sie können festlegen, für wie viele Tage der Verlauf gespeichert wird:

- in den Eigenschaften des Datenpunkts Formular
- bei der Massenaktualisierung von Datenpunkten
- beim Einrichten von Housekeeper-Aufgaben

Ältere Daten werden vom Housekeeper entfernt.

Generell wird dringend empfohlen, den Verlauf für die kleinstmögliche Anzahl von Tagen zu speichern, um die Datenbank nicht mit einer großen Menge historischer Werte zu überlasten.

Anstatt einen langen Verlauf zu speichern, können Sie Trenddaten länger aufbewahren. Zum Beispiel könnten Sie den Verlauf 14 Tage und Trends 5 Jahre lang speichern.

Einen guten Eindruck davon, wie viel Speicherplatz Verlaufsdaten im Vergleich zu Trenddaten benötigen, erhalten Sie auf der Seite zur Datenbankdimensionierung.

Auch bei einer kürzeren Verlaufsspeicherung können Sie ältere Daten weiterhin in Diagrammen überprüfen, da Diagramme für die Anzeige älterer Daten Trendwerte verwenden.

Attention:
Wenn der Verlauf auf „0“ gesetzt ist, aktualisiert der Datenpunkt nur abhängige Datenpunkte und das Inventar. Es werden keine Auslöser-Funktionen ausgewertet, da die Auswertung von Auslösern ausschließlich auf Verlaufsdaten basiert.

Note:
Als alternative Möglichkeit zur Aufbewahrung des Verlaufs können Sie die Funktion Verlaufsexport von ladbaren Modulen verwenden.

Trends beibehalten

Trends sind ein integrierter Mechanismus zur Reduzierung historischer Daten, der für numerische Datentypen pro Stunde den Minimalwert, Maximalwert, Durchschnittswert und die Gesamtzahl der Werte speichert.

Sie können festlegen, für wie viele Tage Trends aufbewahrt werden:

- im Formular der Datenpunkt-Eigenschaften
- beim Massenaktualisieren von Datenpunkten
- beim Einrichten von Housekeeper-Aufgaben

Trends können in der Regel deutlich länger aufbewahrt werden als die Historie. Ältere Daten werden vom Housekeeper entfernt.

Der Zabbix Server sammelt Trenddaten zur Laufzeit im Trend-Cache, während die Daten eingehen. Der Server schreibt die Trends der **vorherigen Stunde** jedes Datenpunkts in die Datenbank (wo das Frontend sie finden kann) in folgenden Situationen:

- der Server empfängt den ersten Wert des Datenpunkts in der aktuellen Stunde
- es verbleiben 5 oder weniger Minuten der aktuellen Stunde und es liegen noch immer keine Werte des Datenpunkts für die aktuelle Stunde vor
- der Server wird gestoppt

Um Trends in einem Diagramm zu sehen, müssen Sie mindestens bis zum Beginn der nächsten Stunde warten (wenn der Datenpunkt häufig aktualisiert wird) und höchstens bis zum Ende der nächsten Stunde (wenn der Datenpunkt selten aktualisiert wird), also maximal 2 Stunden.

Wenn der Server den Trend-Cache schreibt und für diese Stunde bereits Trends in der Datenbank vorhanden sind (zum Beispiel wenn der Server mitten in der Stunde neu gestartet wurde), muss der Server Update-Statements anstelle einfacher Inserts verwenden. Daher ist es bei einer größeren Installation, wenn ein Neustart erforderlich ist, wünschenswert, den Server am Ende einer Stunde zu stoppen und zu Beginn der nächsten Stunde zu starten, um Überschneidungen bei Trenddaten zu vermeiden.

Historientabellen sind in keiner Weise an der Trendgenerierung beteiligt.

Attention:
Wenn Trends auf '0' gesetzt sind, berechnet oder speichert der Zabbix Server überhaupt keine Trends.

Note:
Die Trends werden mit demselben Datentyp wie die ursprünglichen Werte berechnet und gespeichert. Daher werden Durchschnittswertberechnungen von Werten des vorzeichenlosen Datentyps gerundet, und je kleiner das Werteintervall ist, desto ungenauer wird das Ergebnis. Wenn ein Datenpunkt beispielsweise die Werte 0 und 1 hat, ist der Durchschnittswert 0 und nicht 0,5.
Auch ein Neustart des Servers kann zu einem Präzisionsverlust bei den Durchschnittswertberechnungen des vorzeichenlosen Datentyps für die aktuelle Stunde führen.

Benutzerparameter

Overview

A user parameter lets you define a custom item key and the command used to retrieve its value when no built-in Zabbix agent item provides the data you need. A user parameter is a user-defined command referenced by a user-defined item key and executed by the Zabbix agent when the corresponding item is checked.

Note:
On UNIX systems, the agent executes user parameter commands using `/bin/sh`, under the operating system account running the agent (usually `zabbix`). That account must have sufficient permissions to execute the command and access any files, sockets, or credentials it uses.

There are two types of user parameters:

- **Simple user parameters** do not accept arguments; they define a fixed item key and a command to execute.
- **Flexible user parameters** accept arguments (specified as parameters in the item key) and pass them to the command, allowing a single definition to serve multiple items.

User parameter configuration

A user parameter is defined using the `UserParameter` configuration parameter.

A user parameter has the following syntax:

```
UserParameter=<key>,<shell command>
```

where:

- <key> - a unique item key that identifies the user parameter.
 When configuring an item, specify this key to reference the command defined by the user parameter.
 For a simple user parameter, the key is fixed and does not accept arguments. For a flexible user parameter, append `[*]` to the key (for example, `mysql.status[*]`) to indicate that it accepts arguments. Specify these arguments as parameters in the item key when configuring the item. The command can refer to the parameters using positional references $1-$9.
- <shell command> - a shell command or a path to a custom script that the agent executes to obtain the item value.
 For the flexible user parameter, the command can use positional references $1-$9 to refer to the corresponding parameters in the item key (for example, `UserParameter=mysql.status[*],mysqladmin status --user=$1 --password=$2`). Zabbix parses the parameters enclosed in `[]` in the item key (i.e., `mysql.status[zabbix,{$MYSQL.PASSWORD}]`) and substitutes the positional references in the command accordingly.
 $0 is substituted with the original command before

positional references are expanded. Positional references are interpreted regardless of whether they are enclosed in double (") or single (') quotes. To use positional references without substitution, specify a double dollar sign. For example, `awk '{print $$2}'` passes $2 to the command unchanged.
 Note that positional references prefixed with $ are searched for and replaced by the Zabbix agent only in flexible user parameters. For simple user parameters, positional references are not processed, so escaping $ is unnecessary.

Attention:
Certain characters cannot be passed as arguments by default. For the complete list of characters, see UnsafeUserParameters.

`UserParameter` must be configured on each Zabbix agent that uses it: - Directly in the Zabbix agent configuration file (`zabbix_agentd.conf` or `zabbix_agent2.conf`). - In a separate `*.conf` file, provided that the file is included by the agent configuration.

More than one `UserParameter` can be defined per agent, as multiple files can be included with different sets of parameters. Each user parameter must have a unique item key within the monitored host.

The `UserParameterDir` configuration parameter lets you specify the working directory for executing `UserParameter` commands. If configured, the agent changes its working directory to the specified directory before executing a command. Commands can be specified using a relative path with the `./` prefix, for example, `UserParameter=mykey,./myscript.sh`.

To configure a user parameter:

1. Write a custom script or command to retrieve the required data.
2. Add the command and its unique item key to the `UserParameter` configuration parameter in the Zabbix agent configuration file (`zabbix_agentd.conf`).
3. Reload user parameters from the configuration file by running the runtime control command:

   zabbix_agentd -R userparameter_reload

   Alternatively, restart the agent to apply the configuration changes.
4. Create a new item on the monitored host with the item key defined in `UserParameter`.
 Set the item's type to *Zabbix Agent* or *Zabbix Agent (active)*.
 Make sure that the item's *Type of information* is configured correctly; otherwise, Zabbix will reject the returned value.

See Extending Zabbix agents for an example of configuring and using user parameters.

User parameters are subject to the Zabbix agent check timeout. If the timeout is reached, the process executing the user parameter is terminated. See Command execution for more information.

Note that long-running scripts may consume system resources and delay other checks.

Befehlsergebnis

Der Rückgabewert des Befehls ist die Standardausgabe zusammen mit der Standardfehlerausgabe, die vom Befehl erzeugt wird.

Attention:
Ein Datenpunkt, der Text zurückgibt (Zeichen-, Log- oder Texttyp von Informationen), wird im Fall einer Standardfehlerausgabe nicht zu „nicht unterstützt".

Der Rückgabewert ist auf 16 MB begrenzt (einschließlich nachgestellter Leerzeichen, die abgeschnitten werden); Datenbankbeschränkungen gelten ebenfalls.

Benutzerparameter, die Text zurückgeben (Zeichen-, Log- oder Texttyp von Informationen), können ebenfalls Leerraum zurückgeben. Im Fall eines ungültigen Ergebnisses wird der Datenpunkt zu „nicht unterstützt".

Beispiele für einfache Benutzerparameter

Ein einfacher Befehl:

```
UserParameter=ping,echo 1
```

Der Agent wird immer eine "1" für ein Element mit dem Schlüssel "ping" zurückgeben.

Ein komplexeres Beispiel:

```
UserParameter=mysql.ping,mysqladmin -uroot ping | grep -c alive
```

Der Agent gibt '1' zurück, wenn der MySQL-Server aktiv ist, andernfalls '0'.

Beispiel 1

Etwas sehr Einfaches:

```
UserParameter=ping[*],echo $1
```

Wir können eine unbegrenzte Anzahl von Elementen für die Überwachung definieren, die alle das folgende Format haben ping[something].

- ping[0] - wird immer '0' zurückgeben
- ping[aaa] - wird immer 'aaa' zurückgeben

Beispiel 2

Lassen Sie uns mehr Sinn hinzufügen!

```
UserParameter=mysql.ping[*],mysqladmin -u$1 -p$2 ping | grep -c alive
```

Dieser Parameter kann zur Überwachung der Verfügbarkeit einer MySQL-Datenbank verwendet werden. Wir können Benutzernamen und Passwort übergeben:

```
mysql.ping[zabbix,our_password]
```

Beispiel 3

Wie viele Zeilen in einer Datei entsprechen einem regulären Ausdruck?

```
UserParameter=wc[*],grep -c "$2" $1
```

Dieser Parameter kann verwendet werden, um die Anzahl der Zeilen in einer Datei zu berechnen.

```
wc[/etc/passwd,root]
wc[/etc/services,zabbix]
```

Erweiterung von Zabbix Agenten

Dieses Tutorial bietet eine Schritt-für-Schritt-Anleitung, wie die Funktionalität von Zabbix Agent mithilfe eines UserParameter erweitert werden kann.

Schritt 1

Schreiben Sie ein Skript oder eine Befehlszeile, um den erforderlichen Parameter abzurufen.

Beispielsweise können wir den folgenden Befehl schreiben, um die Gesamtzahl der von einem MySQL-Server ausgeführten Abfragen zu ermitteln:

```
mysqladmin -uroot status | cut -f4 -d":" | cut -f1 -d"S"
```

Bei der Ausführung gibt der Befehl die Gesamtzahl der SQL-Abfragen zurück.

Schritt 2

Fügen Sie den Befehl zur Agent-Konfigurationsdatei von Zabbix (`zabbix_agentd.conf`) hinzu:

```
UserParameter=mysql.questions,mysqladmin -uroot status | cut -f4 -d":" | cut -f1 -d"S"
```

mysql.questions ist ein eindeutiger Bezeichner. Er kann jeder gültige Schlüsselbezeichner sein, zum Beispiel *queries*.

Testen Sie diesen Parameter, indem Sie den Zabbix Agent mit dem Flag `-t` verwenden (wenn er jedoch unter root ausgeführt wird, beachten Sie, dass der Agent beim Start als Daemon möglicherweise über andere Berechtigungen verfügt):

```
zabbix_agentd -t mysql.questions
```

Schritt 3

Laden Sie die Benutzerparameter aus der Konfigurationsdatei neu, indem Sie Folgendes ausführen:

```
zabbix_agentd -R userparameter_reload
```

Sie können den Agent auch neu starten, anstatt den Laufzeitsteuerungsbefehl zu verwenden.

Testen Sie den Parameter mit dem Dienstprogramm `zabbix_get`.

Schritt 4

Fügen Sie dem überwachten Host einen neuen Datenpunkt mit `Key=mysql.questions` hinzu. Der Typ des Datenpunkts muss entweder *Zabbix Agent* oder *Zabbix Agent (active)* sein.

Beachten Sie, dass der Typ der zurückgegebenen Werte auf dem Zabbix Server korrekt gesetzt sein muss. Andernfalls akzeptiert Zabbix sie nicht.

Windows-Leistungsindikatoren

Übersicht

Mit dem Schlüssel `perf_counter[]` können Sie Windows-Leistungsindikatoren effektiv überwachen.

Zum Beispiel:

```
perf_counter["\Processor(0)\Interrupts/sec"]
```

oder

```
perf_counter["\Processor(0)\Interrupts/sec", 10]
```

Weitere Informationen zur Verwendung dieses Schlüssels oder seines englischen Äquivalents `perf_counter_en` finden Sie unter Windows-spezifische Datenpunktschlüssel.

Um eine vollständige Liste der zur Überwachung verfügbaren Leistungsindikatoren abzurufen, können Sie Folgendes ausführen:

```
typeperf -qx
```

Sie können auch die Low-Level-Erkennung verwenden, um mehrere Objektinstanzen von Windows-Leistungsindikatoren zu erkennen und die Erstellung von `perf_counter`-Datenpunkten für mehrere Instanzobjekte zu automatisieren.

Numerische Darstellung

Windows verwaltet numerische Darstellungen (Indizes) für Namen von Objekten und Leistungsindikatoren. Zabbix unterstützt diese numerischen Darstellungen als Parameter für die Datanpunkt-Schlüssel `perf_counter`, `perf_counter_en` sowie in den Konfigurationsparametern `PerfCounter`, `PerfCounterEn`.

Es wird jedoch nicht empfohlen, diese zu verwenden, es sei denn, Sie können garantieren, dass Ihre numerischen Indizes auf bestimmten Hosts den korrekten Zeichenfolgen entsprechen. Wenn Sie portable Datenpunkte erstellen müssen, die auf verschiedenen Hosts mit unterschiedlich lokalisierten Windows-Versionen funktionieren, können Sie den Schlüssel `perf_counter_en` oder den Konfigurationsparameter `PerfCounterEn` verwenden. Damit können Sie englische Namen unabhängig von der Systemgebietsschema verwenden.

Um die numerischen Entsprechungen zu ermitteln, führen Sie `regedit` aus und suchen Sie anschließend nach *Counter* unter `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009`.

Der Registrierungseintrag enthält Informationen wie diese:

```
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
....
```

Hier finden Sie die entsprechenden Nummern für jeden Zeichenfolgenteil des Leistungsindikatorpfads, beispielsweise in `\System\% Processor Time`:

```
System → 2
% Processor Time → 6
```

Anschließend können Sie diese Nummern verwenden, um den Pfad numerisch darzustellen:

```
\2\6
```

Parameter für Leistungsindikatoren

Sie können einige PerfCounter-Parameter für die Überwachung von Windows-Leistungsindikatoren bereitstellen.

Beispielsweise können Sie diese zur Konfigurationsdatei des Zabbix Agent hinzufügen:

```
PerfCounter=UserPerfCounter1,"\Memory\Page Reads/sec",30
oder
PerfCounter=UserPerfCounter2,"\4\24",30
```

Mit solchen Parametern können Sie anschließend einfach `UserPerfCounter1` oder `UserPerfCounter2` als Schlüssel zum Erstellen der entsprechenden Datenpunkte verwenden.

Denken Sie daran, den Zabbix Agent nach Änderungen an der Konfigurationsdatei neu zu starten.

Massenaktualisierung

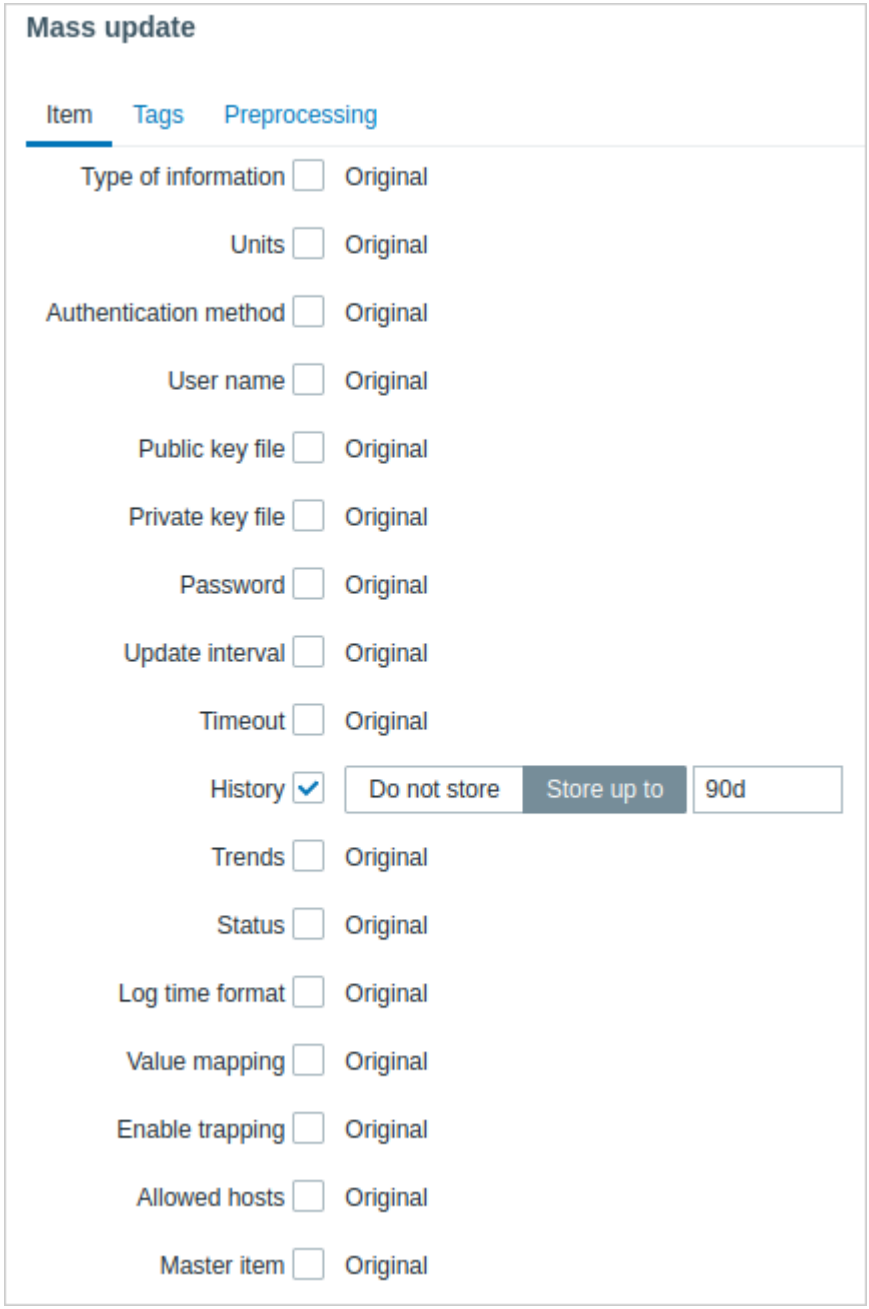
Übersicht

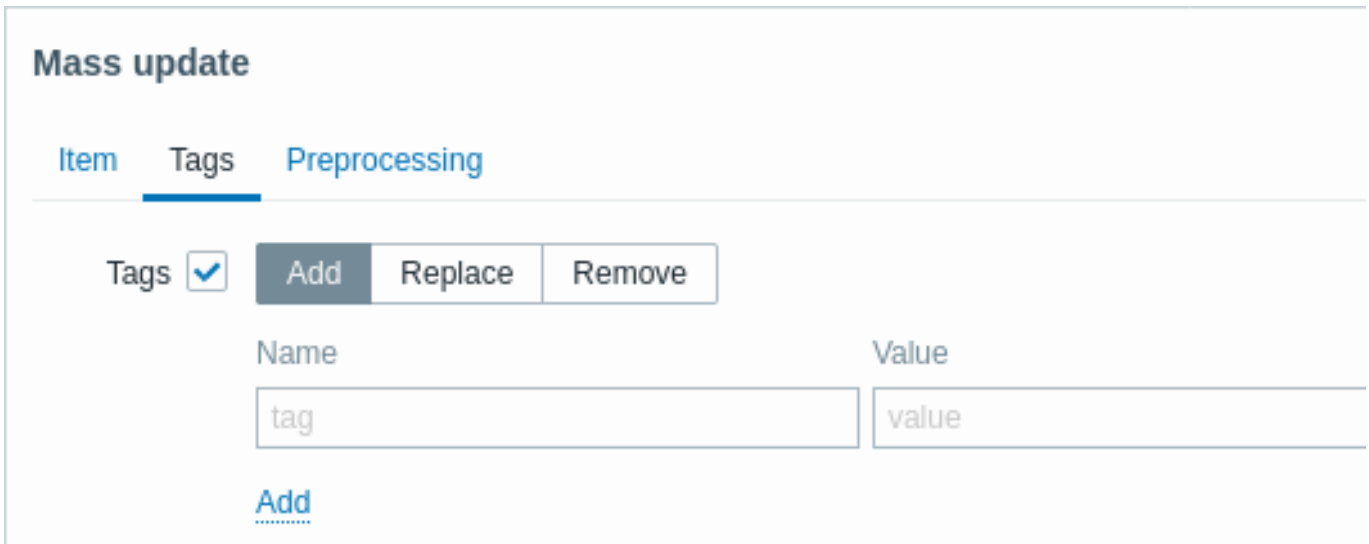
Manchmal möchten Sie ein Attribut für mehrere Datenpunkte gleichzeitig ändern. Anstatt jeden einzelnen Datenpunkt zur Bearbeitung zu öffnen, können Sie dafür die Funktion zur Massenaktualisierung verwenden.

Massenaktualisierung verwenden

Um mehrere Datenpunkte gleichzeitig zu aktualisieren, gehen Sie wie folgt vor:

- Aktivieren Sie in der Liste die Kontrollkästchen der zu aktualisierenden Datenpunkte.
- Klicken Sie unterhalb der Liste auf *Massenaktualisierung*.
- Wechseln Sie zu der Registerkarte mit den erforderlichen Attributen (*Datenpunkt*, *Tags* oder *Vorverarbeitung*).
- Aktivieren Sie die Kontrollkästchen der zu aktualisierenden Attribute.
- Geben Sie neue Werte für die Attribute ein.



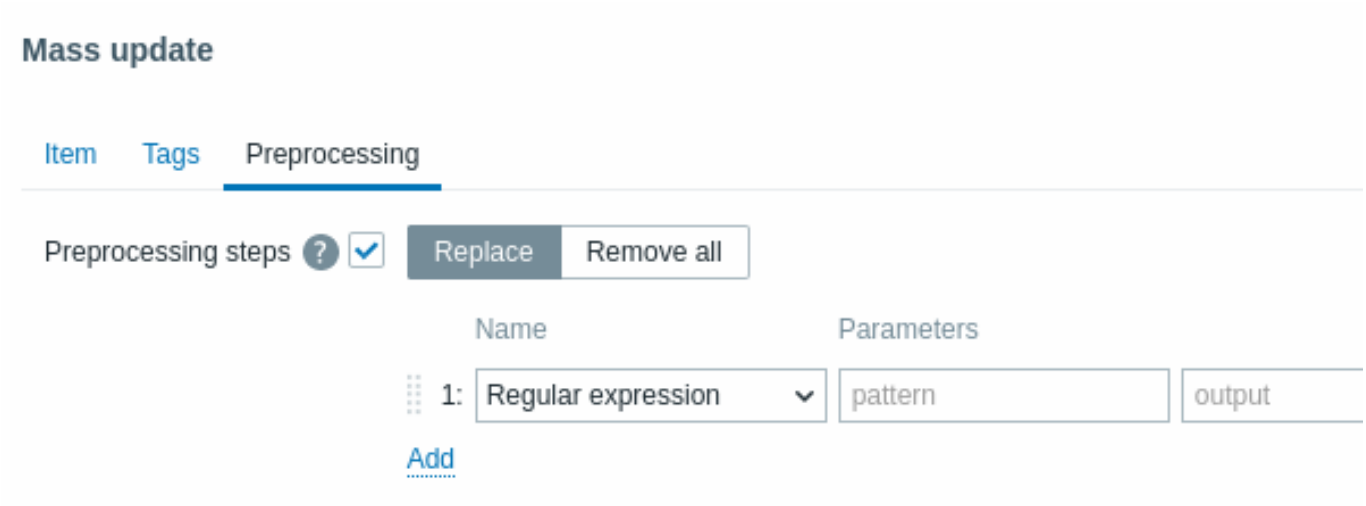


Mit der Option *Tags* können Sie:

- *Hinzufügen* – die angegebenen Tags zu den Datenpunkten hinzufügen (Tags mit demselben Namen, aber unterschiedlichen Werten gelten nicht als „Duplikate“ und können demselben Datenpunkt hinzugefügt werden).
- *Ersetzen* – die angegebenen Tags entfernen und Tags mit neuen Werten hinzufügen

- *Entfernen* – die angegebenen Tags aus den Datenpunkten entfernen

Benutzermakros, {INVENTORY.*}-Makros, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} und {HOST.ID}-Makros werden in Tags unterstützt.



Aktivieren Sie das Kontrollkästchen neben *Vorverarbeitungsschritte*, um die Massenaktualisierung für Vorverarbeitungsschritte zu aktivieren.

Mit der Massenaktualisierung für *Vorverarbeitung* können Sie:

- *Ersetzen* – die vorhandenen Vorverarbeitungsschritte der Datenpunkte durch die unten angegebenen Vorverarbeitungsschritte ersetzen
- *Alle entfernen* – alle Vorverarbeitungsschritte aus den Datenpunkten entfernen

Klicken Sie anschließend auf *Aktualisieren*.

Wertzuordnung

Übersicht

Mit der Wertezuordnung lässt sich eine benutzerfreundlichere Darstellung empfangener Werte konfigurieren, indem Zuordnungen zwischen numerischen/Zeichenfolgenwerten und Zeichenfolgendarstellungen verwendet werden.

Wenn der Wert eines Datenpunkts beispielsweise „0“ oder „1“ ist, können Wertezuordnungen verwendet werden, um diese Werte benutzerfreundlicher darzustellen:

- 0 → Nicht verfügbar
- 1 → Verfügbar

Wertezuordnungen für Datensicherungstypen könnten beispielsweise wie folgt konfiguriert werden:

- F → Vollständig
- D → Differenziell
- I → Inkrementell

Wertebereiche für Spannung könnten beispielsweise wie folgt konfiguriert werden:

- <=209 → Niedrig
- 210-230 → OK
- >=231 → Hoch

Die Wertezuordnung wird im Zabbix Frontend und in Benachrichtigungen verwendet, die von Medientypen gesendet werden.

Attention:
Die Ersetzung des empfangenen Werts durch die konfigurierte Darstellung erfolgt sowohl im Zabbix Frontend als auch auf dem Server; der Server verarbeitet die Ersetzung jedoch nur in den folgenden Fällen:

- beim Befüllen des Host-Inventars;
- beim Erweitern unterstützter Makros {ITEM.VALUE}, {ITEM.LASTVALUE}, {EVENT.OPDATA} und {EVENT.CAUSE.OPDATA}.

Wertezuordnungen werden auf Vorlagen oder Hosts eingerichtet. Nach der Konfiguration stehen sie für alle Datenpunkte innerhalb der jeweiligen Vorlage oder des jeweiligen Hosts zur Verfügung. Beim Konfigurieren von Datenpunkten geben Sie im Parameter *Wertezuordnung* den Namen einer zuvor konfigurierten Wertezuordnung an.

Note:
Es gibt keine Vererbung von Wertezuordnungen – Hosts und Vorlagen übernehmen keine Wertezuordnungen aus verknüpften Vorlagen. Vorlagen-Datenpunkte auf einem Host verwenden weiterhin die auf der Vorlage konfigurierten Wertezuordnungen.

Note:
Wertezuordnungen können mit Datenpunkten verwendet werden, die die Informationstypen *Numerisch (Ganzzahl ohne Vorzeichen)*, *Numerisch (Gleitkommazahl)* und *Zeichen* haben.

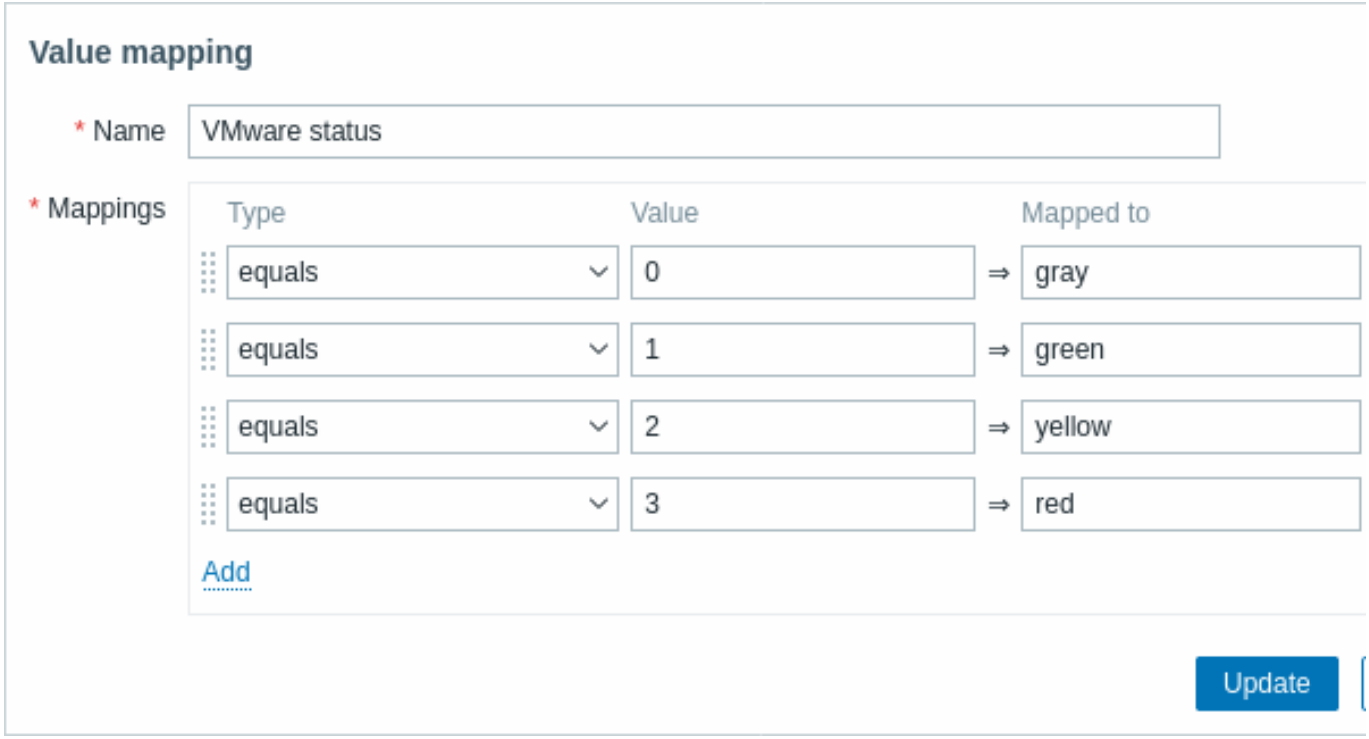
Wertezuordnungen werden zusammen mit der jeweiligen Vorlage oder dem jeweiligen Host exportiert/importiert. Sie können auch mithilfe der Formulare zur Massenaktualisierung für Hosts und Vorlagen massenhaft aktualisiert werden.

Konfiguration

Gehen Sie wie folgt vor, um eine Wertzuordnung zu konfigurieren:

1. Öffnen Sie das Konfigurationsformular des Hosts oder der Vorlage.

2. Klicken Sie auf der Registerkarte *Wertzuordnung* auf *Hinzufügen*, um eine neue Wertzuordnung hinzuzufügen, oder klicken Sie auf den Namen einer vorhandenen Zuordnung, um sie zu bearbeiten.



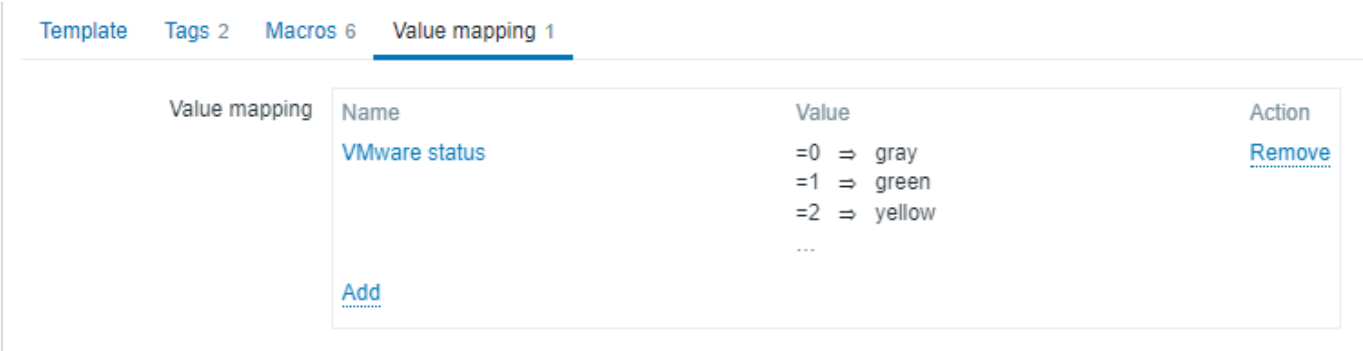
Parameter einer Wertzuordnung:

Parameter	Beschreibung
Name	Eindeutiger Name für den Satz von Wertzuordnungen.
Zuordnungen	Einzelne Regeln für die Zuordnung numerischer Werte oder Zeichenfolgen zu Zeichenfolgendarstellungen. Die Zuordnung wird in der Reihenfolge der Regeln angewendet. Diese Reihenfolge kann durch Ziehen geändert werden.

Parameter	Beschreibung
Typ	Zuordnungstyp: **equals** – Gleiche Werte werden zugeordnet; **is greater than or equals** – Gleiche oder größere Werte werden zugeordnet; **is less than or equals** – Gleiche oder kleinere Werte werden zugeordnet; **in range** – Werte innerhalb eines Bereichs werden zugeordnet; der Bereich wird als <number1>-<number2> oder <number> angegeben; mehrere Bereiche werden unterstützt (zum Beispiel 1-10,101-110,201); **regexp** – Werte, die dem regulären Ausdruck entsprechen, werden zugeordnet (globale reguläre Ausdrücke werden nicht unterstützt); **default** – Alle übrigen Werte werden zugeordnet, sofern für sie keine spezifischen Zuordnungen vorhanden sind. Für die Zuordnung von Bereichen werden nur numerische Werttypen (*is greater than or equals*, *is less than or equals*, *in range*) unterstützt.
Wert	Eingehender Wert (kann abhängig vom Zuordnungstyp einen Bereich oder einen regulären Ausdruck enthalten).
Zugeordnet zu	Zeichenfolgendarstellung (bis zu 64 Zeichen) für den eingehenden Wert.

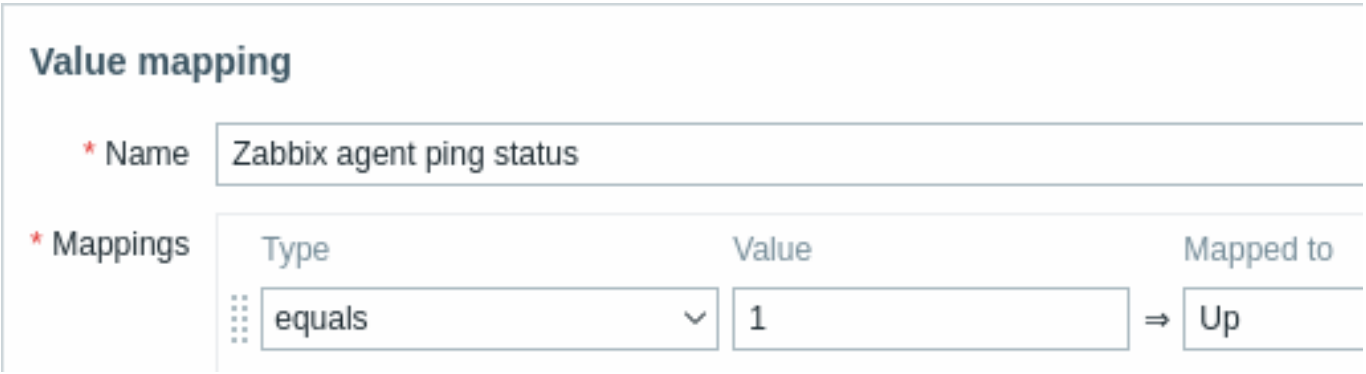
Alle erforderlichen Eingabefelder sind mit einem roten Sternchen gekennzeichnet.

Bei der Anzeige der Wertzuordnung in der Liste sind nur die ersten drei Zuordnungen sichtbar. Drei Punkte zeigen an, dass weitere Zuordnungen vorhanden sind.



Beispiel für eine Wertzuordnung

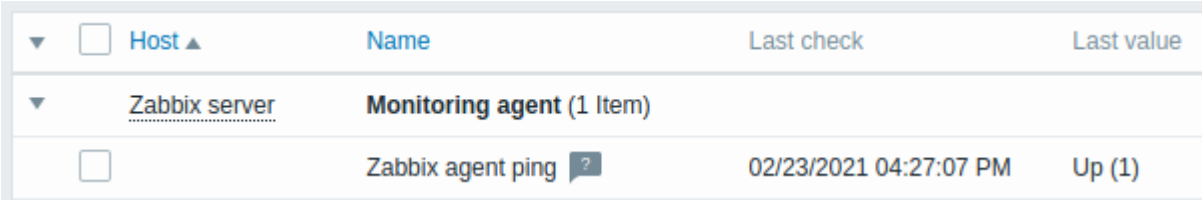
Einer der vordefinierten Agent-Datenpunkte *Zabbix agent ping* verwendet die Wertzuordnung auf Vorlagenebene „Zabbix agent ping status“, um seine Werte anzuzeigen.



Im Datenpunkt-Konfigurationsformular finden Sie im Feld *Value mapping* einen Verweis auf diese Wertzuordnung:

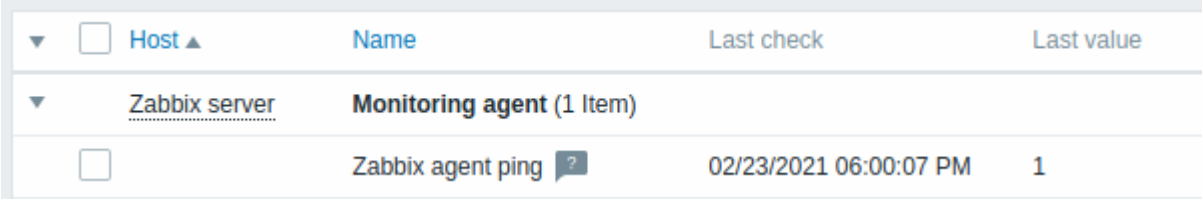


Diese Zuordnung wird im Abschnitt *Monitoring* > *Latest data* verwendet, um „Up“ anzuzeigen (mit dem Rohwert in Klammern).



Note:
Im Abschnitt *Latest data* werden angezeigte Werte auf 20 Zeichen gekürzt. Wenn eine Wertzuordnung verwendet wird, wird diese Kürzung nicht auf den zugeordneten Wert, sondern nur auf den Rohwert angewendet, der in Klammern angezeigt wird.

Ohne eine vordefinierte Wertzuordnung würde nur „1“ angezeigt, was möglicherweise schwer verständlich wäre.



Warteschlange

Übersicht

Die Warteschlange zeigt Datenpunkte an, die auf eine Aktualisierung warten. Die Warteschlange ist lediglich eine **logische** Darstellung von Daten. Es gibt in Zabbix keine IPC-Warteschlange und auch keinen anderen Warteschlangenmechanismus.

Von Proxys überwachte Datenpunkte sind ebenfalls in der Warteschlange enthalten – sie werden für den Aktualisierungszeitraum der Proxy-Verlaufsdaten als in der Warteschlange gezählt.

In der Warteschlange werden nur Datenpunkte mit geplanten Aktualisierungszeiten angezeigt. Das bedeutet, dass die folgenden Datenpunkttypen von der Warteschlange ausgeschlossen sind:

- aktive Zabbix-Agent-Datenpunkte vom Typ log, logrt und event log
- SNMP-Trap-Datenpunkte
- Trapper-Datenpunkte
- Datenpunkte der Web-Überwachung
- abhängige Datenpunkte

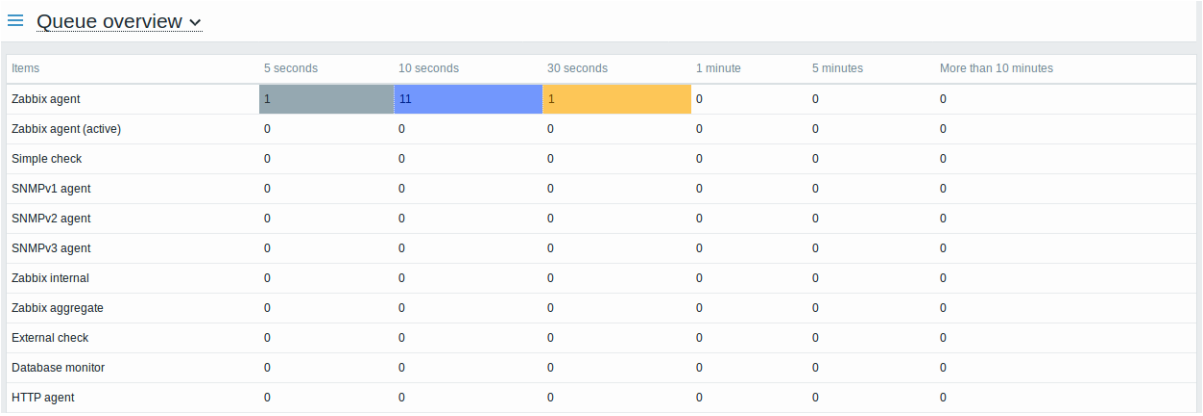
Die von der Warteschlange angezeigten Statistiken sind ein guter Indikator für die Leistung des Zabbix-Servers.

Die Warteschlange wird direkt vom Zabbix-Server über das JSON-Protokoll abgerufen. Die Informationen sind nur verfügbar, wenn der Zabbix-Server läuft.

Attention:
Datenpunkte werden in der Warteschlange nicht gezählt, wenn die Schnittstelle des Datenpunkts aufgrund von Verbindungsproblemen nicht verfügbar wird oder der Agent nicht ordnungsgemäß funktioniert.

Lesen der Warteschlange

Um die Warteschlange zu lesen, gehen Sie zu *Administration* > *Warteschlange* > *Warteschlangenübersicht*.



Die Abbildung hier ist im Allgemeinen „in Ordnung“, daher können wir davon ausgehen, dass der Server ordnungsgemäß funktioniert.

Die Warteschlange zeigt einige Datenpunkte, die bis zu 30 Sekunden warten. Es wäre hilfreich zu wissen, um welche Datenpunkte es sich handelt.

Wählen Sie dazu im Dropdown-Menü des Titels *Warteschlangendetails* aus. Nun wird eine Liste dieser verzögerten Datenpunkte angezeigt.

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

Anhand dieser Details lässt sich möglicherweise herausfinden, warum diese Datenpunkte verzögert sein könnten.

Bei einem oder zwei verzögerten Datenpunkten besteht möglicherweise kein Grund zur Sorge. Sie werden vielleicht innerhalb einer Sekunde aktualisiert. Wenn jedoch viele Datenpunkte über einen längeren Zeitraum verzögert werden, könnte ein schwerwiegenderes Problem vorliegen.

Siehe auch: Zeitzonen beim Verwenden von Planungsintervallen abgleichen.

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

Warteschlangen-Datenpunkt

Ein spezieller interner Datenpunkt `zabbix[queue,<from>,<to>]` kann verwendet werden, um den Zustand der Warteschlange in Zabbix zu überwachen. Er gibt die Anzahl der Datenpunkte zurück, die um die festgelegte Zeit verzögert sind. Weitere Informationen finden Sie unter Interne Datenpunkte.

Wert-Cache

Übersicht

Um die Berechnung von Auslöserausdrücken, berechneten Datenpunkten und einigen Makros deutlich zu beschleunigen, wird vom Zabbix Server eine Wert-Cache-Option unterstützt.

Dieser In-Memory-Cache kann für den Zugriff auf Verlaufsdaten verwendet werden, anstatt direkte SQL-Aufrufe an die Datenbank auszuführen. Wenn Verlaufswerte nicht im Cache vorhanden sind, werden die fehlenden Werte aus der Datenbank angefordert und der Cache entsprechend aktualisiert.

Datenpunkt-Werte verbleiben im Wertespeicher, bis:

- der Datenpunkt gelöscht wird (zwischengespeicherte Werte werden nach der nächsten Konfigurationssynchronisierung gelöscht);

- der Datenpunkt-Wert außerhalb des im Auslöser-/berechneten-Datenpunkt-Ausdruck angegebenen Zeit- oder Zählbereichs liegt (der zwischengespeicherte Wert wird entfernt, wenn ein neuer Wert empfangen wird);
- der im Auslöser-/berechneten-Datenpunkt-Ausdruck angegebene Zeit- oder Zählbereich geändert wird, sodass weniger Daten für die Berechnung erforderlich sind (nicht benötigte zwischengespeicherte Werte werden nach 24 Stunden entfernt).

Note:
Der Status des Wertespeichers kann über die Server-Laufzeitsteuerungs-Option `diaginfo` (oder `diaginfo=valuecache`) beobachtet werden, indem der Abschnitt mit den Diagnoseinformationen zum Wertespeicher geprüft wird. Dies kann hilfreich sein, um falsch konfigurierte Auslöser oder berechnete Datenpunkte zu ermitteln.

Um die Funktionalität des Wert-Cache zu aktivieren, kann in der Server-Konfiguration ein optionaler Parameter `ValueCacheSize` angegeben werden.

Für die Überwachung des Wert-Cache werden zwei interne Datenpunkte unterstützt: `zabbix[vcache,buffer,<mode>]` und `zabbix[vcache,cache,<parameter>]`. Weitere Informationen finden Sie unter internen Datenpunkten.

Jetzt ausführen

Übersicht

Mit der Option *Jetzt ausführen* können passive Prüfungen sofort ausgeführt werden.

Die Erfassung von Datenpunkt-Werten in Zabbix erfolgt gemäß den konfigurierten Aktualisierungsintervallen. Einige Datenpunkte (z. B. Low-Level-Discovery-Regeln) haben lange Intervalle, und es kann erforderlich sein, sofort einen neuen Wert abzurufen – beispielsweise, um Änderungen an erkennbaren Ressourcen festzustellen.

Die Option *Jetzt ausführen* wird für die folgenden Datenpunkt-Typen unterstützt:

- Zabbix-Agent (passiv)
- Einfache Prüfung
- SNMP-Agent (v1/v2/v3)
- Zabbix intern
- Externe Prüfung
- Datenbankmonitor
- HTTP-Agent
- IPMI-Agent
- SSH-Agent
- TELNET-Agent
- JMX-Agent
- Berechnet
- Abhängiger Datenpunkt
- Skript
- Browser

Wenn der Datenpunkt vom Typ „Abhängiger Datenpunkt“ ist, muss auch sein Master-Datenpunkt einer der oben genannten Typen sein.

Konfiguration

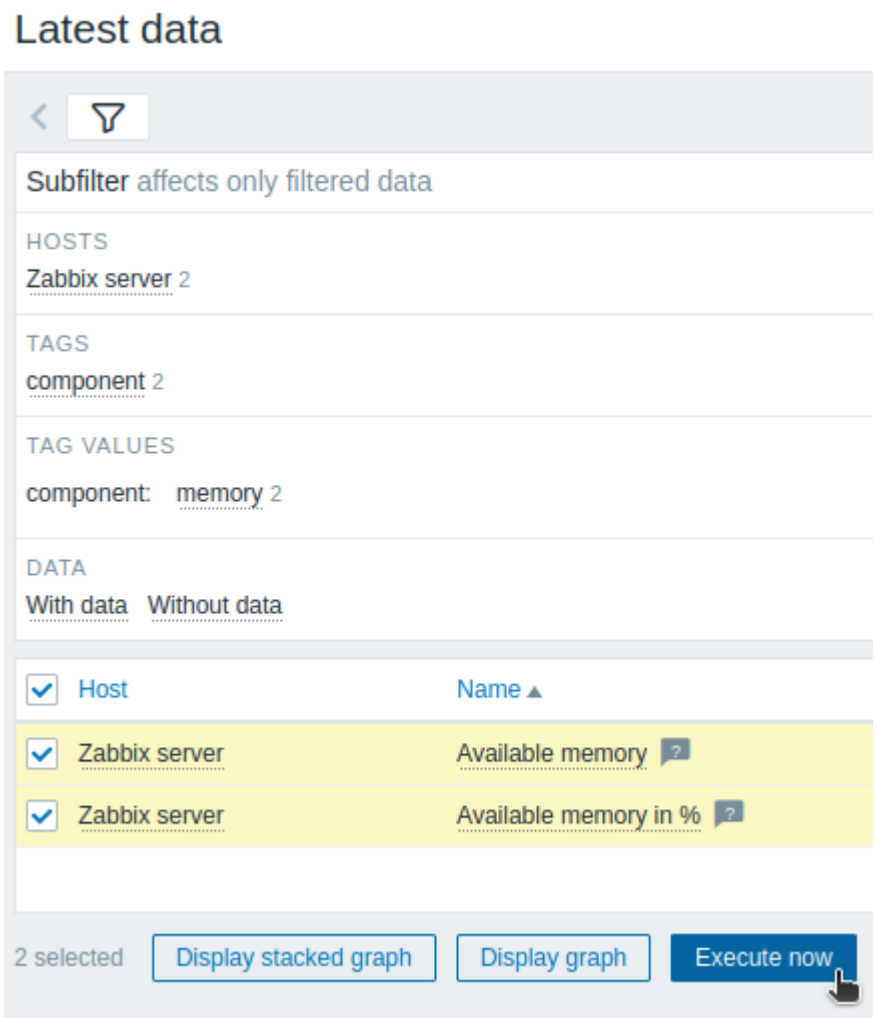
Sie können passive Checks von folgenden Stellen aus ausführen:

- *Monitoring* > *Neueste Daten*
- *Datenerfassung* > *Hosts* > *Datenpunkte* oder *Erkennung*
- Das item-Menü
- Beim Bearbeiten von Datenpunkten oder Low-Level-Erkennungsregeln (in ihren Konfigurationsformularen)

Attention:
Der Check muss im Konfigurations-Cache vorhanden sein, damit er ausgeführt werden kann; siehe auch CacheUpdate-Frequency. Der Cache wird vor der Check-Ausführung **nicht aktualisiert**, daher werden aktuelle Änderungen an der Konfiguration von Datenpunkten oder Low-Level-Erkennungsregeln nicht übernommen. Um Datenpunkte oder LLD-Regeln zu testen, die gerade erstellt werden oder neu erstellt wurden, können Sie stattdessen die Option Test verwenden.

Um beispielsweise einen passiven Check in *Monitoring* > *Neueste Daten* auszuführen:

1. Wählen Sie Datenpunkte aus; Sie können den Check für mehrere Datenpunkte gleichzeitig ausführen.
2. Klicken Sie auf *Jetzt ausführen*.



Note:
In *Neueste Daten* können Benutzer *Jetzt ausführen* für Datenpunkte auf Hosts verwenden, für die sie über die Berechtigung *lesen-schreiben* Berechtigung verfügen, oder über *Lesen* und die für ihre Rolle aktivierte Aktion *"Jetzt ausführen" auf schreibgeschützten Hosts ausführen*.

Einschränkung von Agent-Prüfungen

Übersicht

Sie können steuern, welche Datenpunkt-Schlüssel der Zabbix Agent oder Agent 2 beim Ausführen von Datenpunkt-Prüfungen, Remote-Befehlen oder Skripten verwenden darf oder nicht verwenden darf.

Verwenden Sie dazu diese Parameter der Agent-Konfiguration, um Allow-/Deny-Regeln zu definieren:

- `AllowKey=<pattern>`
- `DenyKey=<pattern>`
- `AllowKeyRegexp=<pattern>`
- `DenyKeyRegexp=<pattern>`

Das `<pattern>` muss einen einzelnen Datenpunkt-Schlüssel enthalten.
 Für `AllowKey` und `DenyKey` unterstützt das `<pattern>` Platzhalter (`*`). Der Platzhalter stimmt an seiner Position mit beliebig vielen beliebigen Zeichen überein und kann verwendet werden, um Datenpunkt-Schlüssel oder Parameter abzugleichen (z. B. `vfs.file.*[*]`).
 Für `AllowKeyRegexp` und `DenyKeyRegexp` unterstützt das `<pattern>` reguläre Ausdrücke. Das Muster des regulären Ausdrucks kann verwendet werden, um Datenpunkt-Schlüssel und deren Parameterzeichenfolgen abzugleichen. Siehe auch Beispiele für korrektes Escaping.
 Bitte beachten Sie, dass Zabbix Agent und Zabbix agent 2 unterschiedliche Syntax für reguläre Ausdrücke unterstützen können.

Attention:
Zur Verbesserung der Sicherheit wird empfohlen, für `AllowKey` und `DenyKey` exakte Datenpunkt-Schlüssel anstelle von Platzhaltern zu verwenden. Außerdem wird empfohlen, für `AllowKeyRegexp` und `DenyKeyRegexp` möglichst spezifische Muster für reguläre Ausdrücke zu verwenden. Weitere Informationen finden Sie unter Absicherung von Allow-/Deny-Regeln.

Im Gegensatz zu anderen Agent-Konfigurationsparametern können Sie eine unbegrenzte Anzahl von AllowKey, DenyKey, AllowKeyRegexp oder DenyKeyRegexp-Parametern angeben.

Wichtige Hinweise

- Alle system.run-Datenpunkte sind standardmäßig deaktiviert (auch wenn DenyKey und DenyKeyRegexp leer sind), so als wäre DenyKey=system.run[*] oder DenyKeyRegexp=^system\.run\[.*\]$ als letzte Regel gesetzt. Dadurch können Sie bestimmte system.run-Datenpunkte zulassen, ohne andere system.run-Datenpunkte ausdrücklich zu verweigern.
- Verwenden Sie nach Möglichkeit AllowKey/AllowKeyRegexp, um nur die erforderlichen Datenpunkte zuzulassen und alles andere zu verweigern. Einige Schlüssel können missbraucht werden, um über Path Traversal unbeabsichtigte Dateien zu lesen (z. B. vfs.file.contents["../../../../etc/passwd"]), und neue Zabbix-Agent-Versionen können Schlüssel einführen, die nicht durch Ihre DenyKey/DenyKeyRegexp-Regeln abgedeckt sind.
- Die Konfiguration von AllowKey, DenyKey, AllowKeyRegexp und DenyKeyRegexp wirkt sich nicht auf die Agent-Parameter HostnameItem, HostMetadataItem oder HostInterfaceItem aus.
- Verweigerte Datenpunkte werden ohne Hinweise oder Fehlermeldungen als nicht unterstützt eingestuft; zum Beispiel:
  - Der Befehlszeilenparameter --print (-p) des Zabbix-Agenten zeigt verweigerte Datenpunktschlüssel nicht an.
  - Der Befehlszeilenparameter --test (-t) des Zabbix-Agenten gibt für verweigerte Datenpunktschlüssel "Unsupported item key." zurück.
  - Wenn die Protokollierung aktiviert ist (LogRemoteCommands=1), protokolliert die Logdatei des Zabbix-Agenten verweigerte Remote-Befehle nicht.

Reihenfolge der Allow-/Deny-Regeln

Sie können eine unbegrenzte Anzahl von AllowKey, DenyKey, AllowKeyRegexp oder DenyKeyRegexp-Regeln angeben, wobei ihre Reihenfolge wichtig ist.

- Die Regeln werden nacheinander von oben nach unten ausgewertet.
- Wenn ein Datenpunkt-Schlüssel zu einer Regel passt, wird er entweder erlaubt oder verweigert, und die Auswertung der Regeln wird beendet.

Wenn beispielsweise vfs.file.contents[/etc/passwd] ausgewertet wird, werden die Regeln wie folgt verarbeitet:

```
AllowKey=vfs.file.contents[/tmp/app.log]    # Das Datenpunkt-Schlüssel-Muster passt nicht; der Agent fährt
AllowKey=vfs.file.contents[/etc/passwd]     # Das Datenpunkt-Schlüssel-Muster passt; der Agent erlaubt die
DenyKey=vfs.file.*[*]                       # Der Agent ignoriert die Regel, da die Auswertung bereits bee
```

Die folgende Reihenfolge der Regeln verweigert die Datenpunktprüfung:

```
DenyKey=vfs.file.*[*]                       # Das Datenpunkt-Schlüssel-Muster passt; der Agent verweigert
AllowKey=vfs.file.contents[/etc/passwd]     # Der Agent ignoriert die Regel, da die Auswertung bereits bee
AllowKey=vfs.file.contents[/tmp/app.log]    # Der Agent ignoriert die Regel, da die Auswertung bereits bee
```

Beispiele

Die folgenden Beispiele zeigen gängige Konfigurationsmuster für AllowKey, DenyKey, AllowKeyRegexp und DenyKeyRegexp.

Zulassen bestimmter Prüfungen und Befehle

Erlauben Sie nur zwei vfs.file Datenpunkt-Prüfungen und zwei system.run Befehle:

mit AllowKey und DenyKey:

```
AllowKey=vfs.file.contents[/tmp/app.log]
AllowKey=vfs.file.size[/tmp/app.log]
AllowKey=system.run[/usr/bin/uptime]
AllowKey=system.run[/usr/bin/df -h /]
DenyKey=vfs.file.*[*]
```

mit AllowKeyRegexp und DenyKeyRegexp:

```
AllowKeyRegexp=^vfs\.file\.(contents|size)\[/tmp/app\.log\]$
AllowKeyRegexp=^system\.run\[/usr/bin/(uptime|df -h /)\]$
DenyKeyRegexp=^vfs\.file\..*\[.*\]$
```

Note:
Das Setzen von DenyKey=system.run[*] oder DenyKeyRegexp=^system\.run\[.*\]$ ist nicht erforderlich, da alle anderen system.run Befehle standardmäßig verweigert werden.

Zulassen von Skripten

Erlauben Sie dem Zabbix Agent, Skripte auf Hosts über alle verfügbaren Methoden auszuführen:

- Globale Skripte, die im Frontend oder über die API ausgeführt werden können (bei dieser Methode wird immer der Schlüssel `system.run[myscript.sh]` verwendet)
- Remote-Befehle aus Aktionen-Operationen (bei dieser Methode wird immer der Schlüssel `system.run[myscript.sh,nowait]` verwendet)
- `system.run` Zabbix Agent-Datenpunkte mit dem Skript, zum Beispiel:
  - `system.run[myscript.sh]`
  - `system.run[myscript.sh,wait]`
  - `system.run[myscript.sh,nowait]`

```
AllowKey=system.run[myscript.sh,*]
```

oder

```
AllowKeyRegexp=^system\.run\[myscript\.sh,.*\]$
```

Um den Parameter `wait/nowait` zu steuern, müssen Sie eine andere Regel festlegen. Sie können zum Beispiel nur `system.run[myscript.sh,wait]`-Datenpunkte zulassen und damit andere Methoden ausschließen:

```
AllowKey=system.run[myscript.sh,wait]
```

oder

```
AllowKeyRegexp=^system\.run\[myscript\.sh,wait\]$
```

Absicherung von Allow-/Deny-Regeln

Diese Beispiele zeigen, wie zu großzügige `AllowKey`-/`AllowKeyRegexp`- oder `DenyKey`-/`DenyKeyRegexp`-Regeln abgesichert werden können.

Betrachten Sie die folgenden Regeln:

mit `AllowKey` und `DenyKey`:

```
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat*"]
DenyKey=vfs.file.*
DenyKey=system.cpu.load[*]
```

mit `AllowKeyRegexp` und `DenyKeyRegexp`:

```
AllowKeyRegexp=^system\.run\["C:\\Program\sFiles\\Zabbix\sAgent\s2\\scripts\\test\.bat.*"\]$
DenyKeyRegexp=^vfs\.file\..*$
DenyKeyRegexp=^system\.cpu\.load\[.*\]$
```

Note:
Unter Windows müssen Sie Leerzeichen im Pfad mit einem Caret (^) für `AllowKey` oder `DenyKey` und mit `\s` für `AllowKeyRegexp` oder `DenyKeyRegexp` maskieren.

Diese Regeln enthalten einen Platzhalter (`*`) (oder `.*` bei Mustern auf Basis regulärer Ausdrücke), der missbraucht werden kann:

- Das Skript `test.bat` kann mit beliebigen Parametern ausgeführt werden, auch mit unbeabsichtigten.
- Das Muster `vfs.file.*` (`^vfs\.file\..*$`) stimmt sowohl mit Datenpunkt-Schlüsseln mit als auch ohne Parameter überein; allerdings erfordern alle `vfs.file`-Datenpunkte Parameter.
- Das Muster `system.cpu.load[*]` (`^system\.cpu\.load\[.*\]$`) stimmt nur mit Datenpunkt-Schlüsseln mit Parametern überein; allerdings benötigen `system.cpu.load`-Datenpunkte keine Parameter.

Um diese Regeln abzusichern, erlauben Sie die Ausführung von `test.bat` nur mit bestimmten Parametern und verweigern Sie korrekte Datenpunkt-Schlüssel-Muster; zum Beispiel:

mit `AllowKey` und `DenyKey`:

```
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat status"]
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat version"]
DenyKey=vfs.file.*[*]
DenyKey=system.cpu.load
DenyKey=system.cpu.load[*]
```

mit `AllowKeyRegexp` und `DenyKeyRegexp`:

```
AllowKeyRegexp=^system\.run\["C:\\Program\sFiles\\Zabbix\sAgent\s2\\scripts\\test\.bat (status|version)"\]
DenyKeyRegexp=^vfs\.file\..+\[.*\]$
DenyKeyRegexp=^system\.cpu\.load(\[.*\])?$
```

Sie können die Regeln testen, indem Sie die folgenden Befehle ausführen, die ZBX_NOTSUPPORTED zurückgeben.

```
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]
```

Musterbeispiele

Die folgenden Tabellen zeigen, wie Datenpunkt-Schlüssel-Muster abgeglichen werden:

- Ein Schlüssel stimmt nur dann mit dem Muster überein, wenn er **alle** Bedingungen in der Spalte *Matches* erfüllt.
- Parameter müssen vollständig in eckige Klammern eingeschlossen sein (z. B. sind `vfs.file.contents[*` und `vfs.file.contents*utf8]` ungültige Muster).
- Die Beispiele veranschaulichen nur den Musterabgleich; tatsächliche Datenpunkte erfordern Parameter.

Für `AllowKey` und `DenyKey`:

Pattern	Matches	Examples
`*`	Beliebiger Schlüssel mit oder ohne Parameter	

`vfs.file.*`	Schlüssel beginnt mit `vfs.file.`
	Mit oder ohne Parameter
		Matches:
		`vfs.file.size`
		`vfs.file.contents`
		`vfs.file.contents[]`
		`vfs.file.size[/var/log/app.log]`

`vfs.*.contents`	Schlüssel beginnt mit `vfs.`
	Schlüssel endet mit `.contents`
	Keine Parameter
		Matches:
		`vfs..contents`
		`vfs.mount.point.file.contents`
		
		Does not match:
		`vfs.contents`
		`vfs.file.contents[]`

`vfs.file.*[*]`	Schlüssel beginnt mit `vfs.file.`
	Beliebige oder leere Parameter
		Matches
		`vfs.file.get.custom[]`
		`vfs.file.size[/var/log/app.log, utf8]`
		
		Does not match:
		`vfs.file.get.custom`

`vfs.file.contents`	Schlüssel ist `vfs.file.contents`
	Keine Parameter
		Matches:
		`vfs.file.contents`
		
		Does not match:
		`vfs.file.contents[/etc/passwd]`

`vfs.file.contents[]`	Schlüssel ist `vfs.file.contents[]`
	Leere Parameter
		Matches:
		`vfs.file.contents[]`
		
		Does not match:
		`vfs.file.contents`

`vfs.file.contents[*]`	Schlüssel ist `vfs.file.contents`
	Beliebige oder leere Parameter
		Matches:
		`vfs.file.contents[/path/to/file]`
		
		Does not match:
		`vfs.file.contents`

Pattern	Matches	Examples
vfs.file.contents[/etc/passwd,*]	Schlüssel ist vfs.file.contents Erster Parameter ist /etc/passwd Beliebiger oder leerer zweiter Parameter	Matches: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8] Does not match: vfs.file.contents[] vfs.file.contents[/etc/passwd]
vfs.file.contents[*passwd*]	Schlüssel ist vfs.file.contents Beliebige Parameter, mindestens einer enthält passwd	Matches: vfs.file.contents[/etc/passwd] vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8]
vfs.file.contents[*passwd*,*]	Schlüssel ist vfs.file.contents Erster Parameter enthält passwd Beliebiger oder leerer zweiter Parameter	Matches: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf8] Does not match: vfs.file.contents[/etc/passwd] vfs.file.contents[/tmp/test]
vfs.file.contents[/etc/passwd,utf8]	Schlüssel ist vfs.file.contents Erster Parameter ist /etc/passwd Zweiter Parameter ist utf8	Matches: vfs.file.contents[/etc/passwd,utf8] Does not match: vfs.file.contents[/etc/passwd,] vfs.file.contents[/etc/passwd,utf16]

Für AllowKeyRegexp und DenyKeyRegexp:

Pattern	Matches	Examples
^.*$	Beliebiger Schlüssel mit oder ohne Parameter	
^vfs\.file\..*$	Schlüssel beginnt mit vfs.file. Mit oder ohne Parameter	Matches: vfs.file.size vfs.file.contents vfs.file.contents[] vfs.file.size[/var/log/app.log]
^vfs\..*\.contents$	Schlüssel beginnt mit vfs. Schlüssel endet mit .contents Keine Parameter	Matches: vfs..contents vfs.mount.point.file.contents Does not match: vfs.contents vfs.file.contents[]
^vfs\.file\..*\[.*\]$	Schlüssel beginnt mit vfs.file. Beliebige oder leere Parameter	Matches vfs.file.get.custom[] vfs.file.size[/var/log/app.log,utf8] Does not match: vfs.file.get.custom
^vfs\.file\.contents$	Schlüssel ist vfs.file.contents Keine Parameter	Matches: vfs.file.contents Does not match: vfs.file.contents[/etc/passwd]
^vfs\.file\.contents\[\]$	Schlüssel ist vfs.file.contents[] Leere Parameter	Matches: vfs.file.contents[] Does not match: vfs.file.contents

Pattern	Matches	Examples

`^vfs\.file\.contents\[.*\]$`	Schlüssel ist `vfs.file.contents`
	Beliebige oder leere Parameter
		Matches:
		`vfs.file.contents[/path/to/file]`
		
		Does not match:
		`vfs.file.contents`

`^vfs\.file\.contents\[/etc/passwd,.*\]$`	Schlüssel ist `vfs.file.contents`
	Erster Parameter ist `/etc/passwd`
	Beliebiger oder leerer zweiter Parameter
		Matches:
		`vfs.file.contents[/etc/passwd,]`
		`vfs.file.contents[/etc/passwd,utf8]`
		
		Does not match:
		`vfs.file.contents[]`
		`vfs.file.contents[/etc/passwd]`

`^vfs\.file\.contents\[.*passwd.*\]$`	Schlüssel ist `vfs.file.contents`
	Beliebige Parameter, mindestens einer enthält `passwd`
		Matches:
		`vfs.file.contents[/etc/passwd]`
		`vfs.file.contents[/etc/passwd,]`
		`vfs.file.contents[/etc/passwd,utf8]`

`^vfs\.file\.contents\[.*passwd.*,.*\]$`	Schlüssel ist `vfs.file.contents`
	Erster Parameter enthält `passwd`
	Beliebiger oder leerer zweiter Parameter
		Matches:
		`vfs.file.contents[/etc/passwd,]`
		`vfs.file.contents[/etc/passwd,utf8]`
		
		Does not match:
		`vfs.file.contents[/etc/passwd]`
		`vfs.file.contents[/tmp/test]`

`^vfs\.file\.contents\[/etc/passwd,(utf8|windows-1252)\]$`	Schlüssel ist `vfs.file.contents`
	Erster Parameter ist `/etc/passwd`
	Zweiter Parameter ist `utf8` oder `windows-1252`
		Matches:
		`vfs.file.contents[/etc/passwd,utf8]`
		`vfs.file.contents[/etc/passwd,windows-1252]`
		
		Does not match:
		`vfs.file.contents[/etc/passwd,]`
		`vfs.file.contents[/etc/passwd,utf16]`

Problemerkennung mit Auslösern

Die Problemerkennung in Zabbix basiert auf **Auslösern**.

Ein Auslöser beschreibt die Problembedingung und ermöglicht es Zabbix, zu reagieren, wenn das Problem auftritt.

Der vollständige Ablauf der Problemerkennung ist:

1. Stellen Sie sicher, dass Sie einen Datenpunkt haben, der Daten sammelt.
2. Definieren Sie den Auslöser für diesen Datenpunkt. Ohne einen Auslöser sammelt Zabbix nur Daten, reagiert aber nie darauf.
3. Beobachten Sie erkannte Probleme:
   - In *Monitoring* > *Problems*
   - Im *Problems*-Widget
4. Benachrichtigungen empfangen, Remote-Befehle ausführen (falls definiert).

Beachten Sie, dass offizielle Zabbix Vorlagen vordefinierte Auslöser enthalten.
Siehe Anleitung zum Anwenden einer Vorlage für Linux überwachen.

Was ist ein „Auslöser“?

Ein Auslöser beschreibt die *Problembedingung* mithilfe des Auslöser-Ausdrucks.

Die Problembedingung tritt in der Praxis dann ein, wenn der Auslöser „auslöst“ (in Zabbix wird ein Problem erstellt).

Die Problembedingung kann auf dem letzten Wert, dem Durchschnittswert, einer erkannten Zeichenfolge und vielen anderen Funktionen basieren. Auslöser können nicht für Datenpunkte erstellt werden, die Werte mit dem Datentyp Binär oder JSON zurückgeben.

Ein einfacher Auslöser-Ausdruck berechnet die Funktion (zum Beispiel `max()` für den Maximalwert) für die in Klammern angegebenen Daten (typischerweise Datenpunkt und Zeitfenster) und vergleicht diese Berechnung dann mit einem Schwellenwert.

Zum Beispiel:

```
max(/host/vfs.fs.size[/,free],5m)<10G
```

Dieser Auslöser löst aus, wenn der freie Festplattenspeicher, gemessen über 5 Minuten, durchgehend unter 10 GB lag.

Ein Auslöser kann die folgenden Zustände haben:

Auslöserstatus	Beschreibung	In Zabbix
OK	Auslöser-Ausdruck wurde zu „0“ (oder FALSE) ausgewertet	Kein Problem erstellt Problem wird gelöst (falls vorhanden)
Problem	Auslöser-Ausdruck wurde zu „1“ (oder TRUE) ausgewertet	Problem wird erstellt

Manchmal ist die Auswertung des Auslöser-Ausdrucks aus irgendeinem Grund nicht möglich. Siehe Unbekannter Ausdruckszustand.

Weitere Ressourcen:

- Auslöser-Beispiele
- Einen Auslöser konfigurieren
- Best Practices

Berechnungshäufigkeit

Ein Auslöser wird jedes Mal neu berechnet, wenn der Zabbix Server einen **neuen Wert** empfängt, der Teil des Ausdrucks ist. Wenn ein neuer Wert empfangen wird, wird jede im Ausdruck enthaltene Funktion neu berechnet (nicht nur diejenige, die den neuen Wert empfangen hat).

Zusätzlich wird ein Auslöser alle 30 Sekunden (durch den History-Syncer) neu berechnet, wenn der Ausdruck Folgendes enthält:

- Funktionen für Datum und Uhrzeit
- die Funktion nodata()

Auslöser, die nur Trend-Funktionen enthalten, werden einmal pro kleinstem Zeitintervall im Ausdruck ausgewertet. Während viele Auslöserfunktionen Verlaufs-Daten von Datenpunkten verwenden, nutzen Trend-Funktionen Trenddaten.

Auslöser-Beispiele

Diese Seite enthält eine Sammlung von Auslöser-Beispielen.

Die Beispiele sind nach Anwendungsfall sortiert:

- Nicht verfügbarer Host
- Nicht verfügbarer Proxy
- Nicht verfügbarer SMTP-Cluster
- Nicht erreichbarer Host
- Unerwarteter Neustart
- Änderungen in wichtigen Dateien
- Änderungen in DNS-Abfragen
- Nicht übereinstimmende Software auf verschiedenen Hosts
- Nicht synchronisierte Uhren
- Veralteter Agent
- Flatternde Schnittstelle
- Hoher eingehender Datenverkehr
- Wenig Speicherplatz auf dem Datenträger
- Wenig Speicherplatz auf dem Datenträger (dynamischer Schwellenwert)
- Hohe CPU-Last
- CPU-Lastvergleich
- Langfristiger CPU-Lastvergleich
- Zeichenfolge mit Sonderzeichen

Nicht verfügbaren Host erkennen

```
max(/host/zabbix[host,agent,available],5m)=0
```

Dieser Auslöser wird ausgelöst, wenn der Zabbix Agent auf dem Host 5 Minuten lang nicht verfügbar war.
 Funktion: `max`
 Datenpunkt: `zabbix[host,agent,available]`

Alternative:

```
nodata(/host/agent.ping,5m)=1
```

Dieser Auslöser wird ausgelöst, wenn 5 Minuten lang keine Daten vom Zabbix Agent empfangen wurden.
 Funktion: nodata
 Datenpunkt: agent.ping

Nicht verfügbaren Proxy erkennen

```
fuzzytime(/host/zabbix[proxy,{$PROXY_NAME},lastaccess],1m)=0
```

Dieser Auslöser wird ausgelöst, wenn die Daten des Zabbix Proxy der Zeit des Zabbix Server um 1 Minute hinterherhinken.
 Funktion: fuzzytime
 Datenpunkt: zabbix[proxy,{$PROXY_NAME},lastaccess]

Nicht verfügbaren SMTP-Cluster erkennen

```
last(/smtp1.example.com/net.tcp.service[smtp])=0 and last(/smtp2.example.com/net.tcp.service[smtp])=0
```

Dieser Auslöser wird ausgelöst, wenn beide SMTP-Server nicht verfügbar sind.
 Funktion: last
 Datenpunkt: net.tcp.service

Nicht erreichbaren Host erkennen

```
count(/host/icmpping,30m,,"0")>5
```

Dieser Auslöser wird ausgelöst, wenn der Host in den letzten 30 Minuten mehr als 5-mal per Ping nicht erreichbar ist.
 Funktion: count
 Datenpunkt: icmpping

Unerwarteten Neustart erkennen

```
change(/host/system.uptime)<0
```

Dieser Auslöser wird ausgelöst, wenn eine negative Änderung des System-Uptime-Werts festgestellt wird (was auf einen Neustart hinweist).
 Funktion: change
 Datenpunkt: system.uptime

Änderungen in wichtigen Dateien erkennen

```
last(/host/vfs.file.cksum[/etc/passwd],#1)<>last(/host/vfs.file.cksum[/etc/passwd],#2)
```

Dieser Auslöser wird ausgelöst, wenn /etc/passwd geändert wurde. Der Ausdruck ist wahr, wenn sich die vorherige Prüfsumme von /etc/passwd von der aktuellsten unterscheidet. Ähnliche Ausdrücke können nützlich sein, um Änderungen in wichtigen Dateien wie /etc/passwd, /etc/inetd.conf, /kernel usw. zu überwachen.
 Funktion: last
 Datenpunkt: vfs.file.cksum

Änderungen in der DNS-Abfrage erkennen

```
last(/Zabbix server/net.dns.record[192.0.2.1,{$WEBSITE_NAME},{$DNS_RESOURCE_RECORD_TYPE},2,1])<>"{$WEBSITE
```

Beachten Sie die Anführungszeichen um den zweiten Operanden.

Dieser Auslöser wird ausgelöst, wenn das Abfrageergebnis nicht dem entspricht, was es normalerweise zurückgibt:

```
example.com             MX      0 mail.example.com
```

Funktion: last
 Datenpunkt: net.dns.record[192.0.2.1,{$WEBSITE_NAME},{$DNS_RESOURCE_RECORD_TYPE},2,1], mit den wie folgt definierten Makros:

```
{$WEBSITE_NAME} = example.com
{$DNS_RESOURCE_RECORD_TYPE} = MX
```

Nicht übereinstimmende Software auf verschiedenen Hosts erkennen

```
last(/host/vfs.file.contents[/etc/os-release])<>last(/host2/vfs.file.contents[/etc/os-release])
```

Dieser Auslöser wird ausgelöst, wenn sich die Ubuntu-Version auf verschiedenen Hosts unterscheidet. Beachten Sie, dass die Operanden hier Funktionen sind, die Zeichenfolgen zurückgeben.
 Funktion: last
 Datenpunkt: vfs.file.contents

Nicht synchronisierte Uhren erkennen

```
fuzzytime(/host/system.localtime,10s)=0
```

Der Auslöser wird ausgelöst, wenn sich die lokale Zeit des Clients und die Zeit des Zabbix-Servers um mehr als 10 Sekunden unterscheiden.
 Funktion: fuzzytime
 Datenpunkt: system.localtime

Beachten Sie, dass system.localtime für den Zabbix Agent als passive Prüfung konfiguriert sein muss; beim Zabbix Agent 2 kann er als aktive Prüfung konfiguriert werden.

Veralteten Agent erkennen

```
find(/host/agent.version,,"like","beta")=1
```

Dieser Auslöser wird ausgelöst, wenn der Zabbix Agent eine Beta-Version hat. Der Zabbix Agent muss aktualisiert werden.
 Funktion: `find`
 Datenpunkt: `agent.version`

Flappende Schnittstelle erkennen

```
changecount(/host/vfs.file.contents["/sys/class/net/eth0/operstate"],1h)>5
```

Dieser Auslöser wird ausgelöst, wenn sich der Betriebszustand (up/down/unknown) von `eth0` innerhalb einer Stunde mehr als 5-mal geändert hat.
 Funktion: `changecount`
 Datenpunkt: `vfs.file.contents`

Hohen eingehenden Datenverkehr erkennen

```
min(/host/net.if.in[eth0,bytes],5m)>100K
```

Dieser Auslöser wird ausgelöst, wenn die Anzahl der auf `eth0` empfangenen Bytes in den letzten fünf Minuten immer über 100 Kilobyte lag. Wahrscheinlich lädt jemand eine große Datei herunter.
 Funktion: `min`
 Datenpunkt: `net.if.in[eth0,bytes]`

Geringen Festplattenspeicher erkennen

```
max(/host/vfs.fs.size[/,free],5m)<10G
```

Der Auslöser wird ausgelöst, wenn der freie Festplattenspeicher durchgehend (5 Minuten lang) unter 10 GB liegt.

Sie können auch einen Wiederherstellungsausdruck definieren:

```
min(/host/vfs.fs.size[/,free],10m)>40G
```

Das Problem wird erst dann **gelöst**, wenn der freie Festplattenspeicher durchgehend (10 Minuten lang) über 40 GB liegt.
 Funktion: `min`
 Datenpunkt: `vfs.fs.size`

Geringen Festplattenspeicher erkennen (dynamischer Schwellenwert)

```
last(/template/hrStorageFree[{#SNMPVALUE}])<last(/template/hrStorageSize[{#SNMPVALUE}])*0.1
```

Der Auslöser wird ausgelöst, wenn der freie Speicherplatz (in Zuordnungseinheiten) unter 10 Prozent fällt. Beachten Sie, dass der Wert eines anderen Datenpunkts verwendet wird, um einen adaptiven Auslöser-Schwellenwert zu erhalten, der auf erkannten Speicher unterschiedlicher Größe anwendbar ist. Funktion: `last`

Hohe CPU-Last erkennen

```
last(/host/system.cpu.load[all,avg1])>5
```

Der Auslöser wird ausgelöst, wenn die durchschnittliche Prozessorlast eine Minute lang über 5 liegt.

Varianten:

```
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
```

Solche Auslöser analysieren 5 Minuten an Daten und werden nur ausgelöst, wenn die CPU-Last nie unter 2 liegt. Zusätzlich werden diese Auslöser ausgelöst:

- nur nachts (00:00-06:00)
- jederzeit außer für 2 Stunden beim Wochenwechsel (Sonntag, 23:00 - Montag, 01:00)

```
(last(/host/system.cpu.load[all,avg1])>5) + (last(/host2/system.cpu.load[all,avg1])>5) + (last(/host3/syst
```

Dieser Auslöser wird ausgelöst, wenn die Prozessorlast auf mindestens zwei der drei Hosts zu hoch ist.
 Funktionen: `last`, `min`, `dayofweek`, `time`
 Datenpunkt: `system.cpu.load`

CPU-Lasten vergleichen

```
avg(/Zabbix server/system.cpu.load,1h)/avg(/Zabbix server/system.cpu.load,1h:now-1d)>2
```

Der Auslöser wird ausgelöst, wenn die durchschnittliche Last heute die durchschnittliche Last derselben Stunde gestern (unter Verwendung der Zeitverschiebung `now-1d`) um mehr als das Zweifache übersteigt.
 Funktion: `avg`
 Datenpunkt: `system.cpu.load`

Langfristige CPU-Lasten vergleichen

```
trendavg(/host/system.cpu.load,1M:now/M)>1.1*trendavg(/host/system.cpu.load,1M:now/M-1M)
```

Dieser Auslöser wird ausgelöst, wenn die CPU-Last auf dem Host im letzten Monat um mehr als 10 % gestiegen ist
 Funktion: `trendavg`
 Datenpunkt: `system.cpu.load`

Sie können auch das Feld Event name in der Auslöser-Konfiguration verwenden, um eine aussagekräftige Warnmeldung zu erstellen, zum Beispiel um etwas wie

```
"Load of Exchange server increased by 24% in July (0.69) comparing to June (0.56)"
```

zu erhalten.

Der Ereignisname muss wie folgt definiert werden:

```
Load of {HOST.HOST} server increased by {{?100*trendavg(//system.cpu.load,1M:now/M)/trendavg(//system.cpu.
```

Es ist für diese Art von Problem auch sinnvoll, in der Auslöser-Konfiguration das manuelle Schließen zu erlauben.

Zeichenfolge mit Sonderzeichen erkennen

```
last(/host/vfs.file.contents[/tmp/hello])={$HELLO_MACRO}
```

Der Auslöser wird ausgelöst, wenn der Inhalt von `/tmp/hello` gleich der in `{$HELLO_MACRO}` definierten Zeichenfolge ist:

```
{$HELLO_MACRO} = \" //hello ?\"
```

Alternativ können Sie direkt mit der Zeichenfolge vergleichen:

```
last(/Zabbix server/vfs.file.contents[/tmp/hello])="\\\" //hello ?\\\""
```

Beachten Sie, dass die Sonderzeichen (\\ und ") maskiert werden, wenn die Zeichenfolge direkt verglichen wird.

Funktion: `last`
 Datenpunkt: `vfs.file.contents`

Konfigurieren eines Auslösers

Diese Seite beschreibt, wie ein Auslöser im Zabbix Frontend konfiguriert wird.

Beachten Sie, dass offizielle Zabbix Vorlagen vordefinierte Auslöser enthalten.
Siehe Anleitung zum Anwenden der Vorlage auf Linux überwachen.
Es ist auch möglich, konfigurierte Auslöser per Massenaktualisierung zu aktualisieren.

Weitere Ressourcen:

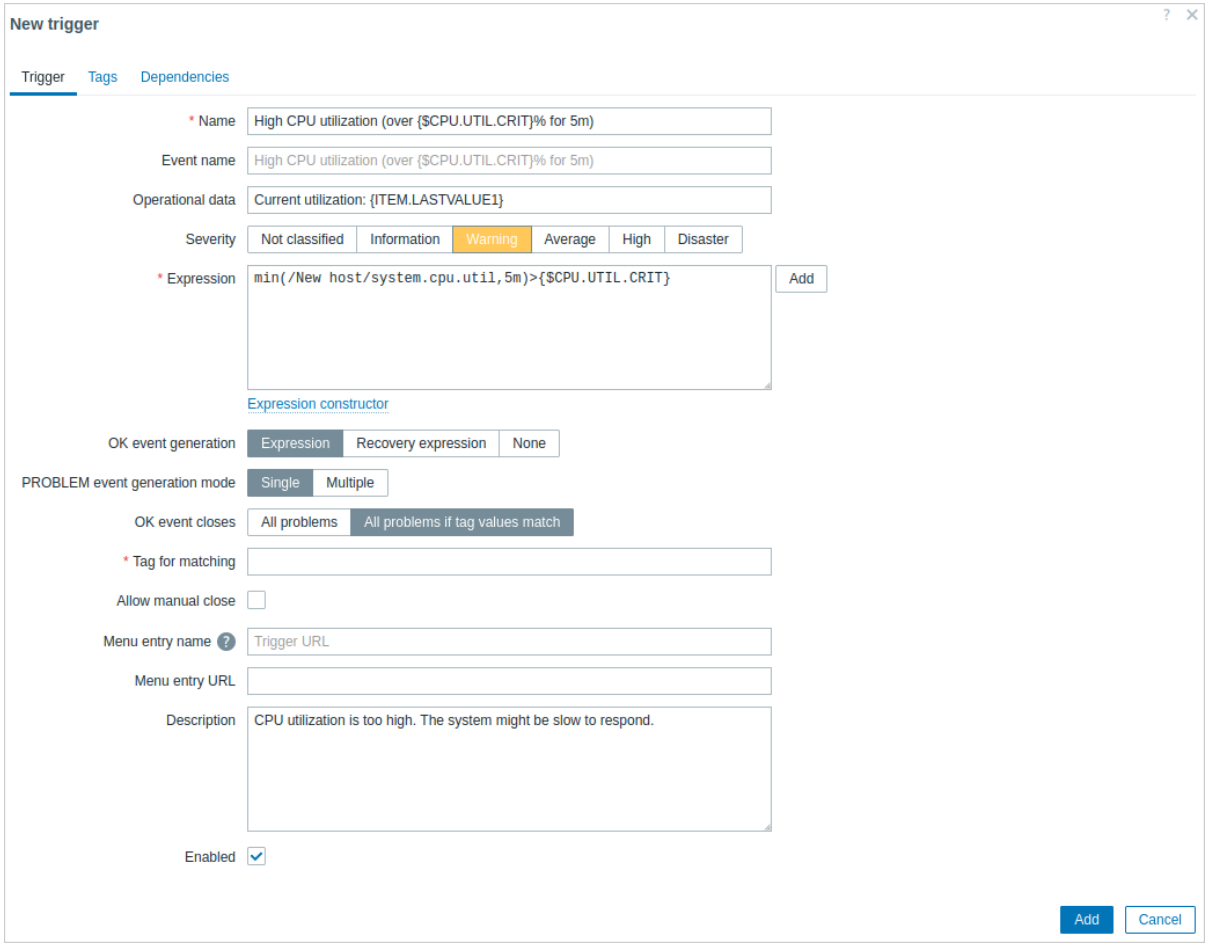
- Problemerkennung mit Auslösern (allgemeine Einführung)
- Auslöserbeispiele (nach Anwendungsfall)
- Ausdruck und Funktionen für Auslöser (Syntaxdetails)
- Bewährte Verfahren

Konfiguration

Um einen Auslöser zu konfigurieren, gehen Sie wie folgt vor:

1. Gehen Sie zu *Datenerfassung* > *Hosts*.
2. Klicken Sie in der Zeile des Hosts auf *Auslöser*.
3. Klicken Sie rechts auf *Auslöser erstellen* (oder auf den Auslösernamen, um einen vorhandenen Auslöser zu bearbeiten).
4. Geben Sie die Parameter des Auslösers im Formular ein.

Die Registerkarte **Auslöser** enthält alle wesentlichen Auslöserattribute.



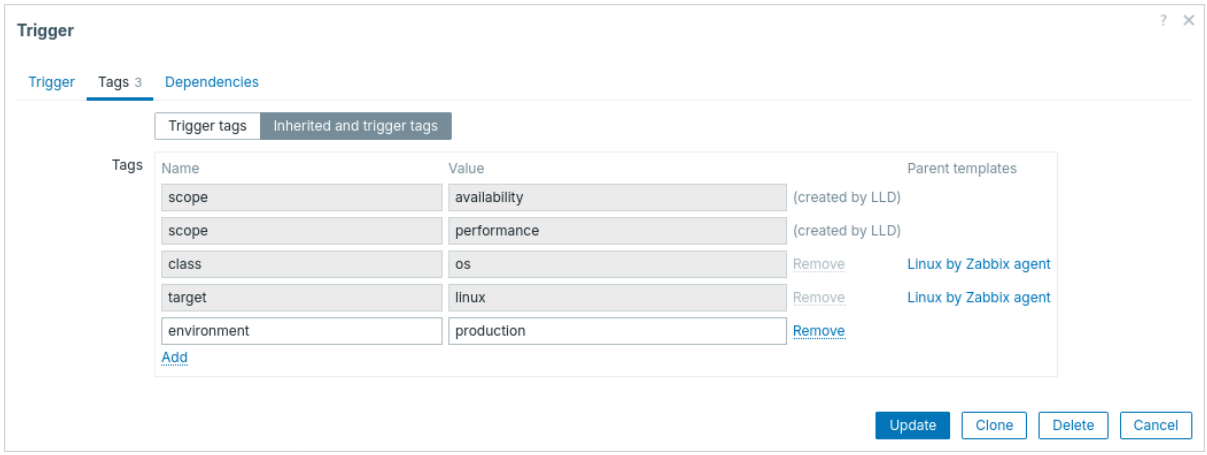
Alle Pflichtfelder sind mit einem roten Sternchen markiert.

Parameter	Beschreibung
Name	Name des Auslösers. Unterstützte Makros sind: {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.*} und {$MACRO} Benutzer-Makros. Makros **$1, $2...$9** können verwendet werden, um auf die erste, zweite ... neunte Konstante des Ausdrucks zu verweisen. *Hinweis*: Makros $1-$9 werden korrekt aufgelöst, wenn sie sich auf Konstanten in relativ einfachen, geradlinigen Ausdrücken beziehen. Der Name ”Processor load above $1 on {HOST.NAME}” wird beispielsweise automatisch in ”Processor load above 5 on New host” geändert, wenn der Ausdruck last(/New host/system.cpu.load[percpu,avg1])>5 lautet.
Event name	Wenn definiert, wird dieser Name zum Erstellen des Problemereignisnamens verwendet. Standardmäßig ist der Ereignisname identisch mit dem Namen des Auslösers. Der Ereignisname kann verwendet werden, um aussagekräftige Alarme mit Problemdaten zu erstellen (siehe Beispiel). Es wird derselbe Satz an Makros unterstützt wie beim Namen des Auslösers, zusätzlich die Ausdrucksmakros {TIME}, {TIMESTAMP} und {?EXPRESSION}.

Parameter	Beschreibung
Operational data	Geben Sie eine Zeichenfolge mit Makros ein, um dynamische Echtzeitdaten in *Monitoring* > *Problems* anzuzeigen. Alternativ können Sie dieses Feld leer lassen, um die neuesten Werte aller Datenpunkte aus dem Auslöserausdruck anzuzeigen. Es wird derselbe Satz an Makros unterstützt wie beim Namen des Auslösers, mit der Möglichkeit, diese dynamisch aufzulösen. Zum Beispiel: {ITEM.VALUE<1-9>} wird zu den Datenpunktwerten zum Zeitpunkt der Zustandsänderung des Auslösers aufgelöst (Problem erstellt, gelöst, manuell geschlossen oder durch Korrelation geschlossen). {ITEM.LASTVALUE<1-9>} wird zu den neuesten Datenpunktwerten aufgelöst. Beachten Sie, dass das manuelle Schließen eines Problems keinen neuen Wert erzeugt, sodass beide Makros weiterhin den Wert zum Problemzeitpunkt anzeigen. Beachten Sie außerdem, dass beide Makros zu *UNKNOWN* aufgelöst werden, wenn der neueste Wert älter ist als der *Max history display period* (siehe *Administration > General*).
Severity	Legen Sie die erforderliche Schweregrad des Auslösers fest, indem Sie auf die Schaltflächen klicken.
Expression	Logischer Ausdruck, der zur Definition der Bedingungen eines Problems verwendet wird. Zeit-Suffixe und Suffixe für Speichergrößen werden unterstützt. Ein Problem wird erstellt, nachdem alle im Ausdruck enthaltenen Bedingungen erfüllt sind, d. h. der Ausdruck zu TRUE ausgewertet wird. Das Problem wird gelöst, sobald der Ausdruck zu FALSE ausgewertet wird, sofern in *Recovery expression* keine zusätzlichen Wiederherstellungsbedingungen angegeben sind.
OK event generation	Optionen zur Erzeugung von OK-Ereignissen: **Expression** - OK-Ereignisse werden auf Grundlage desselben Ausdrucks wie Problemereignisse erzeugt; **Recovery expression** - OK-Ereignisse werden erzeugt, wenn der Problem-Ausdruck zu FALSE und der Wiederherstellungsausdruck zu TRUE ausgewertet wird; **None** - in diesem Fall kehrt der Auslöser niemals von selbst in den OK-Zustand zurück.
Recovery expression	Logischer Ausdruck (optional), der zusätzliche Bedingungen definiert, die erfüllt sein müssen, bevor das Problem gelöst wird, nachdem der ursprüngliche Problem-Ausdruck bereits als FALSE ausgewertet wurde. Der Wiederherstellungsausdruck ist für die Auslöser-Hysterese nützlich. Es ist **nicht** möglich, ein Problem allein durch den Wiederherstellungsausdruck zu lösen, wenn der Problem-Ausdruck weiterhin TRUE ist. Dieses Feld ist nur verfügbar, wenn für *OK event generation* 'Recovery expression' ausgewählt ist.
PROBLEM event generation mode	Modus zur Erzeugung von Problemereignissen: **Single** - ein einzelnes Ereignis wird erzeugt, wenn ein Auslöser zum ersten Mal in den Zustand 'Problem' wechselt; **Multiple** - bei *jeder* 'Problem'-Auswertung des Auslösers wird ein Ereignis erzeugt.
OK event closes	Wählen Sie aus, ob das OK-Ereignis schließt: **All problems** - alle Probleme dieses Auslösers; **All problems if tag values match** - nur diejenigen Auslöserprobleme mit übereinstimmenden Ereignis-Tag-Werten.
Tag for matching	Geben Sie den Namen des Ereignis-Tags ein, der für die Ereigniskorrelation verwendet werden soll. Dieses Feld wird angezeigt, wenn für die Eigenschaft *OK event closes* 'All problems if tag values match' ausgewählt ist, und ist in diesem Fall erforderlich.
Allow manual close	Aktivieren Sie diese Option, um das manuelle Schließen von Problemereignissen zuzulassen, die von diesem Auslöser erzeugt werden. Das manuelle Schließen ist beim Bestätigen von Problemereignissen möglich.
Menu entry name	Wenn nicht leer, wird der hier eingegebene Name (bis zu 64 Zeichen) an mehreren Frontend-Positionen als Bezeichnung für die im Parameter *Menu entry URL* angegebene Auslöser-URL verwendet. Wenn leer, wird standardmäßig der Name *Trigger URL* verwendet. Es wird derselbe Satz an Makros unterstützt wie beim Namen des Auslösers, zusätzlich {EVENT.ID}, {HOST.ID} und {TRIGGER.ID}.
Menu entry URL	Wenn nicht leer, ist die hier eingegebene URL (bis zu 2048 Zeichen) an mehreren Frontend-Positionen als Link im Ereignismenü verfügbar, zum Beispiel beim Klicken auf den Problemnamen in *Monitoring* > *Problems* oder im Dashboard-Widget *Problems*. Es wird derselbe Satz an Makros unterstützt wie beim Namen des Auslösers, zusätzlich {EVENT.ID}, {HOST.ID} und {TRIGGER.ID}. Hinweis: Benutzer-Makros mit geheimen Werten werden in der URL nicht aufgelöst.

Parameter	Beschreibung
Description	Textfeld zur Bereitstellung weiterer Informationen zu diesem Auslöser. Kann Anweisungen zur Behebung eines bestimmten Problems, Kontaktdaten der verantwortlichen Mitarbeiter usw. enthalten. Es wird derselbe Satz an Makros unterstützt wie beim Namen des Auslösers.
Enabled	Wenn Sie dieses Kontrollkästchen deaktivieren, wird der Auslöser bei Bedarf deaktiviert. Probleme eines deaktivierten Auslösers werden im Frontend nicht mehr angezeigt, aber nicht gelöscht.

Die Registerkarte **Tags** ermöglicht es Ihnen, Tags auf Auslöserebene zu definieren. Alle Probleme dieses Auslösers werden mit den hier eingegebenen Werten versehen.



Zusätzlich ermöglicht die Option *Inherited and trigger tags* das Anzeigen von auf Vorlagenebene definierten Tags, wenn der Auslöser von dieser Vorlage stammt. Wenn mehrere Vorlagen denselben Tag haben, werden diese Tags einmal angezeigt und die Vorlagennamen durch Kommas getrennt. Ein Auslöser "erbt" keine Host-Ebene-Tags und zeigt diese auch nicht an.

Parameter	Beschreibung
Name/Value	Legen Sie benutzerdefinierte Tags fest, um Auslöserereignisse zu markieren. Tags sind ein Paar aus Tag-Namen und Wert. Sie können nur den Namen verwenden oder ihn mit einem Wert kombinieren. Ein Auslöser kann mehrere Tags mit demselben Namen, aber unterschiedlichen Werten haben. Benutzer-Makros, Benutzer-Makros mit Kontext, Low-Level-Discovery-Makros und Makro-Funktionen mit `{{ITEM.VALUE}}`, `{{ITEM.LASTVALUE}}` werden in Ereignis-Tags unterstützt. Low-Level-Discovery-Makros können innerhalb des Makro-Kontexts verwendet werden. Das Makro {TRIGGER.ID} wird in Auslöser-Tag-Werten unterstützt. Es kann nützlich sein, um Auslöser zu identifizieren, die aus Auslöserprototypen erstellt wurden, und beispielsweise Probleme dieser Auslöser während Wartungsarbeiten zu unterdrücken. Wenn die Gesamtlänge des erweiterten Werts 255 überschreitet, wird er auf 255 Zeichen gekürzt. Siehe alle für Ereignis-Tags unterstützten Makros. Ereignis-Tags können für die Ereigniskorrelation, in Aktionsbedingungen verwendet werden und sind außerdem in *Monitoring* > *Problems* oder im *Problems*-Widget sichtbar.

Die Registerkarte **Abhängigkeiten** enthält alle Abhängigkeiten des Auslösers.

Klicken Sie auf *Add*, um eine neue Abhängigkeit hinzuzufügen.

Note:
Sie können einen Auslöser auch konfigurieren, indem Sie einen vorhandenen öffnen, auf die Schaltfläche *Clone* klicken und dann unter einem anderen Namen speichern.

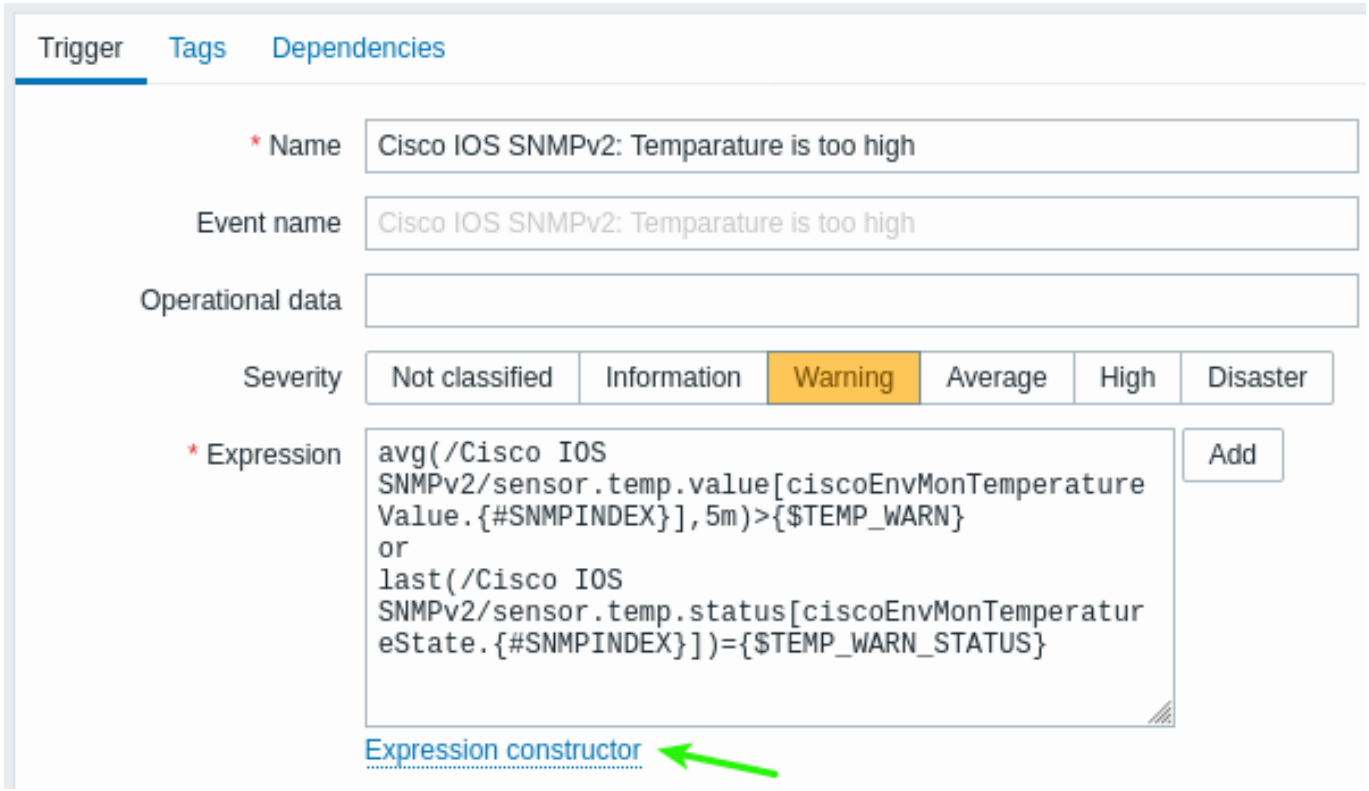
Auslöserausdruck testen

Es ist möglich zu testen, wie das Ergebnis des Ausdrucks in Abhängigkeit vom empfangenen Wert ausfallen würde.

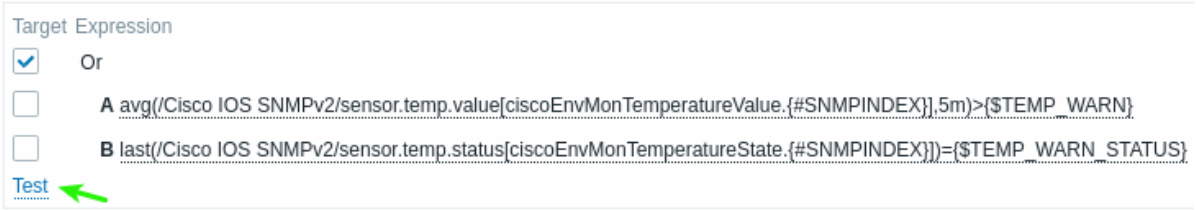
Der folgende Ausdruck aus einer offiziellen Vorlage wird als Beispiel verwendet:

```
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}
```

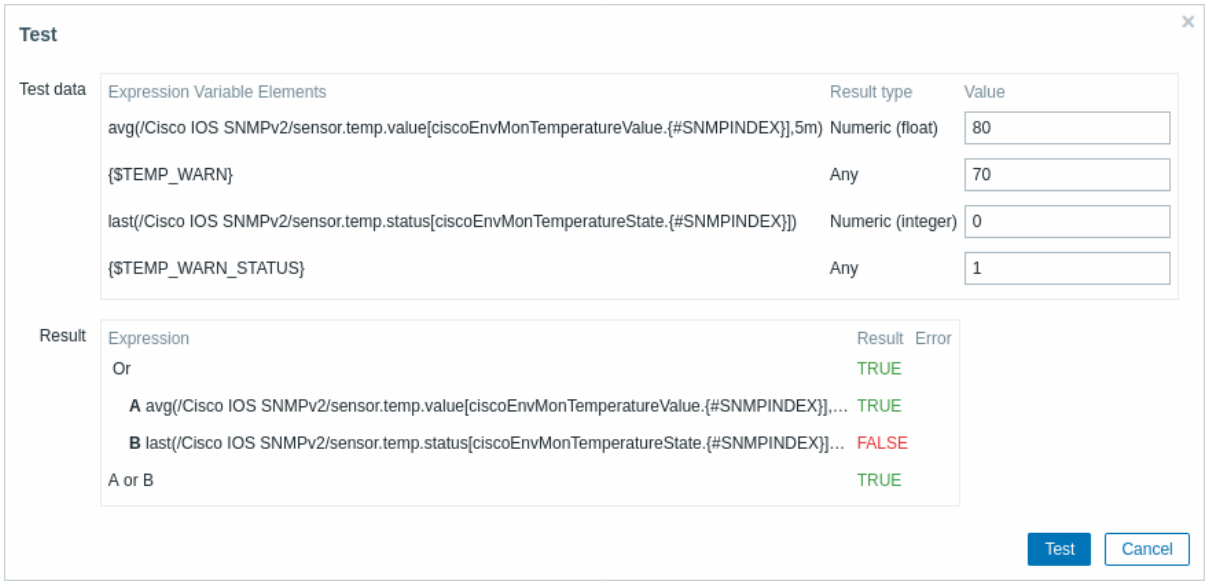
Um den Ausdruck zu testen, klicken Sie unter dem Ausdrucksfeld auf *Ausdruckskonstruktor*.



Im Ausdruckskonstruktor werden alle einzelnen Ausdrücke aufgelistet. Um das Testfenster zu öffnen, klicken Sie unter der Ausdragsliste auf *Test*.



Im Testfenster können Sie Beispielwerte eingeben (’80’, ’70’, ’0’, ’1’ in diesem Beispiel) und anschließend durch Klicken auf die Schaltfläche *Test* das Ergebnis des Ausdrucks anzeigen lassen.



Das Ergebnis der einzelnen Ausdrücke sowie des gesamten Ausdrucks kann angezeigt werden.

”TRUE” bedeutet, dass der angegebene Ausdruck korrekt ist. In diesem speziellen Fall A ist ”80” größer als der angegebene Wert `{$TEMP_WARN}`, hier ”70”. Wie erwartet wird ein Ergebnis von ”TRUE” angezeigt.

”FALSE” bedeutet, dass der angegebene Ausdruck nicht korrekt ist. In diesem speziellen Fall B muss `{$TEMP_WARN_STATUS}` ”1” mit dem angegebenen Wert übereinstimmen, hier ”0”. Wie erwartet wird ein Ergebnis von ”FALSE” angezeigt.

Der gewählte Ausdruckstyp ist ”OR”. Wenn mindestens eine der angegebenen Bedingungen (in diesem Fall A oder B) TRUE ist, ist auch das Gesamtergebnis TRUE. Das bedeutet, dass der aktuelle Wert den Warnwert überschreitet und ein Problem aufgetreten ist.

Auslöser-Ausdruck und Funktionen

Übersicht

Diese Seite beschreibt die Syntax von Auslöserausdrücken und Details zu ihrer Auswertung.

Die Syntax eines einfachen Ausdrucks lautet:

function(/host/item,time_period)**<operator><constant>**

In diesem Ausdruck ist der erste Operand (links vom Operator) eine Funktion mit ihren Parametern in Klammern (typischerweise der Datenpunkt und der Zeitraum).

Die Funktion wird verwendet, um empfangene Daten innerhalb des festgelegten Zeitraums zu analysieren, was zu einem berechneten Wert führt.

Dieser Wert wird dann mithilfe des Operators mit dem zweiten Operanden verglichen. In diesem Beispiel ist der zweite Operand eine Konstante, er kann jedoch auch eine weitere Funktion sein.

Zum Beispiel:

`min(/Zabbix server/net.if.in[eth0,bytes],5m)>100K`

Dieser Auslöser wird ausgelöst, wenn die Anzahl der auf `eth0` empfangenen Bytes während der letzten fünf Minuten immer über 100 Kilobyte lag. In diesem Fall ist der Ausdruck wahr und es wird ein Problem erstellt.

Auslöserausdrücke sind äußerst flexibel. In komplexeren Ausdrücken können mehrere Funktionen, Operatoren und Konstanten kombiniert werden.

Siehe auch:

- Auslöserbeispiele
- Problemerkennung mit Auslösern (allgemeine Einführung)

Funktionen

Funktionen ermöglichen die Analyse gesammelter Werte, beispielsweise die Berechnung des Durchschnitts oder die Suche nach einer bestimmten Zeichenfolge.

Klicken Sie auf die entsprechende Funktionsgruppe, um weitere Details anzuzeigen.

Funktionsgruppe	Funktionen
Aggregatfunktionen	avg, bucket_percentile, count, histogram_quantile, item_count, kurtosis, mad, max, min, skewness, stddevpop, stddevsamp, sum, sumofsquares, varpop, varsamp
Foreach-Funktionen	avg_foreach, bucket_rate_foreach, count_foreach, exists_foreach, last_foreach, max_foreach, min_foreach, sum_foreach
Bitweise Funktionen	bitand, bitlshift, bitnot, bitor, bitrshift, bitxor
Datums- und Zeitfunktionen	date, dayofmonth, dayofweek, now, time

Funktionsgruppe	Funktionen
Verlaufsfunktionen	change, changecount, count, countunique, find, first, firstclock, fuzzytime, jsonpath, last, lastclock, logeventid, logseverity, logsource, logtimestamp, monodec, monoinc, nodata, percentile, rate, xmlxpath
Trendfunktionen	baselinedev, baselinewma, trendavg, trendcount, trendmax, trendmin, trendstl, trendsum
Mathematische Funktionen	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
Operatorfunktionen	between, in
Prädiktive Funktionen	forecast, timeleft
Zeichenfolgenfunktionen	ascii, bitlength, bytelength, char, concat, contains, insert, jsonpath, left, length, ltrim, mid, repeat, replace, right, rtrim, substring, trim, xmlxpath

Sofern nicht anders angegeben, werden diese Funktionen unterstützt in:

- Auslöserausdrücken
- Berechneten Datenpunkten
- Ausdrucksmakros

Foreach-Funktionen werden nur für Aggregatberechnungen unterstützt.

In der Regel geben Funktionen numerische Werte für Vergleiche zurück. Bei der Rückgabe von Zeichenfolgen ist ein Vergleich mit den Operatoren **=** und **<>** möglich (siehe das Beispiel Nicht übereinstimmende Software auf verschiedenen Hosts erkennen).

Funktionsparameter

Mit Funktionsparametern können Sie Folgendes angeben:

- den Datenpunktschlüssel (als `/host/key`) für Funktionen, die auf den Verlauf eines Host-Datenpunkts verweisen
- den Zeitraum (sowie weitere funktionsspezifische Parameter)
- andere Ausdrücke

Datenpunktschlüssel

Der referenzierte Datenpunkt muss sich in einem unterstützten Zustand befinden (mit Ausnahme der Funktion **nodata()**, die auch für nicht unterstützte Datenpunkte berechnet wird).

Das Weglassen des Host-Namens im ersten Parameter (d. h. wie in `function(//key,parameter,...)`) wird nur in bestimmten Kontexten unterstützt:

- In der Formel berechneter Datenpunkte
- In Ausdrucksmakros, die verwendet werden können in:
  - dem Feld *Ereignisname*
  - dem Diagrammnamen
  - der Beschriftung von „Host“- und „Auslöser“- Kartenelementen

In diesen Kontexten können Sie auch das Makro `{HOST.HOST}` verwenden. `{HOST.HOST<1-9>}` kann im Fall des Feldes *Ereignisname* und des Kartenelements „Auslöser“ verwendet werden, um auf einen bestimmten Datenpunkt im Auslöserausdruck zu verweisen. Wenn der Host-Name in diesen Kontexten weggelassen oder durch `{HOST.HOST}` ersetzt wird, verweist die Referenz auf den ersten Datenpunkt im Auslöserausdruck oder auf den ersten Datenpunkt im Diagramm. Außerhalb dieser unterstützten Kontexte führt das Weglassen des Host-Namens in Auslöserausdrücken zu einem Fehler. Ein Beispiel zur Veranschaulichung der Verwendung von Doppelschrägstrichen in Ereignisname-Makros finden Sie unter Vergleich langfristiger CPU-Auslastungen.

Zeitraum

Parameterspezifische Parameter werden nach dem Datenpunktschlüssel angegeben und durch ein Komma vom Datenpunktschlüssel getrennt.

Die meisten numerischen Funktionen akzeptieren einen **Zeitraum** als Parameter.
Damit können Sie das Intervall angeben, für das Sie sich interessieren.
Er kann als Zeitraum (30s, 10m, 1h) oder als Wertebereich (#5 - für die fünf neuesten Werte) angegeben werden.

Sie können Sekunden oder Zeiteinheiten verwenden, um den Zeitraum anzugeben.
Dem Parameter wird ein Rautezeichen vorangestellt, hat er eine andere Bedeutung:

Ausdruck	Beschreibung
sum(/host/key,**10m)**	Summe der Werte der letzten 10 Minuten.
sum(/host/key,**#10)**	Summe der letzten zehn Werte.

Parameter mit einem Rautezeichen haben bei der Funktion **last** eine andere Bedeutung - sie bezeichnen den N-ten vorherigen Wert. Bei den Werten 30, 70, 20, 60, 50 (vom neuesten zum ältesten) gilt daher:

- `last(/host/key,#2)` gibt „70“ zurück
- `last(/host/key,#5)` gibt „50“ zurück

Der Zeitraum wird bis „jetzt“ gemessen - wobei „jetzt“ der Zeitpunkt der letzten Neuberechnung des Auslösers ist (siehe Berechnungshäufigkeit); „jetzt“ ist nicht der aktuelle Zeitpunkt des Servers.

Der Zeitraum gibt entweder Folgendes an:

- Alle Werte zwischen „jetzt - Zeitraum“ und „jetzt“ (oder bei angewendetem Zeitversatz zwischen „jetzt - Zeitversatz - Zeitraum“ und „jetzt“).
- Die angegebene Anzahl von Werten aus der Vergangenheit bis „jetzt“.

Wenn für den angegebenen Auswertungszeitraum keine Daten vorhanden sind, wird der Auslöser oder der berechnete Datenpunkt, der diese Funktion verwendet, nicht unterstützt.

Beachten Sie:

- Wenn im Auslöser nur eine einzelne Funktion (die auf Datenhistorie verweist) verwendet wird, entspricht „jetzt“ immer dem Zeitpunkt des zuletzt empfangenen Werts. Wenn der letzte Wert beispielsweise vor einer Stunde empfangen wurde, wird der Zeitraum bis zum letzten Wert vor einer Stunde betrachtet.
- Ein neuer Auslöser wird berechnet, sobald der erste Wert empfangen wird (Historienfunktionen); bei den Funktionen Datum und Uhrzeit und nodata() erfolgt die Berechnung innerhalb von 30 Sekunden. Der Auslöser wird daher berechnet, obwohl der festgelegte Zeitraum (beispielsweise eine Stunde) seit der Erstellung des Auslösers möglicherweise noch nicht verstrichen ist. Der Auslöser wird auch nach dem ersten Wert berechnet, obwohl der Zeitraum beispielsweise auf die zehn neuesten Werte festgelegt wurde.

Zeitverschiebung

Eine optionale Zeitverschiebung wird mit der Zeit- oder Wertanzahl als Funktionsparameter unterstützt. Mit diesem Parameter können Sie auf Daten aus einem vergangenen Zeitraum verweisen.

Die Zeitverschiebung beginnt mit `now` - zur Angabe der aktuellen Zeit - und wird gefolgt von `+N<time unit>` oder `-N<time unit>` - zum Addieren oder Subtrahieren von N Zeiteinheiten.

Beispielsweise gibt `avg(/host/key,1h:now-1d)` den Durchschnittswert für eine Stunde vor einem Tag zurück.

Attention:
Eine in Monaten (M) und Jahren (y) angegebene Zeitverschiebung wird nur für Trendfunktionen unterstützt. Andere Funktionen unterstützen Sekunden (s), Minuten (m), Stunden (h), Tage (d) und Wochen (w).

Zeitverschiebung mit absoluten Zeiträumen

Absolute Zeiträume werden im Parameter für die Zeitverschiebung unterstützt, beispielsweise Mitternacht bis Mitternacht für einen Tag, Montag bis Sonntag für eine Woche oder der erste bis zum letzten Tag des Monats für einen Monat.

Die Zeitverschiebung für absolute Zeiträume beginnt mit `now` - zur Angabe der aktuellen Zeit - und wird von einer beliebigen Anzahl von Zeitoperationen gefolgt: `/<time unit>` - definiert den Beginn und das Ende der Zeiteinheit, beispielsweise Mitternacht bis Mitternacht für einen Tag, `+N<time unit>` oder `-N<time unit>` - zum Addieren oder Subtrahieren von N Zeiteinheiten.

Beachten Sie, dass der Wert der Zeitverschiebung größer oder gleich 0 sein kann, während der Mindestwert des Zeitraums 1 beträgt.

Parameter	Beschreibung
1d:now/d	Gestern
1d:now/d+1d	Heute
2d:now/d+1d	Letzte 2 Tage
1w:now/w	Letzte Woche
1w:now/w+1w	Diese Woche

Andere Ausdrücke

Funktionsparameter können andere Ausdrücke enthalten, wie in der folgenden Syntax:

```
min(min(/host/key,1h),min(/host2/key2,1h)*10)
```

Beachten Sie, dass andere Ausdrücke nicht verwendet werden dürfen, wenn die Funktion auf die Datenpunkt-Historie verweist. Zum Beispiel ist die folgende Syntax **nicht** zulässig:

~~`min(/host/key,#5*10)`~~

Während andere Ausdrücke von Auslösern als Funktionsparameter in Auslösern auf Nicht-Historienfunktionen beschränkt sind, gilt diese Einschränkung nicht für berechnete Datenpunkte.

Operatoren

Die folgenden Operatoren werden für Auslöser unterstützt **(in absteigender Ausführungspriorität)**:

Priorität	Operator	Definition	**Hinweise zu unbekannten Werten**	Operand zwangsweise in Float umwandeln [1]
1	**-**	Unäres Minus	**-**Unknown → Unknown	Ja
2	**not**	Logisches NICHT	**not** Unknown → Unknown	Ja
3	*	Multiplikation	0 * Unknown → Unknown (ja, Unknown, nicht 0 – damit Unknown bei arithmetischen Operationen nicht verloren geht) 1.2 * Unknown → Unknown	Ja
	/	Division	Unknown / 0 → Fehler Unknown / 1.2 → Unknown 0.0 / Unknown → Unknown	Ja
4	+	Arithmetisches Plus	1.2 + Unknown → Unknown	Ja
	-	Arithmetisches Minus	1.2 - Unknown → Unknown	Ja
5	<	Kleiner als. Der Operator ist wie folgt definiert: A<B ⇔ (A<B-0.000001)	1.2 < Unknown → Unknown	Ja
	<=	Kleiner oder gleich. Der Operator ist wie folgt definiert: A<=B ⇔ (A≤B+0.000001)	Unknown <= Unknown → Unknown	Ja
	>	Größer als. Der Operator ist wie folgt definiert: A>B ⇔ (A>B+0.000001)		Ja
	>=	Größer oder gleich. Der Operator ist wie folgt definiert: A>=B ⇔ (A≥B-0.000001)		Ja

Priorität	Operator	Definition	**Hinweise zu unbekannten Werten**	Operand zwangsweise in Float umwandeln [1]
6	**=**	Ist gleich. Der Operator ist wie folgt definiert: A=B ⇔ (A≥B-0.000001) and (A≤B+0.000001)		Nein [1]
	<>	Ist ungleich. Der Operator ist wie folgt definiert: A<>B ⇔ (A<B-0.000001) or (A>B+0.000001)		Nein [1]
7	**and**	Logisches UND	0 **and** Unknown → 0 1 **and** Unknown → Unknown Unknown **and** Unknown → Unknown	Ja
8	**or**	Logisches ODER	1 **or** Unknown → 1 0 **or** Unknown → Unknown Unknown **or** Unknown → Unknown	Ja

[1] Ein String-Operand wird dennoch in numerisch umgewandelt, wenn:

- ein anderer Operand numerisch ist
- ein anderer Operator als **=** oder **<>** für einen Operanden verwendet wird

(Falls die Umwandlung fehlschlägt, wird der numerische Operand in einen String-Operanden umgewandelt und beide Operanden werden als Strings verglichen.)

Die Operatoren **not**, **and** und **or** sind case-sensitive und müssen kleingeschrieben werden. Außerdem müssen sie von Leerzeichen oder Klammern umgeben sein.

Alle Operatoren außer dem unären **-** und **not** sind linksassoziativ. Das unäre **-** und **not** sind nicht assoziativ (das heißt, **-(-1)** und **not (not 1)** sollten anstelle von **--1** und **not not 1** verwendet werden).

Ergebnis der Auswertung:

- Die Operatoren **<**, **<=**, **>**, **>=**, **=**, **<>** liefern im Auslöser-Ausdruck den Wert '1', wenn die angegebene Relation wahr ist, und '0', wenn sie falsch ist. Wenn mindestens ein Operand Unknown ist, ist das Ergebnis Unknown;
- **and** liefert bei bekannten Operanden '1', wenn beide Operanden ungleich '0' sind; andernfalls liefert es '0'; bei unbekannten Operanden liefert **and** nur dann '0', wenn ein Operand gleich '0' ist; andernfalls liefert es 'Unknown';
- **or** liefert bei bekannten Operanden '1', wenn mindestens einer seiner Operanden ungleich '0' ist; andernfalls liefert es '0'; bei unbekannten Operanden liefert **or** nur dann '1', wenn ein Operand ungleich '0' ist; andernfalls liefert es 'Unknown';
- Das Ergebnis des logischen Negationsoperators **not** für einen bekannten Operanden ist '0', wenn der Wert seines Operanden ungleich '0' ist; '1', wenn der Wert seines Operanden gleich '0' ist. Für einen unbekannten Operanden liefert **not** 'Unknown'.

Unbekannter Ausdruckszustand

Es ist möglich, dass ein unbekannter Operand in einem Auslöserausdruck erscheint, wenn:

- ein nicht unterstützter Datenpunkt verwendet wird
- die Funktionsauswertung für einen unterstützten Datenpunkt zu einem Fehler führt

In diesem Fall wird der Auslöserausdruck im Allgemeinen als `Unknown` ausgewertet (da er nicht ausgewertet werden kann)

Es ist möglich, Benachrichtigungen zu erhalten bei unbekannten Auslösern.

Ausnahmen

Trotz eines unbekannten Operanden können Auslöserausdrücke in einigen Fällen zu einem bekannten Ergebnis (Problem/OK) ausgewertet werden:

- Die Funktion `nodata()` wird unabhängig davon ausgewertet, ob der referenzierte Datenpunkt unterstützt wird oder nicht.
- Ausdrücke mit AND/OR können in zwei Fällen zu einem bekannten Ergebnis ausgewertet werden:
  - **Fall 1**: "`1 or some_function(unsupported_item1) or some_function(unsupported_item2) or ...`" wird zu einem bekannten Ergebnis ausgewertet ('1' oder "Problem"),
  - **Fall 2**: "`0 and some_function(unsupported_item1) and some_function(unsupported_item2) and ...`" wird zu einem bekannten Ergebnis ausgewertet ('0' oder "OK").
- Wenn die Funktionsauswertung für einen unterstützten Datenpunkt zu einem Fehler führt, wird der Funktionswert zu `Unknown` und geht als unbekannter Operand in die weitere Auswertung des Ausdrucks ein.

Unbekannte Operanden können nur in logischen Ausdrücken, wie oben beschrieben, "verschwinden". In arithmetischen Ausdrücken führen unbekannte Operanden immer zu `Unknown` (außer bei Division durch 0).

Attention:
Der Zustand eines unbekannten Ausdrucks ändert den Zustand des Auslösers nicht (Problem/OK). Wenn sich der Zustand des Auslösers also im Zustand "Problem" befand (siehe Fall 1), bleibt er im Problemzustand, selbst wenn der bekannte Teil aufgelöst wird ('1' wird zu '0'), da der Ausdruck nun zu `Unknown` ausgewertet wird und dies den Zustand des Auslösers nicht ändert.

Wenn ein Auslöserausdruck mit mehreren nicht unterstützten Datenpunkten zu `Unknown` ausgewertet wird, bezieht sich die Fehlermeldung im Frontend auf den zuletzt ausgewerteten nicht unterstützten Datenpunkt.

Wert-Caching

Die für die Auslöserauswertung erforderlichen Werte werden vom Zabbix Server zwischengespeichert. Daher verursacht die Auswertung von Auslösern für einige Zeit nach einem Neustart des Servers eine höhere Datenbanklast.

Der Wert-Cache wird nicht geleert, wenn Verlaufswerte von Datenpunkten entfernt werden (entweder manuell oder durch den Housekeeper), sodass der Server die zwischengespeicherten Werte verwendet, bis sie älter sind als die in Auslöserfunktionen definierten Zeiträume oder der Server neu gestartet wird.

Note:
Wenn sich keine aktuellen Daten im Cache befinden und in der Funktion kein Abfragezeitraum definiert ist, greift Zabbix standardmäßig bis zu eine Woche in die Vergangenheit zurück, um historische Werte aus der Datenbank abzufragen.

Aggregatfunktionen

Sofern nicht anders angegeben, werden alle hier aufgeführten Funktionen unterstützt in:

- Ausdrucks für Auslöser
- Formeln für berechnete Datenpunkte
- Ausdrucksmakros

Aggregatfunktionen können entweder mit Folgendem arbeiten:

- der Historie von Datenpunkten, zum Beispiel `min(/host/key,1h)`
- foreach-Funktionen als einzigem Parameter, zum Beispiel `min(last_foreach(/*/key))` (nur in Formeln für berechnete Datenpunkte; um sie in Auslösern zu verwenden, verweisen Sie auf den berechneten Datenpunkt über seinen Schlüssel)

Die Funktionen werden ohne zusätzliche Informationen aufgeführt. Klicken Sie auf die Funktion, um die vollständigen Details anzuzeigen.

Function	Description
avg	Der Durchschnittswert eines Datenpunkts innerhalb des definierten Auswertungszeitraums.
bucket_percentile	Berechnet das Perzentil aus den Buckets eines Histogramms.
count	Die Anzahl der Werte in einem von einer foreach-Funktion zurückgegebenen Array.
histogram_quantile	Berechnet das φ-Quantil aus den Buckets eines Histogramms.
item_count	Die Anzahl der in der Konfiguration vorhandenen Datenpunkte, die den Filterkriterien entsprechen.
kurtosis	Die "Schwanzigkeit" der Wahrscheinlichkeitsverteilung der gesammelten Werte innerhalb des definierten Auswertungszeitraums.
mad	Die mediane absolute Abweichung der gesammelten Werte innerhalb des definierten Auswertungszeitraums.
max	Der höchste Wert eines Datenpunkts innerhalb des definierten Auswertungszeitraums.
min	Der niedrigste Wert eines Datenpunkts innerhalb des definierten Auswertungszeitraums.

Function	Description
skewness	Die Asymmetrie der Wahrscheinlichkeitsverteilung der gesammelten Werte innerhalb des definierten Auswertungszeitraums.
stddevpop	Die Populationsstandardabweichung der gesammelten Werte innerhalb des definierten Auswertungszeitraums.
stddevsamp	Die Stichprobenstandardabweichung der gesammelten Werte innerhalb des definierten Auswertungszeitraums.
sum	Die Summe der gesammelten Werte innerhalb des definierten Auswertungszeitraums.
sumofsquares	Die Summe der Quadrate der gesammelten Werte innerhalb des definierten Auswertungszeitraums.
varpop	Die Populationsvarianz der gesammelten Werte innerhalb des definierten Auswertungszeitraums.
varsamp	Die Stichprobenvarianz der gesammelten Werte innerhalb des definierten Auswertungszeitraums.

Allgemeine Parameter

- `/host/key` ist ein gemeinsamer obligatorischer erster Parameter für die Funktionen, die auf die Verlaufsdaten des Datenpunkts des Hosts verweisen.
- `(sec|#num)<:time shift>` ist ein gemeinsamer zweiter Parameter für die Funktionen, die auf die Verlaufsdaten des Datenpunkts des Hosts verweisen, wobei:
  - **sec** – maximaler Auswertungszeitraum in Sekunden (Zeitsuffixe(/manual/appendix/suffixes) können verwendet werden).
  - **#num** – maximaler Auswertungsbereich in den zuletzt erfassten Werten (wenn ein Rautezeichen vorangestellt ist).
  - **time shift** (optional) ermöglicht, den Auswertungspunkt zeitlich zurückzuverschieben. Weitere Informationen zur Angabe einer Zeitverschiebung finden Sie unter Weitere Details.

Funktionsdetails

Einige allgemeine Hinweise zu Funktionsparametern:

- Funktionsparameter werden durch ein Komma getrennt
- Optionale Funktionsparameter (oder Parameterteile) werden durch < > gekennzeichnet
- Funktionsspezifische Parameter werden bei jeder Funktion beschrieben
- Die Parameter `/host/key` und `(sec|#num)<:time shift>` dürfen niemals in Anführungszeichen gesetzt werden

avg(/host/key,(sec|#num)<:time shift>)

Der Durchschnittswert eines Datenpunkts innerhalb des definierten Auswertungszeitraums.
 Unterstützte Wertetypen: *Float*, *Integer*.
 Unterstützte foreach-Funktionen: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*.

Parameter: siehe allgemeine Parameter.

Die Zeitverschiebung ist nützlich, wenn der aktuelle Durchschnittswert mit dem Durchschnittswert zu einem früheren Zeitpunkt verglichen werden soll.

Beispiele:

```
avg(/host/key,1h) #berechnet den Durchschnittswert für die letzte Stunde bis jetzt
avg(/host/key,#5) #berechnet den Durchschnittswert der fünf neuesten Werte
avg(/host/key,1h:now-1d) #berechnet den Durchschnittswert für eine Stunde im Zeitraum von vor 25 Stunden b
avg(last_foreach(/*/system.cpu.load[,avg1]?[group="MySQL Servers"])) #berechnet die durchschnittliche Proz
avg(/host/proc.num,5m)>300 #Auslöser, wenn die durchschnittliche Anzahl der Prozesse in den letzten 5 Minu
```

bucket_percentile(item filter,time period,percentage)

Berechnet das Perzentil aus den Buckets eines Histogramms.

Parameter:

- **item filter** - siehe item filter;
- **time period** - siehe time period;
- **percentage** - Prozentsatz (0-100).

Kommentare:

- Nur in berechneten Datenpunkten unterstützt;
- Diese Funktion ist ein Alias für `histogram_quantile(percentage/100, bucket_rate_foreach(item filter, time period, 1))`.

Beispiele:

```
bucket_percentile(/*/http_request_duration_seconds_bucket[*],5m,95) #calculate the 95th percentile respons
bucket_percentile(/*/apiserver_request_duration_seconds_bucket[*],10m,99) #calculate the 99th percentile A
```

count(func_foreach(item filter,<time period>),<operator>,<pattern>)

Die Anzahl der Werte in einem Array, das von einer foreach-Funktion zurückgegeben wird.
 Unterstützte foreach-Funktionen: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*.

Parameter:

- **func_foreach** - foreach-Funktion, für die die Anzahl der zurückgegebenen Werte gezählt werden soll. Weitere Informationen finden Sie unter foreach-Funktionen. Beachten Sie, dass count_foreach und bucket_rate_foreach zusätzliche Parameter unterstützen.
- **item filter** - siehe item filter;
- **time period** - siehe time period;
- **operator** (muss in doppelte Anführungszeichen gesetzt werden). Unterstützte `operators`:
eq - gleich
ne - ungleich
gt - größer
ge - größer oder gleich
lt - kleiner
le - kleiner oder gleich
like - stimmt überein, wenn das Muster enthalten ist (Groß-/Kleinschreibung wird beachtet)
bitand - bitweises UND
regexp - Groß-/Kleinschreibung beachtender Abgleich mit dem in `pattern` angegebenen regulären Ausdruck
iregexp - Groß-/Kleinschreibung nicht beachtender Abgleich mit dem in `pattern` angegebenen regulären Ausdruck
- **pattern** - das erforderliche Muster (String-Argumente müssen in doppelte Anführungszeichen gesetzt werden); wird unterstützt, wenn *operator* als dritter Parameter angegeben ist.

Kommentare:

- Die Verwendung von **count()** mit einer auf Historie basierenden foreach-Funktion (max_foreach, avg_foreach usw.) kann Auswirkungen auf die Leistung haben, während die Verwendung von **exists_foreach()**, die nur mit Konfigurationsdaten arbeitet, keinen solchen Effekt hat.
- Optionale Parameter *operator* oder *pattern* dürfen nach einem Komma nicht leer gelassen werden, sondern nur vollständig weggelassen werden.
- Bei *bitand* als drittem Parameter kann der vierte Parameter `pattern` als zwei durch '/' getrennte Zahlen angegeben werden: **number_to_compare_with/mask**. count() berechnet das "bitweise UND" aus dem Wert und der *mask* und vergleicht das Ergebnis mit *number_to_compare_with*. Wenn das Ergebnis des "bitweisen UND" gleich *number_to_compare_with* ist, wird der Wert gezählt.
Wenn *number_to_compare_with* und *mask* gleich sind, muss nur die *mask* angegeben werden (ohne '/').
- Bei *regexp* oder *iregexp* als drittem Parameter kann der vierte Parameter `pattern` ein gewöhnlicher oder globaler (beginnend mit '@') regulärer Ausdruck sein. Bei globalen regulären Ausdrücken wird die Groß-/Kleinschreibung von den Einstellungen des globalen regulären Ausdrucks übernommen. Für den Abgleich per regexp werden Fließkommawerte immer mit 4 Dezimalstellen nach dem '.' dargestellt. Beachten Sie außerdem, dass bei großen Zahlen Unterschiede in der dezimalen (in der Datenbank gespeicherten) und binären (vom Zabbix Server verwendeten) Darstellung die 4. Dezimalstelle beeinflussen können.

Beispiele:

```
count(max_foreach(/*/net.if.in[*],1h)) #berechnet die Anzahl der net.if.in-Datenpunkte, die bis jetzt in d
count(last_foreach(/*/vfs.fs.size[*,pused]),"gt",95) #berechnet die Anzahl der Dateisysteme, bei denen meh
```

histogram_quantile(quantile,bucket1,value1,bucket2,value2,...)

Berechnet das φ-Quantil aus den Buckets eines Histogramms.
 Unterstützte foreach-Funktion: *bucket_rate_foreach*.

Parameter:

- **quantile** - $0 \leq \varphi \leq 1$;
- **bucketN, valueN** - manuell eingegebene Parameterpaare (>=2) oder die Antwort von bucket_rate_foreach.

Kommentare:

- Nur in berechneten Datenpunkten unterstützt;
- Entspricht funktional 'histogram_quantile' aus PromQL;
- Gibt -1 zurück, wenn die Werte des letzten 'Infinity'-Buckets ("+*inf*") gleich 0 sind.

Beispiele:

```
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)

Die Anzahl der in der Konfiguration vorhandenen Datenpunkte, die den Filterkriterien entsprechen.
 Unterstützter Werttyp: *Integer*.

Parameter:

- **item filter** - Kriterien für die Auswahl von Datenpunkten; ermöglicht Referenzen nach Host-Gruppe, Host, Datenpunktschlüssel und Tags. Platzhalter werden unterstützt. Siehe item filter für weitere Details.

Kommentare:

- Nur in berechneten Datenpunkten unterstützt;
- Funktioniert als Alias für die Funktion *count(exists_foreach(item_filter))*.

Beispiele:

```
item_count(/*/agent.ping?[group="Host group 1"]) #calculate the number of hosts with the agent.ping item i
```

kurtosis(/host/key,(sec|#num)<:time shift>)

Die „Schwänzigkeit“ der Wahrscheinlichkeitsverteilung der erfassten Werte innerhalb des definierten Auswertungszeitraums. Siehe auch: Kurtosis.
 Unterstützte Werttypen: *Float*, *Integer*.
 Unterstützte foreach-Funktion: *last_foreach*.

Parameter: siehe allgemeine Parameter.

Beispiele:

```
kurtosis(/host/system.cpu.util[,user],5m)>5 #trigger on sharp spikes of CPU utilization over 5 minutes
```

mad(/host/key,(sec|#num)<:time shift>)

Die mittlere absolute Abweichung vom Median der erfassten Werte innerhalb des definierten Auswertungszeitraums. Siehe auch: Median absolute deviation.
 Unterstützte Wertetypen: *Float*, *Integer*.
 Unterstützte foreach-Funktion: *last_foreach*.

Parameter: siehe allgemeine Parameter.

Beispiele:

```
last(/host/system.cpu.util[,user])>avg(/host/system.cpu.util[,user],1h)+3*mad(/host/system.cpu.util[,user]
```

max(/host/key,(sec|#num)<:time shift>)

Der höchste Wert eines Datenpunkts innerhalb des definierten Auswertungszeitraums.
 Unterstützte Werttypen: *Float*, *Integer*.
 Unterstützte foreach-Funktionen: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*.

Parameter: siehe allgemeine Parameter.

Beispiele:

```
max(/host/key,1h) - min(/host/key,1h) #berechnet die Differenz zwischen den maximalen und minimalen Werten
max(last_foreach(/*/vfs.fs.size[*,pused]?[group="Linux servers"])) #gibt die höchste Festplattenauslastung
```

min(/host/key,(sec|#num)<:time shift>)

Der niedrigste Wert eines Datenpunkts innerhalb des definierten Auswertungszeitraums.
 Unterstützte Wertetypen: *Float*, *Integer*.
 Unterstützte foreach-Funktionen: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*.

Parameter: siehe allgemeine Parameter.

Beispiele:

```
max(/host/key,1h) - min(/host/key,1h) #calculate the difference between the maximum and minimum values wit
min(last_foreach(/*/vfs.file.contents["/sys/class/net/enp0s3/operstate"]?[group="Linux servers"])) #return
```

skewness(/host/key,(sec|#num)<:time shift>)

Die Asymmetrie der Wahrscheinlichkeitsverteilung der erfassten Werte innerhalb des definierten Auswertungszeitraums. Siehe auch: Skewness.
 Unterstützte Wertetypen: *Float*, *Integer*.
 Unterstützte foreach-Funktion: *last_foreach*.

Parameter: siehe allgemeine Parameter.

Beispiele:

```
skewness(/host/vm.memory.size[used],5m)<-20 #Auslöser bei plötzlichen negativen Einbrüchen, die auf Abstür
```

stddevpop(/host/key,(sec|#num)<:time shift>)

Die Populationsstandardabweichung der erfassten Werte innerhalb des definierten Auswertungszeitraums. Siehe auch: Standard-abweichung.
 Unterstützte Wertetypen: *Float*, *Integer*.
 Unterstützte foreach-Funktion: *last_foreach*.

Parameter: siehe allgemeine Parameter.

Beispiele:

```
stddevpop(/host/system.cpu.util[,user],10m)>7 #Auslöser, wenn die CPU-Auslastung stark schwankt (die Popul
```

stddevsamp(/host/key,(sec|#num)<:time shift>)

Die Stichproben-Standardabweichung der erfassten Werte innerhalb des definierten Auswertungszeitraums. Siehe auch: Standard-abweichung.
 Unterstützte Wertetypen: *Float*, *Integer*.
 Unterstützte foreach-Funktion: *last_foreach*.

Parameter: siehe allgemeine Parameter.

Mindestens zwei Datenwerte sind erforderlich, damit diese Funktion arbeitet.

Beispiele:

```
stddevsamp(/host/system.cpu.util[,user],10m)>7 #Auslöser, wenn die CPU-Auslastung stark schwankt (die Stic
```

sum(/host/key,(sec|#num)<:time shift>)

Die Summe der erfassten Werte innerhalb des definierten Auswertungszeitraums.
 Unterstützte Wertetypen: *Float*, *Integer*.
 Unterstützte foreach-Funktionen: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*.

Parameter: siehe allgemeine Parameter.

Beispiele:

```
sum(last_foreach(/*/net.if.in[*]?[group="Linux servers"])) #calculate the total incoming network traffic f
sum(last_foreach(/*/vfs.fs.size[/,total]?[group="MySQL Servers"])) #calculate the total disk space for all
sum(last_foreach(/*/net.dns[*,*,*])) #calculate the total number of successful DNS checks
```

sumofsquares(/host/key,(sec|#num)<:time shift>)

Die Summe der Quadrate der erfassten Werte innerhalb des definierten Auswertungszeitraums.
 Unterstützte Wertetypen: *Float*, *Integer*.
 Unterstützte foreach-Funktion: *last_foreach*.

Parameter: siehe allgemeine Parameter.

Beispiele:

```
sumofsquares(/host/calculated.net.if.rate,1m) #die Summe der Quadrate für die Netzwerkverkehrsrate (eingeh
sumofsquares(last_foreach(/*/net.if.in[*]?[group="Linux servers"])) #die Summe der Quadrate der neuesten e
```

varpop(/host/key,(sec|#num)<:time shift>)

Die Populationsvarianz der erfassten Werte innerhalb des definierten Auswertungszeitraums. Siehe auch: Varianz.
 Unterstützte Wertetypen: *Float*, *Integer*.
 Unterstützte foreach-Funktion: *last_foreach*.

Parameter: siehe allgemeine Parameter.

Beispiele:

```
varpop(/host/system.cpu.util[,user],10m)>50 #Auslöser, wenn die CPU-Auslastung stark schwankt (Varianz lie
```

varsamp(/host/key,(sec|#num)<:time shift>)

Die Stichprobenvarianz der erfassten Werte innerhalb des definierten Auswertungszeitraums. Siehe auch: Varianz.
 Unterstützte Wertetypen: *Float*, *Integer*.
 Unterstützte foreach-Funktion: *last_foreach*.

Parameter: siehe allgemeine Parameter.

Mindestens zwei Datenwerte sind erforderlich, damit diese Funktion funktioniert.

Beispiele:

```
varsamp(/host/system.cpu.util[,user],10m)>50 #Auslöser, wenn die CPU-Auslastung stark schwankt (Stichprobe
```

Siehe alle unterstützten Funktionen.

Foreach-Funktionen

Übersicht

Foreach-Funktionen werden in Aggregatberechnungen verwendet, um für jeden durch den verwendeten **Datenpunktfilter** ausgewählten Datenpunkt einen Aggregatwert zurückzugeben. Es wird ein Array von Werten zurückgegeben.

Beispielsweise gibt die Funktion `avg_foreach` ein Array von Werten zurück, wobei jeder Wert der *Durchschnitts*-Historienwert des ausgewählten Datenpunkts innerhalb des angegebenen Zeitintervalls ist.

> **Attention:**
> Foreach-Funktionen werden nur in Formeln für berechnete Datenpunkte unterstützt. Sie können nicht in einfachen Datenpunktschlüsseln aufgerufen oder direkt zum Erstellen von Ausdrücken für Auslöser verwendet werden. In Ausdrücken für Auslöser kann ein berechneter Datenpunkt verwendet werden, indem auf seinen Schlüssel verwiesen wird.

Der Datenpunktfilter ist Teil der Syntax, die von Foreach-Funktionen verwendet wird. Die Verwendung von Platzhaltern wird im Datenpunktfilter unterstützt, sodass die erforderlichen Datenpunkte sehr flexibel ausgewählt werden können.

Unterstützte Funktionen

Function	Description
avg_foreach	Gibt den Durchschnittswert für jeden Datenpunkt zurück.
bucket_rate_foreach	Gibt Paare (obere Grenze des Buckets, Rate-Wert) zurück, die für die Verwendung in der Funktion `histogram_quantile()` geeignet sind, wobei "obere Grenze des Buckets" der Wert des durch den Parameter <parameter number> definierten item key parameters ist.
count_foreach	Gibt die Anzahl der Werte für jeden Datenpunkt zurück.
exists_foreach	Gibt für jeden aktivierten Datenpunkt '1' zurück.
last_foreach	Gibt den letzten Wert für jeden Datenpunkt zurück.
max_foreach	Gibt den maximalen Wert für jeden Datenpunkt zurück.
min_foreach	Gibt den minimalen Wert für jeden Datenpunkt zurück.
sum_foreach	Gibt die Summe der Werte für jeden Datenpunkt zurück.

Funktionssyntax

Foreach-Funktionen unterstützen zwei gemeinsame Parameter: `Datenpunkt-Filter` (siehe Details unten) und `Zeitperiode`:

```
foreach_function(item filter,time period)
```

Zum Beispiel:

```
avg_foreach(/*/mysql.qps?[group="MySQL Servers"],5m)
```

gibt den Fünf-Minuten-Durchschnitt jedes Datenpunkts „mysql.qps" in der MySQL-Servergruppe zurück.

Beachten Sie, dass einige Funktionen zusätzliche Parameter unterstützen.

Syntax des Datenpunktfilters

Der Datenpunktfilter:

```
/host/key[parameters]?[conditions]
```

besteht aus vier Teilen, wobei:

- host - Host-Name
- key - Datenpunktschlüssel (ohne Parameter)
- parameters - Datenpunktschlüsselparameter
- conditions - auf Host-Gruppen und/oder Datenpunkt-Tags basierende Bedingungen (als Ausdruck)

Leerzeichen sind nur innerhalb des Bedingungsausdrucks zulässig.

Verwendung von Platzhaltern

- Ein Platzhalter kann verwendet werden, um den Host-Namen, den Datenpunktschlüssel oder einen einzelnen Datenpunktschlüsselparameter zu ersetzen.
- Entweder der Host oder der Datenpunktschlüssel muss ohne Platzhalter angegeben werden. Daher sind `/host/*` und `/*/key` gültige Filter, aber `/*/*` ist ungültig.
- Ein Platzhalter kann nicht für einen *Teil* des Host-Namens, des Datenpunktschlüssels oder des Datenpunktschlüsselparameters verwendet werden.

- Ein Platzhalter entspricht nicht mehr als einem einzelnen Datenpunktschlüsselparameter. Daher muss für jeden Parameter separat ein Platzhalter angegeben werden (d. h. `key[abc,*,*]`).

Bedingungsausdruck

Der Bedingungsausdruck unterstützt:

- Operanden:
  - group - Host-Gruppe
  - tag - Datenpunkt-Tag
  - "<text>" - Zeichenkettenkonstante, wobei das Escape-Zeichen \ verwendet wird, um " und \ zu maskieren
- Operatoren für die Groß-/Kleinschreibung beachtende Zeichenkettenvergleiche: =, <>
- logische Operatoren: `and`, `or`, `not`
- Gruppierung mit Klammern: ( )

Die Anführungszeichen für Zeichenkettenkonstanten sind obligatorisch. Es wird nur ein vollständiger, die Groß-/Kleinschreibung beachtender Zeichenkettenvergleich unterstützt.

Warning:
Bei der Angabe von Tags im Filter (d. h. `tag="tagname:value"`) wird der Doppelpunkt ":" als Trennzeichen verwendet. Alles danach wird als Tag-Wert betrachtet. Daher wird die Angabe eines Tag-Namens, der ":" enthält, derzeit nicht unterstützt.

Beispiele

Es kann ein komplexer Filter verwendet werden, der sich auf den Datenpunktschlüssel, die Host-Gruppe und Tags bezieht, wie in den Beispielen dargestellt:

Syntax example	Description
`/host/key[abc,*]`	Entspricht ähnlichen Datenpunkten auf diesem Host.
`/*/key`	Entspricht demselben Datenpunkt auf jedem Host.
`/*/key?[group="ABC" and tag="tagname:value"]`	Entspricht demselben Datenpunkt auf jedem Host aus der Gruppe ABC mit den Tags 'tagname:value'.
`/*/key[a,*,c]?[(group="ABC" and tag="Tag1") or (group="DEF" and (tag="Tag2" or tag="Tag3:value"))]`	Entspricht ähnlichen Datenpunkten auf jedem Host aus der Gruppe ABC oder DEF mit den entsprechenden Tags.

Alle referenzierten Datenpunkte müssen existieren und Daten erfassen. Nur aktivierte Datenpunkte auf aktivierten Hosts werden in die Berechnungen einbezogen. Datenpunkte im nicht unterstützten Zustand werden nicht einbezogen.

Attention:
Wenn der Datenpunktschlüssel eines referenzierten Datenpunkts geändert wird, muss der Filter manuell aktualisiert werden.

Die Angabe einer übergeordneten Host-Gruppe schließt die übergeordnete Gruppe und alle verschachtelten Host-Gruppen mit ihren Datenpunkten ein.

Zeitraum

Mit dem **zweiten** Parameter können Sie den Zeitraum für die Aggregation angeben.
Der Zeitraum kann nur als Zeit ausgedrückt werden; die Anzahl der Werte (mit dem Präfix **#**) wird nicht unterstützt.

In diesem Parameter können zur Vereinfachung unterstützte Einheitensymbole verwendet werden, zum Beispiel `5m` (fünf Minuten) anstelle von `300s` (300 Sekunden) oder `1d` (ein Tag) anstelle von 86400 (86400 Sekunden).

Für die Funktion `last_foreach` ist der Zeitraum ein optionaler Parameter (unterstützt seit Zabbix 7.0) und kann weggelassen werden:

```
last_foreach(/*/key?[group="host group"])
```

Der Zeitraum wird mit der Funktion `exists_foreach` nicht unterstützt.

Zusätzliche Parameter

bucket_rate_foreach

Die Funktion `bucket_rate_foreach` unterstützt einen dritten optionalen Parameter:

```
bucket_rate_foreach(item filter,time period,<parameter number>)
```

wobei <parameter number> die Position des Werts "bucket" im Item-Schlüssel ist. Wenn der Wert "bucket" in myItem[aaa,0.2] beispielsweise 0.2 ist, dann ist seine Position 2.

Der Standardwert von <parameter number> ist 1.

count_foreach

Die Funktion count_foreach unterstützt einen dritten und vierten optionalen Parameter:

```
count_foreach(item filter,time period,<operator>,<pattern>)
```

Dabei gilt:

- **operator** ist der bedingte Operator für Item-Werte (muss in doppelte Anführungszeichen gesetzt werden). Unterstützte operators:
eq - gleich
ne - ungleich
gt - größer
ge - größer oder gleich
lt - kleiner
le - kleiner oder gleich
like - stimmt überein, wenn das Muster enthalten ist (Groß-/Kleinschreibung wird beachtet)
bitand - bitweises UND
regexp - Übereinstimmung mit dem angegebenen regulären Ausdruck unter Beachtung der Groß-/Kleinschreibung in pattern
iregexp - Übereinstimmung mit dem angegebenen regulären Ausdruck ohne Beachtung der Groß-/Kleinschreibung in pattern.
- **pattern** ist das erforderliche Muster (String-Argumente müssen in doppelte Anführungszeichen gesetzt werden); wird unterstützt, wenn *operator* im dritten Parameter angegeben ist.

Hinweise:

- Optionale Parameter *operator* oder *pattern* dürfen nach einem Komma nicht leer gelassen werden, sondern nur vollständig weggelassen werden.
- Bei *bitand* als drittem Parameter kann der vierte Parameter pattern als zwei durch '/' getrennte Zahlen angegeben werden: **number_to_compare_with/mask**. count_foreach() berechnet das "bitweise UND" aus dem Wert und der *mask* und vergleicht das Ergebnis mit *number_to_compare_with*. Wenn das Ergebnis des "bitweisen UND" gleich *number_to_compare_with* ist, wird der Wert gezählt.
Wenn *number_to_compare_with* und *mask* gleich sind, muss nur die *mask* angegeben werden (ohne '/').
- Bei *regexp* oder *iregexp* als drittem Parameter kann der vierte Parameter pattern ein gewöhnlicher oder globaler regulärer Ausdruck sein (beginnend mit '@'). Bei globalen regulären Ausdrücken wird die Groß-/Kleinschreibung von den Einstellungen des globalen regulären Ausdrucks übernommen. Für den Abgleich mit regexp werden Float-Werte immer mit 4 Dezimalstellen nach dem '.' dargestellt. Beachten Sie außerdem, dass bei großen Zahlen Unterschiede in der dezimalen (in der Datenbank gespeicherten) und binären (vom Zabbix Server verwendeten) Darstellung die 4. Dezimalstelle beeinflussen können.

Weitere Details und Beispiele zur Verwendung von foreach-Funktionen finden Sie unter Aggregatberechnungen.

Verhalten in Abhängigkeit von der Verfügbarkeit

Die folgende Tabelle veranschaulicht, wie sich jede Funktion bei eingeschränkter Verfügbarkeit von Host-/Datenpunkt- und Verlaufsdaten verhält.

Function	Deaktivierter Host	Nicht verfügbarer Host mit Daten	Nicht verfügbarer Host ohne Daten	Deaktivierter Datenpunkt	Nicht unterstützter Datenpunkt	Fehler beim Datenabruf (SQL)
avg_foreach	ignorieren	Durchschnitt zurückgeben	ignorieren	ignorieren	ignorieren	ignorieren
bucket_rate_foreach	ignorieren	Bucket-Rate zurückgeben	ignorieren	ignorieren	ignorieren	ignorieren
count_foreach	ignorieren	Anzahl zurückgeben	0	ignorieren	ignorieren	ignorieren
exists_foreach	ignorieren	1	1	ignorieren	1	n/v
last_foreach	ignorieren	Letzten Wert zurückgeben	ignorieren	ignorieren	ignorieren	ignorieren
max_foreach	ignorieren	Maximum zurückgeben	ignorieren	ignorieren	ignorieren	ignorieren
min_foreach	ignorieren	Minimum zurückgeben	ignorieren	ignorieren	ignorieren	ignorieren
sum_foreach	ignorieren	Summe zurückgeben	ignorieren	ignorieren	ignorieren	ignorieren

Wenn der Datenpunkt *ignoriert* wird, wird der Aggregation nichts hinzugefügt.

Bitweise Funktionen

Alle hier aufgeführten Funktionen werden unterstützt in:

- Auslöser-Ausdrücken
- Formeln für berechnete Datenpunkte
- Ausdrucksmakros

Die Funktionen werden ohne zusätzliche Informationen aufgeführt. Klicken Sie auf die Funktion, um die vollständigen Details anzuzeigen.

Function	Description
bitand	Der Wert von ”bitwise AND” eines Datenpunktwerts und einer Maske.
bitlshift	Die bitweise Linksverschiebung eines Datenpunktwerts.
bitnot	Der Wert von ”bitwise NOT” eines Datenpunktwerts.
bitor	Der Wert von ”bitwise OR” eines Datenpunktwerts und einer Maske.
bitrshift	Die bitweise Rechtsverschiebung eines Datenpunktwerts.
bitxor	Der Wert von ”bitwise exclusive OR” eines Datenpunktwerts und einer Maske.

Funktionsdetails

Einige allgemeine Hinweise zu Funktionsparametern:

- Funktionsparameter werden durch ein Komma getrennt
- Ausdrücke werden als Parameter akzeptiert
- Optionale Funktionsparameter (oder Parameterteile) werden durch < > gekennzeichnet

bitand(value,mask)

Der Wert von „bitwise AND“ eines Datenpunktwerts und einer Maske.
 Unterstützte Werttypen: *Integer*.

Parameter:

- **value** - der zu prüfende Wert;
- **mask** (obligatorisch) - eine vorzeichenlose 64-Bit-Ganzzahl (0 - 18446744073709551615).

Obwohl der Vergleich bitweise durchgeführt wird, müssen alle Werte in Dezimaldarstellung angegeben werden und werden auch in Dezimaldarstellung zurückgegeben. Zum Beispiel wird die Prüfung auf das 3. Bit durch den Vergleich mit 4 durchgeführt, nicht mit 100.

Beispiele:

```
bitand(last(/host/key),12)=8 or bitand(last(/host/key),12)=4 #3. oder 4. Bit gesetzt, aber nicht beide gle
bitand(last(/host/key),20)=16 #3. Bit nicht gesetzt und 5. Bit gesetzt
```

bitlshift(value,bits to shift)

Die bitweise Linksverschiebung eines Datenpunktwertes.
 Unterstützte Wertetypen: *Integer*.

Parameter:

- **value** - der zu prüfende Wert;
- **bits to shift** (obligatorisch) - die Anzahl der zu verschiebenden Bits.

Obwohl der Vergleich bitweise durchgeführt wird, müssen alle Werte in Dezimalform angegeben werden und werden auch in Dezimalform zurückgegeben. Zum Beispiel wird die Prüfung des 3. Bits durch den Vergleich mit 4 durchgeführt, nicht mit 100.

bitnot(value)

Der Wert von „bitwise NOT“ eines Datenpunktwertes.
 Unterstützte Werttypen: *Integer*.

Parameter:

- **value** - der zu prüfende Wert.

Obwohl der Vergleich bitweise durchgeführt wird, müssen alle Werte in dezimaler Form angegeben werden und werden auch in dezimaler Form zurückgegeben. Zum Beispiel erfolgt die Prüfung auf das 3. Bit durch den Vergleich mit 4, nicht mit 100.

bitor(value,mask)

Der Wert von „bitwise OR“ eines Datenpunktwerts und einer Maske.
 Unterstützte Wertetypen: *Integer*.

Parameter:

- **value** - der zu prüfende Wert;
- **mask** (obligatorisch) - eine vorzeichenlose 64-Bit-Ganzzahl (0 - 18446744073709551615).

Obwohl der Vergleich bitweise durchgeführt wird, müssen alle Werte in Dezimalform angegeben werden und werden auch in Dezimalform zurückgegeben. Zum Beispiel wird die Prüfung auf das 3. Bit durch den Vergleich mit 4 durchgeführt, nicht mit 100.

bitrshift(value,bits to shift)

Die bitweise Rechtsverschiebung eines Datenpunktwertes.
 Unterstützte Werttypen: *Integer*.

Parameter:

- **value** - der zu prüfende Wert;
- **bits to shift** (obligatorisch) - die Anzahl der zu verschiebenden Bits.

Obwohl der Vergleich bitweise durchgeführt wird, müssen alle Werte in Dezimalform angegeben werden und werden auch in Dezimalform zurückgegeben. Zum Beispiel erfolgt die Prüfung auf das 3. Bit durch den Vergleich mit 4, nicht mit 100.

bitxor(value,mask)

Der Wert von „bitweisem exklusivem ODER“ eines Datenpunkt-Werts und einer Maske.
 Unterstützte Werttypen: *Integer*.

Parameter:

- **value** - der zu prüfende Wert;
- **mask** (obligatorisch) - eine 64-Bit-Ganzzahl ohne Vorzeichen (0 - 18446744073709551615).

Obwohl der Vergleich bitweise durchgeführt wird, müssen alle Werte in Dezimaldarstellung angegeben werden und werden auch in Dezimaldarstellung zurückgegeben. Zum Beispiel wird die Prüfung auf das 3. Bit durch den Vergleich mit 4 durchgeführt, nicht mit 100.

Siehe alle unterstützten Funktionen.

Datums- und Zeitfunktionen

Alle hier aufgeführten Funktionen werden unterstützt in:

- Ausdrucksdefinitionen für Auslöser
- Formeln für berechnete Datenpunkte
- Ausdrucksmakros

> **Attention:**
> Datums- und Zeitfunktionen können nicht allein im Ausdruck verwendet werden; es muss mindestens eine Funktion aus einer anderen Gruppe, die auf den Host-Datenpunkt verweist, in den Ausdruck aufgenommen werden (außer der Funktion nodata()). Detaillierte Informationen dazu, wie Datums- und Zeitfunktionen in Ausdrücken funktionieren, finden Sie unter Berechnungszeit.

Die Funktionen werden ohne zusätzliche Informationen aufgeführt. Klicken Sie auf die Funktion, um die vollständigen Details anzuzeigen.

Function	Description
date	Das aktuelle Datum im Format YYYYMMDD.
dayofmonth	Der Tag des Monats im Bereich von 1 bis 31.
dayofweek	Der Wochentag im Bereich von 1 bis 7.
now	Die Anzahl der Sekunden seit der Unix-Epoche (00:00:00 UTC, 1. Januar 1970).
time	Die aktuelle Uhrzeit im Format HHMMSS.

Funktionsdetails

date

Das aktuelle Datum im Format JJJJMMTT.

Beispiel:

`date()<20220101`

dayofmonth

Der Tag des Monats im Bereich von 1 bis 31.

Beispiel:

```
dayofmonth()=1
```

dayofweek

Der Wochentag im Bereich von 1 bis 7 (Mo - 1, So - 7).

Beispiel (nur Werktage):

```
dayofweek()<6
```

Beispiel (nur Wochenende):

```
dayofweek()>5
```

now

Die Anzahl der Sekunden seit der Epoch (00:00:00 UTC, 1. Januar 1970).

Beispiel:

```
now()<1640998800
```

time

Die aktuelle Uhrzeit im Format HHMMSS.

Beispiel (nur nachts, 00:00-06:00):

```
time()<060000
```

Siehe alle unterstützten Funktionen.

Verlaufsfunktionen

Alle hier aufgeführten Funktionen werden unterstützt in:

- Auslöser-Ausdrücken
- Formeln für berechnete Datenpunkte
- Ausdrucksmakros

Die Funktionen werden ohne zusätzliche Informationen aufgeführt. Klicken Sie auf eine Funktion, um die vollständigen Details anzuzeigen.

Funktion	Beschreibung
change	Die Differenz zwischen dem vorherigen und dem neuesten Wert.
changecount	Die Anzahl der Änderungen zwischen aufeinanderfolgenden Werten innerhalb des definierten Auswertungszeitraums.
count	Die Anzahl der Werte innerhalb des definierten Auswertungszeitraums.
countunique	Die Anzahl der eindeutigen Werte innerhalb des definierten Auswertungszeitraums.
find	Sucht innerhalb des definierten Auswertungszeitraums nach einer Übereinstimmung mit einem Wert.
first	Der erste (älteste) Wert innerhalb des definierten Auswertungszeitraums.
firstclock	Der Zeitstempel des ersten (ältesten) Werts innerhalb des definierten Auswertungszeitraums.
fuzzytime	Prüft, wie stark die Zeit des passiven Agents von der Zeit des Zabbix Servers/Proxys abweicht.
jsonpath	Gibt ein Array mit den Ergebnissen der JSONPath-Abfrage zurück.
last	Der aktuellste Wert.
lastclock	Der Zeitstempel des N.-aktuellsten Werts innerhalb des definierten Auswertungszeitraums.
logeventid	Prüft, ob die Ereignis-ID des letzten Protokolleintrags mit einem regulären Ausdruck übereinstimmt.
logseverity	Der Schweregrad des letzten Protokolleintrags.
logsource	Prüft, ob die Protokollquelle des letzten Protokolleintrags mit einem regulären Ausdruck übereinstimmt.
logtimestamp	Der Zeitstempel der Protokollmeldung des N.-aktuellsten Protokolldatenpunktwerts.
monodec	Prüft, ob die Werte monoton abgenommen haben.
monoinc	Prüft, ob die Werte monoton zugenommen haben.
nodata	Prüft, ob keine Daten empfangen wurden.

Funktion	Beschreibung
percentile	Das P-te Perzentil eines Zeitraums, wobei P (Prozentsatz) durch den dritten Parameter angegeben wird.
rate	Die durchschnittliche Zunahmerate pro Sekunde eines monoton steigenden Zählers innerhalb des definierten Zeitraums.
xmlxpath	Gibt ein Array mit den Ergebnissen der XML-XPath-Abfrage zurück.

Allgemeine Parameter

- `/host/key` ist ein gemeinsamer obligatorischer erster Parameter für die Funktionen, die auf den Verlauf des Datenpunkts des Hosts verweisen.
- `(sec|#num)<:time shift>` ist ein gemeinsamer zweiter Parameter für die Funktionen, die auf den Verlauf des Datenpunkts des Hosts verweisen, wobei:
  - **sec** – maximaler Auswertungszeitraum in Sekunden (Zeit-Suffixe können verwendet werden), oder
  - **#num** – maximaler Auswertungsbereich in den zuletzt erfassten Werten (wenn ein Rautezeichen vorangestellt ist).
  - **time shift** (optional) ermöglicht, den Auswertungspunkt zeitlich zurückzuverschieben. Weitere Details zur Angabe der Zeitverschiebung.

Funktionsdetails

Einige allgemeine Hinweise zu Funktionsparametern:

- Funktionsparameter werden durch ein Komma getrennt
- Optionale Funktionsparameter (oder Parameterteile) werden durch < > angegeben
- Funktionsspezifische Parameter werden bei jeder Funktion beschrieben
- Die Parameter `/host/key` und `(sec|#num)<:time shift>` dürfen niemals in Anführungszeichen gesetzt werden

change(/host/key)

Die Differenz zwischen dem vorherigen und dem neuesten Wert.
 Unterstützte Wertetypen: *Float*, *Integer*, *String*, *Text*, *Log*.
 Für Zeichenfolgen wird zurückgegeben: 0 - Werte sind gleich; 1 - Werte unterscheiden sich.

Parameter: siehe allgemeine Parameter.

Kommentare:

- Die numerische Differenz wird berechnet, wie an diesen eingehenden Beispielwerten zu sehen ist ('vorheriger' und 'neuester' Wert = Differenz):
'1' und '5' = +4
'3' und '1' = −2
'0' und '-2.5' = −2.5.
- In Auslösern nützlich, um plötzliche Spitzen (oder Einbrüche), Zähler-Resets und numerische Zustandsübergänge zu erkennen.
- Siehe auch: abs zum Vergleich.

Beispiele:

```
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>)

Die Anzahl der Änderungen zwischen benachbarten Werten innerhalb des definierten Auswertungszeitraums.
 Unterstützte Werttypen: *Float*, *Integer*, *String*, *Text*, *Log*.

Parameter:

- Siehe allgemeine Parameter
- **mode** (muss in doppelte Anführungszeichen gesetzt werden) – mögliche Werte: *all* – alle Änderungen zählen (Standard); *dec* – Abnahmen zählen; *inc* – Zunahmen zählen

Kommentare:

- Bei nicht numerischen Werttypen wird der Parameter *mode* ignoriert.
- In Auslösern nützlich, um häufige Statusänderungen zu erkennen (ein Hinweis auf Instabilität).

Beispiele:

```
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>)

Die Anzahl der Werte innerhalb des definierten Auswertungszeitraums.
 Unterstützte Werttypen: *Float*, *Integer*, *String*, *Text*, *Log*.

Parameter:

- Siehe allgemeine Parameter
- **operator** (muss in doppelte Anführungszeichen gesetzt werden) Unterstützte `operators`:
eq - gleich (Standard für Integer, Float)
ne - ungleich
gt - größer
ge - größer oder gleich
lt - kleiner
le - kleiner oder gleich
like (Standard für String, Text, Log) - stimmt überein, wenn das Muster enthalten ist (Groß-/Kleinschreibung wird beachtet)
bitand - bitweises UND
regexp - Übereinstimmung mit dem angegebenen regulären Ausdruck unter Beachtung der Groß-/Kleinschreibung in `pattern`
iregexp - Übereinstimmung mit dem angegebenen regulären Ausdruck ohne Beachtung der Groß-/Kleinschreibung in `pattern`
- **pattern** - das erforderliche Muster (String-Argumente müssen in doppelte Anführungszeichen gesetzt werden)

Kommentare:

- Float-Datenpunkte stimmen mit der Genauigkeit von 2.22e-16 überein.
- *like* wird nicht als Operator für Integer-Werte unterstützt.
- *like* und *bitand* werden nicht als Operatoren für Float-Werte unterstützt.
- Für String-, Text- und Log-Werte werden nur die Operatoren *eq*, *ne*, *like*, *regexp* und *iregexp* unterstützt.
- Bei *bitand* als Operator kann der vierte Parameter `pattern` als zwei Zahlen angegeben werden, getrennt durch '/': **number_to_compare_with/mask**. count() berechnet das "bitweise UND" aus dem Wert und der *mask* und vergleicht das Ergebnis mit *number_to_compare_with*. Wenn das Ergebnis des "bitweisen UND" gleich *number_to_compare_with* ist, wird der Wert gezählt.
Wenn *number_to_compare_with* und *mask* gleich sind, muss nur die *mask* angegeben werden (ohne '/').
- Bei *regexp* oder *iregexp* als Operator kann der vierte Parameter `pattern` ein gewöhnlicher oder globaler regulärer Ausdruck sein (beginnend mit '@'). Bei globalen regulären Ausdrücken wird die Groß-/Kleinschreibung von den Einstellungen des globalen regulären Ausdrucks übernommen. Für den Zweck des regexp-Abgleichs werden Float-Werte immer mit 4 Dezimalstellen nach '.' dargestellt. Beachten Sie außerdem, dass bei großen Zahlen Unterschiede in der dezimalen (in der Datenbank gespeicherten) und binären (vom Zabbix Server verwendeten) Darstellung die 4. Dezimalstelle beeinflussen können.

Beispiele:

```
count(/host/icmpping,30m,,"0")>5 #ping ist mehr als 5 Mal in 30 Minuten fehlgeschlagen
count(/host/key,10m,"like","error") #die Anzahl der Werte der letzten 10 Minuten bis jetzt, die 'error' en
count(/host/key,10m,,12) #die Anzahl der Werte der letzten 10 Minuten bis jetzt, die '12' entsprechen
count(/host/key,10m,"gt",12) #die Anzahl der Werte der letzten 10 Minuten bis jetzt, die größer als '12' s
count(/host/key,#10,"gt",12) #die Anzahl der Werte innerhalb der letzten 10 Werte bis jetzt, die größer al
count(/host/key,10m:now-1d,"gt",12) #die Anzahl der Werte zwischen 24 Stunden und 10 Minuten sowie 24 Stun
count(/host/key,10m,"bitand","6/7") #die Anzahl der Werte der letzten 10 Minuten bis jetzt, die in den 3 n
count(/host/key,10m:now-1d) #die Anzahl der Werte zwischen 24 Stunden und 10 Minuten sowie 24 Stunden vor
```

countunique(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

Die Anzahl eindeutiger Werte innerhalb des definierten Auswertungszeitraums.
 Unterstützte Werttypen: *Float*, *Integer*, *String*, *Text*, *Log*.

Parameter:

- Siehe allgemeine Parameter
- **operator** (muss in doppelte Anführungszeichen gesetzt werden). Unterstützte `operators`:
eq - gleich (Standard für Integer, Float)
ne - ungleich
gt - größer
ge - größer oder gleich
lt - kleiner
le - kleiner oder gleich
like (Standard für String, Text, Log) - stimmt überein, wenn das Muster enthalten ist (Groß-/Kleinschreibung wird beachtet)
bitand - bitweises UND
regexp - Übereinstimmung mit dem angegebenen regulären Ausdruck in `pattern` unter Beachtung der Groß-/Kleinschreibung
iregexp - Übereinstimmung mit dem angegebenen regulären Ausdruck in `pattern` ohne Beachtung der Groß-/Kleinschreibung
- **pattern** - das erforderliche Muster (String-Argumente müssen in doppelte Anführungszeichen gesetzt werden)

Kommentare:

- Float-Datenpunkte stimmen mit der Genauigkeit von 2.22e-16 überein.
- *like* wird für Integer-Werte nicht als Operator unterstützt.
- *like* und *bitand* werden für Float-Werte nicht als Operatoren unterstützt.
- Für String-, Text- und Log-Werte werden nur die Operatoren *eq*, *ne*, *like*, *regexp* und *iregexp* unterstützt.
- Bei *bitand* als Operator kann der vierte Parameter `pattern` als zwei Zahlen angegeben werden, getrennt durch '/': **number_to_compare_with/mask**. countunique() berechnet das "bitweise UND" aus dem Wert und der *mask* und vergleicht

das Ergebnis mit *number_to_compare_with*. Wenn das Ergebnis des "bitweisen UND" gleich *number_to_compare_with* ist, wird der Wert gezählt.
Wenn *number_to_compare_with* und *mask* gleich sind, muss nur die *mask* angegeben werden (ohne '/').

- Bei *regexp* oder *iregexp* als Operator kann der vierte Parameter `pattern` ein gewöhnlicher oder globaler regulärer Ausdruck (beginnend mit '@') sein. Bei globalen regulären Ausdrücken wird die Groß-/Kleinschreibung von den Einstellungen des globalen regulären Ausdrucks übernommen. Für den Zweck des regexp-Abgleichs werden Float-Werte immer mit 4 Dezimalstellen nach '.' dargestellt. Beachten Sie außerdem, dass bei großen Zahlen Unterschiede in der dezimalen (in der Datenbank gespeicherten) und binären (vom Zabbix Server verwendeten) Darstellung die 4. Dezimalstelle beeinflussen können.

Beispiele:

```
countunique(/host/key,10m) #die Anzahl eindeutiger Werte für die letzten 10 Minuten bis jetzt
countunique(/host/key,10m,"like","error") #die Anzahl eindeutiger Werte für die letzten 10 Minuten bis jet
countunique(/host/key,10m,,12) #die Anzahl eindeutiger Werte für die letzten 10 Minuten bis jetzt, die '12
countunique(/host/key,10m,"gt",12) #die Anzahl eindeutiger Werte für die letzten 10 Minuten bis jetzt, die
countunique(/host/key,#10,"gt",12) #die Anzahl eindeutiger Werte innerhalb der letzten 10 Werte bis jetzt,
countunique(/host/key,10m:now-1d,"gt",12) #die Anzahl eindeutiger Werte zwischen 24 Stunden und 10 Minuten
countunique(/host/key,10m,"bitand","6/7") #die Anzahl eindeutiger Werte für die letzten 10 Minuten bis jet
countunique(/host/key,10m:now-1d) #die Anzahl eindeutiger Werte zwischen 24 Stunden und 10 Minuten sowie 2
```

find(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

Findet einen Werttreffer innerhalb des definierten Auswertungszeitraums.
 Unterstützte Werttypen: *Float*, *Integer*, *String*, *Text*, *Log*.
 Rückgabewert: 1 - gefunden; 0 - andernfalls.

Parameter:

- Siehe allgemeine Parameter
- **sec** oder **#num** (optional) - standardmäßig der neueste Wert, wenn nicht angegeben
- **operator** (muss in doppelte Anführungszeichen gesetzt werden) Unterstützte `operators`:
eq - gleich (Standard für Integer, Float)
ne - ungleich
gt - größer
ge - größer oder gleich
lt - kleiner
le - kleiner oder gleich
like (Standard für String, Text, Log) - stimmt überein, wenn die in `pattern` angegebene Zeichenfolge enthalten ist (Groß-/Kleinschreibung wird beachtet)
bitand - bitweises UND
regexp - Groß-/Kleinschreibung beachtender Abgleich mit dem in `pattern` angegebenen regulären Ausdruck
iregexp - Groß-/Kleinschreibung nicht beachtender Abgleich mit dem in `pattern` angegebenen regulären Ausdruck
- **pattern** - das erforderliche Muster (String-Argumente müssen in doppelte Anführungszeichen gesetzt werden); Perl Compatible Regular Expression (PCRE)-regulärer Ausdruck, wenn `operator` *regexp*, *iregexp* ist

Kommentare:

- Wenn mehr als ein Wert verarbeitet wird, wird '1' zurückgegeben, wenn mindestens ein Wert übereinstimmt.
- *like* wird nicht als Operator für Integer-Werte unterstützt.
- *like* und *bitand* werden nicht als Operatoren für Float-Werte unterstützt.
- Für String-, Text- und Log-Werte werden nur die Operatoren *eq*, *ne*, *like*, *regexp* und *iregexp* unterstützt.
- Bei *regexp* oder *iregexp* als Operator kann der vierte Parameter `pattern` ein gewöhnlicher oder globaler regulärer Ausdruck sein (beginnend mit '@'). Bei globalen regulären Ausdrücken wird die Groß-/Kleinschreibung von den Einstellungen des globalen regulären Ausdrucks übernommen.

Beispiele:

```
find(/host/key,10m,"like","error") #findet einen Wert, der innerhalb der letzten 10 Minuten bis jetzt 'err
find(/host/agent.version,,"like","beta")=1 #Zabbix Agent hat eine Beta-Version und muss aktualisiert werde
find(/host/log[/var/log/nginx/access.log],,"regexp"," 500 ")=1 #interner Webserverfehler wurde gefunden
```

first(/host/key,sec<:time shift>)

Der erste (der älteste) Wert innerhalb des definierten Auswertungszeitraums.
 Unterstützte Werttypen: *Float*, *Integer*, *String*, *Text*, *Log*.

Parameter:

- Siehe allgemeine Parameter

Siehe auch last().

Beispiel:

```
first(/host/key,1h) #retrieve the oldest value within the last hour until now
```

firstclock(/host/key,sec<:time shift>)

Der Zeitstempel des ältesten Werts innerhalb des definierten Auswertungszeitraums.
 Unterstützte Werttypen: *Float*, *Integer*, *String*, *Text*, *Log*.

Parameter:

- Siehe gemeinsame Parameter.

Die Funktion schlägt mit einem Fehler fehl, wenn im angegebenen Zeitraum keine Daten erfasst wurden.

Siehe auch lastclock().

Beispiele:

```
firstclock(/host/key,1h) #Zeitstempel des ältesten Werts innerhalb der letzten Stunde abrufen
firstclock(/host/key,1h:now-24h) #Zeitstempel des ältesten Werts innerhalb der letzten Stunde vor einem Ta
```

fuzzytime(/host/key,sec)

Prüft, wie stark die Zeit des passiven Agent von der Zeit des Zabbix-Server/Proxy abweicht.
 Unterstützte Werttypen: *Float*, *Integer*.
 Rückgabewert: 1 - die Differenz zwischen dem Wert des passiven Datenpunkts (als Zeitstempel) und dem Zeitstempel des Zabbix-Server/Proxy (der Uhrzeit der Wertsammlung) ist kleiner oder gleich *sec* Sekunden; 0 - andernfalls.

Parameter:

- Siehe allgemeine Parameter.

Kommentare:

- Wird normalerweise mit dem Datenpunkt 'system.localtime' verwendet, um zu prüfen, ob die lokale Zeit mit der lokalen Zeit des Zabbix-Server synchron ist. *Hinweis*: 'system.localtime' muss für den Zabbix Agent als passive Prüfung konfiguriert sein; beim Zabbix Agent 2 kann er als aktive Prüfung konfiguriert werden.
- Kann auch mit dem Schlüssel `vfs.file.time[/path/file,modify]` verwendet werden, um zu prüfen, ob die Datei seit längerer Zeit nicht aktualisiert wurde.
- Diese Funktion wird für die Verwendung in komplexen Ausdrücken von Auslösern (mit mehreren beteiligten Datenpunkten) nicht empfohlen, da sie zu unerwarteten Ergebnissen führen kann (die Zeitdifferenz wird mit der zuletzt erfassten Metrik gemessen), z. B. in `fuzzytime(/Host/system.localtime,60s)=0 or last(/Host/trap)<>0`.

Beispiele:

```
fuzzytime(/host/system.localtime,5m)=0 #client local time differs from Zabbix server/proxy time by more th
fuzzytime(/host/system.localtime,5m)=0 and nodata(/host/system.localtime,10m)=0 #client local time differs
```

jsonpath(/host/key,(sec|#num)<:time shift>,pattern)

Gibt ein Array mit den Ergebnissen der JSONPath-Abfrage zurück.
 Unterstützte Werttypen: *String*, *Text*, *Log*.

Parameter:

- Siehe allgemeine Parameter
- **pattern** – das erforderliche Muster (String-Argumente müssen in doppelte Anführungszeichen eingeschlossen werden)

Kommentare:

- Diese Funktion wird mit Aggregatfunktionen verwendet.
- Eine JSONPath-Abfrage wird für jeden ausgewählten Verlaufswert ausgeführt; jedes nicht leere, erfolgreiche Abfrageergebnis wird dem Array hinzugefügt. Wenn ein Abfrageergebnis für einen einzelnen Verlaufswert mehrere Treffer enthält, wird jeder Treffer separat dem Array hinzugefügt.
- Die Funktion schlägt mit einem Fehler fehl, wenn:
  - Das Format von `pattern` ungültig ist.
  - Einer der Verlaufswerte ungültiges JSON enthält oder die JSONPath-Abfrage für diesen Wert fehlschlägt.
- Die Funktion gibt ein leeres Array zurück, wenn:
  - Keine Verlaufswerte ausgewählt wurden.
  - Alle ausgewählten Verlaufswerte ungültiges JSON enthalten.

Beispiel:

```
max(jsonpath(/host/http.response,5m,"$.response_time")) > 500
```

last(/host/key,<#num<:time shift»)

Der zuletzt empfangene Wert.
 Unterstützte Werttypen: *Float*, *Integer*, *String*, *Text*, *Log*.

Parameter:

- Siehe allgemeine Parameter

- **#num** (optional) - der n-te zuletzt empfangene Wert

Kommentare:

- Beachten Sie, dass ein mit Hash gekennzeichneter Zeitraum (#N) hier anders funktioniert als bei vielen anderen Funktionen. Zum Beispiel: `last(/host/key)` ist immer gleich `last(/host/key,#1)`; `last(/host/key,#3)` - der drittletzte Wert (*nicht* die letzten drei Werte).
- Zabbix garantiert nicht die genaue Reihenfolge der Werte, wenn innerhalb einer Sekunde mehr als zwei Werte in der Historie vorhanden sind.
- Siehe auch first().

Beispiele:

```
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»)

Der Zeitstempel des N-tjüngsten Werts innerhalb des definierten Auswertungszeitraums.
 Unterstützte Werttypen: *Float*, *Integer*, *String*, *Text*, *Log*.

Parameter:

- Siehe allgemeine Parameter;
- **#num** (optional) - der N-tjüngste Wert.

Die Funktion schlägt mit einem Fehler fehl, wenn im angegebenen Zeitraum keine Daten erfasst wurden oder kein N-ter Wert erfasst wurde.

Siehe auch firstclock().

Beispiele:

```
lastclock(/host/key) #Zeitstempel des neuesten Werts abrufen
lastclock(/host/key,#2) #Zeitstempel des zweitneuesten Werts abrufen
now()-lastclock(/host/heartbeat.item)>300 #mehr als 5 Minuten sind vergangen, seit der heartbeat-Datenpunk
lastclock(/host/system.cpu.load[all,avg1])-lastclock(/host/system.cpu.load[all,avg1],#2)>300 #das Aktualis
((now()-lastclock(/host/system.cpu.load[all,avg1])<120 and last(/host/system.cpu.load[all,avg1])>5)) #CPU-
```

logeventid(/host/key,<#num<:time shift»,<pattern>)

Prüft, ob die Ereignis-ID des letzten Logeintrags mit einem regulären Ausdruck übereinstimmt.
 Unterstützte Werttypen: *Log*.
 Rückgabewert: 0 - stimmt nicht überein; 1 - stimmt überein.

Parameter:

- Siehe allgemeine Parameter
- **#num** (optional) - der N-te zuletzt empfangene Wert
- **pattern** (optional) - der reguläre Ausdruck, der das erforderliche Muster beschreibt, im Stil von Perl Compatible Regular Expression (PCRE) (Zeichenfolgenargumente müssen in doppelte Anführungszeichen gesetzt werden)

Beispiele:

```
logeventid(/host/eventlog[Security],,"^4625$")=1 #ein Logeintrag mit der ID "4625" (fehlgeschlagene Authen
logeventid(/host/eventlog[System],,"^(6008|41)$")=1 #ein Logeintrag mit der ID "6008" oder "41" wurde gefu
```

logseverity(/host/key,<#num<:time shift»)

Protokollschwere des letzten Protokolleintrags.
 Unterstützte Werttypen: *Log*.
 Rückgabewerte: 0 - Standardschweregrad; N - Schweregrad (integer, nützlich für Windows-Ereignisprotokolle: 1 - Information, 2 - Warnung, 4 - Fehler, 7 - Überwachungsfehler, 8 - Überwachungserfolg, 9 - Kritisch, 10 - Ausführlich).

Parameter:

- Siehe allgemeine Parameter
- **#num** (optional) - der N-tjüngste Wert

Zabbix übernimmt den Protokollschweregrad aus dem Feld **Information** des Windows-Ereignisprotokolls.

Beispiele:

```
logseverity(/host/log[/var/log/syslog],10m)>3 #a log entry with severity above "3" found
logseverity(/host/eventlog[System],10m)=4 #a log entry with severity equaling "Error" found
```

logsource(/host/key,<#num<:time shift»,<pattern>)

Prüft, ob die Logquelle des letzten Logeintrags mit einem regulären Ausdruck übereinstimmt.
 Unterstützte Werttypen: *Log*.
 Rückgabewert: 0 - stimmt nicht überein; 1 - stimmt überein.

Parameter:

- Siehe allgemeine Parameter
- **#num** (optional) - der N-te zuletzt empfangene Wert
- **pattern** (optional) - der reguläre Ausdruck, der das erforderliche Muster beschreibt, im Stil von Perl Compatible Regular Expression (PCRE) (Zeichenkettenargumente müssen in doppelte Anführungszeichen gesetzt werden)

Wird normalerweise für Windows-Ereignisprotokolle verwendet.

Beispiele:

```
logsource(/host/eventlog[Application],,"MSSQLSERVER")=1 #ein Logeintrag mit einer Quelle, die mit "MSSQLSE
logsource(/host/eventlog[System],,"Service Control Manager")=1 #ein Logeintrag mit einer Quelle, die mit "
logsource(/host/eventlog[System],,"Service Control Manager")=1 and logeventid(/host/eventlog[System],,"^70
```

logtimestamp(/host/key,<#num<:time shift»)

Der Zeitstempel der Log-Nachricht des N-tjüngsten Werts eines Log-Datenpunkts.
 Unterstützte Wertetypen: *Log*.

Parameter:

- Siehe allgemeine Parameter;
- **#num** (optional) - der N-tjüngste Wert.

Kommentare:

- Die Berechnung der Zeitverschiebung basiert auf der Uhrzeit des Datenpunktwerts, nicht auf dem Zeitstempel der Log-Nachricht;
- Die Funktion schlägt mit einem Fehler fehl, wenn:
  - ein Datenpunkt eines Nicht-Log-Typs empfangen wird;
  - im angegebenen Zeitraum keine Daten erfasst wurden;
  - kein N-ter Wert erfasst wurde;
  - die Log-Nachricht keinen Zeitstempelwert enthält.

Beispiele:

```
logtimestamp(/host/key) #Zeitstempel der neuesten Log-Nachricht abrufen
logtimestamp(/host/key,#2) #Zeitstempel der zweitneuesten Log-Nachricht abrufen
logtimestamp(/host/key,#3:now-1d) #Zeitstempel der drittneuesten Log-Nachricht von vor einem Tag abrufen
```

monodec(/host/key,(sec|#num)<:time shift>,<mode>)

Prüft, ob es einen monotonen Rückgang von Werten gegeben hat.
 Unterstützte Wertetypen: *Integer*.
 Rückgabewert: 1 - wenn alle Elemente im Zeitraum fortlaufend abnehmen; 0 - andernfalls.

Parameter:

- Siehe allgemeine Parameter
- **mode** (muss in doppelte Anführungszeichen gesetzt werden) - *weak* (jeder Wert ist kleiner oder gleich dem vorherigen; Standard) oder *strict* (jeder Wert hat abgenommen)

Beispiele:

```
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 #the number of nginx processes has monotonously decreased ov
```

monoinc(/host/key,(sec|#num)<:time shift>,<mode>)

Prüft, ob es einen monotonen Anstieg der Werte gegeben hat.
 Unterstützte Werttypen: *Integer*.
 Rückgabewert: 1 - wenn alle Elemente im Zeitraum kontinuierlich ansteigen; 0 - andernfalls.

Parameter:

- Siehe allgemeine Parameter
- **mode** (muss in doppelte Anführungszeichen gesetzt werden) - *weak* (jeder Wert ist größer als oder gleich dem vorherigen; Standard) oder *strict* (jeder Wert ist gestiegen)

Beispiele:

```
monoinc(/host/system.localtime,#3,"strict")=0 #die lokale Systemzeit ist nicht durchgängig angestiegen
monoinc(/host/vfs.dir.count[/mnt/data/logs],24h,"weak")=0 #Auslöser, wenn die Dateianzahl über 24 Stunden
```

nodata(/host/key,sec,<mode>)

Prüft, ob keine Daten empfangen wurden.
 Unterstützte Werttypen: *Integer*, *Float*, *Character*, *Text*, *Log*.
 Rückgabewert: 1 - wenn während des definierten Zeitraums keine Daten empfangen wurden; 0 - andernfalls.

Parameter:

- Siehe allgemeine Parameter
- **sec** - der Zeitraum darf nicht weniger als 30 Sekunden betragen, da der History-Synchronisierungsprozess diese Funktion nur alle 30 Sekunden berechnet; `nodata(/host/key,0)` ist nicht zulässig
- **mode** - wenn auf *strict* (in doppelten Anführungszeichen) gesetzt, berücksichtigt diese Funktion die Verfügbarkeit des Proxys nicht (Details siehe Kommentare)

Kommentare:

- Die von einem Proxy überwachten „nodata“-Auslöser berücksichtigen standardmäßig die Verfügbarkeit des Proxys. Wenn der Proxy nicht verfügbar ist, werden die „nodata“-Auslöser nicht unmittelbar nach der Wiederherstellung der Verbindung ausgelöst, sondern überspringen die Daten für den Zeitraum der Verzögerung. Beachten Sie, dass bei passiven Proxys die Unterdrückung aktiviert wird, wenn die Verbindung nach mehr als 15 Sekunden und frühestens 2 Sekunden später wiederhergestellt wird. Bei aktiven Proxys wird die Unterdrückung aktiviert, wenn die Verbindung nach mehr als 15 Sekunden wiederhergestellt wird. Um die Berücksichtigung der Proxy-Verfügbarkeit zu deaktivieren, verwenden Sie den dritten Parameter, z. B.: `nodata(/host/key,5m,"strict")`; in diesem Fall wird die Funktion ausgelöst, sobald der Auswertungszeitraum (fünf Minuten) ohne Daten abgelaufen ist.
- Diese Funktion zeigt einen Fehler an, wenn innerhalb des Zeitraums des 1. Parameters:
- keine Daten vorhanden sind und der Zabbix-Server neu gestartet wurde
- keine Daten vorhanden sind und die Wartung beendet wurde
- keine Daten vorhanden sind und der Datenpunkt hinzugefügt oder wieder aktiviert wurde.
- Fehler werden in der Spalte *Info* der Auslöserliste angezeigt.
- Diese Funktion funktioniert möglicherweise nicht ordnungsgemäß, wenn zwischen Zabbix-Server, Proxy und Agent Zeitunterschiede bestehen. Siehe auch: Anforderung an die Zeitsynchronisierung.

Beispiel:

```
nodata(/host/agent.ping,5m)=1 #Auslöser, wenn fünf Minuten lang keine Daten vom Zabbix-Agent empfangen wur
```

percentile(/host/key,(sec|#num)<:time shift>,percentage)

Das P-te Perzentil eines Zeitraums, wobei P (percentage) durch den dritten Parameter angegeben wird.
 Unterstützte Wertetypen: *Float*, *Integer*.

Parameter:

- Siehe allgemeine Parameter
- **percentage** - eine Gleitkommazahl zwischen 0 und 100 (einschließlich) mit bis zu 4 Ziffern nach dem Dezimalpunkt

Beispiele:

```
percentile(/host/net.if.in[eth0,bytes],1h,95)>1000000 #95. Perzentil des Netzwerkeingangs (Bytes/Sek.) übe
percentile(/host/system.cpu.util,5m,95)>80 #95. Perzentil des CPU-Auslastungsprozentsatzes der User-Zeit i
percentile(/host/icmppingsec[192.168.0.2],15m,95)>0.15 #die meisten Latenzmessungen liegen unter 150 ms, a
percentile(/host/net.if.in[eth0,bytes],1h,50) #berechnet das 50. Perzentil (den Medianwert) des eingehende
(percentile(/host/net.if.in[eth0,bytes],1h,50)+percentile(/host/net.if.in[eth0,bytes],1h,51))/2 #berechnet
```

rate(/host/key,sec<:time shift>)

Der durchschnittliche Anstieg pro Sekunde in einem monoton steigenden Zähler innerhalb des definierten Zeitraums.
 Unterstützte Wertetypen: *Float*, *Integer*.

Parameter:

- Siehe allgemeine Parameter

Entspricht funktional ’rate’ aus PromQL.

Beispiele:

```
rate(/host/key,30s) #wenn der monotone Anstieg über 30 Sekunden 20 beträgt, gibt diese Funktion 0,67 zurüc
rate(/host/net.if.in[eth0,bytes],5m)>500000 #die eingehende Schnittstellenverkehrsrate auf eth0 hat in den
rate(/host/app.requests.count,1m)>100 #der Zähler für die Anzahl der Anfragen ist in der letzten Minute au
```

xmlxpath(/host/key,(sec|#num)<:time shift>,pattern)

Gibt ein Array mit den Ergebnissen der XML-XPath-Abfrage zurück.
 Unterstützte Werttypen: *String*, *Text*, *Log*.

Parameter:

- Siehe allgemeine Parameter
- **pattern** – das erforderliche Muster (String-Argumente müssen in doppelte Anführungszeichen eingeschlossen werden)

Kommentare:

- Diese Funktion wird mit Aggregatfunktionen verwendet.
- Eine XML-XPath-Abfrage wird für jeden ausgewählten Verlaufswert ausgeführt; jedes nicht leere, erfolgreiche Abfrageergebnis wird dem Array hinzugefügt. Wenn ein Abfrageergebnis für einen einzelnen Verlaufswert mehrere Treffer enthält, wird jeder Treffer separat dem Array hinzugefügt.
- Die Funktion schlägt mit einem Fehler fehl, wenn:
  - das Format von `pattern` ungültig ist.
  - Zabbix ohne XML-Unterstützung kompiliert wurde.
  - Einer der Verlaufswerte ungültiges XML enthält oder die XPath-Abfrage für diesen Wert fehlschlägt.
- Die Funktion gibt ein leeres Array zurück, wenn:
  - keine Verlaufswerte ausgewählt wurden.
  - alle ausgewählten Verlaufswerte ungültiges XML enthalten.

Beispiel:

```
max(xmlxpath(/host/http.response,5m,"/response/response_time")) > 500
```

Siehe alle unterstützten Funktionen.

Trendfunktionen

Trendfunktionen verwenden im Gegensatz zu Historienfunktionen Trend-Daten für Berechnungen.

Trends speichern stündliche aggregierte Werte. Trendfunktionen verwenden diese stündlichen Durchschnittswerte und sind daher für Langzeitanalysen nützlich.

Die Ergebnisse von Trendfunktionen werden zwischengespeichert, sodass mehrere Aufrufe derselben Funktion mit denselben Parametern Informationen aus der Datenbank nur einmal abrufen. Der Cache der Trendfunktionen wird über den Serverparameter TrendFunctionCacheSize gesteuert.

Verwenden Sie `zabbix[tcache,cache,paccessed]`, um abzuschätzen, wie viel des Caches tatsächlich verwendet wird. Ein niedriger Wert für `paccessed` zusammen mit niedrigen Werten für `pitems` und `pmisses` bedeutet in der Regel, dass die Cache-Größe reduziert werden kann.

Auslöser, die **nur** Trendfunktionen referenzieren, werden einmal pro kleinstem Zeitintervall im Ausdruck ausgewertet. Beispielsweise wird ein Auslöser wie

```
trendavg(/host/key,1d:now/d) > 1 or trendavg(/host/key2,1w:now/w) > 2
```

einmal pro Tag ausgewertet. Wenn der Auslöser sowohl Trend- als auch Historienfunktionen (oder Datum und Uhrzeit und/oder nodata()) enthält, wird er gemäß den üblichen Prinzipien berechnet.

Alle hier aufgeführten Funktionen werden unterstützt in:

- Ausdruck von Auslösern
- Berechnete Datenpunkte
- Ausdrucksmakros

Die Funktionen werden ohne zusätzliche Informationen aufgeführt. Klicken Sie auf die Funktion, um die vollständigen Details anzuzeigen.

Function	Description
baselinedev	Gibt die Anzahl der Abweichungen zurück (nach dem stddevpop-Algorithmus) zwischen dem letzten Datenzeitraum und denselben Datenzeiträumen in vorherigen Saisons.
baselinewma	Berechnet die Basislinie, indem Daten aus demselben Zeitrahmen in mehreren gleich langen Zeiträumen (”Saisons”) mithilfe des gewichteten gleitenden Durchschnittsalgorithmus gemittelt werden.
trendavg	Der Durchschnitt der Trendwerte innerhalb des definierten Zeitraums.
trendcount	Die Anzahl der erfolgreich abgerufenen Historienwerte, die zur Berechnung des Trendwerts innerhalb des definierten Zeitraums verwendet wurden.

Function	Description
trendmax	Der Maximalwert der Trendwerte innerhalb des definierten Zeitraums.
trendmin	Der Minimalwert der Trendwerte innerhalb des definierten Zeitraums.
trendstl	Gibt die Anomalierate während des Erkennungszeitraums zurück - ein Dezimalwert zwischen 0 und 1, der `((die Anzahl der Anomaliewerte)/(Gesamtzahl der Werte))` entspricht.
trendsum	Die Summe der Trendwerte innerhalb des definierten Zeitraums.

Allgemeine Parameter

- `/host/key` ist ein allgemeiner obligatorischer erster Parameter
- `time period:time shift` ist ein allgemeiner zweiter Parameter, wobei:
  - **time period** - der Zeitraum (mindestens „1h“), definiert als <N><Zeiteinheit>, wobei `N` - die Anzahl der Zeiteinheiten ist, `time unit` - h (Stunde), d (Tag), w (Woche), M (Monat) oder y (Jahr).
  - **time shift** - der Versatz des Zeitraums (siehe Funktionsbeispiele)

Funktionsdetails

Einige allgemeine Hinweise zu Funktionsparametern:

- Funktionsparameter werden durch ein Komma getrennt
- Optionale Funktionsparameter (oder Parameterteile) werden durch < > angegeben
- Funktionsspezifische Parameter werden bei jeder Funktion beschrieben
- Die Parameter `/host/key` und `time period:time shift` dürfen niemals in Anführungszeichen gesetzt werden

baselinedev(/host/key,data period:time shift,season unit,num seasons)

Gibt die Anzahl der Abweichungen (nach dem Algorithmus stddevpop) zwischen dem letzten Datenzeitraum und denselben Datenzeiträumen in vorhergehenden Saisons zurück.

Parameter:

- Siehe gemeinsame Parameter
- **data period** - der Datenerfassungszeitraum innerhalb einer Saison, definiert als <N><time unit>, wobei:
`N` - die Anzahl der Zeiteinheiten
`time unit` - h (Stunde), d (Tag), w (Woche), M (Monat) oder y (Jahr); muss gleich oder kleiner als die Saison sein
- **season unit** - die Kalendereinheit, die eine Saison definiert (h, d, w, M, y); darf nicht kleiner als **data period** sein
- **num seasons** - die Anzahl der auszuwertenden Saisons

Beispiele:

```
baselinedev(/host/key,1d:now/d,"M",6) #Berechnung der Anzahl der Standardabweichungen (Population) zwische
baselinedev(/host/key,1h:now/h,"d",10)  #Berechnung der Anzahl der Populations-Standardabweichungen zwisch
```

baselinewma(/host/key,data period:time shift,season unit,num seasons)

Berechnet die Baseline durch Mittelung von Daten aus demselben Zeitrahmen in mehreren gleich langen Zeitperioden („Saisons“) unter Verwendung des Algorithmus des gewichteten gleitenden Durchschnitts.

Parameter:

- Siehe allgemeine Parameter
- **Datenperiode** - der Datenerfassungszeitraum innerhalb einer Saison, definiert als <N><Zeiteinheit>, wobei:
`N` - die Anzahl der Zeiteinheiten
`Zeiteinheit` - h (Stunde), d (Tag), w (Woche), M (Monat) oder y (Jahr); muss gleich oder kleiner als die Saison sein
Zeitverschiebung - der Offset der Zeitperiode; definiert das Ende des Datenerfassungszeitraums in Saisons (siehe Beispiele)
- **Saisoneinheit** - die Kalendereinheit, die eine Saison definiert (h, d, w, M, y); darf nicht kleiner als die Datenperiode sein
- **Anzahl Saisons** - die Anzahl der auszuwertenden Saisons

Beispiele:

```
baselinewma(/host/key,1h:now/h,"d",3) #Berechnung der Baseline aus derselben Tagesstunde über die letzten
baselinewma(/host/key,2h:now/h,"d",3) #Berechnung der Baseline aus demselben Zwei-Stunden-Zeitrahmen über
baselinewma(/host/key,1d:now/d,"M",4) #Berechnung der Baseline aus demselben Kalendertag wie „gestern“ übe
```

trendavg(/host/key,time period:time shift)

Der Durchschnitt der Trendwerte innerhalb des definierten Zeitraums.

Parameter:

- Siehe allgemeine Parameter

Beispiele:

```
trendavg(/host/key,1h:now/h) #der Durchschnitt für die vorherige Stunde (z. B. 12:00-13:00)
trendavg(/host/key,1h:now/h-1h) #der Durchschnitt für vor zwei Stunden (11:00-12:00)
trendavg(/host/key,1h:now/h-2h) #der Durchschnitt für vor drei Stunden (10:00-11:00)
trendavg(/host/key,1M:now/M-1y) #der Durchschnitt für den vorherigen Monat vor einem Jahr
```

trendcount(/host/key,time period:time shift)

Die Anzahl der erfolgreich abgerufenen Verlaufswerte, die zur Berechnung des Trendwerts innerhalb des definierten Zeitraums verwendet werden.

Parameter:

- Siehe allgemeine Parameter

Beispiele:

```
trendcount(/host/key,1h:now/h) #die Anzahl der Werte für die vorherige Stunde (z. B. 12:00-13:00)
trendcount(/host/key,1h:now/h-1h) #die Anzahl der Werte für vor zwei Stunden (11:00-12:00)
trendcount(/host/key,1h:now/h-2h) #die Anzahl der Werte für vor drei Stunden (10:00-11:00)
trendcount(/host/key,1M:now/M-1y) #die Anzahl der Werte für den vorherigen Monat vor einem Jahr
```

trendmax(/host/key,time period:time shift)

Das Maximum der Trendwerte innerhalb des definierten Zeitraums.

Parameter:

- Siehe allgemeine Parameter

Beispiele:

```
trendmax(/host/key,1h:now/h) #das Maximum für die vorherige Stunde (z. B. 12:00-13:00)
trendmax(/host/key,1h:now/h) - trendmin(/host/key,1h:now/h) → berechnet die Differenz zwischen dem Maximum
trendmax(/host/key,1h:now/h-1h) #das Maximum für vor zwei Stunden (11:00-12:00)
trendmax(/host/key,1h:now/h-2h) #das Maximum für vor drei Stunden (10:00-11:00)
trendmax(/host/key,1M:now/M-1y) #das Maximum für den vorherigen Monat vor einem Jahr
```

trendmin(/host/key,time period:time shift)

Das Minimum der Trendwerte innerhalb des definierten Zeitraums.

Parameter:

- Siehe gemeinsame Parameter

Beispiele:

```
trendmin(/host/key,1h:now/h) #das Minimum für die vorherige Stunde (z. B. 12:00-13:00)
trendmax(/host/key,1h:now/h) - trendmin(/host/key,1h:now/h) → berechnet die Differenz zwischen den Maximum
trendmin(/host/key,1h:now/h-1h) #das Minimum für vor zwei Stunden (11:00-12:00)
trendmin(/host/key,1h:now/h-2h) #das Minimum für vor drei Stunden (10:00-11:00)
trendmin(/host/key,1M:now/M-1y) #das Minimum für den vorherigen Monat vor einem Jahr
```

trendstl(/host/key,eval period:time shift,detection period,season,<deviations>,<devalg>,<s window>)

Gibt die Rate der Anomalien während des Erkennungszeitraums zurück – einen Dezimalwert zwischen 0 und 1, der `((Anzahl der Anomaliewerte)/(Gesamtzahl der Werte))` entspricht.

Parameter:

- Siehe allgemeine Parameter
- **eval period** - der Zeitraum, der zerlegt werden muss (Minimum „1h“), definiert als <N><time unit>, wobei
`N` - die Anzahl der Zeiteinheiten
`time unit` - h (Stunde), d (Tag), w (Woche), M (Monat) oder y (Jahr)
- **detection period** - der Zeitraum vor dem Ende von eval period, für den Anomalien berechnet werden (Minimum „1h“, darf nicht länger als eval period sein), definiert als <N><time unit>, wobei
`N` - die Anzahl der Zeiteinheiten
`time unit` - h (Stunde), d (Tag), w (Woche)
- **season** - der kürzeste Zeitraum, in dem ein sich wiederholendes Muster („Saison“) erwartet wird (Minimum „2h“, darf nicht länger als eval period sein, die Anzahl der Einträge in eval period muss größer sein als das Doppelte der resultierenden Frequenz (season/h)), definiert als <N><time unit>, wobei
`N` - die Anzahl der Zeiteinheiten
`time unit` - h (Stunde), d (Tag), w (Woche)

- **deviations** - die Anzahl der Abweichungen (berechnet durch devalg), die als Anomalie gezählt werden (kann dezimal sein), (muss größer oder gleich 1 sein, Standardwert ist 3)
- **devalg** (muss in doppelte Anführungszeichen gesetzt werden) - der Abweichungsalgorithmus; kann *stddevpop*, *stddevsamp* oder *mad* (Standard) sein
- **s window** - die Spannweite (in Lags) des loess-Fensters für die saisonale Extraktion (Standard ist 10 * Anzahl der Einträge in eval period + 1)

Beispiele:

```
trendstl(/host/key,100h:now/h,10h,2h) #analysiert die letzten 100 Stunden der Trenddaten, ermittelt die An
trendstl(/host/key,100h:now/h-10h,100h,2h,2.1,"mad") #analysiert einen Zeitraum von 100 Stunden Trenddaten
trendstl(/host/key,100d:now/d-1d,10d,1d,4,,10) #analysiert 100 Tage Trenddaten bis vor einem Tag, ermittel
trendstl(/host/key,1M:now/M-1y,1d,2h,,"stddevsamp") #analysiert den vorherigen Monat vor einem Jahr, ermit
```

trendsum(/host/key,time period:time shift)

Die Summe der Trendwerte innerhalb des definierten Zeitraums.

Parameter:

- Siehe allgemeine Parameter

Beispiele:

```
trendsum(/host/key,1h:now/h) #die Summe für die vorherige Stunde (z. B. 12:00-13:00)
trendsum(/host/key,1h:now/h-1h) #die Summe für vor zwei Stunden (11:00-12:00)
trendsum(/host/key,1h:now/h-2h) #die Summe für vor drei Stunden (10:00-11:00)
trendsum(/host/key,1M:now/M-1y) #die Summe für den vorherigen Monat vor einem Jahr
```

Siehe alle unterstützten Funktionen.

Mathematische Funktionen

Alle hier aufgeführten Funktionen werden unterstützt in:

- Auslöserausdrücken
- Formeln für berechnete Datenpunkte
- Ausdrucksmakros

Mathematische Funktionen werden für die Werttypen float und integer unterstützt, sofern nicht anders angegeben.

Die Funktionen werden ohne zusätzliche Informationen aufgelistet. Klicken Sie auf die Funktion, um die vollständigen Details anzuzeigen.

Function	Description
abs	Der absolute Wert eines Werts.
acos	Der Arkuskosinus eines Werts als Winkel, ausgedrückt in Radiant.
asin	Der Arkussinus eines Werts als Winkel, ausgedrückt in Radiant.
atan	Der Arkustangens eines Werts als Winkel, ausgedrückt in Radiant.
atan2	Der Arkustangens der Koordinaten Ordinate (Wert) und Abszisse, angegeben als Winkel, ausgedrückt in Radiant.
avg	Der Durchschnittswert der referenzierten Datenpunktwerte.
cbrt	Die Kubikwurzel eines Werts.
ceil	Rundet den Wert auf die nächstgrößere oder gleich große ganze Zahl auf.
cos	Der Kosinus eines Werts, wobei der Wert ein als Radiant ausgedrückter Winkel ist.
cosh	Der hyperbolische Kosinus eines Werts.
cot	Der Kotangens eines Werts, wobei der Wert ein als Radiant ausgedrückter Winkel ist.
degrees	Konvertiert einen Wert von Radiant in Grad.
e	Die Eulersche Zahl (2.718281828459045).
exp	Die Eulersche Zahl potenziert mit einem Wert.
expm1	Die Eulersche Zahl potenziert mit einem Wert minus 1.
floor	Rundet den Wert auf die nächstkleinere oder gleich große ganze Zahl ab.
log	Der natürliche Logarithmus.
log10	Der dekadische Logarithmus.
max	Der höchste Wert der referenzierten Datenpunktwerte.
min	Der niedrigste Wert der referenzierten Datenpunktwerte.
mod	Der Divisionsrest.

Function	Description
pi	Die Pi-Konstante (3.14159265358979).
power	Die Potenz eines Werts.
radians	Konvertiert einen Wert von Grad in Radiant.
rand	Gibt einen zufälligen Ganzzahlwert zurück.
round	Rundet den Wert auf Dezimalstellen.
signum	Gibt '-1' zurück, wenn ein Wert negativ ist, '0' wenn ein Wert null ist, '1' wenn ein Wert positiv ist.
sin	Der Sinus eines Werts, wobei der Wert ein als Radiant ausgedrückter Winkel ist.
sinh	Der hyperbolische Sinus eines Werts, wobei der Wert ein als Radiant ausgedrückter Winkel ist.
sqrt	Die Quadratwurzel eines Werts.
sum	Die Summe der referenzierten Datenpunktwerte.
tan	Der Tangens eines Werts.
truncate	Schneidet den Wert auf Dezimalstellen ab.

Funktionsdetails

Einige allgemeine Hinweise zu Funktionsparametern:

- Funktionsparameter werden durch ein Komma getrennt
- Ausdrücke werden als Parameter akzeptiert
- Optionale Funktionsparameter (oder Parameterteile) werden durch < > gekennzeichnet

abs(value)

Der Absolutwert (ab 0) eines Wertes.

Parameter:

- **value** - der zu prüfende Wert

Zum Beispiel ist der Absolutwert von sowohl „3“ als auch „-3“ gleich „3“.

Beispiel:

```
abs(last(/host/key))>10
```

acos(value)

Der Arkuskosinus eines Werts als Winkel, ausgedrückt in Radiant.

Parameter:

- **value** - der zu prüfende Wert

Der Wert muss zwischen -1 und 1 liegen. Zum Beispiel ist der Arkuskosinus des Werts „0.5“ „2.0943951“.

Beispiel:

```
acos(last(/host/key))
```

asin(value)

Der Arkussinus eines Werts als Winkel, ausgedrückt in Radiant.

Parameter:

- **value** - der zu prüfende Wert

Der Wert muss zwischen -1 und 1 liegen. Zum Beispiel ist der Arkussinus des Werts „0.5“ „-0.523598776“.

Beispiel:

```
asin(last(/host/key))
```

atan(value)

Der Arkustangens eines Werts als Winkel, ausgedrückt in Radiant.

Parameter:

- **value** - der zu prüfende Wert

Zum Beispiel ist der Arkustangens des Werts „1“ gleich „0.785398163“.

Beispiel:

```
atan(last(/host/key))
```

atan2(value,abscissa)

Der Arkustangens der Ordinaten- (value) und Abszissenkoordinaten, angegeben als Winkel und ausgedrückt in Radiant.

Parameter:

- **value** - der zu prüfende Wert;
- **abscissa** - der Abszissenwert.

Zum Beispiel ist der Arkustangens der Ordinaten- und Abszissenkoordinaten eines Werts '1' gleich '2.21429744'.

Beispiel:

```
atan2(last(/host/key),2)
```

avg(<value1>,<value2>,...)

Der Durchschnittswert der referenzierten Datenpunktwerte.

Parameter:

- **valueX** - der von einer anderen Funktion zurückgegebene Wert, die mit der Datenpunkthistorie arbeitet.

Beispiel:

```
avg(avg(/host/key,1h),avg(/host2/key2,1h))
```

cbrt(value)

Die Kubikwurzel eines Wertes.

Parameter:

- **value** - der zu prüfende Wert

Zum Beispiel ist die Kubikwurzel von „64" „4", von „63" „3,97905721".

Beispiel:

```
cbrt(last(/host/key))
```

ceil(value)

Rundet den Wert auf die nächstgrößere oder gleiche ganze Zahl auf.

Parameter:

- **value** - der zu prüfende Wert

Zum Beispiel wird „2.4" auf „3" aufgerundet. Siehe auch floor().

Beispiel:

```
ceil(last(/host/key))
```

cos(value)

Der Kosinus eines Werts, wobei der Wert ein Winkel ist, der in Radiant ausgedrückt wird.

Parameter:

- **value** - der zu prüfende Wert

Zum Beispiel ist der Kosinus des Werts „1" gleich „0.54030230586".

Beispiel:

```
cos(last(/host/key))
```

cosh(value)

Der hyperbolische Kosinus eines Werts. Gibt den Wert als reelle Zahl und nicht in wissenschaftlicher Notation zurück.

Parameter:

- **value** - der zu prüfende Wert

Zum Beispiel ist der hyperbolische Kosinus des Werts „1" gleich „1.54308063482".

Beispiel:

```
cosh(last(/host/key))
```

cot(value)

Der Kotangens eines Werts, wobei der Wert ein in Radiant ausgedrückter Winkel ist.

Parameter:

- **value** - der zu prüfende Wert

Zum Beispiel ist der Kotangens eines Werts ’1’ gleich ’0.54030230586’.

Beispiel:

```
cot(last(/host/key))
```

degrees(value)

Konvertiert einen Wert von Radiant in Grad.

Parameter:

- **value** - der zu prüfende Wert

Beispielsweise wird ein Wert ’1’ bei der Umrechnung in Grad zu ’57.2957795’.

Beispiel:

```
degrees(last(/host/key))
```

e

Die Eulersche Zahl (2.718281828459045).

Beispiel:

```
e()
```

exp(value)

Die Eulersche Zahl hoch dem Wert eines Wertes.

Parameter:

- **value** - der zu prüfende Wert

Zum Beispiel ergibt die Eulersche Zahl hoch dem Wert „2“ den Wert „7.38905609893065“.

Beispiel:

```
exp(last(/host/key))
```

expm1(value)

Die Eulersche Zahl hoch dem Wert minus 1.

Parameter:

- **value** - der zu prüfende Wert

Zum Beispiel ergibt die Eulersche Zahl hoch dem Wert „2“ minus 1 den Wert „6.38905609893065“.

Beispiel:

```
expm1(last(/host/key))
```

floor(value)

Rundet den Wert auf die nächstkleinere oder gleiche Ganzzahl ab.

Parameter:

- **value** – der zu prüfende Wert

Zum Beispiel wird „2.6“ auf „2“ abgerundet. Siehe auch ceil().

Beispiel:

```
floor(last(/host/key))
```

log(value)

Der natürliche Logarithmus.

Parameter:

- **value** - der zu prüfende Wert

Zum Beispiel ist der natürliche Logarithmus des Werts „2“ gleich „0.69314718055994529“.

Beispiel:

```
log(last(/host/key))
```

log10(value)

Der dekadische Logarithmus.

Parameter:

- **value** - der zu prüfende Wert

Zum Beispiel ist der dekadische Logarithmus des Werts „5“ gleich „0.69897000433“.

Beispiel:

```
log10(last(/host/key))
```

max(<value1>,<value2>,...)

Der höchste Wert der referenzierten Datenpunkt-Werte.

Parameter:

- **valueX** - der von einer anderen Funktion zurückgegebene Wert, die mit der Datenpunkt-Historie arbeitet.

Beispiel:

```
max(avg(/host/key,1h),avg(/host2/key2,1h))
```

min(<value1>,<value2>,...)

Der niedrigste Wert der referenzierten Datenpunkt-Werte.

Parameter:

- **valueX** - der von einer anderen Funktion zurückgegebene Wert, die mit der Datenpunkt-Historie arbeitet.

Beispiel:

```
min(avg(/host/key,1h),avg(/host2/key2,1h))
```

mod(value,denominator)

Der Divisionsrest.

Parameter:

- **value** - der zu prüfende Wert;
- **denominator** - der Divisor.

Zum Beispiel ist der Divisionsrest eines Werts „5“ bei einem Divisor „2“ gleich „1“.

Beispiel:

```
mod(last(/host/key),2)
```

pi

Die Pi-Konstante (3.14159265358979).

Beispiel:

```
pi()
```

power(Wert, Potenzwert)

Die Potenz eines Werts.

Parameter:

- **Wert** – der zu prüfende Wert;
- **Potenzwert** – die zu verwendende n-te Potenz.

Zum Beispiel ergibt die 3. Potenz des Werts „2“ den Wert „8“.

Beispiel:

```
power(last(/host/key),3)
```

radians(value)

Konvertiert einen Wert von Grad in Bogenmaß.

Parameter:

- **value** - der zu prüfende Wert

Beispielsweise wird ein Wert ’1’ in Bogenmaß zu ’0.0174532925’ konvertiert.

Beispiel:

```
radians(last(/host/key))
```

rand

Gibt einen zufälligen Ganzzahlwert zurück. Eine pseudozufällig erzeugte Zahl, die die Zeit als Seed verwendet (ausreichend für mathematische Zwecke, aber nicht für Kryptografie).

Beispiel:

```
rand()
```

round(value,decimal places)

Rundet den Wert auf Dezimalstellen.

Parameter:

- **value** - der zu prüfende Wert;
- **decimal places** - gibt die Dezimalstellen für die Rundung an (0 ist ebenfalls möglich).

Beispielsweise wird ein Wert „2.5482“, auf 2 Dezimalstellen gerundet, zu „2.55“.

Beispiel:

```
round(last(/host/key),2)
```

signum(value)

Gibt ’-1’ zurück, wenn ein Wert negativ ist, ’0’, wenn ein Wert null ist, und ’1’, wenn ein Wert positiv ist.

Parameter:

- **value** - der zu prüfende Wert.

Beispiel:

```
signum(last(/host/key))
```

sin(value)

Der Sinus eines Werts, wobei der Wert ein in Radiant ausgedrückter Winkel ist.

Parameter:

- **value** - der zu prüfende Wert

Zum Beispiel ist der Sinus des Werts „1“ gleich „0.8414709848“.

Beispiel:

```
sin(last(/host/key))
```

sinh(value)

Der hyperbolische Sinus eines Wertes.

Parameter:

- **value** - der zu prüfende Wert

Zum Beispiel ist der hyperbolische Sinus des Wertes „1“ gleich „1.17520119364“.

Beispiel:

```
sinh(last(/host/key))
```

sqrt(value)

Die Quadratwurzel eines Wertes.
 Diese Funktion schlägt bei einem negativen Wert fehl.

Parameter:

- **value** - der zu prüfende Wert

Zum Beispiel ist die Quadratwurzel des Wertes „3.5“ gleich „1.87082869339“.

Beispiel:

```
sqrt(last(/host/key))
```

sum(<value1>,<value2>,...)

Die Summe der referenzierten Datenpunkt-Werte.

Parameter:

- **valueX** – der von einer anderen Funktion zurückgegebene Wert, die mit der Datenpunkt-Historie arbeitet.

Beispiel:

```
sum(avg(/host/key,1h),avg(/host2/key2,1h))
```

tan(value)

Der Tangens eines Werts.

Parameter:

- **value** - der zu prüfende Wert

Zum Beispiel ist der Tangens des Werts „1“ gleich „1.55740772465“.

Beispiel:

```
tan(last(/host/key))
```

truncate(value,decimal places)

Den Wert auf Dezimalstellen kürzen.

Parameter:

- **value** - der zu prüfende Wert;
- **decimal places** - Dezimalstellen für das Kürzen angeben (0 ist ebenfalls möglich).

Zum Beispiel wird ein Wert ’2.5482’, der auf 2 Dezimalstellen gekürzt wird, zu ’2.54’.

Beispiel:

```
truncate(last(/host/key),2)
```

Siehe alle unterstützten Funktionen.

Operatorfunktionen

Alle hier aufgeführten Funktionen werden unterstützt in:

- Auslöser-Ausdrücken
- Formeln für berechnete Datenpunkte
- Ausdrucksmakros

Die Funktionen werden ohne zusätzliche Informationen aufgeführt. Klicken Sie auf eine Funktion, um die vollständigen Details anzuzeigen.

Funktion	Beschreibung
between	Prüft, ob der Wert innerhalb des angegebenen Bereichs liegt.
in	Prüft, ob der Wert mindestens einem der aufgeführten Werte entspricht.

Funktionsdetails

Einige allgemeine Hinweise zu Funktionsparametern:

- Funktionsparameter werden durch ein Komma getrennt
- Ausdrücke werden als Parameter akzeptiert

between(value,min,max)

Prüft, ob der Wert zum angegebenen Bereich gehört.
 Unterstützte Werttypen: *Integer*, *Float*.
 Rückgabewert: 1 - im Bereich; 0 - andernfalls.

Parameter:

- **value** - der zu prüfende Wert;
- **min** - der Mindestwert;
- **max** - der Höchstwert.

Beispiel:

```
between(last(/host/key),1,10)=1 #Auslöser, wenn der Wert zwischen 1 und 10 liegt
```

in(value,value1,value2,...valueN)

Prüft, ob der Wert mindestens einem der aufgeführten Werte entspricht.
 Unterstützte Werttypen: *Integer*, *Float*, *Character*, *Text*, *Log*.
 Rückgabewert: 1 - wenn gleich; 0 - andernfalls.

Parameter:

- **value** - der zu prüfende Wert;
- **valueX** - aufgeführte Werte (Zeichenfolgenwerte müssen in doppelte Anführungszeichen gesetzt werden).

Der Wert wird mit den aufgeführten Werten als Zahl verglichen, wenn alle diese Werte in numerische Werte umgewandelt werden können; andernfalls erfolgt der Vergleich als Zeichenfolgen.

Beispiel:

```
in(last(/host/key),5,10)=1 #Auslöser, wenn der letzte Wert 5 oder 10 entspricht
in("text",last(/host/key),last(/host/key,#2))=1 #Auslöser, wenn "text" einem der letzten 2 Werte entsprich
```

Siehe alle unterstützten Funktionen.

Prädiktive Funktionen

Alle hier aufgeführten Funktionen werden unterstützt in:

- Auslöserausdrücken
- Formeln für berechnete Datenpunkte
- Ausdrucksmakros

Die Funktionen werden ohne zusätzliche Informationen aufgelistet. Klicken Sie auf die Funktion, um die vollständigen Details anzuzeigen.

Function	Description
forecast	Der zukünftige Wert, das Maximum, Minimum, Delta oder der Durchschnitt des Datenpunkts.
timeleft	Die in Sekunden benötigte Zeit, bis ein Datenpunkt den angegebenen Schwellenwert erreicht.

Allgemeine Parameter

- `/host/key` ist ein gemeinsamer obligatorischer erster Parameter für die Funktionen, die auf die Historie des Datenpunkts des Hosts verweisen.
- `(sec|#num)<:time shift>` ist ein gemeinsamer zweiter Parameter für die Funktionen, die auf die Historie des Datenpunkts des Hosts verweisen, wobei:
  - **sec** – maximaler Auswertungszeitraum in Sekunden (Zeit-Suffixe können verwendet werden), oder
  - **#num** – maximaler Auswertungsbereich in den zuletzt erfassten Werten (wenn ein Rautezeichen vorangestellt ist).
  - **time shift** (optional) – ermöglicht, den Auswertungspunkt zeitlich zurückzuverschieben. Weitere Details zur Angabe einer Zeitverschiebung.

Funktionsdetails

Einige allgemeine Hinweise zu Funktionsparametern:

- Funktionsparameter werden durch ein Komma getrennt
- Optionale Funktionsparameter (oder Parameterteile) werden durch < > angegeben
- Funktionsspezifische Parameter werden bei jeder Funktion beschrieben
- Die Parameter `/host/key` und `(sec|#num)<:time shift>` dürfen niemals in Anführungszeichen gesetzt werden

forecast(/host/key,(sec|#num)<:time shift>,time,<fit>,<mode>)

Der zukünftige Wert, das Maximum, das Minimum, die Differenz oder der Durchschnitt des Datenpunkts.
 Unterstützte Werttypen: *Float*, *Integer*.

Parameter:

- Siehe allgemeine Parameter;
- **time** - der Prognosezeitraum in Sekunden (z. B. 1, 60) (Zeitsuffixe können verwendet werden (z. B. 30s, 1m)); negative Werte werden unterstützt;
- **fit** (optional; muss in doppelte Anführungszeichen eingeschlossen werden) - die Funktion, die zur Anpassung historischer Daten verwendet wird. Unterstützte Anpassungen:
linear - lineare Funktion (Standard)
polynomialN - Polynom vom Grad N (1 <= N <= 6)
exponential - Exponentialfunktion
logarithmic - logarithmische Funktion
power - Potenzfunktion
Beachten Sie, dass *polynomial1 linear* entspricht;
- **mode** (optional; muss in doppelte Anführungszeichen eingeschlossen werden) - die gewünschte Ausgabe. Unterstützte Modi:
value - Wert (Standard)
max - Maximum
min - Minimum
delta - *max-min*
avg - Durchschnitt
Beachten Sie, dass *value* den Datenpunktwert zum Zeitpunkt `now + time` schätzt; *max*, *min*, *delta* und *avg* untersuchen die Schätzung des Datenpunktwerts im Intervall zwischen `now` und `now + time`.

Hinweise:

- Wenn der zurückzugebende Wert größer als 1.7976931348623158E+308 oder kleiner als -1.7976931348623158E+308 ist, wird der Rückgabewert entsprechend auf 1.7976931348623158E+308 oder -1.7976931348623158E+308 begrenzt;
- Wird nur dann nicht unterstützt, wenn die Funktion im Ausdruck falsch verwendet wird (falscher Datenpunkttyp, ungültige Parameter); andernfalls wird bei Fehlern -1 zurückgegeben;
- Weitere Informationen finden Sie unter prädiktiven Auslöserfunktionen.

Beispiele:

```
forecast(/host/key,#10,1h) #Prognose des Datenpunktwerts in einer Stunde basierend auf den letzten 10 Wert
forecast(/host/key,1h,30m) #Prognose des Datenpunktwerts in 30 Minuten basierend auf den Daten der letzten
forecast(/host/key,1h:now-1d,12h) #Prognose des Datenpunktwerts in 12 Stunden basierend auf einer Stunde D
forecast(/host/key,1h,10m,"exponential") #Prognose des Datenpunktwerts in 10 Minuten basierend auf den Dat
forecast(/host/key,1h,2h,"polynomial3","max") #Prognose des maximalen Werts, den der Datenpunkt in den näc
forecast(/host/key,#2,-20m) #Schätzung des Datenpunktwerts vor 20 Minuten basierend auf den letzten beiden
```

timeleft(/host/key,(sec|#num)<:time shift>,threshold,<fit>)

Die Zeit in Sekunden, die ein Datenpunkt benötigt, um den angegebenen Schwellenwert zu erreichen.
 Unterstützte Werttypen: *Float*, *Integer*.

Parameter:

- Siehe allgemeine Parameter;
- **threshold** - der zu erreichende Wert (Einheiten-Suffixe können verwendet werden);
- **fit** (optional; muss in doppelte Anführungszeichen eingeschlossen werden) - siehe forecast().

Kommentare:

- Wenn der zurückzugebende Wert größer als 1.7976931348623158E+308 ist, wird der Rückgabewert auf 1.7976931348623158E+308 begrenzt;
- Gibt 1.7976931348623158E+308 zurück, wenn der Schwellenwert nicht erreicht werden kann;
- Wird nur dann nicht unterstützt, wenn die Funktion im Ausdruck falsch verwendet wird (falscher Datenpunkttyp, ungültige Parameter); andernfalls wird bei Fehlern -1 zurückgegeben;
- Siehe auch zusätzliche Informationen zu prädiktiven Auslöserfunktionen.

Beispiele:

```
timeleft(/host/key,#10,0) #die Zeit, bis der Datenpunktwert basierend auf den letzten 10 Werten null errei
timeleft(/host/key,1h,100) #die Zeit, bis der Datenpunktwert basierend auf den Daten der letzten Stunde 10
timeleft(/host/key,1h:now-1d,100) #die Zeit, bis der Datenpunktwert 100 erreicht, basierend auf einer Stun
timeleft(/host/key,1h,200,"polynomial2") #die Zeit, bis der Datenpunktwert 200 erreicht, basierend auf den
```

Siehe alle unterstützten Funktionen.

Prädiktive Auslöserfunktionen

Übersicht

Manchmal gibt es Anzeichen für ein bevorstehendes Problem. Diese Anzeichen können erkannt werden, sodass im Voraus Maßnahmen ergriffen werden können, um die Auswirkungen des Problems zu verhindern oder zumindest zu minimieren.

Zabbix verfügt über Werkzeuge, um das zukünftige Verhalten des überwachten Systems auf Grundlage historischer Daten vorherzusagen. Diese Werkzeuge werden durch prädiktive Auslöserfunktionen umgesetzt.

Funktionen

Bevor ein Auslöser eingerichtet wird, muss definiert werden, was ein Problemzustand ist und wie viel Zeit benötigt wird, um Maßnahmen zu ergreifen. Danach gibt es zwei Möglichkeiten, einen Auslöser einzurichten, der auf eine potenziell unerwünschte Situation hinweist. Erstens: Der Auslöser muss auslösen, wenn erwartet wird, dass sich das System nach der „time to act" in einem Problemzustand befindet. Zweitens: Der Auslöser muss auslösen, wenn das System den Problemzustand in weniger als „time to act" erreichen wird. Die entsprechenden Auslöserfunktionen sind **forecast** und **timeleft**. Beachten Sie, dass die zugrunde liegende statistische Analyse für beide Funktionen im Wesentlichen identisch ist. Sie können einen Auslöser auf die von Ihnen bevorzugte Weise mit ähnlichen Ergebnissen einrichten.

Parameter

Beide Funktionen verwenden nahezu denselben Parametersatz. Verwenden Sie die Liste der unterstützten Funktionen als Referenz.

Zeitintervall

Zunächst sollten Sie den historischen Zeitraum angeben, den Zabbix analysieren soll, um die Vorhersage zu erstellen. Dies geschieht auf die gewohnte Weise mithilfe des Parameters `time period` und einer optionalen Zeitverschiebung, so wie Sie es auch bei den Funktionen **avg**, **count**, **delta**, **max**, **min** und **sum** tun.

Prognosehorizont

(nur **forecast**)
 Der Parameter `time` gibt an, wie weit in die Zukunft Zabbix die Abhängigkeiten extrapolieren soll, die es in historischen Daten findet. Unabhängig davon, ob Sie `time_shift` verwenden oder nicht, wird `time` immer ab dem aktuellen Zeitpunkt gezählt.

Zu erreichender Schwellenwert

(nur **timeleft**)
 Der Parameter `threshold` gibt einen Wert an, den der analysierte Datenpunkt erreichen muss, unabhängig davon, ob von oben oder von unten. Sobald wir f(t) bestimmt haben (siehe unten), sollten wir die Gleichung f(t) = `threshold` lösen und die Nullstelle zurückgeben, die näher an jetzt liegt und rechts von jetzt liegt, oder 1.7976931348623158E+308, wenn es keine solche Nullstelle gibt.

Note:
Wenn sich Datenpunktwerte dem Schwellenwert annähern und ihn dann überschreiten, geht **timeleft** davon aus, dass dieser Schnittpunkt bereits in der Vergangenheit liegt, und wechselt daher zum nächsten Schnittpunkt mit dem Niveau `threshold`, falls vorhanden. Es empfiehlt sich, Vorhersagen als Ergänzung zur gewöhnlichen Problemdiagnose zu verwenden, nicht als Ersatz.[1]

Anpassungsfunktionen

Standardmäßig ist `fit` die *lineare* Funktion. Wenn Ihr überwachtes System komplexer ist, stehen Ihnen weitere Optionen zur Auswahl.

`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	$x = a*t^b$

Modi

(nur **forecast**)
 Jedes Mal, wenn eine Auslöserfunktion ausgewertet wird, ruft sie Daten aus dem angegebenen Verlaufszeitraum ab und passt eine angegebene Funktion an die Daten an. Wenn sich die Daten also leicht unterscheiden, wird sich auch die angepasste Funktion leicht unterscheiden. Wenn wir einfach den Wert der angepassten Funktion zu einem angegebenen Zeitpunkt in der Zukunft berechnen, wissen Sie nichts darüber, wie sich der analysierte Datenpunkt zwischen jetzt und diesem Zeitpunkt in der Zukunft voraussichtlich verhalten wird. Für einige `fit`-Optionen (wie *polynomial*) kann ein einfacher Wert aus der Zukunft irreführend sein.

mode	**forecast**-Ergebnis
value	f(now + `time`)
max	$\max_{\text{now <= t <= now + time}}$ f(t)
min	$\min_{\text{now <= t <= now + time}}$ f(t)
delta	*max* - *min*
avg	Durchschnitt von f(t) (now <= t <= now + `time`) gemäß Definition

Details

Um Berechnungen mit sehr großen Zahlen zu vermeiden, betrachten wir den Zeitstempel des ersten Werts im angegebenen Zeitraum plus 1 ns als neue Nullzeit (die aktuelle Epoch-Zeit liegt in der Größenordnung von 10^9, das Quadrat der Epoch-Zeit bei 10^{18}, die doppelte Genauigkeit bei etwa 10^{-16}). 1 ns wird hinzugefügt, um für *logarithmische* und *Potenz*-Anpassungen, bei denen log(t) berechnet wird, ausschließlich positive Zeitwerte bereitzustellen. Die Zeitverschiebung beeinflusst *lineare*, *polynomiale* und *exponentielle* Anpassungen nicht (abgesehen von einfacheren und genaueren Berechnungen), verändert jedoch die Form von *logarithmischen* und *Potenz*-Funktionen.

Mögliche Fehler

Funktionen geben in solchen Situationen -1 zurück:

- der angegebene Auswertungszeitraum enthält keine Daten;
- das Ergebnis der mathematischen Operation ist nicht definiert[3];
- numerische Komplikationen (leider werden bei einigen Eingabedatensätzen Bereich und Genauigkeit des Gleitkommaformats mit doppelter Genauigkeit unzureichend)[4].

Note:
Es werden keine Warnungen oder Fehler markiert, wenn die gewählte Anpassung die bereitgestellten Daten schlecht beschreibt oder einfach zu wenige Daten für eine genaue Vorhersage vorhanden sind.

Beispiele und Umgang mit Fehlern

Um eine Warnung zu erhalten, wenn Ihnen auf Ihrem Host bald der freie Festplattenspeicher ausgeht, können Sie einen Auslöser-Ausdruck wie diesen verwenden:

```
timeleft(/host/vfs.fs.size[/,free],1h,0)<1h
```

Allerdings kann der Fehlercode -1 ins Spiel kommen und Ihren Auslöser in einen Problemzustand versetzen. Im Allgemeinen ist das gut, weil Sie eine Warnung erhalten, dass Ihre Vorhersagen nicht korrekt funktionieren und Sie diese genauer untersuchen sollten, um herauszufinden, warum. Manchmal ist es jedoch schlecht, weil -1 einfach bedeuten kann, dass in der letzten Stunde keine Daten über den freien Festplattenspeicher des Hosts erfasst wurden. Wenn Sie zu viele Fehlalarme erhalten, sollten Sie einen komplizierteren Auslöser-Ausdruck in Betracht ziehen [5]:

```
timeleft(/host/vfs.fs.size[/,free],1h,0)<1h and timeleft(/host/vfs.fs.size[/,free],1h,0)<>-1
```

Bei **forecast** ist die Situation etwas schwieriger. Zunächst einmal kann -1 den Auslöser je nachdem, ob Sie einen Ausdruck wie `forecast(/host/item,(...))<...` oder wie `forecast(/host/item,(...))>...` haben, in einen Problemzustand versetzen oder auch nicht.

Außerdem kann -1 eine gültige Vorhersage sein, wenn es normal ist, dass der Datenpunkt-Wert negativ ist. Die Wahrscheinlichkeit, dass diese Situation in der Praxis auftritt, ist jedoch vernachlässigbar (siehe wie der Operator = funktioniert). Fügen Sie also `... or forecast(/host/item,(...))=-1` oder `... and forecast(/host/item,(...))<>-1` hinzu, je nachdem, ob Sie -1 als Problem behandeln möchten oder nicht.

Fußnoten

1 Zum Beispiel kann ein einfacher Auslöser wie `timeleft(/host/item,1h,X) < 1h` in einen Problemzustand wechseln, wenn sich der Datenpunkt-Wert X nähert, und sich dann plötzlich erholen, sobald der Wert X erreicht ist. Wenn das Problem darin besteht, dass der Datenpunkt-Wert unter X liegt, verwenden Sie: `last(/host/item) < X or timeleft(/host/item,1h,X) < 1h` Wenn das Problem darin besteht, dass der Datenpunkt-Wert über X liegt, verwenden Sie: `last(/host/item) > X or timeleft(/host/item,1h,X) < 1h`

2 Der Polynomgrad kann zwischen 1 und 6 liegen, *polynomial1* ist äquivalent zu *linear*. Verwenden Sie jedoch Polynome höheren Grades mit Vorsicht. Wenn der Auswertungszeitraum weniger Punkte enthält als zur Bestimmung der Polynomkoeffizienten erforderlich sind, wird der Polynomgrad verringert (z. B. wird *polynomial5* angefordert, es gibt aber nur 4 Punkte, daher wird *polynomial3* angepasst).

[3] Zum Beispiel umfasst das Anpassen von Funktionen wie *exponential* oder *power* die Berechnung von log() aus Datenpunkt-Werten. Wenn die Daten Nullen oder negative Zahlen enthalten, erhalten Sie einen Fehler, da log() nur für positive Werte definiert ist.

[4] Für *linear*, *exponential*, *logarithmic* und *power* können alle erforderlichen Berechnungen explizit geschrieben werden. Für *polynomial* kann nur *value* ohne zusätzliche Schritte berechnet werden. Die Berechnung von *avg* umfasst die Berechnung der Stammfunktion des Polynoms (analytisch). Die Berechnung von *max*, *min* und *delta* umfasst die Berechnung der Ableitung des Polynoms (analytisch) und das Finden ihrer Nullstellen (numerisch). Das Lösen von f(t) = 0 umfasst das Finden der Polynom-Nullstellen (numerisch).

[5] In diesem Fall kann -1 jedoch dazu führen, dass sich Ihr Auslöser aus dem Problemzustand erholt. Um vollständig geschützt zu sein, verwenden Sie: `timeleft(/host/vfs.fs.size[/,free],1h,0)<1h and ({TRIGGER.VALUE}=0 and timeleft(/host/vfs.fs.size[/,free],1h,0)<>-1 or {TRIGGER.VALUE}=1)`

Zeichenfolgenfunktionen

Alle hier aufgeführten Funktionen werden unterstützt in:

- Auslöserausdrücken
- Formeln für berechnete Datenpunkte
- Ausdrucksmakros

Die Funktionen werden ohne zusätzliche Informationen aufgeführt. Klicken Sie auf eine Funktion, um die vollständigen Details anzuzeigen.

Funktion	Beschreibung
ascii	Der ASCII-Code des am weitesten links stehenden Zeichens des Werts.
bitlength	Die Länge des Werts in Bits.
bytelength	Die Länge des Werts in Bytes.
char	Gibt das Zeichen zurück, indem der Wert als ASCII-Code interpretiert wird.
concat	Die Zeichenfolge, die durch die Verkettung der referenzierten Datenpunktwerte oder konstanten Werte entsteht.
contains	Gibt 1 zurück, wenn der Wert die durch `search value` angegebene Teilzeichenfolge enthält, andernfalls 0.
insert	Fügt die angegebenen Zeichen oder Leerzeichen ab der angegebenen Position in der Zeichenfolge in die Zeichenfolge ein.
jsonpath	Gibt das JSONPath-Ergebnis zurück.
left	Gibt die am weitesten links stehenden Zeichen des Werts zurück.
length	Die Länge des Werts in Zeichen.
ltrim	Entfernt die angegebenen Zeichen vom Anfang der Zeichenfolge.
mid	Gibt eine Teilzeichenfolge mit N Zeichen zurück, die an der durch „start“ angegebenen Zeichenposition beginnt. Diese Funktion ist ein Alias für die Funktion `substring()`.
repeat	Wiederholt eine Zeichenfolge.
replace	Sucht das Muster im Wert und ersetzt es durch den Ersatzwert.
right	Gibt die am weitesten rechts stehenden Zeichen des Werts zurück.
rtrim	Entfernt die angegebenen Zeichen vom Ende der Zeichenfolge.
substring	Gibt eine Teilzeichenfolge mit N Zeichen zurück, die an der durch „start“ angegebenen Zeichenposition beginnt.
trim	Entfernt die angegebenen Zeichen vom Anfang und Ende der Zeichenfolge.
xmlxpath	Gibt das XML-XPath-Ergebnis zurück.

Funktionsdetails

Einige allgemeine Hinweise zu Funktionsparametern:

- Funktionsparameter werden durch ein Komma getrennt
- Ausdrücke werden als Parameter akzeptiert
- Zeichenkettenparameter müssen in doppelte Anführungszeichen gesetzt werden; andernfalls könnten sie falsch interpretiert werden
- Optionale Funktionsparameter (oder Parameterteile) werden durch < > gekennzeichnet

ascii(value)

Der ASCII-Code des am weitesten links stehenden Zeichens des Werts.
 Unterstützte Werttypen: *String*, *Text*, *Log*.

Parameter:

- **value** - der zu prüfende Wert

Zum Beispiel gibt ein Wert wie 'Abc' '65' zurück (ASCII-Code für 'A').

Beispiel:

```
ascii(last(/host/key))
```

bitlength(value)

Die Länge von value in Bits.
 Unterstützte Werttypen: *String*, *Text*, *Log*, *Integer*.

Parameter:

- **value** - der zu prüfende Wert

Beispiel:

```
bitlength(last(/host/key))
```

bytelength(value)

Die Länge von value in Byte.
 Unterstützte Werttypen: *String*, *Text*, *Log*, *Integer*.

Parameter:

- **value** - der zu prüfende Wert

Beispiel:

```
bytelength(last(/host/key))
```

char(value)

Gibt das Zeichen zurück, indem der Wert als ASCII-Code interpretiert wird.
 Unterstützte Werttypen: *Integer*.

Parameter:

- **value** - der zu prüfende Wert

Der Wert muss im Bereich 0-255 liegen. Zum Beispiel gibt ein Wert wie '65' (als ASCII-Code interpretiert) 'A' zurück.

Beispiel:

```
char(last(/host/key))
```

concat(<value1>,<value2>,...)

Die Zeichenkette, die durch das Verketten der referenzierten Datenpunktwerte oder konstanter Werte entsteht.
 Unterstützte Werttypen: *String*, *Text*, *Log*, *Float*, *Integer*.

Parameter:

- **valueX** - der von einer der Verlaufsfunktionen zurückgegebene Wert oder ein konstanter Wert (Zeichenkette, Ganzzahl oder Gleitkommazahl). Muss mindestens zwei Parameter enthalten.

Zum Beispiel ergibt ein Wert wie 'Zab', der mit 'bix' (der konstanten Zeichenkette) verkettet wird, 'Zabbix'.

Beispiele:

```
concat(last(/host/key),"bix")
concat("1 min: ",last(/host/system.cpu.load[all,avg1]),", 15 min: ",last(/host/system.cpu.load[all,avg15])
```

contains(value, search value)

Gibt 1 zurück, wenn der Wert die als `search value` angegebene Teilzeichenfolge enthält, andernfalls 0.| Unterstützte Werttypen: *String*, *Text*, *Log*.

Parameter:

- **value** – der zu überprüfende Wert;
- **search value** – der zu suchende Wert.

Kommentare:

- Bei der Suche wird zwischen Groß- und Kleinschreibung unterschieden.
- Wenn der Wert gültig und `search value` leer ist, gibt die Funktion 1 zurück.

Beispielsweise gibt die Funktion 1 zurück, wenn der Wert „Zabbix" enthält.

Beispiel:

```
contains(last(/host/key),"Zabbix")
```

insert(value,start,length,replacement)

Fügt angegebene Zeichen oder Leerzeichen in die Zeichenkette ein, beginnend an der angegebenen Position in der Zeichenkette.
 Unterstützte Werttypen: *String*, *Text*, *Log*.

Parameter:

- **value** - der zu prüfende Wert;
- **start** - Startposition;
- **length** - zu ersetzende Positionen;
- **replacement** - Ersetzungszeichenkette.

Zum Beispiel wird ein Wert wie 'Zabbbix' durch 'Zabbix' ersetzt, wenn 'bb' (Startposition 3, zu ersetzende Positionen 2) durch 'b' ersetzt wird.

Beispiel:

```
insert(last(/host/key),3,2,"b")
```

jsonpath(value,path,<default>)

Gibt das JSONPath-Ergebnis zurück.
 Unterstützte Werttypen: *String*, *Text*, *Log*.

Parameter:

- **value** - der zu prüfende Wert;
- **path** - der Pfad (muss in Anführungszeichen gesetzt werden);
- **default** - der optionale Ersatzwert, falls die JSONPath-Abfrage keine Daten zurückgibt. Beachten Sie, dass bei anderen Fehlern ein Fehlschlag zurückgegeben wird (z. B. „unsupported construct").

Beispiel:

```
jsonpath(last(/host/proc.get[zabbix_agentd,,,summary]),"$..size")
```

left(value,count)

Gibt die Zeichen ganz links im Wert zurück.
 Unterstützte Werttypen: *String*, *Text*, *Log*.

Parameter:

- **value** - der zu prüfende Wert;
- **count** - die Anzahl der zurückzugebenden Zeichen.

Zum Beispiel können Sie aus 'Zabbix' 'Zab' zurückgeben, indem Sie angeben, dass die 3 Zeichen ganz links zurückgegeben werden sollen. Siehe auch right().

Beispiel:

```
left(last(/host/key),3) #gibt die drei Zeichen ganz links zurück
```

length(value)

Die Länge von value in Zeichen.
 Unterstützte Werttypen: *String*, *Text*, *Log*.

Parameter:

- **value** – der zu prüfende Wert.

Beispiele:

```
length(last(/host/key)) #die Länge des letzten Werts
length(last(/host/key,#3)) #die Länge des drittletzten Werts
length(last(/host/key,#1:now-1d)) #die Länge des letzten Werts vor einem Tag
```

ltrim(value,<chars>)

Entfernt angegebene Zeichen vom Anfang einer Zeichenkette.
 Unterstützte Wertetypen: *String*, *Text*, *Log*.

Parameter:

- **value** - der zu prüfende Wert;
- **chars** (optional) - gibt die zu entfernenden Zeichen an.

Standardmäßig werden führende Leerraumzeichen entfernt (wenn keine optionalen Zeichen angegeben sind). Siehe auch: rtrim(), trim().

Beispiele:

```
ltrim(last(/host/key)) #entfernt Leerraum vom Anfang der Zeichenkette
ltrim(last(/host/key),"Z") #entfernt alle 'Z' vom Anfang der Zeichenkette
ltrim(last(/host/key)," Z") #entfernt alle Leerzeichen und 'Z' vom Anfang der Zeichenkette
```

mid(value,start,length)

Gibt eine Teilzeichenfolge mit N Zeichen zurück, beginnend an der durch „start“ angegebenen Zeichenposition.
Diese Funktion ist ein Alias für die Funktion `substring()`.
 Unterstützte Werttypen: *String*, *Text*, *Log*.

Parameter:

- **value** – der zu prüfende Wert;
- **start** – Startposition der Teilzeichenfolge;
- **length** – Anzahl der in der Teilzeichenfolge zurückzugebenden Zeichen.

Beispielsweise kann aus einem Wert wie „Zabbix“ „abbi“ zurückgegeben werden, wenn die Startposition 2 und die Anzahl der zurückzugebenden Zeichen 4 ist.

Beispiel:

```
mid(last(/host/key),2,4)="abbi"
```

repeat(value,count)

Wiederholt eine Zeichenkette.
 Unterstützte Wertetypen: *String*, *Text*, *Log*.

Parameter:

- **value** - der zu prüfende Wert;
- **count** - die Anzahl der Wiederholungen.

Beispiel:

```
repeat(last(/host/key),2) #Wert zweimal wiederholen
```

replace(value,pattern,replacement)

Sucht das Muster im Wert und ersetzt es durch replacement. Alle Vorkommen des Musters werden ersetzt.
 Unterstützte Werttypen: *String*, *Text*, *Log*.

Parameter:

- **value** - der zu prüfende Wert;
- **pattern** - das zu findende Muster;
- **replacement** - die Zeichenfolge, durch die das Muster ersetzt wird.

Beispiel:

```
replace(last(/host/key),"ibb","abb") #alle 'ibb' durch 'abb' ersetzen
```

right(value,count)

Gibt die Zeichen ganz rechts im Wert zurück.
 Unterstützte Werttypen: *String*, *Text*, *Log*.

Parameter:

- **value** - der zu prüfende Wert;
- **count** - die Anzahl der zurückzugebenden Zeichen.

Zum Beispiel können Sie aus ’Zabbix’ ’bix’ zurückgeben, indem Sie angeben, dass die 3 Zeichen ganz rechts zurückgegeben werden sollen. Siehe auch left().

Beispiel:

```
right(last(/host/key),3) #gibt die drei Zeichen ganz rechts zurück
```

rtrim(value,<chars>)

Entfernt angegebene Zeichen vom Ende einer Zeichenkette.
 Unterstützte Werttypen: *String*, *Text*, *Log*.

Parameter:

- **value** - der zu prüfende Wert;
- **chars** (optional) - gibt die zu entfernenden Zeichen an.

Leerraum wird standardmäßig rechts abgeschnitten (wenn keine optionalen Zeichen angegeben sind). Siehe auch: ltrim(), trim().

Beispiele:

```
rtrim(last(/host/key)) #Leerraum vom Ende der Zeichenkette entfernen
rtrim(last(/host/key),"x") #beliebiges 'x' vom Ende der Zeichenkette entfernen
rtrim(last(/host/key),"x ") #beliebiges 'x' und Leerzeichen vom Ende der Zeichenkette entfernen
```

substring(value,start,length)

Gibt einen Teilstring mit N Zeichen zurück, der an der durch „start“ angegebenen Zeichenposition beginnt.
 Unterstützte Werttypen: *String*, *Text*, *Log*.

Parameter:

- **value** – der zu prüfende Wert;
- **start** – Startposition des Teilstrings;
- **length** – die Anzahl der im Teilstring zurückzugebenden Zeichen.

Beispielsweise kann aus einem Wert wie „Zabbix“ der Teilstring „abbi“ zurückgegeben werden, wenn die Startposition 2 und die Anzahl der zurückzugebenden Zeichen 4 ist.

Beispiel:

```
substring(last(/host/key),2,4)="abbi"
```

trim(value,<chars>)

Entfernt angegebene Zeichen vom Anfang und Ende einer Zeichenkette.
 Unterstützte Werttypen: *String*, *Text*, *Log*.

Parameter:

- **value** - der zu prüfende Wert;
- **chars** (optional) - gibt die zu entfernenden Zeichen an.

Standardmäßig werden Leerraumzeichen auf beiden Seiten entfernt (wenn keine optionalen Zeichen angegeben sind). Siehe auch: ltrim(), rtrim().

Beispiele:

```
trim(last(/host/key)) #entfernt Leerraumzeichen vom Anfang und Ende der Zeichenkette
trim(last(/host/key),"_") #entfernt '_' vom Anfang und Ende der Zeichenkette
```

xmlxpath(value,path,<default>)

Gibt das XML-XPath-Ergebnis zurück.
 Unterstützte Werttypen: *String*, *Text*, *Log*.

Parameter:

- **value** - der zu prüfende Wert;
- **path** - der Pfad (muss in Anführungszeichen gesetzt werden);
- **default** - der optionale Ersatzwert, falls die XML-XPath-Abfrage eine leere Knotenmenge zurückgibt. Er wird nicht zurückgegeben, wenn das leere Ergebnis keine Knotenmenge ist (d. h. eine leere Zeichenkette). Bei anderen Fehlern wird ein Fehlschlag zurückgegeben (z. B. „ungültiger Ausdruck“).

Beispiel:

```
xmlxpath(last(/host/xml_result),"/response/error/status")
```

Siehe alle unterstützten Funktionen.

Best Practices für Auslöser

Auslöser sind ein leistungsstarkes Werkzeug, können jedoch auch unerwünschte Alarmmeldungen verursachen. Um mehr echte Signale und weniger Rauschen zu sehen, befolgen Sie diese Tipps:

1. Entschärfen Sie Auslöser. Statt auf den letzten Wert zu alarmieren (zu hoch/zu niedrig), analysieren Sie den Durchschnitt über einen längeren Zeitraum mithilfe von Funktionen wie **avg**, **min** und **max**.
2. Erwägen Sie die Verwendung der Funktion **percentile** (setzen Sie sie auf 95 % oder 5 %) in Auslösern, wenn Sie Alarme bei zufälligen Spitzen und Einbrüchen vermeiden möchten.
3. Verwenden Sie Hysterese, um ein Flattern von Auslösern zu vermeiden - häufige Zustandswechsel von Auslösern (OK ↔ Problem). Ein Kontinuum für den Problemzustand kann durch Hinzufügen eines Wiederherstellungsausdrucks definiert werden (eine separate Bedingung für die Problembehebung).
4. Verwenden Sie Auslöser-Abhängigkeiten, um Alarme zu vermeiden, die nicht mit der eigentlichen Ursache zusammenhängen.
5. Verwenden Sie die Auslöser-Schweregrad, um nur bei schwerwiegenderen Problemen zu alarmieren.

6. Definieren Sie Wartungs-Fenster.

Hysterese

Manchmal wird ein Intervall zwischen Problem- und Wiederherstellungszuständen benötigt, anstatt eines einfachen Schwellenwerts. Wenn wir zum Beispiel einen Auslöser definieren möchten, der ein Problem meldet, wenn die Temperatur im Serverraum über 20 °C steigt, und wir möchten, dass er im Problemzustand bleibt, bis die Temperatur unter 15 °C fällt, dann reicht ein einfacher Auslöser-Schwellenwert bei 20 °C nicht aus.

Stattdessen müssen wir zunächst einen Auslöser-Ausdruck für das Problemereignis definieren (Temperatur über 20 °C). Dann müssen wir eine zusätzliche Wiederherstellungsbedingung definieren (Temperatur unter 15 °C). Dies geschieht durch das Definieren eines *Wiederherstellungsausdrucks* beim Konfigurieren eines Auslösers.

In diesem Fall erfolgt die Problembehebung in zwei Schritten:

- Zuerst muss der Problemausdruck (Temperatur über 20 °C) zu FALSE ausgewertet werden
- Danach muss der Wiederherstellungsausdruck (Temperatur unter 15 °C) zu TRUE ausgewertet werden

Der Wiederherstellungsausdruck wird erst ausgewertet, nachdem das Problemereignis behoben wurde. Dass der Wiederherstellungsausdruck allein TRUE ist, behebt ein Problem nicht, wenn der Problemausdruck weiterhin TRUE ist!

Beispiel

Die Temperatur im Serverraum ist zu hoch.

Problemausdruck:

```
last(/server/temp)>20
```

Wiederherstellungsausdruck:

```
last(/server/temp)<=15
```

Note:
Es ist nicht sinnvoll, das Makro {TRIGGER.VALUE} in einem Wiederherstellungsausdruck zu verwenden, da dieser Ausdruck nur ausgewertet wird, wenn sich der Auslöser im Status „Problem“ befindet. Folglich wird {TRIGGER.VALUE} bei der Auswertung des Ausdrucks immer zu „1“ aufgelöst (was auf den Status „Problem“ hinweist).

Auslöser-Abhängigkeiten

Auslöser-Abhängigkeiten können verwendet werden, um Warnmeldungen zu vermeiden, die nicht mit der Grundursache zusammenhängen.

Siehe alle Best Practices.

Übersicht

Manchmal hängt die Verfügbarkeit eines Hosts von einem anderen ab. Ein Server hinter einem Router wird nicht mehr erreichbar sein, wenn der Router ausfällt. Wenn für beide Auslöser konfiguriert sind, erhalten Sie möglicherweise Benachrichtigungen über zwei ausgefallene Hosts, obwohl nur der Router die Ursache war.

In diesem Fall kann eine Abhängigkeit zwischen Hosts hilfreich sein. Wenn eine Abhängigkeit festgelegt ist, können Benachrichtigungen für abhängige Hosts zurückgehalten und nur die Benachrichtigung über das grundlegende Problem gesendet werden.

Zabbix unterstützt zwar keine direkten Abhängigkeiten zwischen Hosts, diese können jedoch mit einer anderen, flexibleren Methode definiert werden: Auslöserabhängigkeiten. Ein Auslöser kann von einem oder mehreren anderen Auslösern abhängen.

In unserem einfachen Beispiel öffnen wir das Konfigurationsformular des Server-Auslösers und legen fest, dass er vom entsprechenden Auslöser des Routers abhängt. Bei einer solchen Abhängigkeit ändert der Server-Auslöser seinen Status nicht, solange der Auslöser, von dem er abhängt, den Status „PROBLEM“ hat. Daher werden keine abhängigen Aktionen ausgeführt und keine Benachrichtigungen gesendet.

Wenn sowohl der Server als auch der Router ausgefallen sind und eine Abhängigkeit zwischen dem Auslöser „Server ausgefallen“ und dem Auslöser „Router ausgefallen“ festgelegt ist, führt Zabbix keine Aktionen für den abhängigen Auslöser aus.

Solange der übergeordnete Auslöser den Status PROBLEM hat, melden seine abhängigen Auslöser möglicherweise Werte, denen nicht vertraut werden kann. Daher werden abhängige Auslöser erst dann erneut ausgewertet, wenn der übergeordnete Auslöser (im obigen Beispiel der Router):

- vom Status „PROBLEM“ wieder in den Status „OK“ wechselt;
- seinen Status von „PROBLEM“ in „UNKNOWN“ ändert;
- manuell, durch Korrelation oder mithilfe der Funktionen date and time und/oder nodata() geschlossen wird;

- durch den Wert eines Datenpunkts aufgelöst wird, der nicht am abhängigen Auslöser beteiligt ist;
- deaktiviert wird, einen deaktivierten Datenpunkt hat oder zu einem deaktivierten Host gehört.

In allen oben genannten Fällen wird der abhängige Auslöser (Server) erst dann erneut ausgewertet, wenn eine neue Metrik für ihn empfangen wird. Das bedeutet, dass der abhängige Auslöser möglicherweise nicht sofort aktualisiert wird.

Außerdem gilt:

- Eine Auslöserabhängigkeit kann von einem Host-Auslöser zu jedem anderen Host-Auslöser hinzugefügt werden, sofern dadurch keine zirkuläre Abhängigkeit entsteht.
- Eine Auslöserabhängigkeit kann von einer Vorlage zu einer anderen Vorlage hinzugefügt werden. Wenn ein Auslöser aus Vorlage A von einem Auslöser aus Vorlage B abhängt, kann Vorlage A nur zusammen mit Vorlage B mit einem Host (oder einer anderen Vorlage) verknüpft werden. Vorlage B kann jedoch auch allein mit einem Host (oder einer anderen Vorlage) verknüpft werden.
- Eine Auslöserabhängigkeit kann von einem Vorlagen-Auslöser zu einem Host-Auslöser hinzugefügt werden. In diesem Fall wird durch die Verknüpfung einer solchen Vorlage mit einem Host ein Host-Auslöser erstellt, der vom selben Auslöser der Vorlage abhängt, von dem der ursprüngliche Auslöser abhängig war. So können Sie beispielsweise eine Vorlage erstellen, in der einige Auslöser von den Router-Auslösern eines Hosts abhängen. Alle mit dieser Vorlage verknüpften Hosts hängen von diesem bestimmten Router ab.
- Eine Auslöserabhängigkeit kann nicht von einem Host-Auslöser zu einem Vorlagen-Auslöser hinzugefügt werden.
- Eine Auslöserabhängigkeit kann von einem Auslöserprototyp zu einem anderen Auslöserprototyp (innerhalb derselben Low-Level-Discovery-Regel) oder zu einem tatsächlichen Auslöser hinzugefügt werden. Ein Auslöserprototyp darf nicht von einem Auslöserprototyp aus einer anderen LLD-Regel oder von einem aus einem Auslöserprototyp erstellten Auslöser abhängen. Ein Host-Auslöserprototyp kann nicht von einem Auslöser aus einer Vorlage abhängen.

Konfiguration

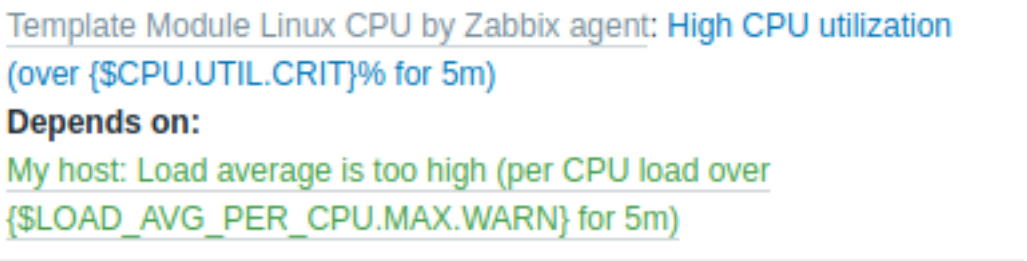
So definieren Sie eine Abhängigkeit:

1. Öffnen Sie die Registerkarte *Abhängigkeiten* im Konfigurationsformular für Auslöser.
2. Klicken Sie im Abschnitt *Abhängigkeiten* auf *Hinzufügen* und wählen Sie einen oder mehrere Auslöser aus, von denen der Auslöser abhängig sein soll.



3. Klicken Sie auf *Aktualisieren*.

Nun wird die Abhängigkeit des Auslösers in der Liste angezeigt.



Beispiel für mehrere Abhängigkeiten

Zum Beispiel befindet sich der Host hinter dem Router2 und der Router2 hinter dem Router1.

```
Zabbix - Router1 - Router2 - Host
```

Wenn der Router1 ausgefallen ist, dann sind natürlich auch der Host und der Router2 nicht erreichbar, doch drei Benachrichtigungen darüber zu erhalten, dass der Host, der Router1 und der Router2 ausgefallen sind, ist übertrieben.

In diesem Fall definieren wir also zwei Abhängigkeiten:

```
the 'Host is down' trigger depends on the 'Router2 is down' trigger
the 'Router2 is down' trigger depends on the 'Router1 is down' trigger
```

Bevor der Status des Auslösers 'Host is down' geändert wird, prüft Zabbix die entsprechenden Auslöserabhängigkeiten. Wenn solche gefunden werden und einer dieser Auslöser sich im Zustand `Problem` befindet, wird der Auslöserstatus nicht geändert, die Aktionen werden nicht ausgeführt und es werden keine Benachrichtigungen gesendet.

Zabbix führt diese Prüfung rekursiv durch. Wenn der Router1 oder der Router2 nicht erreichbar ist, wird der Host-Auslöser nicht aktualisiert.

Auslöser-Schweregrad

Der Auslöser-Schweregrad kann verwendet werden, um Alarmrauschen bei weniger schwerwiegenden Problemen zu vermeiden.

Siehe alle Best Practices.

Überblick

Der Schweregrad stellt die relative Wichtigkeit eines Auslösers dar.



Die Zuweisung eines Schweregrads ist nicht nur für die visuelle Darstellung nützlich, sondern auch, um Benachrichtigungen bei trivialen Problemen zu vermeiden.

Auslöser-Schweregrade werden verwendet für:

- Visuelle Darstellung (eine eigene Farbe für jedes erstellte Problem, basierend auf dem ursprünglichen Auslöser-Schweregrad)
- Einschränkung von Benachrichtigungen basierend auf dem Schweregrad (siehe Aktionsbedingungen)
- Benutzermedien - unterschiedliche Medien (Benachrichtigungskanäle) für unterschiedliche Schweregrade (zum Beispiel SMS für Auslöser mit dem Schweregrad *Hoch* und *Katastrophe* sowie E-Mail für Auslöser mit anderen Schweregraden)
- Audio in globalen Alarmen - unterschiedliche Audiosignale für unterschiedliche Schweregrade

Die folgenden Auslöser-Schweregrade werden standardmäßig unterstützt:

Schweregrad	Farbe	Beschreibung
Nicht klassifiziert	Grau	Kann verwendet werden, wenn der Schweregrad eines Ereignisses unbekannt ist, noch nicht bestimmt wurde, nicht zum regulären Überwachungsumfang gehört usw., zum Beispiel während der Erstkonfiguration, als Platzhalter für eine spätere Bewertung oder als Teil eines Integrationsprozesses.
Information	Hellblau	Kann für informative Ereignisse verwendet werden, die keine sofortige Aufmerksamkeit erfordern, aber dennoch wertvolle Einblicke liefern können.
Warnung	Gelb	Kann verwendet werden, um auf ein potenzielles Problem hinzuweisen, das möglicherweise untersucht werden muss oder Maßnahmen erfordert, aber nicht kritisch ist.
Durchschnitt	Orange	Kann verwendet werden, um auf ein erhebliches Problem hinzuweisen, das relativ bald behoben werden sollte, um weitere Probleme zu vermeiden.
Hoch	Hellrot	Kann verwendet werden, um auf kritische Probleme hinzuweisen, die sofortige Aufmerksamkeit erfordern, um erhebliche Störungen zu vermeiden.
Katastrophe	Rot	Kann verwendet werden, um auf einen schwerwiegenden Vorfall hinzuweisen, der sofortiges Handeln erfordert, um beispielsweise Systemausfälle oder Datenverlust zu verhindern.

Namen und Farben der Auslöser-Schweregrade können unter *Administration* > *General* > *Trigger displaying options* **angepasst** werden. Farben werden von allen Frontend-Themes gemeinsam verwendet.

Übersetzen benutzerdefinierter Schweregradnamen

> **Attention:**
> Wenn Übersetzungen des Zabbix Frontends verwendet werden, überschreiben benutzerdefinierte Schweregradnamen standardmäßig die übersetzten Namen.

Standardnamen für Auslöser-Schweregrade stehen in allen Gebietsschemata zur Übersetzung zur Verfügung.
Wenn ein Schweregradname geändert wird, wird in allen Gebietsschemata ein benutzerdefinierter Name verwendet, und eine zusätzliche manuelle Übersetzung ist erforderlich.

Vorgehensweise zum Übersetzen benutzerdefinierter Schweregradnamen:

- den erforderlichen benutzerdefinierten Schweregradnamen festlegen, zum Beispiel „Important“
- folgende Datei bearbeiten: <frontend_dir>/locale/<required_locale>/LC_MESSAGES/frontend.po
- 2 Zeilen hinzufügen:

```
msgid "Important"
msgstr "<Übersetzungszeichenfolge>"
```

und die Datei speichern.

- .mo-Dateien erstellen, wie in <frontend_dir>/locale/README beschrieben

Dabei muss **msgid** mit dem neuen benutzerdefinierten Schweregradnamen übereinstimmen, und **msgstr** muss dessen Übersetzung in die jeweilige Sprache sein.

Dieser Vorgang muss nach jeder Änderung eines Schweregradnamens durchgeführt werden.

Visualisierung

Überblick

Die Visualisierung bietet verschiedene Möglichkeiten, die von Zabbix erfassten Monitoring-Daten anzuzeigen. Diagramme zeigen Trends und Metriken, Netzwerkkarten stellen die Infrastrukturtopologie dar, und Dashboards kombinieren anpassbare Widgets, um umfassende Monitoring-Ansichten bereitzustellen.

Visualisierungswerkzeuge

- Graphen - Daten von Datenpunkten als einfache automatisch generierte Graphen anzeigen, benutzerdefinierte Graphen erstellen oder mehrere Datenpunkte in Ad-hoc-Graphen vergleichen.
- Netzwerkkarten - visuelle Darstellungen der Netzwerkinfrastruktur mit anpassbaren Elementen und Indikatoren erstellen.
- Dashboards - Layouts anpassen, die verschiedene Dashboard-Widgets kombinieren, welche Daten in unterschiedlichen Formaten visualisieren.

Diagramme

Übersicht

Wenn viele Daten in Zabbix einfließen, wird es für die Benutzer wesentlich einfacher, eine visuelle Darstellung der Vorgänge zu betrachten, anstatt nur Zahlen zu sehen.

Hier kommen Diagramme ins Spiel. Diagramme ermöglichen es, den Datenfluss auf einen Blick zu erfassen, Probleme miteinander in Beziehung zu setzen, festzustellen, wann etwas begonnen hat, oder zu veranschaulichen, wann sich etwas zu einem Problem entwickeln könnte.

Zabbix bietet Benutzern:

- Integrierte einfache Diagramme für die Daten eines Datenpunkts.
- Die Möglichkeit, komplexere benutzerdefinierte Diagramme zu erstellen.
- Schnellen Zugriff auf einen Vergleich mehrerer Datenpunkte in Ad-hoc-Diagrammen.
- Moderne, anpassbare Vektor-Diagramme und Kreisdiagramme.

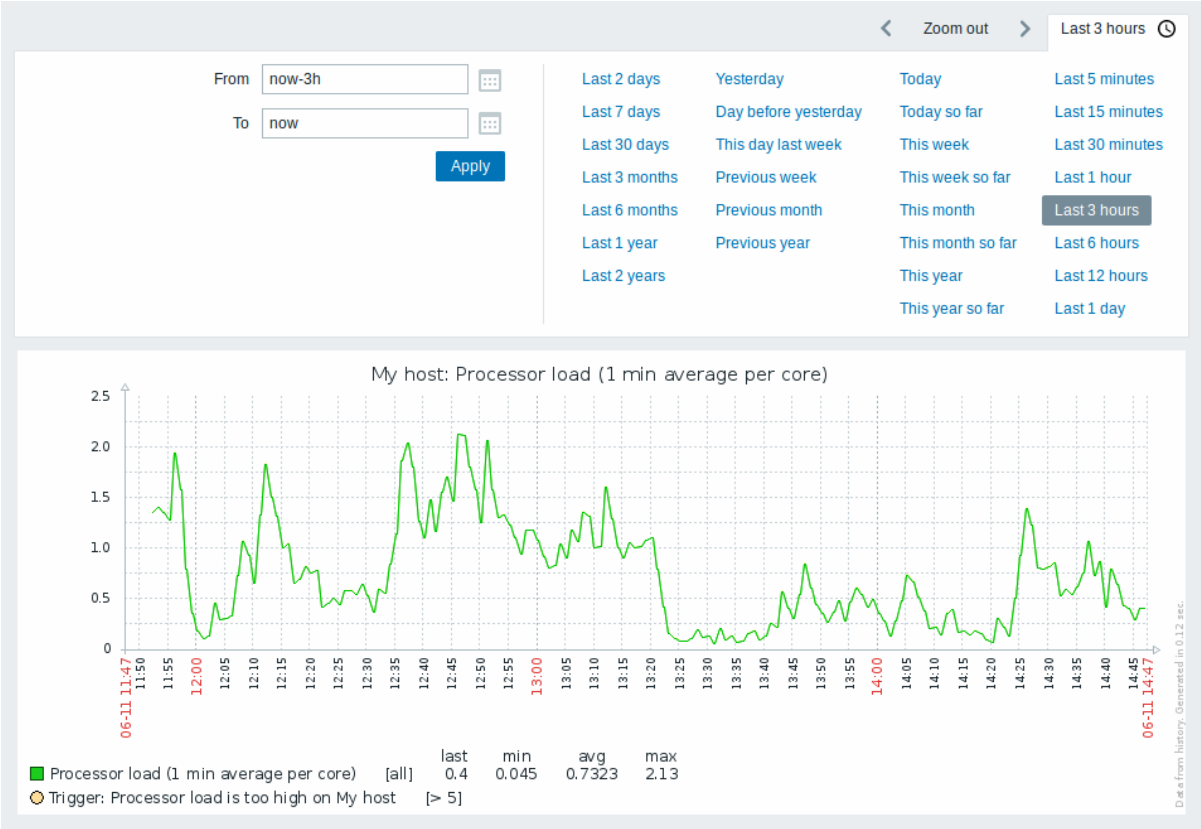
Einfache Diagramme

Übersicht

Einfache Diagramme dienen zur Visualisierung der von Datenpunkten erfassten Daten.

Für die Anzeige einfacher Diagramme ist keine Konfiguration durch den Benutzer erforderlich. Sie werden von Zabbix frei zur Verfügung gestellt.

Gehen Sie einfach zu *Monitoring* > *Latest data* und klicken Sie beim entsprechenden Datenpunkt auf den Link *Graph*. Anschließend wird ein Diagramm angezeigt.



Note:
Einfache Diagramme werden für alle numerischen Datenpunkte bereitgestellt. Für Textdatenpunkte ist unter *Monitoring* > *Latest data* ein Link zu *History* verfügbar.

Zeitraum-Auswahl

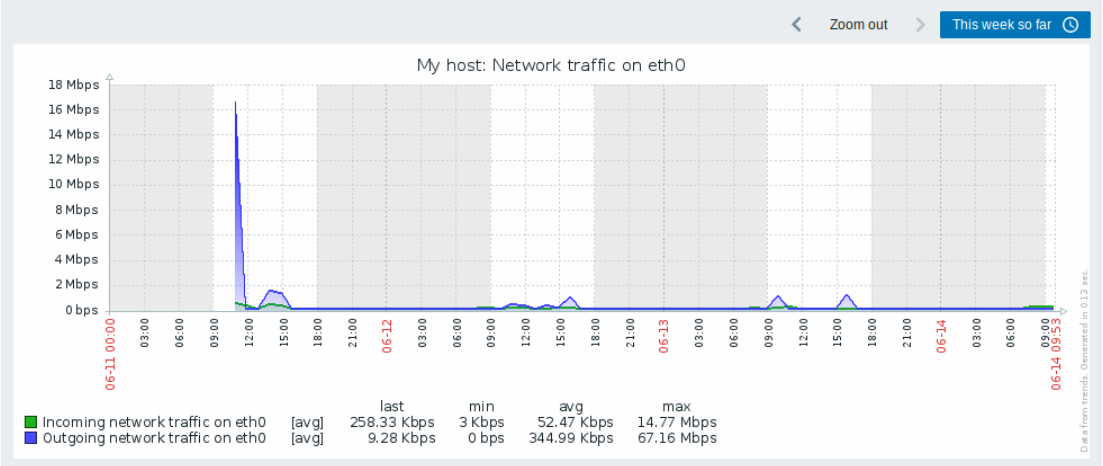
Mit der oben im Diagramm angezeigten Zeitraum-Auswahl können Sie häufig benötigte Zeiträume mit einem Mausklick auswählen. Weitere Informationen finden Sie unter Zeitraum- und Host-Auswahl.

Aktuelle Daten im Vergleich zu längeren Zeiträumen

Für sehr aktuelle Daten wird eine **einzelne** Linie gezeichnet, die jeden empfangenen Wert verbindet. Die einzelne Linie wird gezeichnet, solange mindestens ein horizontaler Pixel für einen Wert verfügbar ist.

Für Daten, die einen längeren Zeitraum anzeigen, werden **drei Linien** gezeichnet – eine dunkelgrüne zeigt den Durchschnitt, während eine hellrosa und eine hellgrüne Linie die Maximal- bzw. Minimalwerte zu diesem Zeitpunkt anzeigen. Der Bereich zwischen den Höchst- und Tiefstwerten wird mit einem gelben Hintergrund gefüllt.

Die Arbeitszeit (Arbeitstage) wird in Diagrammen mit weißem Hintergrund angezeigt, während die arbeitsfreie Zeit grau dargestellt wird (mit dem standardmäßigen Frontend-Theme *Original blue*).



Die Arbeitszeit wird in einfachen Diagrammen immer angezeigt, während ihre Anzeige in benutzerdefinierten Diagrammen von den Benutzereinstellungen abhängt.

Die Arbeitszeit wird nicht angezeigt, wenn das Diagramm mehr als 3 Monate umfasst.

Auslöser-Linien

Einfache Auslöser werden als Linien mit schwarzen Strichen über der Farbe des Auslöser-Schweregrads angezeigt - beachten Sie die blaue Linie im Diagramm und die in der Legende angezeigten Auslöser-Informationen. Im Diagramm können bis zu 3 Auslöser-Linien angezeigt werden; wenn es mehr Auslöser gibt, werden die Auslöser mit niedrigerem Schweregrad priorisiert. Auslöser werden in einfachen Diagrammen immer angezeigt, während ihre Anzeige in benutzerdefinierten Diagrammen eine Benutzereinstellung ist.



Generierung aus Verlauf/Trends

Diagramme können entweder auf Grundlage des Verlaufs oder der Trends eines Datenpunkts erstellt werden.

Für Benutzer, bei denen der Debug-Modus im Frontend aktiviert ist, wird unten rechts in einem Diagramm eine graue, vertikale Beschriftung angezeigt, die angibt, woher die Daten stammen.

Mehrere Faktoren beeinflussen, ob der Verlauf oder die Trends verwendet werden:

- Aufbewahrungsdauer des Verlaufs eines Datenpunkts. Der Verlauf eines Datenpunkts kann beispielsweise 14 Tage lang aufbewahrt werden. In diesem Fall stammen alle Daten, die älter als vierzehn Tage sind, aus den Trends.
- Datenmenge im Diagramm. Wenn die Anzahl der anzuzeigenden Sekunden pro horizontalem Diagrammpixel 3600/16 überschreitet, werden Trenddaten angezeigt (auch wenn der Verlauf des Datenpunkts für denselben Zeitraum noch verfügbar ist).
- Wenn Trends deaktiviert sind, wird der Verlauf des Datenpunkts zum Erstellen des Diagramms verwendet, sofern er für diesen Zeitraum verfügbar ist.

Fehlen von Daten

Bei Datenpunkten mit einem regulären Aktualisierungsintervall wird im Diagramm nichts angezeigt, wenn keine Daten für den Datenpunkt erfasst werden.

Bei Trapper-Datenpunkten und Datenpunkten mit einem geplanten Aktualisierungsintervall (und einem auf 0 gesetzten regulären Aktualisierungsintervall) wird jedoch eine gerade Linie bis zum ersten erfassten Wert und vom letzten erfassten Wert bis zum Ende des Diagramms gezeichnet; die Linie befindet sich jeweils auf der Höhe des ersten bzw. letzten Werts.

Zu Rohwerten wechseln

Über ein Dropdown-Menü oben rechts können Sie vom einfachen Diagramm zu den Listen *Werte/500 neueste Werte* wechseln. Dies kann hilfreich sein, um die numerischen Werte anzuzeigen, aus denen das Diagramm besteht.

Die hier dargestellten Werte sind Rohwerte, d. h., es werden weder Einheiten noch eine Nachbearbeitung der Werte verwendet. Die Wertzuordnung wird jedoch angewendet.

Bekannte Probleme

Siehe bekannte Probleme für Graphen.

Benutzerdefinierte Diagramme

Übersicht

Benutzerdefinierte Diagramme bieten, wie der Name schon sagt, Möglichkeiten zur Anpassung.

Einfache Diagramme eignen sich zwar gut zur Anzeige von Daten eines einzelnen Datenpunkts, bieten jedoch keine Konfigurationsmöglichkeiten.

Wenn Sie also beispielsweise den Diagrammstil oder die Art der Darstellung von Linien ändern oder mehrere Datenpunkte vergleichen möchten, etwa eingehenden und ausgehenden Datenverkehr in einem einzigen Diagramm, benötigen Sie ein benutzerdefiniertes Diagramm.

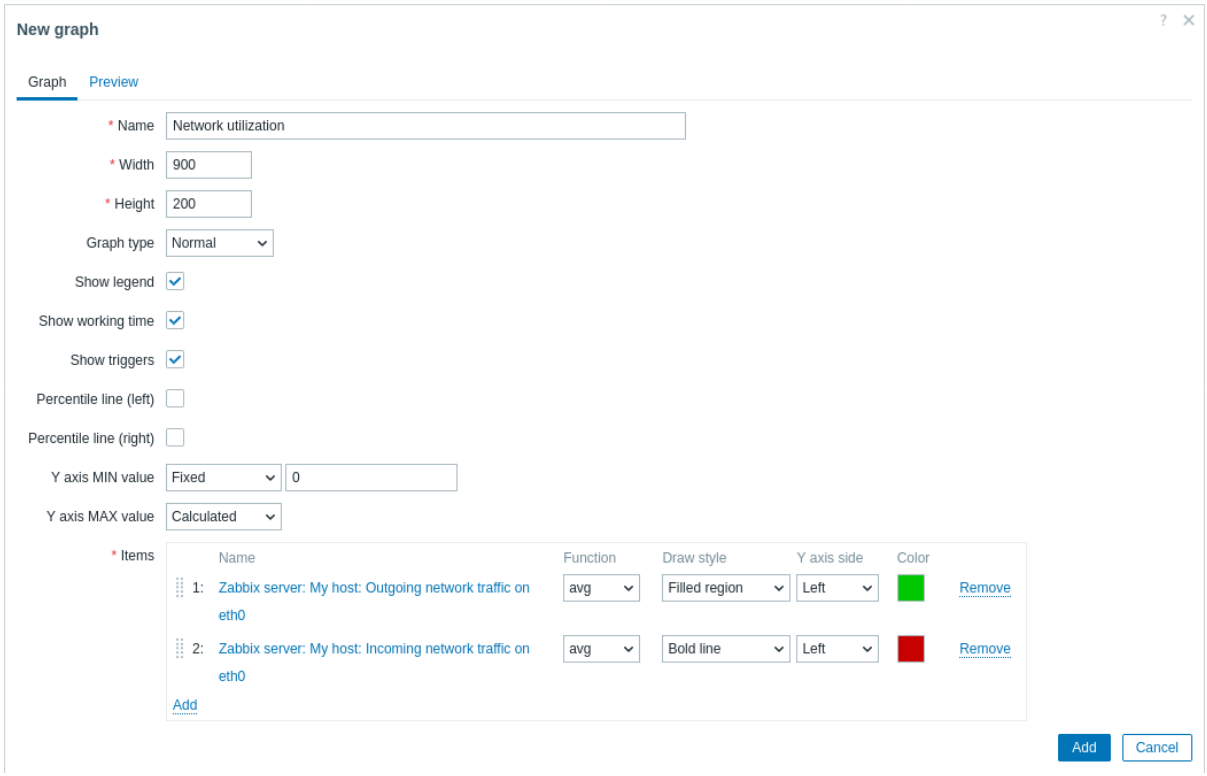
Benutzerdefinierte Diagramme werden manuell konfiguriert.

Sie können für einen Host oder mehrere Hosts oder für eine einzelne Vorlage erstellt werden.

Konfigurieren benutzerdefinierter Diagramme

Gehen Sie wie folgt vor, um ein benutzerdefiniertes Diagramm zu erstellen:

1. Wechseln Sie zu *Datenaufzeichnung* > *Hosts* (oder *Vorlagen*).
2. Klicken Sie in der Zeile neben dem gewünschten Host oder der gewünschten Vorlage auf *Diagramme*.
3. Klicken Sie im Bildschirm *Diagramme* auf *Diagramm erstellen*.
4. Bearbeiten Sie die Diagrammattribute.



Alle Pflichtfelder sind mit einem roten Sternchen markiert.

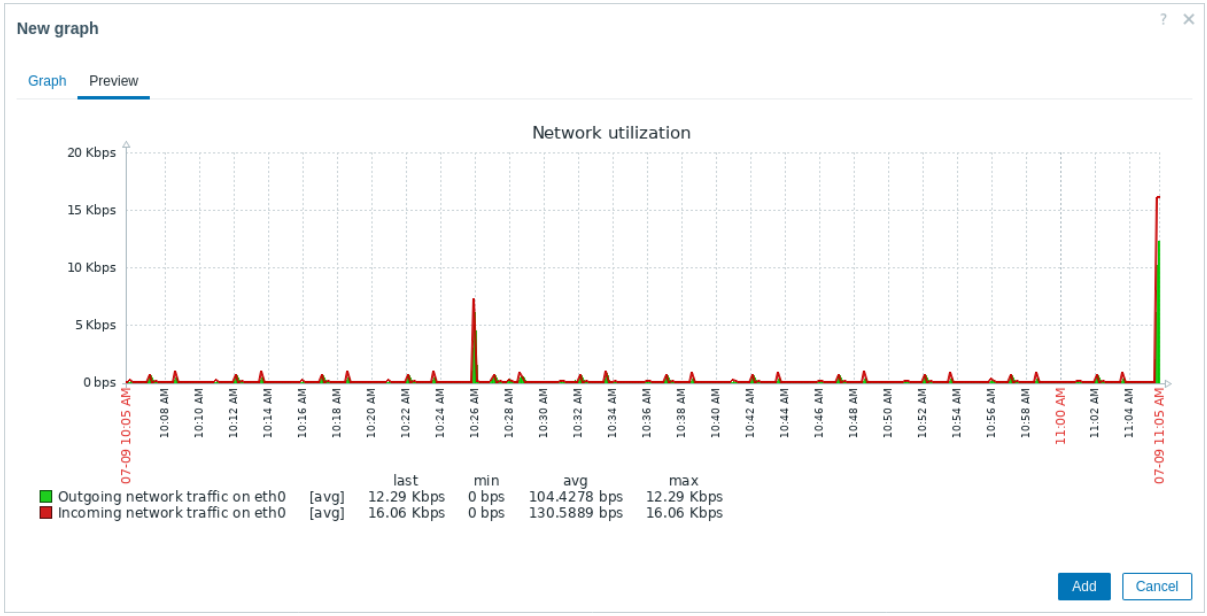
Diagrammattribute:

Parameter	Beschreibung
Name	Eindeutiger Name des Diagramms. Ausdrucks-Makros werden in diesem Feld unterstützt, jedoch nur mit den Funktionen `avg`, `last`, `min` und `max` sowie Zeit als Parameter (zum Beispiel `{?avg(/host/key,1h)}`). Makros vom Typ {HOST.HOST<1-9>} werden für die Verwendung innerhalb dieses Makros unterstützt und verweisen auf den ersten, zweiten, dritten usw. Host im Diagramm, zum Beispiel `{?avg(/{HOST.HOST2}/key,1h)}`. Beachten Sie, dass die Referenzierung des ersten Hosts mit diesem Makro überflüssig ist, da auf den ersten Host implizit verwiesen werden kann, zum Beispiel `{?avg(//key,1h)}`.
Width	Diagrammbreite in Pixeln (nur für Vorschau- sowie Kreis-/aufgeschnittene Kreisdiagramme).
Height	Diagrammhöhe in Pixeln.
Graph type	Diagrammtyp: **Normal** - normales Diagramm, Werte werden als Linien angezeigt **Stacked** - gestapeltes Diagramm, gefüllte Bereiche werden angezeigt **Pie** - Kreisdiagramm **Exploded** - "aufgeschnittenes" Kreisdiagramm, die Segmente werden als aus dem Kreis "herausgeschnitten" angezeigt
Show legend	Wenn dieses Kontrollkästchen aktiviert ist, wird die Diagrammlegende angezeigt.
Show working time	Wenn diese Option ausgewählt ist, werden Nichtarbeitszeiten mit einem grauen Hintergrund angezeigt. Dieser Parameter ist für Kreis- und aufgeschnittene Kreisdiagramme nicht verfügbar.
Show triggers	Wenn diese Option ausgewählt ist, werden einfache Auslöser als Linien mit schwarzen Strichen über der Schweregradfarbe des Auslösers angezeigt. Dieser Parameter ist für Kreis- und aufgeschnittene Kreisdiagramme nicht verfügbar.
Percentile line (left)	Perzentil für die linke Y-Achse anzeigen. Wenn beispielsweise das 95-%-Perzentil festgelegt ist, liegt die Perzentillinie auf dem Niveau, unter dem 95 Prozent der Werte liegen. Wird als hellgrüne Linie angezeigt. Nur für normale Diagramme verfügbar.
Percentile line (right)	Perzentil für die rechte Y-Achse anzeigen. Wenn beispielsweise das 95-%-Perzentil festgelegt ist, liegt die Perzentillinie auf dem Niveau, unter dem 95 Prozent der Werte liegen. Wird als hellrote Linie angezeigt. Nur für normale Diagramme verfügbar.
Y axis MIN value	Minimalwert der Y-Achse: **Calculated** - der Minimalwert der Y-Achse wird automatisch berechnet. **Fixed** - fester Minimalwert für die Y-Achse. **Item** - der letzte Wert des ausgewählten Datenpunkts wird der Minimalwert. Dieser Parameter ist für Kreis- und aufgeschnittene Kreisdiagramme nicht verfügbar.
Y axis MAX value	Maximalwert der Y-Achse: **Calculated** - der Maximalwert der Y-Achse wird automatisch berechnet. **Fixed** - fester Maximalwert für die Y-Achse. **Item** - der letzte Wert des ausgewählten Datenpunkts wird der Maximalwert. Dieser Parameter ist für Kreis- und aufgeschnittene Kreisdiagramme nicht verfügbar.
3D view	3D-Darstellung aktivieren. Nur für Kreis- und aufgeschnittene Kreisdiagramme.
Items	Datenpunkte, deren Daten in diesem Diagramm angezeigt werden sollen. Klicken Sie auf *Add*, um Datenpunkte auszuwählen. Sie können außerdem verschiedene Anzeigeoptionen auswählen (Funktion, Darstellungsstil, Anzeige auf linker/rechter Achse, Farbe).
Sort order (0→100)	Zeichenreihenfolge. 0 wird zuerst verarbeitet. Kann verwendet werden, um Linien oder Bereiche hinter (oder vor) einem anderen Element zu zeichnen. Sie können Elemente per Drag-and-drop mithilfe des Symbols am Anfang einer Zeile verschieben, um die Sortierreihenfolge oder festzulegen, welches Element vor dem anderen angezeigt wird.
Name	Der Name des ausgewählten Datenpunkts wird als Link angezeigt. Wenn Sie auf den Link klicken, wird die Liste der anderen verfügbaren Datenpunkte geöffnet.
Type	Typ (nur für Kreis- und aufgeschnittene Kreisdiagramme verfügbar): **Simple** - der Wert des Datenpunkts wird proportional im Kreis dargestellt **Graph sum** - der Wert des Datenpunkts stellt den gesamten Kreis dar Beachten Sie, dass die Färbung des Elements "graph sum" nur insoweit sichtbar ist, wie sie nicht von "proportionalen" Elementen belegt wird.

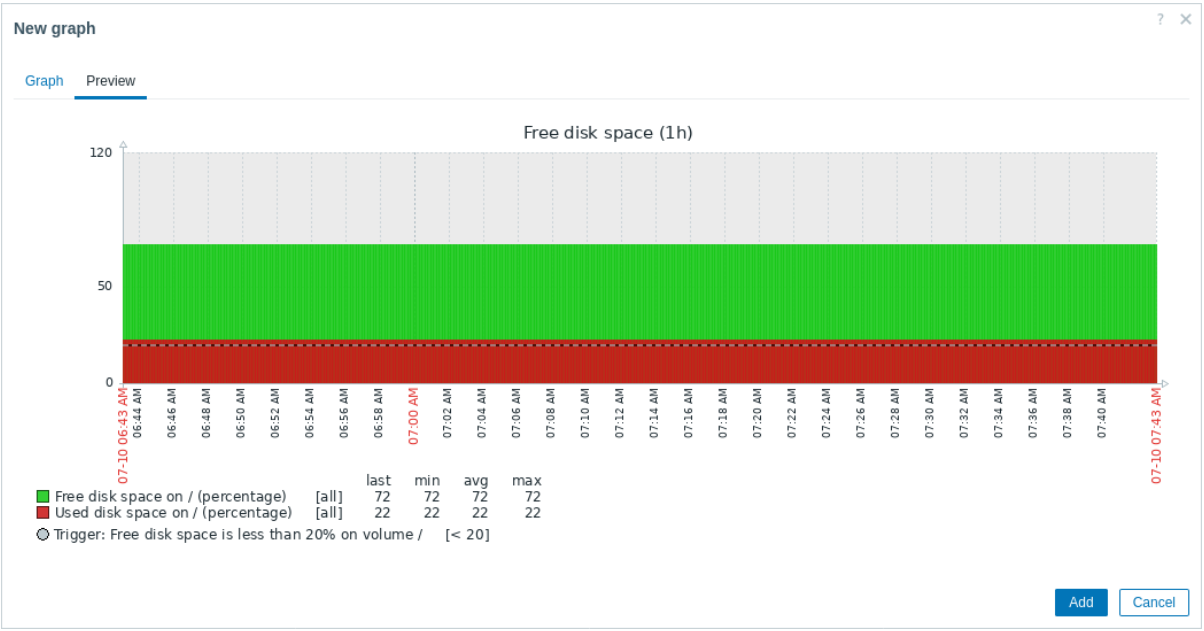
Parameter	Beschreibung
Function	Wählen Sie aus, welche Werte angezeigt werden sollen, wenn für einen Datenpunkt mehr als ein Wert pro vertikalem Diagrammpixel vorhanden ist: **all** - alle möglichen Werte (Minimum, Maximum, Durchschnitt) im Diagramm anzeigen. Beachten Sie, dass diese Einstellung bei kürzeren Zeiträumen keine Auswirkung hat; nur bei längeren Zeiträumen, wenn die Datendichte in einem vertikalen Diagrammpixel zunimmt, zeigt 'all' Minimum-, Maximum- und Durchschnittswerte an. Diese Funktion ist nur für den Diagrammtyp *Normal* verfügbar. Siehe auch: Diagramme aus Verlauf/Trends generieren. **avg** - Durchschnittswerte anzeigen **last** - die neuesten Werte anzeigen. Diese Funktion ist nur verfügbar, wenn als Diagrammtyp entweder *Pie/Exploded pie* ausgewählt ist. **max** - Maximalwerte anzeigen **min** - Minimalwerte anzeigen
Draw style	Wählen Sie den Darstellungsstil aus (nur für normale Diagramme verfügbar; bei gestapelten Diagrammen wird immer ein gefüllter Bereich verwendet), der auf die Daten des Datenpunkts angewendet werden soll - *Line*, *Bold line*, *Filled region*, *Dot*, *Dashed line*, *Gradient line*.
Y axis side	Wählen Sie die Seite der Y-Achse aus, auf der die Daten des Datenpunkts angezeigt werden sollen - *Left*, *Right*.
Color	Wählen Sie die Farbe aus, die auf die Daten des Datenpunkts angewendet werden soll.

Graph-Vorschau

Auf der Registerkarte *Vorschau* wird eine Vorschau des Graphen angezeigt, sodass Sie sofort sehen können, was Sie erstellen.



Beachten Sie, dass in der Vorschau für Datenpunkte aus Vorlagen keine Daten angezeigt werden.



Achten Sie in diesem Beispiel auf die gestrichelte, dicke Linie, die die Auslöser-Schwelle anzeigt, sowie auf die in der Legende angezeigten Auslöser-Informationen.

Note:
Es können nicht mehr als 3 Auslöser-Linien angezeigt werden. Wenn es mehr Auslöser gibt, werden die Auslöser mit niedrigerem Schweregrad bevorzugt angezeigt.

 Wenn die Graph-Höhe auf weniger als 120 Pixel festgelegt ist, wird in der Legende kein Auslöser angezeigt.

Ad-hoc-Diagramme

Übersicht

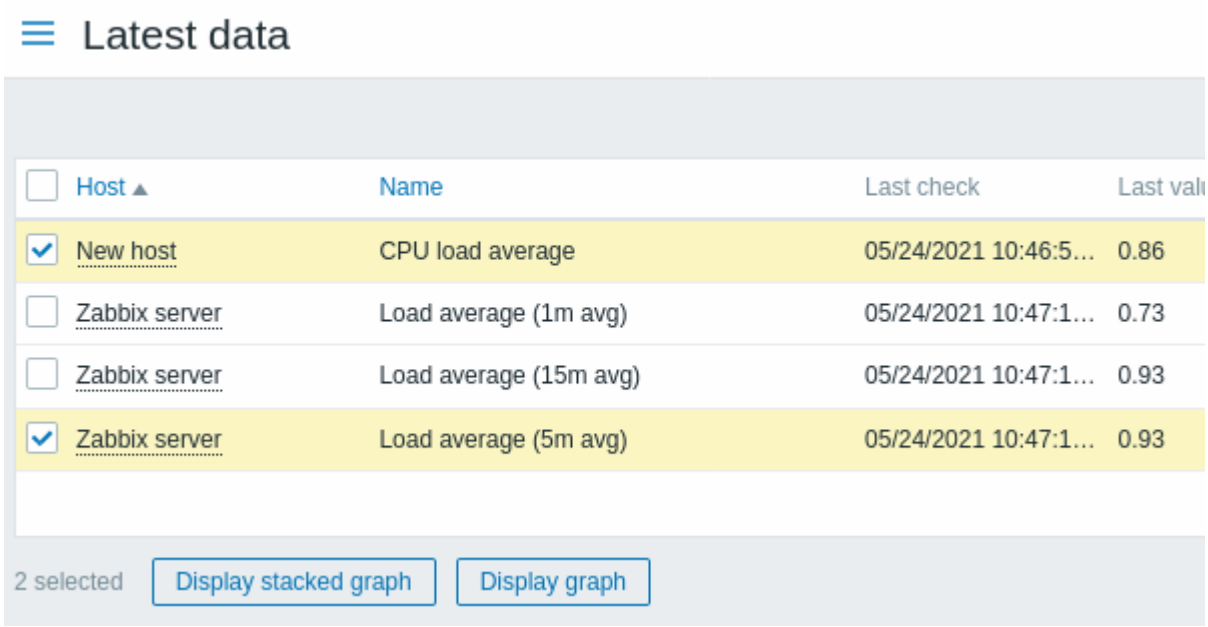
Während ein einfaches Diagramm nützlich ist, um Daten aus einem einzelnen Datenpunkt anzuzeigen, und benutzerdefinierte Diagramme Anpassungsmöglichkeiten bieten, ermöglicht keines von beiden, schnell ein Vergleichsdiagramm für mehrere Datenpunkte mit minimalem Aufwand und Wartungsaufwand zu erstellen.

Um dieses Problem zu lösen, können Ad-hoc-Diagramme für mehrere Datenpunkte sehr schnell erstellt werden.

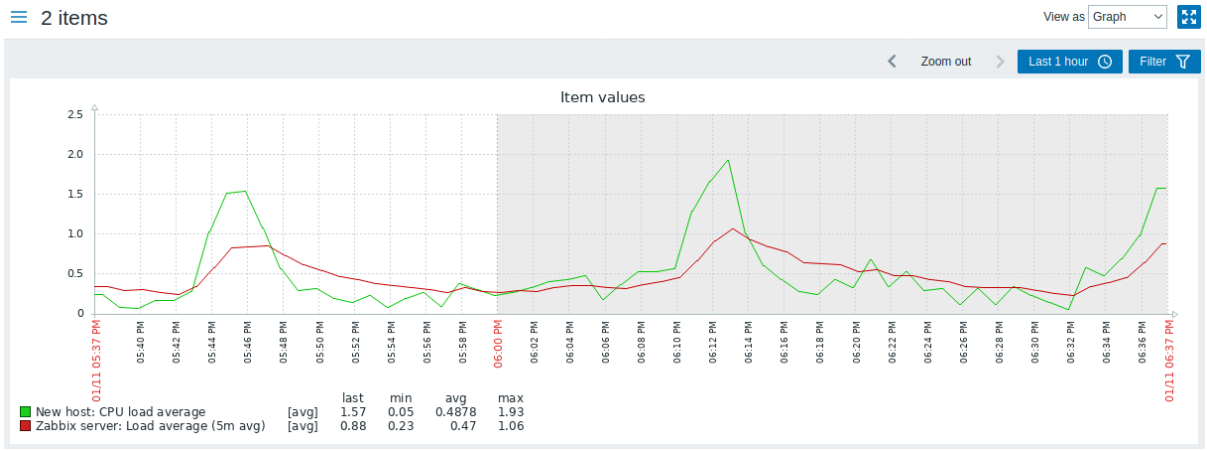
Konfiguration

Gehen Sie wie folgt vor, um ein Ad-hoc-Diagramm zu erstellen:

1. Gehen Sie zu *Monitoring > Neueste Daten*.
2. Verwenden Sie den Filter, um die gewünschten Datenpunkte anzuzeigen.
3. Markieren Sie die Kontrollkästchen der Datenpunkte, die Sie im Diagramm anzeigen möchten.
4. Klicken Sie auf die Schaltflächen *Gestapeltes Diagramm anzeigen* oder *Diagramm anzeigen*.

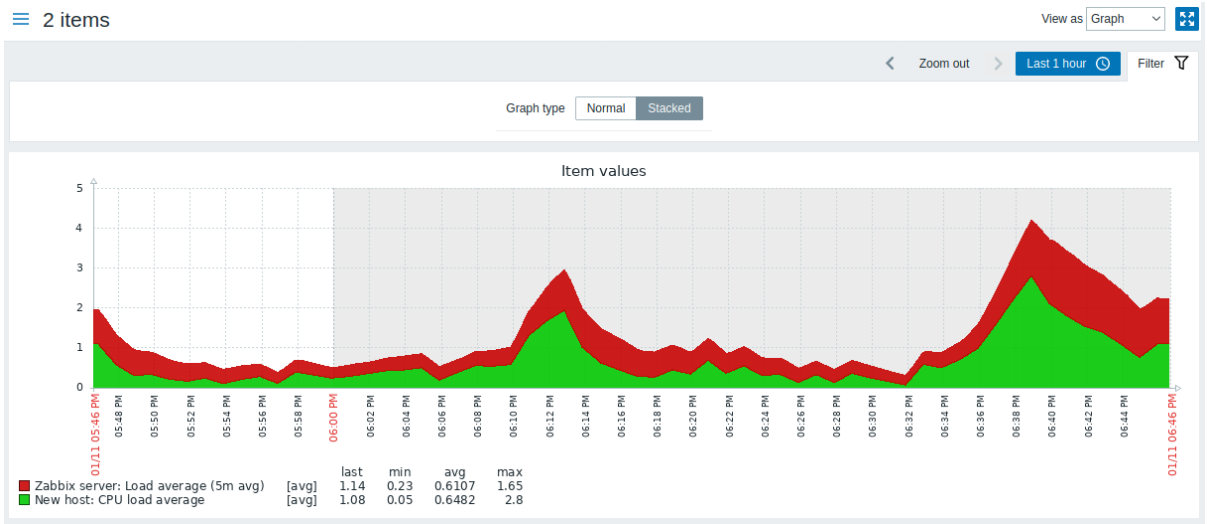


Ihr Diagramm wird sofort erstellt:



Beachten Sie, dass zur Vermeidung einer zu großen Anzahl von Linien im Diagramm nur der Durchschnittswert für jeden Datenpunkt angezeigt wird (Min-/Max-Wert-Linien werden nicht angezeigt). Auslöser und Auslöserinformationen werden im Diagramm nicht angezeigt.

Im Fenster des erstellten Diagramms steht Ihnen der Zeitraumselektor zur Verfügung, und Sie können vom ”normalen” Liniendiagramm zu einem gestapelten Diagramm wechseln und umgekehrt.



Netzwerkkarten

Übersicht

Wenn Sie ein Netzwerk betreuen, möchten Sie möglicherweise irgendwo einen Überblick über Ihre Infrastruktur haben. Zu diesem Zweck können Sie in Zabbix Karten erstellen – von Netzwerken und von allem, was Sie möchten.

Alle Benutzer können Netzwerkkarten erstellen. Die Karten können öffentlich sein (für alle Benutzer verfügbar) oder privat (für ausgewählte Benutzer verfügbar).

Fahren Sie mit der Konfiguration einer Netzwerkkarte fort.

Konfigurieren einer Netzwerkkarte

Übersicht

Das Konfigurieren einer Karte in Zabbix erfordert, dass Sie zunächst eine Karte erstellen, indem Sie ihre allgemeinen Parameter definieren, und anschließend die eigentliche Karte mit Elementen und deren Verknüpfungen füllen.

Sie können die Karte mit Elementen befüllen, die ein Host, eine Hostgruppe, ein Auslöser, ein Bild oder eine andere Karte sind.

Symbole werden verwendet, um Kartenelemente darzustellen. Sie können die Informationen definieren, die zusammen mit den Symbolen angezeigt werden, und festlegen, dass aktuelle Probleme auf besondere Weise dargestellt werden. Sie können die Symbole verknüpfen und Informationen definieren, die auf den Verknüpfungen angezeigt werden.

Sie können benutzerdefinierte URLs hinzufügen, die durch Klicken auf die Symbole aufgerufen werden können. So können Sie beispielsweise ein Host-Symbol mit den Host-Eigenschaften oder ein Kartensymbol mit einer anderen Karte verknüpfen.

Die Problemanzahl in Karten wird nur für Ursache-Probleme angezeigt.

Karten werden unter *Monitoring* > *Maps* verwaltet, wo sie konfiguriert, verwaltet und angezeigt werden können. In der Monitoring-Ansicht können Sie auf die Symbole klicken und die Verknüpfungen zu einigen Skripten und URLs nutzen.

Netzwerkkarten basieren auf Vektorgrafiken (SVG).

Öffentliche und private Karten

Alle Benutzer in Zabbix (einschließlich Nicht-Admin-Benutzern) können Netzwerkkarten erstellen. Karten haben einen Eigentümer – den Benutzer, der sie erstellt hat. Karten können öffentlich oder privat sein.

- *Öffentliche* Karten sind für alle Benutzer sichtbar. Um sie jedoch sehen zu können, muss der Benutzer mindestens Leserechte für ein Kartenelement haben. Öffentliche Karten können bearbeitet werden, wenn ein Benutzer/eine Benutzergruppe Lese-/Schreibrechte für diese Karte und mindestens Leserechte für alle Elemente der entsprechenden Karte hat, einschließlich Auslösern in den Verknüpfungen.
- *Private* Karten sind nur für ihren Eigentümer und die Benutzer/Benutzergruppen sichtbar, für die die Karte vom Eigentümer freigegeben wurde. Normale Benutzer (keine Super-Admins) können Karten nur für die Gruppen freigeben, denen sie angehören, sowie für Benutzer, die diesen Gruppen angehören. Benutzer mit Admin-Rechten können private Karten sehen, unabhängig davon, ob sie Eigentümer sind oder zur Liste der freigegebenen Benutzer gehören. Private Karten können vom Eigentümer der Karte bearbeitet werden sowie dann, wenn ein Benutzer/eine Benutzergruppe Lese-/Schreibrechte für diese Karte und mindestens Leserechte für alle Elemente der entsprechenden Karte hat, einschließlich Auslösern in den Verknüpfungen.

Kartenelemente, für die der Benutzer keine Leserechte hat, werden mit einem ausgegrauten Symbol angezeigt, und alle Textinformationen des Elements werden ausgeblendet. Die Beschriftung des Auslösers ist jedoch sichtbar, auch wenn der Benutzer keine Berechtigung für den Auslöser hat.

Um ein Element zur Karte hinzuzufügen, muss der Benutzer ebenfalls mindestens Leserechte für das Element haben.

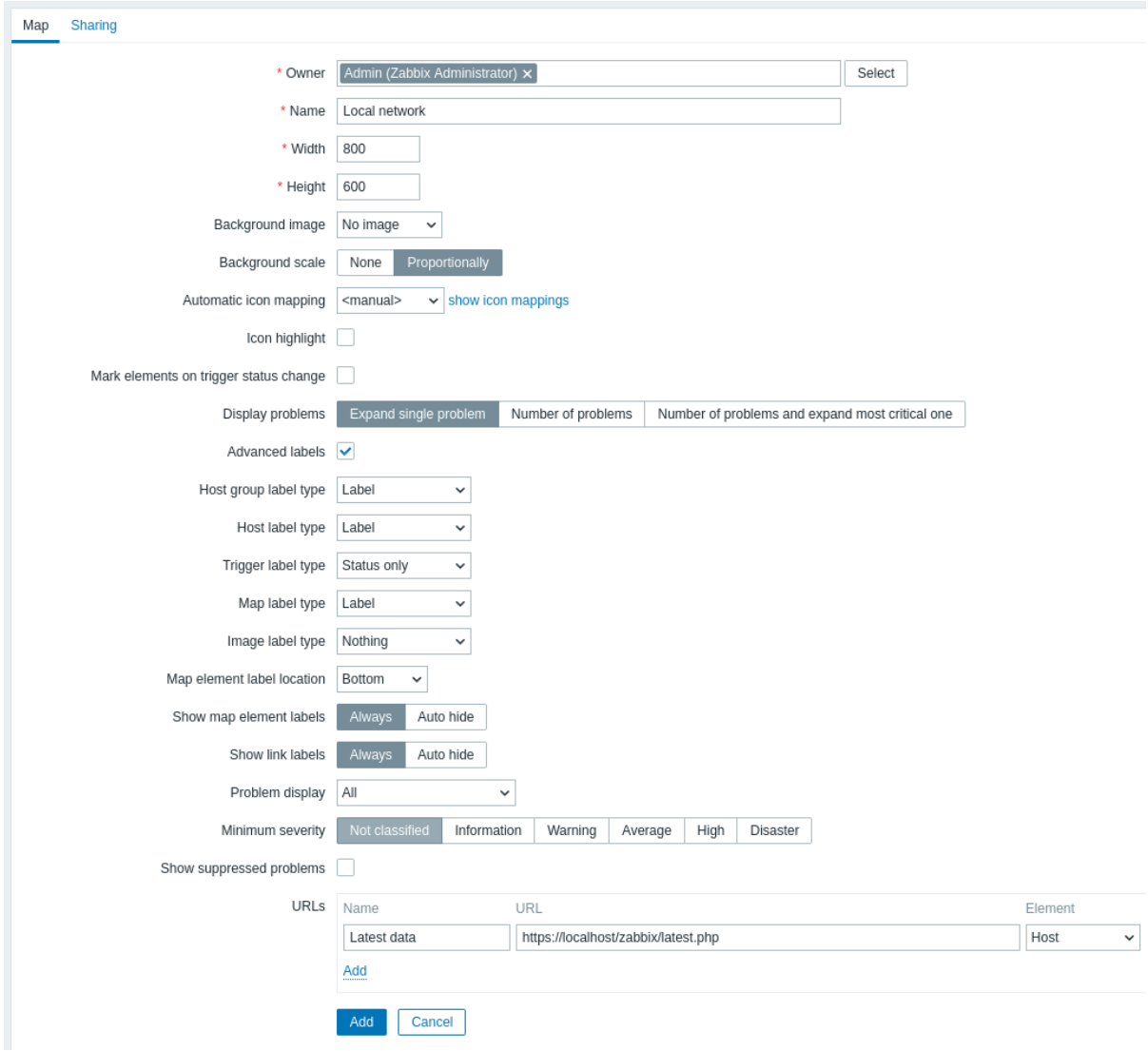
Erstellen einer Karte

Gehen Sie zum Erstellen einer Karte wie folgt vor:

1. Gehen Sie zu *Monitoring* > *Maps*.
2. Öffnen Sie die Ansicht mit allen Karten.
3. Klicken Sie auf *Create map*.
4. Konfigurieren Sie die Kartendetails.

Sie können auch die Schaltfläche *Clone* im Konfigurationsformular einer vorhandenen Karte verwenden, um eine neue Karte zu erstellen. Diese Karte verfügt über alle Eigenschaften der vorhandenen Karte, einschließlich der allgemeinen Layoutattribute sowie der Elemente der vorhandenen Karte.

Die Registerkarte **Map** enthält allgemeine Kartenattribute:



Alle Pflichtfelder sind mit einem roten Sternchen gekennzeichnet.

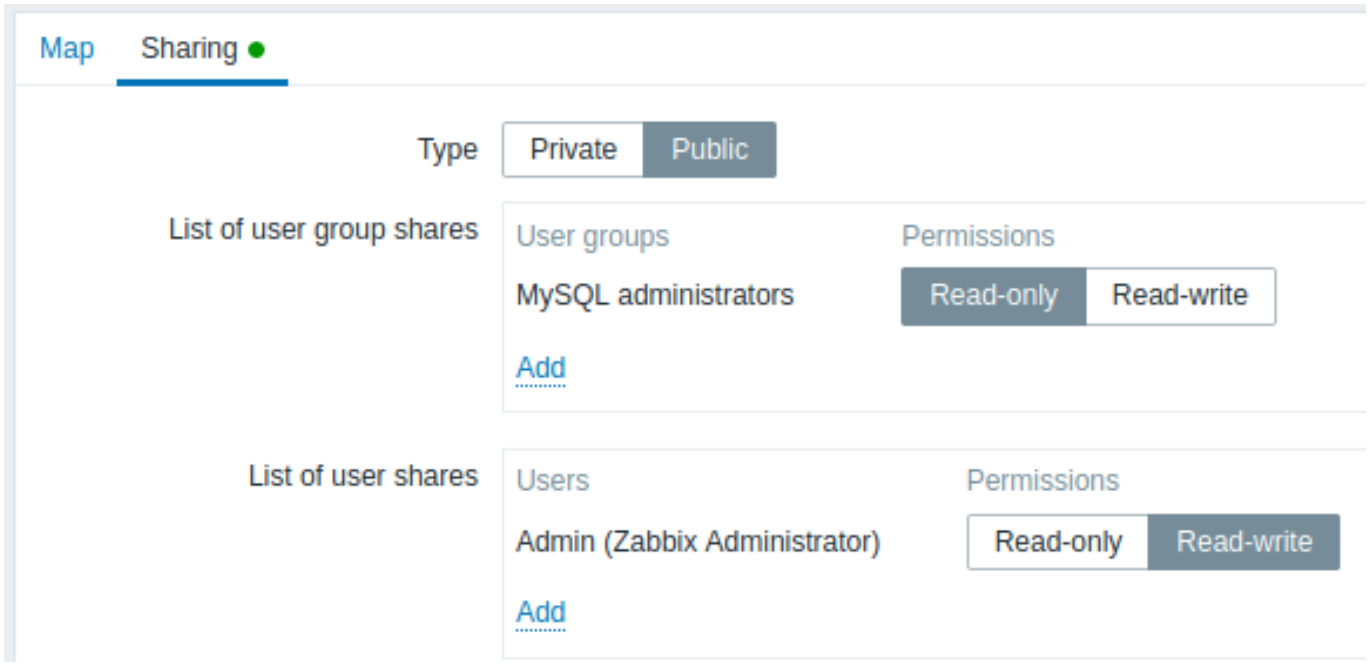
Allgemeine Kartenattribute:

Parameter	Beschreibung
Owner	Name des Kartenbesitzers.
Name	Eindeutiger Kartenname.
Width	Kartenbreite in Pixeln.
Height	Kartenhöhe in Pixeln.
Background image	Hintergrundbild verwenden: **No image** – kein Hintergrundbild (weißer Hintergrund) **Image** – ausgewähltes Bild, das als Hintergrundbild verwendet wird. Sie können eine geografische Karte oder ein beliebiges anderes Bild verwenden, um Ihre Karte zu erweitern.
Background scale	Hintergrundskalierung verwenden: **None** – keine Skalierung; **Proportionally** – den gesamten Kartenhintergrund abdecken, dabei jedoch die Bildproportionen beibehalten (Standard).
Automatic icon mapping	Sie können eine automatische Symbolzuordnung verwenden, die unter *Administration* > *General* > *Icon mapping* konfiguriert wird. Mit der Symbolzuordnung können bestimmte Symbole bestimmten Inventarfeldern eines Hosts zugeordnet werden.
Icon highlight	Wenn Sie dieses Kontrollkästchen aktivieren, werden Kartenelemente hervorgehoben. Elemente mit einem aktiven Auslöser erhalten einen runden Hintergrund in derselben Farbe wie der Auslöser mit dem höchsten Schweregrad. Außerdem wird ein dicker grüner Rand um den Kreis angezeigt, wenn alle Probleme bestätigt wurden. Elemente mit dem Status „disabled“ oder „in maintenance“ erhalten einen quadratischen Hintergrund, der jeweils grau bzw. orange ist. Siehe auch: Karten anzeigen

Parameter	Beschreibung
Mark elements on trigger status change	Eine kürzlich erfolgte Änderung des Auslöserstatus (neues Problem oder Behebung) wird durch Markierungen (nach innen zeigende rote Dreiecke) an den drei Seiten des Elementsymbols hervorgehoben, die nicht vom Label belegt sind. Die Markierungen werden 30 Minuten lang angezeigt.
Display problems	Wählen Sie aus, wie Probleme mit einem Kartenelement angezeigt werden: **Expand single problem** – wenn nur ein Problem vorliegt, wird der Problemname angezeigt. Andernfalls wird die Gesamtzahl der Probleme angezeigt. **Number of problems** – die Gesamtzahl der Probleme wird angezeigt **Number of problems and expand most critical one** – der Name des kritischsten Problems und die Gesamtzahl der Probleme werden angezeigt. „Kritischstes“ wird anhand des Problemschweregrads und, bei Gleichstand, anhand der Problemereignis-ID bestimmt (höhere ID bzw. späteres Problem wird zuerst angezeigt). Bei einem *trigger map element* wird dies anhand des Problemschweregrads und, bei Gleichstand, anhand der Position des Auslösers in der Auslöserliste bestimmt. Bei mehreren Problemen desselben Auslösers wird das aktuellste angezeigt.
Advanced labels	Wenn Sie dieses Kontrollkästchen aktivieren, können Sie separate Labeltypen für verschiedene Elementtypen definieren.
Host group label type	Für das Kartenelement verwendeter Labeltyp: **Label** – Label des Kartenelements **IP address** – IP-Adresse **Element name** – Elementname (z. B. Hostname) **Status only** – nur Status (OK oder PROBLEM) **Nothing** – es werden keine Labels angezeigt
Host label type *Trigger label type* *Map label type* *Image label type*	
Map element label location	Position des Labels relativ zum Kartenelement: **Bottom** – unterhalb des Kartenelements **Left** – links davon **Right** – rechts davon **Top** – oberhalb des Kartenelements
Show map element labels	Wählen Sie aus, wie Labels von Kartenelementen angezeigt werden: **Always** – immer angezeigt (Standard); **Auto hide** – Label ausblenden, wenn der Mauszeiger nicht darüber bewegt wird und es nicht ausgewählt ist.
Show link labels	Wählen Sie aus, wie Linklabels angezeigt werden: **Always** – immer angezeigt (Standard); **Auto hide** – Label ausblenden, wenn der Mauszeiger nicht darüber bewegt wird und es nicht ausgewählt ist.
Problem display	Problemanzahl anzeigen als: **All** – die vollständige Problemanzahl wird angezeigt **Separated** – die Anzahl der nicht bestätigten Probleme wird getrennt als Teil der Gesamtproblemanzahl angezeigt **Unacknowledged only** – nur die Anzahl der nicht bestätigten Probleme wird angezeigt
Minimum severity	Probleme unterhalb des ausgewählten Mindestschweregrads werden nicht auf der Karte angezeigt. Wenn beispielsweise *Warning* ausgewählt ist, werden Änderungen mit Auslösern des Schweregrads *Information* und *Not classified* nicht auf der Karte dargestellt.
Show suppressed problems	Aktivieren Sie das Kontrollkästchen, um Probleme anzuzeigen, die andernfalls aufgrund der Wartung des Hosts unterdrückt (nicht angezeigt) würden.
URLs	Hier können URLs für jeden Elementtyp definiert werden, wobei: **Name** – ein Label für die URL; **URL** – eine URL (bis zu 2048 Zeichen); **Element** – der Elementtyp (z. B. *Host*, *Host group* usw.). Diese werden als Links angezeigt, wenn ein Benutzer im Kartenanzeigemodus auf das Element klickt. In Namen und Werten von Karten-URLs können Makros verwendet werden. Eine vollständige Liste finden Sie unter unterstützte Makros; suchen Sie dort nach „map URL names and values“.

Freigabe

Die Registerkarte **Freigabe** enthält den Kartentyp sowie Freigabeoptionen (Benutzergruppen, Benutzer) für private Karten:



Parameter	Description
Type	Kartentyp auswählen: **Private** - die Karte ist nur für ausgewählte Benutzergruppen und Benutzer sichtbar **Public** - die Karte ist für alle sichtbar
List of user group shares	Benutzergruppen auswählen, für die die Karte zugänglich ist. Sie können Lesezugriff oder Lese- und Schreibzugriff zulassen.
List of user shares	Benutzer auswählen, für die die Karte zugänglich ist. Sie können Lesezugriff oder Lese- und Schreibzugriff zulassen.

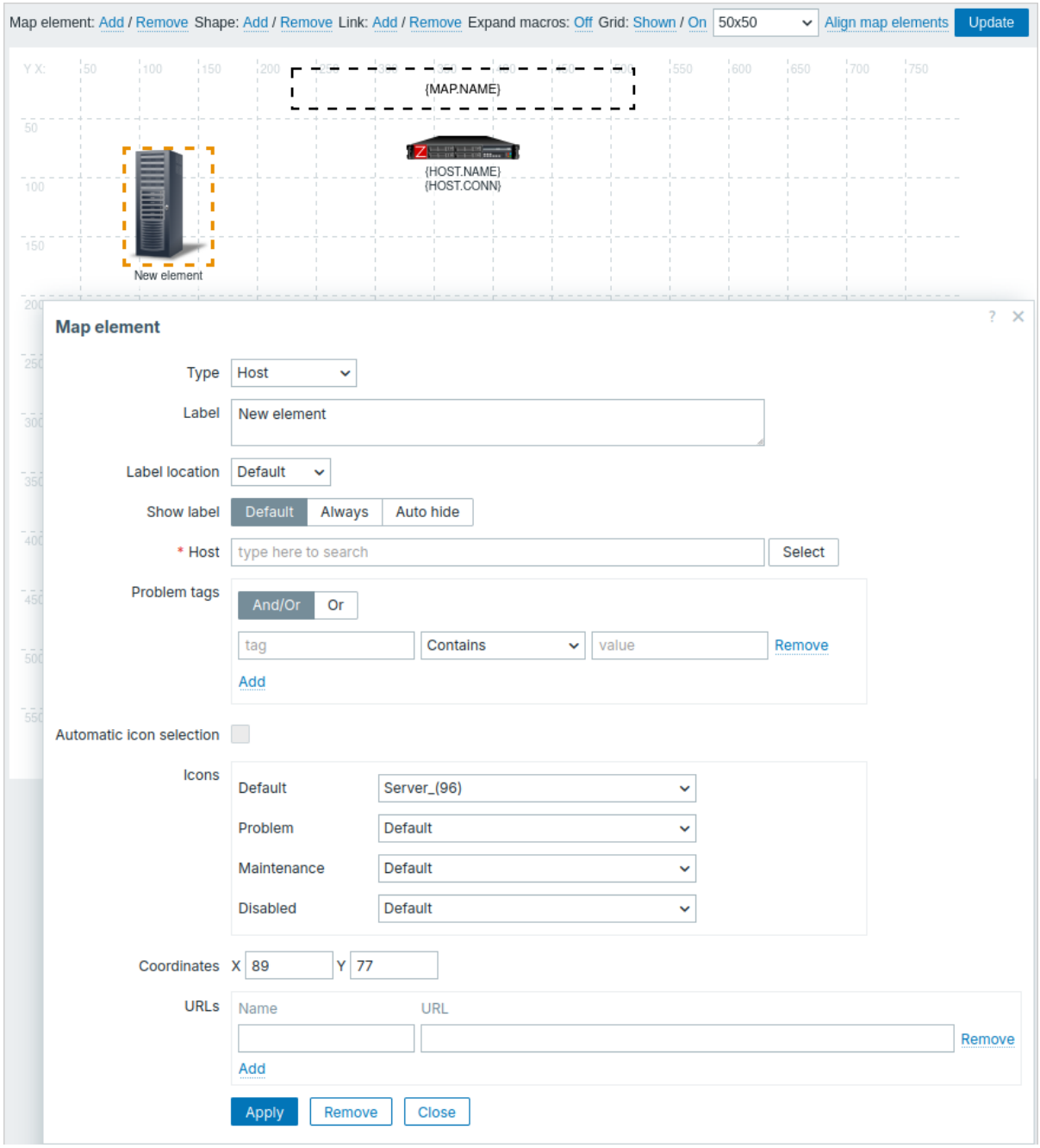
Wenn Sie auf *Add* klicken, um diese Karte zu speichern, haben Sie eine leere Karte mit einem Namen, Abmessungen und bestimmten Einstellungen erstellt. Jetzt müssen Sie einige Elemente hinzufügen. Klicken Sie dazu in der Kartenliste auf *Edit*, um den bearbeitbaren Bereich zu öffnen.

Hinzufügen von Elementen

Um ein Element hinzuzufügen, klicken Sie neben `Map element` auf *Add*. Das neue Element wird in der oberen linken Ecke der Karte angezeigt. Ziehen Sie es per Drag-and-drop an die gewünschte Stelle.

Beachten Sie, dass bei aktivierter Raster-Option ”On” Elemente immer am Raster ausgerichtet werden (Sie können im Dropdown verschiedene Rastergrößen auswählen und das Raster auch ein- oder ausblenden). Wenn Sie Elemente beliebig ohne Ausrichtung platzieren möchten, stellen Sie die Option auf ”Off”. (Sie können zufällig platzierte Elemente später durch Klicken auf *Align map elements* am Raster ausrichten.)

Nachdem Sie einige Elemente platziert haben, möchten Sie sie möglicherweise durch Namen usw. unterscheiden. Wenn Sie auf das Element klicken, wird ein Formular angezeigt, in dem Sie den Elementtyp festlegen, einen Namen vergeben, ein anderes Symbol auswählen usw. können.



Attribute von Kartenelementen:

Parameter	Beschreibung
Type	Typ des Elements: **Host** - Symbol, das den Status aller Auslöser des ausgewählten Hosts darstellt **Map** - Symbol, das den Status aller Elemente einer Karte darstellt **Trigger** - Symbol, das den Status eines oder mehrerer Auslöser darstellt **Host group** - Symbol, das den Status aller Auslöser aller Hosts darstellt, die zur ausgewählten Gruppe gehören **Image** - ein Symbol, das mit keiner Ressource verknüpft ist
Label	Symbolbeschriftung, beliebige Zeichenfolge. Makros und mehrzeilige Zeichenfolgen können verwendet werden. Ausdrucks-Makros werden in diesem Feld unterstützt, jedoch nur mit den Funktionen `avg`, `last`, `min` und `max` sowie Zeit als Parameter (zum Beispiel `{?avg(/host/key,1h)}`). Eine vollständige Liste der unterstützten Makros finden Sie unter unterstützte Makros; suchen Sie nach "map element labels".

Parameter	Beschreibung
Label location	Position der Beschriftung in Bezug auf das Symbol: **Default** - Standardposition der Kartenbeschriftung; **Bottom** - unterhalb des Symbols; **Left** - links; **Right** - rechts; **Top** - oberhalb des Symbols.
Show label	Wählen Sie aus, wie die Elementbeschriftung angezeigt werden soll: **Default** - die Einstellungen *Show map element labels* aus der Kartenkonfiguration verwenden (Standard); **Always** - immer anzeigen (Standard); **Auto hide** - die Beschriftung ausblenden, wenn sie nicht mit der Maus überfahren oder ausgewählt wird.
Host	Wählen Sie einen Host aus oder geben Sie alternativ den Hostnamen ein. Dieses Feld unterstützt Auto-Vervollständigung, sodass beim Tippen des Hostnamens eine Dropdown-Liste mit passenden Hosts angezeigt wird. Scrollen Sie nach unten, um einen auszuwählen. Klicken Sie auf das Entfernen-Symbol (x), um die Auswahl zu löschen. Dieses Feld ist nur für Elemente vom Typ *Host* verfügbar.
Map	Wählen Sie eine Karte aus oder geben Sie alternativ den Kartennamen ein. Dieses Feld unterstützt Auto-Vervollständigung, sodass beim Tippen des Kartennamens eine Dropdown-Liste mit passenden Karten angezeigt wird. Scrollen Sie nach unten, um eine auszuwählen. Klicken Sie auf das Entfernen-Symbol (x), um die Auswahl zu löschen. Dieses Feld ist nur für Elemente vom Typ *Map* verfügbar.
Triggers	Wählen Sie im Feld *New triggers* unten einen oder mehrere Auslöser aus und klicken Sie auf *Add*. Alternativ können Sie mit der Eingabe des Namens eines Auslösers beginnen und einen passenden Eintrag aus der Dropdown-Liste auswählen. Klicken Sie auf das Entfernen-Symbol (x), um die Auswahl zu löschen. Dieses Feld ist nur für Elemente vom Typ *Trigger* verfügbar. Die Reihenfolge der ausgewählten Auslöser kann geändert werden, jedoch nur innerhalb derselben Schweregrade. Die Auswahl mehrerer Auslöser wirkt sich auch auf die Auflösung von {HOST.*}-Makros sowohl im Bearbeitungs- als auch im Ansichtsmodus aus. Im Bearbeitungsmodus werden die zuerst angezeigten {HOST.*}-Makros abhängig vom ersten Auslöser in der Liste aufgelöst (basierend auf dem Schweregrad des Auslösers). Der Ansichtsmodus hängt vom Parameter Display problems in den allgemeinen Kartenattributen ab: - Wenn der Modus *Expand single problem* gewählt ist, werden die zuerst angezeigten {HOST.*}-Makros abhängig vom zuletzt erkannten Problem-Auslöser aufgelöst (unabhängig vom Schweregrad) oder vom ersten Auslöser in der Liste (falls kein Problem erkannt wurde). - Wenn der Modus *Number of problems and expand most critical one* gewählt ist, werden die zuerst angezeigten {HOST.*}-Makros abhängig vom Schweregrad des Auslösers aufgelöst.
Host group	Wählen Sie eine Hostgruppe aus oder geben Sie alternativ die Hostgruppe ein. Dieses Feld unterstützt Auto-Vervollständigung, sodass beim Tippen des Namens einer Hostgruppe eine Dropdown-Liste mit passenden Hostgruppen angezeigt wird. Scrollen Sie nach unten, um eine auszuwählen. Klicken Sie auf das Entfernen-Symbol (x), um die Auswahl zu löschen. Dieses Feld ist nur für Elemente vom Typ *Host group* verfügbar.

Parameter	Beschreibung
Problem tags	Geben Sie Tags an, um die Anzahl der im Widget angezeigten Probleme zu begrenzen. Es ist möglich, bestimmte Tags und Tag-Werte einzuschließen oder auszuschließen. Es können mehrere Bedingungen festgelegt werden. Die Übereinstimmung des Tag-Namens ist immer groß-/kleinschreibungssensitiv. Für jede Bedingung stehen mehrere Operatoren zur Verfügung: **Exists** - die angegebenen Tag-Namen einschließen **Equals** - die angegebenen Tag-Namen und Werte einschließen (groß-/kleinschreibungssensitiv) **Contains** - die angegebenen Tag-Namen einschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv) **Does not exist** - die angegebenen Tag-Namen ausschließen **Does not equal** - die angegebenen Tag-Namen und Werte ausschließen (groß-/kleinschreibungssensitiv) **Does not contain** - die angegebenen Tag-Namen ausschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv) Für Bedingungen gibt es zwei Berechnungstypen: **And/Or** - alle Bedingungen müssen erfüllt sein, Bedingungen mit demselben Tag-Namen werden durch die Or-Bedingung gruppiert **Or** - es genügt, wenn eine Bedingung erfüllt ist Dieses Feld ist nur für Elemente vom Typ *Host* und *Host group* verfügbar.
Automatic icon selection	Wenn das Kontrollkästchen aktiviert ist, wird die Symbolzuordnung verwendet, um festzulegen, welches Symbol angezeigt wird.
Icons	Sie können für das Element in den folgenden Fällen unterschiedliche Symbole anzeigen lassen: *Default*, *Problem*, *Maintenance* und *Disabled*. Dieses Feld ist nur für Elemente vom Typ *Host* verfügbar.
Coordinates	**X**- und **Y**-Koordinaten des Kartenelements.
URLs	Hier können elementspezifische URLs für das Element festgelegt werden, wobei gilt: **Name** - eine Beschriftung für die URL; **URL** - eine URL (bis zu 2048 Zeichen). Diese werden als Links angezeigt, wenn ein Benutzer im Ansichtsmodus der Karte auf das Element klickt. Wenn das Element eigene URLs hat und für seinen Typ auf Kartenebene definierte URLs vorhanden sind, werden diese im selben Menü zusammengeführt. Makros können in Namen und Werten von Kartenelementen verwendet werden. Eine vollständige Liste finden Sie unter unterstützte Makros; suchen Sie nach ”map URL names and values”.

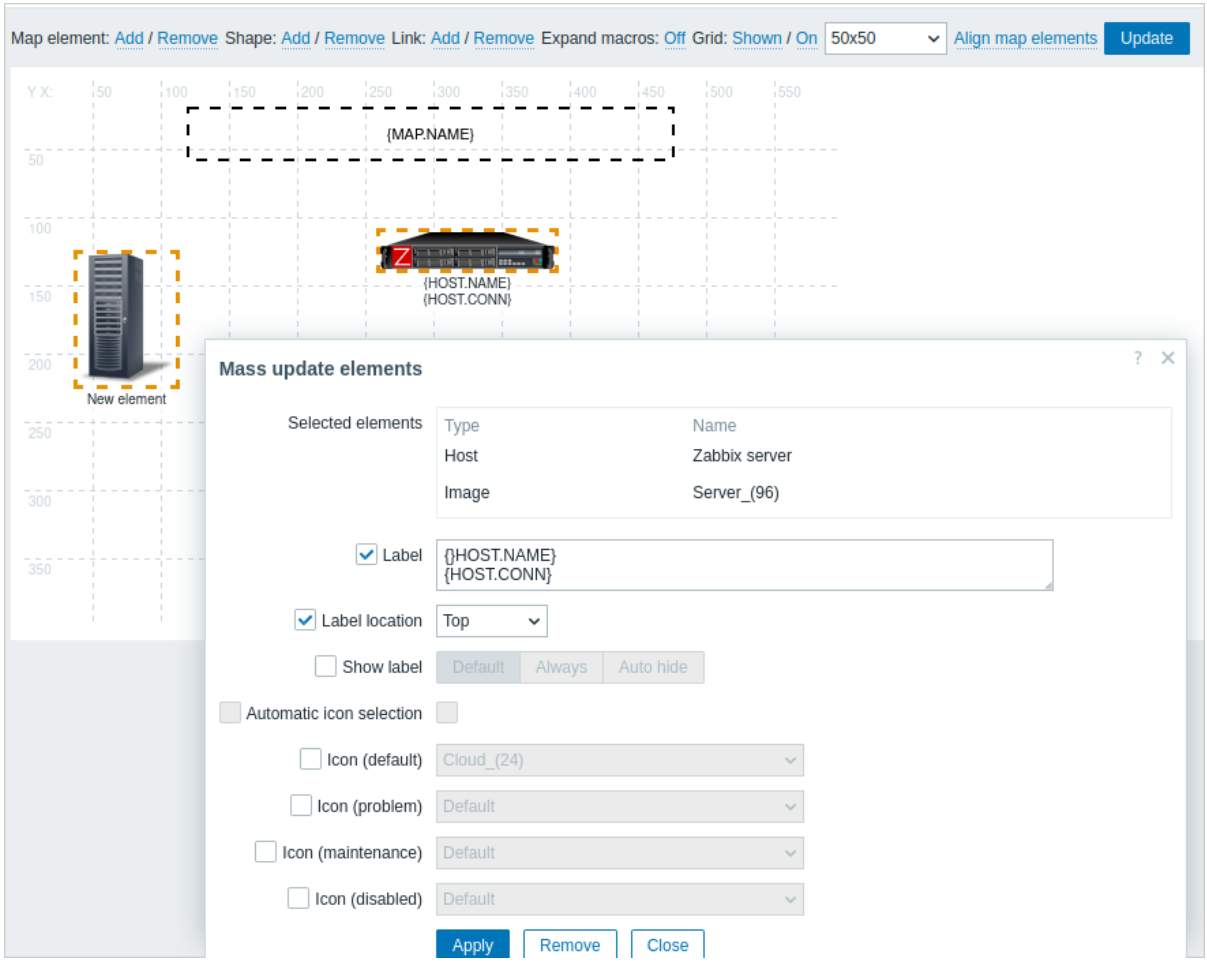
Attention:
Hinzugefügte Elemente werden nicht automatisch gespeichert. Wenn Sie die Seite verlassen, können alle Änderungen verloren gehen.
Daher ist es ratsam, oben rechts auf die Schaltfläche **Update** zu klicken. Nach dem Klicken werden die Änderungen unabhängig davon gespeichert, was Sie im folgenden Popup auswählen.
Ausgewählte Rasteroptionen werden ebenfalls mit jeder Karte gespeichert.

Elemente auswählen

Um Elemente auszuwählen, wählen Sie eines aus und halten Sie dann *Strg* gedrückt, um die anderen auszuwählen.

Sie können auch mehrere Elemente auswählen, indem Sie im bearbeitbaren Bereich ein Rechteck aufziehen und alle darin enthaltenen Elemente auswählen.

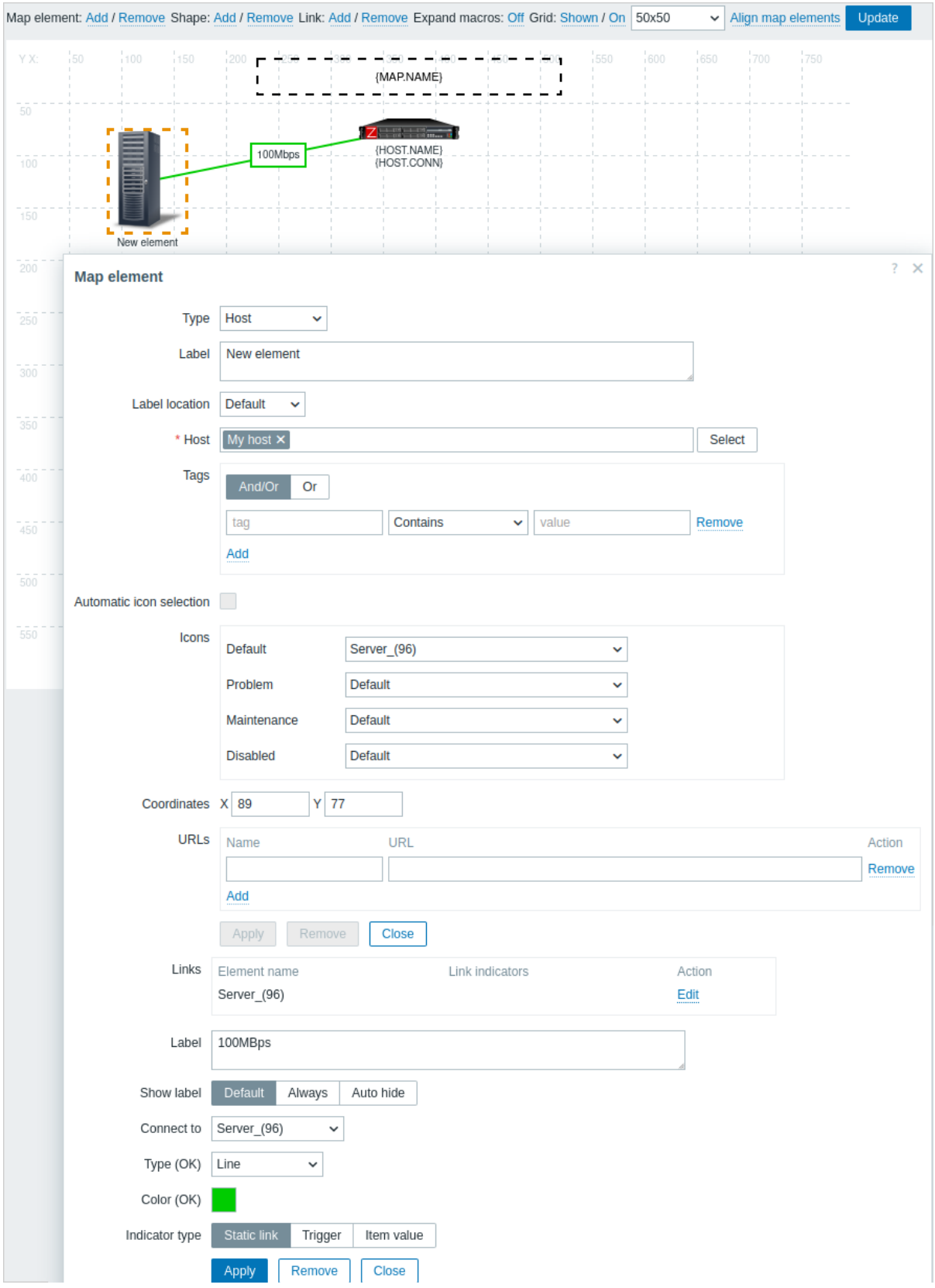
Sobald Sie mehr als ein Element auswählen, wechselt das Formular für die Elementeigenschaften in den Massenaktualisierungsmodus, sodass Sie Attribute der ausgewählten Elemente in einem Schritt ändern können. Markieren Sie dazu das Attribut mit dem Kontrollkästchen und geben Sie einen neuen Wert dafür ein. Sie können hier Makros verwenden (zum Beispiel {HOST.NAME} für die Elementbeschriftung).



Verknüpfung von Elementen

Sobald Sie einige Elemente auf der Karte platziert haben, ist es an der Zeit, mit dem Verknüpfen zu beginnen. Um zwei Elemente zu verknüpfen, müssen Sie sie zuerst auswählen. Klicken Sie bei ausgewählten Elementen neben Link auf *Add*.

Nachdem eine Verknüpfung erstellt wurde, enthält das Formular für das einzelne Element nun einen zusätzlichen Abschnitt *Links*. Klicken Sie auf *Edit*, um die Attribute der Verknüpfung zu bearbeiten.



Attribute der Verknüpfung:

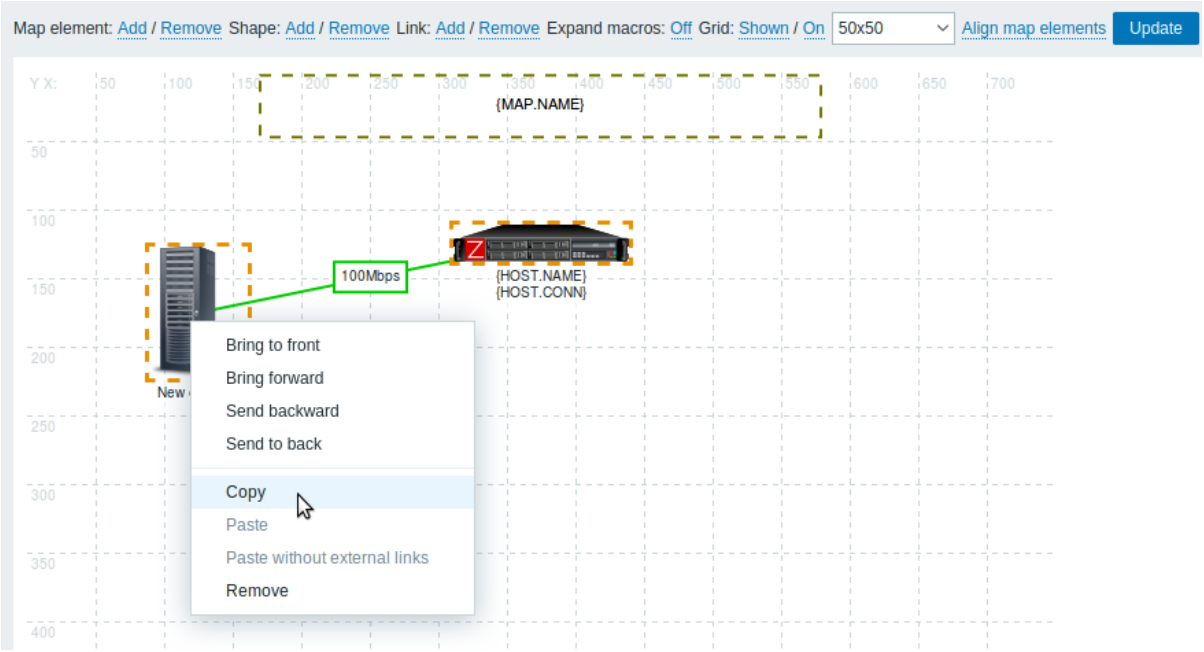
Parameter	Beschreibung
Label	Beschriftung, die oberhalb der Verknüpfung angezeigt wird. Ausdrucks-Makros werden in diesem Feld unterstützt, jedoch nur mit den Funktionen `avg`, `last`, `min` und `max` sowie Zeit als Parameter (zum Beispiel `{?avg(/host/key,1h)}`).

Parameter	Beschreibung
Show label	Wählen Sie aus, wie die Beschriftung der Verknüpfung angezeigt werden soll: **Default** - die Einstellungen *Show map element labels* aus der Kartenkonfiguration verwenden (Standard); **Always** - immer anzeigen (Standard); **Auto hide** - die Beschriftung ausblenden, wenn sie nicht mit der Maus darübergefahren oder ausgewählt ist.
Connect to	Das Element, mit dem die Verknüpfung verbunden ist.
Type (OK)	Standardstil der Verknüpfung. Wählen Sie den Stil: *Line*, *Bold line*, *Dot* oder *Dashed line*.
Color (OK)	Standardfarbe der Verknüpfung. Klicken Sie auf das Farbfeld, um eine andere Farbe auszuwählen.
Indicator type	Wählen Sie den Typ des Verknüpfungsindikators aus: **Static link** - keine Indikatoren; **Trigger** - ermöglicht triggerbasierte Verknüpfungsindikatoren; **Item value** - ermöglicht auf Datenpunktwerten basierende Verknüpfungsindikatoren.
Item	Wählen Sie den Datenpunkt aus. Der Wert dieses Datenpunkts beeinflusst den Stil der Verknüpfung. Dieses Feld ist verfügbar, wenn für *Indicator type Item value* ausgewählt ist.
Indicators	Die Liste der Verknüpfungsindikatoren. Dieses Feld ist verfügbar, wenn *Indicator type* auf *Trigger* oder *Item value* gesetzt ist. Wenn *Trigger* ausgewählt ist, können Auslöser als Verknüpfungsindikatoren zugewiesen werden. Wenn sich ein Auslöser im Zustand *Problem* befindet, wird sein Stil (ausgewählte Farbe und Linientyp) auf die Verknüpfung angewendet. Wenn *Item value* ausgewählt ist und im Feld *Item* ein Datenpunkt gewählt wurde, können Datenpunktwerte als Verknüpfungsindikatoren zugewiesen werden. Wenn der Datenpunktwert den angegebenen Schwellenwert erreicht (bei einem numerischen Datenpunkt) oder dem Muster entspricht (bei einem Textdatenpunkt), wird sein Stil (ausgewählte Farbe und Linientyp) auf die Verknüpfung angewendet.

Verschieben und Kopieren/Einfügen von Elementen

Mehrere ausgewählte Elemente können an eine andere Stelle in der Karte **verschoben** werden, indem Sie auf eines der ausgewählten Elemente klicken, die Maustaste gedrückt halten und den Cursor an die gewünschte Position bewegen.

Ein oder mehrere Elemente können **kopiert** werden, indem Sie die Elemente auswählen, dann mit der rechten Maustaste auf ein ausgewähltes Element klicken und im Menü *Copy* auswählen.

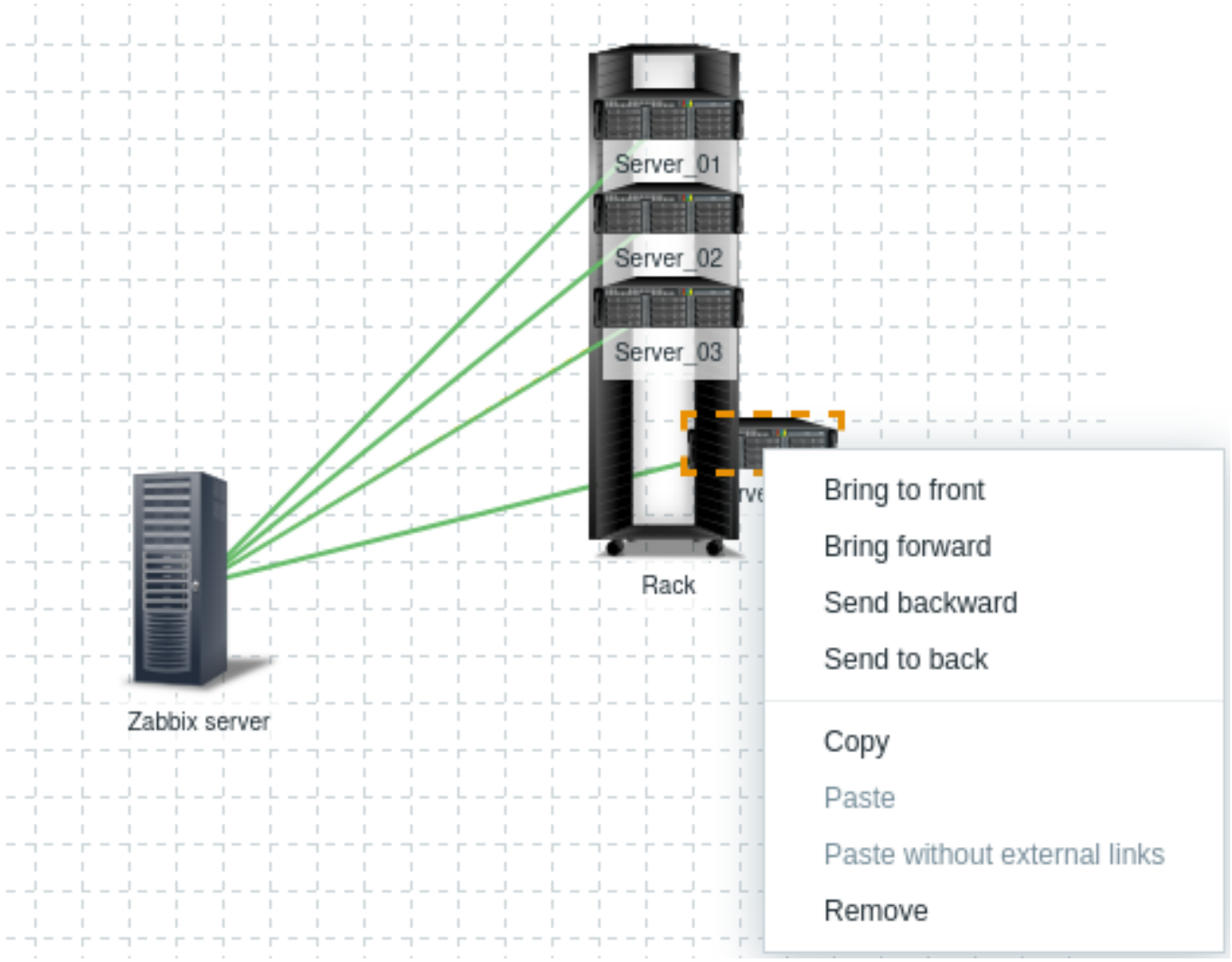


Um die Elemente einzufügen, klicken Sie mit der rechten Maustaste auf einen Bereich der Karte und wählen Sie im Menü *Paste* aus. Die Option *Paste without external links* fügt die Elemente ein und behält dabei nur die Verknüpfungen zwischen den ausgewählten Elementen bei.

Kopieren und Einfügen funktioniert innerhalb desselben Browserfensters. Tastenkombinationen werden nicht unterstützt.

Elemente anordnen

Um ein Element vor ein anderes zu bringen (oder umgekehrt), klicken Sie mit der rechten Maustaste auf das Element und wählen Sie *Nach vorne bringen/In den Vordergrund bringen* oder *Nach hinten senden/In den Hintergrund senden*.

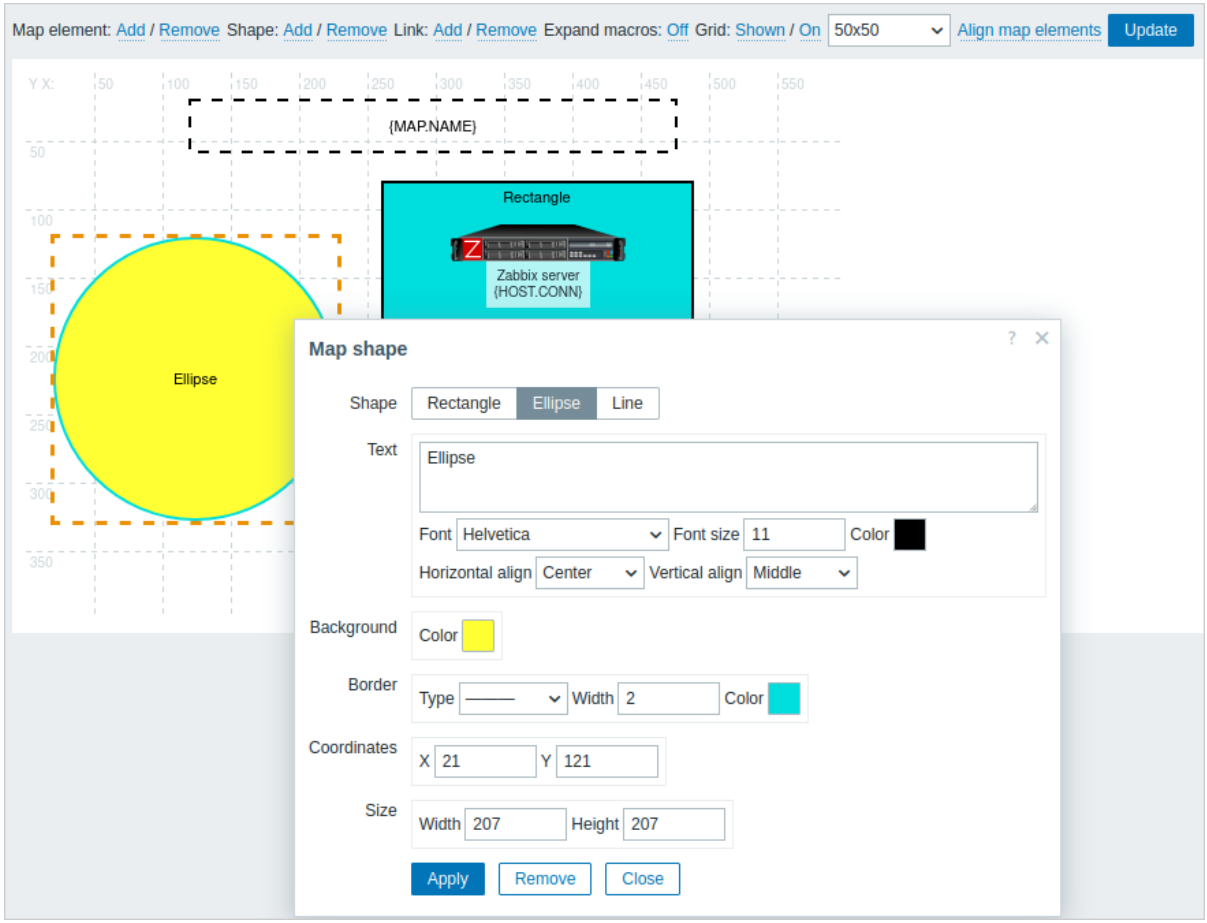


Hinzufügen von Formen

Zusätzlich zu Kartenelementen ist es auch möglich, einige Formen hinzuzufügen. Formen sind keine Kartenelemente; sie sind lediglich eine visuelle Darstellung. So kann beispielsweise eine Rechteckform als Hintergrund verwendet werden, um einige Hosts zu gruppieren. Rechteck- und Ellipsenformen können hinzugefügt werden.

Um eine Form hinzuzufügen, klicken Sie neben Form auf *Hinzufügen*. Die neue Form wird in der oberen linken Ecke der Karte angezeigt. Ziehen Sie sie per Drag-and-drop an die gewünschte Stelle.

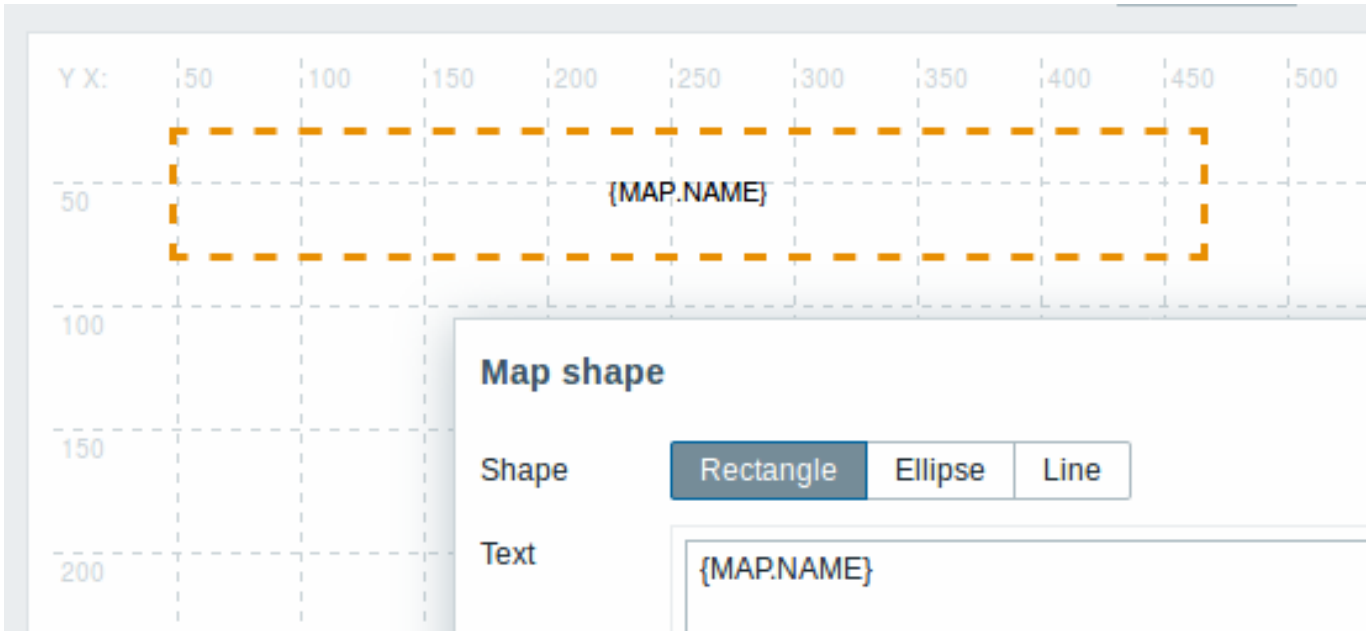
Eine neue Form wird mit Standardfarben hinzugefügt. Wenn Sie auf die Form klicken, wird ein Formular angezeigt, in dem Sie das Aussehen der Form anpassen, Text hinzufügen usw. können.



Um Formen auszuwählen, wählen Sie eine aus und halten Sie dann *Strg* gedrückt, um die anderen auszuwählen. Wenn mehrere Formen ausgewählt sind, können gemeinsame Eigenschaften massenweise aktualisiert werden, ähnlich wie bei Elementen.

In den Formen kann Text hinzugefügt werden. Ausdrucks-Makros werden im Text unterstützt, jedoch nur mit den Funktionen `avg`, `last`, `min` und `max` sowie Zeit als Parameter (zum Beispiel `{?avg(/host/key,1h)}`).

Um nur Text anzuzeigen, kann die Form unsichtbar gemacht werden, indem der Rahmen der Form entfernt wird (wählen Sie im Feld *Rahmen* die Option 'None'). Beachten Sie zum Beispiel, dass das oben im Screenshot sichtbare Makro {MAP.NAME} tatsächlich eine Rechteckform mit Text ist, was sichtbar wird, wenn Sie auf das Makro klicken:



{MAP.NAME} wird beim Anzeigen der Karte in den konfigurierten Kartennamen aufgelöst.

Wenn im Text Hyperlinks verwendet werden, werden diese beim Anzeigen der Karte anklickbar.

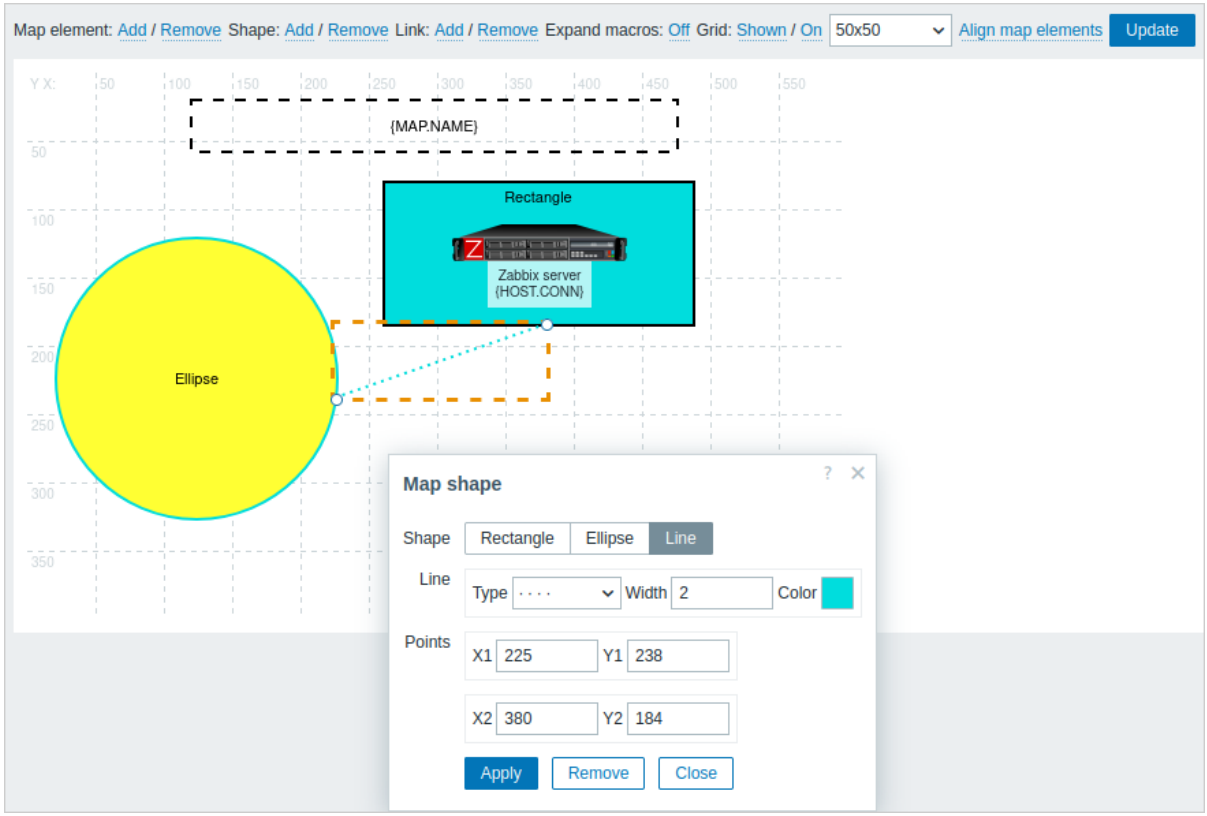
Der Zeilenumbruch für Text ist innerhalb von Formen immer aktiviert. Innerhalb einer Ellipse werden die Zeilen jedoch so umbrochen, als wäre die Ellipse ein Rechteck. Wortumbruch ist nicht implementiert, daher werden lange Wörter (Wörter, die nicht

in die Form passen) nicht umbrochen, sondern maskiert (auf der Seite zur Kartenbearbeitung) oder abgeschnitten (auf anderen Seiten mit Karten).

Linien hinzufügen

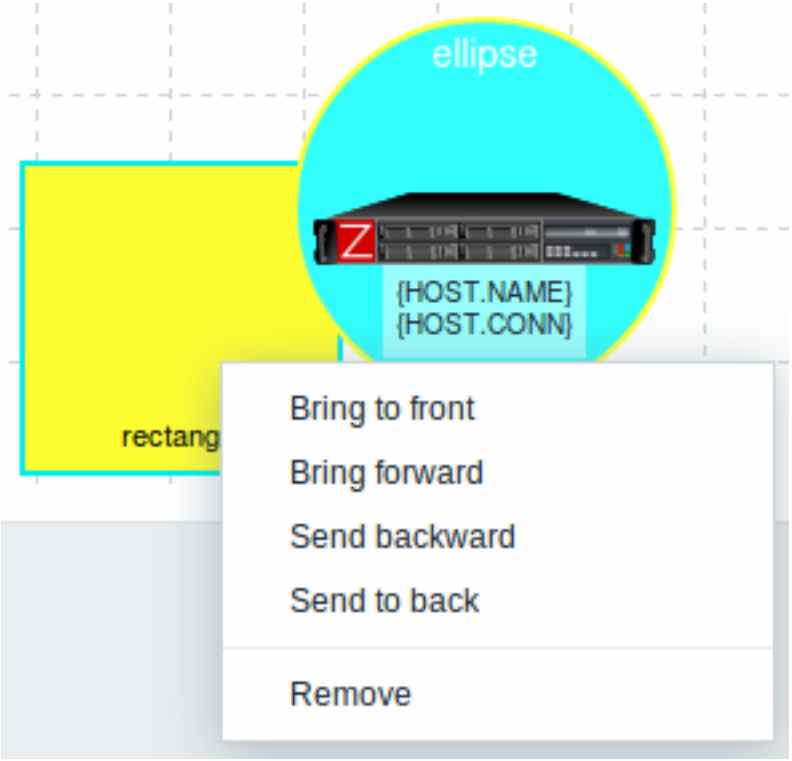
Zusätzlich zu Formen ist es auch möglich, einige Linien hinzuzufügen. Linien können verwendet werden, um Elemente oder Formen in einer Karte zu verbinden.

Um eine Linie hinzuzufügen, klicken Sie neben Form auf *Hinzufügen*. Eine neue Form wird in der oberen linken Ecke der Karte angezeigt. Wählen Sie sie aus und klicken Sie im Bearbeitungsformular auf *Linie*, um die Form in eine Linie umzuwandeln. Passen Sie anschließend die Linieneigenschaften an, z. B. Linientyp, Breite, Farbe usw.



Formen und Linien anordnen

Um eine Form vor die andere zu bringen (oder umgekehrt), klicken Sie mit der rechten Maustaste auf die Form, um das Menü für Kartenformen zu öffnen.

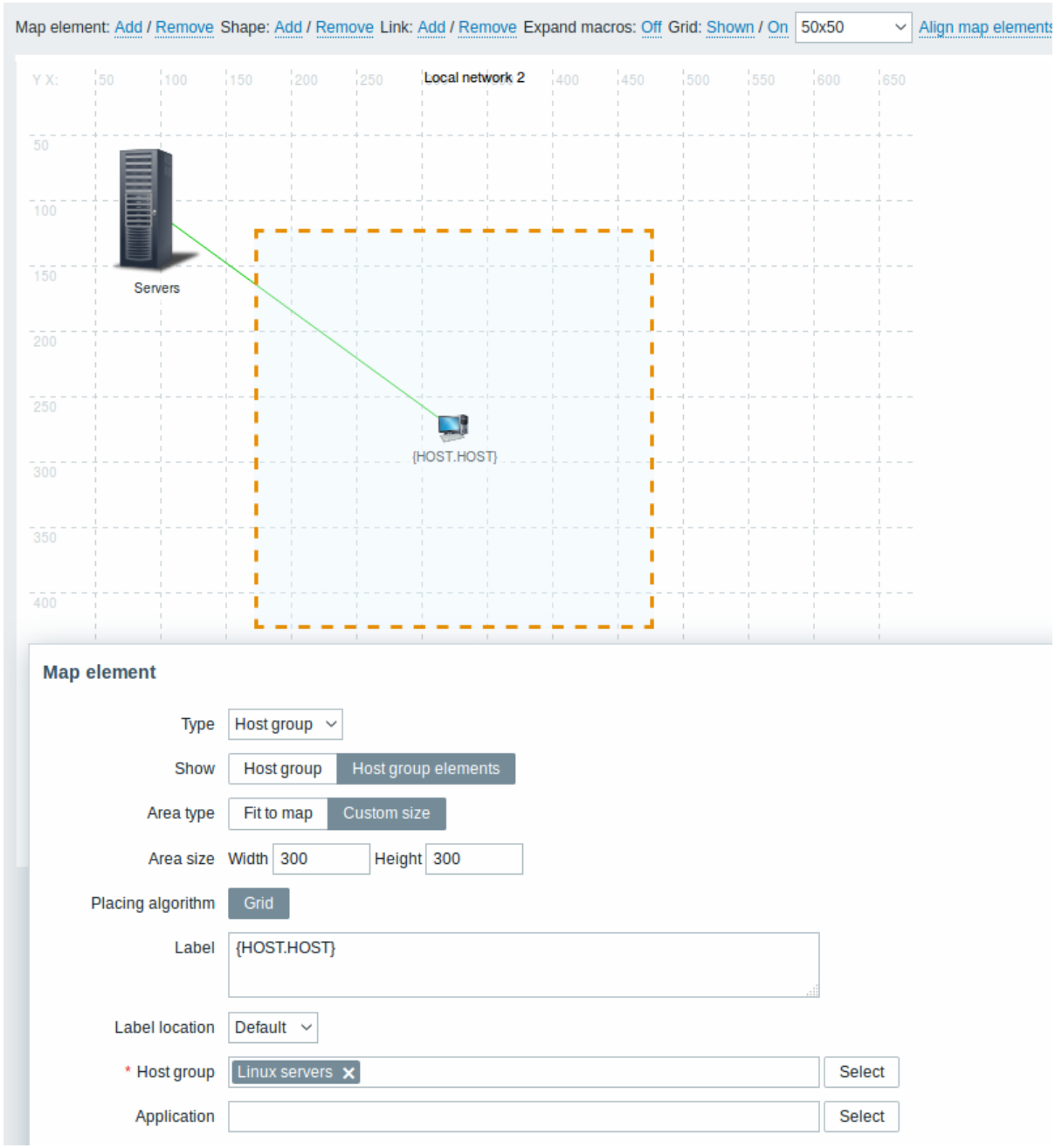


Hostgruppen-Elemente

Übersicht

In diesem Abschnitt wird erläutert, wie beim Konfigurieren einer Netzwerkkarte ein Element vom Typ *Host-Gruppe* hinzugefügt wird.

Konfiguration



Alle Pflichtfelder sind mit einem roten Sternchen gekennzeichnet.

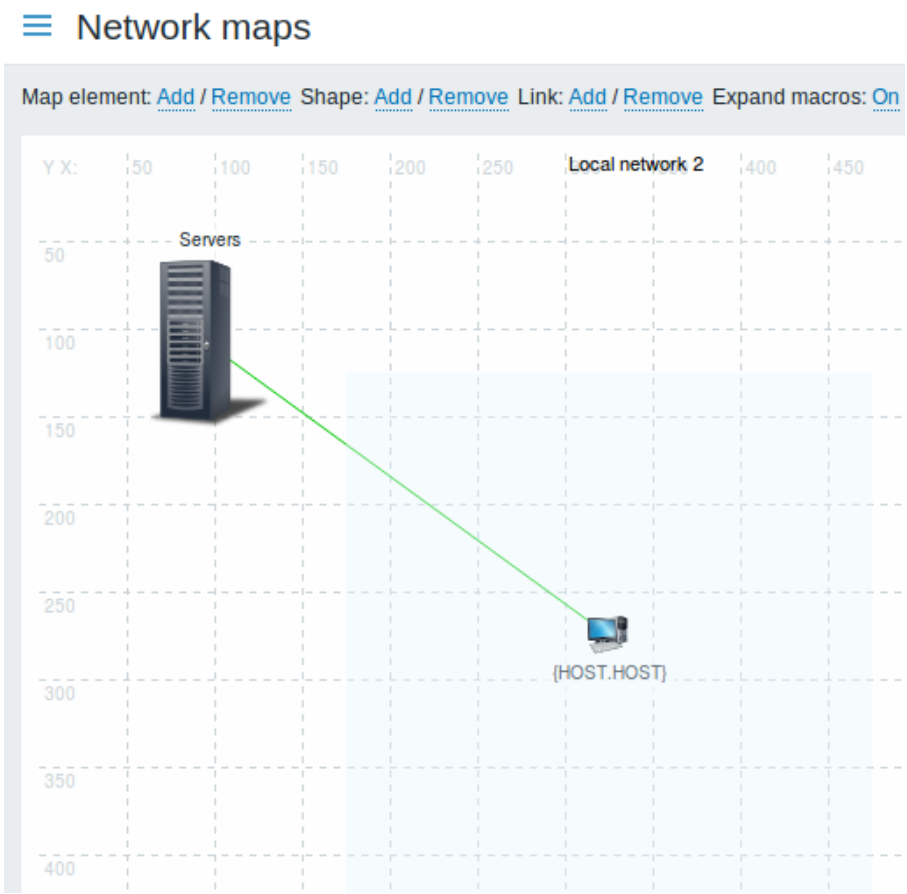
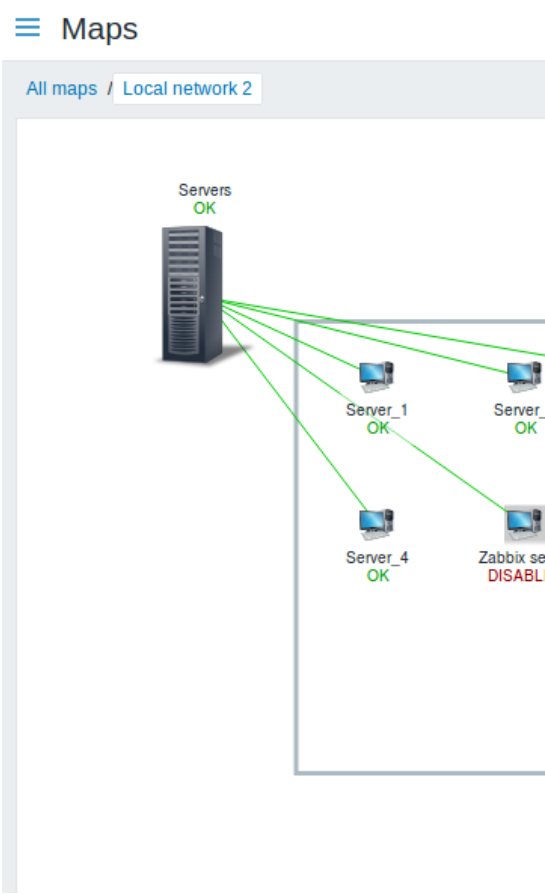
Diese Tabelle enthält Parameter, die für den Elementtyp *Host group* typisch sind:

Parameter	Beschreibung
Type	Wählen Sie den Elementtyp aus: **Host group** – Symbol, das den Status aller Auslöser aller Hosts der ausgewählten Gruppe darstellt.
Show	Optionen für die Anzeige: **Host group** – Bei Auswahl dieser Option wird ein einzelnes Symbol mit den entsprechenden Informationen zur betreffenden Hostgruppe angezeigt. **Host group elements** – Bei Auswahl dieser Option werden mehrere Symbole mit den entsprechenden Informationen zu jedem einzelnen Element (Host) der betreffenden Hostgruppe angezeigt.
Area type	Diese Einstellung ist verfügbar, wenn *Show* auf *Host group elements* gesetzt ist: **Fit to map** – Alle Elemente der Hostgruppe werden gleichmäßig innerhalb der Karte platziert. **Custom size** – Manuelle Einstellung des Kartenbereichs für alle anzuzeigenden Elemente der Hostgruppe.
Area size	Diese Einstellung ist verfügbar, wenn *Area type* auf *Custom size* gesetzt ist: **Width** – Wert in Pixeln zur Angabe der Breite des Kartenbereichs. **Height** – Wert in Pixeln zur Angabe der Höhe des Kartenbereichs.

Parameter	Beschreibung
Placing algorithm	**Grid** – Einzige verfügbare Option zur Anzeige aller Elemente der Hostgruppe.
Label	Beschriftung des Symbols, beliebige Zeichenfolge. Makros und mehrzeilige Zeichenfolgen können in Beschriftungen verwendet werden. Wenn der Typ des Kartenelements *Host group* ist, wirkt sich die Angabe bestimmter Makros auf die Kartenansicht aus, indem entsprechende Informationen zu jedem einzelnen Host angezeigt werden. Wenn beispielsweise das Makro `{HOST.IP}` verwendet wird, zeigt die Kartenansicht beim Bearbeiten nur das Makro `{HOST.IP}` selbst an, während die Kartenansicht die eindeutige IP-Adresse jedes Hosts enthält und anzeigt.

Anzeigen von Elementen einer Hostgruppe

Diese Option ist verfügbar, wenn die Anzeigeoption *Elemente der Hostgruppe* ausgewählt ist.
Wenn Sie *Elemente der Hostgruppe* als Anzeigeoption auswählen, sehen Sie zunächst nur ein Symbol für die Hostgruppe.
Wenn Sie die Karte jedoch speichern und anschließend zur Kartenansicht wechseln, sehen Sie, dass die Karte alle Elemente (Hosts) der betreffenden Hostgruppe enthält:

Kartenbearbeitungsansicht	Kartenansicht
	

Beachten Sie, wie das Makro `{HOST.HOST}` verwendet wird.
Bei der Kartenbearbeitung wird der Makroname nicht aufgelöst, während in der Kartenansicht die eindeutigen Namen aller Hosts angezeigt werden.

Link-Indikatoren

Übersicht

Sie können Indikatoren einer Verbindung zwischen Elementen in einer Netzwerkkarte zuweisen.

Die Indikatoren können auf Auslösern oder Datenpunktwerten basieren. Es ist möglich, unterschiedliche Verbindungsstile und -farben anzuzeigen:

- wenn Auslöser in einen Problemzustand wechseln;

- wenn der Datenpunktwert:
  - einen Schwellenwert erreicht (bei numerischen Datenpunkten);
  - mit einem regulären Ausdruck übereinstimmt (bei Text-Datenpunkten).

Wenn Sie eine Verbindung konfigurieren, legen Sie den Standard-Verbindungstyp und die Standardfarbe fest. Durch das Zuweisen von Indikatoren zu einer Verbindung wird es möglich, den Verbindungsstil und die Farbe vom Auslöserzustand oder Datenpunktwert abhängig zu machen.

Wenn zum Beispiel einer der zugewiesenen Auslöser in einen Problemzustand wechselt, ändern sich Verbindungsstil und -farbe entsprechend. Vielleicht war Ihre Standardverbindung also eine grüne Linie. Befindet sich der Auslöser nun im Problemzustand, kann Ihre Verbindung fett und rot werden (wenn Sie dies so definiert haben).

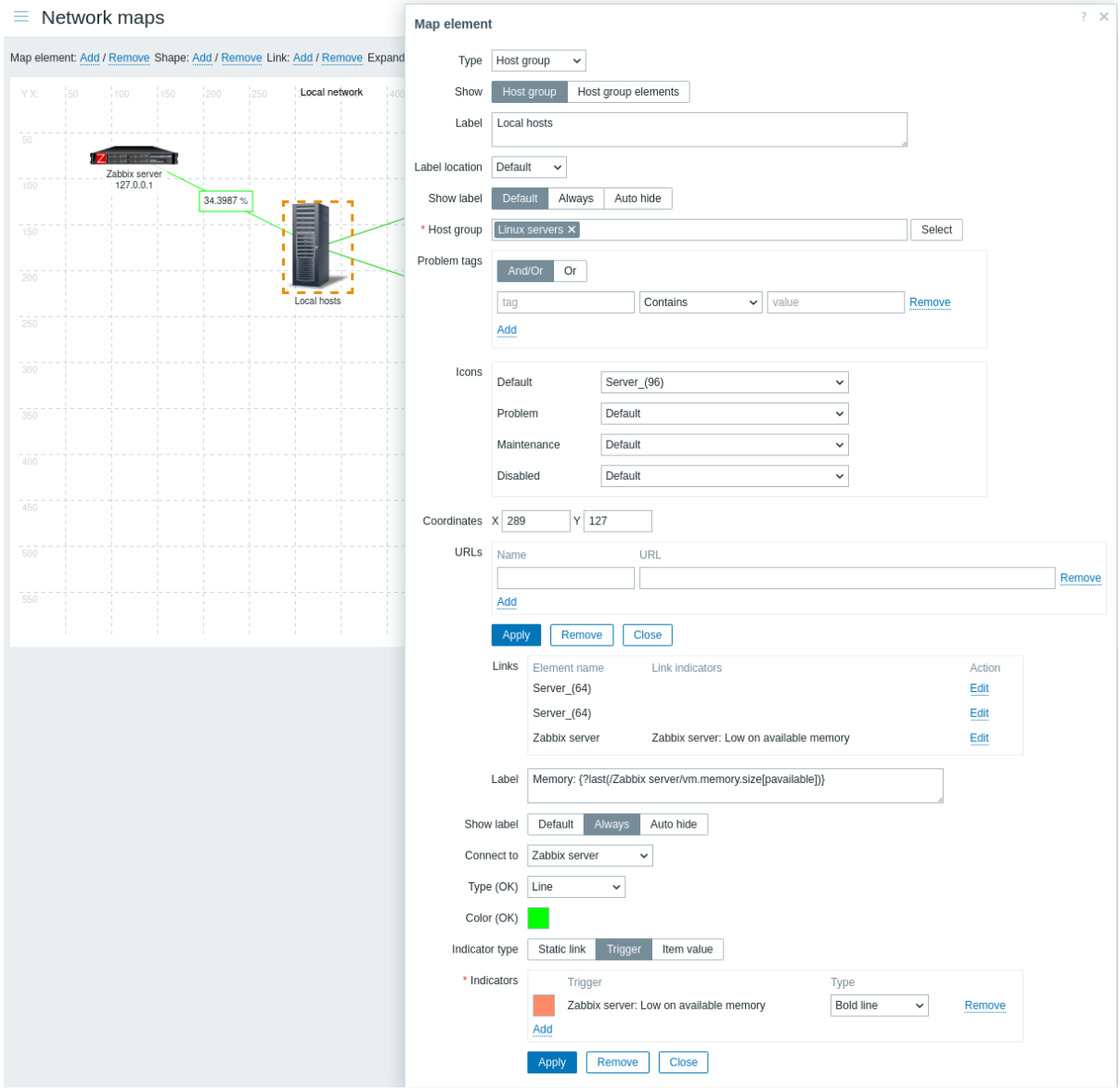
Ebenso kann der Verbindungsstil dies widerspiegeln, wenn ein Datenpunktwert einen angegebenen Schwellenwert erreicht oder mit einem angegebenen regulären Ausdruck übereinstimmt.

Konfiguration

Auslöser

Um Auslöser als Link-Indikatoren zuzuweisen, gehen Sie wie folgt vor:

- wählen Sie ein Map-Element aus
- klicken Sie im Abschnitt *Links* für den entsprechenden Link auf *Edit*
- wählen Sie *Trigger* als Indikatortyp aus
- klicken Sie im Block *Indicators* auf *Add* und wählen Sie einen oder mehrere Auslöser aus

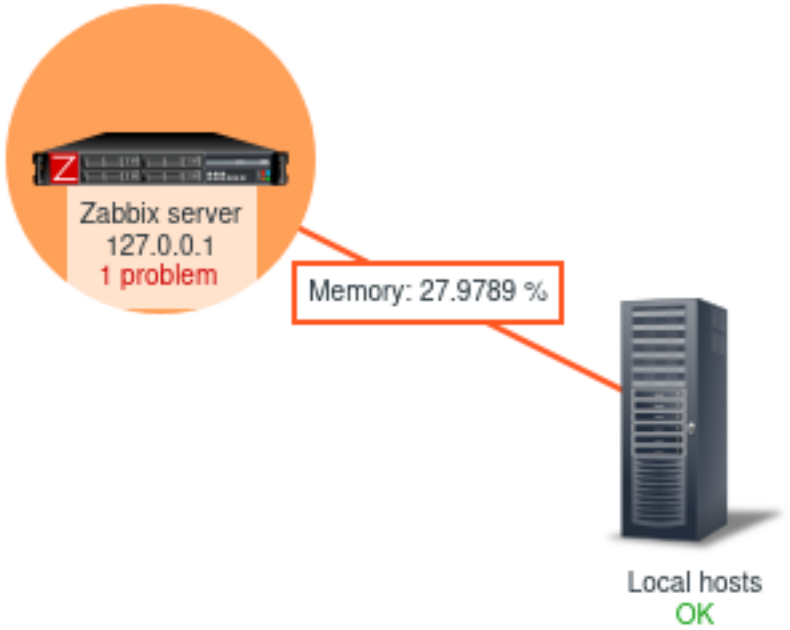


Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert.

Hinzugefügte Auslöser werden in der Liste *Indicators* angezeigt.

Sie können den Linktyp und die Farbe für jeden Indikator direkt in der Liste festlegen. Wenn Sie fertig sind, klicken Sie auf *Apply*, schließen Sie das Formular und klicken Sie auf *Update*, um die Änderungen an der Map zu speichern.

In *Monitoring* > *Maps* werden der entsprechende Linktyp und die Farbe angezeigt, wenn der **trigger** in einen Problemzustand wechselt.



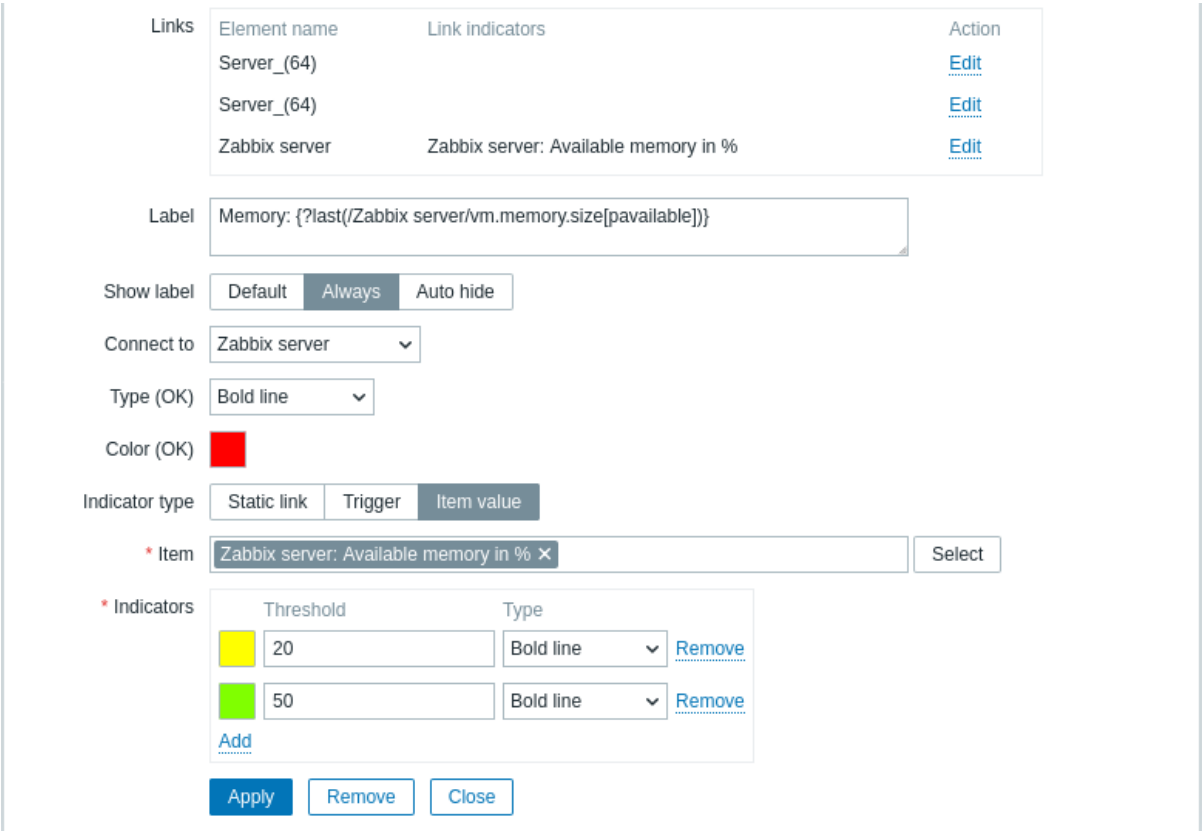
Note:
Wenn mehrere Auslöser in einen Problemzustand wechseln, bestimmt das Problem mit der höchsten Schwere den Linkstil und die Farbe. Wenn mehreren Auslösern mit derselben Schwere derselbe Map-Link zugewiesen ist, hat derjenige mit der niedrigsten ID Vorrang. Beachten Sie außerdem:

1. Die Einstellungen *Minimum trigger severity* und *Show suppressed problem* aus der Map-Konfiguration beeinflussen, welche Probleme berücksichtigt werden.
2. Bei Auslösern mit mehreren Problemen (multiple problem generation) kann jedes Problem eine von der Auslöser-Schwere abweichende Schwere haben (manuell geändert), andere Tags aufweisen (aufgrund von Makros) und unterdrückt sein.

Item-Werte

Um Item-Werte als Link-Indikatoren zuzuweisen, gehen Sie wie folgt vor:

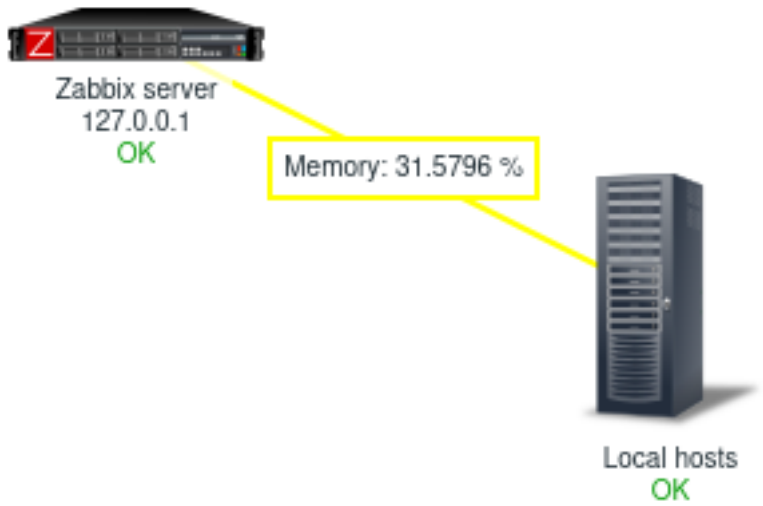
- wählen Sie ein Map-Element aus
- klicken Sie im Abschnitt *Links* für den entsprechenden Link auf *Edit*
- wählen Sie *Item value* als Indikatortyp aus
- wählen Sie das Datenpunkt aus
- fügen Sie im Block *Indicators* einen oder mehrere Schwellenwerte oder Muster für Item-Werte hinzu



Hinzugefügte Item-Schwellenwerte/Muster werden in der Liste *Indicators* angezeigt.

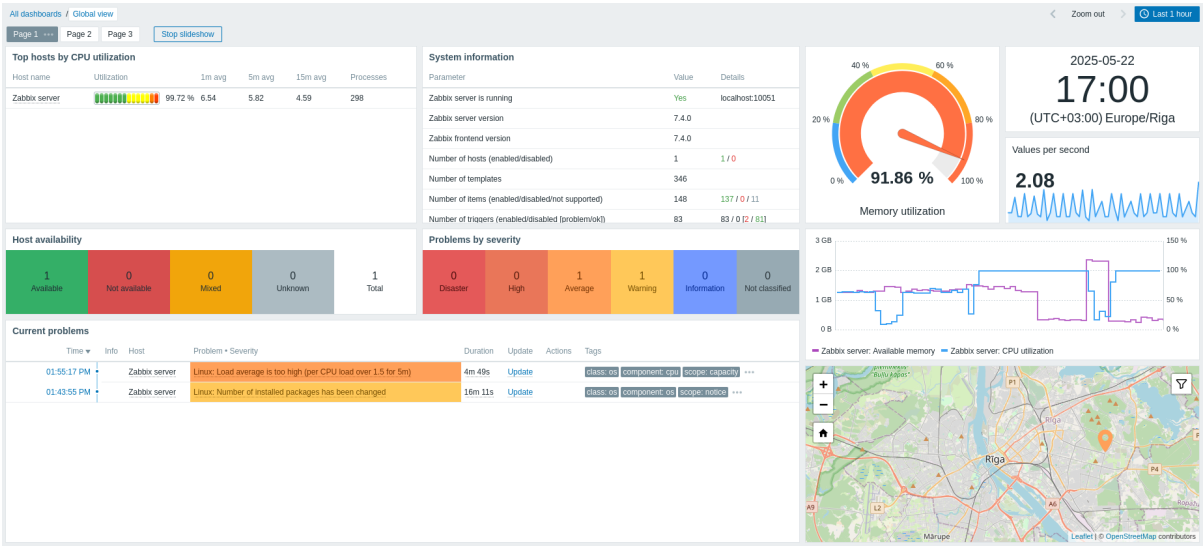
Sie können den Linktyp und die Farbe für jeden Indikator direkt in der Liste festlegen. Wenn Sie fertig sind, klicken Sie auf *Apply*, schließen Sie das Formular und klicken Sie auf *Update*, um die Änderungen an der Map zu speichern.

In *Monitoring* > *Maps* werden der entsprechende Linktyp und die Farbe angezeigt, wenn der **item value** den festgelegten Schwellenwert erreicht (bei numerischen Datentypen) oder dem regulären Ausdrucksmuster entspricht (bei Textdatentypen).



Dashboards

Dashboards - sowohl globale Dashboards als auch Host-Dashboards - bieten eine leistungsstarke Visualisierungsplattform mit solchen Widgets und Werkzeugen wie modernen Diagrammen, Karten, Diashows und mehr.



Vorlagen und Vorlagengruppen

Übersicht

Die Verwendung von Vorlagen ist eine hervorragende Möglichkeit, den eigenen Arbeitsaufwand zu reduzieren und die Zabbix-Konfiguration zu optimieren. Eine Vorlage ist eine Sammlung von Entitäten, die bequem auf mehrere Hosts angewendet werden kann.

Die Entitäten können sein:

- Datenpunkte
- Auslöser
- Diagramme
- Dashboards
- Low-Level-Discovery-Regeln
- Webszenarien

Da viele Hosts in der Praxis identisch oder recht ähnlich sind, liegt es nahe, dass die Menge an Entitäten (Datenpunkte, Auslöser, Diagramme, ...), die Sie für einen Host erstellt haben, auch für viele andere nützlich sein kann. Natürlich könnten Sie diese auf jeden neuen Host kopieren, aber das würde viel manuelle Arbeit bedeuten. Stattdessen können Sie sie mit Vorlagen in eine Vorlage kopieren und die Vorlage dann auf so viele Hosts anwenden, wie benötigt.

Wenn eine Vorlage mit einem Host verknüpft ist, werden alle Entitäten (Datenpunkte, Auslöser, Diagramme, ...) der Vorlage zum Host hinzugefügt. Vorlagen werden jedem einzelnen Host direkt zugewiesen (und nicht einer Hostgruppe).

Vorlagen werden häufig verwendet, um Entitäten für bestimmte Dienste oder Anwendungen zu gruppieren (wie Apache, MySQL, PostgreSQL, Postfix ...) und dann auf Hosts anzuwenden, auf denen diese Dienste ausgeführt werden.

Ein weiterer Vorteil der Verwendung von Vorlagen ist, dass Änderungen für alle Hosts zentral vorgenommen werden können. Wenn etwas einmal auf Vorlagenebene geändert wird, wird diese Änderung an alle verknüpften Hosts weitergegeben.

Vorlagen sind in Vorlagengruppen organisiert.

Fahren Sie mit dem Abschnitt Erstellen und Konfigurieren einer Vorlage fort.

Vorlagen out of the box

Übersicht

Zabbix stellt eine wachsende Anzahl vorkonfigurierter Vorlagen bereit, um die Einrichtung von Überwachungszielen zu vereinfachen und zu beschleunigen.

Alle sofort einsatzbereiten Vorlagen sind unter *Datensammlung* > *Vorlagen* verfügbar.

Beim Upgrade von Zabbix werden Vorlagen nicht automatisch aktualisiert. Dadurch wird verhindert, dass von Ihnen vorgenommene benutzerdefinierte Änderungen überschrieben werden. Informationen zum manuellen Aktualisieren oder Hinzufügen von Vorlagen finden Sie im folgenden Abschnitt *Vorlagen-Upgrade*.

Verwenden Sie die Seitenleiste, um Informationen zu bestimmten Vorlagentypen und den erforderlichen Betriebsbedingungen aufzurufen.

Globale Dashboards, die für die Verwendung mit vorhandenen Zabbix-Vorlagen entwickelt wurden, können aus dem Dashboards-Repository heruntergeladen und in Ihre Zabbix-Umgebung importiert werden. Eine detaillierte Beschreibung jedes Dashboards einschließlich der Voraussetzungen finden Sie in der Datei `README` des jeweiligen Dashboards.

Siehe auch:

- Eine Vorlage verknüpfen
- Bekannte Probleme bei Vorlagen

Vorlagen-Upgrade

So aktualisieren oder fügen Sie eine **einzelne Vorlage** hinzu:

1. Öffnen Sie das Verzeichnis `templates` im Zabbix-Git-Repository.

2. Wählen Sie den Branch (unter der Bezeichnung *Source*), der Ihrer Zabbix-Version entspricht (z. B. *master* für die Entwicklungsversion).

3. Suchen Sie die Vorlage, die Sie hinzufügen oder aktualisieren möchten (z. B. MySQL by Zabbix agent).

4. Kopieren Sie den Inhalt der Vorlagendatei (z. B. `template_db_mysql_agent.yaml`) und speichern Sie ihn in einer lokalen Datei.

5. Gehen Sie in der Zabbix-Benutzeroberfläche zu *Datensammlung* > *Vorlagen* und importieren Sie die lokale Datei.

> **Note:**
> Webhook-Medientypen können auf ähnliche Weise aktualisiert werden: Rufen Sie sie aus dem Verzeichnis `media` ab und importieren Sie sie unter *Alarme* > *Medientypen*.

So aktualisieren oder fügen Sie **mehrere Vorlagen** hinzu:

1. Laden Sie das Archiv mit den Vorlagenpaketen abhängig von Ihrem Webserver herunter:

- Nginx - `templates_bundles-8.0-1M-nginx.zip`
- Apache - `templates_bundles-8.0-2M-apache.zip`

2. Entpacken Sie das Archiv. Es enthält YAML-Dateien mit jeweils mehreren Vorlagen sowie eine Indexdatei, in der aufgeführt ist, welche Vorlagen in den einzelnen YAML-Dateien enthalten sind.

3. Gehen Sie in der Zabbix-Benutzeroberfläche zu *Datensammlung* > *Vorlagen* und importieren Sie die YAML-Dateien nacheinander.

> **Note:**
> Webhook-Medientypen können auf ähnliche Weise aktualisiert werden: Nehmen Sie die Datei `media_bundle-8.0-000.yaml` aus dem Archiv mit den Vorlagenpaketen und importieren Sie sie unter *Alarme* > *Medientypen*.

Um **alle Vorlagen** gleichzeitig zu aktualisieren oder hinzuzufügen, können Sie das im Archiv mit den Vorlagenpaketen enthaltene Skript `import_templates.sh` verwenden. Das Skript erfordert die Dienstprogramme `curl` und `jq` und verwendet die Zabbix-API-Methode `configuration.import`. Geben Sie beim Ausführen des Skripts das Verzeichnis mit den YAML-Dateien sowie die Zabbix-API-URL und das Authentifizierungstoken an.

> **Warning:**
> Das Skript `import_templates.sh` überschreibt alle vorhandenen Vorlagen. Es bietet keine Möglichkeit, Vorlagenänderungen zu überprüfen oder rückgängig zu machen.

Beispiel:

```
/tmp/templates_bundles-8.0-2M-apache/import_templates.sh \
  /tmp/templates_bundles-8.0-2M-apache \
  https://example.com/zabbix/api_jsonrpc.php \
  d8d6b5c78ee2a8333db11bfcd3e448c49c239d8ce00702759e2b56f42e24dbe4
```

Vorlagenkompatibilität mit dem Host Wizard

Der Host Wizard ist mit allen Vorlagen „out of the box“ kompatibel.

Sie können auch Ihre benutzerdefinierten Vorlagen mit dem Host Wizard kompatibel machen:

1. Gehen Sie im Zabbix Frontend zu *Datensammlung* > *Vorlagen* und exportieren Sie die benutzerdefinierte Vorlage.

2. Bearbeiten Sie die exportierte Datei:

   - Fügen Sie das Vorlagenelement `wizard_ready` hinzu, um die Kompatibilität mit dem Host Wizard anzugeben.
   - Fügen Sie bei Bedarf optional das Vorlagenelement `readme` oder `config` hinzu.

3. Gehen Sie im Zabbix Frontend zu *Datensammlung* > *Vorlagen* und importieren Sie die aktualisierte Vorlagendatei.

Betrieb der Vorlage für den Zabbix Agent

Schritte, um den korrekten Betrieb von Vorlagen sicherzustellen, die Metriken mit dem Zabbix Agent erfassen:

1. Stellen Sie sicher, dass der Zabbix Agent auf dem Host installiert ist. Stellen Sie bei aktiven Prüfungen außerdem sicher, dass die Adresse des Zabbix Servers/Proxys dem Parameter „ServerActive“ der Agent-Konfigurationsdatei hinzugefügt wurde.
2. Verknüpfen Sie die Vorlage mit einem Ziel-Host (falls die Vorlage in Ihrer Zabbix-Installation nicht verfügbar ist, müssen Sie die Vorlage möglicherweise zuerst importieren).
3. Passen Sie bei Bedarf die Werte der Vorlagenmakros an.
4. Konfigurieren Sie die überwachte Instanz so, dass sie die gemeinsame Nutzung von Daten mit Zabbix erlaubt.

Eine detaillierte Beschreibung einer Vorlage einschließlich der vollständigen Liste der Makros, Datenpunkte und Auslöser ist in der README-Datei der Vorlage verfügbar (sie kann durch Klicken auf den Namen einer Vorlage aufgerufen werden).

Die folgenden Vorlagen sind verfügbar:

- Apache by Zabbix agent
- Apache by Zabbix agent active
- HAProxy by Zabbix agent
- IIS by Zabbix agent
- IIS by Zabbix agent active
- MariaDB by Zabbix agent
- MariaDB by Zabbix agent active
- Microsoft Exchange Server 2016 by Zabbix agent
- Microsoft Exchange Server 2016 by Zabbix agent active
- MySQL by Zabbix agent
- MySQL by Zabbix agent active
- Nginx by Zabbix agent
- Nginx by Zabbix agent active
- Percona by Zabbix agent
- Percona by Zabbix agent active
- PHP-FPM by Zabbix agent
- PHP-FPM by Zabbix agent active
- Podman by Zabbix agent
- Podman by Zabbix agent active
- PostgreSQL by Zabbix agent
- PostgreSQL by Zabbix agent active
- RabbitMQ cluster by Zabbix agent

Zabbix Agent 2 Vorlagebetrieb

Schritte zur Sicherstellung des korrekten Betriebs von Vorlagen, die Metriken mit Zabbix agent 2 erfassen:

1. Stellen Sie sicher, dass der Agent 2 auf dem Host installiert ist und dass die installierte Version das erforderliche Plugin enthält. In einigen Fällen müssen Sie den Agent 2 zuerst aktualisieren.
2. Verknüpfen Sie die Vorlage mit einem Ziel-Host (falls die Vorlage in Ihrer Zabbix-Installation nicht verfügbar ist, müssen Sie die Vorlage möglicherweise zuerst importieren).
3. Passen Sie bei Bedarf die Werte der Vorlagen-Makros an. Beachten Sie, dass Benutzermakros verwendet werden können, um Konfigurationsparameter zu überschreiben.
4. Konfigurieren Sie die zu überwachende Instanz so, dass der Datenaustausch mit Zabbix erlaubt ist.

Attention:
Zabbix agent 2 Vorlagen arbeiten in Verbindung mit den Plugins. Während die Grundkonfiguration einfach durch Anpassung von Benutzermakros erfolgen kann, lässt sich die tiefere Anpassung durch die Konfiguration des Plugins selbst erreichen. Wenn ein Plugin beispielsweise benannte Sitzungen unterstützt, können mehrere Entitäten desselben Typs (z. B. MySQL1 und MySQL2) überwacht werden, indem in der Konfigurationsdatei für jede Entität eine benannte Sitzung mit eigener URI, Benutzername und Passwort angegeben wird.

Eine detaillierte Beschreibung einer Vorlage, einschließlich der vollständigen Liste der Makros, Datenpunkte und Auslöser, ist in der README-Datei der Vorlage verfügbar (zugänglich durch Klicken auf den Vorlagennamen).

Die folgenden Vorlagen sind verfügbar:

- Ceph by Zabbix agent 2
- Docker
- Memcached
- MongoDB cluster by Zabbix agent 2
- MongoDB node by Zabbix agent 2
- MySQL by Zabbix agent 2
- MySQL by Zabbix agent 2 active
- Oracle by Zabbix agent 2
- PostgreSQL by Zabbix agent 2
- PostgreSQL by Zabbix agent 2 active
- Redis
- SMART by Zabbix agent 2
- SMART by Zabbix agent 2 active
- Systemd by Zabbix agent 2
- Website certificate by Zabbix agent 2

HTTP-Vorlagenoperation

Schritte, um den korrekten Betrieb von Vorlagen sicherzustellen, die Metriken mit dem HTTP-Agent erfassen:

1. Erstellen Sie in Zabbix einen Host, der dem Überwachungsziel entspricht. Geben Sie die IP-Adresse bzw. den DNS-Namen des Überwachungsziels als primäre Host-Schnittstelle an. Dies ist erforderlich, damit das Makro `{HOST.CONN}` in den Datenpunkten der Vorlage korrekt aufgelöst wird.
2. Verknüpfen Sie die Vorlage mit dem in Schritt 1 erstellten Host (falls die Vorlage in Ihrer Zabbix-Installation nicht verfügbar ist, müssen Sie die Vorlage möglicherweise zuerst importieren).
3. Passen Sie bei Bedarf die Werte der Makros auf Vorlagenebene an.
4. Konfigurieren Sie die überwachte Instanz so, dass sie die gemeinsame Nutzung von Daten mit Zabbix erlaubt.

Eine detaillierte Beschreibung einer Vorlage einschließlich der vollständigen Liste der Makros, Datenpunkte und Auslöser ist in der `README`-Datei der Vorlage verfügbar (auf die Sie durch Klicken auf einen Vorlagennamen zugreifen können).

Die folgenden Vorlagen sind verfügbar:

- Acronis Cyber Protect Cloud by HTTP
- Apache by HTTP
- Asterisk by HTTP
- AWS by HTTP
- AWS Cost Explorer by HTTP
- AWS EC2 by HTTP
- AWS ECS Cluster by HTTP
- AWS ECS Serverless Cluster by HTTP
- AWS ELB Application Load Balancer by HTTP
- AWS ELB Network Load Balancer by HTTP
- AWS Lambda by HTTP
- AWS RDS instance by HTTP
- AWS S3 bucket by HTTP
- Azure by HTTP
- Cisco Meraki organization by HTTP
- Cisco SD-WAN by HTTP
- Cisco Secure Firewall Threat Defense by HTTP
- Claude API by HTTP
- ClickHouse by HTTP
- Cloudflare by HTTP

- CockroachDB by HTTP
- Control-M enterprise manager by HTTP
- Control-M server by HTTP
- Cradlepoint NCM v2 by HTTP
- DELL PowerEdge R720 by HTTP
- DELL PowerEdge R740 by HTTP
- DELL PowerEdge R820 by HTTP
- DELL PowerEdge R840 by HTTP
- Domain RDAP by HTTP
- Elasticsearch Cluster by HTTP
- Envoy Proxy by HTTP
- Etcd by HTTP
- FortiGate by HTTP
- GitHub repository by HTTP
- GitLab by HTTP
- GLPI by HTTP
- Google Cloud Platform (GCP) by HTTP
- Hadoop by HTTP
- HAProxy by HTTP
- HashiCorp Consul Cluster by HTTP
- HashiCorp Consul Node by HTTP
- HashiCorp Nomad by HTTP
- HashiCorp Vault by HTTP
- Hikvision camera by HTTP
- HPE iLO by HTTP
- HPE MSA 2040 Storage by HTTP
- HPE MSA 2060 Storage by HTTP
- HPE Primera by HTTP
- HPE Synergy by HTTP
- InfluxDB by HTTP
- Jenkins by HTTP
- Kubernetes API server by HTTP, Kubernetes cluster state by HTTP, Kubernetes Controller manager by HTTP, Kubernetes kubelet by HTTP, Kubernetes nodes by HTTP, Kubernetes Scheduler by HTTP sind veraltet und wurden durch die Vorlagen *Kubernetes Cluster by HTTP* und *Kubernetes Node by HTTP* ersetzt.
- Kubernetes Cluster by HTTP
- Kubernetes Node by HTTP
- MantisBT by HTTP
- Microsoft 365 reports by HTTP
- Microsoft SharePoint by HTTP
- NetApp AFF A700 by HTTP
- Nextcloud by HTTP
- NGINX by HTTP
- NGINX Plus by HTTP
- Nutanix Prism Element by HTTP
- OpenAI Platform by HTTP
- OpenStack by HTTP
- OpenWeatherMap by HTTP
- Oracle Cloud by HTTP
- Palo Alto PA-440 by HTTP
- PHP-FPM by HTTP
- Podman by HTTP
- Proxmox VE by HTTP
- Pure Storage FlashArray v1 and v2 by HTTP
- RabbitMQ cluster by HTTP
- Ribbon SBC Edge by HTTP
- Ribbon SBC SWe CE by HTTP
- Ribbon SBC SWe Core by HTTP
- TiDB by HTTP
- TiDB PD by HTTP
- TiDB TiKV by HTTP
- Travis CI by HTTP
- Veeam Backup Enterprise Manager by HTTP
- Veeam Backup and Replication by HTTP

- VeloCloud SD-WAN by HTTP
- VeloCloud SD-WAN Edge by HTTP
- YugabyteDB by HTTP
- ZooKeeper by HTTP

IPMI-Vorlage verwenden

IPMI-Vorlagen erfordern keine spezielle Einrichtung. Um mit der Überwachung zu beginnen, verknüpfen Sie die Vorlage mit einem Ziel-Host (falls die Vorlage in Ihrer Zabbix-Installation nicht verfügbar ist, müssen Sie die Vorlage möglicherweise zuerst importieren).

Eine detaillierte Beschreibung einer Vorlage, einschließlich der vollständigen Liste der Makros, Datenpunkte und Auslöser, finden Sie in der README-Datei der Vorlage (zugänglich durch Klicken auf den Vorlagennamen).

Verfügbare Vorlage:

- Chassis by IPMI

JMX-Vorlage verwenden

Schritte, um den korrekten Betrieb von Vorlagen sicherzustellen, die Metriken über JMX erfassen:

1. Stellen Sie sicher, dass das Zabbix Java gateway installiert und korrekt eingerichtet ist.
2. Verknüpfen Sie die Vorlage mit dem Ziel-Host. Der Host sollte über die konfigurierte JMX-Schnittstelle verfügen. Wenn die Vorlage in Ihrer Zabbix-Installation nicht verfügbar ist, müssen Sie die Vorlage möglicherweise zuerst importieren.
3. Passen Sie bei Bedarf die Werte der Vorlagenmakros an.
4. Konfigurieren Sie die zu überwachende Instanz so, dass sie den Datenaustausch mit Zabbix zulässt.

Eine detaillierte Beschreibung einer Vorlage, einschließlich der vollständigen Liste der Makros, Datenpunkte und Auslöser, finden Sie in der README-Datei der Vorlage (zugänglich durch Klicken auf den Vorlagennamen).

Die folgenden Vorlagen sind verfügbar:

- 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

ODBC-Vorlagenbetrieb

Schritte, um den korrekten Betrieb von Vorlagen sicherzustellen, die Metriken über ODBC-Monitoring erfassen:

1. Stellen Sie sicher, dass der erforderliche ODBC-Treiber auf dem Zabbix Server oder Proxy installiert ist.
2. Verknüpfen Sie die Vorlage mit einem Ziel-Host (falls die Vorlage in Ihrer Zabbix-Installation nicht verfügbar ist, müssen Sie die Vorlage möglicherweise zuerst importieren).
3. Passen Sie bei Bedarf die Werte der Vorlagenmakros an.
4. Konfigurieren Sie die zu überwachende Instanz so, dass sie den Datenaustausch mit Zabbix erlaubt.

Eine detaillierte Beschreibung einer Vorlage, einschließlich der vollständigen Liste der Makros, Datenpunkte und Auslöser, ist in der README-Datei der Vorlage verfügbar (zugänglich durch Klicken auf den Vorlagennamen).

Die folgenden Vorlagen sind verfügbar:

- MariaDB by ODBC
- MSSQL by ODBC
- MySQL by ODBC
- Oracle by ODBC
- Percona by ODBC
- PostgreSQL by ODBC

Standardisierte Vorlagen für Netzwerkgeräte

Übersicht

Um Monitoring für Netzwerkgeräte wie Switches und Router bereitzustellen, haben wir zwei sogenannte Modelle erstellt: eines für das Netzwerkgerät selbst (im Wesentlichen sein Gehäuse) und eines für die Netzwerkschnittstelle.

Für viele Familien von Netzwerkgeräten werden Vorlagen bereitgestellt. Alle Vorlagen decken, soweit diese Daten vom Gerät ermittelt werden können, Folgendes ab:

- Überwachung von Gehäusefehlern (Netzteile, Lüfter und Temperatur, Gesamtstatus)
- Überwachung der Gehäuseleistung (CPU- und Speicher-Datenpunkte)
- Erfassung des Gehäuse-Inventars (Seriennummern, Modellname, Firmware-Version)
- Überwachung der Netzwerkschnittstelle mit IF-MIB und EtherLike-MIB (Schnittstellenstatus, Schnittstellenverkehrslast, Duplex-Status für Ethernet)

Wenn die Vorlage in Ihrer Zabbix-Installation nicht verfügbar ist, müssen Sie die Vorlage möglicherweise zuerst importieren.

Wenn Sie die neuen Out-of-the-box-Vorlagen importieren, sollten Sie möglicherweise auch den globalen regulären Ausdruck `@Network interfaces for discovery` auf Folgendes aktualisieren:

```
Does not match regular expression: ^Software Loopback Interface
Does not match regular expression: ^(In)?[lL]oop[bB]ack[0-9._]*$
Does not match regular expression: ^NULL[0-9.]*$
Does not match regular expression: ^[lL]o[0-9.]*$
Does not match regular expression: ^[sS]ystem$
Does not match regular expression: ^Nu[0-9.]*$
```

um Loopback- und Null-Schnittstellen auf den meisten Systemen herauszufiltern.

Geräte

Liste der Gerätefamilien, für die Vorlagen verfügbar sind:

Name der Vorlage	Hersteller	Gerätefamilie	Bekannte Modelle	OS	Verwendete MIBs	**Tags**
Alcatel Timetra TiMOS SNMP	Alcatel	Alcatel Timetra	ALCATEL SR 7750	TiMOS	TIMETRA-SYSTEM-MIB,TIMETRA-CHASSIS-MIB	Zertifiziert
Aruba CX 8300s by SNMP	HPE Aruba Network-ing	Aruba CX Switch-Serie	Aruba JL636A 8325, Aruba JL717A 8360	Aruba AOS-CX	ARUBAWIRED-FAN-MIB, ARUBAWIRED-POWERSUPPLY-MIB, ARUBAWIRED-SYSTEMINFO-MIB, ARUBAWIRED-TEMPSENSOR-MIB, OSPF-MIB	Zertifiziert
Brocade FC SNMP	Brocade	Brocade-FC-Switches	Brocade 300 SAN Switch-	-	SW-MIB,ENTITY-MIB	Leistung Fehler
Brocade_Foundry Stackable SNMP	Brocade	Brocade ICX	Brocade ICX6610, Brocade ICX7250-48, Brocade ICX7450-48F		FOUNDRY-SN-AGENT-MIB, FOUNDRY-SN-STACKING-MIB	Zertifiziert
Brocade_Foundry Nonstackable SNMP	Brocade, Foundry	Brocade MLX, Foundry	Brocade MLXe, Foundry FLS648, Foundry FWSX424		FOUNDRY-SN-AGENT-MIB	Leistung Fehler
Check Point Next Generation Firewall by SNMP	Check Point	Next-Generation-Firewall	-	Gaia	HOST-RESOURCES-MIB, CHECKPOINT-MIB, UCD-SNMP-MIB, SNMPv2-MIB, IF-MIB	Zertifiziert

Name der Vorlage	Hersteller	Gerätefamilie	Bekannte Modelle	OS	Verwendete MIBs	**Tags**
Ciena 3906 by SNMP	Ciena	Ciena CPE	Ciena 3906	SAOS	WWP-LEOS-BLADE-MIB, WWP-LEOS-CHASSIS-MIB, WWP-LEOS-SYSTEM-CONFIG-MIB	Zertifiziert
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	Zertifiziert
Cisco IOS SNMP	Cisco	Cisco IOS ver > 12.2 3.5	Cisco C2950	IOS	CISCO-PROCESS-MIB,CISCO-MEMORY-POOL-MIB,CISCO-ENVMON-MIB	Zertifiziert
Cisco IOS versions 12.0_3_T-12.2_3.5 SNMP	Cisco	Cisco IOS > 12.0 3 T und 12.2 3.5	-	IOS	CISCO-PROCESS-MIB,CISCO-MEMORY-POOL-MIB,CISCO-ENVMON-MIB	Zertifiziert
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	Zertifiziert
D-Link DES_DGS Switch SNMP	D-Link	DES/DGX-Switches	D-Link DES-xxxx/DGS-xxxx,DLINK DGS-3420-26SC	-	DLINK-AGENT-MIB,EQUIPMENT-MIB,ENTITY-MIB	Zertifiziert
D-Link DES 7200 SNMP	D-Link	DES-7xxx	D-Link DES 7206	-	ENTITY-MIB,MY-SYSTEM-MIB,MY-PROCESS-MIB,MY-MEMORY-MIB	Leistung Fehler Schnittstellen
Dell Force S-Series SNMP	Dell	Dell Force S-Serie	S4810		F10-S-SERIES-CHASSIS-MIB	Zertifiziert
Extreme Exos SNMP	Extreme	Extreme EXOS	X670V-48x	EXOS	EXTREME-SYSTEM-MIB,EXTREME-SOFTWARE-MONITOR-MIB	Zertifiziert
FortiGate by SNMP	Fortinet	FortiGate (NGFW)	-	FortiOS	HOST-RESOURCES-MIB FORTINET-FORTIGATE-MIB FORTINET-CORE-MIB SNMPv2-MIB IF-MIB ENTITY-MIB	Leistung Inventar
HP Comware HH3C SNMP	HP	HP (H3C) Comware	HP A5500-24G-4SFP HI Switch		HH3C-ENTITY-EXT-MIB,ENTITY-MIB	Zertifiziert

Name der Vorlage	Hersteller	Gerätefamilie	Bekannte Modelle	OS	Verwendete MIBs	**Tags**
HP Enterprise Switch SNMP	HP	HP Enterprise Switch	HP ProCurve J4900B Switch 2626, HP J9728A 2920-48G Switch		STATISTICS-MIB,NETSWITCH-MIB,HP-ICF-CHASSIS,ENTITY-MIB,SEMI-MIB	Zertifiziert
Huawei AR600 by SNMP	Huawei	Huawei NetEngine AR600 Series Router	Huawei AR600/AR6100/AR6200/AR6300 Series Router mit VRP-Software (V300R019/V300R021/V300R023 oder höher)		HUAWEI-CBQOS-MIB, HUAWEI-NQA-MIB, ENTITY-MIB, IF-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Zertifiziert
Huawei OceanStor 5300 V5 by SNMP	Huawei	Huawei OceanStor Dorado	Huawei OceanStor Dorado mit V5-Software		HUAWEI-STORAGE-HARDWARE-MIB, HUAWEI-STORAGE-SPACE-MIB, ISM-PERFORMANCE-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Zertifiziert
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 mit V6/V700-Software		HUAWEI-STORAGE-HARDWARE-MIB, HUAWEI-STORAGE-SPACE-MIB, ISM-PERFORMANCE-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Zertifiziert
Huawei VRP by SNMP	Huawei	Huawei VRP	S2352P-EI		ENTITY-MIB,HUAWEI-ENTITY-EXTENT-MIB	Zertifiziert
Intel_Qlogic Infiniband SNMP	Intel/QLogic	Intel/QLogic Infiniband-Geräte	Infiniband 12300		ICS-CHASSIS-MIB	Fehler Inventar
Juniper SNMP	Juniper	MX-, SRX-, EX-Modelle	Juniper MX240, Juniper EX4200-24F	JunOS	JUNIPER-MIB	Zertifiziert
Juniper MX NETCONF	Juniper	MX-Modelle	Juniper MX204 Edge Router	JunOS 24.2R1-S1.10	*Nicht zutreffend (verwendet NETCONF über SSH)*	Zertifiziert
Juniper MX SNMP	Juniper	MX-Modelle	Juniper MX204 Edge Router	JunOS 24.2R1-S1.10	OSPF-MIB, JUNIPER-DOM-MIB, JUNIPER-MIB, BGP4-V2-MIB-JUNIPER, OSPFV3-MIB-JUNIPER	Zertifiziert

Name der Vorlage	Hersteller	Gerätefamilie	Bekannte Modelle	OS	Verwendete MIBs	**Tags**
Mellanox SNMP	Mellanox	Mellanox-Infiniband-Geräte	SX1036	MLNX-OS	HOST-RESOURCES-MIB,ENTITY-MIB,ENTITY-SENSOR-MIB,MELLANOX-MIB	Zertifiziert
MikroTik CCR<device model> SNMP	MikroTik	MikroTik Cloud Core Router (CCR-Serie)	Separate dedizierte Vorlagen sind verfügbar für 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	Zertifiziert

Name der Vorlage	Hersteller	Gerätefamilie	Bekannte Modelle	OS	Verwendete MIBs	**Tags**
MikroTik CRS<device model> SNMP	MikroTik	MikroTik Cloud Router Switches (CRS-Serie)	Separate dedizierte Vorlagen sind verfügbar für 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	Zertifiziert
MikroTik CSS<device model> SNMP	MikroTik	MikroTik Cloud Smart Switches (CSS-Serie)	Separate dedizierte Vorlagen sind verfügbar für MikroTik CSS326-24G-2S+RM, MikroTik CSS610-8G-2S+IN	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Zertifiziert
MikroTik FiberBox SNMP	MikroTik	MikroTik FiberBox	MikroTik FiberBox	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Zertifiziert
MikroTik hEX <device model> SNMP	MikroTik	MikroTik hEX	Separate dedizierte Vorlagen sind verfügbar für MikroTik hEX, MikroTik hEX lite, MikroTik hEX PoE, MikroTik hEX PoE lite, MikroTik hEX S	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Zertifiziert

Name der Vorlage	Hersteller	Gerätefamilie	Bekannte Modelle	OS	Verwendete MIBs	**Tags**
MikroTik netPower <device model> SNMP	MikroTik	MikroTik net-Power	Separate dedizierte Vorlagen sind verfügbar für MikroTik netPower 15FR, MikroTik netPower 16P SNMP, MikroTik netPower Lite 7R	RouterOS/SwitchOS, SwitchOS Lite	MIKROTIK-MIB,HOST-RESOURCES-MIB	Zertifiziert
MikroTik PowerBox <device model> SNMP	MikroTik	MikroTik Power-Box	Separate dedizierte Vorlagen sind verfügbar für MikroTik PowerBox, MikroTik PowerBox Pro	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Zertifiziert
MikroTik RB<device model> SNMP	MikroTik	MikroTik Router der RB-Serie	Separate dedizierte Vorlagen sind verfügbar für 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	Zertifiziert
MikroTik SNMP	MikroTik	MikroTik RouterOS-Geräte	MikroTik CCR1016-12G, MikroTik RB2011UAS-2HnD, MikroTik 912UAG-5HPnD, MikroTik 941-2nD, MikroTik 951G-2HnD, MikroTik 1100AHx2	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Zertifiziert
Netgear Fastpath SNMP	Netgear	Netgear Fastpath	M5300-28G		FASTPATH-SWITCHING-MIB,FASTPATH-BOXSERVICES-PRIVATE-MIB	Fehler Inventar

Name der Vorlage	Hersteller	Gerätefamilie	Bekannte Modelle	OS	Verwendete MIBs	**Tags**
QTech QSW SNMP	QTech	Qtech-Geräte	Qtech QSW-2800-28T	-	QTECH-MIB,ENTITY-MIB	Leistung Inventar
Stormshield SNS by SNMP	Stormshield	Stormshield Network Security (SNS)-Firewalls	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	Zertifiziert
TP-LINK SNMP	TP-LINK	TP-LINK	T2600G-28TS v2.0		TPLINK-SYSMONITOR-MIB,TPLINK-SYSINFO-MIB	Leistung Inventar
Ubiquiti AirOS SNMP	Ubiquiti	Ubiquiti AirOS-WLAN-Geräte	NanoBridge,NanoStation,UniFi	AirOS	FROGFOOT-RESOURCES-MIB,IEEE802dot11-MIB	Leistung
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	Leistung Inventar

Vorlagendesign

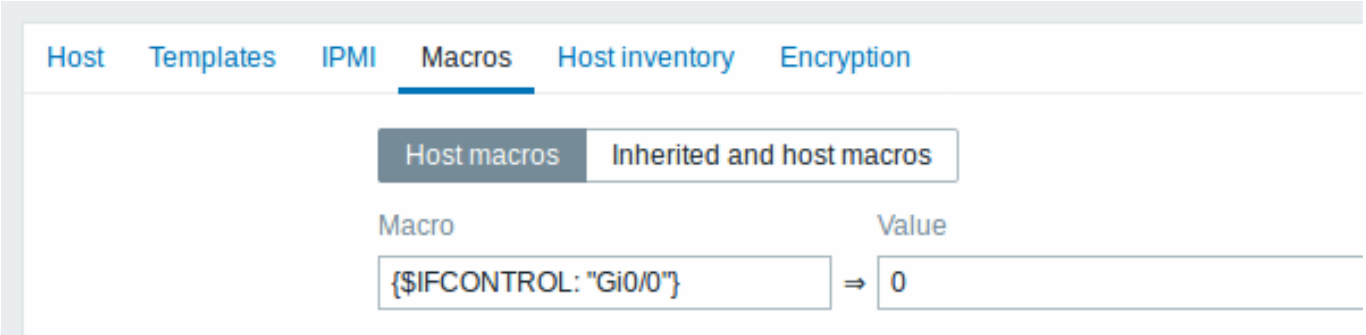
Vorlagen wurden unter Berücksichtigung der folgenden Punkte entwickelt:

- Benutzermakros werden so weit wie möglich verwendet, damit Auslöser vom Benutzer angepasst werden können;
- Low-Level-Discovery wird so weit wie möglich verwendet, um die Anzahl nicht unterstützter Datenpunkte zu minimieren;
- Alle Vorlagen hängen von Template ICMP Ping ab, sodass alle Geräte zusätzlich per ICMP geprüft werden;
- Datenpunkte verwenden keine MIBs - in Datenpunkten und Low-Level-Discoverys werden SNMP-OIDs verwendet. Daher ist es nicht erforderlich, MIBs in Zabbix zu laden, damit Vorlagen funktionieren;
- Loopback-Netzwerkschnittstellen werden bei der Erkennung herausgefiltert, ebenso wie Schnittstellen mit ifAdminStatus = down(2)
- Nach Möglichkeit werden 64-Bit-Zähler aus IF-MIB::ifXTable verwendet. Falls diese nicht unterstützt werden, werden stattdessen die standardmäßigen 32-Bit-Zähler verwendet.

Alle erkannten Netzwerkschnittstellen verfügen über einen Auslöser, der ihren Betriebsstatus (Link) überwacht, zum Beispiel:

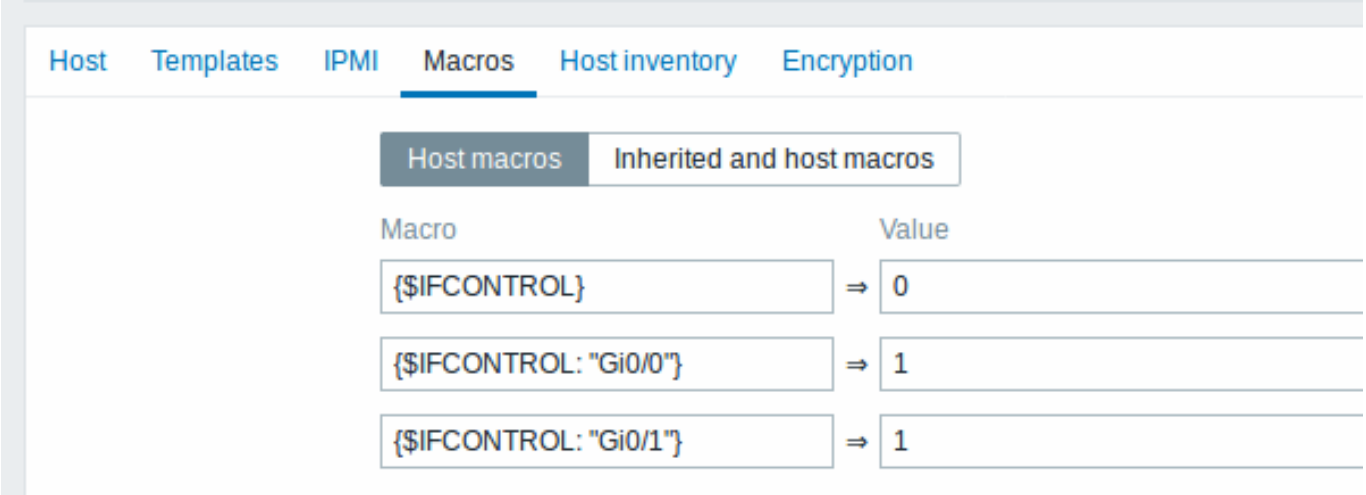
```
{$IFCONTROL:"{#IFNAME}"}=1 and last(/Alcatel Timetra TiMOS SNMP/net.if.status[ifOperStatus.{#SNMPINDEX}])
```

- Wenn Sie diesen Zustand für eine bestimmte Schnittstelle nicht überwachen möchten, erstellen Sie ein Benutzermakro mit Kontext und dem Wert 0. Zum Beispiel:



wobei Gi0/0 {#IFNAME} ist. Auf diese Weise wird der Auslöser für diese bestimmte Schnittstelle nicht mehr verwendet.

- Sie können auch das Standardverhalten so ändern, dass alle Auslöser standardmäßig nicht ausgelöst werden, und diesen Auslöser nur für eine begrenzte Anzahl von Schnittstellen wie Uplinks aktivieren:



Tags

- Leistung - MIBs der Gerätefamilie bieten eine Möglichkeit, CPU- und Speicherdatenpunkte zu überwachen;
- Fehler - MIBs der Gerätefamilie bieten eine Möglichkeit, mindestens einen Temperatursensor zu überwachen;
- Inventar - MIBs der Gerätefamilie bieten eine Möglichkeit, mindestens die Seriennummer und den Modellnamen des Geräts zu erfassen;
- Zertifiziert - alle drei oben genannten Hauptkategorien werden abgedeckt.

VMware-Vorlagenoperation

Übersicht

Zabbix enthält eine Reihe sofort einsatzbereiter Vorlagen für die Überwachung virtueller Maschinen, die für VMware vCenter- und ESXi-Hypervisoren entwickelt wurden. Die verfügbaren Vorlagen sind in zwei separate Vorlagensätze unterteilt:

- VMware - verwendet UUID-Daten für entsprechende Makros
- VMware FQDN - verwendet FQDN-Daten für entsprechende Makros

Die für die Überwachung von VMware vCenter- oder ESXi-Hypervisoren entwickelten Vorlagen enthalten vorkonfigurierte Low-Level-Discovery-Regeln sowie verschiedene integrierte Prüfungen zur Überwachung virtueller Installationen.

Note:
Für die korrekte Funktion der Vorlage *VMware FQDN* sollte jede überwachte virtuelle Maschine einen eindeutigen OS-Namen haben, der den FQDN-Regeln entspricht. Zusätzlich müssen VMware Tools/Open Virtual Machine Tools auf jeder Maschine installiert sein. Wenn diese Voraussetzungen erfüllt sind, wird die Verwendung der Vorlage *VMware FQDN* empfohlen. Die Vorlage *VMware FQDN* ist seit Zabbix 5.2 verfügbar, als die Möglichkeit eingeführt wurde, Hosts mit benutzerdefinierten Schnittstellen zu erstellen.

 Eine klassische Vorlage *VMware* ist ebenfalls verfügbar und kann verwendet werden, wenn die FQDN-Anforderungen nicht erfüllt sind. Die Vorlage *VMware* hat jedoch ein bekanntes Problem. Hosts für erkannte virtuelle Maschinen werden mit Namen erstellt, die in vCenter gespeichert sind (zum Beispiel „VM1“, „VM2“ usw.). Wenn auf diesen Hosts der Zabbix-Agent installiert ist und die aktive automatische Registrierung des Zabbix-Agenten aktiviert ist, liest der Autoregistrierungsprozess die Hostnamen so, wie sie beim Start registriert wurden (zum Beispiel „vm1.example.com“, „vm2.example.com“ usw.). Dies kann zur Erstellung neuer Hosts für bereits vorhandene virtuelle Maschinen führen (da keine Namensübereinstimmungen gefunden wurden), was zu doppelten Hosts mit unterschiedlichen Namen führt.

Bitte beachten Sie:

- Die Vorlage *VMware Hypervisor* kann manuell mit einem Host verknüpft sowie in der Discovery verwendet werden;
- Die Vorlage *VMware Guest* sollte nicht manuell mit einem Host verknüpft werden und kann nur in der Discovery verwendet werden.

Konfiguration von Host-Makros

Um einfache VMware-Prüfungen zu verwenden, muss der Host die folgenden Benutzermakros definiert haben:

- {$VMWARE.URL} - SDK-URL des VMware-Dienstes (vCenter oder ESXi-Hypervisor) (https://servername/sdk)
- {$VMWARE.USERNAME} - Benutzername des VMware-Dienstes
- {$VMWARE.PASSWORD} - Passwort des Benutzers {$VMWARE.USERNAME} des VMware-Dienstes

OTLP-Vorlagenbetrieb

Zabbix stellt eine Reihe sofort einsatzbereiter Vorlagen zum Erfassen und Überwachen von OpenTelemetry-Traces, -Metriken und -Protokollen mithilfe von *Telemetry query*-Datenpunkten bereit.

Schritte zur Sicherstellung des korrekten Betriebs der Vorlagen:

1. Prüfen Sie die Datei `README` der Vorlage auf vorlagenspezifische Konfigurationsanforderungen.
2. Erstellen Sie in Zabbix einen Host, der dem Überwachungsziel entspricht.
3. Stellen Sie sicher, dass der Parameter `TelemetryProvider` in der Konfigurationsdatei des Zabbix-Servers oder Zabbix-Proxys konfiguriert ist.
4. Verknüpfen Sie die Vorlage mit dem erstellten Host. (Falls die Vorlage in Ihrer Zabbix-Installation nicht verfügbar ist, müssen Sie sie möglicherweise zuerst importieren.)
5. Passen Sie bei Bedarf die Werte der Makros auf Vorlagenebene an.

Eine ausführliche Beschreibung der Vorlagen, einschließlich der vollständigen Liste der Makros, Datenpunkte und Auslöser, finden Sie in der Datei `README` der jeweiligen Vorlage. Sie können darauf zugreifen, indem Sie auf den Namen der Vorlage klicken.

Die folgenden Vorlagen sind verfügbar:

- Envoy Proxy by OTLP
- Generic OpenTelemetry by OTLP
- OpenRouter traces by OTLP
- Proxmox VE by OTLP

Benachrichtigungen bei Ereignissen

Übersicht

Angenommen, wir haben einige Datenpunkte und Auslöser konfiguriert und erhalten nun einige Ereignisse, die dadurch entstehen, dass Auslöser ihren Status ändern, dann ist es an der Zeit, einige Aktionen in Betracht zu ziehen.

Zunächst möchten wir nicht ständig auf die Liste der Auslöser oder Ereignisse schauen. Es wäre viel besser, eine Benachrichtigung zu erhalten, wenn etwas Bedeutendes (z. B. ein Problem) passiert ist. Außerdem möchten wir bei auftretenden Problemen sehen, dass alle betroffenen Personen informiert werden.

Deshalb ist das Senden von Benachrichtigungen eine der wichtigsten von Zabbix angebotenen Aktionen. Es kann festgelegt werden, wer und wann bei einem bestimmten Ereignis benachrichtigt werden soll.

Um Benachrichtigungen von Zabbix senden und empfangen zu können, müssen Sie:

- einige Medien definieren
- eine Aktion konfigurieren, die eine Nachricht an eines der definierten Medien sendet

Aktionen bestehen aus *Bedingungen* und *Operationen*. Grundsätzlich werden Operationen ausgeführt, wenn die Bedingungen erfüllt sind. Die zwei wichtigsten Operationen sind das Senden einer Nachricht (Benachrichtigung) und das Ausführen eines Remote-Befehls.

Für durch Discovery und Autoregistrierung erzeugte Ereignisse stehen einige zusätzliche Operationen zur Verfügung. Dazu gehören das Hinzufügen oder Entfernen eines Hosts, das Verknüpfen einer Vorlage usw.

Medientypen

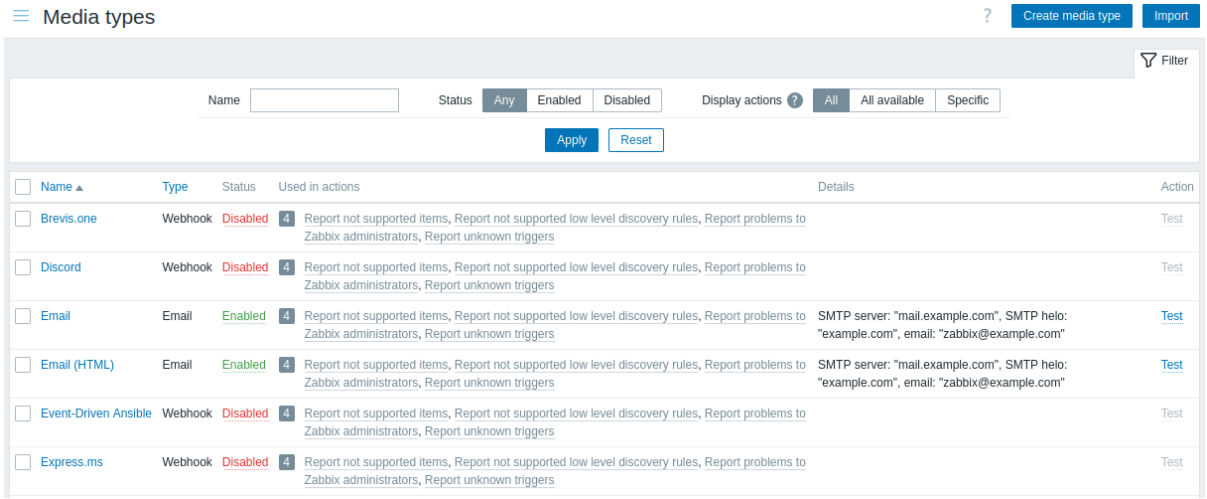
Übersicht

Medientypen sind die Zustellkanäle, die zum Senden von Benachrichtigungen und Warnungen aus Zabbix verwendet werden.

Medientypen unterstützen die folgenden Zustellmethoden:

- E-Mail
- Push-Benachrichtigung
- SMS
- Benutzerdefiniertes Skript
- Webhook

Medientypen werden unter *Alarme* > *Medientypen* verwaltet. Einige Medientypen sind im Standarddatensatz vordefiniert. Sie müssen lediglich ihre Parameter anpassen, damit sie funktionieren.



Um zu sehen, wie Medientypen in den Alarmierungsprozess eingebunden sind, betrachten wir die drei Voraussetzungen für die Zustellung von Benachrichtigungen aus Zabbix an Endbenutzer:

1. Eine Operation einer Aktion muss definiert werden, die Benachrichtigungen sendet.
2. Ein funktionierender Medientyp muss definiert werden (z. B. *E-Mail*, über die Warnungen mithilfe von SMTP gesendet werden).
3. Zustelldetails auf Benutzerebene (z. B. E-Mail-Adressen, Telefonnummern usw.) müssen unter Benutzermedien definiert werden.

Testen von Medientypen

Um zu testen, ob ein konfigurierter Medientyp funktioniert, klicken Sie in der Liste der Medientypen auf *Test*.

Die Testanfrage wird an den Zabbix Server gesendet. Der Zabbix Server versucht, mithilfe des angegebenen Medientyps eine Benachrichtigung zu senden, und gibt das Ergebnis an das Frontend zurück. Das Frontend wartet, bis der Server die Ergebnisse zurückgibt. Das Testen von Medientypen hat standardmäßig ein Zeitlimit von 65 Sekunden (konfigurierbar unter *Administration* > *Allgemein* > *Zeitüberschreitungen*).

Siehe auch die Testdetails für:

- E-Mail
- Push-Benachrichtigung
- Webhook
- Skript

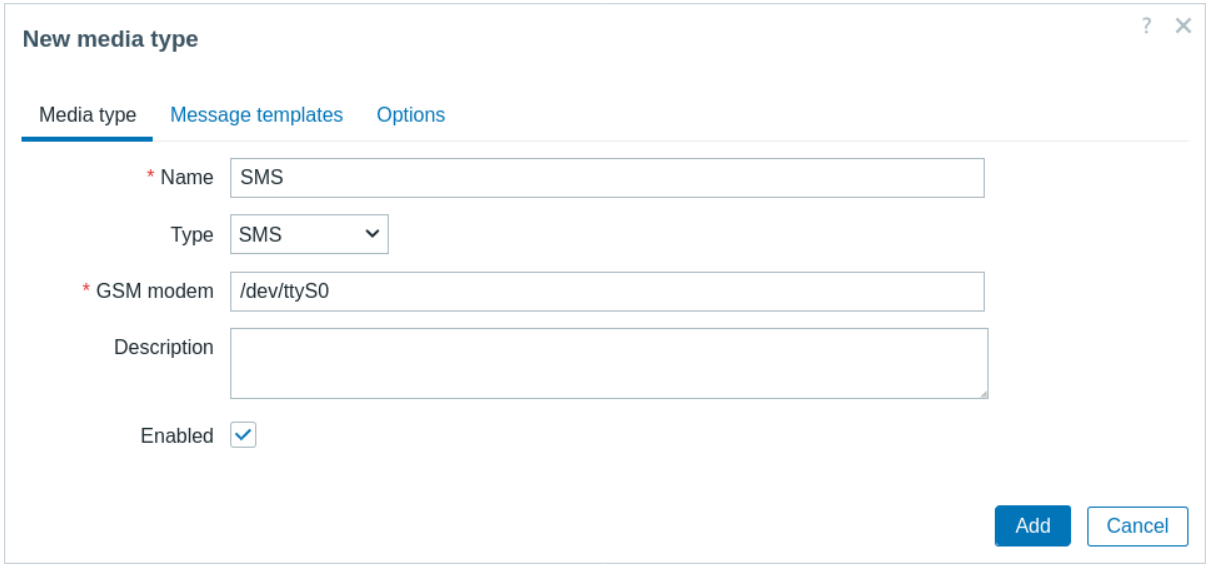
Konfiguration

So erstellen Sie einen Medientyp im Zabbix Frontend:

1. Gehen Sie zu *Alarme* > *Medientypen*.
2. Klicken Sie auf *Medientyp erstellen*.
3. Geben Sie die Parameter des Medientyps im Formular ein.

Einige Parameter sind für alle Zustellungsmethoden **allgemeingültig**.

Note:
Wenn ein oder mehrere Zustellungsmethoden im Feature-Flag `$ZBX_FEATURE_FLAGS['media_type_denylist']` in der Frontend-Konfigurationsdatei (`zabbix.conf.php`) angegeben sind, sind diese Zustellungsmethoden in der Liste *Typ* nicht verfügbar.

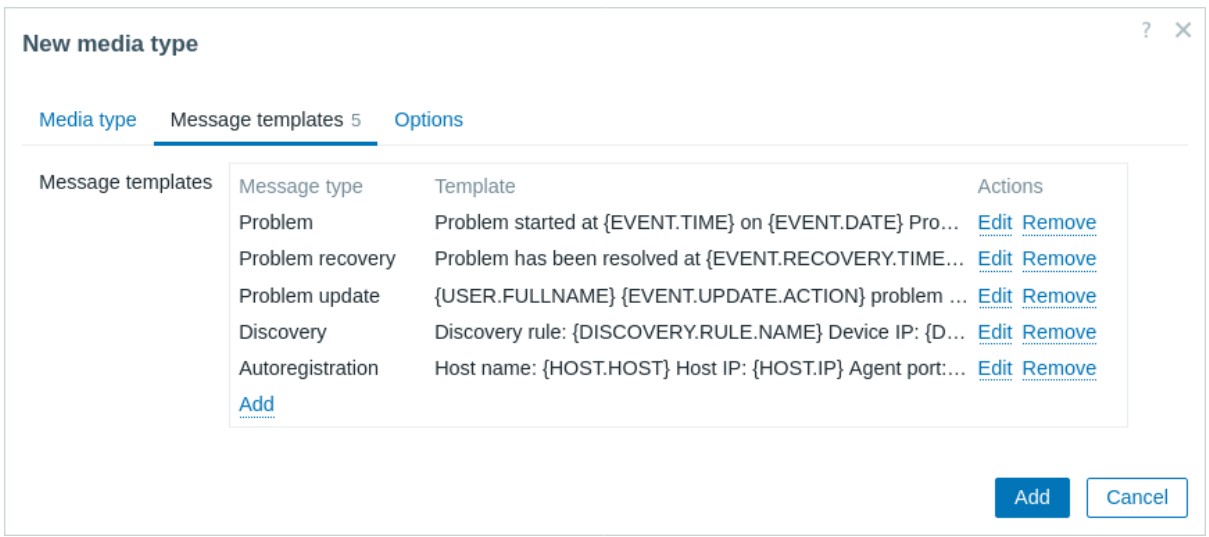


Parameter	Beschreibung
Name	Name des Medientyps.
Typ	Wählen Sie die Zustellungsmethode für den Medientyp aus.
Beschreibung	Geben Sie eine Beschreibung für den Medientyp ein.
Aktiviert	Aktivieren Sie das Kontrollkästchen, um den Medientyp zu aktivieren.

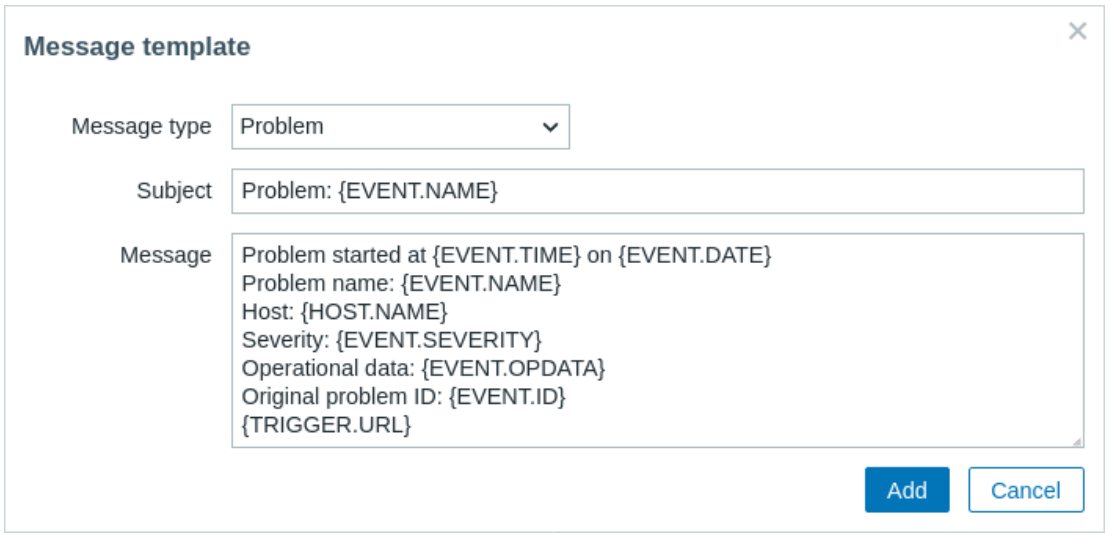
Methodenspezifische Parameter finden Sie auf den Seiten zu E-Mail, Push-Benachrichtigungen, SMS, benutzerdefiniertem Alertscript oder webhook.

Nachrichtenvorlagen

Die Registerkarte **Nachrichtenvorlagen** enthält Standardnachrichten pro Ereignistyp (Problem, Problembehebung, Erkennung usw.).



Klicken Sie auf **Hinzufügen**, um eine Standardnachricht zu definieren (oder auf **Bearbeiten**, um eine vorhandene Nachricht zu aktualisieren):



Parameter	Beschreibung
Nachrichtentyp	Typ eines Ereignisses, für das die Standardnachricht verwendet werden soll. Für jeden Ereignistyp kann nur eine Standardnachricht definiert werden.
Betreff	Betreff der Standardnachricht. Der Betreff kann Makros enthalten. Er ist auf 255 Zeichen begrenzt. Der Betreff ist für den SMS-Medientyp nicht verfügbar.
Nachricht	Die Standardnachricht. Sie ist je nach Datenbanktyp auf eine bestimmte Anzahl von Zeichen begrenzt (weitere Informationen finden Sie unter Nachrichten senden). Die Nachricht kann unterstützte Makros enthalten. In Nachrichten zu Problemen und Problemaktualisierungen werden Ausdrucksmakros unterstützt (zum Beispiel `{?avg(/host/key,1h)}`).

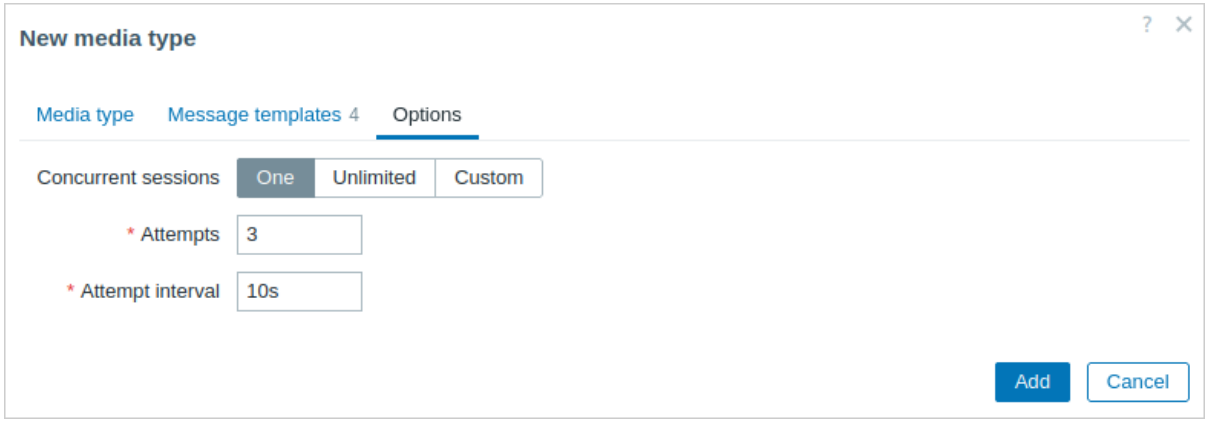
Beachten Sie, dass Standardnachrichten durch benutzerdefinierte Nachrichten überschrieben werden, sofern diese in Aktionsoperationen definiert sind.

Warning:
Das Definieren von Nachrichtenvorlagen ist für alle Zustellmethoden obligatorisch, einschließlich webhooks oder benutzerdefinierter Alarmskripte, die keine Standardnachrichten für Benachrichtigungen verwenden. Beispielsweise schlägt die Aktion "Nachricht an Pushover webhook senden" fehl, wenn die Problemnachricht für den Pushover webhook nicht definiert ist.

Optionen

Die Registerkarte **Optionen** enthält Einstellungen für die Alarmverarbeitung. Für jeden Medientyp kann derselbe Satz von Optionen konfiguriert werden.

Alle Medientypen werden parallel verarbeitet. Während die maximale Anzahl gleichzeitiger Sitzungen pro Medientyp konfiguriert werden kann, kann die Gesamtzahl der Alerter-Prozesse auf dem Server nur durch den Parameter StartAlerters begrenzt werden. Alarme, die von einem Auslöser generiert werden, werden sequenziell verarbeitet. Daher können mehrere Benachrichtigungen nur dann gleichzeitig verarbeitet werden, wenn sie von mehreren Auslösern generiert wurden.



Parameter	Beschreibung
Gleichzeitige Sitzungen	Wählen Sie die Anzahl paralleler Alerter-Sitzungen für den Medientyp aus: **Eine** - eine Sitzung **Unbegrenzt** - unbegrenzte Anzahl von Sitzungen **Benutzerdefiniert** - eine benutzerdefinierte Anzahl von Sitzungen auswählen Unbegrenzte/hohe Werte bedeuten mehr parallele Sitzungen und eine höhere Kapazität zum Senden von Benachrichtigungen. Unbegrenzte/hohe Werte sollten in großen Umgebungen verwendet werden, in denen möglicherweise viele Benachrichtigungen gleichzeitig gesendet werden müssen. Wenn mehr Benachrichtigungen gesendet werden müssen, als gleichzeitige Sitzungen vorhanden sind, werden die übrigen Benachrichtigungen in die Warteschlange gestellt; sie gehen nicht verloren.
Versuche	Anzahl der Versuche, eine Benachrichtigung zu senden. Es können bis zu 100 Versuche angegeben werden; der Standardwert ist „3“. Wenn „1“ angegeben wird, sendet Zabbix die Benachrichtigung nur einmal und versucht es nicht erneut, wenn das Senden fehlschlägt.
Versuchsintervall	Häufigkeit, mit der versucht wird, eine Benachrichtigung erneut zu senden, falls das Senden fehlgeschlagen ist, in Sekunden (0-3600). Wenn „0“ angegeben wird, versucht Zabbix es sofort erneut. Zeitsuffixe werden unterstützt (z. B. `30s`, `1m`, `1h`).

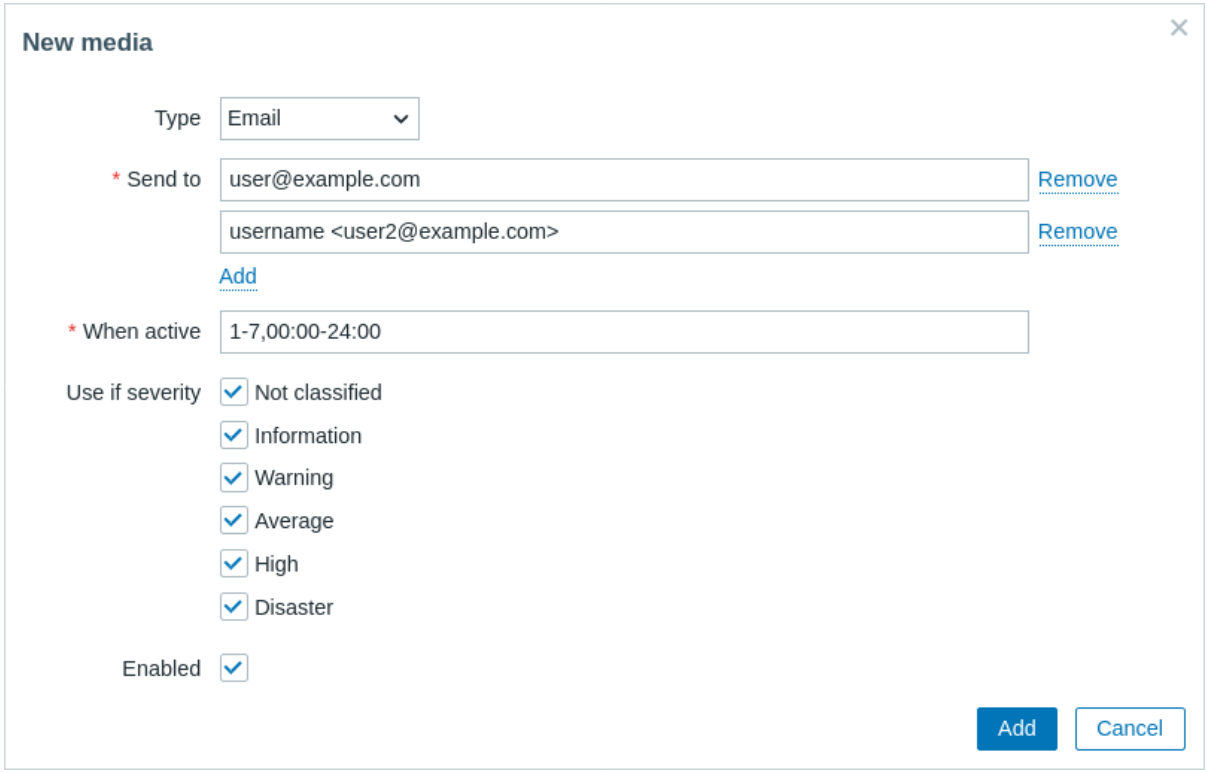
Benutzermedien

Während Medientypen festlegen, **wie** eine Benachrichtigung gesendet wird, legen Benutzermedien fest, **wohin** die Benachrichtigung gesendet werden muss.

Benutzermedien (z. B. E-Mail-Adresse, Benutzer-ID eines Webhooks usw.) müssen unabhängig von der Zustellmethode im Benutzerprofil definiert werden. Eine Aktion, die Nachrichten an den Benutzer *Admin* über den webhook *X* sendet, kann nicht zugestellt werden, wenn die Zustelldetails des webhooks *X* nicht im Benutzerprofil von Admin definiert sind.

So definieren Sie Benutzermedien:

1. Gehen Sie zu *Benutzer* > *Benutzer* und öffnen Sie das Formular mit den Benutzereigenschaften. Alternativ können Sie in Ihrem eigenen Benutzerprofil zu *Benutzereinstellungen* > *Benachrichtigungen* gehen.
2. Klicken Sie auf der Registerkarte *Medien* auf *Hinzufügen*.
3. Geben Sie die Parameter des Benutzermediums in das Formular ein.



Parameter	Beschreibung
Typ	Die Dropdown-Liste enthält die Namen der aktivierten Medientypen. Beachten Sie, dass beim Bearbeiten eines Mediums eines deaktivierten Medientyps der Typ rot angezeigt wird.
Senden an	Geben Sie die Kontaktinformationen ein, an die Nachrichten gesendet werden sollen. Beim Medientyp E-Mail können mehrere Adressen hinzugefügt werden, indem Sie auf die Schaltfläche Add unter dem Adressfeld klicken. In diesem Fall werden Benachrichtigungen an alle aufgeführten Adressen gesendet. Beispiele für Adressen finden Sie in der Beschreibung des Parameters *E-Mail* für den Medientyp E-Mail. Beim Medientyp Push-Benachrichtigung können Sie aus den aktiven Mobilgeräten des Benutzers auswählen oder ein oder mehrere bestimmte Geräte hinzufügen.
Aktiv, wenn	Sie können den Zeitraum begrenzen, in dem Nachrichten gesendet werden, z. B. nur auf Werktage (1-5,09:00-18:00). Beachten Sie, dass diese Einschränkung auf der Zeitzone des Benutzers basiert. Wenn die Zeitzone des Benutzers geändert wird und von der Zeitzone des Systems abweicht, muss diese Einschränkung möglicherweise entsprechend angepasst werden, damit wichtige Nachrichten nicht verpasst werden. Eine Beschreibung des Formats finden Sie auf der Seite Angabe von Zeiträumen. Benutzermakros werden unterstützt.
Verwenden bei Schweregrad	Aktivieren Sie die Kontrollkästchen der Auslöser-Schweregrade, für die Sie Benachrichtigungen erhalten möchten. *Hinweis*: Der Standardschweregrad („Nicht klassifiziert“) **muss** aktiviert sein, wenn Sie Benachrichtigungen für Ereignisse erhalten möchten, die nicht von Auslösern stammen. Nach dem Speichern werden die ausgewählten Auslöser-Schweregrade in den entsprechenden Schweregradfarben angezeigt, während nicht ausgewählte Schweregrade ausgegraut dargestellt werden.
Status	Status des Benutzermediums. **Aktiviert** – wird verwendet. **Deaktiviert** – wird nicht verwendet.

E-Mail

Übersicht

Um E-Mail als Zustellkanal für Nachrichten zu konfigurieren, müssen Sie E-Mail als Medientyp konfigurieren und Benutzern bestimmte Adressen zuweisen.

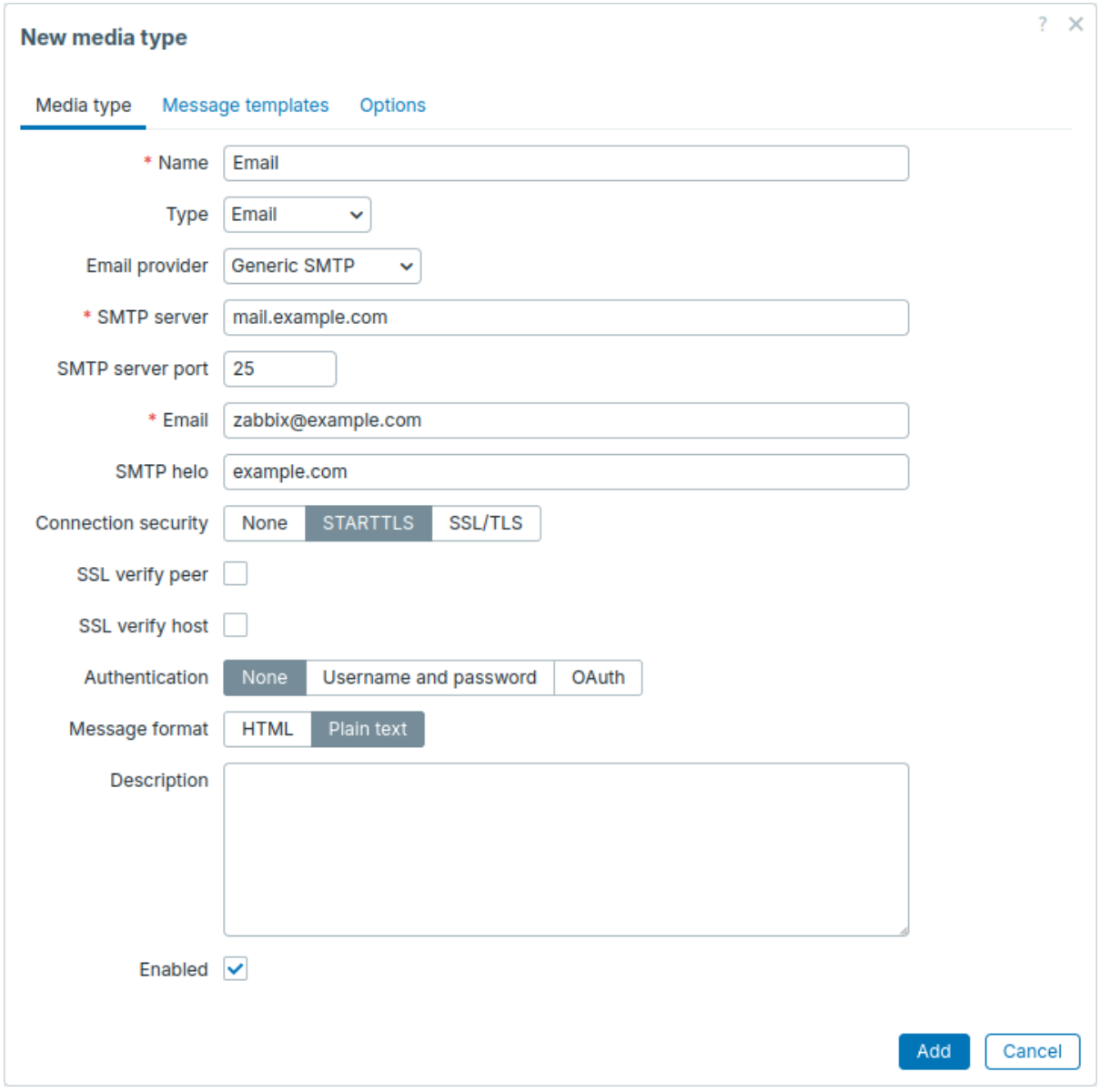
Note:
Mehrere Benachrichtigungen für ein einzelnes Ereignis werden im selben E-Mail-Thread zusammengefasst.

Konfiguration

So konfigurieren Sie E-Mail als Medientyp:

1. Gehen Sie zu *Alerts* > *Media types*.
2. Klicken Sie auf *Create media type* (oder klicken Sie in der Liste der vordefinierten Medientypen auf *Email*).
3. Geben Sie die Parameter des E-Mail-Medientyps im Formular ein.

Die Registerkarte **Media type** enthält allgemeine Attribute des Medientyps:



Alle Pflichtfelder sind mit einem roten Sternchen gekennzeichnet.

Warning:
Das Passwort wird beim Exportieren der Konfiguration des E-Mail-Medientyps im Klartext exportiert.

Die folgenden Parameter sind spezifisch für den E-Mail-Medientyp:

Parameter	Beschreibung
Email provider	Wählen Sie den E-Mail-Anbieter aus: *Generic SMTP*, *Gmail*, *Gmail relay*, *Office365* oder *Office365 relay*. Wenn Sie eine der Gmail-/Office365-bezogenen Optionen auswählen, müssen Sie nur die E-Mail-Adresse des Absenders und das Passwort angeben; Optionen wie *SMTP server*, *SMTP server port*, *SMTP helo* und *Connection security* werden von Zabbix automatisch ausgefüllt. Siehe auch: Automated Gmail/Office365 media types.
SMTP server	Geben Sie einen SMTP-Server an, der ausgehende Nachrichten verarbeitet. Dieses Feld ist verfügbar, wenn *Generic SMTP* als E-Mail-Anbieter ausgewählt ist.
SMTP server port	Geben Sie den SMTP-Serverport an, der ausgehende Nachrichten verarbeitet. Dieses Feld ist verfügbar, wenn *Generic SMTP* als E-Mail-Anbieter ausgewählt ist.
Email	Die hier eingegebene Adresse wird als **From**-Adresse für die gesendeten Nachrichten verwendet. Das Hinzufügen eines Anzeigenamens zum tatsächlichen E-Mail-Adresse (z. B. `Zabbix_info <zabbix@company.com>`) wird unterstützt. Für Anzeigenamen in Zabbix-E-Mails gelten im Vergleich zu den von RFC 5322 erlaubten Anzeigenamen einige Einschränkungen, wie die folgenden Beispiele zeigen. Gültige Beispiele: `zabbix@company.com` (nur eine E-Mail-Adresse; spitze Klammern sind nicht erforderlich) `Zabbix_info <zabbix@company.com>` (ein Anzeigename gefolgt von der E-Mail-Adresse in spitzen Klammern) `Ω-monitoring <zabbix@company.com>` (UTF-8-Zeichen im Anzeigenamen) Ungültige Beispiele: `Zabbix HQ zabbix@company.com` (ein Anzeigename ist vorhanden, aber die E-Mail-Adresse steht nicht in spitzen Klammern) `"Zabbix@<H(comment)Q>" <zabbix@company.com>` (obwohl dies gemäß RFC 5322 gültig ist, werden Anführungszeichen und Kommentare in Zabbix-E-Mails nicht unterstützt).
SMTP helo	Geben Sie einen korrekten SMTP-HELO-Wert an, normalerweise einen Domainnamen. Wenn das Feld leer ist, wird der Domainname der E-Mail gesendet (also der Teil nach `@` im Feld *Email*). Wenn der Domainname nicht ermittelt werden kann, wird eine Warnung auf Debug-Ebene protokolliert und der Hostname des Servers als Domain für den HELO-Befehl gesendet. Dieses Feld ist verfügbar, wenn *Generic SMTP* als E-Mail-Anbieter ausgewählt ist.
Connection security	Wählen Sie die Stufe der Verbindungssicherheit aus: **None** - die Option CURLOPT_USE_SSL nicht verwenden **STARTTLS** - die Option CURLOPT_USE_SSL mit dem Wert CURLUSESSL_ALL verwenden **SSL/TLS** - die Verwendung von CURLOPT_USE_SSL ist optional.
SSL verify peer	Aktivieren Sie das Kontrollkästchen, um das SSL-Zertifikat des SMTP-Servers zu überprüfen. Der Wert der Serverkonfigurationsdirektive "SSLCALocation" muss zur Zertifikatsvalidierung in CURLOPT_CAPATH eingetragen werden. Dadurch wird die cURL-Option CURLOPT_SSL_VERIFYPEER gesetzt.
SSL verify host	Aktivieren Sie das Kontrollkästchen, um zu überprüfen, ob das Feld *Common Name* oder das Feld *Subject Alternate Name* des SMTP-Serverzertifikats übereinstimmt. Dadurch wird die cURL-Option CURLOPT_SSL_VERIFYHOST gesetzt.
Authentication	Wählen Sie die Authentifizierungsstufe aus: **None** - es werden keine cURL-Optionen gesetzt **Username and password** - entspricht "AUTH=*", wobei cURL die Auswahl des Authentifizierungsmechanismus überlassen wird **OAuth** - OAuth-Authentifizierung Die OAuth-Authentifizierung wird für den E-Mail-Anbieter *Office365 relay* nicht unterstützt.
Username	Benutzernamen für die Authentifizierung. Dadurch wird der Wert von CURLOPT_USERNAME gesetzt. Benutzermakros werden unterstützt.
Password	Passwort für die Authentifizierung. Dadurch wird der Wert von CURLOPT_PASSWORD gesetzt. Benutzermakros werden unterstützt. Der hier eingegebene Wert wird beim Exportieren der Medientypkonfiguration unverändert exportiert. Um zu verhindern, dass vertrauliche Zugangsdaten in exportierten Dateien offengelegt werden, verwenden Sie anstelle eines wörtlichen Passworts ein Benutzermakro (z. B. `{$EMAIL_SMTP_PASSWORD}`) – der Makroverweis wird exportiert, während der geheime Wert auf dem Zielsystem konfiguriert bzw. neu konfiguriert werden sollte.
OAuth tokens	Klicken Sie auf *Configure*, um in einem neuen Fenster Parameter zum Abrufen von OAuth tokens einzurichten. Dieses Feld ist nur verfügbar, wenn im Feld *Authentication* die Option "OAuth" ausgewählt ist.

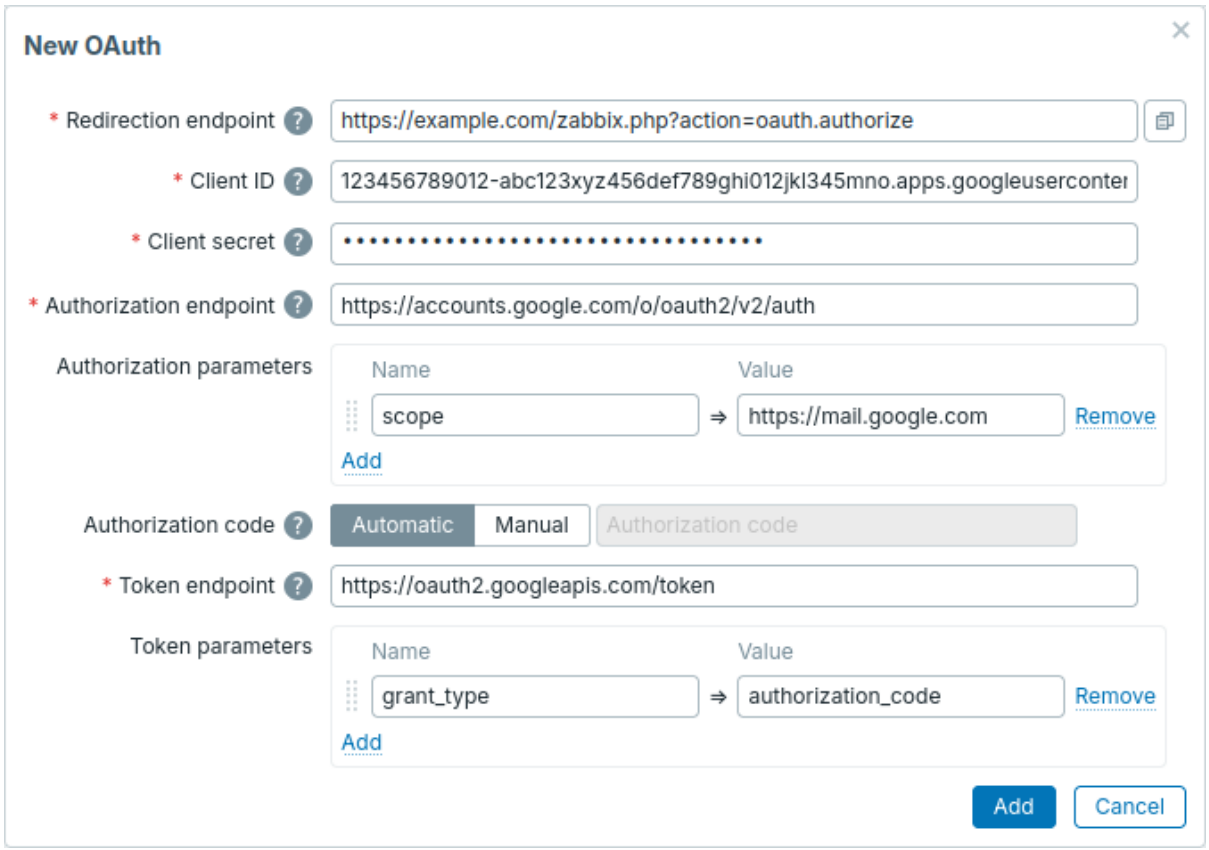
Parameter	Beschreibung
Message format	Wählen Sie das Nachrichtenformat aus: **HTML** - als HTML senden; **Plain text** - als Klartext senden.

Attention:
Um die SMTP-Authentifizierung zu aktivieren, muss der Zabbix-Server mit der Option `--with-libcurl` zur Kompilierung kompiliert werden. Diese Option erfordert libcurl (siehe die Anforderungen für Server für Details zu den Versionen).

Weitere Informationen zur Konfiguration von Standardnachrichten und Optionen zur Verarbeitung von Alerts finden Sie unter common media type parameters.

OAuth-Tokens

Die folgenden Parameter sind für das Abrufen von OAuth-Tokens erforderlich:



Warning:
Das Client-Geheimnis wird im Klartext exportiert, wenn die Konfiguration des E-Mail-Medientyps exportiert wird.

Parameter	Beschreibung
Redirection endpoint	Geben Sie die Zabbix-Frontend-URL ein, an die der OAuth-Dienst die OAuth-Autorisierung zurückleitet (im Format `https://<zabbix-frontend-url>/zabbix.php?action=oauth.authorize`). Sie wird für einen neuen E-Mail-Medientyp automatisch gesetzt, wenn die Frontend-URL definiert ist.
Client ID	Geben Sie die eindeutige Kennung der Client-App ein, die beim OAuth-Autorisierungsserver registriert ist.
Client secret	Geben Sie das private Geheimnis der Client-App ein, die beim OAuth-Autorisierungsserver registriert ist.
Authorization endpoint	Geben Sie die URL des OAuth-Autorisierungsservers für die Anforderung der Benutzerautorisierung ein.
Authorization parameters	Geben Sie die Parameter für den Autorisierungsendpunkt ein.

Parameter	Beschreibung
Authorization code	Geben Sie den Autorisierungscode ein: **Automatic** - der Code wird automatisch über eine Umleitungsseite abgerufen; **Manual** - geben Sie den Code manuell ein, wenn der automatische Abruf fehlschlägt.
Token endpoint	Geben Sie die URL des OAuth-Autorisierungsservers ein, um den Autorisierungscode gegen Zugriffs- und Aktualisierungstokens auszutauschen.
Token parameters	Geben Sie die Parameter für das Zugriffstoken ein.

Note:
Das Abrufen von OAuth-Tokens verfügt über automatisierte Funktionen für die E-Mail-Anbieter **Gmail**, **Gmail relay** und **Office365**. Es müssen nur die Parameterwerte *Redirection endpoint*, *Client ID* und *Client secret* angegeben werden. Zabbix füllt die anderen erforderlichen Werte automatisch aus (siehe OAuth-URL-Standardwerte nach Anbieter).

Abruf von OAuth-Zugriffs- und Aktualisierungs-Token

Nach dem Absenden der OAuth-Parameter:

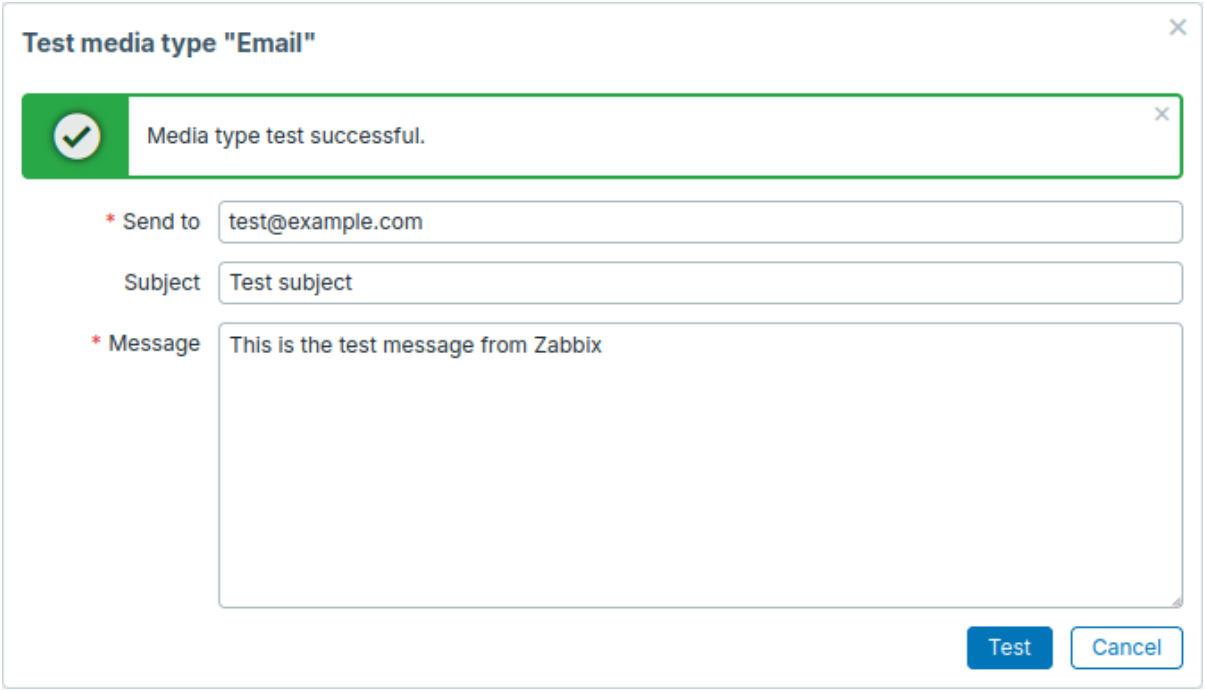
1. Ein Browser-Popup wird geöffnet und leitet den Benutzer zum *Authorization endpoint* weiter.
2. Der Benutzer autorisiert Zabbix im OAuth-Dienst.
3. Der OAuth-Dienst leitet den Benutzer zur Zabbix-Frontend-Aktion `oauth.authorize` mit dem Autorisierungscode und dem Scope-Wert weiter.
4. Als Antwort sendet Zabbix eine Anfrage an den *Token endpoint*, um den Autorisierungscode gegen Zugriffs- und Aktualisierungs-Token auszutauschen.

Testen

Um zu testen, ob ein konfigurierter E-Mail-Medientyp korrekt funktioniert:

1. Suchen Sie die relevante E-Mail in der Liste der Medientypen.
2. Klicken Sie in der letzten Spalte der Liste auf *Test* (ein Testfenster wird geöffnet).
3. Geben Sie eine Empfängeradresse für *Senden an*, den Nachrichtentext und optional einen Betreff ein.
4. Klicken Sie auf *Test*, um eine Testnachricht zu senden.

Im selben Fenster wird eine Meldung über Erfolg oder Fehler des Tests angezeigt:



Benutzer-Medien

Nachdem der E-Mail-Medientyp konfiguriert wurde, wechseln Sie zum Abschnitt *Users* > *Users* und bearbeiten Sie das Benutzerprofil, um dem Benutzer E-Mail-Medien zuzuweisen. Die Schritte zum Einrichten von Benutzer-Medien, die für alle Medientypen gleich sind, werden auf der Seite Medientypen beschrieben.

Automatisierte Gmail-/Office365-Medientypen

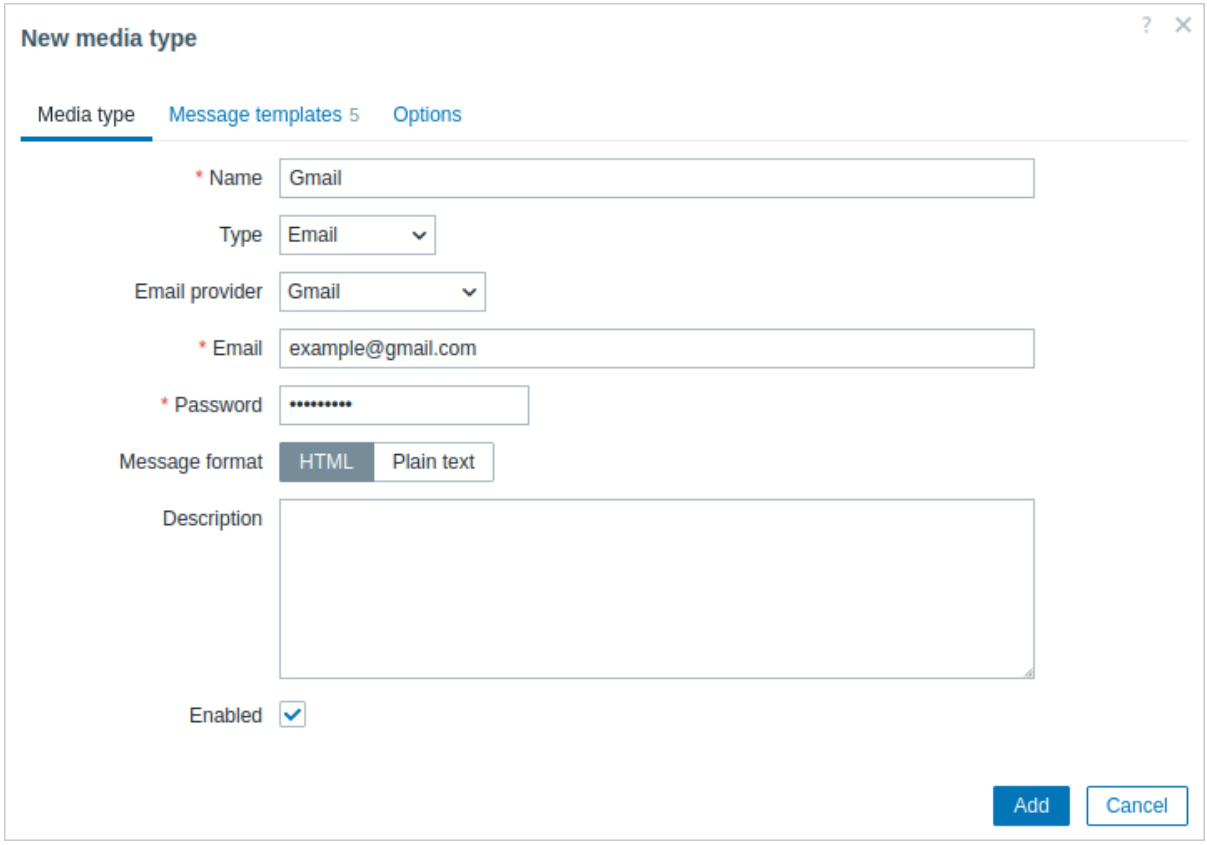
Übersicht

Gmail- oder Office365-Benutzer können von automatisierten Funktionen in der Medientyp-Konfiguration profitieren.

Konfiguration

Im Feld *E-Mail-Anbieter* der Konfiguration des E-Mail-Medientyps können Sie vorkonfigurierte Optionen für Gmail und Office 365 auswählen.

Bei Auswahl der Optionen für Gmail/Office365 müssen Sie lediglich die E-Mail-Adresse und das Passwort des Absenders angeben, um einen funktionierenden Medientyp zu erstellen.



Sobald die E-Mail-Adresse und das Passwort angegeben wurden, kann Zabbix alle erforderlichen Einstellungen für Gmail/Office365-Medientypen automatisch mit den tatsächlichen bzw. empfohlenen Werten ausfüllen, d. h. *SMTP-Server*, *SMTP-Server-Port*, *SMTP-Helo* und *Verbindungssicherheit*. Aufgrund dieser Automatisierung werden diese Felder nicht einmal angezeigt. Es ist jedoch möglich, die SMTP-Server- und E-Mail-Details in der Liste der Medientypen einzusehen (siehe die Spalte *Details*).

Beachten Sie außerdem:

- Für die Relay-Optionen ist kein Passwort erforderlich.
- Für das Office365-Relay wird der Domänenname der angegebenen E-Mail-Adresse verwendet, um den SMTP-Server dynamisch auszufüllen (d. h. „example.com“ in `example-com.mail.protection.outlook.com` wird durch den tatsächlichen Wert ersetzt).

OAuth-Token

Das Abrufen von OAuth-Token verfügt über automatisierte Funktionen für die E-Mail-Anbieter *Gmail*, *Gmail relay* und *Office365*.



Für den OAuth-Zugriff und das Abrufen des Aktualisierungstokens muss nur ein begrenzter Satz von Parametern angegeben werden – *Umleitungsendpunkt*, *Client-ID* und *Client-Secret*. Zabbix füllt die anderen erforderlichen Werte automatisch aus (siehe Standardwerte für OAuth-URLs nach Anbieter).

Beachten Sie, dass es auch möglich ist, das Formular Generic SMTP zu verwenden, um die OAuth-Autorisierung für diese Anbieter einzurichten.

Note:
SmtpClientAuthentication muss entweder pro Benutzer/freigegebenem Postfach oder für den gesamten Tenant in Office365 aktiviert sein (in den Standardeinstellungen nicht aktiviert).

Standardwerte der OAuth-URLs nach Anbieter

Die folgende Tabelle listet die standardmäßigen OAuth-URL-Werte und Parameter pro Anbieter auf, die von Zabbix verwendet werden.

Parameter	Gmail	Office365	Generisches SMTP
Autorisierungsendpunkt	`https://accounts.google.com/o/oauth2/auth`	`https://login.microsoftonline.com/common/oauth2/v2.0/authorize`	Kein Standardwert
scope	`https://mail.google.com`	`https://outlook.office.com/SMTP.Send offline_access`	Kein Standardwert
access_type	`offline`	Nicht verwendet	Kein Standardwert
prompt	`consent`	Nicht verwendet	Kein Standardwert
redirect_uri	Der Wert des Feldes *Umleitungsendpunkt* wird automatisch per Code hinzugefügt. Er ist in den Parametern des *Autorisierungsendpunkts* nicht aufgeführt.		
state	Ein eindeutiger Hash zur Identifizierung des Mediums, das mit dem OAuth-Token aktualisiert wird, wird automatisch per Code hinzugefügt. Er ist in den Parametern des *Autorisierungsendpunkts* nicht aufgeführt.		
Token-Endpunkt	`https://oauth2.googleapis.com/token`	`https://login.microsoftonline.com/common/oauth2/v2.0/token`	Kein Standardwert
grant_type	`authorization_code`	`authorization_code`	Kein Standardwert
redirect_uri	Der Wert des Feldes *Umleitungsendpunkt* wird automatisch per Code hinzugefügt. Er ist in den Parametern des *Token-Endpunkts* nicht aufgeführt.		
state	Ein eindeutiger Hash zur Identifizierung des Mediums, das mit dem OAuth-Token aktualisiert wird, wird automatisch per Code hinzugefügt. Er ist in den Parametern des *Token-Endpunkts* nicht aufgeführt.		

Office365-Workaround für persönliche Konten

Persönliche Office365-Konten unterstützen keine Abfragezeichenfolgen in der Umleitungs-URL.

Um dieses Problem zu umgehen, kann der Administrator des Apache-Webservers die folgende Rewrite-Regel zur Datei `.htaccess` hinzufügen:

```
RewriteEngine On
RewriteRule ^/oauth\.authorize$ /zabbix.php?action=oauth.authorize [QSA,L,PT]
```

Dadurch wird die Verwendung einer vereinfachten Umleitungs-URL wie der folgenden ermöglicht:

```
http://server-name/zabbix/oauth.authorize
```

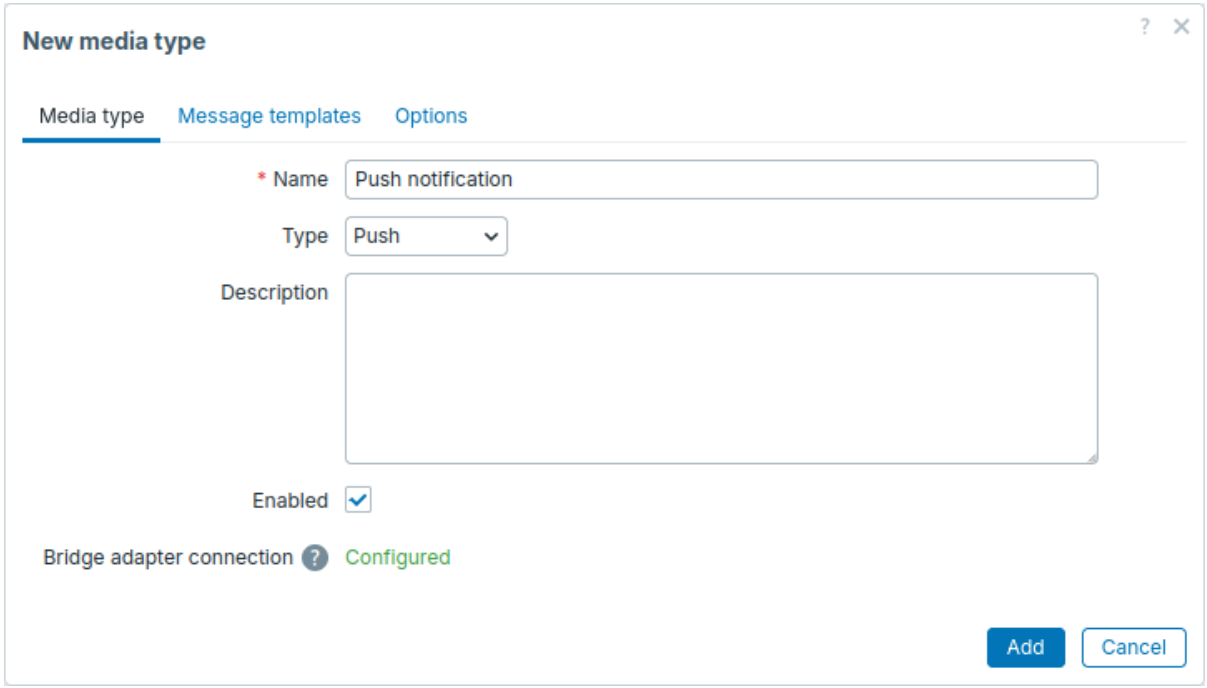
anstelle einer URL mit Abfrageparametern, wodurch die Kompatibilität mit persönlichen Office365-Konten sichergestellt wird.

Push-Benachrichtigung

Dieser Medientyp ermöglicht es dem Zabbix Server, Push-Benachrichtigungen an die Zabbix mobile app auf Ihrem Mobilgerät zu senden.

Konfiguration

Wählen Sie beim Erstellen eines Medientyps *Push* als *Typ* aus:



Das Konfigurationsformular enthält allgemeine Parameter, Nachrichtenvorlagen und Optionen für den Medientyp.

Das Formular zeigt außerdem an, ob der Zabbix-Bridge-Adapter konfiguriert ist.
 Wenn der Adapter nicht konfiguriert ist, können Sie den Medientyp trotzdem erstellen, er wird jedoch nicht funktionieren.

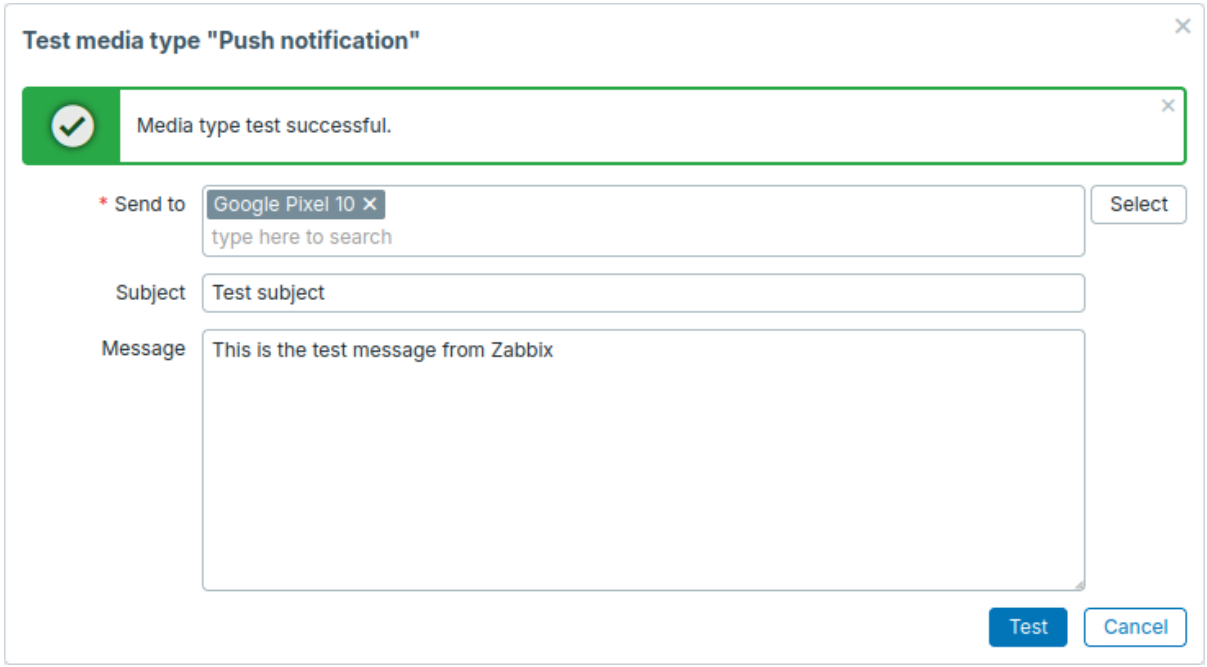
Nachdem Sie den Medientyp erstellt und aktiviert haben, fügen Sie ein Gerät hinzu, damit ein Benutzer den Medientyp automatisch erhält (oder fügen Sie Benutzermedien manuell hinzu). Dadurch können Benutzer Push-Benachrichtigungen auf ihrem Mobilgerät empfangen.

Testen

So testen Sie, ob dieser Medientyp funktioniert:

1. Suchen Sie den Medientyp in der Liste der Medientypen.
2. Klicken Sie in der letzten Spalte der Liste auf *Test*, um das Testformular zu öffnen.
3. Wählen Sie im Feld *Senden an* ein Gerät aus.
4. Klicken Sie auf *Test*, um eine Push-Testbenachrichtigung an das ausgewählte Gerät zu senden.

Im selben Fenster wird eine Erfolgs- oder Fehlermeldung angezeigt:



SMS

Übersicht

Zabbix unterstützt das Senden von SMS-Nachrichten über ein serielles GSM-Modem, das mit dem seriellen Port des Zabbix Server verbunden ist.

Stellen Sie sicher, dass:

- die Geschwindigkeit des seriellen Geräts (unter Linux normalerweise /dev/ttyS0) mit der des GSM-Modems übereinstimmt. Zabbix setzt die Geschwindigkeit der seriellen Verbindung nicht. Es verwendet die Standardeinstellungen.
- der Benutzer 'zabbix' Lese-/Schreibzugriff auf das serielle Gerät hat. Führen Sie den Befehl ls -l /dev/ttyS0 aus, um die aktuellen Berechtigungen des seriellen Geräts anzuzeigen.
- beim GSM-Modem die PIN eingegeben wurde und diese nach einem Neustart der Stromversorgung erhalten bleibt. Alternativ können Sie die PIN auf der SIM-Karte deaktivieren. Die PIN kann durch Eingabe des Befehls AT+CPIN="NNNN" (NNNN ist Ihre PIN, die Anführungszeichen müssen vorhanden sein) in einer Terminalsoftware wie Unix minicom oder Windows HyperTerminal gesetzt werden.

Zabbix wurde mit folgenden GSM-Modems getestet:

- Siemens MC35
- Teltonika ModemCOM/G10

Um SMS als Übertragungskanal für Nachrichten zu konfigurieren, müssen Sie außerdem SMS als Medientyp konfigurieren und die entsprechenden Telefonnummern für die Benutzer eintragen.

Konfiguration

So konfigurieren Sie SMS als Medientyp:

1. Gehen Sie zu *Alerts* > *Media types*.
2. Klicken Sie auf *Create media type* (oder klicken Sie in der Liste der vordefinierten Medientypen auf *SMS*).
3. Geben Sie die Parameter des SMS-Medientyps im Formular ein.

Die folgenden Parameter sind spezifisch für den SMS-Medientyp:

Parameter	Beschreibung
GSM modem	Legen Sie den Namen des seriellen Geräts des GSM-Modems fest. Der hier eingegebene Pfad wird anhand des Serverparameters SMSDevices validiert (falls angegeben).

Weitere Informationen zum Konfigurieren von Standardnachrichten und Optionen für die Alarmverarbeitung finden Sie unter allgemeine Parameter für Medientypen. Beachten Sie, dass die parallele Verarbeitung des Versands von SMS-Benachrichtigungen nicht möglich ist.

Benutzermedien

Nachdem der SMS-Medientyp konfiguriert wurde, wechseln Sie zum Abschnitt *Benutzer* > *Benutzer* und bearbeiten Sie das Benutzerprofil, um dem Benutzer das SMS-Medium zuzuweisen. Die Schritte zum Einrichten von Benutzermedien, die für alle Medientypen gelten, werden auf der Seite Medientypen beschrieben.

Benutzerdefinierte Alarmskripte

Übersicht

Wenn Sie mit den vorhandenen Medientypen zum Senden von Benachrichtigungen nicht zufrieden sind, gibt es eine alternative Möglichkeit dafür. Sie können ein Skript erstellen, das die Benachrichtigung auf Ihre Weise verarbeitet.

Benutzerdefinierte Alarmskripte werden auf dem Zabbix Server ausgeführt. Diese Skripte müssen sich in dem Verzeichnis befinden, das durch den Parameter AlertScriptsPath in der Server-Konfigurationsdatei angegeben ist.

Hier ist ein Beispiel für ein benutzerdefiniertes Alarmskript:

```
#####!/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 prüft den Exit-Code der ausgeführten Befehle und Skripte. Jeder Exit-Code, der ungleich **0** ist, wird als Fehler bei der Befehlsausführung betrachtet. In solchen Fällen versucht Zabbix, die fehlgeschlagene Ausführung zu wiederholen.

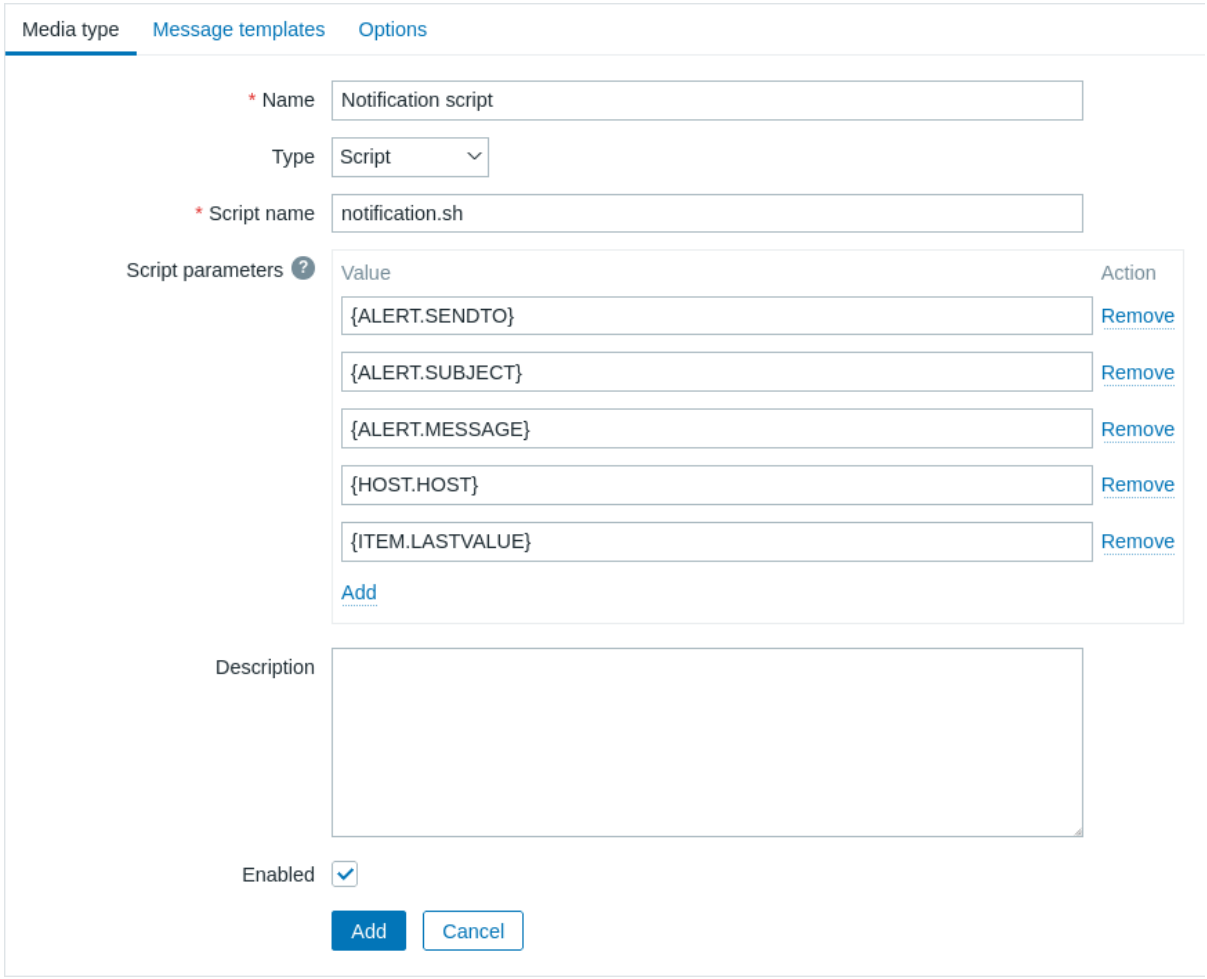
Umgebungsvariablen werden für das Skript weder beibehalten noch erstellt und müssen daher explizit behandelt werden.

Konfiguration

So konfigurieren Sie benutzerdefinierte Alarmskripte als Medientyp:

1. Gehen Sie zu *Alerts* > *Media types*.
2. Klicken Sie auf *Create media type*.
3. Geben Sie die Parameter des Skript-Medientyps im Formular ein.

Die Registerkarte **Media type** enthält allgemeine Attribute des Medientyps:



Alle Pflichtfelder sind mit einem roten Sternchen gekennzeichnet.

Die folgenden Parameter gelten speziell für den Skript-Medientyp:

Parameter	Beschreibung
Script name	Geben Sie den Namen der Skriptdatei (z. B. notification.sh) ein, die sich in dem Verzeichnis befindet, das im Serverkonfigurationsparameter AlertScriptsPath angegeben ist.
Script parameters	Fügen Sie optionale Skriptparameter hinzu, die in der Reihenfolge ihrer Definition als Befehlszeilenargumente an das Skript übergeben werden. Skriptparameter unterstützen die Makros {ALERT.SENDTO}, {ALERT.SUBJECT}, {ALERT.MESSAGE} sowie alle Makros, die in Benachrichtigungen unterstützt werden, und außerdem Benutzermakros.

Weitere Informationen zum Konfigurieren von Standardnachrichten und Optionen für die Alarmverarbeitung finden Sie unter allgemeine Medientypparameter.

Warning:
Auch wenn ein Alarmskript keine Standardnachrichten verwendet, müssen die Nachrichtenvorlagen für die von diesem Medientyp verwendeten Aktionstypen definiert sein. Andernfalls wird keine Benachrichtigung gesendet.

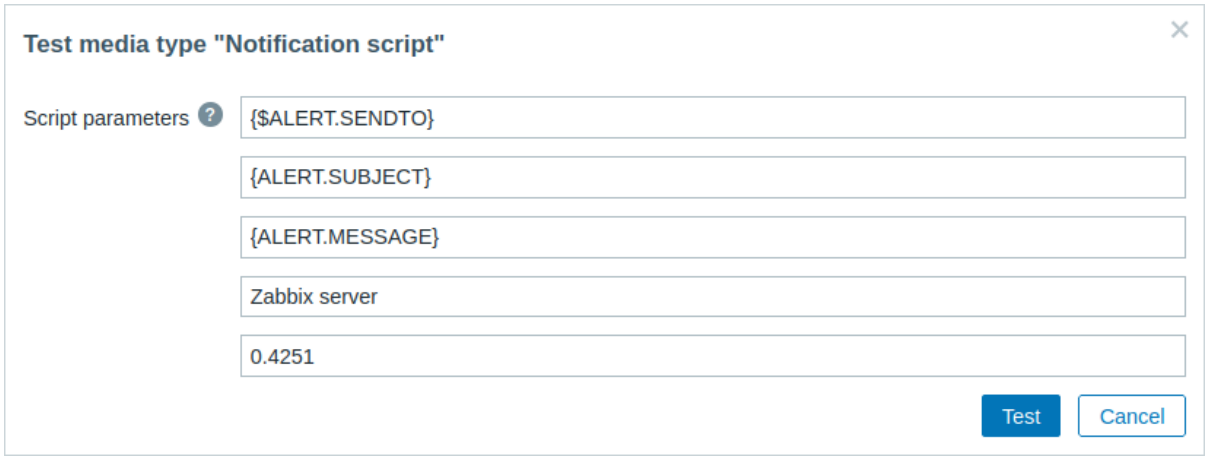
Attention:
Wenn mehr als ein Skript-Medientyp konfiguriert ist, können diese Skripte von den Alerter-Prozessen parallel verarbeitet werden. Die Gesamtzahl der Alerter-Prozesse wird durch den Parameter `StartAlerters` in der Serverkonfigurationsdatei begrenzt.

Testen

Um einen konfigurierten Script-Medientyp zu testen:

1. Suchen Sie das relevante Script in der Liste der Medientypen.

2. Klicken Sie in der letzten Spalte der Liste auf *Test*; ein Testformular wird in einem Popup-Fenster geöffnet. Das Testformular enthält dieselbe Anzahl von Parametern, die für den Script-Medientyp konfiguriert sind.
3. Bearbeiten Sie bei Bedarf die Werte der Script-Parameter. Die Bearbeitung wirkt sich nur auf den Testvorgang aus; die tatsächlichen Werte werden nicht geändert.
4. Klicken Sie auf *Test*.



Note:
Beim Testen eines konfigurierten Script-Medientyps werden {ALERT.SENDTO}, {ALERT.SUBJECT}, {ALERT.MESSAGE} und Benutzermakros in ihre Werte aufgelöst, Makros, die sich auf Ereignisse beziehen (z. B. {HOST.HOST}, {ITEM.LASTVALUE} usw.), werden jedoch nicht aufgelöst, da beim Testen kein zugehöriges Ereignis vorhanden ist, aus dem die Details ermittelt werden können. Beachten Sie, dass Makros innerhalb der Makros {ALERT.SUBJECT} und {ALERT.MESSAGE} ebenfalls nicht aufgelöst werden. Wenn der Wert von {ALERT.SUBJECT} beispielsweise aus "Problem: {EVENT.NAME}" besteht, wird das Makro {EVENT.NAME} nicht aufgelöst.

Benutzer-Medien

Sobald der Medientyp konfiguriert ist, gehen Sie zum Abschnitt *Benutzer* > *Benutzer* und bearbeiten Sie ein Benutzerprofil, indem Sie diesem Benutzer diesen Medientyp zuweisen. Die Schritte zum Einrichten von Benutzer-Medien, die für alle Medientypen gleich sind, werden auf der Seite Medientypen beschrieben.

Beachten Sie, dass beim Definieren der Benutzer-Medien das Feld *Senden an* nicht leer sein darf. Wenn dieses Feld im Alarmskript nicht verwendet wird, geben Sie eine beliebige Kombination unterstützter Zeichen ein, um die Validierungsanforderungen zu umgehen.

webhook

Übersicht

Der webhook-Medientyp ist nützlich, um HTTP-Aufrufe mithilfe von benutzerdefiniertem JavaScript-Code für eine unkomplizierte Integration mit externer Software wie Helpdesk-Systemen, Chats oder Messengern durchzuführen. Sie können eine von Zabbix bereitgestellte Integration importieren oder eine benutzerdefinierte Integration von Grund auf neu erstellen.

Integrationen

Die folgenden Integrationen sind verfügbar. Sie ermöglichen die Verwendung vordefinierter webhook-Medientypen, um Zabbix-Benachrichtigungen an folgende Dienste zu senden:

- brevis.one
- Discord
- Event-Driven Ansible
- Express.ms messenger
- GitHub
- GLPI
- IBM Maximo Service Request
- iLert
- iTop
- Jira
- Jira Service Management
- ManageEngine ServiceDesk

- Mantis Bug Tracker
- Mattermost
- MS Teams Workflows
- LINE
- Opsgenie
- OTRS CE
- Pagerduty
- Pushover
- Redmine
- Rocket.Chat
- ServiceNow
- SIGNL4
- Slack
- SolarWinds
- SysAid
- Telegram
- TOPdesk
- VictorOps
- Zammad
- Zendesk

Note:
Zusätzlich zu den hier aufgeführten Diensten kann Zabbix in **Spiceworks** integriert werden (ein webhook ist nicht erforderlich). Um Zabbix-Benachrichtigungen in Spiceworks-Tickets umzuwandeln, erstellen Sie einen E-Mail-Medientyp und geben Sie die E-Mail-Adresse des Spiceworks-Helpdesks (z. B. help@zabbix.on.spiceworks.com) in den Profileinstellungen eines dafür vorgesehenen Zabbix-Benutzers ein.

Konfiguration

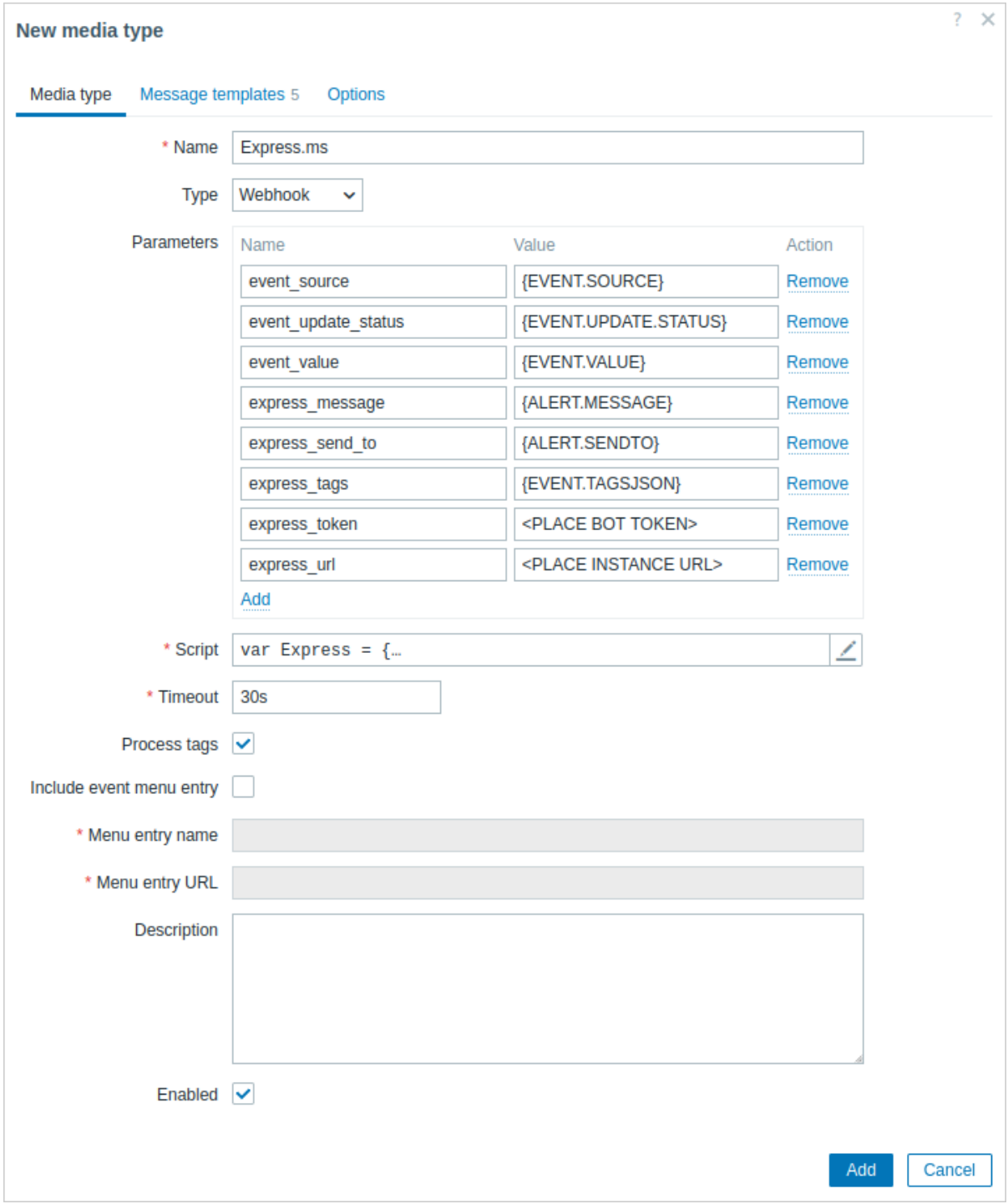
So beginnen Sie mit der Verwendung einer webhook-Integration:

1. Suchen Sie die erforderliche `.yaml`-Datei im Verzeichnis `templates/media` der heruntergeladenen Zabbix-Version oder laden Sie sie aus dem Zabbix-Git-Repository herunter.
2. Importieren Sie die Datei in Ihre Zabbix-Installation. Die webhook wird in der Liste der Medientypen angezeigt.
3. Konfigurieren Sie die webhook gemäß den Anweisungen in der Datei `Readme.md` (Sie können oben auf den Namen einer webhook klicken, um schnell auf `Readme.md` zuzugreifen).

So erstellen Sie eine benutzerdefinierte webhook von Grund auf:

1. Gehen Sie zu *Alarme* > *Medientypen*.
2. Klicken Sie auf *Medientyp erstellen*.
3. Geben Sie die Parameter des webhook-Medientyps im Formular ein.

Die Registerkarte **Medientyp** enthält verschiedene Attribute, die für diesen Medientyp spezifisch sind:



Alle Pflichtfelder sind mit einem roten Sternchen gekennzeichnet.

Die folgenden Parameter sind für den webhook-Medientyp spezifisch:

Parameter	Beschreibung
Parameter	Geben Sie die webhook-Variablen als Attribut-Wert-Paare an. Bei vorkonfigurierten webhooks variiert die Liste der Parameter je nach Dienst. Eine Beschreibung der Parameter finden Sie in der *Readme.md*-Datei der webhook. Bei neuen webhooks sind standardmäßig mehrere allgemeine Variablen enthalten (URL:<empty>, HTTPProxy:<empty>, To:{ALERT.SENDTO}, Subject:{ALERT.SUBJECT}, Message:{ALERT.MESSAGE}); Sie können diese beibehalten oder entfernen. Webhook-Parameter unterstützen Benutzermakros, alle Makros, die in Problembenachrichtigungen unterstützt werden, sowie zusätzlich die Makros {ALERT.SENDTO}, {ALERT.SUBJECT} und {ALERT.MESSAGE}. Wenn Sie einen HTTP-Proxy angeben, unterstützt das Feld dieselbe Funktionalität wie das Feld HTTP-Proxy in der Datenpunkt-Konfiguration. Der Proxy-String kann mit `[scheme]://` beginnen, um anzugeben, welche Art von Proxy verwendet wird (z. B. https, socks4, socks5; siehe Dokumentation).
Skript	Geben Sie JavaScript-Code im modalen Editor ein, der geöffnet wird, wenn Sie in das Parameterfeld oder auf das Stiftsymbol daneben klicken. Dieser Code führt die webhook-Operation aus. Das Skript ist ein Funktionscode, der Parameter-Wert-Paare akzeptiert. Die Werte sollten mithilfe der Methode JSON.parse() in JSON-Objekte konvertiert werden, zum Beispiel: `var params = JSON.parse(value);`. Der Code hat Zugriff auf alle Parameter, kann HTTP-GET-, POST-, PUT- und DELETE-Anforderungen ausführen, zusätzliche Methoden wie CONNECT, PATCH, HEAD, OPTIONS und TRACE unterstützen und HTTP-Header sowie den Anforderungstext steuern. Das Skript muss einen return-Operator enthalten, andernfalls ist es ungültig. Es kann den Status OK zusammen mit einer optionalen Liste von Tags und Tag-Werten (siehe Option *Tags verarbeiten*) oder eine Fehlerzeichenfolge zurückgeben. Wiederherstellungsereignisse (unabhängig davon, ob sie automatisch oder infolge eines manuellen Schließens erzeugt wurden) werden vom Server erstellt und enthalten aufgelöste Ereignis-Tags (einschließlich von Vorlagen, Hosts und Auslösern geerbter Tags). Webhook-Skripte werden ausgeführt, nachdem der Alarm erstellt wurde. Daher werden von einem webhook-Skript zurückgegebene Tags erst nach der ursprünglichen Alarmerstellung hinzugefügt und sind in den Makros `{EVENT.TAGS}` und `{EVENT.RECOVERY.TAGS}` der ursprünglichen Problemmeldung oder der unmittelbaren Wiederherstellungsmeldung nicht vorhanden. *Hinweis*: Es wird empfohlen, lokale Variablen (z. B. `var local = 1`) anstelle globaler Variablen (z. B. `global = 1`) zu verwenden, damit jedes Skript mit seinen eigenen Daten arbeitet und Kollisionen zwischen gleichzeitigen Aufrufen vermieden werden (siehe bekannte Probleme). Siehe auch: Richtlinien für die webhook-Entwicklung, Beispiele für webhook-Skripte, Zusätzliche JavaScript-Objekte.
Zeitüberschreitung	Zeitüberschreitung bei der JavaScript-Ausführung (1–60 s, Standardwert 30 s). Zeitsuffixe werden unterstützt (z. B. `30s`, `1m`).
Tags verarbeiten	Aktivieren Sie das Kontrollkästchen, um zurückgegebene JSON-Eigenschaftswerte als Tags zu verarbeiten. Diese Tags werden zu allen vorhandenen Problem-Tags hinzugefügt. Beachten Sie, dass bei Verwendung von webhook-Tags die webhook ein JSON-Objekt zurückgeben muss, das mindestens ein leeres Tags-Objekt enthält: `var result = {tags: {}};` Beispiele für zurückgegebene Tags: *jira-id:prod-1234*, *responsible:John Smith*, *processed:<no value>*

Parameter	Beschreibung
Ereignismenüeintrag einschließen	Aktivieren Sie das Kontrollkästchen, um im Ereignismenü einen Eintrag einzuschließen, der auf ein erstelltes externes Ticket verweist. Für jede aktivierte webhook, bei der dieses Kontrollkästchen aktiviert ist, wird ein Eintrag eingeschlossen. Beachten Sie, dass ein Eintrag nur dann eingeschlossen wird, wenn die Makros {EVENT.TAGS.<tag name>} in den Parametern *Name des Menüeintrags* und *URL des Menüeintrags* aufgelöst werden können (das heißt, wenn für das Ereignis diese Tags definiert sind). Wenn diese Option aktiviert ist, sollte die webhook nicht zum Senden von Benachrichtigungen an verschiedene Benutzer verwendet werden (erwägen Sie stattdessen, einen dedizierten Benutzer zu erstellen) und nicht in mehreren Alarmaktionen für ein einzelnes Problemereignis verwendet werden.
Name des Menüeintrags	Geben Sie den Namen des Menüeintrags an. Das Makro {EVENT.TAGS.<tag name>} wird unterstützt. Dieses Feld ist nur erforderlich, wenn *Ereignismenüeintrag einschließen* aktiviert ist.
URL des Menüeintrags	Geben Sie die zugrunde liegende URL des Menüeintrags an. Das Makro {EVENT.TAGS.<tag name>} wird unterstützt. Dieses Feld ist nur erforderlich, wenn *Ereignismenüeintrag einschließen* aktiviert ist.

Unter allgemeine Medientypparameter finden Sie Einzelheiten zur Konfiguration von Standardnachrichten und Optionen für die Alarmverarbeitung.

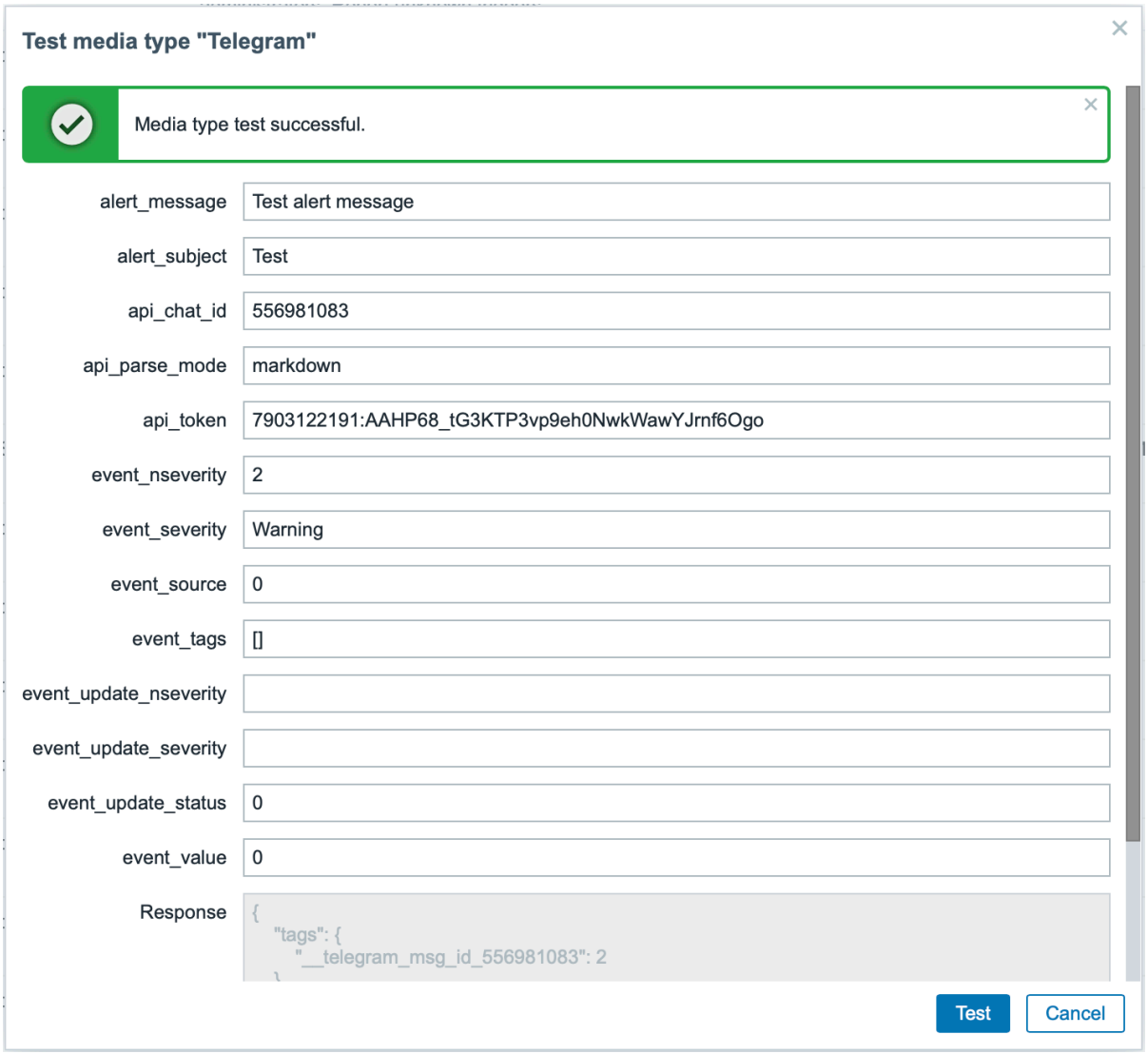
Warning:
Auch wenn eine webhook keine Standardnachrichten verwendet, müssen Nachrichtenvorlagen für die von dieser webhook verwendeten Aktionstypen weiterhin definiert werden.

Testen

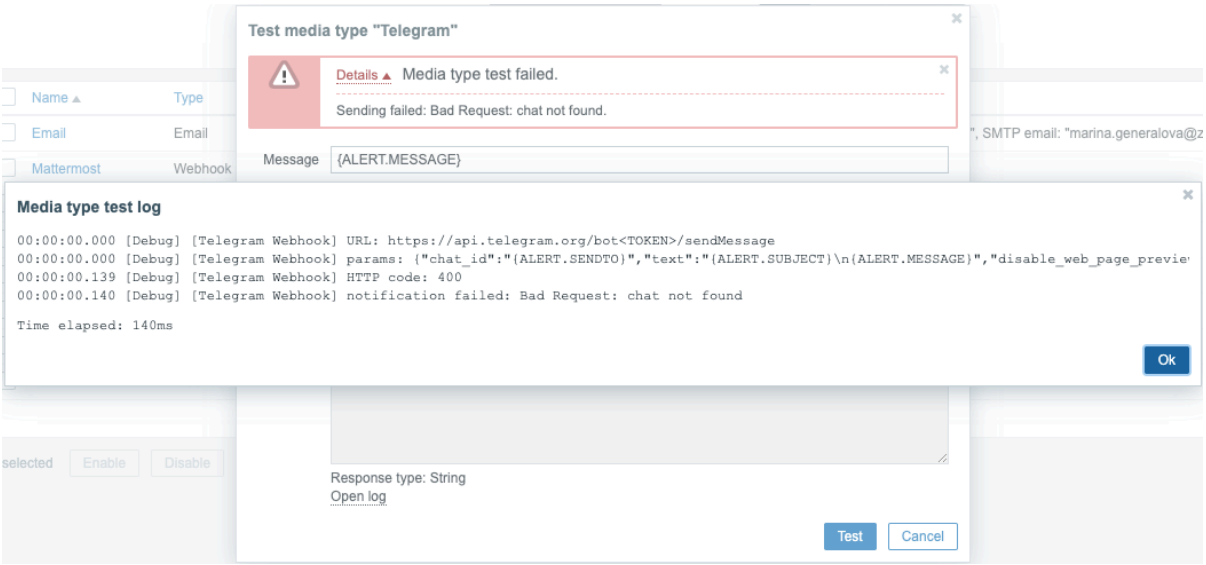
So testen Sie einen konfigurierten webhook-Medientyp:

1. Suchen Sie den entsprechenden webhook in der Liste der Medientypen.
2. Klicken Sie in der letzten Spalte der Liste auf *Test* (ein Testfenster wird geöffnet).
3. Bearbeiten Sie die Werte der webhook-Parameter nach Bedarf.
   Ersetzen Sie Makros durch Beispielwerte. Andernfalls werden die Makros nicht aufgelöst und der Test schlägt fehl.
4. Klicken Sie auf *Test*.

Das Ersetzen oder Löschen von Werten im Testfenster wirkt sich nur auf den Testvorgang aus. Die tatsächlichen Attributwerte des webhooks bleiben unverändert.



Um die Testprotokolleinträge des Medientyps anzuzeigen, ohne das Testfenster zu verlassen, klicken Sie auf *Protokoll öffnen* (ein neues Pop-up-Fenster wird geöffnet).



Wenn der webhook-Test erfolgreich ist:

- Die Meldung *„Medientyp-Test erfolgreich.“* wird angezeigt.
- Die Server-Antwort erscheint im grauen Feld *Antwort*.
- Der Antworttyp (JSON oder String) wird unterhalb des Feldes *Antwort* angegeben.

Wenn der webhook-Test fehlschlägt:

- Die Meldung *„Medientyp-Test fehlgeschlagen.“* wird angezeigt, gefolgt von zusätzlichen Details zum Fehler.

Benutzermedien

Nachdem der Medientyp konfiguriert wurde, wechseln Sie zum Abschnitt *Benutzer* > *Benutzer* und weisen Sie das webhook-Medium einem vorhandenen Benutzer zu oder erstellen Sie einen neuen Benutzer, der das webhook repräsentiert. Die Schritte zum Einrichten von Benutzermedien für einen vorhandenen Benutzer, die für alle Medientypen gelten, werden auf der Seite Medientypen beschrieben.

Wenn ein webhook Tags zum Speichern der Ticket\Nachrichten-ID verwendet, vermeiden Sie es, dasselbe webhook verschiedenen Benutzern als Medium zuzuweisen, da dies zu webhook-Fehlern führen kann (gilt für die Mehrheit der Webhooks, die die Option *Include event menu entry* verwenden). In diesem Fall empfiehlt es sich, einen dedizierten Benutzer zu erstellen, der das webhook repräsentiert:

1. Nachdem Sie den webhook-Medientyp konfiguriert haben, wechseln Sie zum Abschnitt *Benutzer* > *Benutzer* und erstellen Sie einen dedizierten Zabbix-Benutzer, der das webhook repräsentiert - zum Beispiel mit dem Benutzernamen *Slack* für das Slack-webhook. Alle Einstellungen außer dem Medium können auf ihren Standardwerten belassen werden, da sich dieser Benutzer nicht bei Zabbix anmelden wird.
2. Wechseln Sie im Benutzerprofil zur Registerkarte *Medien* und fügen Sie ein webhook hinzu, indem Sie die erforderlichen Kontaktinformationen angeben. Wenn das webhook kein Feld *Send to* verwendet, geben Sie eine beliebige Kombination unterstützter Zeichen ein, um die Validierungsanforderungen zu umgehen.
3. Gewähren Sie diesem Benutzer mindestens Leseberechtigungen für alle Hosts, für die er die Alarme senden soll, unter Berechtigungen.

Fügen Sie beim Konfigurieren der Alarmauslösung diesen Benutzer im Feld *Send to users* unter den Operationsdetails hinzu - dadurch wird Zabbix angewiesen, das webhook für Benachrichtigungen aus dieser Aktion zu verwenden.

Konfigurieren von Alarmaktionen

Aktionen bestimmen, welche Benachrichtigungen über das webhook gesendet werden sollen. Die Schritte zum Konfigurieren von Aktionen, die webhooks betreffen, sind mit Ausnahme der folgenden Punkte für alle anderen Medientypen gleich:

- Wenn ein webhook webhook-Tags verwendet, um Ticket\Nachrichten-ID zu speichern und Update- bzw. Resolve-Operationen zu verarbeiten, vermeiden Sie die Verwendung desselben webhooks in mehreren Alarmaktionen für ein einzelnes Problemereignis. Wenn {EVENT.TAGS.<tag name>} vorhanden ist und im webhook aktualisiert wird, ist der resultierende Wert undefiniert. Um dies zu vermeiden, verwenden Sie im webhook einen neuen Tag-Namen zum Speichern aktualisierter Werte. Dies gilt für die von Zabbix bereitgestellten webhooks für Jira, Jira Service Desk, Mattermost, Opsgenie, OTRS, Redmine, ServiceNow, Slack, Zammad und Zendesk sowie für die meisten webhooks, die die Option *Include event menu entry* verwenden. Beachten Sie jedoch, dass ein einzelner webhook in mehreren Operationen oder Eskalationsschritten derselben Aktion verwendet werden kann, ebenso wie in verschiedenen Aktionen, die aufgrund unterschiedlicher Bedingungen nicht durch dasselbe Problemereignis ausgelöst werden.
- Wenn Sie einen webhook in Aktionen für interne Ereignisse verwenden, stellen Sie sicher, dass Sie das Kontrollkästchen *Custom message* aktivieren und in der Konfiguration der Aktionsoperation eine benutzerdefinierte Nachricht definieren. Andernfalls wird keine Benachrichtigung gesendet.

webhook-Skriptbeispiele

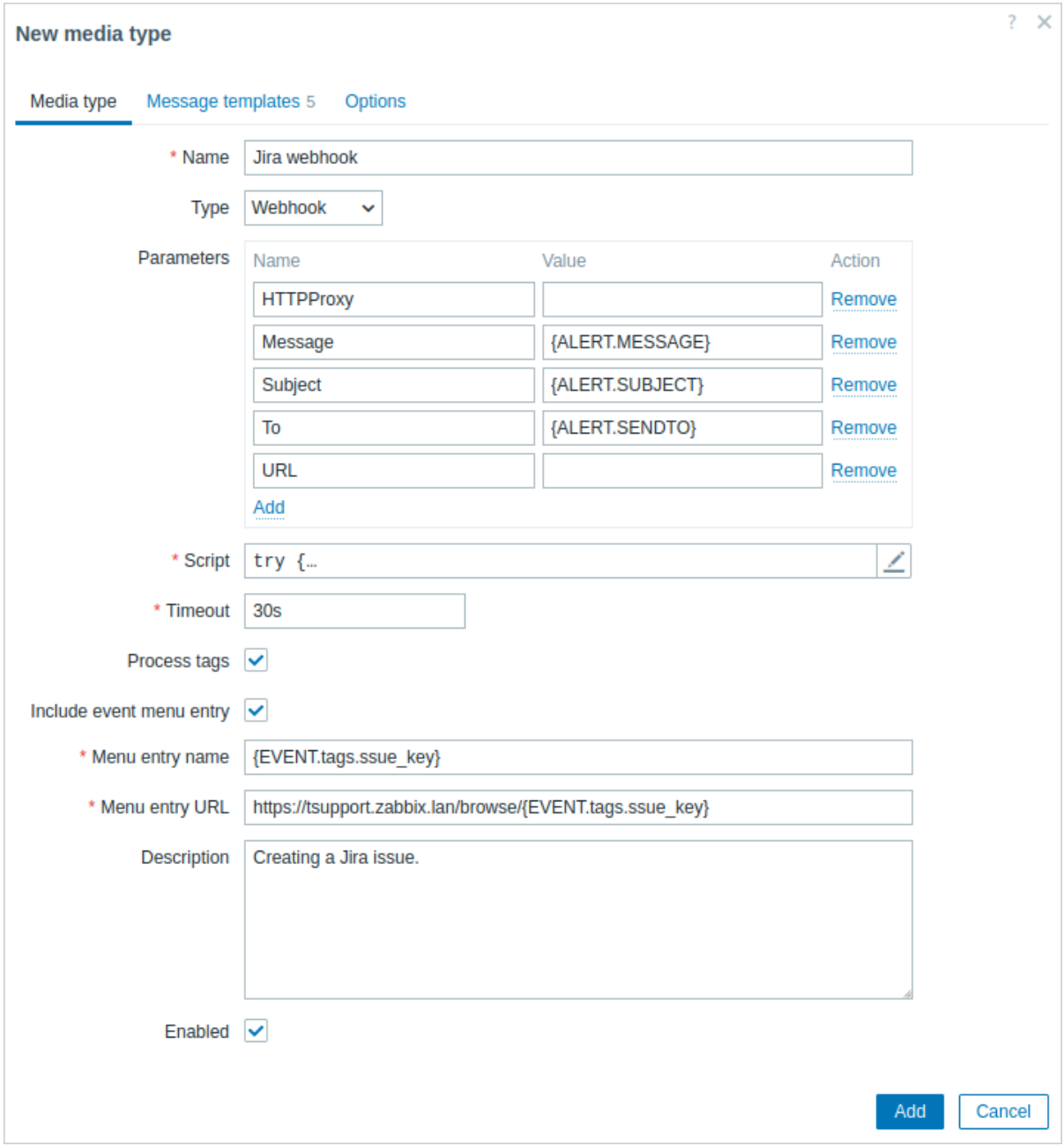
Übersicht

Obwohl Zabbix eine große Anzahl von webhook-Integrationen direkt einsatzbereit anbietet, möchten Sie möglicherweise stattdessen Ihre eigenen webhooks erstellen. Dieser Abschnitt enthält Beispiele für benutzerdefinierte webhook-Skripte (verwendet im Parameter *Script*). Siehe den Abschnitt webhook für eine Beschreibung weiterer webhook-Parameter.

> **Attention:**
> Verwenden Sie in JavaScript für die Vorverarbeitung keine nicht deklarierten Zuweisungen. Verwenden Sie `var`, um lokale Variablen zu deklarieren.

Wiederherstellungsereignisse (unabhängig davon, ob sie automatisch oder nach einem manuellen Schließen erzeugt werden) enthalten aufgelöste Ereignistags (einschließlich von Vorlagen, Hosts und Auslösern geerbter Tags). Webhook-Skripte werden nach der Erstellung des Alarms ausgeführt; von einem webhook-Skript zurückgegebene Tags werden daher erst nach der anfänglichen Alarm-Erstellung angewendet und sind möglicherweise im Alarm, der den webhook ausgelöst hat, nicht vorhanden. Wenn eine Integration erfordert, dass von einem webhook erzeugte Tags in der ursprünglichen Benachrichtigung vorhanden sind, rufen Sie die Ereignistags vom Server ab (zum Beispiel über die Event-API) oder speichern Sie von webhooks erzeugte Tags in einem externen persistenten Speicher und korrelieren Sie sie dort.

Jira webhook (benutzerdefiniert)



Dieses Skript erstellt ein JIRA-Ticket und gibt einige Informationen zum erstellten Ticket zurück.

```
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();
    }

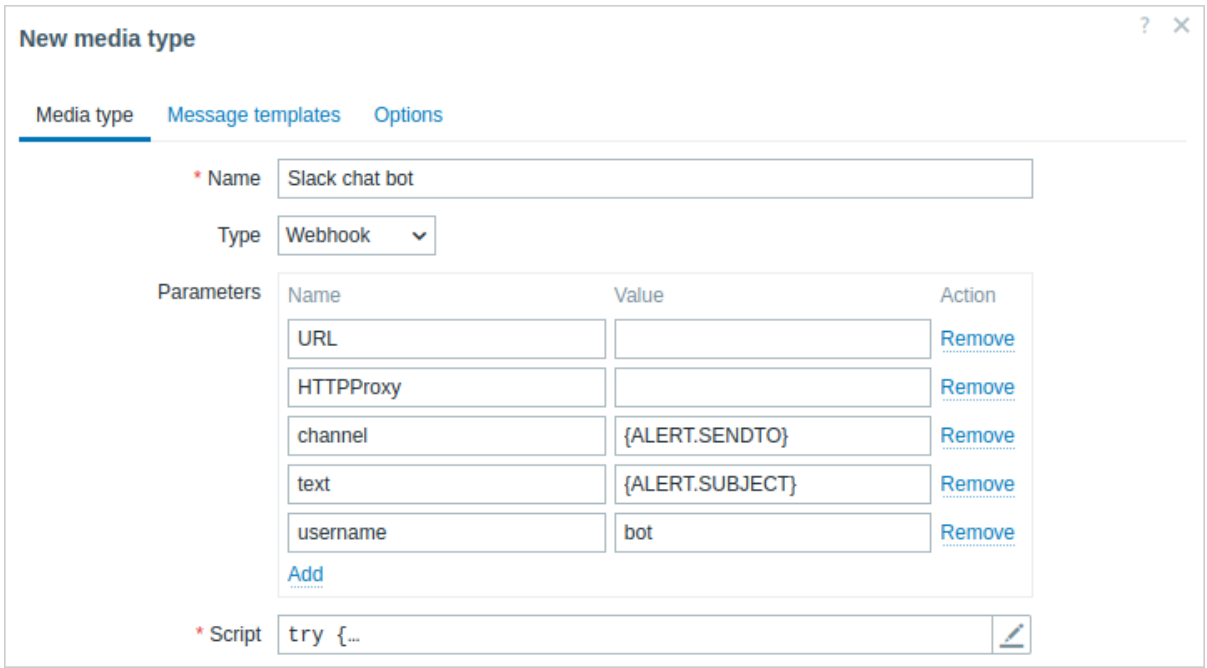
    resp = JSON.parse(resp);
    result.tags.issue_id = resp.id;
    result.tags.issue_key = resp.key;

    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;
}
```

Slack webhook (custom)

Dieses webhook leitet Benachrichtigungen von Zabbix an einen Slack-Kanal weiter.




```
try {
    var params = JSON.parse(value),
        req = new HttpRequest(),
        response;

    if (params.HTTPProxy) {
        req.setProxy(params.HTTPProxy);
    }

    req.addHeader('Content-Type: application/x-www-form-urlencoded');

    Zabbix.log(4, '[ Slack webhook ] Webhook request with value=' + value);
```

```
    response = req.post(params.URL, 'payload=' + encodeURIComponent(value));
    Zabbix.log(4, '[ Slack webhook ] Responded with code: ' + req.getStatus() + '. Response: ' + response)

    try {
        response = JSON.parse(response);
    }
    catch (error) {
        if (req.getStatus() < 200 || req.getStatus() >= 300) {
            throw 'Request failed with status code ' + req.getStatus();
        }
        else {
            throw 'Request success, but response parsing failed.';
        }
    }

    if (req.getStatus() !== 200 || !response.ok || response.ok === 'false') {
        throw response.error;
    }

    return 'OK';
}
catch (error) {
    Zabbix.log(3, '[ Slack webhook ] Sending failed. Error: ' + error);

    throw 'Failed with error: ' + error;
}
```

Aktionen

Übersicht

Wenn Sie möchten, dass bestimmte Operationen infolge von Ereignissen ausgeführt werden (zum Beispiel das Senden von Benachrichtigungen), müssen Sie Aktionen konfigurieren.

Aktionen können als Reaktion auf Ereignisse aller unterstützten Typen definiert werden:

- Auslöser-Aktionen – für Ereignisse, wenn sich der Auslöserstatus zwischen `OK` und `Problem` ändert
- Service-Aktionen – für Ereignisse, wenn sich der Servicestatus zwischen `OK` und `Problem` ändert
- Discovery-Aktionen – für Ereignisse, wenn eine Netzwerk-Discovery stattfindet
- Autoregistrierungs-Aktionen – für Ereignisse, wenn sich neue aktive Agents automatisch registrieren (oder sich die Host-Metadaten bereits registrierter Hosts ändern)
- Interne Aktionen – für Ereignisse, wenn Datenpunkte nicht mehr unterstützt werden oder Auslöser in einen unbekannten Zustand wechseln

Die wichtigsten Unterschiede bei Service-Aktionen sind:

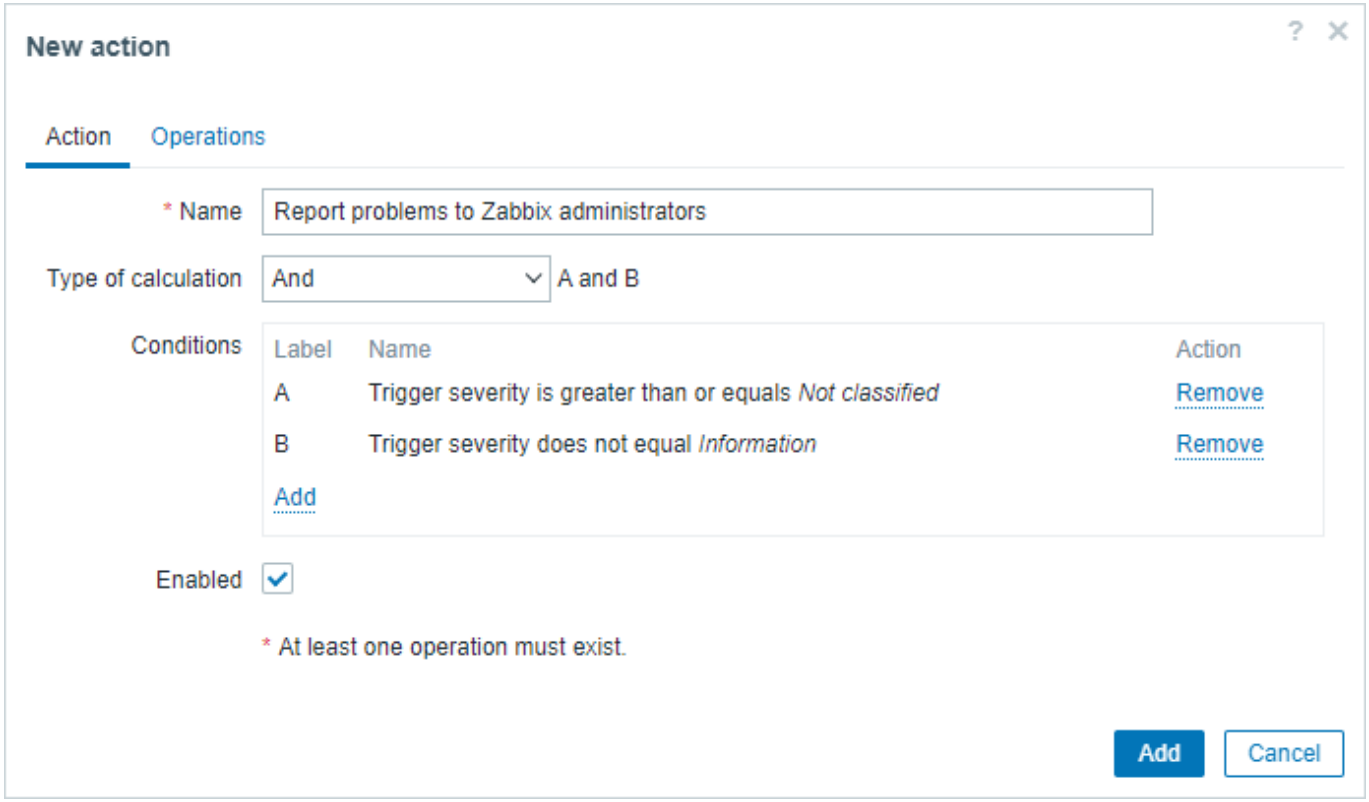
- Der Benutzerzugriff auf Service-Aktionen hängt von den Zugriffsrechten auf Services ab, die durch die Rolle des Benutzers gewährt werden
- Service-Aktionen unterstützen einen anderen Satz von Bedingungen

Konfigurieren einer Aktion

Um eine Aktion zu konfigurieren, gehen Sie wie folgt vor:

1. Gehen Sie zu *Alerts* > *Actions* und wählen Sie im Untermenü den gewünschten Aktionstyp aus (Sie können später über das Titeldropdown zu einem anderen Typ wechseln).
2. Klicken Sie auf *Create action*.
3. Geben Sie der Aktion einen Namen.
4. Wählen Sie die Bedingungen aus, unter denen die Operationen ausgeführt werden.
5. Wählen Sie die auszuführenden Operationen aus.

Allgemeine Aktionsattribute:



Alle Pflichtfelder sind mit einem roten Sternchen markiert.

Parameter	Beschreibung
Name	Eindeutiger Aktionsname.
Type of calculation	Wählen Sie die Auswertungs-Option für Aktionsbedingungen aus (bei mehr als einer Bedingung): **And** - alle Bedingungen müssen erfüllt sein. **Or** - es reicht aus, wenn eine Bedingung erfüllt ist. **And/Or** - Kombination aus beidem: AND bei unterschiedlichen Bedingungstypen und OR bei gleichem Bedingungstyp. **Custom expression** - eine benutzerdefinierte Berechnungsformel zur Auswertung von Aktionsbedingungen.
Conditions	Liste der Aktionsbedingungen. Klicken Sie auf *Add*, um eine neue Bedingung hinzuzufügen. Wenn keine Bedingungen konfiguriert sind, wird die Aktion für jedes Ereignis ausgeführt, das dem zu konfigurierenden Aktionstyp entspricht.
Enabled	Aktivieren Sie das Kontrollkästchen, um die Aktion zu aktivieren. Andernfalls wird sie deaktiviert.

Bedingungen

Übersicht

Es ist möglich festzulegen, dass eine Aktion nur ausgeführt wird, wenn das Ereignis einer definierten Menge von Bedingungen entspricht. Bedingungen werden bei der Konfiguration der Aktion festgelegt.

Der Abgleich von Bedingungen ist groß-/kleinschreibungssensitiv.

Auslöser-Aktionen

Die folgenden Bedingungen können in auslöserbasierten Aktionen verwendet werden:

Bedingungstyp	Unterstützte Operatoren	Beschreibung
Host-Gruppe	gleich ungleich	Geben Sie Host-Gruppen oder auszuschließende Host-Gruppen an. **gleich** - das Ereignis gehört zu dieser Host-Gruppe. **ungleich** - das Ereignis gehört nicht zu dieser Host-Gruppe. Die Angabe einer übergeordneten Host-Gruppe wählt implizit auch alle untergeordneten Host-Gruppen aus. Um nur die übergeordnete Gruppe anzugeben, müssen alle untergeordneten Gruppen zusätzlich mit dem Operator **ungleich** festgelegt werden.
Vorlage	gleich ungleich	Geben Sie Vorlagen oder auszuschließende Vorlagen an. **gleich** - das Ereignis gehört zu einem von dieser Vorlage geerbten Auslöser. **ungleich** - das Ereignis gehört nicht zu einem von dieser Vorlage geerbten Auslöser.
Host	gleich ungleich	Geben Sie Hosts oder auszuschließende Hosts an. **gleich** - das Ereignis gehört zu diesem Host. **ungleich** - das Ereignis gehört nicht zu diesem Host.
Tag-Name	gleich ungleich enthält enthält nicht	Geben Sie ein Ereignis-Tag oder ein auszuschließendes Ereignis-Tag an. **gleich** - das Ereignis hat dieses Tag. **ungleich** - das Ereignis hat dieses Tag nicht. **enthält** - das Ereignis hat ein Tag, das diese Zeichenfolge enthält. **enthält nicht** - das Ereignis hat kein Tag, das diese Zeichenfolge enthält.
Tag-Wert	gleich ungleich enthält enthält nicht	Geben Sie eine Kombination aus Ereignis-Tag und Wert oder eine auszuschließende Kombination aus Tag und Wert an. **gleich** - das Ereignis hat dieses Tag und diesen Wert. **ungleich** - das Ereignis hat dieses Tag und diesen Wert nicht. **enthält** - das Ereignis hat ein Tag und einen Wert, die diese Zeichenfolgen enthalten. **enthält nicht** - das Ereignis hat kein Tag und keinen Wert, die diese Zeichenfolgen enthalten.
Auslöser	gleich ungleich	Geben Sie Auslöser oder auszuschließende Auslöser an. **gleich** - das Ereignis wird von diesem Auslöser erzeugt. **ungleich** - das Ereignis wird von einem beliebigen anderen Auslöser außer diesem erzeugt.
Ereignisname	enthält enthält nicht	Geben Sie eine Zeichenfolge im Namen des vom Auslöser erzeugten Ereignisses oder eine auszuschließende Zeichenfolge an. Standardmäßig entspricht der Ereignisname dem Auslösernamen, sofern nicht in der Auslöser-Konfiguration ein benutzerdefinierter Ereignisname angegeben ist. **enthält** - der Ereignisname enthält diese Zeichenfolge. **enthält nicht** - diese Zeichenfolge ist vom Ereignisnamen ausgeschlossen. Hinweis: Der eingegebene Wert wird mit dem Ereignisnamen verglichen, wobei alle Makros expandiert werden.
Auslöser-Schweregrad	gleich ungleich ist größer oder gleich ist kleiner oder gleich	Geben Sie den Auslöser-Schweregrad an. **gleich** - entspricht dem Auslöser-Schweregrad. **ungleich** - entspricht nicht dem Auslöser-Schweregrad. **ist größer oder gleich** - größer oder gleich dem Auslöser-Schweregrad. **ist kleiner oder gleich** - kleiner oder gleich dem Auslöser-Schweregrad.
Zeitperiode	in nicht in	Geben Sie eine Zeitperiode oder eine auszuschließende Zeitperiode an. **in** - die Ereigniszeit liegt innerhalb der Zeitperiode. **nicht in** - die Ereigniszeit liegt nicht innerhalb der Zeitperiode. Eine Beschreibung des Formats finden Sie auf der Seite Spezifikation der Zeitperiode. Benutzermakros werden unterstützt.

Bedingungstyp	Unterstützte Operatoren	Beschreibung
Problem ist unterdrückt	nein ja	Geben Sie an, ob das Problem aufgrund einer Host-Wartung unterdrückt (nicht angezeigt) wird. **nein** - das Problem ist nicht unterdrückt. **ja** - das Problem ist unterdrückt.

Service-Aktionen

Für Service-Aktionen können die folgenden Bedingungen verwendet werden:

Condition type	Supported operators	Description
Service	equals does not equal	Geben Sie einen Service oder einen auszuschließenden Service an. **equals** - das Ereignis gehört zu diesem Service. **does not equal** - das Ereignis gehört nicht zu diesem Service. Die Angabe eines übergeordneten Service wählt implizit alle untergeordneten Services aus. Um nur den übergeordneten Service anzugeben, müssen alle verschachtelten Services zusätzlich mit dem Operator **does not equal** gesetzt werden.
Service name	contains does not contain	Geben Sie eine Zeichenfolge im Servicenamen oder eine auszuschließende Zeichenfolge an. **contains** - das Ereignis wird von einem Service erzeugt, dessen Name diese Zeichenfolge enthält. **does not contain** - diese Zeichenfolge kann im Servicenamen nicht gefunden werden.
Service tag name	equals does not equal contains does not contain	Geben Sie ein Ereignistag oder ein auszuschließendes Ereignistag an. Service-Ereignistags können im Konfigurationsabschnitt *Tags* des Service definiert werden. **equals** - das Ereignis hat dieses Tag. **does not equal** - das Ereignis hat dieses Tag nicht. **contains** - das Ereignis hat ein Tag, das diese Zeichenfolge enthält. **does not contain** - das Ereignis hat kein Tag, das diese Zeichenfolge enthält.
Service tag value	equals does not equal contains does not contain	Geben Sie eine Kombination aus Ereignistag und Wert oder eine auszuschließende Kombination aus Tag und Wert an. Service-Ereignistags können im Konfigurationsabschnitt *Tags* des Service definiert werden. **equals** - das Ereignis hat dieses Tag und diesen Wert. **does not equal** - das Ereignis hat dieses Tag und diesen Wert nicht. **contains** - das Ereignis hat ein Tag und einen Wert, die diese Zeichenfolgen enthalten. **does not contain** - das Ereignis hat kein Tag und keinen Wert, die diese Zeichenfolgen enthalten.

Attention:
Stellen Sie sicher, dass Sie Nachrichtenvorlagen für Service-Aktionen im Menü *Alerts* > *Media types* definieren. Andernfalls werden die Benachrichtigungen nicht gesendet.

Discovery-Aktionen

Die folgenden Bedingungen können in ereignisbasierten Aktionen verwendet werden:

Bedingungstyp	Unterstützte Operatoren	Beschreibung
Host-IP	ist gleich ist nicht gleich	Geben Sie einen IP-Adressbereich oder einen auszuschließenden Bereich für einen erkannten Host an. **ist gleich** - Die Host-IP befindet sich im Bereich. **ist nicht gleich** - Die Host-IP befindet sich nicht im Bereich. Folgende Formate sind möglich: Einzelne IP: 192.168.1.33 IP-Adressbereich: 192.168.1-10.1-254 IP-Maske: 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 Leerzeichen im Listenformat werden unterstützt.
Servicetyp	ist gleich ist nicht gleich	Geben Sie einen Servicetyp eines erkannten Dienstes oder einen auszuschließenden Servicetyp an. **ist gleich** - Entspricht dem erkannten Dienst. **ist nicht gleich** - Entspricht nicht dem erkannten Dienst. Verfügbare Servicetypen: SSH, LDAP, SMTP, FTP, HTTP, HTTPS, POP, NNTP, IMAP, TCP, Zabbix agent, SNMPv1 agent, SNMPv2 agent, SNMPv3 agent, ICMP ping, telnet.
Serviceport	ist gleich ist nicht gleich	Geben Sie einen TCP-Portbereich eines erkannten Dienstes oder einen auszuschließenden Bereich an. **ist gleich** - Der Serviceport befindet sich im Bereich. **ist nicht gleich** - Der Serviceport befindet sich nicht im Bereich.
Discovery-Regel	ist gleich ist nicht gleich	Geben Sie eine Discovery-Regel oder eine auszuschließende Discovery-Regel an. **ist gleich** - Diese Discovery-Regel wird verwendet. **ist nicht gleich** - Eine andere Discovery-Regel als diese wird verwendet.
Discovery-Prüfung	ist gleich ist nicht gleich	Geben Sie eine Discovery-Prüfung oder eine auszuschließende Discovery-Prüfung an. **ist gleich** - Diese Discovery-Prüfung wird verwendet. **ist nicht gleich** - Eine andere Discovery-Prüfung als diese wird verwendet.
Discovery-Objekt	ist gleich	Geben Sie das erkannte Objekt an. **ist gleich** - Entspricht dem erkannten Objekt (einem Gerät oder einem Dienst).
Discovery-Status	ist gleich	**Up** - Entspricht den Ereignissen „Host Up“ und „Service Up“. **Down** - Entspricht den Ereignissen „Host Down“ und „Service Down“. **Discovered** - Entspricht den Ereignissen „Host Discovered“ und „Service Discovered“. **Lost** - Entspricht den Ereignissen „Host Lost“ und „Service Lost“.
Betriebszeit/Ausfallzeit	ist größer oder gleich ist kleiner oder gleich	Betriebszeit für die Ereignisse „Host Up“ und „Service Up“. Ausfallzeit für die Ereignisse „Host Down“ und „Service Down“. **ist größer oder gleich** - Ist größer oder gleich. Der Parameter wird in Sekunden angegeben. **ist kleiner oder gleich** - Ist kleiner oder gleich. Der Parameter wird in Sekunden angegeben.
Empfangener Wert	ist gleich ist nicht gleich ist größer oder gleich ist kleiner oder gleich enthält enthält nicht	Geben Sie den Wert an, der von einer Agent-Prüfung (Zabbix, SNMP) in einer Discovery-Regel empfangen wurde. Zeichenfolgenvergleich. Wenn mehrere Zabbix-Agent- oder SNMP-Prüfungen für eine Regel konfiguriert sind, werden die empfangenen Werte für jede Prüfung überprüft (jede Prüfung erzeugt ein neues Ereignis, das mit allen Bedingungen abgeglichen wird). **ist gleich** - Entspricht dem Wert. **ist nicht gleich** - Entspricht nicht dem Wert. **ist größer oder gleich** - Ist größer oder gleich dem Wert. **ist kleiner oder gleich** - Ist kleiner oder gleich dem Wert. **enthält** - Enthält die Teilzeichenfolge. Der Parameter wird als Zeichenfolge angegeben. **enthält nicht** - Enthält die Teilzeichenfolge nicht. Der Parameter wird als Zeichenfolge angegeben.

Bedingungstyp	Unterstützte Operatoren	Beschreibung
Proxy	ist gleich ist nicht gleich	Geben Sie einen Proxy oder einen auszuschließenden Proxy an. **ist gleich** - Dieser Proxy wird verwendet. **ist nicht gleich** - Ein anderer Proxy als dieser wird verwendet.
Proxy-Gruppe	ist gleich ist nicht gleich	Geben Sie eine Proxy-Gruppe oder eine auszuschließende Proxy-Gruppe an. **ist gleich** - Diese Proxy-Gruppe wird verwendet. **ist nicht gleich** - Eine andere Proxy-Gruppe als diese wird verwendet.

Note:
Dienstprüfungen in einer Discovery-Regel, die Discovery-Ereignisse erzeugen, finden nicht gleichzeitig statt. Wenn daher **mehrere** Werte für die Bedingungen `Servicetyp`, `Serviceport` oder `Empfangener Wert` in der Aktion konfiguriert sind, werden sie jeweils mit einem Discovery-Ereignis verglichen, jedoch **nicht** gleichzeitig mit mehreren Ereignissen. Daher werden Aktionen mit mehreren Werten für dieselben Prüfungstypen möglicherweise nicht korrekt ausgeführt.

Aktionen zur automatischen Registrierung

Die folgenden Bedingungen können in Aktionen verwendet werden, die auf der automatischen Registrierung aktiver Agenten basieren:

Bedingungstyp	Unterstützte Operatoren	Beschreibung
Host-Metadaten	enthält enthält nicht entspricht entspricht nicht	Geben Sie Host-Metadaten oder auszuschließende Host-Metadaten an. **enthält** – Host-Metadaten enthalten die Zeichenfolge. **enthält nicht** – Host-Metadaten enthalten die Zeichenfolge nicht. Host-Metadaten können in einer Agent-Konfigurationsdatei angegeben werden. **entspricht** – Host-Metadaten entsprechen einem regulären Ausdruck. **entspricht nicht** – Host-Metadaten entsprechen keinem regulären Ausdruck.
Host-Name	enthält enthält nicht entspricht entspricht nicht	Geben Sie einen Host-Namen oder einen auszuschließenden Host-Namen an. **enthält** – Der Host-Name enthält die Zeichenfolge. **enthält nicht** – Der Host-Name enthält die Zeichenfolge nicht. **entspricht** – Der Host-Name entspricht einem regulären Ausdruck. **entspricht nicht** – Der Host-Name entspricht keinem regulären Ausdruck.
Proxy	ist gleich ist nicht gleich	Geben Sie einen Proxy oder einen auszuschließenden Proxy an. **ist gleich** – Dieser Proxy wird verwendet. **ist nicht gleich** – Jeder andere Proxy (oder Server) außer diesem Proxy wird verwendet.
Proxy-Gruppe	ist gleich ist nicht gleich	Geben Sie eine Proxy-Gruppe oder eine auszuschließende Proxy-Gruppe an. **ist gleich** – Diese Proxy-Gruppe wird verwendet. **ist nicht gleich** – Jede andere Proxy-Gruppe außer dieser wird verwendet.

Aktionen für interne Ereignisse

Die folgenden Bedingungen können für Aktionen auf Basis interner Ereignisse festgelegt werden:

Bedingungstyp	Unterstützte Operatoren	Beschreibung
Ereignistyp	gleich	**Datenpunkt im Status „nicht unterstützt“** - entspricht Ereignissen, bei denen ein Datenpunkt von einem „normalen“ in einen Status „nicht unterstützt“ wechselt. **Regel für Low-Level-Discovery im Status „nicht unterstützt“** - entspricht Ereignissen, bei denen eine Regel für Low-Level-Discovery von einem „normalen“ in einen Status „nicht unterstützt“ wechselt. **Auslöser im Status „unbekannt“** - entspricht Ereignissen, bei denen ein Auslöser von einem „normalen“ in einen Status „unbekannt“ wechselt.
Host-Gruppe	gleich ungleich	Geben Sie Host-Gruppen oder auszuschließende Host-Gruppen an. **gleich** - das Ereignis gehört zu dieser Host-Gruppe. **ungleich** - das Ereignis gehört nicht zu dieser Host-Gruppe.
Tag-Name	gleich ungleich enthält enthält nicht	Geben Sie ein Ereignis-Tag oder ein auszuschließendes Ereignis-Tag an. **gleich** - das Ereignis hat dieses Tag. **ungleich** - das Ereignis hat dieses Tag nicht. **enthält** - das Ereignis hat ein Tag, das diese Zeichenfolge enthält. **enthält nicht** - das Ereignis hat kein Tag, das diese Zeichenfolge enthält.
Tag-Wert	gleich ungleich enthält enthält nicht	Geben Sie eine Kombination aus Ereignis-Tag und Wert oder eine auszuschließende Kombination aus Tag und Wert an. **gleich** - das Ereignis hat dieses Tag und diesen Wert. **ungleich** - das Ereignis hat dieses Tag und diesen Wert nicht. **enthält** - das Ereignis hat ein Tag und einen Wert, die diese Zeichenfolgen enthalten. **enthält nicht** - das Ereignis hat kein Tag und keinen Wert, die diese Zeichenfolgen enthalten.
Vorlage	gleich ungleich	Geben Sie Vorlagen oder auszuschließende Vorlagen an. **gleich** - das Ereignis gehört zu einem Datenpunkt/Auslöser/einer Regel für Low-Level-Discovery, der/die von dieser Vorlage geerbt wurde. **ungleich** - das Ereignis gehört nicht zu einem Datenpunkt/Auslöser/einer Regel für Low-Level-Discovery, der/die von dieser Vorlage geerbt wurde.
Host	gleich ungleich	Geben Sie Hosts oder auszuschließende Hosts an. **gleich** - das Ereignis gehört zu diesem Host. **ungleich** - das Ereignis gehört nicht zu diesem Host.

Berechnungstyp

Die folgenden Optionen zur Berechnung von Bedingungen sind verfügbar:

- **Und** - alle Bedingungen müssen erfüllt sein

Beachten Sie, dass die Verwendung der Berechnung „Und“ zwischen mehreren Auslösern nicht zulässig ist, wenn diese als Bedingung `Trigger=` ausgewählt sind. Aktionen können nur auf Grundlage des Ereignisses eines einzelnen Auslösers ausgeführt werden.

- **Oder** - es reicht aus, wenn eine Bedingung erfüllt ist
- **Und/Oder** - Kombination aus beiden: UND bei unterschiedlichen Bedingungstypen und ODER bei demselben Bedingungstyp, zum Beispiel:

Host-Gruppe ist gleich Oracle-Server
 Host-Gruppe ist gleich MySQL-Server
 Ereignisname enthält 'Database is down'
 Ereignisname enthält 'Database is unavailable'

wird ausgewertet als

(Host-Gruppe ist gleich Oracle-Server **or** Host-Gruppe ist gleich MySQL-Server**) and (**Ereignisname enthält 'Database is down' **or** Ereignisname enthält 'Database is unavailable'**)**

- **Benutzerdefinierter Ausdruck** - eine benutzerdefinierte Berechnungsformel zur Auswertung von Aktionsbedingungen. Sie muss alle Bedingungen enthalten (dargestellt durch Großbuchstaben A, B, C, ...) und kann Leerzeichen, Tabulatoren, Klammern ( ), **and** (Groß-/Kleinschreibung beachten), **or** (Groß-/Kleinschreibung beachten), **not** (Groß-/Kleinschreibung beachten) enthalten.

Während das vorherige Beispiel mit `And/Or` als (A or B) and (C or D) dargestellt würde, können Sie in einem benutzerdefinierten Ausdruck auch viele andere Berechnungsarten verwenden:

(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
 usw.

Löschen von Objekten, die in Aktionen verwendet werden

Beim Versuch, Objekte zu löschen, die in einer Aktionsbedingung/-operation verwendet werden, erhält der Benutzer eine entsprechende Fehlermeldung.

Aktionen werden nicht deaktiviert, wenn Aktionsbedingungen oder -operationen:

- gelöschte Objekte enthalten;
- Objekte enthalten, die nicht mehr durch Low-Level-Discovery erkannt werden und entfernt wurden (nach dem Zeitraum *Delete lost resources*).

Aktionsbedingungen/-operationen werden nicht entfernt, wenn ihnen Objekte fehlen. Stattdessen werden gelöschte Objekte als *Deleted host group*, *Deleted host*, *Deleted trigger* usw. angezeigt. In der Spalte „Info“ der Aktionsliste wird ein Warnsymbol mit der Meldung „This action has conditions or operations referencing deleted object(s).“ angezeigt.

Vorgänge

Übersicht

Sie können die folgenden Operationen für alle Ereignisse definieren:

- Eine Nachricht senden
- Einen Remote-Befehl ausführen

> **Attention:**
> Der Zabbix Server erstellt keine Benachrichtigungen, wenn der Zugriff auf den Host für den als Empfänger der Aktionsoperation definierten Benutzer explizit „verweigert“ wird oder wenn für den Benutzer überhaupt keine Berechtigungen für den Host definiert sind.

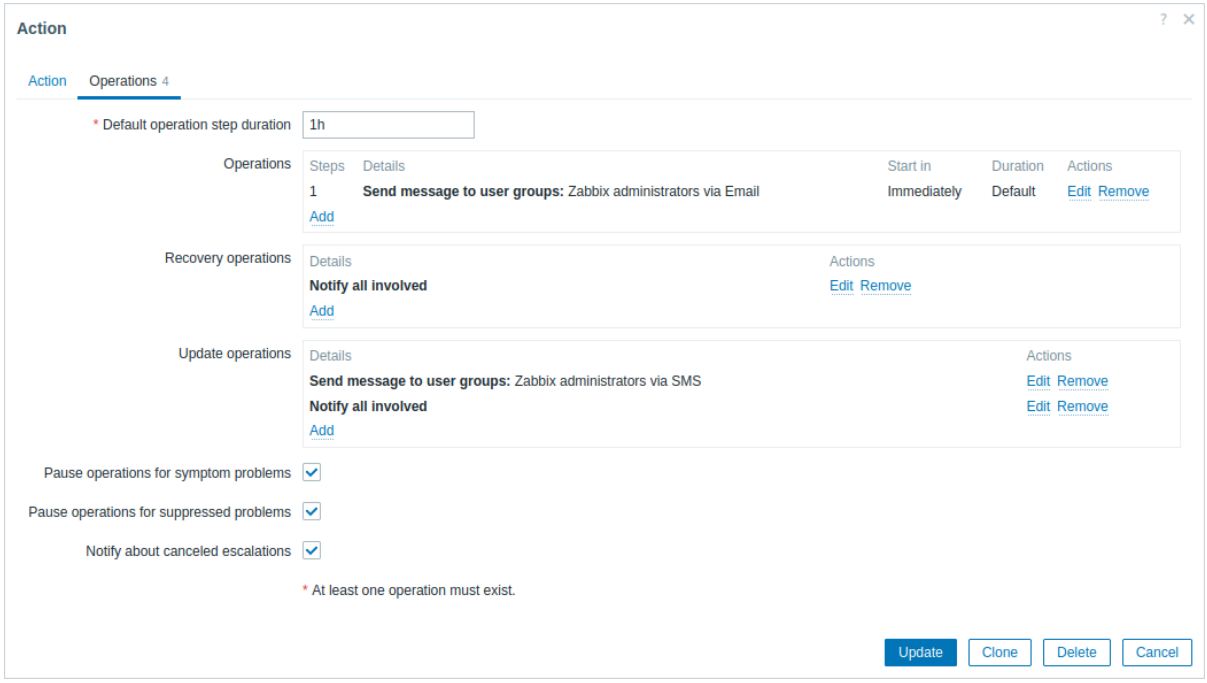
Für Discovery- und Autoregistrierungsereignisse sind zusätzliche Operationen verfügbar:

- Host hinzufügen
- Host entfernen
- Host aktivieren
- Host deaktivieren
- Zu Hostgruppe hinzufügen
- Aus Hostgruppe entfernen
- Host-Tags hinzufügen
- Host-Tags entfernen
- Vorlage verknüpfen
- Verknüpfung der Vorlage aufheben
- Modus des Host-Inventars festlegen

Konfigurieren eines Vorgangs

So konfigurieren Sie einen Vorgang:

1. Wechseln Sie in der Konfiguration der Aktion zur Registerkarte *Vorgänge*.
2. Konfigurieren Sie die Details des Vorgangs.



Allgemeine Attribute von Vorgängen:

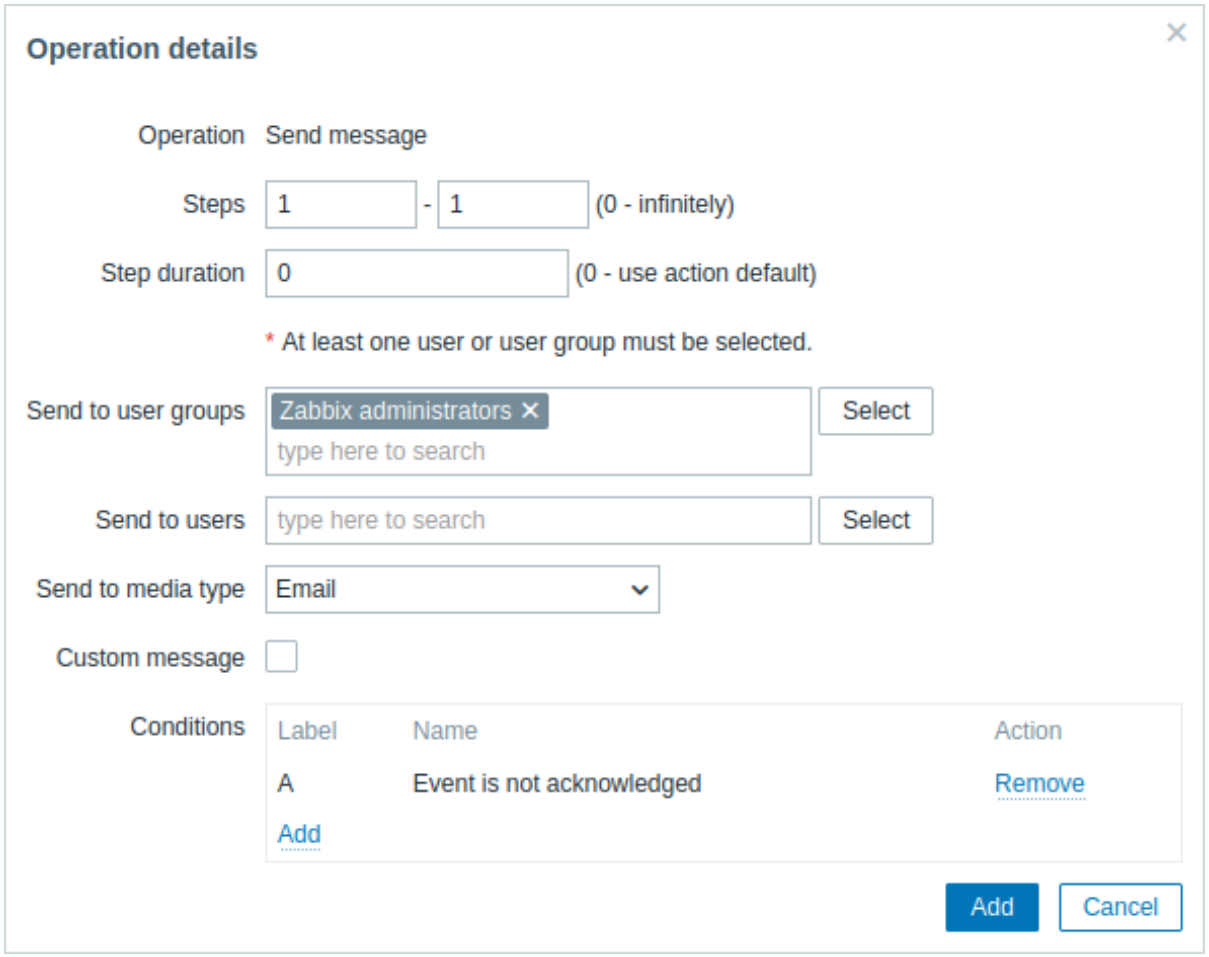
Parameter	Beschreibung
Standarddauer eines Vorgangsschritts	Standarddauer eines Vorgangsschritts (60 Sekunden bis 1 Woche). Beispiel: Bei einer Schrittdauer von einer Stunde vergeht nach der Ausführung eines Vorgangs eine Stunde, bevor der nächste Schritt ausgeführt wird. Zeitsuffixe werden unterstützt, z. B. 60s, 1m, 2h, 1d. Benutzermakros werden unterstützt.
Vorgänge	Aktionsvorgänge (falls vorhanden) werden mit folgenden Details angezeigt: **Schritte** – Eskalationsschritt(e), denen der Vorgang zugewiesen ist. **Details** – Typ des Vorgangs und dessen Empfänger/Ziel. Die Vorgangsliste zeigt außerdem den verwendeten Medientyp (E-Mail, SMS oder Skript) sowie den Vor- und Nachnamen (in Klammern nach dem Benutzernamen) eines Benachrichtigungsempfängers an. **Start in** – wie lange nach einem Ereignis der Vorgang ausgeführt wird. **Dauer (Sek.)** – die Schrittdauer wird angezeigt. *Standard* wird angezeigt, wenn der Schritt die Standarddauer verwendet; bei Verwendung einer benutzerdefinierten Dauer wird die Zeit angezeigt. **Aktionen** – Links zum Bearbeiten und Entfernen eines Vorgangs werden angezeigt.
Wiederherstellungsvorgänge	Aktionsvorgänge (falls vorhanden) werden mit folgenden Details angezeigt: **Details** – Typ des Vorgangs und dessen Empfänger/Ziel. Die Vorgangsliste zeigt außerdem den verwendeten Medientyp (E-Mail, SMS oder Skript) sowie den Vor- und Nachnamen (in Klammern nach dem Benutzernamen) eines Benachrichtigungsempfängers an. **Aktionen** – Links zum Bearbeiten und Entfernen eines Vorgangs werden angezeigt.
Aktualisierungsvorgänge	Aktionsvorgänge (falls vorhanden) werden mit folgenden Details angezeigt: **Details** – Typ des Vorgangs und dessen Empfänger/Ziel. Die Vorgangsliste zeigt außerdem den verwendeten Medientyp (E-Mail, SMS oder Skript) sowie den Vor- und Nachnamen (in Klammern nach dem Benutzernamen) eines Benachrichtigungsempfängers an. **Aktionen** – Links zum Bearbeiten und Entfernen eines Vorgangs werden angezeigt.
Vorgänge für Symptomprobleme pausieren	Aktivieren Sie dieses Kontrollkästchen, um Vorgänge (nach dem ersten Vorgang) für Symptomprobleme zu pausieren. Beachten Sie, dass diese Einstellung nur Problemeskalationen betrifft; Wiederherstellungs- und Aktualisierungsvorgänge sind nicht betroffen. Diese Option ist nur für *Auslöseraktionen* verfügbar.
Vorgänge für unterdrückte Probleme pausieren	Aktivieren Sie dieses Kontrollkästchen, um den Start von Vorgängen für die Dauer eines Wartungszeitraums zu verzögern. Nach dem Wartungszeitraum werden beim Start der Vorgänge alle Vorgänge ausgeführt, einschließlich der Vorgänge für Ereignisse während des Wartungszeitraums. Beachten Sie, dass diese Einstellung nur Problemeskalationen betrifft; Wiederherstellungs- und Aktualisierungsvorgänge sind nicht betroffen. Wenn Sie dieses Kontrollkästchen deaktivieren, werden Vorgänge auch während eines Wartungszeitraums ohne Verzögerung ausgeführt. Diese Option ist für *Serviceaktionen* nicht verfügbar.

Parameter	Beschreibung
Über abgebrochene Eskalationen benachrichtigen	Deaktivieren Sie dieses Kontrollkästchen, um Benachrichtigungen über abgebrochene Eskalationen zu deaktivieren (wenn Host, Datenpunkt, Auslöser oder Aktion deaktiviert wird).

Alle Pflichtfelder sind mit einem roten Sternchen gekennzeichnet.

Um die Details eines neuen Vorgangs zu konfigurieren, klicken Sie im Block *Vorgänge* auf Add. Um einen vorhandenen Vorgang zu bearbeiten, klicken Sie neben dem Vorgang auf Edit. Es wird ein Popup-Fenster geöffnet, in dem Sie die Details des Vorgangsschritts bearbeiten können.

Operationsdetails



Parameter	Beschreibung
Operation	Wählen Sie die Operation aus: **Send message** – Nachricht an den Benutzer senden. **<remote command name>** – einen Remote-Befehl ausführen. Befehle können ausgeführt werden, wenn sie zuvor in global scripts mit dem als Geltungsbereich ausgewählten *Action operation* definiert wurden. Für Ereignisse der Erkennung und automatischen Registrierung sind weitere Operationen verfügbar (siehe oben).
Steps	Wählen Sie den bzw. die Schritte aus, denen die Operation in einem Eskalations-Zeitplan zugewiesen werden soll: **From** – ab diesem Schritt ausführen. **To** – bis zu diesem Schritt ausführen (0=unendlich, die Ausführung wird nicht begrenzt).

Parameter	Beschreibung
Step duration	Benutzerdefinierte Dauer für diese Schritte (0=Standarddauer des Schritts verwenden). Zeitsuffixe werden unterstützt, z. B. 60s, 1m, 2h, 1d. Benutzermakros werden unterstützt. Mehrere Operationen können demselben Schritt zugewiesen werden. Wenn für diese Operationen unterschiedliche Schritt-Dauern definiert sind, wird die kürzeste berücksichtigt und auf den Schritt angewendet.
Operation type: send message	
Send to user groups	Wählen Sie die Benutzergruppen aus, an die die Nachricht gesendet werden soll. Die Benutzergruppe muss mindestens über „Lesen“-Berechtigungen für den Host verfügen, damit sie benachrichtigt werden kann.
Send to users	Wählen Sie die Benutzer aus, an die die Nachricht gesendet werden soll. Der Benutzer muss mindestens über „Lesen“-Berechtigungen für den Host verfügen, damit er benachrichtigt werden kann.
Send to media type	Senden Sie die Nachricht an alle verfügbaren (konfigurierten und aktivierten) Medientypen oder nur an einen bestimmten Medientyp.
Custom message	Wenn diese Option ausgewählt ist, kann die benutzerdefinierte Nachricht konfiguriert werden. Für Benachrichtigungen über interne Ereignisse über webhooks ist eine benutzerdefinierte Nachricht erforderlich.
Subject	Betreff der benutzerdefinierten Nachricht. Der Betreff kann Makros enthalten. Er ist auf 255 Zeichen begrenzt.
Message	Die benutzerdefinierte Nachricht. Die Nachricht kann Makros enthalten. Sie ist abhängig vom Datenbanktyp auf eine bestimmte Anzahl von Zeichen begrenzt (weitere Informationen finden Sie unter Sending message).
Operation type: remote command	
Target list	Wählen Sie die Ziele aus, auf denen der Befehl ausgeführt werden soll: **Current host** – der Befehl wird auf dem Host des Auslösers ausgeführt, der das Problemereignis verursacht hat. Diese Option funktioniert nicht, wenn der Auslöser mehrere Hosts enthält. **Host** – wählen Sie den bzw. die Hosts aus, auf denen der Befehl ausgeführt werden soll. **Host group** – wählen Sie die Hostgruppe(n) aus, auf denen der Befehl ausgeführt werden soll. Durch Angabe einer übergeordneten Hostgruppe werden implizit alle verschachtelten Hostgruppen ausgewählt. Daher wird der Remote-Befehl auch auf Hosts aus verschachtelten Gruppen ausgeführt. Ein Befehl auf einem Host wird nur einmal ausgeführt, selbst wenn der Host mehr als einmal übereinstimmt (z. B. aus mehreren Hostgruppen, einzeln und aus einer Hostgruppe). Die Zielliste ist bedeutungslos, wenn ein benutzerdefiniertes Skript auf dem Zabbix-Server ausgeführt wird. Die Auswahl weiterer Ziele führt in diesem Fall lediglich dazu, dass das Skript mehrmals auf dem Server ausgeführt wird. Beachten Sie, dass die Zielauswahl für globale Skripte auch von der Einstellung *Host group* in der Konfiguration des globalen Skripts abhängt. Die Option *Target list* ist für *Service actions* nicht verfügbar, da Remote-Befehle in diesem Fall immer auf dem Zabbix-Server ausgeführt werden.

Parameter	Beschreibung
Conditions	Bedingung für die Ausführung der Operation: **Event is not acknowledged** – nur, wenn das Ereignis nicht bestätigt wurde. **Event is acknowledged** – nur, wenn das Ereignis bestätigt wurde. Die Option *Conditions* ist nur für *Trigger actions* verfügbar.

Klicken Sie anschließend auf *Add*, um die Operation zur Liste der *Operations* hinzuzufügen.

Senden einer Nachricht

Übersicht

Das Senden einer Nachricht ist eine der besten Möglichkeiten, Personen über ein Problem zu benachrichtigen. Deshalb ist es eine der wichtigsten von Zabbix angebotenen Aktionen.

Konfiguration

Um Benachrichtigungen von Zabbix senden und empfangen zu können, müssen Sie:

- Das Medium definieren, an das eine Nachricht gesendet werden soll.

Wenn der Vorgang außerhalb des für das ausgewählte Medium in der Benutzerkonfiguration definierten Zeitraums Wenn aktiv stattfindet, wird die Nachricht nicht gesendet.

Der standardmäßige Auslöser-Schweregrad („Nicht klassifiziert“) **muss** in der Benutzermedienkonfiguration aktiviert sein, wenn Sie Benachrichtigungen für Ereignisse ohne Auslöser empfangen möchten, z. B. für Discovery, die aktive automatische Agent-Registrierung oder interne Ereignisse.

- Eine Aktionsoperation konfigurieren, die eine Nachricht an eines der definierten Medien sendet.

> **Attention:**
> Zabbix sendet Benachrichtigungen nur an Benutzer, die mindestens über „Lesen“-Berechtigungen für den Host verfügen, auf dem das Ereignis generiert wurde. Mindestens ein Host eines Auslöserausdrucks muss zugänglich sein.

Sie können benutzerdefinierte Szenarien zum Senden von Nachrichten mithilfe von Eskalationen konfigurieren.

Damit E-Mails von Zabbix erfolgreich empfangen und gelesen werden können, müssen E-Mail-Server/-Clients das standardmäßige Format „SMTP/MIME email“ unterstützen, da Zabbix UTF-8-Daten sendet (wenn der Betreff ausschließlich ASCII-Zeichen enthält, wird er nicht UTF-8-codiert). Der Betreff und der Nachrichtentext werden gemäß dem Standardformat „SMTP/MIME email“ base64-codiert.

Das Nachrichtenlimit nach der Auflösung aller Makros entspricht dem Nachrichtenlimit für Remote-Befehle.

Nachrichtenverfolgung

Sie können den Status der gesendeten Nachrichten unter *Monitoring* > *Problems* einsehen.

In der Spalte *Actions* sehen Sie zusammengefasste Informationen zu ausgeführten Aktionen. Dabei stehen grüne Zahlen für gesendete Nachrichten, rote Zahlen für fehlgeschlagene Nachrichten. *In progress* zeigt an, dass eine Aktion eingeleitet wurde. *Failed* informiert darüber, dass keine Aktion erfolgreich ausgeführt wurde.

Wenn Sie auf die Ereigniszeit klicken, um die Ereignisdetails anzuzeigen, können Sie im Block *Actions* Details zu den aufgrund des Ereignisses gesendeten (oder nicht gesendeten) Nachrichten sehen.

Unter *Reports* > *Action log* sehen Sie Details zu allen ausgeführten Aktionen für jene Ereignisse, für die eine Aktion konfiguriert ist.

Remote-Befehle

Übersicht

Mit Remote-Befehlen können Sie festlegen, dass ein bestimmter vordefinierter Befehl bei einer bestimmten Bedingung automatisch auf dem überwachten Host ausgeführt wird.

Daher sind Remote-Befehle ein leistungsfähiger Mechanismus für eine intelligente proaktive Überwachung.

Bei den naheliegendsten Anwendungsfällen können Sie beispielsweise:

- eine Anwendung (Webserver, Middleware, CRM) automatisch neu starten, wenn sie nicht reagiert
- den IPMI-Befehl „reboot“ verwenden, um einen entfernten Server neu zu starten, wenn dieser keine Anfragen beantwortet
- automatisch Speicherplatz freigeben (ältere Dateien entfernen, `/tmp` bereinigen), wenn der Speicherplatz knapp wird
- eine VM abhängig von der CPU-Auslastung von einem physischen Rechner auf einen anderen migrieren
- bei unzureichenden CPU-, Festplatten-, Arbeitsspeicher- oder sonstigen Ressourcen neue Knoten zu einer Cloud-Umgebung hinzufügen

Das Konfigurieren einer Aktion für Remote-Befehle ähnelt dem Konfigurieren einer Aktion zum Senden einer Nachricht. Der einzige Unterschied besteht darin, dass Zabbix einen Befehl anstelle einer Nachricht ausführt.

Remote-Befehle können vom Zabbix Server, Zabbix Proxy oder Zabbix Agent ausgeführt werden. Beachten Sie Folgendes:

- Remote-Befehle auf dem Zabbix Agent können direkt vom Zabbix Server oder über einen Zabbix Proxy ausgeführt werden.
- Remote-Befehle, die vom Zabbix Server ausgeführt werden, werden wie unter Befehlsausführung beschrieben ausgeführt, einschließlich der Überprüfung des Exit-Codes.
- Remote-Befehle werden auch dann ausgeführt, wenn sich der Ziel-Host im Wartungsmodus befindet.

Remote-Befehle sind sowohl auf dem Zabbix Agent als auch auf dem Zabbix Proxy standardmäßig deaktiviert. Sie können wie folgt aktiviert werden:

- Für Remote-Befehle, die vom Zabbix Agent ausgeführt werden, fügen Sie in der Agent-Konfiguration für jeden zulässigen Befehl einen Parameter `AllowKey=system.run[<command>,*]` oder `AllowKeyRegexp=^system\.run\[<command>,.*\]$` hinzu, wobei `*` entweder den Ausführungsmodus (`wait` oder `nowait`) bezeichnet.
- Für Remote-Befehle, die vom Zabbix Proxy ausgeführt werden, setzen Sie den Parameter `EnableRemoteCommands` in der Proxy-Konfiguration auf 1. Beachten Sie, dass Remote-Befehle auf dem Zabbix Proxy nicht aktiviert werden müssen, wenn sie vom Zabbix Agent auf einem vom Proxy überwachten Host ausgeführt werden.

Attention:
Verwenden Sie `AllowKey=system.run[*]` oder `AllowKeyRegexp=^system\.run\[.*\]$` nur, wenn dies erforderlich ist, da dadurch die Ausführung beliebiger Befehle auf dem Host ermöglicht wird. Aus Sicherheitsgründen wird empfohlen, nur exakt festgelegte Befehle zuzulassen. `system.run[*]` ist standardmäßig abgelehnt (`DenyKey=system.run[*]`).

Siehe auch Agent-Prüfungen einschränken.

Limit für Remote-Befehle

Das Limit für Remote-Befehle nach dem Auflösen aller Makros hängt vom Typ der Datenbank und vom Zeichensatz ab (Nicht-ASCII-Zeichen benötigen mehr als ein Byte zur Speicherung):

Datenbank	Limit in Zeichen	Limit in Byte
MySQL	65535	65535
PostgreSQL	65535	nicht begrenzt
SQLite (nur Zabbix Proxy)	65535	nicht begrenzt

Die Ausgabe der Ausführung von Remote-Befehlen (Rückgabewert) ist auf 16 MB begrenzt (einschließlich nachgestellter Leerzeichen, die abgeschnitten werden). Das Limit für IPMI-Remote-Befehle basiert auf der installierten IPMI-Bibliothek. Beachten Sie, dass Datenbanklimits für alle Remote-Befehle gelten.

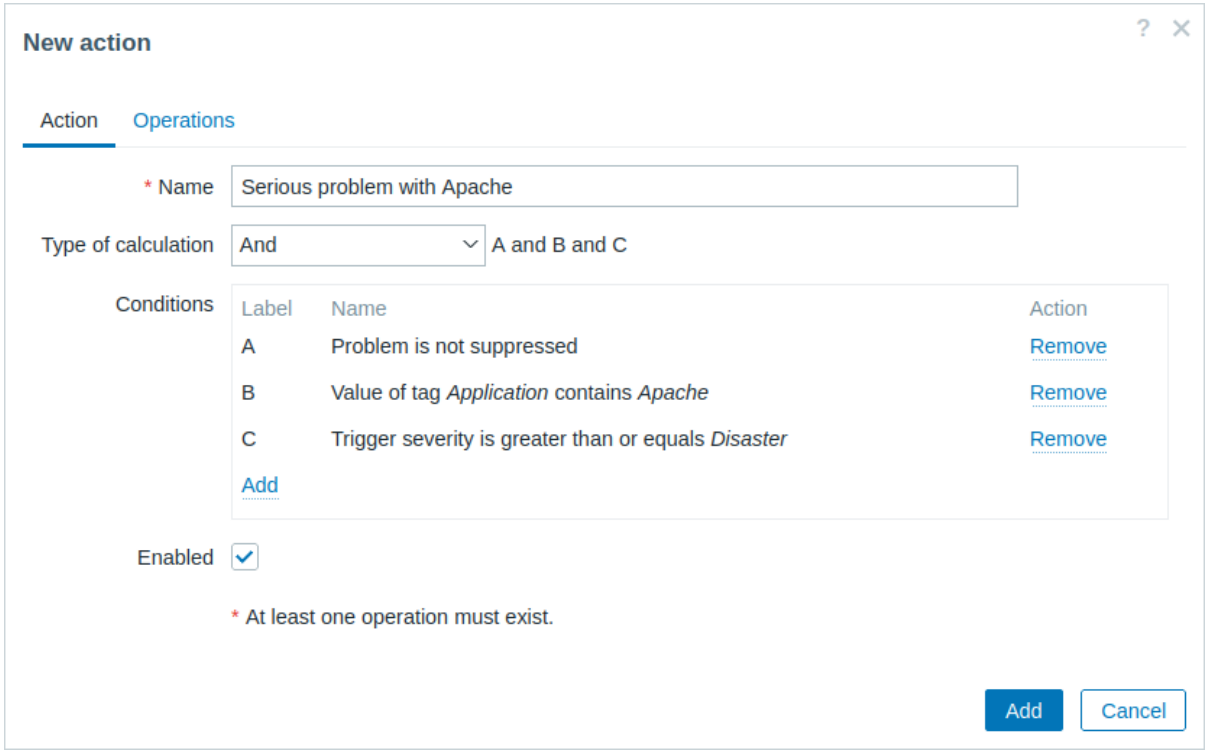
Konfiguration

Remote-Befehle, die vom Zabbix Agent ausgeführt werden, müssen zunächst in der Agent-Konfiguration mithilfe des Parameters `AllowKey` oder `AllowKeyRegexp` erlaubt werden.

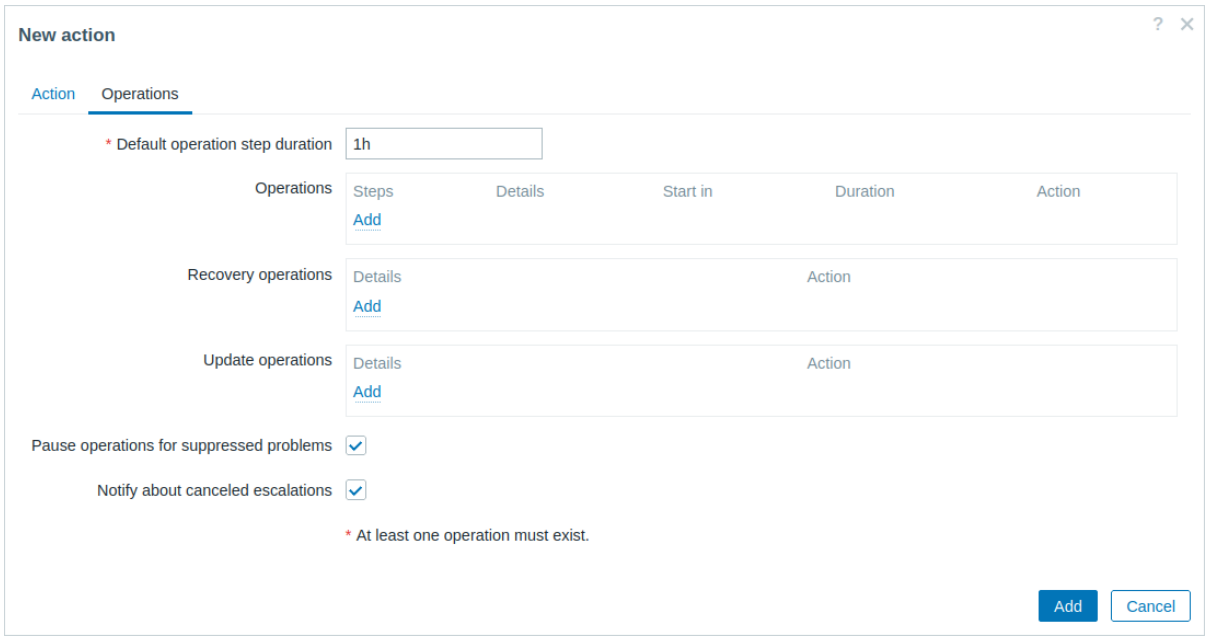
Stellen Sie sicher, dass der Parameter `AllowKey=system.run[<command>,*]` oder `AllowKeyRegexp=^system\.run\[<command>,.*\` für jeden erlaubten Befehl zur Agent-Konfiguration hinzugefügt wird, damit dessen Ausführung im `nowait`-Modus möglich ist. Starten Sie den Agent-Daemon nach der Änderung des Parameters neu.

Wenn Sie anschließend unter *Alerts* > *Actions* > *Trigger actions* eine neue Aktion konfigurieren:

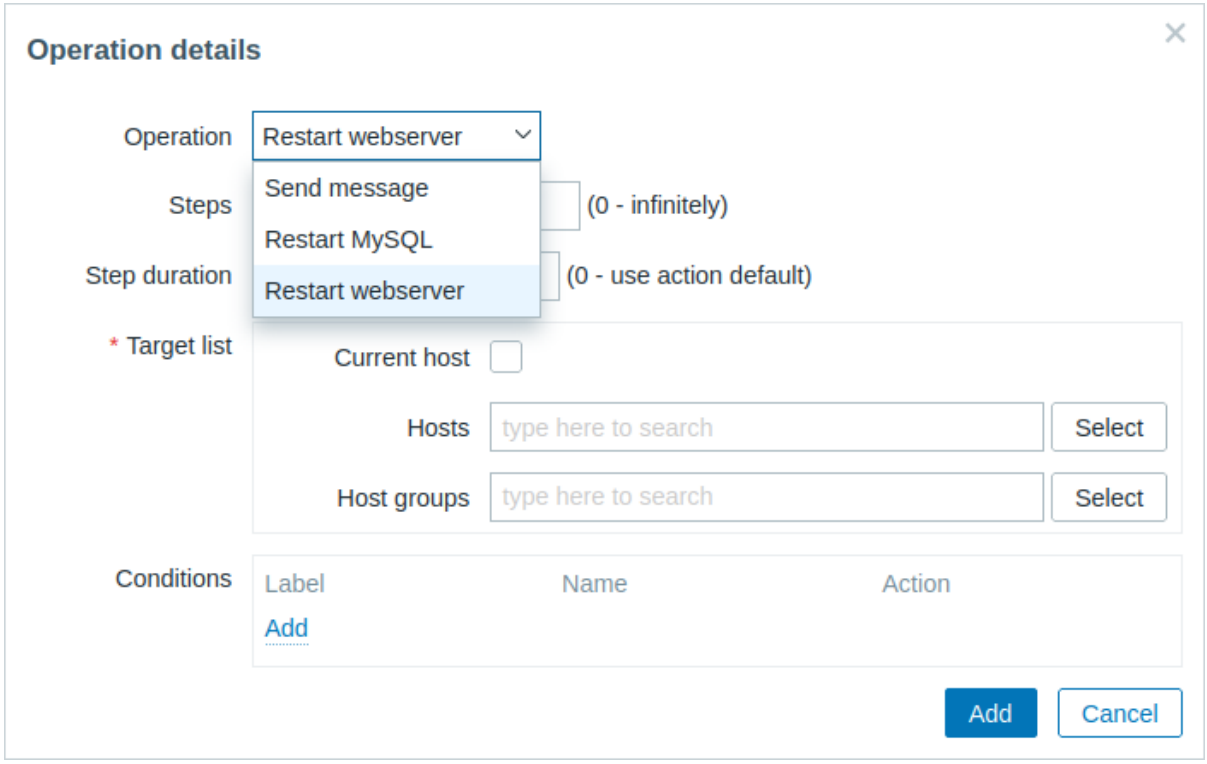
1. Legen Sie die entsprechenden Bedingungen fest, zum Beispiel, dass die Aktion bei beliebigen Problemen mit dem Schweregrad „Disaster“ in einer der Apache-Anwendungen aktiviert wird.



2. Klicken Sie auf der Registerkarte *Operations* im Abschnitt *Operations*, *Recovery operations* oder *Update operations* auf *Add*.



3. Wählen Sie einen der vordefinierten Skripte aus der Dropdown-Liste *Operation* aus und legen Sie die *Target list* für das Skript fest.



Vorgegebene Skripte

Skripte, die für Aktionsoperationen (webhook, Skript, SSH, Telnet, IPMI) verfügbar sind, werden in globalen Skripten mit der ausgewählten Geltung *Action operation* definiert.

Zum Beispiel:

```
sudo /etc/init.d/apache restart
```

In diesem Fall versucht Zabbix, einen Apache-Prozess neu zu starten. Stellen Sie bei diesem Befehl sicher, dass der Befehl auf dem Zabbix Agent ausgeführt wird (klicken Sie auf die Schaltfläche *Zabbix agent* neben *Execute on*).

Attention:
Beachten Sie die Verwendung von `sudo` - der Zabbix-Benutzer hat standardmäßig keine Berechtigungen, Systemdienste neu zu starten. Siehe unten für Hinweise zur Konfiguration von `sudo`.

Note:
Ab Zabbix Agent 7.0 können Remote-Befehle auch auf einem Agent ausgeführt werden, der im aktiven Modus arbeitet. Der Zabbix Agent - ob aktiv oder passiv - sollte auf dem entfernten Host ausgeführt werden und führt die Befehle im Hintergrund aus.

Remote-Befehle auf dem Zabbix Agent werden ohne Timeout mit dem Schlüssel `system.run[,nowait]` ausgeführt und nicht auf Ausführungsergebnisse geprüft. Auf dem Zabbix Server und dem Zabbix Proxy werden Remote-Befehle mit dem Timeout ausgeführt, das im Parameter `TrapperTimeout` der Datei `zabbix_server.conf` oder `zabbix_proxy.conf` festgelegt ist, und auf Ausführungsergebnisse geprüft. Weitere Informationen finden Sie unter *Script timeout*.

Zugriffsberechtigungen

Stellen Sie sicher, dass der Benutzer `zabbix` Ausführungsberechtigungen für die konfigurierten Befehle hat. Möglicherweise möchten Sie `sudo` verwenden, um Zugriff auf privilegierte Befehle zu gewähren. Um den Zugriff zu konfigurieren, führen Sie als root aus:

```
visudo
```

Beispielzeilen, die in der Datei `sudoers` verwendet werden könnten:

```
# erlaubt dem Benutzer 'zabbix', alle Befehle ohne Passwort auszuführen.
zabbix ALL=NOPASSWD: ALL

# erlaubt dem Benutzer 'zabbix', apache ohne Passwort neu zu starten.
zabbix ALL=NOPASSWD: /etc/init.d/apache restart
```

Note:
Auf einigen Systemen verhindert die Datei `sudoers`, dass nicht-lokale Benutzer Befehle ausführen. Um dies zu ändern, kommentieren Sie die Option `requiretty` in `/etc/sudoers` aus.

Remote-Befehle mit mehreren Schnittstellen

Wenn das Zielsystem mehrere Schnittstellen des ausgewählten Typs (Zabbix-Agent oder IPMI) hat, werden Remote-Befehle auf der Standard- Schnittstelle ausgeführt.

Es ist möglich, Remote-Befehle über SSH und Telnet unter Verwendung einer anderen Schnittstelle als der des Zabbix-Agenten auszuführen. Die verfügbare zu verwendende Schnittstelle wird in der folgenden Reihenfolge ausgewählt:

- Zabbix-Agent-Standard-Schnittstelle
- SNMP-Standard-Schnittstelle
- JMX-Standard-Schnittstelle
- IPMI-Standard-Schnittstelle

IPMI-Fernbefehle

Für IPMI-Fernbefehle sollte die folgende Syntax verwendet werden:

`<command> [<value>]`

wobei

- <command> - einer der IPMI-Befehle ohne Leerzeichen
- <value> - 'on', 'off' oder eine beliebige vorzeichenlose Ganzzahl. <value> ist ein optionaler Parameter.

Beispiele

Beispiele für globale Skripte, die als Remote-Befehle in Aktionsoperationen verwendet werden können.

Beispiel 1

Neustart von Windows unter bestimmten Bedingungen.

Um Windows bei einem von Zabbix erkannten Problem automatisch neu zu starten, definieren Sie das folgende Skript:

Script parameter	Value
Scope	'Action operation'
Type	'Script'
Command	`c:\windows\system32\shutdown.exe -r -f`

Beispiel 2

Starten Sie den Host mithilfe der IPMI-Steuerung neu.

Skriptparameter	Wert
Geltungsbereich	'Aktionsoperation'
Typ	'IPMI'
Befehl	reset

Beispiel 3

Schalten Sie den Host aus, indem Sie die IPMI-Steuerung verwenden.

Skriptparameter	Wert
Bereich	'Aktionsoperation'
Typ	'IPMI'
Befehl	power off

Zusätzliche Operationen

Übersicht

In diesem Abschnitt finden Sie einige Details zu zusätzlichen Operationen für Discovery-/Autoregistrierungsereignisse.

Host hinzufügen

Hosts werden während des Discovery-Prozesses hinzugefügt, sobald ein Host entdeckt wird, und nicht erst am Ende des Discovery-Prozesses.

> **Note:**
> Da die Netzwerk-Discovery aufgrund vieler nicht verfügbarer Hosts/Services einige Zeit in Anspruch nehmen kann, wird empfohlen, Geduld zu haben und sinnvolle IP-Bereiche zu verwenden.

Beim Hinzufügen eines Hosts wird sein Name durch die Standardfunktion **gethostbyname** festgelegt. Wenn der Host aufgelöst werden kann, wird der aufgelöste Name verwendet. Andernfalls wird die IP-Adresse verwendet. Wenn außerdem eine IPv6-Adresse als Hostname verwendet werden muss, werden alle ":" (Doppelpunkte) durch "_" (Unterstriche) ersetzt, da Doppelpunkte in Hostnamen nicht zulässig sind.

> **Attention:**
> Wenn die Discovery durch einen Proxy durchgeführt wird, findet die Hostname-Auflösung derzeit weiterhin auf dem Zabbix Server statt.

> **Attention:**
> Wenn in der Zabbix-Konfiguration bereits ein Host mit demselben Namen wie ein neu entdeckter existiert, fügt Zabbix **_N** zum Hostnamen hinzu, wobei **N** eine aufsteigende Zahl ist, beginnend mit 2.

Verwendung von Makros in Nachrichten

Übersicht

In Nachrichtenbetreffs und Nachrichtentexten können Sie Makros für eine effizientere Problemberichterstattung verwenden.

Zusätzlich zu einer Reihe integrierter Makros werden auch Benutzermakros und Ausdrucksmakros unterstützt. Eine vollständige Liste der von Zabbix unterstützten Makros ist verfügbar.

Examples

Die folgenden Beispiele zeigen, wie Sie Makros in Nachrichten verwenden können.

Beispiel 1

Betreff der Nachricht:

```
Problem: {TRIGGER.NAME}
```

Wenn Sie die Nachricht erhalten, wird der Betreff der Nachricht durch etwa Folgendes ersetzt:

```
Problem: Prozessorlast ist auf dem Zabbix Server zu hoch
```

Beispiel 2

Nachricht:

```
Prozessorlast ist: {?last(/zabbix.zabbix.com/system.cpu.load[,avg1])}
```

Wenn Sie die Nachricht erhalten, wird die Nachricht durch etwa Folgendes ersetzt:

```
Prozessorlast ist: 1.45
```

Beispiel 3

Nachricht:

```
Letzter Wert: {?last(/{HOST.HOST}/{ITEM.KEY})}
MAX für 15 Minuten: {?max(/{HOST.HOST}/{ITEM.KEY},15m)}
MIN für 15 Minuten: {?min(/{HOST.HOST}/{ITEM.KEY},15m)}
```

Wenn Sie die Nachricht erhalten, wird die Nachricht durch etwa Folgendes ersetzt:

```
Letzter Wert: 1.45
MAX für 15 Minuten: 2.33
MIN für 15 Minuten: 1.01
```

Beispiel 4

Nachricht:

```
http://<server_ip_or_name>/zabbix/tr_events.php?triggerid={TRIGGER.ID}&eventid={EVENT.ID}
```

Wenn Sie die Nachricht erhalten, enthält sie einen Link zur Seite *Ereignisdetails*, die Informationen über das Ereignis, seinen Auslöser und eine Liste der neuesten Ereignisse bereitstellt, die durch denselben Auslöser erzeugt wurden.

Beispiel 5

Information über Werte von mehreren Hosts in einem Auslöserausdruck.

Nachricht:

```
Problemname: {TRIGGER.NAME}
Auslöserausdruck: {TRIGGER.EXPRESSION}

1. Datenpunktwert auf {HOST.NAME1}: {ITEM.VALUE1} ({ITEM.NAME1})
2. Datenpunktwert auf {HOST.NAME2}: {ITEM.VALUE2} ({ITEM.NAME2})
```

Wenn Sie die Nachricht erhalten, wird sie etwa wie folgt ersetzt:

```
Problemname: Prozessorlast ist auf einem lokalen Host zu hoch
Auslöserausdruck: last(/Myhost/system.cpu.load[percpu,avg1])>5 or last(/Myotherhost/system.cpu.load[percpu

1. Datenpunktwert auf Myhost: 0.83 (Prozessorlast (1-Minuten-Durchschnitt pro Kern))
2. Datenpunktwert auf Myotherhost: 5.125 (Prozessorlast (1-Minuten-Durchschnitt pro Kern))
```

Beispiel 6

Empfangen von Details sowohl zum Problemereignis als auch zum Wiederherstellungsereignis in einer Wiederherstellungs-Nachricht:

Nachricht:

```
Problem:

Ereignis-ID: {EVENT.ID}
Ereigniswert: {EVENT.VALUE}
Ereignisstatus: {EVENT.STATUS}
Ereigniszeit: {EVENT.TIME}
Ereignisdatum: {EVENT.DATE}
Ereignisalter: {EVENT.AGE}
Ereignisbestätigung: {EVENT.ACK.STATUS}
Ereignisaktualisierungsverlauf: {EVENT.UPDATE.HISTORY}

Wiederherstellung:

Ereignis-ID: {EVENT.RECOVERY.ID}
Ereigniswert: {EVENT.RECOVERY.VALUE}
Ereignisstatus: {EVENT.RECOVERY.STATUS}
Ereigniszeit: {EVENT.RECOVERY.TIME}
Ereignisdatum: {EVENT.RECOVERY.DATE}
Betriebsdaten: {EVENT.OPDATA}
```

Wenn Sie die Nachricht erhalten, werden die Makros durch etwa Folgendes ersetzt:

```
Problem:

Ereignis-ID: 21874
Ereigniswert: 1
Ereignisstatus: PROBLEM
Ereigniszeit: 13:04:30
Ereignisdatum: 2018.01.02
Ereignisalter: 5m 0s
Ereignisbestätigung: Ja
Ereignisaktualisierungsverlauf: 2018.01.02 13:05:51 "John Smith (Admin)"
Aktionen: bestätigt.
```

```
Wiederherstellung:

Ereignis-ID: 21896
Ereigniswert: 0
Ereignisstatus: OK
Ereigniszeit: 13:10:07
Ereignisdatum: 2018.01.02
Betriebsdaten: Aktueller Wert ist 0.83
```

Wiederherstellungsaktionen

Übersicht

Wiederherstellungsaktionen ermöglichen es Ihnen, benachrichtigt zu werden, wenn Probleme behoben wurden.

Sowohl Nachrichten als auch Remote-Befehle werden in Wiederherstellungsaktionen unterstützt. Es können zwar mehrere Aktionen hinzugefügt werden, eine Eskalation wird jedoch nicht unterstützt – alle Aktionen werden einem einzelnen Schritt zugewiesen und daher gleichzeitig ausgeführt.

Anwendungsfälle

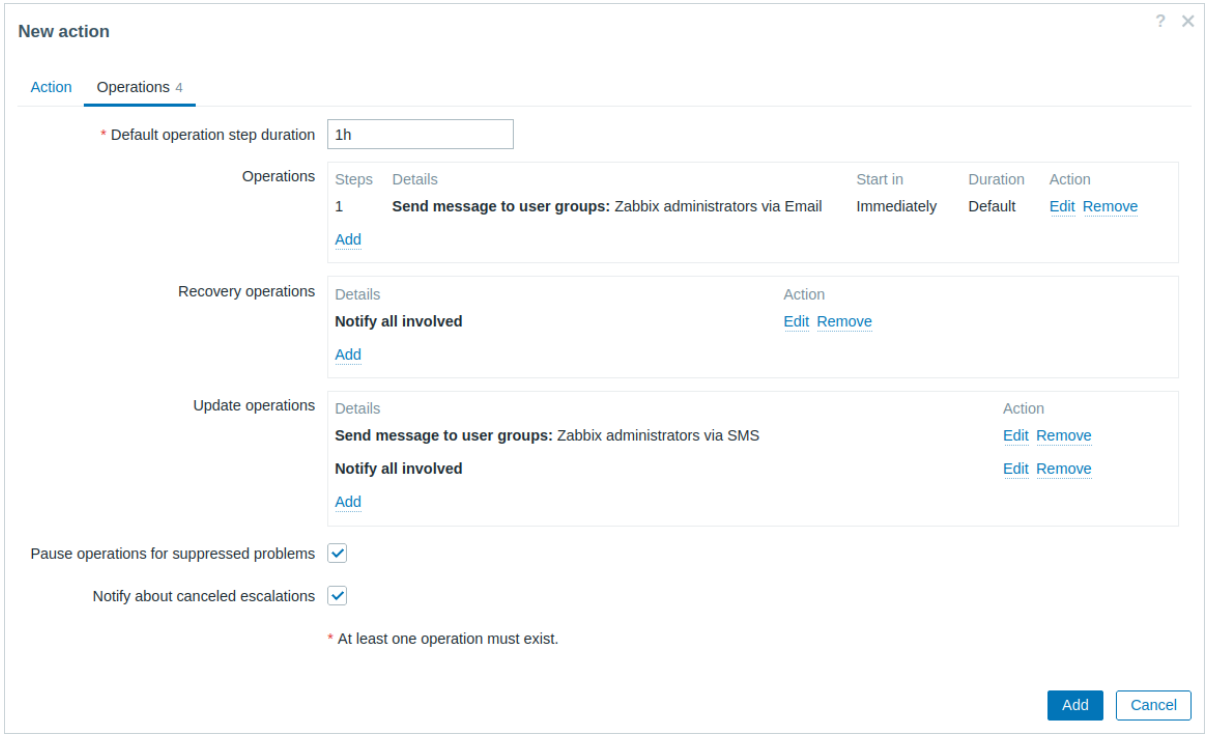
Einige Anwendungsfälle für Wiederherstellungsoperationen sind wie folgt:

1. Bei einer Wiederherstellung alle Benutzer benachrichtigen, die über das Problem benachrichtigt wurden:
    - Wählen Sie *Alle Beteiligten benachrichtigen* als Operationstyp aus.
2. Mehrere Operationen bei einer Wiederherstellung ausführen: eine Benachrichtigung senden und einen Remote-Befehl ausführen:
    - Fügen Sie Operationstypen zum Senden einer Nachricht und zum Ausführen eines Befehls hinzu.
3. Ein Ticket in einem externen Helpdesk-/Ticketsystem eröffnen und es schließen, wenn das Problem behoben ist:
    - Erstellen Sie ein externes Skript, das mit dem Helpdesk-System kommuniziert.
    - Erstellen Sie eine Aktion mit einer Operation, die dieses Skript ausführt und dadurch ein Ticket eröffnet.
    - Richten Sie eine Wiederherstellungsoperation ein, die dieses Skript mit anderen Parametern ausführt und das Ticket schließt.
    - Verwenden Sie das Makro {EVENT.ID}, um auf das ursprüngliche Problem zu verweisen.

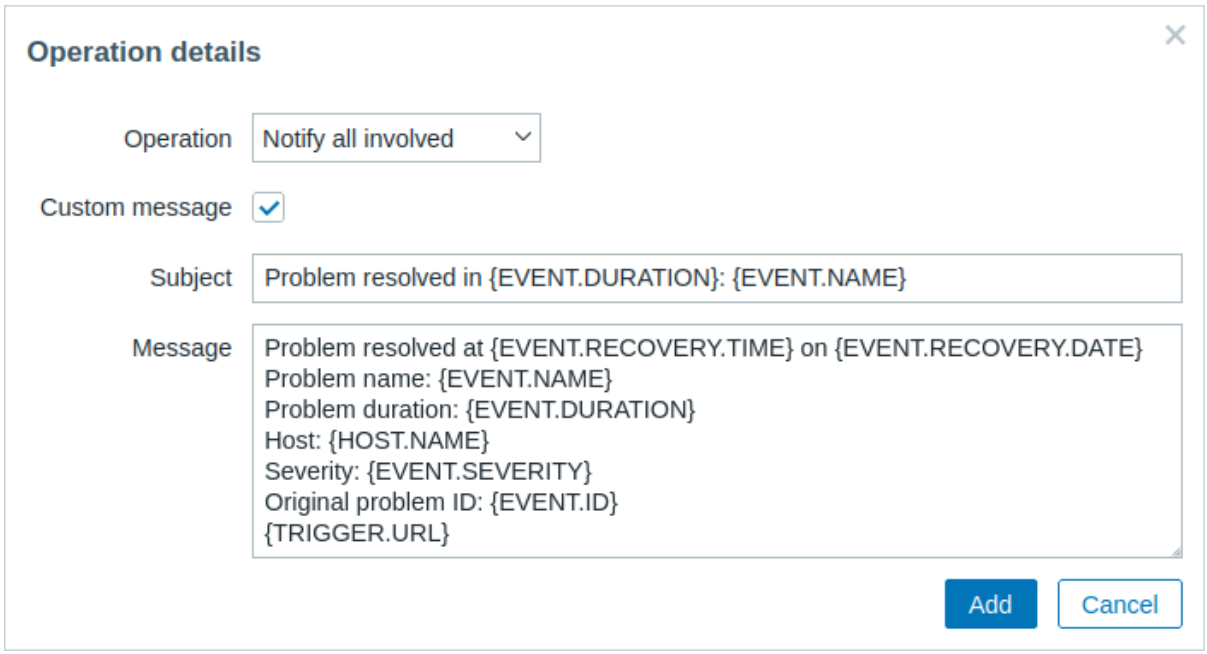
Konfigurieren einer Wiederherstellungsoperation

Gehen Sie wie folgt vor, um eine Wiederherstellungsoperation zu konfigurieren:

1. Wechseln Sie im Formular zur Aktionskonfiguration auf die Registerkarte *Operationen*.
2. Um eine neue Wiederherstellungsoperation zu konfigurieren, klicken Sie im Abschnitt *Wiederherstellungsoperationen* auf *Hinzufügen*. Um eine vorhandene Operation zu bearbeiten, klicken Sie neben der Operation auf *Bearbeiten*.
3. Konfigurieren Sie die Operationsdetails.



Details zu Wiederherstellungsoperationen



Für Wiederherstellungsereignisse sind drei Operationstypen verfügbar:

- **Nachricht senden** - eine Wiederherstellungsnachricht an den angegebenen Benutzer senden;
- **Alle Beteiligten benachrichtigen** - eine Wiederherstellungsnachricht an alle Benutzer senden, die über das Problemereignis benachrichtigt wurden;
- **<Name des Remote-Befehls>** - einen Remote-Befehl ausführen. Befehle stehen zur Ausführung zur Verfügung, wenn sie zuvor in globalen Skripten mit *Aktionsoperation* als Geltungsbereich definiert wurden.

Die Parameter für jeden Operationstyp werden unten beschrieben. Alle erforderlichen Eingabefelder sind mit einem roten Sternchen markiert. Klicken Sie anschließend auf *Hinzufügen*, um die Operation zur Liste der *Wiederherstellungsoperationen* hinzuzufügen.

Note:
Beachten Sie, dass keine doppelten Benachrichtigungen gesendet werden, wenn derselbe Empfänger in mehreren Operationstypen ohne angegebene *Benutzerdefinierte Nachricht* definiert ist.

Operationstyp: Nachricht senden

Parameter	Beschreibung
An Benutzergruppen senden	Wählen Sie Benutzergruppen aus, an die die Wiederherstellungsnachricht gesendet werden soll. Die Benutzergruppe muss mindestens über „Lesen“- Berechtigungen für den Host verfügen, um benachrichtigt zu werden.
An Benutzer senden	Wählen Sie Benutzer aus, an die die Wiederherstellungsnachricht gesendet werden soll. Der Benutzer muss mindestens über „Lesen“- Berechtigungen für den Host verfügen, um benachrichtigt zu werden.
An Medientyp senden	Senden Sie die Standard-Wiederherstellungsnachricht an alle verfügbaren (konfigurierten und aktivierten) Medientypen oder nur an einen bestimmten.
Benutzerdefinierte Nachricht	Falls ausgewählt, kann eine benutzerdefinierte Nachricht definiert werden.
Betreff	Betreff der benutzerdefinierten Nachricht. Der Betreff kann Makros enthalten.
Nachricht	Die benutzerdefinierte Nachricht. Die Nachricht kann Makros enthalten.

Operationstyp: Remote-Befehl

Parameter	Beschreibung
Zielliste	Wählen Sie die Ziele aus, auf denen der Befehl ausgeführt werden soll: **Aktueller Host** – Der Befehl wird auf dem Host des Auslösers ausgeführt, der das Problemereignis verursacht hat. Diese Option funktioniert nicht, wenn der Auslöser mehrere Hosts enthält. **Host** – Wählen Sie den oder die Hosts aus, auf denen der Befehl ausgeführt werden soll. **Hostgruppe** – Wählen Sie die Hostgruppe(n) aus, auf denen der Befehl ausgeführt werden soll. Durch die Angabe einer übergeordneten Hostgruppe werden implizit alle verschachtelten Hostgruppen ausgewählt. Daher wird der Remote-Befehl auch auf Hosts aus verschachtelten Gruppen ausgeführt. Ein Befehl auf einem Host wird nur einmal ausgeführt, selbst wenn der Host mehr als einmal übereinstimmt (z. B. über mehrere Hostgruppen, einzeln und über eine Hostgruppe). Die Zielliste ist bedeutungslos, wenn der Befehl auf dem Zabbix-Server ausgeführt wird. Wenn Sie in diesem Fall weitere Ziele auswählen, wird der Befehl lediglich mehrmals auf dem Server ausgeführt. Beachten Sie, dass bei globalen Skripten die Zielauswahl auch von der Einstellung *Hostgruppe* in der Konfiguration des globalen Skripts abhängt.

Operationstyp: Alle Beteiligten benachrichtigen

Parameter	Beschreibung
Benutzerdefinierte Nachricht	Wenn diese Option ausgewählt ist, kann eine benutzerdefinierte Nachricht definiert werden.
Betreff	Betreff der benutzerdefinierten Nachricht. Der Betreff kann Makros enthalten.
Nachricht	Die benutzerdefinierte Nachricht. Die Nachricht kann Makros enthalten.

Update-Operationen

Übersicht

Aktualisierungsoperationen sind in Aktionen mit den folgenden Ereignisquellen verfügbar:

- *Auslöser* - wenn Probleme von anderen Benutzern aktualisiert werden, d. h. kommentiert, bestätigt, in ihrer Schwere geändert oder (manuell) geschlossen werden;
- *Services* - wenn sich die Schwere eines Services geändert hat, der Service jedoch noch nicht wiederhergestellt ist.

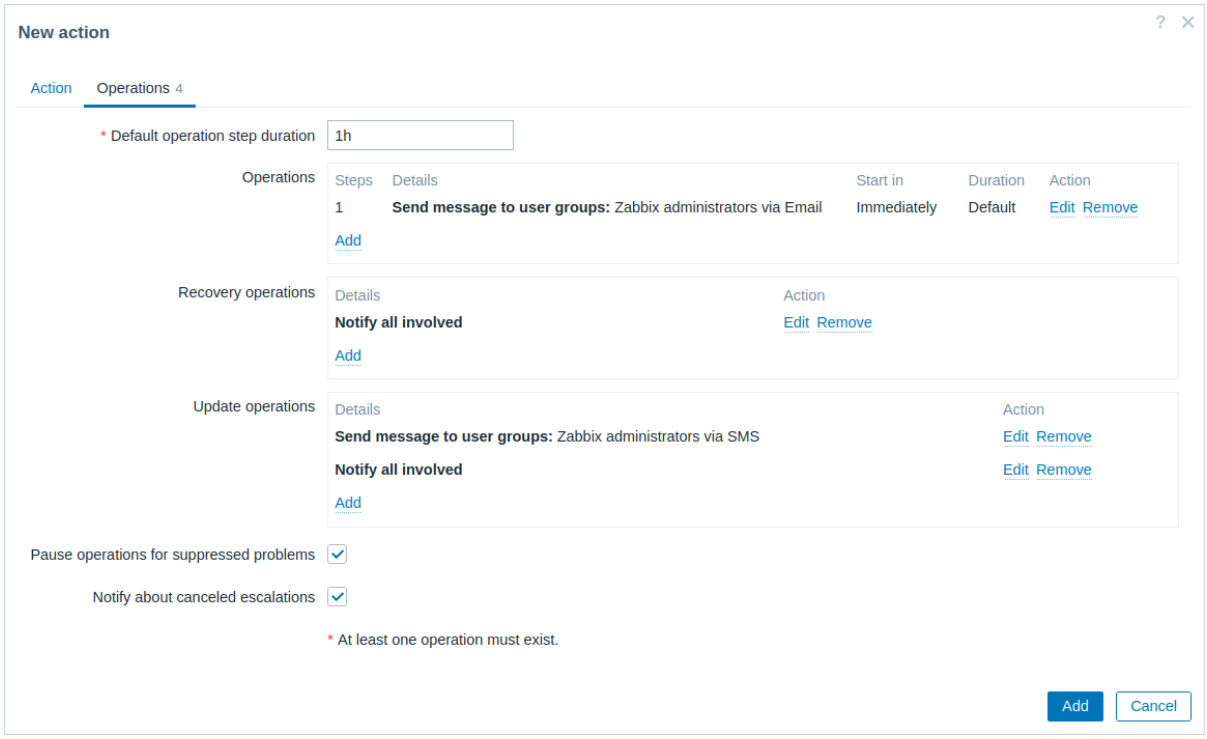
Bitte beachten Sie, dass Benutzer keine Benachrichtigungen über ihre eigenen Aktualisierungen erhalten.

Sowohl Nachrichten als auch Remote-Befehle werden in Aktualisierungsoperationen unterstützt. Obwohl mehrere Operationen hinzugefügt werden können, wird keine Eskalation unterstützt - alle Operationen werden einem einzelnen Schritt zugewiesen und daher gleichzeitig ausgeführt.

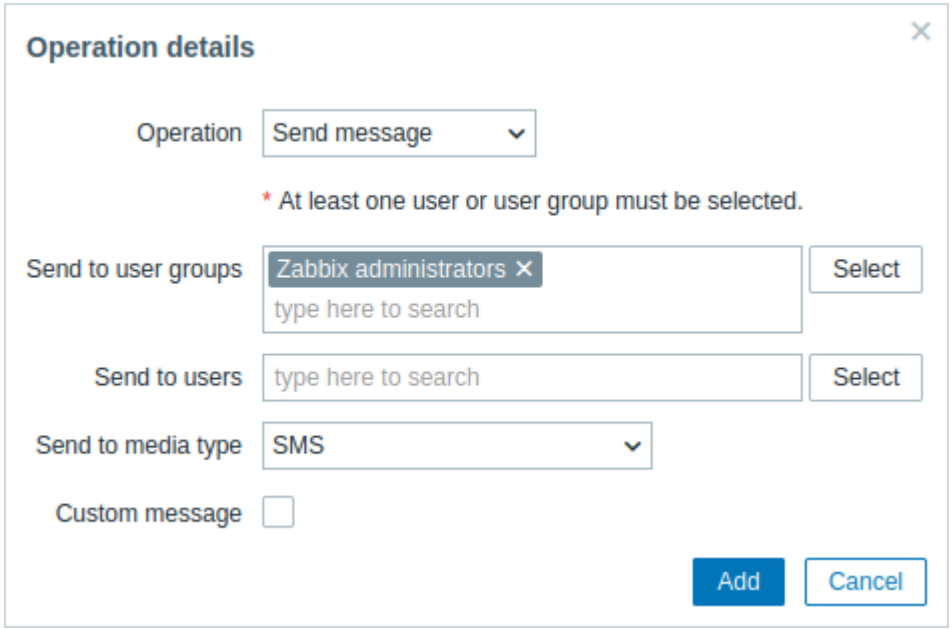
Konfigurieren einer Aktualisierungsoperation

Um eine Aktualisierungsoperation zu konfigurieren, gehen Sie wie folgt vor:

1. Wechseln Sie im Formular zur Aktionskonfiguration auf die Registerkarte *Operations*.
2. Um eine neue Aktualisierungsoperation zu konfigurieren, klicken Sie im Abschnitt *Update operations* auf *Add*. Um eine vorhandene Operation zu bearbeiten, klicken Sie neben der Operation auf *Edit*.
3. Konfigurieren Sie die Operationsdetails.



Details zur Aktualisierungsoperation



Aktualisierungsoperationen bieten denselben Parametersatz wie Wiederherstellungsoperationen.

Eskalationen

Übersicht

Mit Eskalationen können Sie benutzerdefinierte Szenarien zum Senden von Benachrichtigungen oder zum Ausführen von Remote-Befehlen erstellen.

In der Praxis bedeutet das:

- Benutzer können sofort über neue Probleme informiert werden.
- Benachrichtigungen können wiederholt werden, bis das Problem behoben ist.
- Das Senden einer Benachrichtigung kann verzögert werden.
- Benachrichtigungen können an eine andere „höhere“ Benutzergruppe eskaliert werden.
- Remote-Befehle können sofort ausgeführt werden oder wenn ein Problem über einen längeren Zeitraum nicht behoben wird.

Aktionen werden basierend auf dem **Eskalationsschritt** eskaliert. Jeder Schritt hat eine zeitliche Dauer.

Sie können sowohl die Standarddauer als auch eine benutzerdefinierte Dauer eines einzelnen Schritts festlegen. Die Mindestdauer eines Eskalationsschritts beträgt 60 Sekunden.

Sie können Aktionen wie das Senden von Benachrichtigungen oder das Ausführen von Befehlen von jedem Schritt aus starten. Schritt eins ist für sofortige Aktionen vorgesehen. Wenn Sie eine Aktion verzögern möchten, können Sie sie einem späteren Schritt zuweisen. Für jeden Schritt können mehrere Aktionen definiert werden.

Die Anzahl der Eskalationsschritte ist nicht begrenzt.

Eskalationen werden beim Konfigurieren einer Operation definiert. Eskalationen werden nur für Problemoperationen unterstützt, nicht für Wiederherstellungen.

Verschiedene Aspekte des Eskalationsverhaltens

Sehen wir uns an, was unter verschiedenen Umständen geschieht, wenn eine Aktion mehrere Eskalationsschritte enthält.

Situation	Verhalten
Der betreffende Host wechselt in den Wartungsmodus, nachdem die erste Problembenachrichtigung gesendet wurde	Abhängig von der Einstellung *Operationen für unterdrückte Probleme pausieren* in der Aktionskonfiguration werden alle verbleibenden Eskalationsschritte entweder mit einer durch den Wartungszeitraum verursachten Verzögerung oder ohne Verzögerung ausgeführt. Ein Wartungszeitraum bricht Operationen nicht ab.
*Der im Aktionszustand **Zeitraum** definierte Zeitraum endet, nachdem die erste Benachrichtigung gesendet wurde*	Alle verbleibenden Eskalationsschritte werden ausgeführt. Die Bedingung *Zeitraum* kann Operationen nicht stoppen; sie wirkt sich darauf aus, wann Aktionen gestartet bzw. nicht gestartet werden, nicht auf Operationen.
Ein Problem beginnt während des Wartungszeitraums und besteht nach dessen Ende weiterhin (wurde nicht behoben)	Abhängig von der Einstellung *Operationen für unterdrückte Probleme pausieren* in der Aktionskonfiguration werden alle Eskalationsschritte entweder ab dem Ende des Wartungszeitraums oder sofort ausgeführt.
Ein Problem beginnt während eines Wartungszeitraums ohne Daten und besteht nach dessen Ende weiterhin (wurde nicht behoben)	Es muss gewartet werden, bis der Auslöser ausgelöst wird, bevor alle Eskalationsschritte ausgeführt werden.
Verschiedene Eskalationen folgen in kurzer Folge aufeinander und überschneiden sich	Die Ausführung jeder neuen Eskalation setzt die vorherige Eskalation außer Kraft, mit Ausnahme von mindestens einem Eskalationsschritt, der bei der vorherigen Eskalation immer ausgeführt wird. Dieses Verhalten ist bei Aktionen für Ereignisse relevant, die bei JEDER Problembewertung des Auslösers erstellt werden.

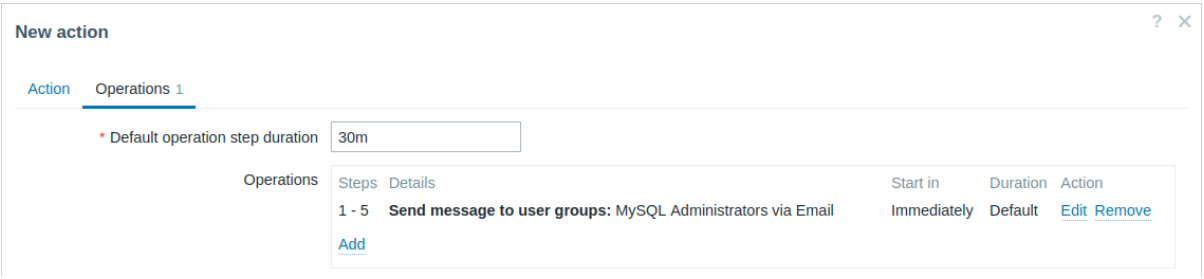
Situation	Verhalten
Während einer laufenden Eskalation (z. B. während eine Nachricht gesendet wird), basierend auf einem beliebigen Ereignistyp: - die Aktion wird deaktiviert Basierend auf einem Auslöserereignis: - der Auslöser wird deaktiviert - der Host oder Datenpunkt wird deaktiviert Basierend auf einem internen Ereignis zu Auslösern: - der Auslöser wird deaktiviert Basierend auf einem internen Ereignis zu Datenpunkten/Regeln für die Low-Level-Discovery: - der Datenpunkt wird deaktiviert - der Host wird deaktiviert	Die laufende Nachricht wird gesendet, anschließend wird noch eine weitere Nachricht zur Eskalation gesendet. Die Folgemeldung enthält am Anfang des Nachrichtentexts den Abbruchtext (*HINWEIS: Eskalation abgebrochen*) mit Angabe des Grundes (z. B. *HINWEIS: Eskalation abgebrochen: Aktion „<Aktionsname>“ deaktiviert*). Auf diese Weise wird der Empfänger darüber informiert, dass die Eskalation abgebrochen wurde und keine weiteren Schritte ausgeführt werden. Diese Nachricht wird an alle gesendet, die zuvor Benachrichtigungen erhalten haben. Der Abbruchgrund wird außerdem in die Server-Protokolldatei geschrieben (ab Debug-Level 3=Warnung). Beachten Sie, dass die Nachricht *Eskalation abgebrochen* auch gesendet wird, wenn die Operationen abgeschlossen sind, aber Wiederherstellungsoperationen konfiguriert und noch nicht ausgeführt wurden.
Während einer laufenden Eskalation (z. B. während eine Nachricht gesendet wird) wird die Aktion gelöscht	Es werden keine weiteren Nachrichten gesendet. Die Information wird in die Server-Protokolldatei geschrieben (ab Debug-Level 3=Warnung), zum Beispiel: `escalation canceled: action id:334 deleted`

Eskalationsbeispiele

Beispiel 1

Senden einer wiederholten Benachrichtigung einmal alle 30 Minuten (insgesamt 5-mal) an eine Gruppe „MySQL-Administratoren“. So konfigurieren Sie dies:

- Legen Sie auf der Registerkarte *Operations* die *Default operation step duration* auf „30m“ (30 Minuten) fest.
- Setzen Sie die Eskalations-*Steps* von „1“ bis „5“.
- Wählen Sie die Gruppe „MySQL-Administratoren“ als Empfänger der Nachricht aus.



Benachrichtigungen werden 0:00, 0:30, 1:00, 1:30 und 2:00 Stunden nach Beginn des Problems gesendet (es sei denn, das Problem wird natürlich früher behoben).

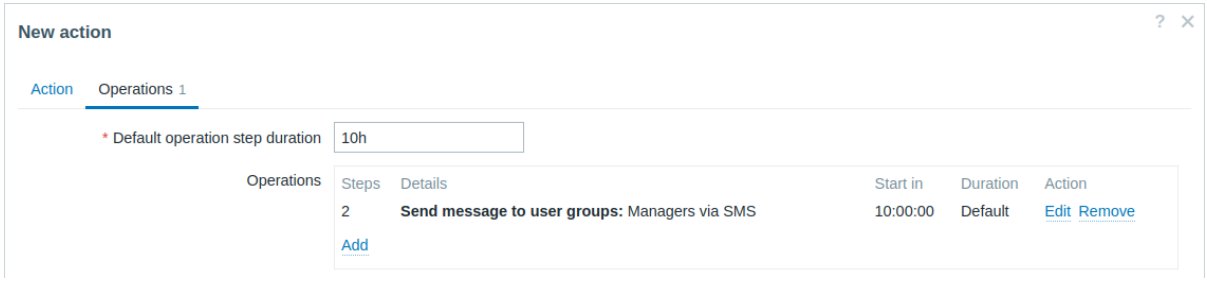
Wenn das Problem behoben wird und eine Wiederherstellungsnachricht konfiguriert ist, wird sie an diejenigen gesendet, die innerhalb dieses Eskalationsszenarios mindestens eine Problembenachrichtigung erhalten haben.

Note:
Wenn der Auslöser, der eine aktive Eskalation erzeugt hat, deaktiviert wird, sendet Zabbix eine entsprechende Informationsnachricht an alle, die bereits Benachrichtigungen erhalten haben.

Beispiel 2

Senden einer verzögerten Benachrichtigung über ein seit Langem bestehendes Problem. Zur Konfiguration:

- Setzen Sie im Reiter *Operations* die *Default operation step duration* auf „10h“ (10 Stunden).
- Setzen Sie die Eskalations-*Steps* von „2“ bis „2“.



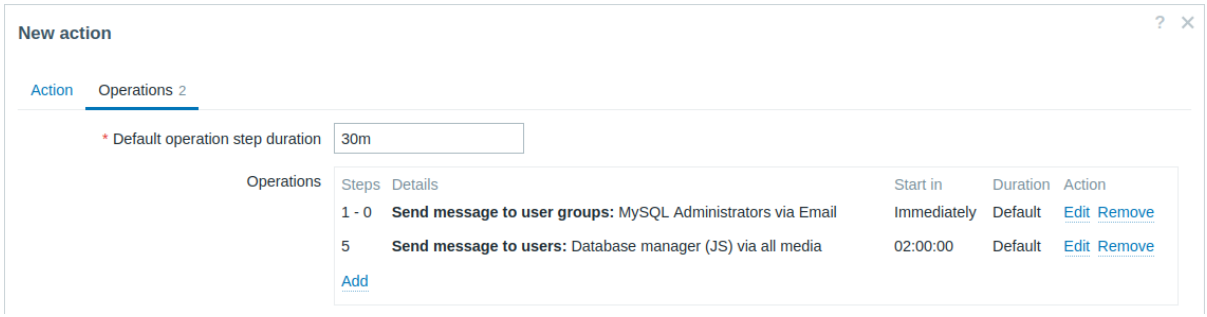
Eine Benachrichtigung wird nur in Schritt 2 des Eskalationsszenarios gesendet, also 10 Stunden nach Beginn des Problems.

Sie können den Nachrichtentext beispielsweise in „Das Problem besteht seit mehr als 10 Stunden“ ändern.

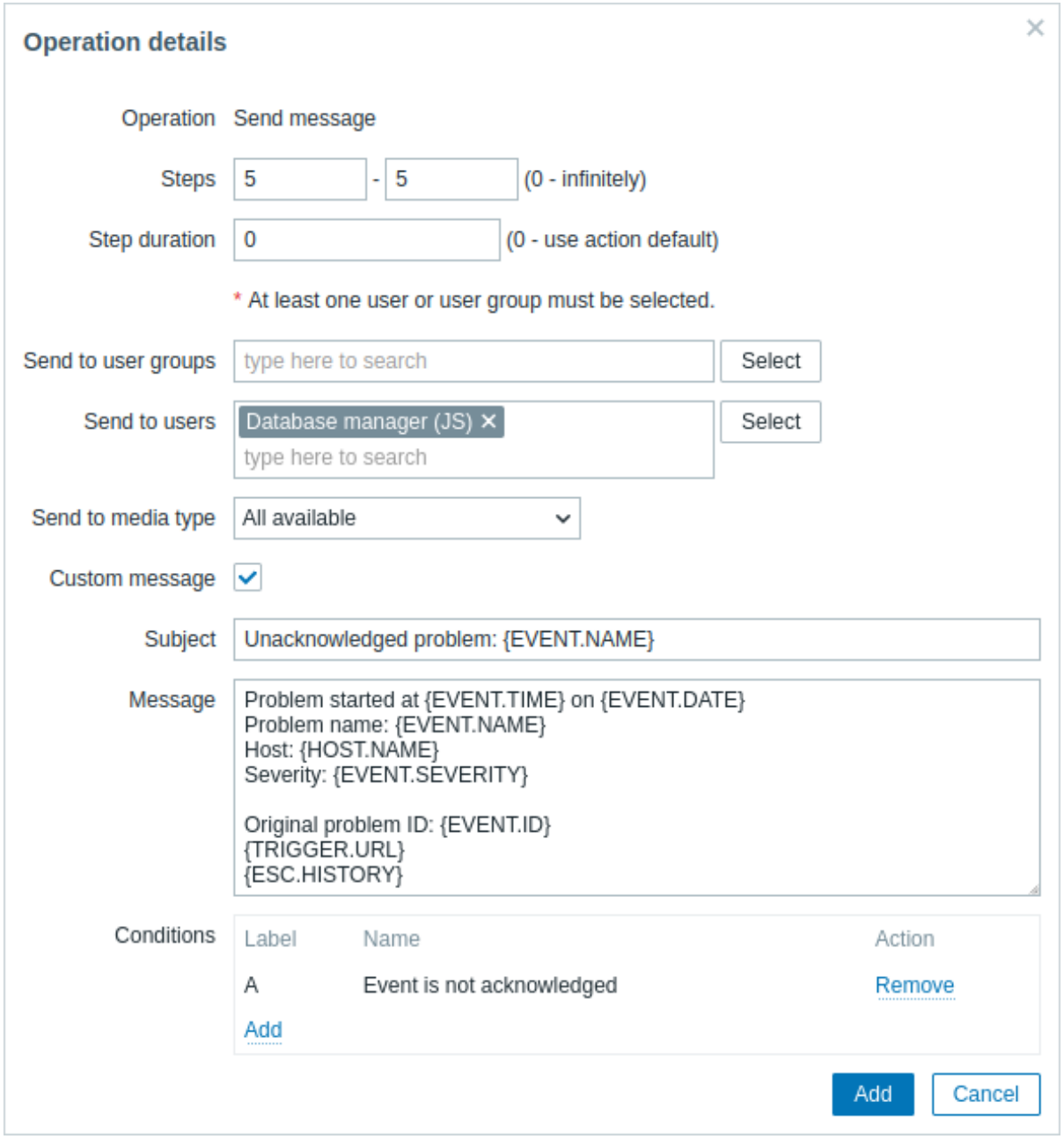
Beispiel 3

Eskalation des Problems an den Vorgesetzten.

Im ersten obigen Beispiel haben wir den periodischen Versand von Nachrichten an MySQL-Administratoren konfiguriert. In diesem Fall erhalten die Administratoren vier Nachrichten, bevor das Problem an den Datenbankmanager eskaliert wird. Beachten Sie, dass der Manager nur dann eine Nachricht erhält, wenn das Problem noch nicht bestätigt wurde, also vermutlich noch niemand daran arbeitet.



Details zu Operation 2:



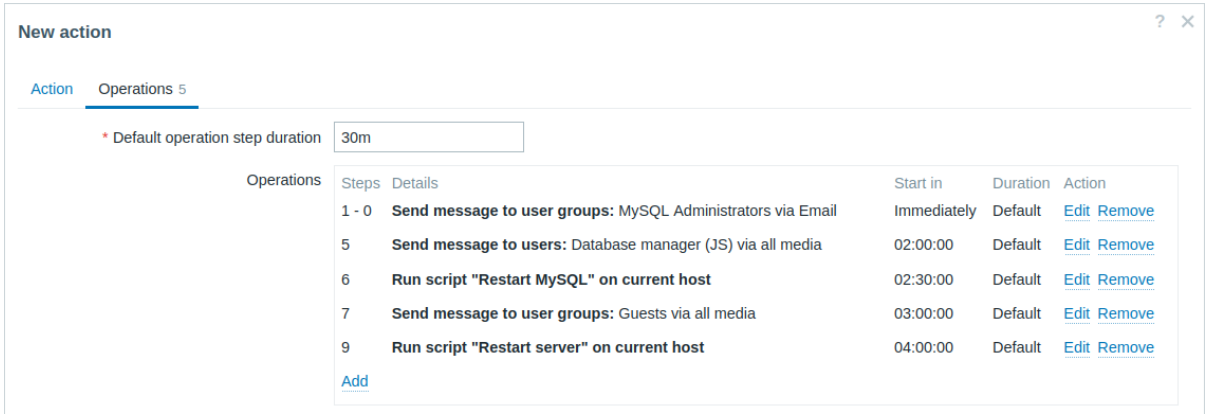
Beachten Sie die Verwendung des Makros {ESC.HISTORY} in der angepassten Nachricht. Das Makro enthält Informationen über alle zuvor ausgeführten Schritte dieser Eskalation, wie gesendete Benachrichtigungen und ausgeführte Befehle.

Beispiel 4

Ein komplexeres Szenario. Nach mehreren Nachrichten an die MySQL-Administratoren und einer Eskalation an den Manager wird Zabbix versuchen, die MySQL-Datenbank neu zu starten. Dies geschieht, wenn das Problem seit 2:30 Stunden besteht und nicht bestätigt wurde.

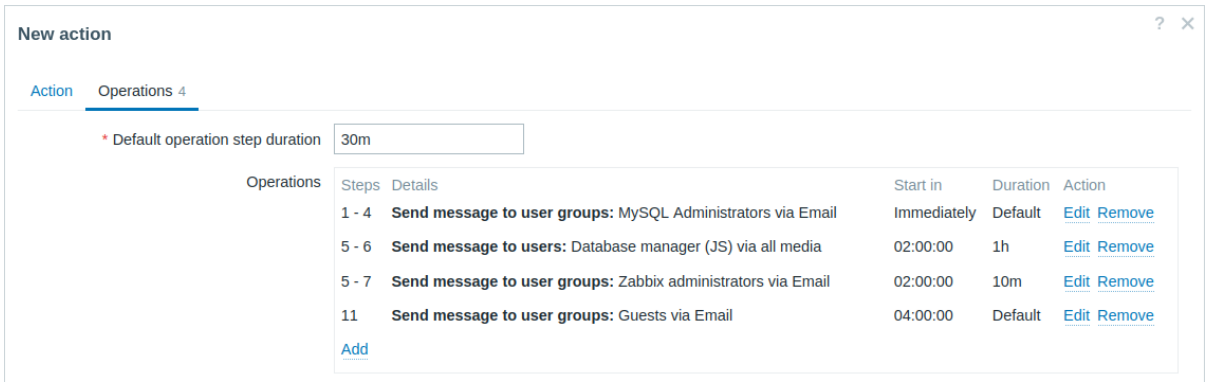
Wenn das Problem weiterhin besteht, sendet Zabbix nach weiteren 30 Minuten eine Nachricht an alle Gastbenutzer.

Wenn dies nicht hilft, startet Zabbix nach einer weiteren Stunde den Server mit der MySQL-Datenbank (zweiter Remote-Befehl) mithilfe von IPMI-Befehlen neu.



Beispiel 5

Eine Eskalation mit mehreren Operationen, die sich überschneidende Schrittbereiche und benutzerdefinierte Intervalle haben. Die standardmäßige Operations-Schrittdauer beträgt 30 Minuten.



Benachrichtigungen werden wie folgt gesendet:

- An MySQL-Administratoren um 0:00, 0:30, 1:00 und 1:30 nach Beginn des Problems.
- An den Datenbankmanager um 2:00 und 2:10 (die kürzere benutzerdefinierte Schrittdauer von 10 Minuten, die in der nachfolgenden Operation definiert ist, überschreibt die hier konfigurierte längere Schrittdauer von 1 Stunde, wie in Operationsdetails für *Schrittdauer* beschrieben, wenn sich Schritte überschneiden).
- An Zabbix-Administratoren um 2:00, 2:10 und 2:20 nach Beginn des Problems (die benutzerdefinierte Schrittdauer von 10 Minuten wird angewendet).
- An Gastbenutzer um 4:00 nach Beginn des Problems (die standardmäßige Schrittdauer von 30 Minuten gilt zwischen den Schritten 8 und 11).

Benachrichtigung bei nicht unterstützten Datenpunkten erhalten

Übersicht

Es ist möglich, Benachrichtigungen zu nicht unterstützten Datenpunkten in Zabbix zu erhalten.

Dies ist Teil des Konzepts interner Ereignisse in Zabbix und ermöglicht es Benutzern, über solche Vorkommnisse benachrichtigt zu werden. Interne Ereignisse spiegeln eine Zustandsänderung wider:

- wenn Datenpunkte von „normal“ zu „nicht unterstützt“ wechseln (und zurück);
- wenn Auslöser von „normal“ zu „unknown“ wechseln (und zurück);
- wenn Low-Level-Discovery-Regeln von „normal“ zu „nicht unterstützt“ wechseln (und zurück).

In diesem Abschnitt wird beschrieben, wie Sie eine **Benachrichtigung erhalten**, wenn ein Datenpunkt nicht unterstützt wird.

Konfiguration

Insgesamt sollte sich der Prozess zum Einrichten der Benachrichtigung für diejenigen vertraut anfühlen, die zuvor bereits Warnmeldungen in Zabbix eingerichtet haben.

Schritt 1

Konfigurieren Sie ein Medium, z. B. E-Mail, SMS oder ein Skript, das für die Benachrichtigungen verwendet werden soll. Informationen zur Durchführung dieser Aufgabe finden Sie in den entsprechenden Abschnitten des Handbuchs.

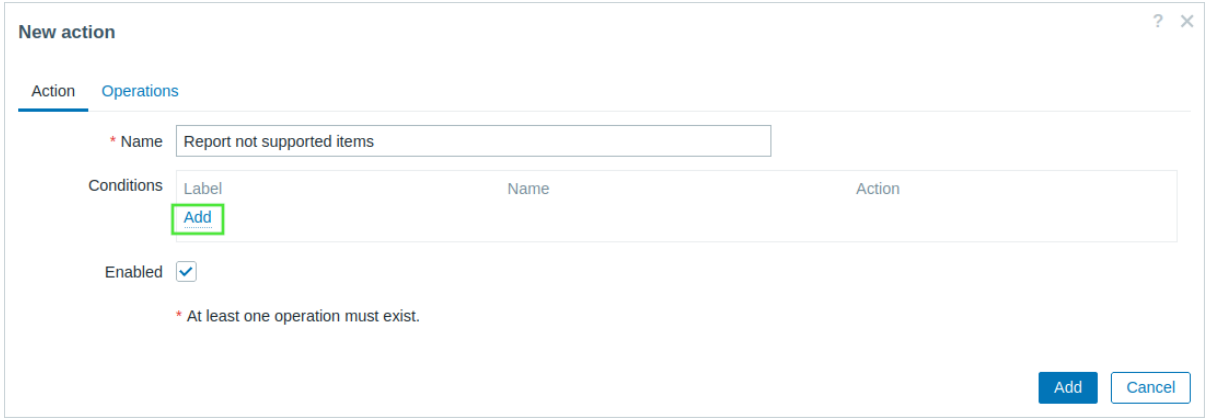
Attention:
Für Benachrichtigungen über interne Ereignisse wird der Standard- Schweregrad („Nicht klassifiziert“) verwendet. Lassen Sie ihn daher beim Konfigurieren von Benutzermedien aktiviert, wenn Sie Benachrichtigungen für interne Ereignisse erhalten möchten.

Schritt 2

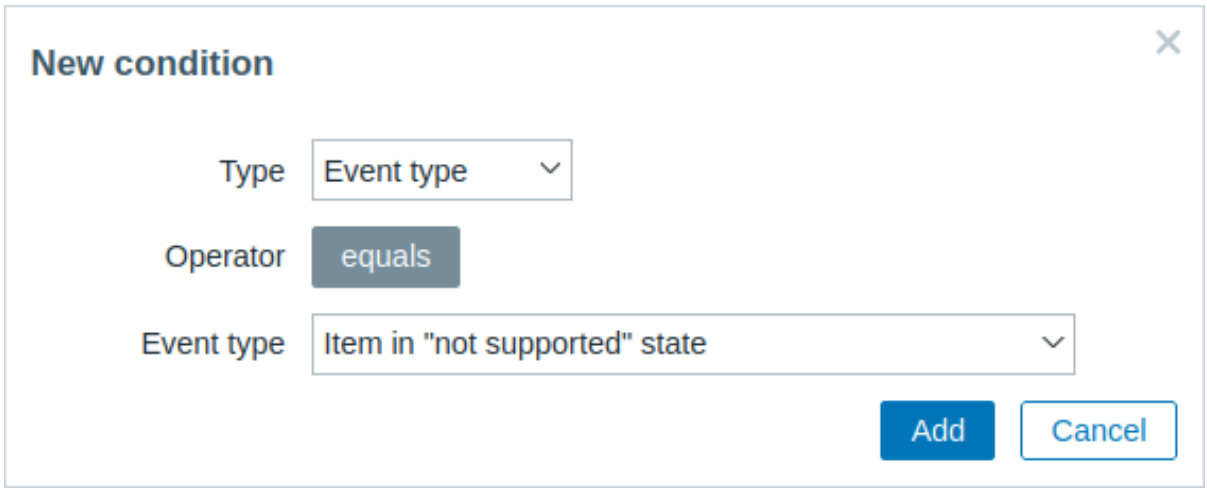
1. Gehen Sie zu *Alerts* > *Actions* > *Internal actions*.
2. Klicken Sie oben rechts auf der Seite auf *Create action*, um ein Formular zur Konfiguration einer Aktion zu öffnen.

Schritt 3

1. Geben Sie auf der Registerkarte *Action* einen Namen für die Aktion ein.
2. Klicken Sie im Block *Conditions* auf *Add*, um eine neue Bedingung hinzuzufügen.



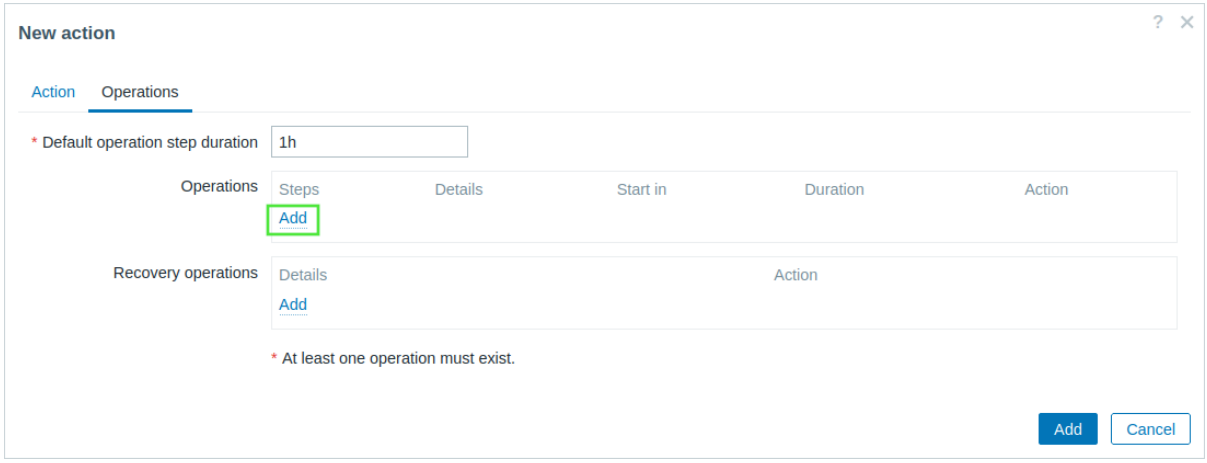
3. Wählen Sie im Pop-up-Fenster *New condition* als Bedingungstyp „Event type“ und anschließend als Ereignistyp „Item in ’not supported’ state“ aus.



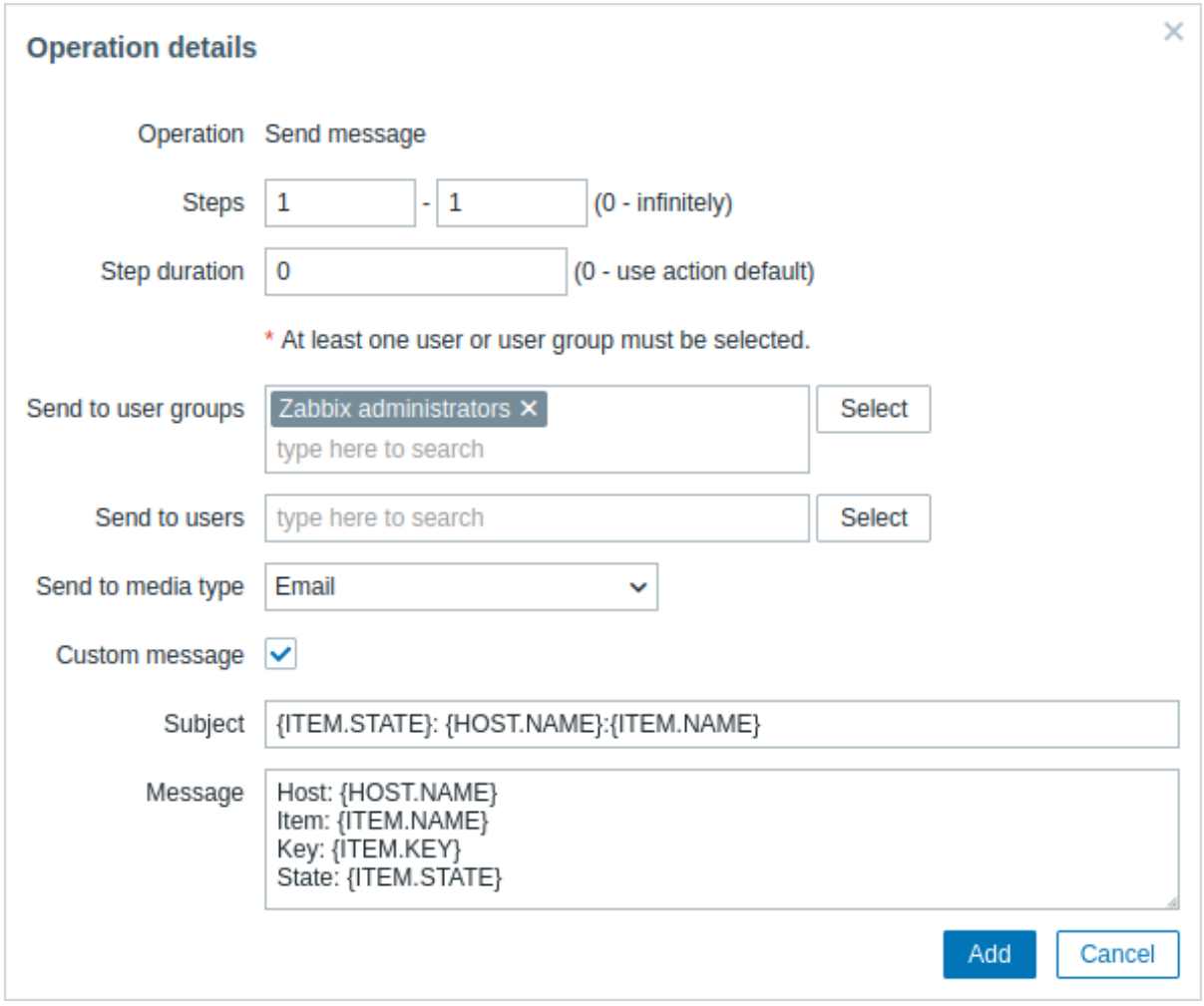
4. Klicken Sie auf *Add*, um die Bedingung tatsächlich im Block *Conditions* aufzulisten.

Schritt 4

1. Klicken Sie auf der Registerkarte *Operations* im Block *Operations* auf *Add*, um eine neue Operation hinzuzufügen.



2. Wählen Sie einige Empfänger der Nachricht (Benutzergruppen/Benutzer) und den für die Zustellung zu verwendenden Medientyp (oder „All“) aus.
3. Aktivieren Sie das Kontrollkästchen *Custom message*, wenn Sie den benutzerdefinierten Betreff/Inhalt der Problemmeldung eingeben möchten.



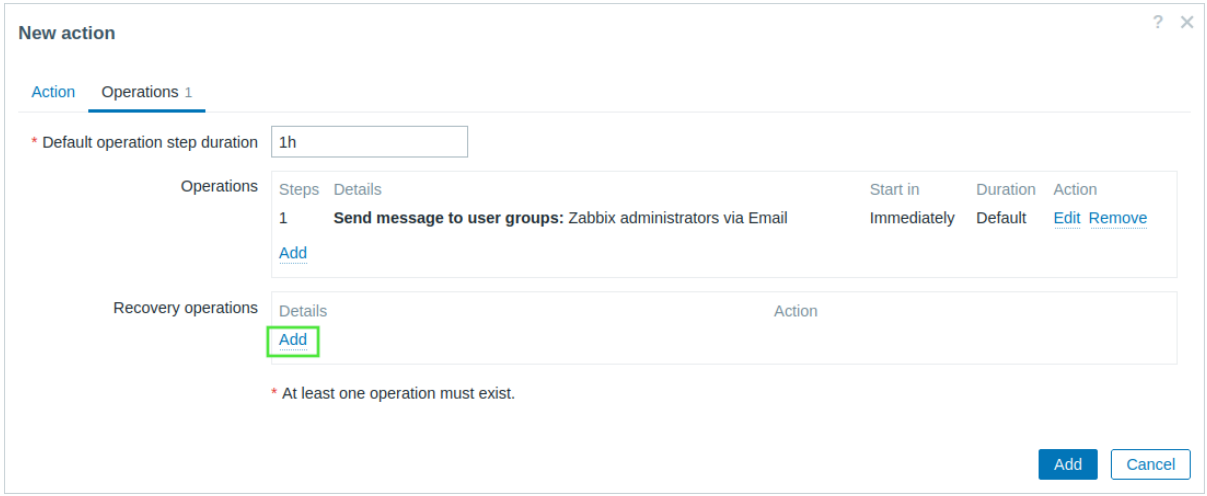
4. Klicken Sie auf *Add*, um die Operation tatsächlich im Block *Operations* aufzulisten.

Wenn Sie mehr als eine Benachrichtigung erhalten möchten, legen Sie die Dauer des Operationsschritts (Intervall zwischen den gesendeten Nachrichten) fest und fügen Sie einen weiteren Schritt hinzu.

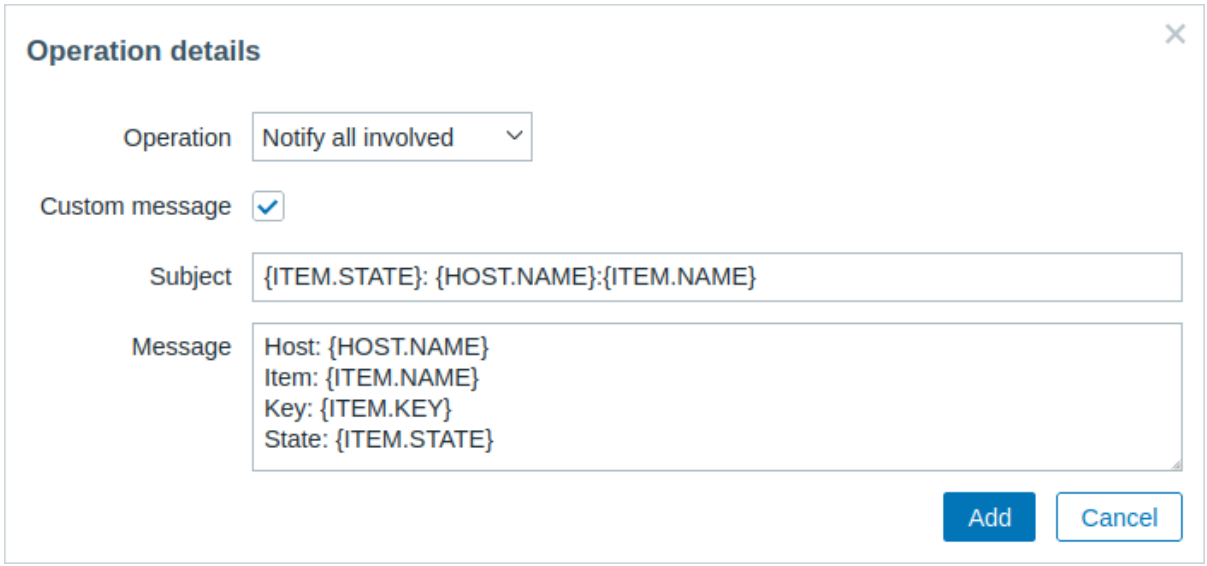
Schritt 5

Im Block *Wiederherstellungsoperationen* können Sie eine Wiederherstellungsbenachrichtigung konfigurieren, wenn ein Datenpunkt wieder in den Normalzustand zurückkehrt.

1. Klicken Sie im Block *Wiederherstellungsoperationen* auf *Hinzufügen*, um eine neue Wiederherstellungsoperation hinzuzufügen.



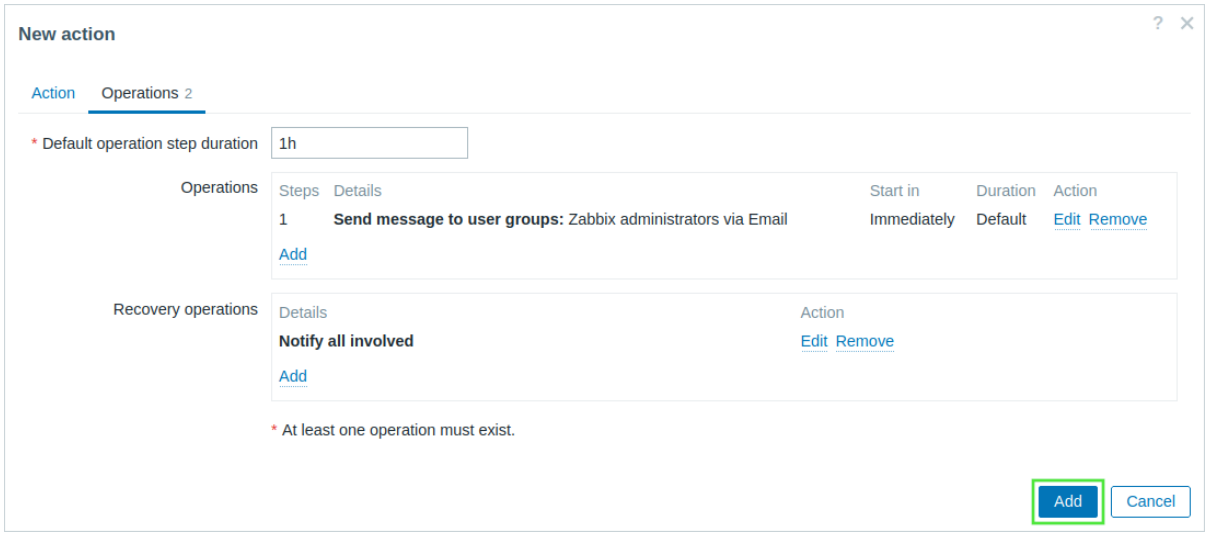
2. Wählen Sie den Operationstyp „Alle Beteiligten benachrichtigen“ aus.
3. Aktivieren Sie das Kontrollkästchen *Benutzerdefinierte Nachricht*, wenn Sie einen benutzerdefinierten Betreff bzw. Inhalt der Wiederherstellungsnachricht eingeben möchten.



4. Klicken Sie im Pop-up-Fenster *Operationsdetails* auf *Hinzufügen*, um die Operation tatsächlich im Block *Wiederherstellungsoperationen* aufzulisten.

Schritt 6

Wenn Sie fertig sind, klicken Sie unten im Formular auf die Schaltfläche *Add*.



Und das war’s, Sie sind fertig! Jetzt können Sie sich darauf freuen, Ihre erste Benachrichtigung von Zabbix zu erhalten, wenn ein Datenpunkt nicht mehr unterstützt wird.

Makros

Übersicht

Zabbix unterstützt eine Reihe integrierter Makros, die in verschiedenen Situationen verwendet werden können. Diese Makros sind Variablen, die durch eine bestimmte Syntax gekennzeichnet sind:

`{MACRO}`

Makros werden abhängig vom Kontext in einen bestimmten Wert aufgelöst.

Die effektive Verwendung von Makros ermöglicht es Ihnen, Zeit zu sparen und die Zabbix-Konfiguration transparenter zu gestalten.

In einem typischen Anwendungsfall kann ein Makro in einer Vorlage verwendet werden. So kann ein Auslöser in einer Vorlage den Namen „Prozessorauslastung auf {HOST.NAME} ist zu hoch“ tragen. Wenn die Vorlage auf den Host, beispielsweise den Zabbix-Server, angewendet wird, wird der Name bei der Anzeige des Auslösers im Bereich „Monitoring“ in „Prozessorauslastung auf Zabbix-Server ist zu hoch“ aufgelöst.

Makros können in Parametern von Datenpunktschlüsseln verwendet werden. Ein Makro kann nur für einen Teil des Parameters verwendet werden, zum Beispiel `item.key[server_{HOST.HOST}_local]`. Das Setzen des Parameters in doppelte Anführungszeichen ist nicht erforderlich, da Zabbix sich um eventuell vorhandene mehrdeutige Sonderzeichen kümmert, wenn das Makro aufgelöst wird.

In Zabbix gibt es weitere Makrotypen.

Zabbix unterstützt die folgenden Makros:

- `{MACRO}` – integriertes Makro (siehe vollständige Liste)
- `{<macro>.<func>(<params>)}` – Makro-Funktionen
- `{$MACRO}` – benutzerdefiniertes Makro, optional mit Kontext
- `{#MACRO}` – Makro für die Low-Level-Discovery
- `{?EXPRESSION}` – Ausdrucksmakro

Makrofunktionen

Übersicht

Makrofunktionen bieten die Möglichkeit, Makro-Werte anzupassen (zum Beispiel zu kürzen oder bestimmte Teilzeichenfolgen zu extrahieren), sodass sie einfacher zu verwenden sind.

Die Syntax einer Makrofunktion lautet:

```
{macro.func(params)}
```

wobei

- **macro** - das anzupassende Makro;
- **func** - die anzuwendende Funktion (siehe unterstützte Funktionen);
- **params** - eine durch Kommas getrennte Liste von Funktionsparametern, die **in doppelte Anführungszeichen** gesetzt werden müssen, wenn sie:
  - mit einem Leerzeichen oder doppelten Anführungszeichen beginnen;
  - schließende Klammern oder ein Komma enthalten.

Zum Beispiel:

```
{{TIME}.fmttime(format,time_shift)}
{{ITEM.VALUE}.regsub(pattern, output)}
{{$USERMACRO}.regsub(pattern, output)}
{{#LLDMACRO}.regsub(pattern, output)}
```

Makrofunktionen werden unterstützt für

- Integrierte Makros
- Benutzermakros
- Makros der Low-Level-Discovery
- Ausdrucksmakros

Makrofunktionen können an allen Stellen verwendet werden, die die aufgeführten Makros unterstützen. Dies gilt, sofern nicht ausdrücklich angegeben ist, dass nur ein Makro erwartet wird (zum Beispiel bei der Konfiguration von Host-Makros oder von Filtern für Low-Level-Discovery-Regeln).

Pro Makro wird nur eine einzelne Funktion unterstützt; mehrere Makrofunktionen in einer Kette werden nicht unterstützt.

Note:
Bitte sehen Sie sich Beispiele zum Escaping an, wenn Makrofunktionen in anderen Kontexten verwendet werden (Funktion, Datenpunkt-Schlüssel, anderes Makro usw.).

Unterstützte Funktionen

Die Funktionen werden ohne zusätzliche Informationen aufgelistet.
Klicken Sie auf die Funktion, um die vollständigen Details anzuzeigen.

Funktion	Beschreibung
btoa	Kodierung des Makrowerts in Base64.
fmtnum	Zahlenformatierung zur Steuerung der Anzahl der nach dem Dezimaltrennzeichen ausgegebenen Stellen.
fmttime	Zeitformatierung.
htmldecode	Dekodierung des Makrowerts aus der HTML-Kodierung.
htmlencode	Kodierung des Makrowerts in HTML.
iregsub	Extraktion eines Teilstrings anhand einer Übereinstimmung mit einem regulären Ausdruck (ohne Beachtung der Groß-/Kleinschreibung).
jsonpath	Gibt das JSONPath-Ergebnis zurück.
lowercase	Umwandlung der Zeichen des Makrowerts in Kleinbuchstaben.
regrepl	Ersetzung eines Zeichens oder Teilstrings im Makrowert.
regsub	Extraktion eines Teilstrings anhand einer Übereinstimmung mit einem regulären Ausdruck (unter Beachtung der Groß-/Kleinschreibung).
tr	Transliteration der Zeichen des Makrowerts.
uppercase	Umwandlung der Zeichen des Makrowerts in Großbuchstaben.
urldecode	Dekodierung des Makrowerts aus der URL-Kodierung.
urlencode	Kodierung des Makrowerts in URL-Kodierung.
xmlxpath	Gibt das XML-XPath-Ergebnis zurück.

Funktionsdetails

Optionale Funktionsparameter werden durch < > angegeben.

btoa

Kodierung eines Makrowerts in die Base64-Kodierung. Die Base64-Kodierung ist eine Methode zur Darstellung von Binärdaten als Text und ist nützlich für die Speicherung sowie die sichere Übertragung binärer Inhalte über textbasierte Protokolle.

Beispiel:

```
{{ITEM.VALUE}.btoa()} - kodiert einen Wert wie "zabbix" in Base64 zu "emFiYml4"
```

fmtnum(digits)

Zahlenformatierung zur Steuerung der Anzahl der nach dem Dezimalpunkt ausgegebenen Stellen.

Parameter:

- **digits** - die Anzahl der Stellen nach dem Dezimalpunkt. Gültiger Bereich: 0-20. Es werden keine nachgestellten Nullen erzeugt.

Beispiele:

```
{{ITEM.VALUE}.fmtnum(2)} - gibt "24.35" für einen empfangenen Wert von "24.3483523" zurück
{{ITEM.VALUE}.fmtnum(0)} - gibt "24" für einen empfangenen Wert von "24.3483523" zurück
```

fmttime(format,<time_shift>)

Zeitformatierung.
 Beachten Sie, dass diese Funktion mit Makros verwendet werden kann, die einen Wert in einem der folgenden Zeitformate auflösen:

- `hh:mm:ss`
- `yyyy-mm-ddThh:mm:ss[tz]` (ISO8601-Standard)

- UNIX-Zeitstempel

Parameter:

- **format** - obligatorische Formatzeichenfolge, kompatibel mit der Formatierung der Funktion `strftime`;
- **time_shift** (optional) - die Zeitverschiebung, die vor der Formatierung auf die Zeit angewendet wird; sollte mit `-<N><time_unit>` oder `+<N><time_unit>` beginnen, wobei:
  - `N` - die Anzahl der Zeit-Einheiten, die addiert oder subtrahiert werden;
  - `time_unit` - h (Stunde), d (Tag), w (Woche), M (Monat) oder y (Jahr).

Kommentare:

- Der Parameter `time_shift` unterstützt mehrstufige Zeitoperationen und kann `/<time_unit>` enthalten, um an den Anfang der Zeiteinheit zu springen (`/d` - Mitternacht, `/w` - 1. Tag der Woche (Montag), `/M` - 1. Tag des Monats usw.). Beispiele: `-1w` - genau 7 Tage zurück; `-1w/w` - Montag der vorherigen Woche; `-1w/w+1d` - Dienstag der vorherigen Woche.
- Zeitoperationen werden von links nach rechts ohne Prioritäten berechnet. Zum Beispiel wird `-1M/d+1h/w` als `((-1M/d)+1h)/w` interpretiert.

Beispiele:

```
{{TIMESTAMP}.fmttime(%B)} - gibt "October" aus einem empfangenen Wert von "1633098961" zurück
{{TIMESTAMP}.fmttime(%d %B,-1M/M)} - gibt "1 September" aus einem empfangenen Wert von "1633098961" zurück
{{EVENT.TIMESTAMP}.fmttime("%x %X")} - gibt die gebietsschemaabhängige Darstellung von Datum und Uhrzeit z
{{EVENT.TIME}.fmttime("%x %X")} - gibt die gebietsschemaabhängige Darstellung von Datum und Uhrzeit zurück
```

htmldecode

Dekodieren eines Makrowerts aus der HTML-Kodierung.

Die folgenden Zeichen werden unterstützt:

Wert	Dekodierter Wert
&	&
<	<
>	>
"	"
'	'
'	'

Beispiel:

```
{{ITEM.VALUE}.htmldecode()} - dekodiert einen Wert wie "&lt;" per HTML-Dekodierung zu "<"
```

htmlencode

Kodierung eines Makrowerts in HTML-Kodierung.

Die folgenden Zeichen werden unterstützt:

Wert	Kodierter Wert
&	&
<	<
>	>
"	"
'	'

Beispiel:

```
{{ITEM.VALUE}.htmlencode()} - kodiert ein Zeichen wie "<" in "&lt;" mit HTML
```

iregsub(pattern,output)

Extraktion von Teilzeichenfolgen durch eine Übereinstimmung mit einem regulären Ausdruck (Groß-/Kleinschreibung wird nicht beachtet).

Parameter:

- **pattern** - der abzugleichende reguläre Ausdruck;

- **output** - die Ausgabeoptionen. **\1 - \9** Platzhalter werden für Erfassungsgruppen unterstützt. **\0** gibt den übereinstimmenden Text zurück.

Kommentare:

- Wenn es keine Übereinstimmung für den regulären Ausdruck gibt, gibt die Funktion eine leere Zeichenfolge zurück.
- Wenn das Funktionsmuster ein fehlerhafter regulärer Ausdruck ist, wird das Makro zu 'UNKNOWN' ausgewertet (außer bei Low-Level-Discovery-Makros; in diesem Fall wird die Funktion ignoriert und das Makro bleibt ungelöst).
- Verweise auf nicht vorhandene Erfassungsgruppen in der Ersetzungszeichenfolge werden durch eine leere Zeichenfolge ersetzt.

Beispiel:

```
{{ITEM.VALUE}.iregsub("fail|error|fault|problem","ERROR")} - wird zu "ERROR" aufgelöst, wenn die Teilzeich
```

jsonpath(pattern,<default>)

Gibt das JSONPath-Ergebnis zurück.
 Unterstützte Werttypen: *String*, *Text*, *Log*.

Parameter:

- **pattern** – das erforderliche Muster;
- **default** – der optionale Fallback-Wert, wenn die JSONPath-Abfrage keine Daten zurückgibt. Beachten Sie, dass bei anderen Fehlern ein Fehler zurückgegeben wird.

Hinweise:

- Die Funktion schlägt mit einem Fehler fehl, wenn:
  - das JSON ungültig ist.
  - die JSONPath-Abfrage fehlschlägt.
  - das JSONPath-Ergebnis keine Daten enthält und der Parameter `default` nicht angegeben wurde.

Beispiel:

```
{{ITEM.VALUE}.jsonpath("$.columns.servicename", "NA")}
```

Kleinbuchstaben

Umwandlung aller Zeichen eines Makrowerts in Kleinbuchstaben.
Funktioniert mit Single-Byte-Zeichensätzen (wie ASCII) und unterstützt kein UTF-8.

Beispiel:

```
{{ITEM.VALUE}.lowercase()} - wandelt einen Wert wie "Zabbix SERVER" in "zabbix server" um (Kleinbuchstaben
```

regrepl(pattern,replacement,<pattern2>,<replacement2>,...)

Ersetzung von Zeichen/Teilzeichenfolgen im Makrowert.

Parameter:

- **pattern** - der reguläre Ausdruck, der abgeglichen werden soll;
- **replacement** - die Ersetzungszeichenfolge. **\1 - \9** Platzhalter werden in Ersetzungszeichenfolgen unterstützt, um Gruppen zu erfassen.

Kommentare:

- Alle Übereinstimmungen mit dem Muster werden ersetzt.
- Die Muster und Ersetzungen werden nacheinander verarbeitet, wobei jedes nachfolgende Paar entsprechend dem Ergebnis der vorherigen Ersetzung angewendet wird.
- Verweise auf nicht vorhandene Erfassungsgruppen in der Ersetzungszeichenfolge werden durch eine leere Zeichenfolge ersetzt.

Beispiele:

```
{{ITEM.VALUE}.regrepl("oldParam", "newParam")} - ersetzt "oldParam" durch "newParam"
{{ITEM.VALUE}.regrepl("([^a-z])","\\1")} - alle Nicht-Buchstaben-Zeichen werden mit einem Backslash maski
{$THRESHOLD:"{{#FSNAME}.regrepl(\"\\$\",\"\")}"} - entfernt einen abschließenden Backslash (z. B. um "C:\"
{{ITEM.VALUE}.regrepl("_v1\.0", "_v2.0", "\(final\)", "")} - ersetzt mehrere Teile im Datenpunktwert
```

regsub(pattern,output)

Teilzeichenfolgenextraktion durch eine Übereinstimmung mit einem regulären Ausdruck (Groß-/Kleinschreibung wird beachtet).

Parameter:

- **pattern** - der abzugleichende reguläre Ausdruck;
- **output** - die Ausgabeoptionen. **\1 - \9** Platzhalter werden für Erfassungsgruppen unterstützt. **\0** gibt den übereinstimmenden Text zurück.

Kommentare:

- Wenn es keine Übereinstimmung für den regulären Ausdruck gibt, gibt die Funktion eine leere Zeichenfolge zurück.
- Wenn das Funktionsmuster ein fehlerhafter regulärer Ausdruck ist, wird das Makro zu 'UNKNOWN' ausgewertet (außer bei Low-Level-Discovery-Makros; in diesem Fall wird die Funktion ignoriert und das Makro bleibt ungelöst).
- Verweise auf nicht vorhandene Erfassungsgruppen in der Ersetzungszeichenfolge werden durch eine leere Zeichenfolge ersetzt.

Beispiele:

```
{{ITEM.VALUE}.regsub("^([0-9]+)", Problem ID: \1)} - wird zu "Problem ID: 123" aufgelöst, wenn ein Wert wi
{{ITEM.VALUE}.regsub("fail|error|fault|problem","ERROR")} - wird zu "ERROR" aufgelöst, wenn die Teilzeiche
```

Siehe weitere Beispiele.

tr(characters,replacement)

Transliteration von Makrowert-Zeichen.

- **characters** - die Menge der zu ersetzenden Zeichen;
- **replacement** - die Menge der positionsbezogen entsprechenden Ersetzungszeichen.

Beispiele:

```
{{ITEM.VALUE}.tr(abc, xyz)} - ersetzt alle Vorkommen von "a" durch "x", "b" durch "y", "c" durch "z"
{{ITEM.VALUE}.tr(abc, xyzq)} - ersetzt alle Vorkommen von "a" durch "x", "b" durch "y", "c" durch "z" ("q"
{{ITEM.VALUE}.tr(abcde, xyz)} - ersetzt alle Vorkommen von "a" durch "x", "b" durch "y", "c" durch "z", "d
{{ITEM.VALUE}.tr("\\\'", "\/\"")} - ersetzt alle Vorkommen von Backslash durch Schrägstrich, einfache Anfü
{{ITEM.VALUE}.tr(A-Z,a-z)} - wandelt alle Buchstaben in Kleinbuchstaben um
{{ITEM.VALUE}.tr(0-9a-z,*)} - ersetzt alle Zahlen und Kleinbuchstaben durch "*"
{{ITEM.VALUE}.tr(0-9,ab)} - ersetzt alle Vorkommen von 0 durch "a" und alle Vorkommen von 1, 2, 3, 4, 5, 6
{{ITEM.VALUE}.tr(0-9abcA-L,*)} - ersetzt alle Zahlen, die Zeichen "abc" und den Bereich A-L durch "*"
{{ITEM.VALUE}.tr("\n","*")} - ersetzt alle Zeilenende-Vorkommen durch *
{{ITEM.VALUE}.tr("e", "\n")} - ersetzt alle "e" durch ein Zeilenende
```

Um literale Zeichen einzuschließen:

```
backslash - muss als \\ maskiert werden
einfaches Anführungszeichen - muss als \' maskiert werden
doppeltes Anführungszeichen - muss als \" maskiert werden
```

Unterstützte Escape-Sequenzen mit Backslash:

```
\\\\ => \\ - doppelter Backslash zu einfachem Backslash
\\a  => \a - Alarm
\\b  => \b - Rückschritt
\\f  => \f - Seitenvorschub
\\n  => \n - Zeilenumbruch
\\r  => \r - Wagenrücklauf
\\t  => \t - horizontaler Tabulator
\\v  => \v - vertikaler Tabulator
```

Großbuchstaben

Umwandlung aller Zeichen eines Makrowerts in Großbuchstaben.
Funktioniert mit Einzelbyte-Zeichensätzen (wie ASCII) und unterstützt kein UTF-8.

Beispiel:

```
{{ITEM.VALUE}.uppercase()} - wandelt einen Wert wie "Zabbix Server" in "ZABBIX SERVER" um (Großbuchstaben)
```

urldecode

Dekodiert einen Makrowert aus der URL-Kodierung.

Beispiel:

```
{{ITEM.VALUE}.urldecode()} - dekodiert einen Wert wie "%2F" per URL-Dekodierung zu "/"
```

urlencode

Kodierung eines Makrowerts in URL-Kodierung.

Beispiel:

```
{{ITEM.VALUE}.urlencode()} - kodiert ein Zeichen wie "/" per URL-Kodierung zu "%2F"
```

xmlxpath(pattern,<default>)

Gibt das Ergebnis des XML-XPath zurück.
 Unterstützte Werttypen: *String*, *Text*, *Log*.

Parameter:

- **pattern** – das erforderliche Muster;
- **default** – der optionale Fallback-Wert, wenn die XPath-Abfrage keine Daten zurückgibt. Beachten Sie, dass bei anderen Fehlern ein Fehler zurückgegeben wird.

Kommentare:

- Die Funktion schlägt mit einem Fehler fehl, wenn:
  - das XML ungültig ist.
  - die XPath-Abfrage fehlschlägt.
  - das XPath-Ergebnis keine Daten enthält und der Parameter `default` nicht angegeben ist.
  - Zabbix ohne XML-Unterstützung kompiliert wurde.

Beispiel:

```
{{ITEM.VALUE}.xmlxpath("/response/error/status","unknown")}
```

Zusätzliche Beispiele

Die folgende Tabelle zeigt weitere Beispiele für die Verwendung von Makrofunktionen.

`{#IFALIAS}` ist ein LLD-Makro und wird nur in Low-Level-Discovery-Kontexten definiert (Discovery-Regeln, Prototypen und die daraus erstellten Datenpunkte/Auslöser). Bei Verwendung außerhalb von LLD bleibt das Token unausgewertet.

Makrofunktion	Empfangener Wert	Ausgabe
`{{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*` (ungültiger regulärer Ausdruck)
`{{#IFALIAS}.regsub("(.*)_([0-9]+)", \1)}`	`customername_1`	`customername`
`{{#IFALIAS}.regsub("(.*)_([0-9]+)", \2)}`	`customername_1`	`1`
`{{#IFALIAS}.regsub("(.*)_([0-9]+", \1)}`	`customername_1`	`{{#IFALIAS}.regsub("(.*)_([0-9]+", \1)}` (ungültiger regulärer Ausdruck)
`{$MACRO:"{{#IFALIAS}.regsub(\"(.*)_([0-9]+\", \1)}"}`	`customername_1`	`{$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)}"}` (ungültiger regulärer Ausdruck)
`"{$MACRO:"\{{#IFALIAS}.regsub("(.*)_([0-9]+", \1)}\"}"`	`customername_1`	`"{$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)}\"}"` (ungültiger regulärer Ausdruck)

Vollständige Datenpunktwerte anzeigen

Lange Werte aufgelöster Makros {ITEM.VALUE} und {ITEM.LASTVALUE} für Text-/Log-Datenpunkte werden an einigen Stellen im Frontend auf 20 Zeichen gekürzt. Um die vollständigen Werte dieser Makros anzuzeigen, können Sie Makrofunktionen verwenden, z. B.:

```
{{ITEM.VALUE}.regsub("(.*)", \1)}
{{ITEM.LASTVALUE}.regsub("(.*)", \1)}
```

Siehe auch: Makrodetails zu {ITEM.VALUE} und {ITEM.LASTVALUE}.

Benutzermakros

Übersicht

Benutzermakros werden in Zabbix für mehr Flexibilität zusätzlich zu den standardmäßig unterstützten Makros unterstützt.

Benutzermakros können auf globaler Ebene sowie auf Vorlagen- und Host-Ebene definiert werden. Diese Makros haben eine spezielle Syntax:

`{$MACRO}`

Zabbix löst Makros entsprechend der folgenden Priorität auf:

1. Makros auf Host-Ebene (werden zuerst geprüft).
2. Makros, die für die Vorlagen der ersten Ebene des Hosts definiert sind (d. h. Vorlagen, die direkt mit dem Host verknüpft sind), sortiert nach Vorlagen-ID.
3. Makros, die für die Vorlagen der zweiten Ebene des Hosts definiert sind, sortiert nach Vorlagen-ID.
4. Makros, die für die Vorlagen der dritten Ebene des Hosts definiert sind, sortiert nach Vorlagen-ID usw.
5. Globale Makros (werden zuletzt geprüft).

Wenn ein Makro für einen Host nicht existiert, versucht Zabbix also, es in den Vorlagen des Hosts mit zunehmender Verschachtelungstiefe zu finden. Wenn es weiterhin nicht gefunden wird, wird ein globales Makro verwendet, sofern vorhanden.

Warning:
Wenn ein Makro mit demselben **Namen** in mehreren verknüpften Vorlagen derselben Ebene existiert, wird das Makro aus der Vorlage mit der niedrigsten ID verwendet. Daher stellt die Verwendung von Makros mit demselben Namen in mehreren Vorlagen ein Konfigurationsrisiko dar.

Wenn Zabbix ein Makro nicht finden kann, wird das Makro nicht aufgelöst.

Attention:
Makros (einschließlich Benutzermakros) werden im Bereich „Konfiguration“ (zum Beispiel in der Auslöserliste) absichtlich nicht aufgelöst, um komplexe Konfigurationen transparenter zu machen.

Benutzermakros können verwendet werden in:

- Datenpunktname
- Datenpunktschlüsselparameter
- Aktualisierungsintervallen und flexiblen Intervallen von Datenpunkten
- Name und Beschreibung des Auslösers
- Parametern und Konstanten des Auslöserausdrucks (siehe Beispiele)
- vielen anderen Bereichen – siehe die vollständige Liste

Häufige Anwendungsfälle von globalen Makros und Host-Makros

- Verwenden Sie ein globales Makro an mehreren Stellen; ändern Sie dann den Makrowert und wenden Sie Konfigurationsänderungen mit einem Klick auf alle Stellen an.
- Nutzen Sie Vorlagen mit Host-spezifischen Attributen: Passwörter, Portnummern, Dateinamen, reguläre Ausdrücke usw.

Note:
Es wird empfohlen, Host-Makros anstelle von globalen Makros zu verwenden, da das Hinzufügen, Aktualisieren oder Löschen globaler Makros eine inkrementelle Konfigurationsaktualisierung für alle Hosts erzwingt. Weitere Informationen finden Sie unter Passive and active agent checks.

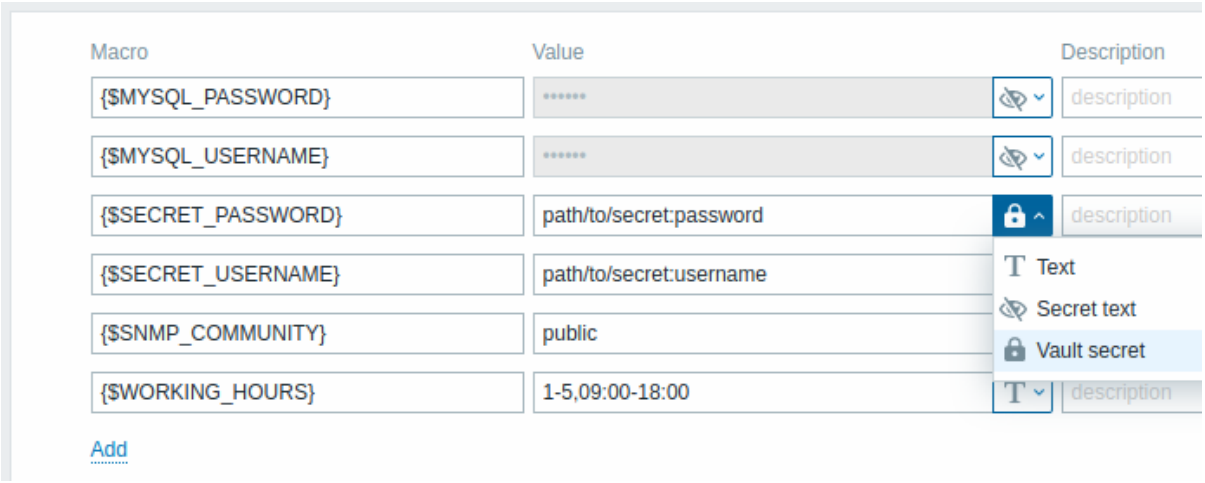
Konfiguration

Um Benutzer-Makros zu definieren, gehen Sie im Frontend zum entsprechenden Bereich:

- Für globale Makros öffnen Sie *Administration* > *Makros*.

- Für Makros auf Host- und Vorlagenebene öffnen Sie die Eigenschaften des Hosts oder der Vorlage und suchen Sie nach der Registerkarte *Makros*.

Ein Benutzer-Makro hat die folgenden Attribute:



Parameter	Beschreibung
Macro	Name des Makros. Der Name muss in geschweifte Klammern gesetzt werden und mit einem Dollarzeichen beginnen. Beispiel: `{$FRONTEND_URL}`. In Makronamen sind die folgenden Zeichen zulässig: **A-Z** (nur Großbuchstaben), **0-9**, _, **.**
Value	Makrowert. Es werden drei Werttypen unterstützt: **Text** (Standard) - Klartextwert **Secret text** - der Wert wird mit Sternchen maskiert **Vault secret** - der Wert enthält einen Pfad/eine Abfrage zu einem vault secret. Um den Werttyp zu ändern, klicken Sie auf die Schaltfläche am Ende des Eingabefelds für den Wert. Die maximale Länge eines Benutzer-Makrowerts beträgt 2048 Zeichen.
Description	Textfeld, das verwendet wird, um weitere Informationen zu diesem Makro bereitzustellen.

Beachten Sie bei der Konfiguration von Benutzer-Makros die folgenden kontextabhängigen Verhaltensweisen:

- wenn Benutzer-Makros in Vorlagen-Datenpunkten oder Auslösern verwendet werden, sollten Sie diese Makros ebenfalls zur Vorlage hinzufügen (auch wenn sie global definiert sind); auf diese Weise funktionieren Makros vom Typ *Text* nach dem Export der Vorlage nach XML und dem Import in ein anderes System wie erwartet (Werte von Secret-Makros werden nicht exportiert)
- wenn Benutzer-Makros in Auslöser-Ausdrücken verwendet werden, werden diese Makros nur aufgelöst, wenn sie auf einen Parameter oder eine Konstante verweisen; sie werden NICHT aufgelöst, wenn sie auf einen Host, einen Datenpunkt-Schlüssel, eine Funktion, einen Operator oder einen anderen Auslöser-Ausdruck verweisen (Secret-Makros können nicht in Auslöser-Ausdrücken verwendet werden)
- wenn Benutzer-Makros auf einem Host verwendet werden, der über eine Low-Level-Discovery-Regel mit Host-Prototypen verfügt, übernehmen entdeckte Hosts alle auf diesem Host definierten Benutzer-Makros

Beispiele

Beispiel 1

Verwendung eines Makros auf Host-Ebene im Schlüssel des Datenpunkts „Status des SSH-Daemons“:

```
net.tcp.service[ssh,,{$SSH_PORT}]
```

Dieser Datenpunkt kann mehreren Hosts zugewiesen werden, sofern der Wert von `{$SSH_PORT}` auf diesen Hosts definiert ist.

Beispiel 2

Verwendung eines Host-Makros im Auslöser „CPU load is too high“:

```
last(/ca_001/system.cpu.load[,avg1])>{$MAX_CPULOAD}
```

Ein solcher Auslöser würde in der Vorlage erstellt und nicht in einzelnen Hosts bearbeitet werden.

Note:
Wenn Sie die Anzahl der Werte als Funktionsparameter verwenden möchten (zum Beispiel `max(/host/key,#3)`), fügen Sie das Rautezeichen in die Makrodefinition ein, etwa so: `SOME_PERIOD => #3`.

Beispiel 3

Verwendung von zwei Makros im Auslöser „CPU-Auslastung ist zu hoch“:

```
min(/ca_001/system.cpu.load[,avg1],{$CPULOAD_PERIOD})>{$MAX_CPULOAD}
```

Beachten Sie, dass ein Makro als Parameter einer Auslöserfunktion verwendet werden kann, in diesem Beispiel der Funktion `min()`.

Beispiel 4

Synchronisieren Sie die Bedingung für die Nichtverfügbarkeit des Agent mit dem Aktualisierungsintervall des Datenpunkts:

- Definieren Sie das Makro `{$INTERVAL}` und verwenden Sie es im Aktualisierungsintervall des Datenpunkts.
- Verwenden Sie `{$INTERVAL}` als Parameter des Auslösers für die Nichtverfügbarkeit des Agent:

```
nodata(/ca_001/agent.ping,{$INTERVAL})=1
```

Beispiel 5

Zentralisieren Sie die Konfiguration der Arbeitszeiten:

- Erstellen Sie ein globales Makro `{$WORKING_HOURS}` mit dem Wert `1-5,09:00-18:00`.
- Verwenden Sie es im Feld *Arbeitszeit* unter *Administration* > *Allgemein* > *GUI*.
- Verwenden Sie es im Feld *Aktiv wenn* unter *Benutzer* > *Benutzer*, Registerkarte *Medien* eines Benutzers.
- Verwenden Sie es, um während der Arbeitszeit eine häufigere Abfrage von Datenpunkten einzurichten:

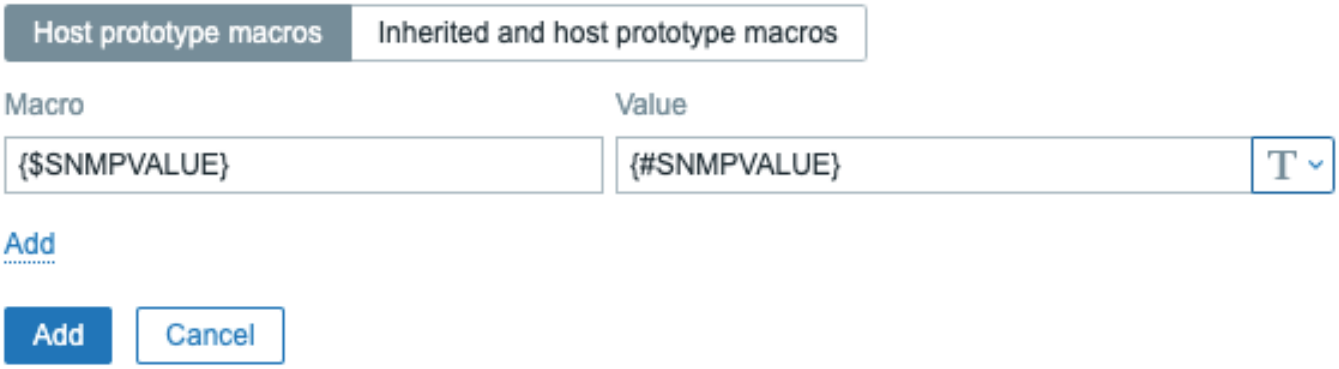


- Verwenden Sie es in der Aktionsbedingung *Zeitraum*.
- Passen Sie die Arbeitszeit bei Bedarf unter *Administration* > *Makros* an.

Beispiel 6

Verwenden Sie ein Host-Prototyp-Makro, um Datenpunkte für erkannte Hosts zu konfigurieren:

1. Definieren Sie in einem Host-Prototyp das Benutzermakro `{$SNMPVALUE}` mit dem Makro `{#SNMPVALUE}` für die Low-Level-Discovery als Wert:



2. Weisen Sie dem Host-Prototyp die Vorlage *Generic SNMPv2* zu.
3. Verwenden Sie `{$SNMPVALUE}` im Feld *SNMP OID* der Datenpunkte der Vorlage *Generic SNMPv2*.

Kontext von Benutzermakros

Siehe Benutzermakros mit Kontext.

Benutzermakros mit Kontext

Übersicht

Ein optionaler Kontext kann in Benutzermakros verwendet werden, sodass der Standardwert durch einen kontextspezifischen Wert überschrieben werden kann.

Der Kontext wird an den Makronamen angehängt. Die Syntax hängt davon ab, ob der Kontext ein statischer Textwert ist:

```
{$MACRO:"static text"}
```

oder ein regulärer Ausdruck:

```
{$MACRO:regex:"regular expression"}
```

Beachten Sie, dass ein Makro mit einem Kontext als regulärem Ausdruck nur in der Konfiguration von Benutzermakros definiert werden kann. Wenn das Präfix `regex:` an anderer Stelle als Kontext eines Benutzermakros verwendet wird, beispielsweise in einem Ausdruck eines Auslösers, wird es als statischer Kontext behandelt.

Das Setzen von Anführungszeichen für den Kontext ist optional (siehe auch wichtige Hinweise).

Beispiele für Makrokontexte:

Beispiel	Beschreibung
`{$LOW_SPACE_LIMIT}`	Benutzermakro ohne Kontext.
`{$LOW_SPACE_LIMIT:/tmp}`	Benutzermakro mit Kontext (statische Zeichenfolge).
`{$LOW_SPACE_LIMIT:regex:"^/tmp$"}`	Benutzermakro mit Kontext (regulärer Ausdruck). Entspricht `{$LOW_SPACE_LIMIT:/tmp}`.
`{$LOW_SPACE_LIMIT:regex:"^/var/log/.*$"}`	Benutzermakro mit Kontext (regulärer Ausdruck). Entspricht allen Zeichenfolgen, die mit /var/log/ beginnen.

Anwendungsfälle

Benutzer-Makros mit Kontext können definiert werden, um flexiblere Schwellwerte in Ausdrücken von Auslösern zu erreichen (basierend auf den von Low-Level-Discovery abgerufenen Werten). Sie können beispielsweise die folgenden Makros definieren:

- {$LOW_SPACE_LIMIT} = 10
- {$LOW_SPACE_LIMIT:/home} = 20
- {$LOW_SPACE_LIMIT:regex:”^/[a-z]+$”} = 30

Dann kann ein Low-Level-Discovery-Makro als Makro-Kontext in einem Auslöserprototyp für die Erkennung eingehängter Dateisysteme verwendet werden:

```
last(/host/vfs.fs.size[{#FSNAME},pfree])<{$LOW_SPACE_LIMIT:"{#FSNAME}"}
```

Nach der Erkennung gelten in Auslösern unterschiedliche Schwellwerte für wenig freien Speicherplatz, abhängig von den erkannten Einhängepunkten oder Dateisystemtypen. Problemereignisse werden generiert, wenn:

- der Ordner /home weniger als 20 % freien Speicherplatz hat
- Ordner, die dem regulären Ausdruck entsprechen (z. B. /etc, /tmp oder /var), weniger als 30 % freien Speicherplatz haben
- Ordner, die nicht dem regulären Ausdruck entsprechen und nicht /home sind, weniger als 10 % freien Speicherplatz haben

Wichtige Hinweise

- Wenn mehr als ein Benutzermakro mit Kontext vorhanden ist, versucht Zabbix zunächst, die Makros mit einfachem Kontext abzugleichen, und danach die Kontextmakros mit regulären Ausdrücken in einer nicht definierten Reihenfolge.

> **Warning:**
> Erstellen Sie keine unterschiedlichen Kontextmakros, die mit derselben Zeichenfolge übereinstimmen, um undefiniertes Verhalten zu vermeiden.

- Wenn ein Makro mit seinem Kontext weder auf dem Host, in verknüpften Vorlagen noch global gefunden wird, wird nach dem Makro ohne Kontext gesucht.
- Im Kontext werden nur Low-Level-Discovery-Makros unterstützt. Alle anderen Makros werden ignoriert und als Klartext behandelt.

Technisch wird der Makrokontext mit Regeln angegeben, die den Parametern des Datenpunkt-Schlüssels ähneln, mit der Ausnahme, dass der Makrokontext nicht als mehrere Parameter geparst wird, wenn ein `,`-Zeichen vorhanden ist:

- Der Makrokontext muss mit " in Anführungszeichen gesetzt werden, wenn der Kontext ein }-Zeichen enthält oder mit einem "-Zeichen beginnt. Anführungszeichen innerhalb eines in Anführungszeichen gesetzten Kontexts müssen mit dem Zeichen \ maskiert werden.
- Das Zeichen \ selbst wird nicht maskiert, was bedeutet, dass es unmöglich ist, einen in Anführungszeichen gesetzten Kontext zu haben, der mit dem Zeichen \ endet - das Makro {$MACRO:"a:\b\c\"} ist ungültig.
- Führende Leerzeichen im Kontext werden ignoriert, nachfolgende Leerzeichen jedoch nicht:
  - Zum Beispiel ist {$MACRO:A} dasselbe wie {$MACRO: A}, aber nicht {$MACRO:A }.
- Alle Leerzeichen vor führenden Anführungszeichen und nach schließenden Anführungszeichen werden ignoriert, alle Leerzeichen innerhalb der Anführungszeichen jedoch nicht:
  - Die Makros {$MACRO:"A"}, {$MACRO: "A"}, {$MACRO:"A" } und {$MACRO: "A" } sind gleich, aber die Makros {$MACRO:"A"} und {$MACRO:" A "} sind es nicht.

Die folgenden Makros sind alle äquivalent, da sie denselben Kontext haben: {$MACRO:A}, {$MACRO: A} und {$MACRO:"A"}. Dies steht im Gegensatz zu Datenpunkt-Schlüsseln, bei denen 'key[a]', 'key[a]' und 'key["a"]' semantisch gleich sind, sich jedoch im Hinblick auf die Eindeutigkeit unterscheiden.

Geheime Benutzermakros

Übersicht

Zabbix bietet zwei Optionen zum Schutz sensibler Informationen in Benutzer-Makrowerten:

- Geheimer Text
- Vault-Geheimnis

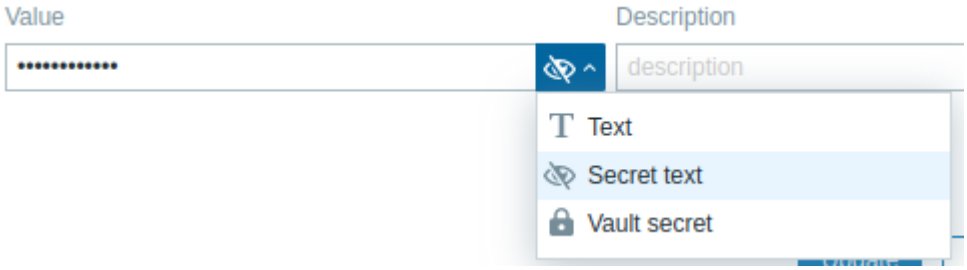
Note:
Obwohl der Wert eines geheimen Makros ausgeblendet ist, kann er durch die Verwendung in Datenpunkten offengelegt werden. Beispielsweise kann in einem externen Skript eine `echo`-Anweisung, die auf ein geheimes Makro verweist, verwendet werden, um den Makrowert im Frontend offenzulegen, da der Zabbix Server Zugriff auf den tatsächlichen Makrowert hat. Siehe Orte, an denen geheime Makrowerte eingeblendet werden.

Geheime Makros können nicht in Auslöser-Ausdrücken verwendet werden.

Geheimtext

Bei Geheimtext-Makros wird der Makrowert mit Sternchen maskiert.

Um einen Makrowert geheim zu machen, klicken Sie auf die Schaltfläche am Ende des Felds *Value* und wählen Sie die Option *Secret text*:



Sobald die Konfiguration gespeichert wurde, ist es nicht mehr möglich, den Wert anzuzeigen.

Um den Makrowert zu ändern, bewegen Sie den Mauszeiger über das Feld *Value* und klicken Sie auf die Schaltfläche *Set new value* (wird beim Darüberfahren angezeigt):



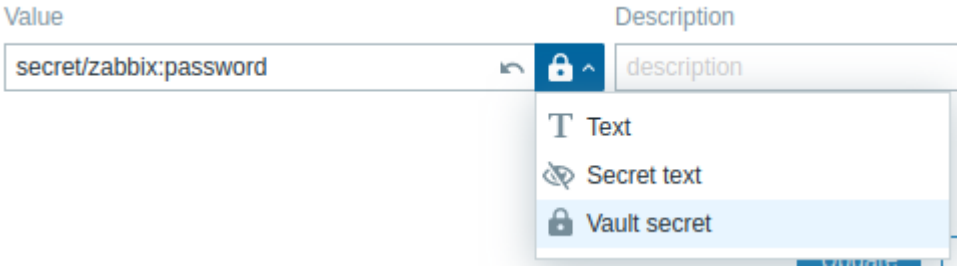
Wenn Sie auf die Schaltfläche *Set new value* klicken (oder den Makrowerttyp ändern), wird der aktuelle Wert gelöscht. Sie können den ursprünglichen Wert wiederherstellen, indem Sie auf den Pfeil am Ende des Felds *Value* klicken (nur vor dem Speichern der neuen Konfiguration verfügbar). Beachten Sie, dass das Wiederherstellen des ursprünglichen Werts ihn nicht sichtbar macht.

Note:
URLs, die ein geheimes Makro enthalten, funktionieren nicht, da das Makro darin als "******" aufgelöst wird.

Vault secret

Mit Vault secret-Makros wird der Makrowert in einer externen Software zur Geheimnisverwaltung (Vault) gespeichert.

Um ein Vault secret-Makro zu konfigurieren, klicken Sie auf die Schaltfläche am Ende des Feldes *Value* und wählen Sie die Option *Vault secret* aus:



Der Makrowert muss auf ein Vault-Geheimnis verweisen. Das Eingabeformat hängt vom Vault-Anbieter ab. Beispiele für anbieterspezifische Konfigurationen finden Sie unter:

- HashiCorp
- CyberArk

Werte von Vault-Geheimmakros werden vom Zabbix Server (und vom Zabbix Proxy, wenn *Resolve secret vault macros by* gesetzt ist auf *Zabbix server and proxy*) bei jeder Aktualisierung der Konfigurationsdaten aus dem Vault abgerufen und anschließend im Konfigurations-Cache gespeichert.
Zabbix Server und Zabbix Proxy können unterschiedliche Vaults verwenden.

Wenn *Resolve secret vault macros by* gesetzt ist auf *Zabbix server*, dann werden Vault-Geheimnisse nur vom Server abgerufen, und der Zabbix Proxy erhält die Werte der Vault-Geheimmakros bei jeder Konfigurationssynchronisierung vom Zabbix Server und speichert sie in seinem eigenen Konfigurations-Cache.
Das bedeutet, dass ein Zabbix Proxy nach einem Neustart die Datenerfassung erst starten kann, nachdem er die Konfigurationsaktualisierung vom Zabbix Server erhalten hat.

Um Geheimniswerte manuell aus dem Vault zu aktualisieren, verwenden Sie die Option `secrets_reload` der Laufzeitsteuerung (nur Server).

Die Verschlüsselung zwischen Zabbix Server und Proxy muss aktiviert sein; andernfalls wird eine Server-Warnmeldung protokolliert.

Warning:
Wenn ein Makrowert nicht erfolgreich abgerufen werden kann, wird der entsprechende Datenpunkt, der diesen Wert verwendet, nicht unterstützt.

Unmaskierte Orte

Diese Liste enthält Orte von Parametern, an denen geheime Makrowerte nicht maskiert sind.

Note:
Geheime Makrowerte bleiben an den unten aufgeführten Orten maskiert, wenn sie indirekt referenziert werden. Beispielsweise werden {ITEM.KEY}, {ITEM.KEY<1-9>}, {LLDRULE.KEY} integrierte Makros, die in Medientypen (Script- oder webhook-Parameter) verwendet werden, zu Item-Schlüsseln aufgelöst, die maskierte geheime Makros enthalten, z. B. `net.tcp.port[******,******]` statt `net.tcp.port[192.0.2.1,80]`.

Kontext		Parameter
Items, Item-Prototypen, LLD-Regeln		
	Item	*Item-Schlüsselparameter*
	Item-Prototyp	*Item-Prototyp-Schlüsselparameter*
	Low-Level-Discovery-Regel	*Discovery-Item-Schlüsselparameter*
	SNMP Agent	*SNMP community*
		Kontextname (SNMPv3)
		Sicherheitsname (SNMPv3)
		Authentifizierungs-Passphrase (SNMPv3)
		Privatsphären-Passphrase (SNMPv3)
	HTTP Agent	*URL*
		Abfragefelder
		Anforderungstext
		Header
		Benutzername

Kontext		Parameter
		Passwort
		SSL-Schlüsselpasswort
	Script	*Parameter*
		Script
	Browser	*Parameter*
		Script
	Datenbanküberwachung	*SQL-Abfrage*
	TELNET Agent	*Script*
		Benutzername
		Passwort
	SSH Agent	*Script*
		Benutzername
		Passwort
	Einfache Prüfung	*Benutzername*
		Passwort
	JMX Agent	*Benutzername*
		Passwort
Vorverarbeitung von Item-Werten		
	JavaScript-Vorverarbeitungsschritt	*Script*
Webszenarien		
	Webszenario	*Variablenwert*
		Headerwert
		URL
		Wert des Abfragefelds
		Wert des Post-Felds
		Roh-Post
	Webszenario-Authentifizierung	*Benutzer*
		Passwort
		SSL-Schlüsselpasswort
Connectoren		
	Connector	*URL*
		Benutzername
		Passwort
		Token
		HTTP-Proxy
		SSL-Zertifikatsdatei
		SSL-Schlüsseldatei
		SSL-Schlüsselpasswort
Netzwerkerkennung		
	SNMP	*SNMP community*
		Kontextname (SNMPv3)
		Sicherheitsname (SNMPv3)
		Authentifizierungs-Passphrase (SNMPv3)
		Privatsphären-Passphrase (SNMPv3)
Globale Scripts		
	Webhook	*JavaScript-Script*
		Wert des JavaScript-Script-Parameters
	Telnet	*Benutzername*
		Passwort
	SSH	*Benutzername*
		Passwort
	Script	*Script*
Medientypen		
	Script	*Script-Parameter*
	Webhook	*Parameter*
IPMI-Verwaltung		
	Host	*Benutzername*
		Passwort

Low-level discovery-Makros

Übersicht

Es gibt einen Makrotyp, der innerhalb der Funktion Low-Level-Discovery (LLD) verwendet wird:

```
{#MACRO}
```

Dabei handelt es sich um ein Makro, das in einer LLD-Regel verwendet wird und reale Werte des Dateisystemnamens, der Netzwerkschnittstelle, der SNMP-OID usw. zurückgibt.

Diese Makros können verwendet werden, um Entitätsprototypen (Datenpunkt-, Auslöser-, Graph-, LLD-Regel-, Host- und Hostgruppen-Prototypen) zu erstellen. Bei der Erkennung realer Dateisysteme, Netzwerkschnittstellen, virtueller Maschinen usw. werden diese Makros durch reale Werte ersetzt und bilden die Grundlage für die Erstellung realer Entitäten.

Einige Low-Level-Discovery-Makros werden in Zabbix zusammen mit der LLD-Funktion „mitgeliefert“ - {#FSNAME}, {#FSTYPE}, {#IFNAME}, {#SNMPINDEX}, {#SNMPVALUE}. Beim Erstellen einer benutzerdefinierten Low-Level-Discovery-Regel ist es jedoch nicht zwingend erforderlich, diese Namen zu verwenden. Dann können Sie jeden anderen LLD-Makronamen verwenden und auf diesen Namen verweisen.

Unterstützte Datentypen

Beim Definieren benutzerdefinierter Discovery-Regeln müssen Eigenschaftswerte, die in JSON-Objekten für LLD-Makros zurückgegeben werden, einer der folgenden primitiven Typen sein:

- string
- number
- boolean

Arrays, Objekte und null-Werte werden nicht unterstützt. Jedes LLD-Makro, das auf einen solchen Wert verweist, bleibt unausgewertet und wird bei der Vorverarbeitung und Erstellung von Datenpunkten wörtlich angezeigt (z. B. `'{#MY_MACRO}'`).

Unterstützte Stellen

LLD-Makros können verwendet werden:

- im Filter der Low-Level-Discovery-Regel
- in verschachtelten Low-Level-Discovery-Regeln, in
  - JSONPath-Preprocessing-Parametern
  - dem JSONPath-Feld für benutzerdefinierte LLD-Makros
- für Datenpunkt-Prototypen und Discovery-Prototypen in
  - Name
  - Schlüsselparametern
  - Einheit
  - Aktualisierungsintervall[1]
  - Timeout[1]
  - Aufbewahrungszeitraum für den Verlauf[1]
  - Aufbewahrungszeitraum für Trends[1]
  - Preprocessing-Schritten für Datenpunktwerte
  - SNMP-OID
  - IPMI-Sensorfeld
  - Ausdruck für berechnete/aggregierte Datenpunkte, in:
    - Ausdruckskonstanten und Funktionsparametern
    - Datenpunktschlüsselparametern
  - Filterbedingungen für aggregierte Datenpunkte (Hostgruppenname und Tag-Name)
  - SSH-Skript und Telnet-Skript
  - SQL-Abfrage für die Datenbanküberwachung
  - Endpunktfeld des JMX-Datenpunkts
  - Beschreibung
  - URL-Feld des HTTP-Agenten
  - Feld für HTTP-Abfragefelder des HTTP-Agenten
  - Feld für den Request-Body des HTTP-Agenten
  - Feld für erforderliche Statuscodes des HTTP-Agenten
  - Schlüssel und Wert im Header-Feld des HTTP-Agenten
  - Feld für den HTTP-Authentifizierungs-Benutzernamen des HTTP-Agenten
  - Feld für das HTTP-Authentifizierungs-Passwort des HTTP-Agenten
  - Feld für den HTTP-Proxy des HTTP-Agenten
  - Feld für die HTTP-SSL-Zertifikatsdatei des HTTP-Agenten
  - Feld für die HTTP-SSL-Schlüsseldatei des HTTP-Agenten
  - Feld für das HTTP-SSL-Schlüsselpasswort des HTTP-Agenten
  - Tags

- für Auslöser-Prototypen in
  - Name
  - Betriebsdaten
  - Ausdruck (nur in Konstanten und Funktionsparametern)
  - URL
  - Beschreibung
  - Tags
- für Graph-Prototypen in
  - Name
- für Host-Prototypen in
  - Name
  - sichtbarem Namen
  - benutzerdefinierten Schnittstellenfeldern: IP, DNS, Port, SNMP-v1/v2-Community, SNMP-v3-Kontextname, SNMP-v3-Sicherheitsname, SNMP-v3-Authentifizierungs-Passphrase, SNMP-v3-Privacy-Passphrase
  - Namen der Hostgruppen-Prototypen
  - Wert des Host-Tags
  - Wert des Host-Makros
  - (siehe die vollständige Liste)

An all diesen Stellen, mit Ausnahme des Filters der Low-Level-Discovery-Regel, können LLD-Makros innerhalb eines statischen Benutzer-Makrokontexts verwendet werden.

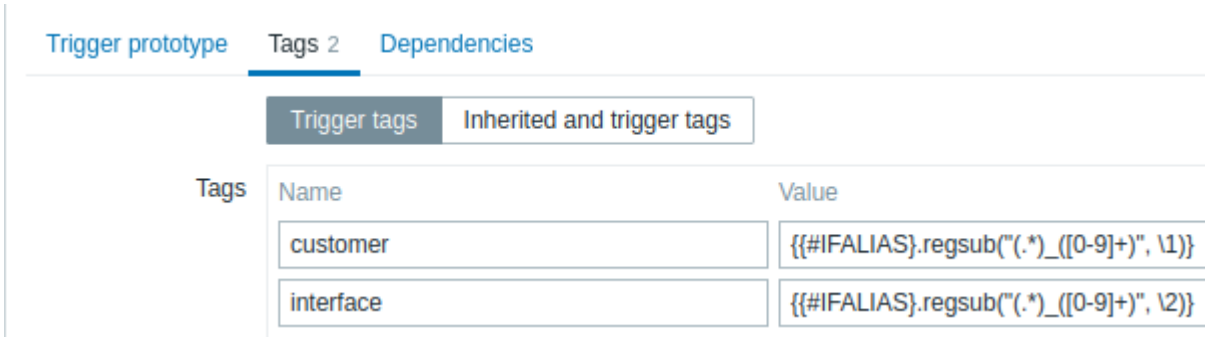
Makrofunktionen verwenden

Makrofunktionen werden mit Makros der Low-Level-Discovery unterstützt (außer in der Filter-Option der Low-Level-Discovery-Regel). Dadurch kann mithilfe eines regulären Ausdrucks ein bestimmter Teil des Makrowerts extrahiert werden.

Beispielsweise möchten Sie möglicherweise den Kundennamen und die Schnittstellennummer aus dem folgenden LLD-Makro für die Ereigniskennzeichnung extrahieren:

```
{#IFALIAS}=customername_1
```

Dazu kann die Makrofunktion `regsub` mit dem Makro im Wertfeld der Ereigniskennzeichnung eines Auslöser-Prototyps verwendet werden:



Beachten Sie, dass in nicht in Anführungszeichen gesetzten Parametern von Datenpunkt-Schlüsseln keine Kommas zulässig sind. Daher muss der Parameter, der eine Makrofunktion enthält, in Anführungszeichen gesetzt werden. Das Backslash-Zeichen (\) sollte verwendet werden, um doppelte Anführungszeichen innerhalb des Parameters zu maskieren. Beispiel:

```
net.if.in["{{#IFALIAS}.regsub(\"(.*)_([0-9]+)\", \1)}",bytes]
```

Weitere Informationen zur Syntax von Makrofunktionen finden Sie unter: Makrofunktionen

Makrofunktionen werden seit Zabbix 4.0 in Makros der Low-Level-Discovery unterstützt.

Datenpunkt-Prototypen ohne Schlüsselparameter

Wenn keine Datenpunktschlüsselparameter verwendet werden, platzieren Sie das LLD-Makro innerhalb der Parameterklammern `[...]` des Datenpunktschlüssels, damit es als Parameter behandelt und während der Discovery ersetzt wird:

```
v_[{#MACRO}]
```

Fußnoten

[1] In den mit [1] markierten Feldern muss ein einzelnes Makro das gesamte Feld ausfüllen. Mehrere Makros in einem Feld oder mit Text gemischte Makros werden nicht unterstützt.

Ausdrucksmakros

Übersicht

Ausdrucksmakros ermöglichen es Ihnen, Berechnungen in Feldern durchzuführen.

Ihr Wert wird berechnet, indem zuerst alle inneren Makros aufgelöst und anschließend der resultierende Ausdruck ausgewertet wird.

Syntax:

```
{?EXPRESSION}
```

EXPRESSION verwendet dieselbe Syntax und unterstützt dieselben Funktionen wie Ausdrücke für Auslöser.

Beispiel:

```
{?trendavg(/host/item1,1M:now/M)/trendavg(/host/item1,1M:now/M-1y)*100}
```

Für Stellen, die Ausdrucksmakros unterstützen, suchen Sie in der Tabelle der unterstützten Makros nach "{?EXPRESSION}".

Hinweise zur Verwendung:

- Verwenden Sie Makros {FUNCTION.*}, um auf Funktionswerte von Ausdrücken für Auslöser/Wiederherstellungsausdrücken zu verweisen.
- Verwenden Sie die Makros {HOST.HOST<1-9>} und {ITEM.KEY<1-9>}, um auf Hosts und Datenpunkte zu verweisen.
- Verwenden Sie in Vorlagen die Makros {HOST.HOST<1-9>} oder lassen Sie den Host für den ersten Host ganz weg - zum Beispiel {?avg(//item1,1h)} - anstatt Vorlagennamen zu verwenden, da Vorlagennamen beim Verknüpfen von Vorlagen nicht durch Hostnamen ersetzt werden.

```
{?{FUNCTION.VALUE2} - {FUNCTION.VALUE3}}
{?max(/{HOST.HOST}/{ITEM.KEY},3h)}
```

Siehe auch Beispiele für Ausdrücke für Auslöser für ein Beispiel zur Verwendung eines Ausdrucksmakros in einem Ereignisnamen.

Benutzer und Benutzergruppen

Übersicht

Alle Benutzer in Zabbix greifen über das webbasierte Frontend auf die Zabbix-Anwendung zu. Jedem Benutzer wird ein eindeutiger Anmeldename und ein Passwort zugewiesen.

Alle Benutzerpasswörter werden verschlüsselt und in der Zabbix-Datenbank gespeichert. Benutzer können ihre Benutzer-ID und ihr Passwort nicht verwenden, um sich direkt am UNIX-Server anzumelden, sofern sie nicht auch entsprechend in UNIX eingerichtet wurden. Die Kommunikation zwischen dem Webserver und dem Browser des Benutzers kann mithilfe von SSL geschützt werden.

Mit einem flexiblen Benutzerberechtigungsschema können Sie Rechte einschränken und differenzieren für:

- Zugriff auf administrative Zabbix-Frontend-Funktionen.
- Ausführen bestimmter Aktionen im Frontend.
- Zugriff auf überwachte Hosts in Hostgruppen.
- Verwendung bestimmter API-Methoden.

Einen Benutzer konfigurieren

Übersicht

Die anfängliche Zabbix-Installation enthält zwei vordefinierte Benutzer:

- *Admin* – ein Zabbix-Superuser mit vollständigen Berechtigungen.
- *guest* – ein spezieller Zabbix-Benutzer.

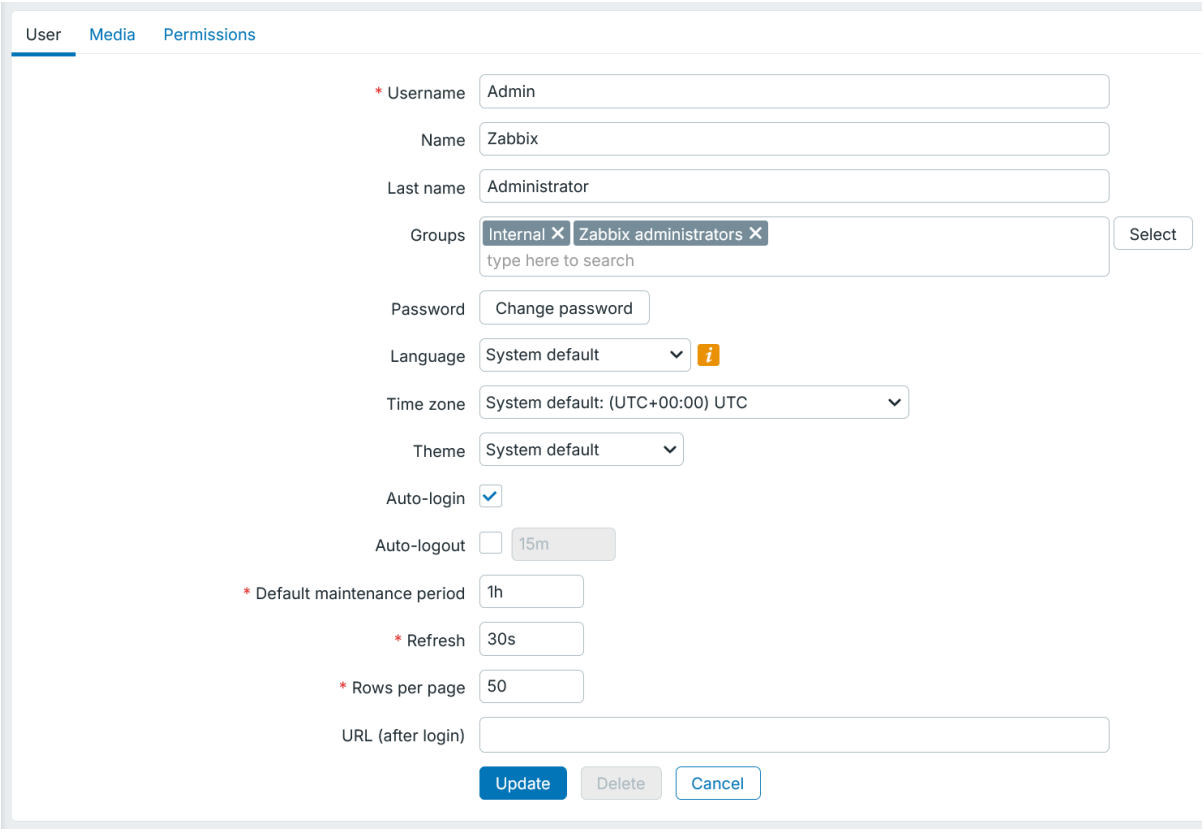
Der Benutzer „guest" ist standardmäßig deaktiviert.
Um den Benutzer zu aktivieren, entfernen Sie die vordefinierte Gruppe *Disabled* aus den Benutzergruppen.
Sie können sich mit diesem Benutzer anmelden und auf die Überwachungsseiten in Zabbix zugreifen.
Beachten Sie, dass „guest" standardmäßig keine Berechtigungen für Zabbix-Objekte besitzt.

So konfigurieren Sie einen Benutzer:

1. Gehen Sie zu *Benutzer* > *Benutzer*.
2. Klicken Sie auf *Benutzer erstellen* (oder auf einen Benutzernamen, um einen vorhandenen Benutzer zu bearbeiten).
3. Bearbeiten Sie die Benutzerattribute im Formular.

Allgemeine Attribute

Die Registerkarte *Benutzer* enthält allgemeine Benutzerattribute:



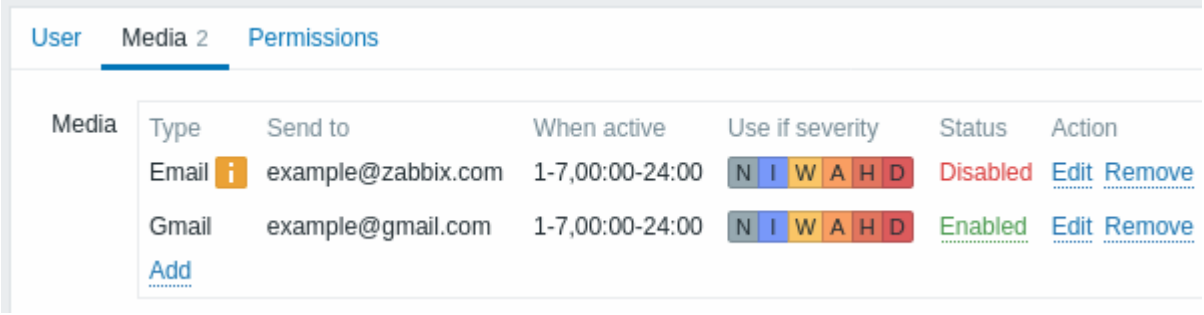
Alle Pflichtfelder sind mit einem roten Sternchen gekennzeichnet.

Parameter	Beschreibung
Benutzername	Eindeutiger Benutzername, der als Anmeldename verwendet wird.
Vorname	Vorname des Benutzers (optional). Wenn nicht leer, wird er in Bestätigungsinformationen und Informationen zu Benachrichtigungsempfängern angezeigt.
Nachname	Nachname des Benutzers (optional). Wenn nicht leer, wird er in Bestätigungsinformationen und Informationen zu Benachrichtigungsempfängern angezeigt.
Gruppen	Wählen Sie die Benutzergruppen aus, denen der Benutzer angehört. Dieses Feld unterstützt die automatische Vervollständigung. Wenn Sie beginnen, den Namen einer Benutzergruppe einzugeben, wird eine Dropdown-Liste mit passenden Gruppen angezeigt. Scrollen Sie nach unten, um eine Gruppe auszuwählen. Alternativ können Sie auf *Auswählen* klicken, um Gruppen hinzuzufügen. Klicken Sie auf „x“, um die ausgewählte Gruppe zu entfernen. Die Zugehörigkeit zu Benutzergruppen bestimmt, auf welche Hostgruppen und Hosts der Benutzer zugreifen kann.
Passwort	Zwei Felder zur Eingabe des Benutzerpassworts oder eine Schaltfläche *Passwort ändern*, wenn der Benutzer bereits vorhanden ist. Durch Klicken auf die Schaltfläche *Passwort ändern* werden zwei Felder zur Eingabe eines neuen Passworts geöffnet. Wenn ein Benutzer mit der Rolle *Superadministrator* sein eigenes Passwort ändert, wird durch Klicken auf die Schaltfläche *Passwort ändern* ein zusätzliches Feld zur Eingabe des aktuellen (alten) Passworts geöffnet. Nach einer erfolgreichen Passwortänderung wird der Benutzer, dessen Passwort geändert wurde, von allen aktiven Sitzungen abgemeldet. Beachten Sie, dass das Passwort nur für Benutzer geändert werden kann, die die interne Authentifizierung von Zabbix verwenden.
Sprache	Sprache des Zabbix Frontend. Damit die Übersetzungen funktionieren, ist die PHP-gettext-Erweiterung erforderlich.
Zeitzone	Wählen Sie die Zeitzone aus, um die globale Zeitzone auf Benutzerebene zu überschreiben, oder wählen Sie **Systemstandard**, um die globalen Zeitzoneneinstellungen zu verwenden.

Parameter	Beschreibung
Theme	Legt das Erscheinungsbild des Frontend fest: **Systemstandard** – Standardsystemeinstellungen verwenden **Blau** – standardmäßiges blaues Theme **Blau (klassisch)** – standardmäßiges blaues Theme mit Legacy-Schriftarten **Dunkel** – alternatives dunkles Theme **Dunkelblau** – dunkelblaues Theme **Dunkel (klassisch)** – alternatives dunkles Theme mit Legacy-Schriftarten **Hell mit hohem Kontrast** – helles Theme mit hohem Kontrast **Dunkel mit hohem Kontrast** – dunkles Theme mit hohem Kontrast
Automatische Anmeldung	Aktivieren Sie dieses Kontrollkästchen, damit Zabbix den Benutzer 30 Tage lang speichert und automatisch anmeldet. Bei der Anmeldung mit *Für 30 Tage speichern*: - Die automatische Abmeldung wird zurückgesetzt (die Sitzung bleibt 30 Tage lang bestehen). - Die automatische Anmeldung wird für eine nahtlose erneute Authentifizierung aktiviert. Bei der Anmeldung ohne *Für 30 Tage speichern*: - Die automatische Anmeldung wird für eine nahtlose erneute Authentifizierung aktiviert. - Die automatische Abmeldung wird weiterhin durch die standardmäßige Zeitüberschreitung gesteuert. Hierfür werden Browser-Cookies verwendet.
Automatische Abmeldung	Wenn dieses Kontrollkästchen aktiviert ist, wird der Benutzer nach der festgelegten Anzahl von Sekunden automatisch abgemeldet (mindestens 90 Sekunden, höchstens 1 Tag). Beachten Sie, dass diese Einstellung überschrieben wird, wenn *Für 30 Tage speichern* aktiviert ist, da die Sitzung für den gesamten Zeitraum verlängert wird. Zeitsuffixe werden unterstützt, z. B. 90s, 5m, 2h, 1d. Beachten Sie, dass diese Option nicht funktioniert: * Wenn die globale Konfigurationsoption „Warnung anzeigen, wenn der Zabbix-Server nicht verfügbar ist“ aktiviert ist und das Zabbix Frontend geöffnet bleibt. * Wenn Seiten des Menüs „Überwachung“ Hintergrundaktualisierungen von Informationen durchführen. * Wenn die Anmeldung mit aktivierter Option *Für 30 Tage speichern* erfolgt.
Standardwartungszeitraum	Legen Sie den standardmäßigen Wartungszeitraum für Ad-hoc-Wartungen fest.
Aktualisierung	Legen Sie die Aktualisierungsrate für Diagramme, Klartextdaten usw. fest. Kann auf 0 gesetzt werden, um die Aktualisierung zu deaktivieren. Zeitsuffixe werden unterstützt, z. B. 90s, 5m, 1h.
Zeilen pro Seite	Sie können festlegen, wie viele Zeilen pro Seite in Listen angezeigt werden.
URL (nach der Anmeldung)	Sie können Zabbix so konfigurieren, dass der Benutzer nach einer erfolgreichen Anmeldung zu einer bestimmten URL weitergeleitet wird, beispielsweise zur Seite *Probleme*.

Benutzermedien

Die Registerkarte *Medien* enthält eine Auflistung aller für den Benutzer definierten Medien.
Medien werden zum Senden von Benachrichtigungen verwendet.



Klicken Sie auf *Hinzufügen*, um dem Benutzer ein Medium zuzuweisen.

Wenn der Medientyp deaktiviert wurde:

- Wird nach dem Namen ein gelbes Informationssymbol angezeigt.
- Wird in der Spalte Status *Deaktiviert* angezeigt.

Note:
Benutzerberechtigungen zum Ändern eigener Mediendetails können basierend auf ihrer Benutzerrolle gewährt oder entzogen werden (siehe die Option *Eigene Medien erstellen und bearbeiten*). Super-Admin-Benutzerberechtigungen zum Ändern von Mediendetails anderer Benutzer können ebenfalls basierend auf ihrer Benutzerrolle gewährt oder entzogen werden (siehe die Option *Benutzermedien erstellen und bearbeiten*).

Weitere Informationen zur Konfiguration von Benutzermedien finden Sie im Abschnitt Medientypen.

Berechtigungen

Die Registerkarte *Berechtigungen* enthält Informationen zu den folgenden Elementen:

- Benutzerrolle (für jeden neu erstellten Benutzer erforderlich), die nur von einem Benutzer des Typs *Super-Admin* geändert werden kann.

Attention:
Benutzer können nicht ohne eine Benutzerrolle erstellt werden (außer über die Zabbix-User API). Bereits erstellte Benutzer ohne Rolle können weiterhin bearbeitet werden, ohne ihnen eine Rolle zuzuweisen. Sobald jedoch eine Rolle zugewiesen wurde, kann sie nur geändert, nicht aber entfernt werden.

 Beachten Sie, dass sich Benutzer ohne Rolle nur über die LDAP- oder SAML-Authentifizierung bei Zabbix anmelden können, sofern ihre LDAP-/SAML-Informationen mit den in Zabbix konfigurierten Benutzergruppen-Zuordnungen übereinstimmen.

- Benutzertyp (*Benutzer*, *Administrator*, *Super-Admin*), der in der Konfiguration der Benutzerrolle definiert ist.
- Host- und Vorlagengruppen, auf die der Benutzer Zugriff hat.
  - Benutzer des Typs *Benutzer* und *Administrator* haben standardmäßig keinen Zugriff auf Gruppen, Vorlagen und Hosts. Um diesen Zugriff zu gewähren, müssen Benutzer in Benutzergruppen aufgenommen werden, die mit Berechtigungen für die entsprechenden Entitäten konfiguriert sind.
- Zugriffsrechte auf Bereiche und Elemente des Zabbix Frontends, Module und API-Methoden.
  - Elemente mit erlaubtem Zugriff werden grün angezeigt, während Elemente mit verweigertem Zugriff hellgrau angezeigt werden.
  - Wenn der Zugriff auf die API aktiviert ist, werden die API-Methoden der Benutzerrolle unter *Erlaubte Methoden* oder *Verweigerte Methoden* aufgelistet, abhängig davon, ob in der Konfiguration der Benutzerrolle *Zulassungsliste* oder *Sperrliste* ausgewählt ist. Wenn keine API-Methoden angegeben sind, wird *Keine* angezeigt. *Keine* unter *Erlaubte Methoden* bedeutet, dass keine API-Methoden erlaubt sind, während *Keine* unter *Verweigerte Methoden* bedeutet, dass alle für den Benutzertyp verfügbaren API-Methoden erlaubt sind.
- Berechtigungen zum Ausführen bestimmter Aktionen.
  - Aktionen, die der Benutzer ausführen darf, werden grün angezeigt, während verweigerte Aktionen hellgrau angezeigt werden.

Weitere Informationen finden Sie auf der Seite Berechtigungen.

Berechtigungen

Übersicht

Berechtigungen in Zabbix hängen vom Benutzertyp, von angepassten Benutzerrollen und vom Zugriff auf Hosts ab, der auf Grundlage der Benutzergruppe festgelegt wird.

Benutzertypen

Die Berechtigungen in Zabbix hängen in erster Linie vom Benutzertyp ab:

- *Benutzer* – hat eingeschränkte Zugriffsrechte auf Menüabschnitte (siehe unten) und standardmäßig keinen Zugriff auf Ressourcen. Berechtigungen für Host- oder Vorlagengruppen müssen ausdrücklich zugewiesen werden.
- *Administrator* – hat unvollständige Zugriffsrechte auf Menüabschnitte (siehe unten). Der Benutzer hat standardmäßig keinen Zugriff auf Hostgruppen. Berechtigungen für Host- oder Vorlagengruppen müssen ausdrücklich erteilt werden.
- *Superadministrator* – hat Zugriff auf alle Menüabschnitte. Der Benutzer hat Lese- und Schreibzugriff auf alle Host- und Vorlagengruppen. Berechtigungen können nicht durch die Verweigerung des Zugriffs auf bestimmte Gruppen entzogen werden.

Menüzugriff

Note:
Der eingeschränkte Zugriff auf einige UI-Elemente verhindert lediglich das Öffnen dieser Seite. Die Möglichkeit, auf zugrunde liegende Daten in anderen Teilen der Oberfläche zuzugreifen, wird dadurch nicht entfernt.

Die folgende Tabelle veranschaulicht den Zugriff auf die Menüabschnitte von Zabbix nach Benutzertyp:

Menüabschnitt		Benutzer	Administrator	Superadministrator
Dashboards		+	+	+
Überwachung		+	+	+
	Probleme	+	+	+
	Hosts	+	+	+
	Neueste Daten	+	+	+
	Maps	+	+	+
	Discovery		+	+
APM		+	+	+
	Traces	+	+	+
	Metriken	+	+	+
	Protokolle	+	+	+
Services		+	+	+
	Services	+	+	+
	SLA		+	+
	SLA-Bericht	+	+	+
Inventar		+	+	+
	Übersicht	+	+	+
	Hosts	+	+	+
Berichte		+	+	+
	Systeminformationen		+	+
	Geplante Berichte		+	+
	Verfügbarkeitsbericht	+	+	+
	Top-100-Auslöser	+	+	+
	Auditprotokoll			+
	Aktionsprotokoll			+
	Benachrichtigungen		+	+
Datenerfassung			+	+
	Vorlagengruppen		+	+
	Hostgruppen		+	+
	Vorlagen		+	+
	Hosts		+	+
	Wartung		+	+
	Ereigniskorrelation			+
	Discovery		+	+
Alarme			+	+
	Auslöseraktionen		+	+
	Serviceaktionen		+	+
	Discovery-Aktionen		+	+
	Autoregistrierungsaktionen		+	+
	Interne Aktionen		+	+
	Medientypen			+
	Skripte			+
Benutzer				+
	Benutzergruppen			+
	Benutzerrollen			+
	Benutzer			+
	API-Tokens			+
	Authentifizierung			+
	Geräte			+
Administration				+
	Datenquelle			+
	Allgemein			+
	Auditprotokoll			+
	Housekeeping			+
	Proxygruppen			+
	Proxys			+
	Makros			+
	Warteschlange			+

Benutzerrollen

Benutzerrollen ermöglichen benutzerdefinierte Anpassungen der durch den Benutzertyp festgelegten Berechtigungen.
Es können zwar keine Berechtigungen hinzugefügt werden (da dies die Berechtigungen des Benutzertyps überschreiten würde), einige Berechtigungen können jedoch entzogen werden.

Darüber hinaus bestimmt eine Benutzerrolle den Zugriff nicht nur auf Menüabschnitte, sondern auch auf Services, Module, API-Methoden und verschiedene Aktionen im Frontend.

Benutzerrollen werden von Super-Admin-Benutzern im Abschnitt *Benutzer* > *Benutzerrollen* konfiguriert.

Benutzerrollen werden von Super-Admin-Benutzern im Tab *Berechtigungen* des Benutzerkonfigurationsformulars Benutzern zugewiesen.

Users

User | Media | Permissions

* Role: Admin role ✕ | Select

User type: Admin

Permissions

User group permissions

Group	Type	Permissions
All groups	Hosts	None
All groups	Templates	None

Proxies: None

Proxy groups: None

Permissions can be assigned for user groups only.

Access to UI elements

Dashboards: Dashboards

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

APM: Traces | Metrics | Logs

Services: Services | SLA | SLA report

Inventory: Overview | Hosts

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

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

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

User profile settings

Redirect URL after login

Access to services

Read-write access to services: None

Read-only access to services: All

Access to modules

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

Access to API

Disabled

Devices

Enabled | Manage own devices

Access to actions

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

Add | Cancel

Zugriff auf Hosts

Der Zugriff auf beliebige Host- und Vorlagendaten in Zabbix wird Benutzergruppen ausschließlich auf der Ebene der Host-/Vorlagengruppen gewährt.

Das bedeutet, dass einem einzelnen Benutzer nicht direkt Zugriff auf einen Host (oder eine Host-Gruppe) gewährt werden kann. Zugriff auf einen Host kann nur gewährt werden, wenn der Benutzer Teil einer Benutzergruppe ist, der Zugriff auf die Host-Gruppe gewährt wurde, die den Host enthält.

Ebenso kann einem Benutzer Zugriff auf eine Vorlage nur gewährt werden, wenn er Teil einer Benutzergruppe ist, der Zugriff auf die Vorlagengruppe gewährt wurde, die die Vorlage enthält.

Benutzergruppen

Übersicht

Benutzergruppen ermöglichen es Ihnen, Benutzer sowohl zu organisatorischen Zwecken als auch zur Vergabe von Berechtigungen für Daten zu gruppieren. Berechtigungen zum Anzeigen und Konfigurieren von Daten aus Hostgruppen und Vorlagengruppen werden Benutzergruppen und nicht einzelnen Benutzern zugewiesen.

Oft kann es sinnvoll sein, zu trennen, welche Informationen für eine Benutzergruppe und welche für eine andere verfügbar sind. Dies kann erreicht werden, indem Benutzer gruppiert und anschließend unterschiedliche Berechtigungen für Host- und Vorlagengruppen zugewiesen werden.

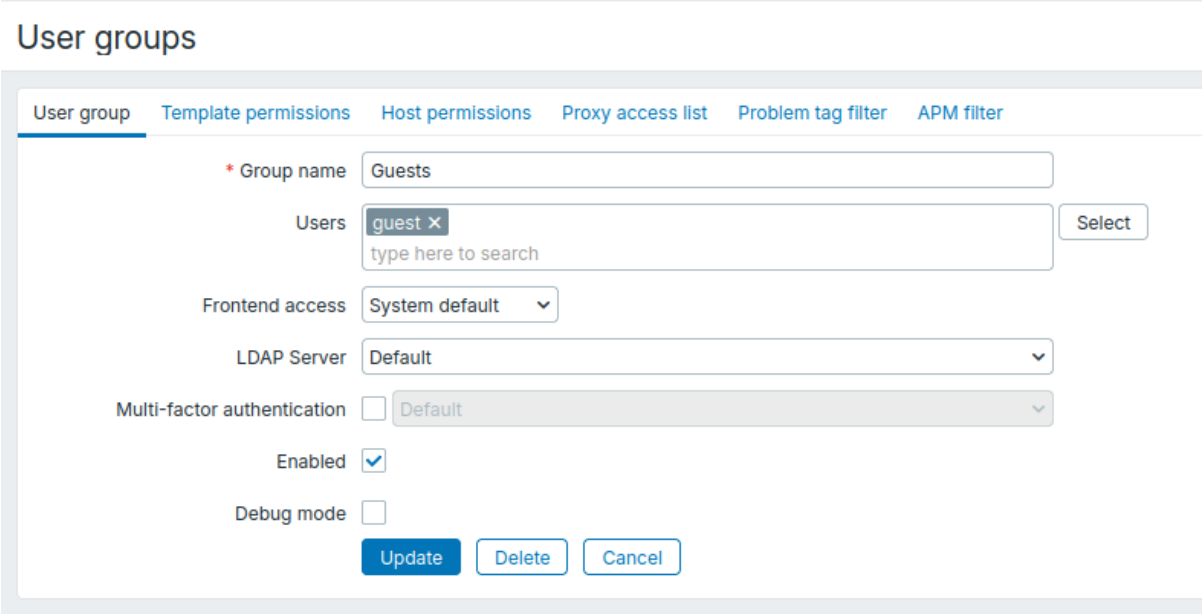
Ein Benutzer kann einer beliebigen Anzahl von Gruppen angehören.

Konfiguration

So konfigurieren Sie eine Benutzergruppe:

1. Gehen Sie zu *Benutzer* > *Benutzergruppen*.
2. Klicken Sie auf *Benutzergruppe erstellen* (oder auf den Namen der Gruppe, um eine vorhandene Gruppe zu bearbeiten).
3. Bearbeiten Sie die Gruppenattribute im Formular.

Die Registerkarte **Benutzergruppe** enthält allgemeine Gruppenattribute:

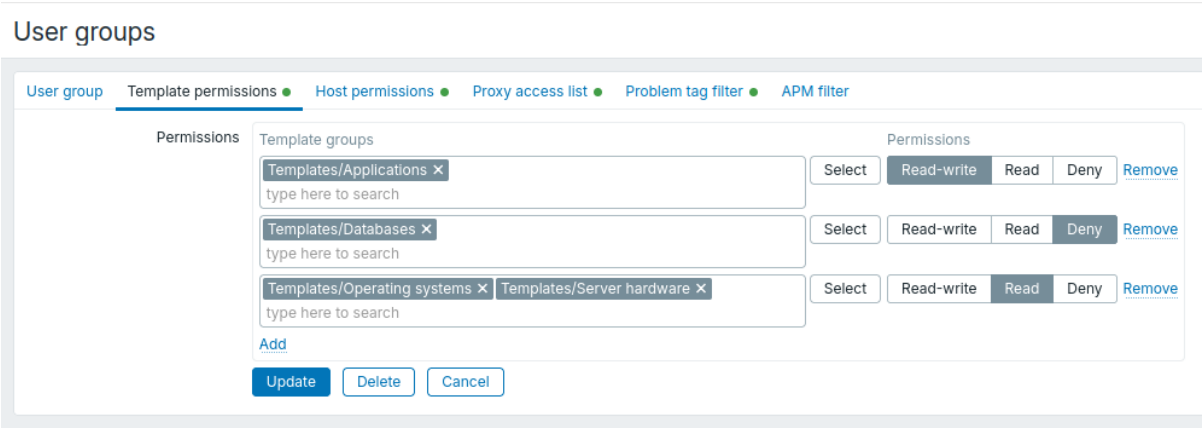


Alle Pflichtfelder sind mit einem roten Sternchen gekennzeichnet.

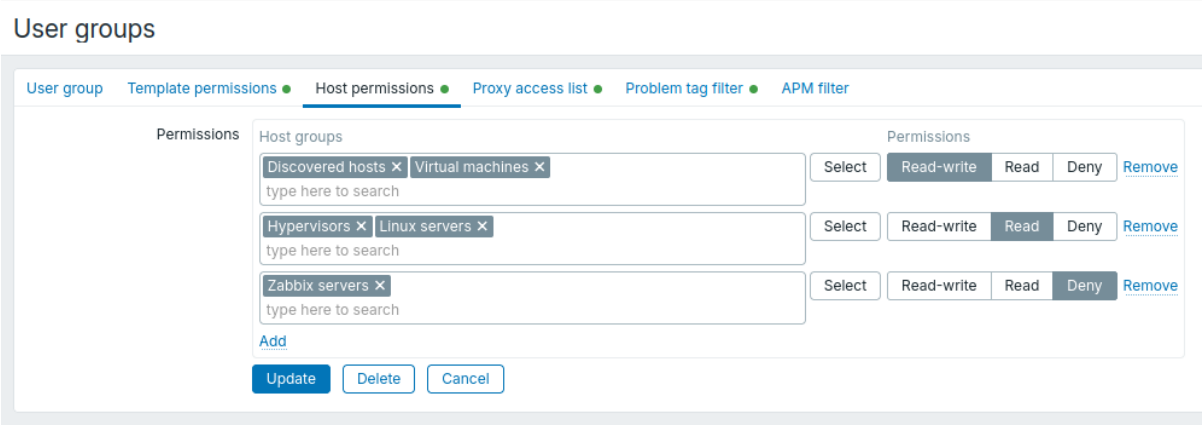
Parameter	Beschreibung
Gruppenname	Eindeutiger Gruppenname.
Benutzer	Um Benutzer zur Gruppe hinzuzufügen, beginnen Sie mit der Eingabe des Namens eines vorhandenen Benutzers. Wenn die Dropdown-Liste mit passenden Benutzernamen angezeigt wird, scrollen Sie nach unten, um einen Benutzer auszuwählen. Alternativ können Sie auf die Schaltfläche *Auswählen* klicken, um Benutzer in einem Popup-Fenster auszuwählen.

Parameter	Beschreibung
Frontend-Zugriff	Legt fest, wie die Benutzer der Gruppe authentifiziert werden. **Systemstandard** – verwendet die standardmäßige Authentifizierungsmethode (global festgelegt unter Globale Einstellungen) **Intern** – verwendet die interne Zabbix-Authentifizierung (auch wenn global die LDAP-Authentifizierung verwendet wird). Wird ignoriert, wenn die HTTP-Authentifizierung global als Standard festgelegt ist. **LDAP** – verwendet die LDAP-Authentifizierung (auch wenn global die interne Authentifizierung verwendet wird). Wird ignoriert, wenn die HTTP-Authentifizierung global als Standard festgelegt ist. **Deaktiviert** – der Zugriff auf das Zabbix-Frontend ist für diese Gruppe verboten
LDAP-Server	Wählen Sie den LDAP-Server aus, der zur Authentifizierung des Benutzers verwendet werden soll. Dieses Feld ist nur aktiviert, wenn *Frontend-Zugriff* auf LDAP oder Systemstandard gesetzt ist.
Multi-Faktor-Authentifizierung	Wählen Sie die Methode für die Multi-Faktor-Authentifizierung aus, die zur Authentifizierung des Benutzers verwendet werden soll: **Standard** – verwendet die in der MFA-Konfiguration als Standard festgelegte Methode; diese Option ist standardmäßig für neue Benutzergruppen ausgewählt, wenn MFA aktiviert ist; **<Methodenname>** – verwendet die ausgewählte Methode (z. B. „Zabbix TOTP“); **Deaktiviert** – MFA ist für diese Gruppe deaktiviert; diese Option ist standardmäßig für neue Benutzergruppen ausgewählt, wenn MFA deaktiviert ist. Beachten Sie, dass bei einem Benutzer, der mehreren Benutzergruppen mit aktivierter MFA angehört (oder wenn mindestens eine Gruppe MFA aktiviert hat), die folgenden Authentifizierungsregeln gelten: Wenn eine Gruppe die MFA-Methode „Standard“ verwendet, wird der Benutzer mit dieser Methode authentifiziert. Andernfalls wird die erste Methode (alphabetisch sortiert) zur Authentifizierung verwendet.
Aktiviert	Status der Benutzergruppe und ihrer Mitglieder. *Aktiviert* – Benutzergruppe und Benutzer sind aktiviert *Deaktiviert* – Benutzergruppe und Benutzer sind deaktiviert
Debug-Modus	Aktivieren Sie dieses Kontrollkästchen, um den Debug-Modus für die Benutzer zu aktivieren.

Auf der Registerkarte **Vorlagenberechtigungen** können Sie den Zugriff der Benutzergruppe auf Vorlagengruppen und darüber auf Vorlagendaten festlegen:



Auf der Registerkarte **Host-Berechtigungen** können Sie den Zugriff der Benutzergruppe auf Hostgruppen und darüber auf Hostdaten festlegen:



Klicken Sie auf Add, um die Vorlagen-/Hostgruppen (übergeordnete oder verschachtelte Gruppen) auszuwählen und ihnen Berechtigungen zuzuweisen. Beginnen Sie mit der Eingabe des Gruppennamens (eine Dropdown-Liste mit passenden Gruppen wird angezeigt) oder klicken Sie auf *Auswählen*, um ein Popup-Fenster mit allen Gruppen zu öffnen.

Verwenden Sie anschließend die Optionsschaltflächen, um den ausgewählten Gruppen Berechtigungen zuzuweisen. Folgende Berechtigungen sind möglich:

- **Lesen-Schreiben** – Lese- und Schreibzugriff auf eine Gruppe
- **Lesen** – schreibgeschützter Zugriff auf eine Gruppe
- **Verweigern** – Zugriff auf eine Gruppe verweigert

Wenn dieselbe Vorlagen-/Hostgruppe in mehreren Zeilen mit unterschiedlichen Berechtigungen hinzugefügt wird, wird die strengste Berechtigung angewendet.

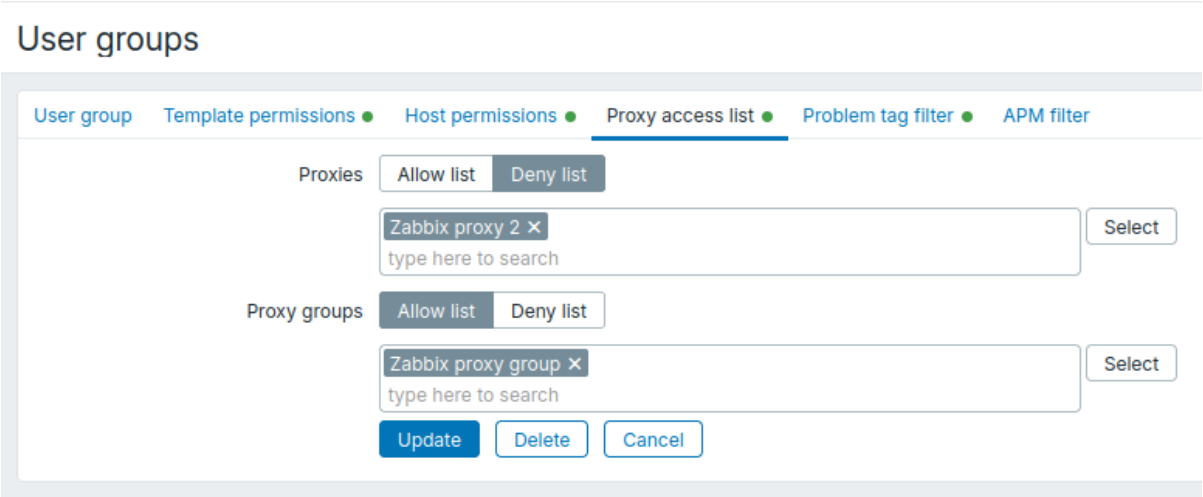
Beachten Sie, dass ein Benutzer mit der Rolle *Super-Admin* erzwingen kann, dass verschachtelte Gruppen dieselbe Berechtigungsstufe wie die übergeordnete Gruppe erhalten. Dies kann im Konfigurationsformular der Hostgruppen/Vorlagengruppen festgelegt werden.

Die Registerkarten **Vorlagenberechtigungen** und **Host-Berechtigungen** unterstützen denselben Parametersatz.

Die aktuellen Berechtigungen für Gruppen werden im Block *Berechtigungen* angezeigt und können dort geändert oder entfernt werden.

Note:
Wenn eine Benutzergruppe **Lesen-Schreiben**-Berechtigungen für einen Host, aber **Verweigern** oder keine Berechtigungen für eine mit diesem Host verknüpfte Vorlage besitzt, können die Benutzer dieser Gruppe keine Vorlagen-Datenpunkte auf dem Host bearbeiten. Der Vorlagenname wird als *Nicht zugängliche Vorlage* angezeigt.

Auf der Registerkarte **Proxy-Zugriffsliste** können Sie den Zugriff der Benutzergruppe auf Proxys einzeln oder über Proxygruppen festlegen:



Verwenden Sie das Feld *Proxygruppen*, um die Berechtigung auf die ausgewählten Gruppen anzuwenden. Ein Proxy, der zu einer Gruppe gehört, kann nur über diese Gruppe erlaubt oder verweigert werden. Verwenden Sie das Feld *Proxys*, um einzelne Proxys hinzuzufügen, die keiner Proxygruppe angehören.

Proxys und *Proxygruppen* unterstützen denselben Parametersatz.

Klicken Sie neben dem Feld *Proxys* oder *Proxygruppen* auf die Schaltfläche *Auswählen*, um Proxys bzw. Proxygruppen auszuwählen. Alternativ können Sie im entsprechenden Feld mit der Eingabe des Namens eines Proxys oder einer Proxygruppe beginnen. Eine Dropdown-Liste mit passenden Proxys oder Proxygruppen wird angezeigt.

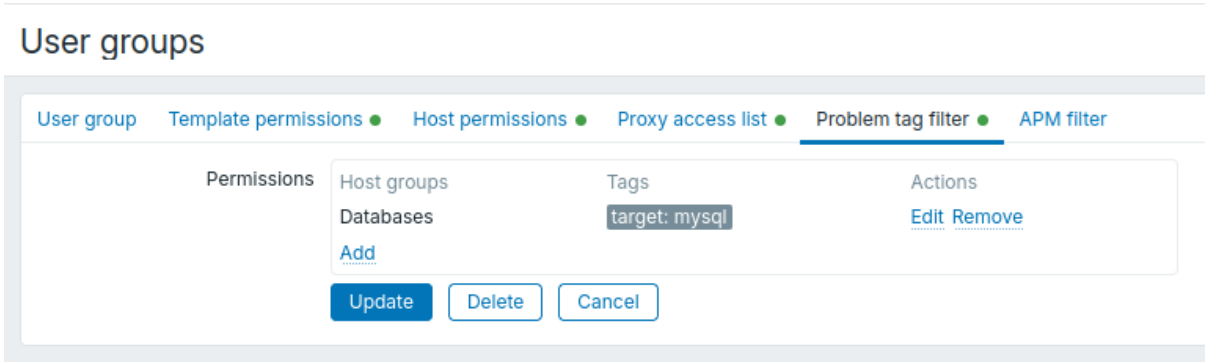
Wählen Sie *Zulassungsliste*, um der Benutzergruppe nur den Zugriff auf die ausgewählten Proxys und Proxygruppen zu erlauben.
 Wenn die Auswahl *Proxys* für die *Zulassungsliste* leer ist, hat die Benutzergruppe keinen Zugriff auf Proxys, die keiner Proxygruppe angehören. Wenn die Auswahl *Proxygruppen* für die *Zulassungsliste* leer ist, hat die Benutzergruppe keinen Zugriff auf Proxygruppen.

Wählen Sie *Sperrliste*, um der Benutzergruppe den Zugriff auf die ausgewählten Proxys und Proxygruppen zu verweigern. Der Zugriff auf alle anderen Proxys und Proxygruppen ist erlaubt.
 Wenn die Auswahl *Proxys* für die *Sperrliste* leer ist, hat die Benutzergruppe vollständigen Zugriff auf alle Proxys, die keiner Proxygruppe angehören. Wenn die Auswahl *Proxygruppen* für die *Sperrliste* leer ist, hat die Benutzergruppe vollständigen Zugriff auf alle Proxygruppen.

Regeln für den Vorrang von Berechtigungen für Benutzer, die mehreren Benutzergruppen angehören:

- Die *Sperrliste* hat Vorrang vor der *Zulassungsliste*. Wenn eine Gruppe den Zugriff auf einen Proxy/eine Proxygruppe verweigert, kann der Benutzer nicht darauf zugreifen.
- Die Berechtigungen der *Zulassungsliste* werden kumuliert. Wenn mehrere Gruppen den Zugriff erlauben, kann der Benutzer auf alle von diesen Gruppen erlaubten Proxys zugreifen.

Auf der Registerkarte **Problem-Tag-Filter** können Sie tagbasierte Berechtigungen für Benutzergruppen festlegen, damit Probleme nach Tagname und Tagwert gefiltert angezeigt werden:

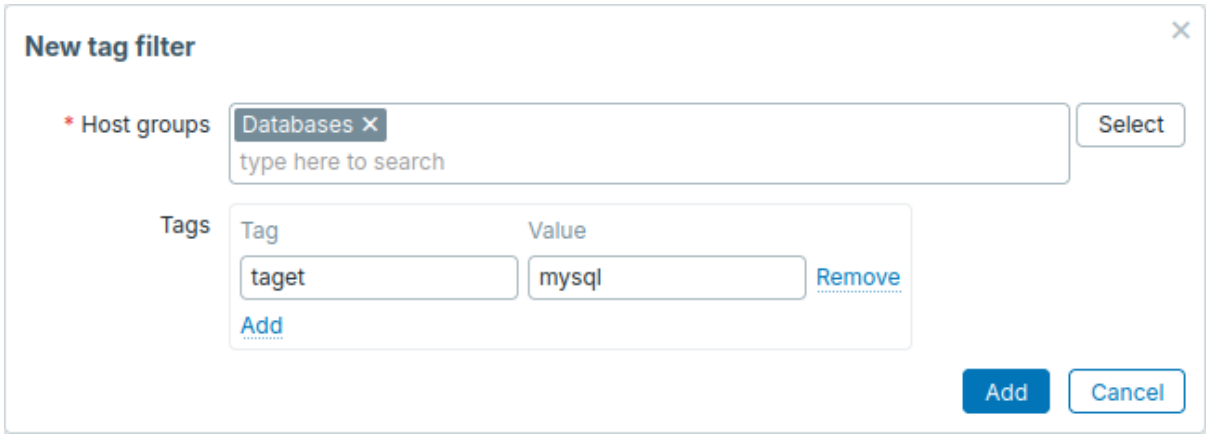


Klicken Sie auf Add, um die Hostgruppen auszuwählen. Um eine Hostgruppe auszuwählen, auf die ein Tag-Filter angewendet werden soll, klicken Sie auf *Auswählen*, um die vollständige Liste der vorhandenen Hostgruppen anzuzeigen, oder beginnen Sie mit der Eingabe des Namens einer Hostgruppe, um eine Dropdown-Liste mit passenden Gruppen anzuzeigen. Es werden nur Hostgruppen angezeigt, da der Problem-Tag-Filter nicht auf Vorlagengruppen angewendet werden kann.

Tagnamen ohne Werte können hinzugefügt werden, Werte ohne Namen jedoch nicht. Nur die ersten drei Tags (gegebenenfalls mit Werten) werden im Block *Berechtigungen* angezeigt. Wenn weitere Tags vorhanden sind, können Sie diese durch Klicken auf das Symbol ••• oder durch Bewegen des Mauszeigers darüber anzeigen.

Mit dem Tag-Filter kann der Zugriff auf eine Hostgruppe von der Möglichkeit getrennt werden, Probleme anzuzeigen.

Wenn ein Datenbankadministrator beispielsweise nur „MySQL“-Datenbankprobleme sehen soll, muss zunächst eine Benutzergruppe für Datenbankadministratoren erstellt und anschließend der Tagname „target“ sowie der Wert „mysql“ angegeben werden.



Wenn der Tagname „target“ angegeben und das Wertefeld leer gelassen wird, sieht die Benutzergruppe für die ausgewählte Hostgruppe alle Probleme mit dem Tagname „target“.

Stellen Sie sicher, dass Tagname und Tagwert korrekt angegeben sind. Andernfalls sieht die Benutzergruppe keine Probleme.

Sehen wir uns ein Beispiel an, bei dem ein Benutzer Mitglied mehrerer ausgewählter Benutzergruppen ist. Die Filterung verwendet in diesem Fall eine ODER-Bedingung für Tags.

Benutzergruppe A			**Benutzergruppe B**			**Für einen Benutzer (Mitglied) beider Gruppen sichtbares Ergebnis**
Tag-Filter						
Hostgruppe	*Tagname*	*Tagwert*	*Hostgruppe*	*Tagname*	*Tagwert*	
Datenbanken	target	mysql	Datenbanken	target	oracle	Probleme mit target:mysql oder target:oracle sichtbar
Nicht im **Problem-Tag-Filter** konfiguriert			Datenbanken	target	oracle	Probleme mit target:oracle sichtbar

Zugriff aus mehreren Benutzergruppen

Ein Benutzer kann einer beliebigen Anzahl von Benutzergruppen angehören. Diese Gruppen können unterschiedliche Zugriffsberechtigungen auf Hosts oder Vorlagen haben.

Daher ist es wichtig zu wissen, auf welche Entitäten ein nicht privilegierter Benutzer dadurch letztlich zugreifen kann. Betrachten Sie im folgenden Beispiel, wie sich der Zugriff auf Host **X** (in Hostgruppe 1) in verschiedenen Situationen für einen Benutzer auswirkt, der sich in den Benutzergruppen A und B befindet.

- Wenn Gruppe A nur *Lesezugriff* auf Hostgruppe 1 hat, Gruppe B jedoch *Lese-/Schreibzugriff* auf Hostgruppe 1, erhält der Benutzer **Lese-/Schreibzugriff** auf „X“.

Attention:
Berechtigungen vom Typ „Lese-/Schreibzugriff“ haben Vorrang vor Berechtigungen vom Typ „Lesezugriff“.

- Im selben Szenario wie oben gilt: Wenn sich „X“ gleichzeitig auch in Hostgruppe 2 befindet, die für Gruppe A oder B **verweigert** ist, ist der Zugriff auf „X“ **nicht verfügbar**, trotz *Lese-/Schreibzugriff* auf Hostgruppe 1.
- Wenn für Gruppe A keine Berechtigungen definiert sind und Gruppe B *Lese-/Schreibzugriff* auf Hostgruppe 1 hat, erhält der Benutzer **Lese-/Schreibzugriff** auf „X“.
- Wenn Gruppe A *Verweigern*-Zugriff auf Hostgruppe 1 hat und Gruppe B *Lese-/Schreibzugriff* auf Hostgruppe 1 hat, wird dem Benutzer der Zugriff auf „X“ **verweigert**.

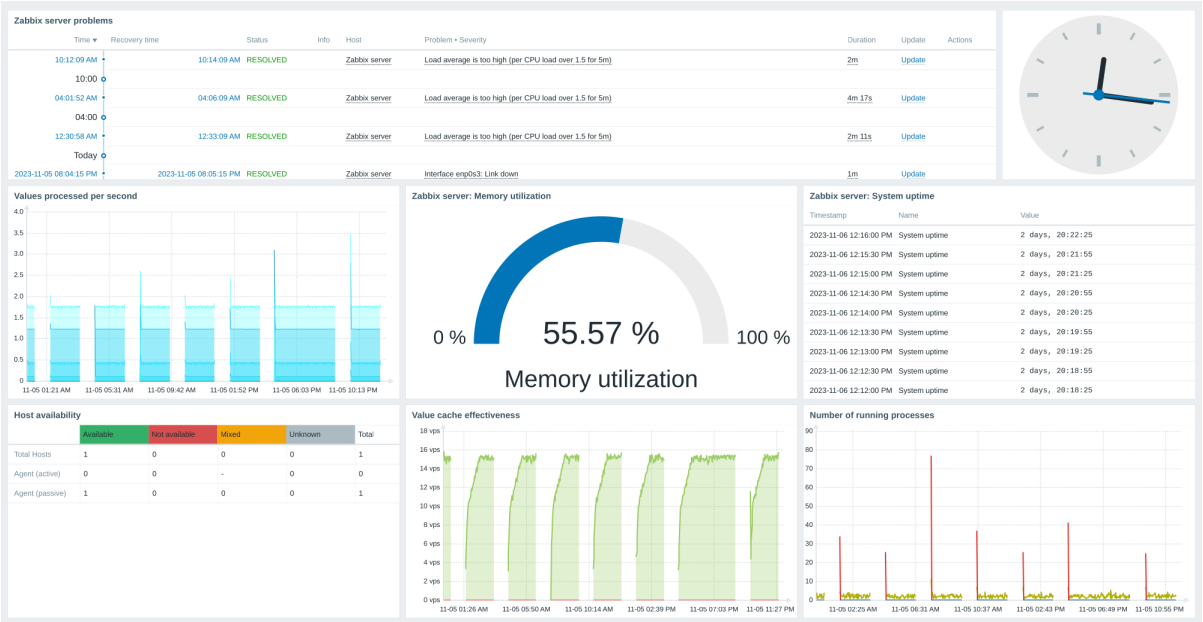
Weitere Details

- Ein Benutzer auf Admin-Ebene mit *Lese-/Schreibzugriff* auf einen Host kann Vorlagen nicht verknüpfen/entfernen, wenn er keinen Zugriff auf die Vorlagengruppe hat, zu der sie gehören. Mit *Lesezugriff* auf die Vorlagengruppe kann er Vorlagen mit dem Host verknüpfen/deren Verknüpfung aufheben, sieht jedoch keine Vorlagen in der Vorlagenliste und kann Vorlagen an anderen Stellen nicht verwenden.
- Ein Benutzer auf Admin-Ebene mit *Lesezugriff* auf einen Host sieht den Host nicht in der Host-Liste im Konfigurationsbereich; die Host-Auslöser sind jedoch in der IT-Service-Konfiguration zugänglich.
- Jeder Benutzer, der kein Super-Admin ist (einschließlich ’guest’), kann Netzwerkkarten sehen, solange die Karte leer ist oder nur Bilder enthält. Wenn Hosts, Host-Gruppen oder Auslöser zur Karte hinzugefügt werden, werden die Berechtigungen berücksichtigt.
- Der Zabbix Server sendet keine Benachrichtigungen an Benutzer, die als Empfänger von Aktionsoperationen definiert sind, wenn der Zugriff auf den betreffenden Host ausdrücklich auf ”verweigert” gesetzt ist.

Geplante Berichte

Übersicht

Mit der Funktion *Geplante Berichte* können Sie eine PDF-Version eines bestimmten Dashboards einrichten, die in regelmäßigen Abständen an festgelegte Empfänger gesendet wird.



Voraussetzungen:

- Der Zabbix-Webservice muss installiert und korrekt konfiguriert sein, damit die Generierung geplanter Berichte aktiviert wird – Anweisungen finden Sie unter Einrichten geplanter Berichte.
- Ein Benutzer muss über eine Benutzerrolle vom Typ *Admin* oder *Super-Admin* mit den folgenden Berechtigungen verfügen:
  - *Geplante Berichte* im Block *Zugriff auf UI-Elemente* (zum Anzeigen der Berichtseinstellungen)
  - *Geplante Berichte verwalten* im Block *Zugriff auf Aktionen* (zum Erstellen/Bearbeiten von Berichten)

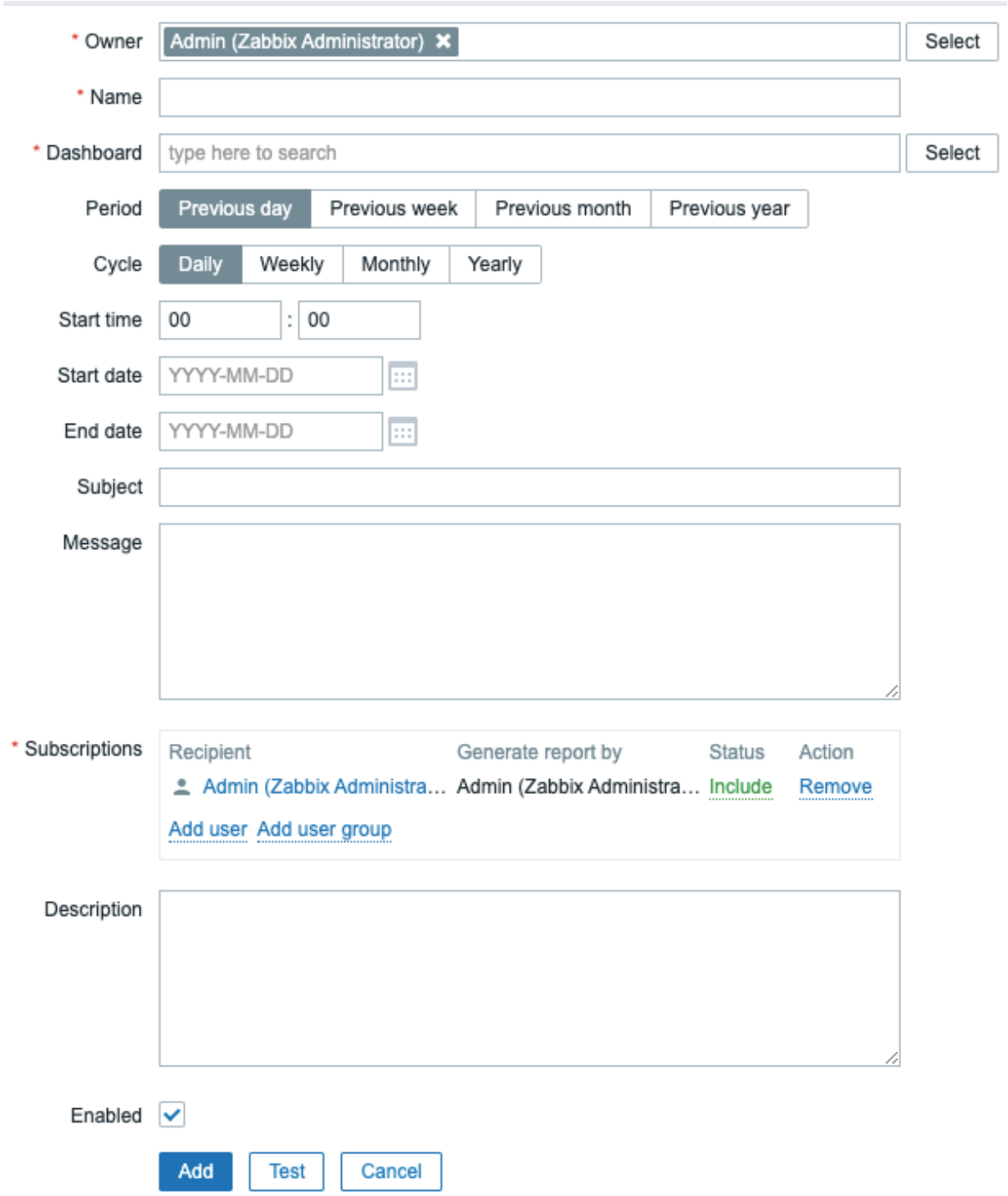
Gehen Sie wie folgt vor, um einen geplanten Bericht im Zabbix Frontend zu erstellen:

1. Gehen Sie zu *Berichte* > *Geplante Berichte*.
2. Klicken Sie oben rechts auf dem Bildschirm auf *Bericht erstellen*.
3. Geben Sie die Berichtsparameter im Formular ein.

Alternativ können Sie einen Bericht erstellen, indem Sie einen vorhandenen Bericht öffnen, auf die Schaltfläche *Klonen* klicken und ihn unter einem anderen Namen speichern.

Konfiguration

Die Registerkarte *Geplante Berichte* enthält allgemeine Berichtattribute.



Alle Pflichtfelder sind mit einem roten Sternchen gekennzeichnet.

Parameter	Beschreibung
Eigentümer	Benutzer, der einen Bericht erstellt. Benutzer mit der Berechtigungsstufe *Super-Admin* dürfen den Eigentümer ändern. Für Benutzer mit der Berechtigungsstufe *Admin* ist dieses Feld schreibgeschützt.
Name	Name des Berichts; muss eindeutig sein.
Dashboard	Dashboard, auf dem der Bericht basiert; es kann jeweils nur ein Dashboard ausgewählt werden. Um ein Dashboard auszuwählen, beginnen Sie mit der Eingabe des Namens - eine Liste übereinstimmender Dashboards wird angezeigt; scrollen Sie nach unten, um eines auszuwählen. Alternativ können Sie neben dem Feld auf *Auswählen* klicken und ein Dashboard aus der angezeigten Liste auswählen.
Zeitraum	Zeitraum, für den der Bericht erstellt wird. Wählen Sie den vorherigen Tag, die vorherige Woche, den vorherigen Monat oder das vorherige Jahr aus.
Zyklus	Häufigkeit der Berichtserstellung. Die Berichte können täglich, wöchentlich, monatlich oder jährlich gesendet werden. Im Modus „Wöchentlich“ können Sie die Wochentage auswählen, an denen der Bericht gesendet wird.

Parameter	Beschreibung
Startzeit	Tageszeit im Format hh:mm, zu der der Bericht erstellt wird. Beachten Sie, dass die Zeitzone des Zabbix-Servers verwendet wird.
Wiederholen an	Wochentage, an denen der Bericht gesendet wird. Dieses Feld ist nur verfügbar, wenn *Zyklus* auf „Wöchentlich“ gesetzt ist.
Startdatum	Datum, an dem die regelmäßige Berichtserstellung beginnen soll.
Enddatum	Datum, an dem die regelmäßige Berichtserstellung beendet werden soll.
Betreff	Betreff der Bericht-E-Mail. Unterstützte Makros: {TIME}, {TIMESTAMP}.
Nachricht	Text der Bericht-E-Mail. Unterstützte Makros: {TIME}, {TIMESTAMP}.
Abonnements	Liste der Berichtsempfänger. Standardmäßig ist nur der Eigentümer des Berichts enthalten. Jeder Zabbix-Benutzer mit konfiguriertem E-Mail-Medium kann als Berichtsempfänger angegeben werden. Klicken Sie auf *Benutzer hinzufügen* oder *Benutzergruppe hinzufügen*, um weitere Empfänger hinzuzufügen. Klicken Sie auf den Benutzernamen, um die Einstellungen zu bearbeiten: *Bericht erstellen anhand von* - legt fest, ob die Berichtsdaten anhand der Dashboard-Berechtigungen des aktuellen Benutzers oder des Empfängers erstellt werden sollen. *Status* - wählen Sie „Einschließen“, um den Bericht an den Benutzer zu senden, oder „Ausschließen“, um das Senden des Berichts an diesen Benutzer zu verhindern. Mindestens ein Benutzer muss den Status „Einschließen“ haben. Mit dem Status „Ausschließen“ können bestimmte Benutzer aus einer eingeschlossenen Benutzergruppe ausgeschlossen werden. Beachten Sie, dass Benutzer mit unzureichenden Berechtigungen (d. h. Benutzer mit einer auf dem Benutzertyp *Admin* basierenden Rolle, die nicht Mitglied derselben Benutzergruppe wie der Empfänger oder der Eigentümer des Berichts sind) in den Feldern *Empfänger* und *Bericht erstellen anhand von* anstelle der tatsächlichen Namen „Nicht zugänglicher Benutzer“ oder „Nicht zugängliche Benutzergruppe“ sehen. Die Felder *Status* und *Aktion* werden schreibgeschützt angezeigt.
Aktiviert	Berichtsstatus. Wenn Sie dieses Kontrollkästchen deaktivieren, wird der Bericht deaktiviert.
Beschreibung	Eine optionale Beschreibung des Berichts. Diese Beschreibung ist für die interne Verwendung bestimmt und wird nicht an Berichtsempfänger gesendet.

Formularschaltflächen

Mit den Schaltflächen am unteren Rand des Formulars können Sie mehrere Vorgänge ausführen.

Add	Einen Bericht hinzufügen. Diese Schaltfläche ist nur für neue Berichte verfügbar.
Update	Die Eigenschaften eines Berichts aktualisieren.
Clone	Einen weiteren Bericht auf Grundlage der Eigenschaften des aktuellen Berichts erstellen.
Test	Testen, ob die Berichtskonfiguration korrekt ist, indem ein Bericht an den aktuellen Benutzer gesendet wird.
Delete	Den Bericht löschen.
Cancel	Die Bearbeitung der Berichtseigenschaften abbrechen.

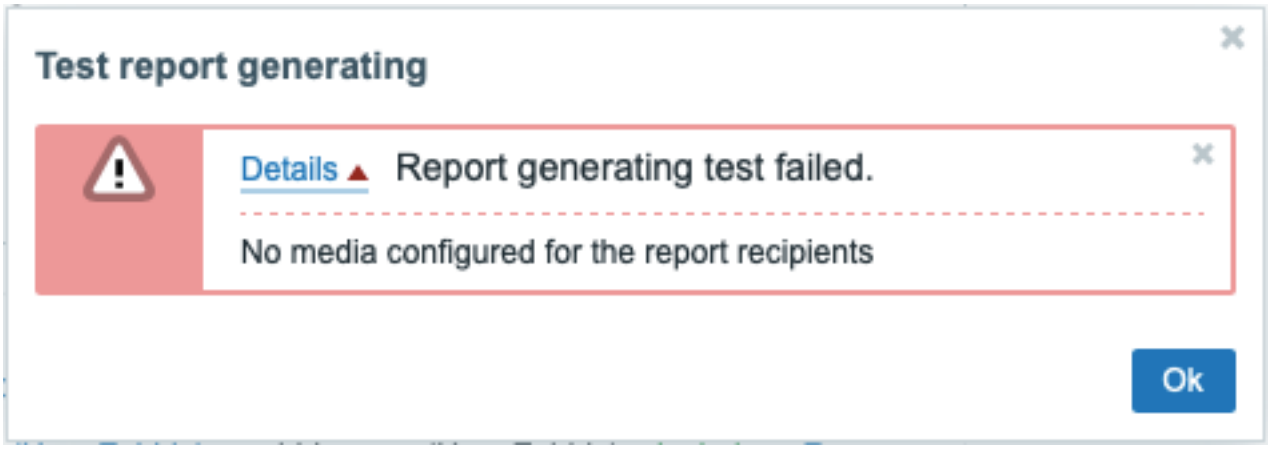
Testen

Um einen Bericht zu testen, klicken Sie unten im Berichtskonfigurationsformular auf die Schaltfläche *Test*.

Note:
Die Schaltfläche *Test* ist nicht verfügbar, wenn das Berichtskonfigurationsformular über das Aktionsmenü des Dashboards geöffnet wurde.

Wenn die Konfiguration korrekt ist, wird der Testbericht sofort an den aktuellen Benutzer gesendet. Bei Testberichten werden die Einstellungen für Abonnenten und den Benutzer *Generate report by* ignoriert.

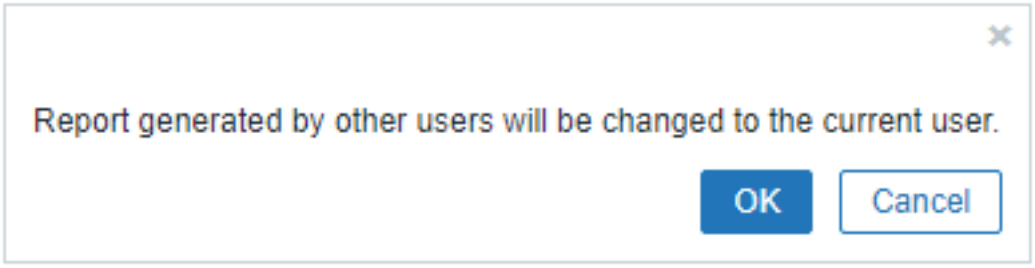
Wenn die Konfiguration fehlerhaft ist, wird eine Fehlermeldung angezeigt, die die mögliche Ursache beschreibt.



Einen Bericht aktualisieren

Um einen vorhandenen Bericht zu aktualisieren, klicken Sie auf den Berichtsnamen, nehmen Sie die erforderlichen Konfigurationsänderungen vor und klicken Sie dann auf die Schaltfläche *Update*.

Wenn ein vorhandener Bericht von einem anderen Benutzer aktualisiert wird und dieser Benutzer das Dashboard ändert, wird beim Klicken auf die Schaltfläche *Update* eine Warnmeldung mit dem Text „Von anderen Benutzern erzeugter Bericht wird auf den aktuellen Benutzer geändert“ angezeigt.



Ein Klick auf *OK* in diesem Schritt führt zu den folgenden Änderungen:

- Die Einstellungen für *Generate report by* werden aktualisiert, sodass der Benutzer angezeigt wird, der den Bericht zuletzt bearbeitet hat (es sei denn, *Generate report by* ist auf den Empfänger gesetzt).
- Benutzer, die als „Inaccessible user“ oder „Inaccessible user group“ angezeigt wurden, werden aus der Liste der Berichtsabonnenten gelöscht.

Ein Klick auf *Cancel* schließt das Konfigurationsformular und bricht die Aktualisierung des Berichts ab.

Klonen eines Berichts

Um einen vorhandenen Bericht schnell zu klonen, klicken Sie unten in einem vorhandenen Berichtskonfigurationsformular auf die Schaltfläche *Klonen*. Wenn ein von einem anderen Benutzer erstellter Bericht geklont wird, wird der aktuelle Benutzer zum Eigentümer des neuen Berichts.

Die Berichtseinstellungen werden unter Berücksichtigung der Benutzerberechtigungen in das neue Berichtskonfigurationsformular kopiert:

- Wenn der Benutzer, der einen Bericht klont, keine Berechtigungen für ein Dashboard hat, wird das Feld *Dashboard* geleert.
- Wenn der Benutzer, der einen Bericht klont, keine Berechtigungen für einige Benutzer oder Benutzergruppen in der Liste *Abonnements* hat, werden nicht zugängliche Empfänger nicht geklont.
- Die Einstellungen für *Bericht erstellen durch* werden aktualisiert, sodass der aktuelle Benutzer angezeigt wird (es sei denn, *Bericht erstellen durch* ist auf den Empfänger gesetzt).

Ändern Sie die erforderlichen Einstellungen und den Berichtsnamen und klicken Sie dann auf *Hinzufügen*.

Datenexport

Übersicht

Zabbix unterstützt den Datenexport in Echtzeit auf zwei Arten:

- Export in Dateien
- Streaming an externe Systeme

Die folgenden Entitäten können exportiert werden:

- Auslöser-Ereignisse
- Datenpunkt-Werte
- Trends (nur Export in Dateien)

Export in Dateien

Übersicht

Es ist möglich, den Echtzeit-Export von Auslöser-Ereignissen, Datenpunkt-Werten und Trends im newline-delimited JSON-Format zu konfigurieren.

Der Export erfolgt in Dateien, wobei jede Zeile der Exportdatei ein JSON-Objekt ist. Wertzuordnungen werden nicht angewendet.

Im Fehlerfall (Daten können nicht in die Exportdatei geschrieben werden, die Exportdatei kann nicht umbenannt werden oder nach dem Umbenennen kann keine neue Datei erstellt werden), wird der Datenpunkt verworfen und nie in die Exportdatei geschrieben. Er wird nur in der Zabbix-Datenbank gespeichert. Das Schreiben von Daten in die Exportdatei wird fortgesetzt, sobald das Schreibproblem behoben ist.

Die Exportdatei wird mit Lese- und Schreibberechtigungen nur für den Dateieigentümer erstellt. Zusätzlich ist die Datei für die Eigentümergruppe lesbar. Alle anderen Berechtigungen werden verweigert.

Beachten Sie, dass Host/Datenpunkt keine Metadaten (Hostgruppen, Hostname, Datenpunktname) haben kann, wenn der Host/Datenpunkt entfernt wurde, nachdem die Daten empfangen wurden, aber bevor der Server die Daten exportiert hat.

Konfiguration

Der Echtzeit-Export von Auslöser-Ereignissen, Datenpunktwerten und Trends wird konfiguriert, indem ein Verzeichnis für die Exportdateien angegeben wird – siehe den Parameter ExportDir in der Server-Konfiguration.

Zwei weitere Parameter sind verfügbar:

- Mit `ExportFileSize` kann die maximal zulässige Größe einer einzelnen Exportdatei festgelegt werden. Wenn ein Prozess in eine Datei schreiben muss, überprüft er zunächst deren Größe. Überschreitet sie das konfigurierte Größenlimit, wird die Datei durch Anhängen von `.old` an ihren Namen umbenannt und eine neue Datei mit dem ursprünglichen Namen erstellt.

Attention:
Für jeden Prozess, der Daten schreibt, wird eine Datei erstellt (d. h. ungefähr 4–30 Dateien). Da die Standardsgröße pro Exportdatei 1G beträgt, kann das Aufbewahren großer Exportdateien den Speicherplatz schnell aufbrauchen.

- Mit `ExportType` können Sie festlegen, welche Entitätstypen (Ereignisse, Verlauf, Trends) exportiert werden.

Newline-delimited JSON-Exportprotokoll

Dieser Abschnitt enthält Details zum Exportprotokoll in einem newline-delimited JSON-Format, das verwendet wird in:

- Datenexport in Dateien
- Streaming an externe Systeme

Alle Dateien haben die Erweiterung `.ndjson`. Jede Zeile der Exportdatei ist ein JSON-Objekt.

Auslöser-Ereignis-Export

Die folgenden Informationen werden für ein Problemereignis exportiert:

Field	Type	Description
clock	number	Anzahl der Sekunden seit der Unix-Epoche bis zum Zeitpunkt, an dem das Problem erkannt wurde (Ganzzahlteil).
ns	number	Anzahl der Nanosekunden, die zu `clock` addiert werden, um den genauen Zeitpunkt der Problemerkennung zu erhalten.
value	number	1 (immer).
eventid	number	ID des Problemereignisses.
name	string	Name des Problemereignisses.
severity	number	Schweregrad des Problemereignisses (0 - Nicht klassifiziert, 1 - Information, 2 - Warnung, 3 - Durchschnittlich, 4 - Hoch, 5 - Katastrophal).
hosts	array	Liste der Hosts, die an dem Ausdrück des Auslösers beteiligt sind; im Array muss mindestens ein Element vorhanden sein.

Field		Type	Description
-		object	
	host	string	Host-Name.
	name	string	Sichtbarer Host-Name.
groups		array	Liste der Host-Gruppen aller Hosts, die an dem Ausdrück des Auslösers beteiligt sind; im Array muss mindestens ein Element vorhanden sein.
-		string	Name der Host-Gruppe.
tags		array	Liste der Problem-Tags (kann leer sein).
-		object	
	tag	string	Tag-Name.
	value	string	Tag-Wert (kann leer sein).

Die folgenden Informationen werden für ein Wiederherstellungsereignis exportiert:

Field	Type	Description
clock	number	Anzahl der Sekunden seit der Unix-Epoche bis zum Zeitpunkt, an dem das Problem behoben wurde (Ganzzahlteil).
ns	number	Anzahl der Nanosekunden, die zu `clock` addiert werden, um den genauen Zeitpunkt der Problembehebung zu erhalten.
value	number	0 (immer).
eventid	number	ID des Wiederherstellungsereignisses.
p_eventid	number	ID des Problemereignisses.

Beispiele für den Export von Auslöser-Ereignissen

Problem:

```
{"clock":1519304285,"ns":123456789,"value":1,"name":"Either Zabbix agent is unreachable on Host B or polle
```

Wiederherstellung:

```
{"clock":1519304345,"ns":987654321,"value":0,"eventid":43,"p_eventid":42}
```

Problem (Generierung mehrerer Problemereignisse):

```
{"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
```

Wiederherstellung:

```
{"clock":1519304346,"ns":987654321,"value":0,"eventid":44,"p_eventid":43}

{"clock":1519304346,"ns":987654321,"value":0,"eventid":44,"p_eventid":42}
```

Item-Wert-Export

Die folgenden Informationen werden für einen erfassten Datenpunktwert exportiert:

Field		Type	Description
host		object	Host-Name des Datenpunkt-Hosts.
	host	string	Host-Name.
	name	string	Angezeigter Host-Name.
groups		array	Liste der Host-Gruppen des Datenpunkt-Hosts; das Array sollte mindestens ein Element enthalten.
	-	string	Name der Host-Gruppe.
item_tags		array	Liste der Datenpunkt-Tags (kann leer sein).
	-	object	
	tag	string	Tag-Name.
	value	string	Tag-Wert (kann leer sein).
itemid		number	Datenpunkt-ID.
name		string	Angezeigter Datenpunktname.

Field	Type	Description
clock	number	Anzahl der Sekunden seit der Unix-Epoche zum Zeitpunkt der Wert-Erfassung (Ganzzahlteil).
ns	number	Anzahl der Nanosekunden, die zu `clock` addiert werden, um den genauen Zeitpunkt der Wert-Erfassung zu erhalten.
timestamp (*Log* only)	number	0, falls nicht verfügbar.
source (*Log* only)	string	Leerer String, falls nicht verfügbar.
severity (*Log* only)	number	0, falls nicht verfügbar.
eventid (*Log* only)	number	0, falls nicht verfügbar.
value	number (for numeric items) or string (for text items)	Erfasster Datenpunktwert.
type	number	Typ des erfassten Werts: 0 - numerisch float, 1 - Zeichen, 2 - Log, 3 - numerisch unsigned, 4 - Text, 5 - binär, 6 - JSON

Beispiel für den Export von Datenpunktwerten

Numerischer Wert (vorzeichenlos):

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Numerischer Wert (Gleitkomma):

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Zeichen- und Textwert:

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Logwert:

```
{"host":{"host":"Host A","name":"Host A visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Trends-Export

Die folgenden Informationen werden für einen berechneten Trendwert exportiert:

Field	Type	Description
host	object	Host-Name des Hosts des Datenpunkts.
host	string	Host-Name.
name	string	Angezeigter Host-Name.
groups	array	Liste der Host-Gruppen des Hosts des Datenpunkts; das Array sollte mindestens ein Element enthalten.
-	string	Name der Host-Gruppe.
item_tags	array	Liste der Datenpunkt-Tags (kann leer sein).
-	object	
tag	string	Tag-Name.
value	string	Tag-Wert (kann leer sein).
itemid	number	Datenpunkt-ID.
name	string	Angezeigter Datenpunkt-Name.
clock	number	Anzahl der Sekunden seit der Unix-Epoche zum Zeitpunkt der Erfassung des Werts (Ganzzahlteil).
count	number	Anzahl der für eine bestimmte Stunde erfassten Werte.
min	number	Minimaler Datenpunktwert für eine bestimmte Stunde.
avg	number	Durchschnittlicher Datenpunktwert für eine bestimmte Stunde.
max	number	Maximaler Datenpunktwert für eine bestimmte Stunde.

Field	Type	Description
type	number	Werttyp: 0 - numerisch Gleitkomma, 3 - numerisch ohne Vorzeichen

Beispiele für den Export von Trends

Numerischer Wert (unsigned):

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Numerischer Wert (float):

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Streaming an externe Systeme

Übersicht

Es ist möglich, Datenpunktwerte und Ereignisse von Zabbix über HTTP an externe Systeme zu streamen (siehe Protokolldetails).

Der Tag-Filter kann verwendet werden, um Teilmengen von Datenpunktwerten oder Ereignissen zu streamen.

Zwei Zabbix-Serverprozesstypen sind für das Datenstreaming verantwortlich: `connector manager` und `connector worker`. Mit dem internen Zabbix-Datenpunkt `zabbix[connector_queue]` können Sie die Anzahl der in die Connector-Warteschlange eingereihten Werte überwachen.

Konfiguration

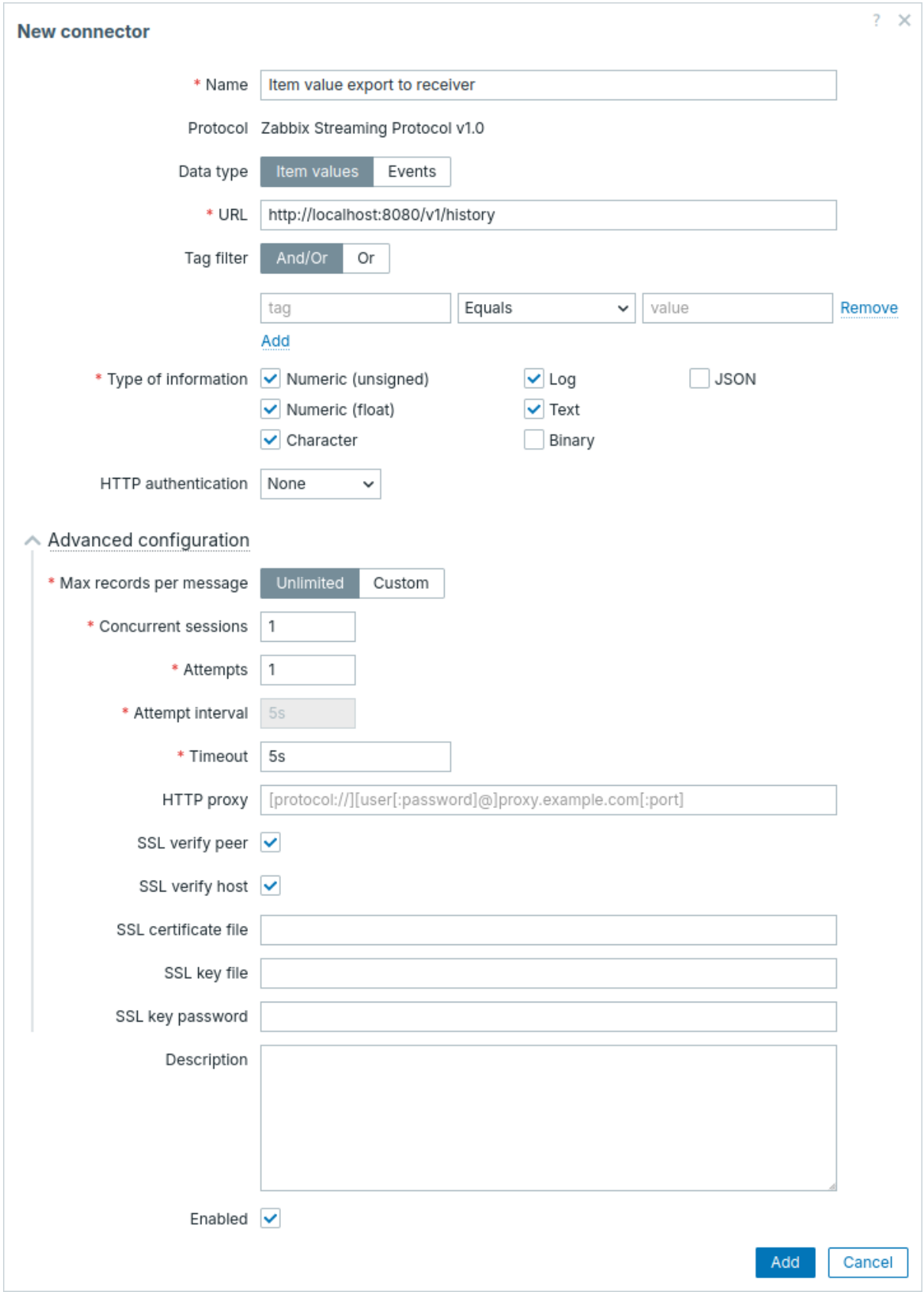
Die folgenden Schritte sind erforderlich, um das Streamen von Daten an ein externes System zu konfigurieren:

1. Richten Sie ein Remotesystem für den Empfang von Daten von Zabbix ein. Zu diesem Zweck stehen die folgenden Tools zur Verfügung:

- Ein Beispiel für einen einfachen Empfänger, der die empfangenen Informationen in den Dateien `events.ndjson` und `history.ndjson` protokolliert.
- Kafka-Connector für den Zabbix-Server – ein schlanker, in Go geschriebener Server, der dafür ausgelegt ist, Datenpunktwerte und Ereignisse von einem Zabbix-Server an einen Kafka-Broker weiterzuleiten.

2. Legen Sie die erforderliche Anzahl von Connector-Workern in Zabbix fest, indem Sie den Parameter `StartConnectors` in `zabbix_server.conf` anpassen. Die Anzahl der Connector-Worker sollte der konfigurierten Connector-Anzahl im Zabbix-Frontend entsprechen (oder diese überschreiten, wenn mehr als eine gleichzeitige Sitzung konfiguriert ist). Starten Sie anschließend den Zabbix-Server neu.

3. Konfigurieren Sie einen neuen Connector im Zabbix-Frontend (*Administration* > *General* > *Connectors*) und laden Sie den Server-Cache mit dem Befehl `zabbix_server -R config_cache_reload` neu.



Pflichtfelder sind mit einem Sternchen gekennzeichnet.

Parameter	Beschreibung
Name	Geben Sie den Namen des Connectors ein.
Data type	Wählen Sie den zu streamenden Datentyp aus: **Item values** – streamt Datenpunktwerte von Zabbix an externe Systeme; **Events** – streamt Ereignisse von Zabbix an externe Systeme.
URL	Geben Sie die URL des Empfängers ein. Benutzermakros werden unterstützt.

Parameter	Beschreibung
Tag filter	Exportiert nur Datenpunktwerte oder Ereignisse, die dem Tag-Filter entsprechen. Wenn kein Filter festgelegt ist, wird alles exportiert. Bestimmte Tags und Tagwerte können eingeschlossen oder ausgeschlossen werden. Es können mehrere Bedingungen festgelegt werden. Beim Abgleich von Tagnamen wird immer zwischen Groß- und Kleinschreibung unterschieden. Für jede Bedingung stehen mehrere Operatoren zur Verfügung: **Exists** – schließt die angegebenen Tagnamen ein; **Equals** – schließt die angegebenen Tagnamen und -werte ein (Groß-/Kleinschreibung wird berücksichtigt); **Contains** – schließt die angegebenen Tagnamen ein, deren Tagwerte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich ohne Berücksichtigung der Groß-/Kleinschreibung); **Does not exist** – schließt die angegebenen Tagnamen aus; **Does not equal** – schließt die angegebenen Tagnamen und -werte aus (Groß-/Kleinschreibung wird berücksichtigt); **Does not contain** – schließt die angegebenen Tagnamen aus, deren Tagwerte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich ohne Berücksichtigung der Groß-/Kleinschreibung). Für Bedingungen stehen zwei Berechnungstypen zur Verfügung: **And/Or** – alle Bedingungen müssen erfüllt sein; Bedingungen mit demselben Tagnamen werden mit der Or-Bedingung gruppiert; **Or** – es genügt, wenn eine Bedingung erfüllt ist.
Type of information	Wählen Sie den Informationstyp (numerisch (unsigned), numerisch (Gleitkomma), Zeichen usw.) aus, nach dem die Datenpunktwerte gefiltert werden sollen, die der Connector streamen soll. Dieses Feld ist verfügbar, wenn *Data type* auf „Item values“ gesetzt ist.
HTTP authentication	Wählen Sie die Authentifizierungsoption aus: **None** – keine Authentifizierung wird verwendet; **Basic** – die Basisauthentifizierung wird verwendet; **NTLM** – die NTLM-Authentifizierung (Windows NT LAN Manager) wird verwendet; **Kerberos** – die Kerberos-Authentifizierung wird verwendet (siehe auch: Kerberos mit Zabbix konfigurieren); **Digest** – die Digest-Authentifizierung wird verwendet; **Bearer** – die Bearer-Authentifizierung wird verwendet.
Username	Geben Sie den Benutzernamen ein (bis zu 255 Zeichen). Benutzermakros werden unterstützt. Dieses Feld ist verfügbar, wenn *HTTP authentication* auf „Basic“, „NTLM“, „Kerberos“ oder „Digest“ gesetzt ist.
Password	Geben Sie das Benutzerpasswort ein (bis zu 255 Zeichen). Benutzermakros werden unterstützt. Dieses Feld ist verfügbar, wenn *HTTP authentication* auf „Basic“, „NTLM“, „Kerberos“ oder „Digest“ gesetzt ist.
Bearer token	Geben Sie das Bearer-Token ein. Benutzermakros werden unterstützt. Dieses Feld ist verfügbar und erforderlich, wenn *HTTP authentication* auf „Bearer“ gesetzt ist.
Advanced configuration	Klicken Sie auf die Überschrift *Advanced configuration*, um die erweiterten Konfigurationsoptionen anzuzeigen (siehe unten).
Max records per message	Geben Sie die maximale Anzahl von Werten oder Ereignissen an, die innerhalb einer Nachricht gestreamt werden können.
Concurrent sessions	Wählen Sie die Anzahl der Senderprozesse aus, die für diesen Connector ausgeführt werden sollen. Es können bis zu 100 Sitzungen angegeben werden; der Standardwert ist „1“.
Attempts	Anzahl der Versuche zum Streamen von Daten. Es können bis zu 5 Versuche angegeben werden; der Standardwert ist „1“.
Attempt interval	Geben Sie an, wie lange der Connector nach einem erfolglosen Versuch, Daten zu streamen, warten soll. Es können bis zu 10 s angegeben werden; der Standardwert ist „5 s“. Dieses Feld ist verfügbar, wenn *Attempts* auf „2“ oder höher gesetzt ist. Als erfolglos gelten Versuche, bei denen der Verbindungsaufbau fehlgeschlagen ist oder der HTTP-Antwortcode nicht 200, 201, 202, 203 oder 204 lautet. Wiederholungen werden bei Kommunikationsfehlern oder wenn der HTTP-Antwortcode nicht 200, 201, 202, 203, 204, 400, 401, 403, 404, 405, 415 oder 422 lautet, **ausgelöst**. Weiterleitungen werden verfolgt, daher ist 302 -> 200 eine positive Antwort, während 302 -> 503 eine Wiederholung auslöst.
Timeout	Geben Sie das Nachrichten-Timeout an (1–60 Sekunden, Standardwert: 5 Sekunden). Zeit-Suffixe werden unterstützt (z. B. `30s`, `1m`). Benutzermakros werden unterstützt.

Parameter	Beschreibung
HTTP proxy	Sie können einen HTTP-Proxy im folgenden Format angeben: `[protocol://][username[:password]@]proxy.example.com[:port]` Benutzermakros werden unterstützt. Das optionale Präfix `protocol://` kann verwendet werden, um alternative Proxy-Protokolle anzugeben (die Unterstützung für das Protokollpräfix wurde in cURL 7.21.7 hinzugefügt). Wenn kein Protokoll angegeben ist, wird der Proxy als HTTP-Proxy behandelt. Standardmäßig wird Port 1080 verwendet. Wenn *HTTP proxy* angegeben ist, überschreibt der Proxy proxybezogene Umgebungsvariablen wie `http_proxy` und `HTTPS_PROXY`. Wenn kein Proxy angegeben ist, werden proxybezogene Umgebungsvariablen nicht überschrieben. Der eingegebene Wert wird unverändert übernommen; es findet keine Plausibilitätsprüfung statt. Sie können auch eine SOCKS-Proxyadresse eingeben. Wenn Sie das falsche Protokoll angeben, kann der Connector keine Datenpunktwerte oder Ereignisse von Zabbix streamen. Beachten Sie, dass mit einem HTTP-Proxy nur eine einfache Authentifizierung unterstützt wird.
SSL verify peer	Aktivieren Sie das Kontrollkästchen, um das SSL-Zertifikat des Webservers zu überprüfen. Das Serverzertifikat wird anhand der Zertifikate vertrauenswürdiger Zertifizierungsstellen (CA) am systemweiten CA-Speicherort überprüft. Sie können den Speicherort der CA-Dateien mit dem Zabbix-Server- oder Proxy-Konfigurationsparameter `SSLCALocation` überschreiben.
SSL verify host	Aktivieren Sie das Kontrollkästchen, um zu überprüfen, ob das Feld *Common Name* oder das Feld *Subject Alternate Name* des Webserverzertifikats übereinstimmt. Dadurch wird die cURL-Option `CURLOPT_SSL_VERIFYHOST` gesetzt.
SSL certificate file	Name der SSL-Zertifikatsdatei, die für die Clientauthentifizierung verwendet wird. Die Zertifikatsdatei muss im PEM[1]-Format vorliegen. Benutzermakros werden unterstützt. Wenn die Zertifikatsdatei auch den privaten Schlüssel enthält, lassen Sie das Feld *SSL key file* leer. Wenn der Schlüssel verschlüsselt ist, geben Sie das Passwort im Feld *SSL key password* an. Das Verzeichnis, das diese Datei enthält, wird durch den Zabbix-Server- oder Proxy-Konfigurationsparameter `SSLCertLocation` angegeben.
SSL key file	Name der privaten SSL-Schlüsseldatei, die für die Clientauthentifizierung verwendet wird. Die Datei mit dem privaten Schlüssel muss im PEM[1]-Format vorliegen. Benutzermakros werden unterstützt. Das Verzeichnis, das diese Datei enthält, wird durch den Zabbix-Server- oder Proxy-Konfigurationsparameter `SSLKeyLocation` angegeben.
SSL key password	Passwort der privaten SSL-Schlüsseldatei. Benutzermakros werden unterstützt.
Description	Geben Sie die Beschreibung des Connectors ein.
Enabled	Aktivieren Sie das Kontrollkästchen, um den Connector zu aktivieren.

Wenn der Kafka-Connector mit einer durch Kommas getrennten Liste von Bootstrap-Brokeradressen konfiguriert ist (zum Beispiel `Kafka.URL=kafka1.example.com:9093,kafka2.example.com:9093`), stellt der Kafka-Client eine Verbindung zu dem oder den Brokern her, die zuerst antworten, und verwendet deren Cluster-Metadaten. Wenn die Liste Adressen aus verschiedenen Kafka-Clustern enthält, wird nur der am schnellsten antwortende Cluster verwendet und die anderen Adressen werden als nicht verfügbar protokolliert. Daher können beim Start Warnungen wie die folgende erscheinen, obwohl der Connector verbunden ist:

```
kafka cluster connected, but broker(s) "kafka1.example.com:9093, kafka2.example.com:9093" unavailable; wil
```

In einigen Umgebungen (privaten Netzwerken, Container-Netzwerken oder bei nicht standardmäßigen DNS-/Hosts-Konfigurationen) können Hostnamen oder IP-Adressen in Loopback-Adressen (zum Beispiel `127.0.0.1/localhost`) aufgelöst oder vom Client normalisiert werden, wodurch solche Warnungen irreführend sein können. Um Verwirrung zu vermeiden, stellen Sie sicher, dass alle `Kafka.URL`-Adressen zum selben Kafka-Cluster gehören, überprüfen Sie die DNS-Auflösung vom Connector-Host sowie die `advertised.listeners` der Broker und bevorzugen Sie Adressen, die in die vom Broker angekündigte Adresse aufgelöst werden.

Protokoll

Die Kommunikation zwischen dem Server und dem Receiver erfolgt über HTTP mithilfe der REST-API, NDJSON und "Content-Type: application/x-ndjson".

Weitere Informationen finden Sie unter Newline-delimited JSON export protocol.

Server-Anfrage

Beispiel für das Streamen von Datenpunkt-Werten:

```
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
```

Beispiel für das Streamen von Ereignissen:

```
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}
```

Antwort des Empfängers

Die Antwort besteht aus dem HTTP-Antwortstatuscode und der JSON-Nutzlast. Der HTTP-Antwortstatuscode muss für erfolgreich verarbeitete Anfragen ”200”, ”201”, ”202”, ”203” oder ”204” sein, andernfalls gilt die Anfrage als fehlgeschlagen.

Beispiel für einen Erfolg:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Content-Type-Options: nosniff
Date: Tue, 21 Apr 2026 10:13:04 GMT
Content-Length: 23

{"response":"success"}
```

Beispiel mit Fehlern:

```
HTTP/1.1 422 Unprocessable Entity
Content-Type: application/json
X-Content-Type-Options: nosniff
Date: Tue, 21 Apr 2026 12:15:01 GMT
Content-Length: 55

{"error":"invalid character '{' after top-level value"}
```

SNMP gateway

Übersicht

Das Zabbix SNMP gateway ist eine AgentX-Erweiterung für snmpd und unterstützt sowohl SNMP-Abfragen als auch Traps.

Mit dem Zabbix SNMP gateway ist es möglich, das SNMP-Protokoll zu verwenden, um Folgendes abzurufen:

- Auslöser-Daten;
- Daten zu Problem-Auslösern;
- Status von Host-Gruppen (Anzahl der Auslöser nach Auslöserstatus pro Gruppe)

Die Daten werden über die OID abgerufen, die aus einer gemeinsamen Basis und einem spezifischen Suffix besteht. Die gemeinsame **Basis** wird in der Konfigurationsdatei des SNMP gateway festgelegt, zum Beispiel:

- `BaseOID=1.3.6.1.4.1.3043.7.55` - für beliebige Auslöser-Daten;
- `ProblemBaseOID=1.3.6.1.4.1.3047.7.55` - für Daten zu Problem-Auslösern;
- `BaseOID=1.3.6.1.4.1.3046.7.55` - für den Status von Host-Gruppen.

Das OID-**Suffix** wird in der Konfiguration der Host-Auslöser als **Tag** (zum Beispiel `OIDSuffix:3`) im Frontend festgelegt.

In diesem Fall sind alle Informationen für den Auslöser unter `OID=1.3.6.1.4.1.3043.7.55.X.3` verfügbar. „X“ ist hier die Nummer der Auslöser-Datenfelder (d. h. 1 - Suffix, 2 - ID, 3 - Ausdruck, 4 - Beschreibung usw.).

Eine ausführlichere Beschreibung und ein Beispiel für die Konfigurationsdatei finden Sie in der SNMP gateway readme file.

Installation und Einrichtung

Anweisungen zu folgenden Themen finden Sie in der Datei readme des SNMP gateway:

- Installation und Konfiguration von snmpd;
- Aktivierung der AgentX-Unterstützung;
- Konfiguration des Zabbix SNMP gateway;
- Konfiguration von SNMP-Traps für Änderungen des Auslöserstatus.

Daten abrufen

Wenn alles korrekt eingerichtet ist, können Sie die Befehle `snmpwalk` und `snmpget` verwenden, um Daten abzurufen:

```
[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
...
```

Filteroptionen

Sie können die Informationen zu Problem-Auslösern in der SNMP-Gateway-Konfiguration einschränken:

- nach Schweregrad (standardmäßig `ProblemMinSeverity=-1`)
- durch Ausblenden bestätigter Probleme (standardmäßig `ProblemHideAck=false`)

Sie können die Problemanzahl pro Hostgruppe in der SNMP-Gateway-Konfiguration einschränken:

- nach Auslösern im unbekannten Zustand (standardmäßig `CountUnknown=false`)
- nach Auslösern mit bestätigten/unbestätigten/allen Problemen (standardmäßig `CountAcknowledgeStatus=all`)

Konfigurations-Export/Import

Übersicht

Die Export-/Import-Funktionalität von Zabbix ermöglicht den Austausch verschiedener Konfigurationselemente zwischen einem Zabbix-System und einem anderen.

Typische Anwendungsfälle für diese Funktionalität:

- Vorlagen oder Netzwerkkarten teilen - Zabbix-Benutzer können ihre Konfigurationsparameter weitergeben
- eine Vorlage in Zabbix Community templates hochladen. Anschließend können andere die Vorlage herunterladen und die Datei in Zabbix importieren.
- Integration mit Drittanbieter-Tools - universelle YAML-, XML- und JSON- Formate ermöglichen die Integration sowie den Datenimport/-export mit Drittanbieter- Tools und Anwendungen

Was kann exportiert/importiert werden

Objekte, die exportiert/importiert werden können, sind:

- Host groups (*nur über die Zabbix API*)
- Template groups (*nur über die Zabbix API*)
- Vorlagen

- Hosts
- Netzwerkkarten
- Medientypen
- Bilder
- Dashboards
- Reguläre Ausdrücke

Exportformat

Daten können über das Zabbix-Frontend oder die Zabbix API exportiert werden. Unterstützte Exportformate sind YAML, XML und JSON.

Details zum Export

- Alle unterstützten Elemente werden in einer Datei exportiert.
- Host- und Vorlagen-Entitäten (Datenpunkte, Auslöser, Diagramme, Erkennungsregeln), die von verknüpften Vorlagen geerbt wurden, werden nicht exportiert. Alle Änderungen, die an diesen Entitäten auf Host-Ebene vorgenommen wurden (z. B. geändertes Datenpunkt-Intervall, geänderte reguläre Ausdrücke oder hinzugefügte Prototypen zur Low-Level-Erkennungsregel), gehen beim Export verloren; beim Import werden alle Entitäten aus verknüpften Vorlagen so neu erstellt wie in der ursprünglich verknüpften Vorlage.
- Durch Low-Level-Erkennung erstellte Entitäten und alle von ihnen abhängigen Entitäten werden nicht exportiert. Beispielsweise wird ein Auslöser, der für einen von einer LLD-Regel generierten Datenpunkt erstellt wurde, nicht exportiert.
- Wenn der exportierte Host/die exportierte Vorlage Entitäten enthält, die Timeouts unterstützen, werden die Timeout-Werte exportiert, sofern für diese Entitäten eigene Timeouts konfiguriert sind.

Details zum Import

- Der Import wird beim ersten Fehler abgebrochen.
- Beim Aktualisieren vorhandener Bilder während des Bildimports wird das Feld ”imagetype” ignoriert, d. h. es ist nicht möglich, den Bildtyp per Import zu ändern.
- Beim Import von Hosts/Vorlagen mit der Option ”Fehlende löschen” werden Host-/Vorlagen-Makros, die in der Importdatei nicht vorhanden sind, nach dem Import vom Host/Vorlage gelöscht.
- Leere Tags für Datenpunkte, Auslöser, Diagramme, discoveryRules, itemPrototypes, triggerPrototypes, graphPrototypes sind bedeutungslos, d. h. es ist dasselbe, als wären sie nicht vorhanden.
- Wenn Entitäten des importierten Hosts/Vorlage eigene konfigurierte Timeouts haben, werden diese angewendet; andernfalls werden Proxy-/globale Timeouts angewendet.
- Der Import unterstützt YAML, XML und JSON. Die Importdatei muss die korrekte Dateiendung haben: .yaml und .yml für YAML, .xml für XML und .json für JSON. Siehe Kompatibilitätsinformationen zu den unterstützten XML-Versionen.
- Der Import unterstützt Konfigurationsdateien nur in UTF-8-Kodierung (mit oder ohne BOM); andere Kodierungen (UTF16LE, UTF16BE, UTF32LE, UTF32BE usw.) führen zu einem Fehler bei der Importkonvertierung.

YAML-Basisformat

Das YAML-Exportformat enthält die folgenden Knoten:

- Wurzelknoten für den Zabbix-YAML-Export
- Exportversion

```
zabbix_export:
  version: '8.0'
```

Weitere Knoten hängen von den exportierten Objekten ab.

XML-Format

Das XML-Exportformat enthält die folgenden Tags:

- Standard-Header für XML-Dokumente
- Root-Tag für den Zabbix-XML-Export
- Exportversion

```
<?xml version="1.0" encoding="UTF-8"?>
<zabbix_export>
    <version>8.0</version>
</zabbix_export>
```

Andere Tags hängen von den exportierten Objekten ab.

JSON-Format

Das JSON-Exportformat enthält die folgenden Objekte:

- Wurzelobjekt für den Zabbix-JSON-Export
- Exportversion

```
{
    "zabbix_export": {
        "version": "8.0"
    }
}
```

Andere Objekte sind von den exportierten Objekten abhängig.

Vorlagengruppen

Übersicht

Im Frontend können Vorlagengruppen nur zusammen mit dem Vorlagenexport exportiert werden. Wenn eine Vorlage exportiert wird, werden alle Gruppen, denen sie angehört, automatisch mit exportiert.

Die API ermöglicht den unabhängigen Export von Vorlagengruppen ohne Vorlagen.

Exportformat

```
template_groups:
  - uuid: 36bff6c29af64692839d077febfc7079
    name: 'Network devices'
```

Exportierte Elemente

Element	Type	Description
uuid	string	Eindeutige Kennung für diese Vorlagengruppe.
name	string	Gruppenname.

Host groups

Übersicht

Im Frontend können Host groups nur zusammen mit dem Host-Export exportiert werden. Wenn ein Host exportiert wird, werden alle Gruppen, denen er angehört, automatisch mit exportiert.

Die API ermöglicht den unabhängigen Export von Host groups, getrennt von Hosts.

Exportformat

```
host_groups:
  - uuid: 6f6799aa69e844b4b3918f779f2abf08
    name: 'Zabbix servers'
```

Exportierte Elemente

Element	Type	Description
uuid	string	Eindeutige Kennung für diese Hostgruppe.
name	string	Gruppenname.

Vorlagen

Übersicht

Vorlagen werden zusammen mit vielen zugehörigen Objekten und Objektbeziehungen exportiert.

Der Vorlagenexport enthält:

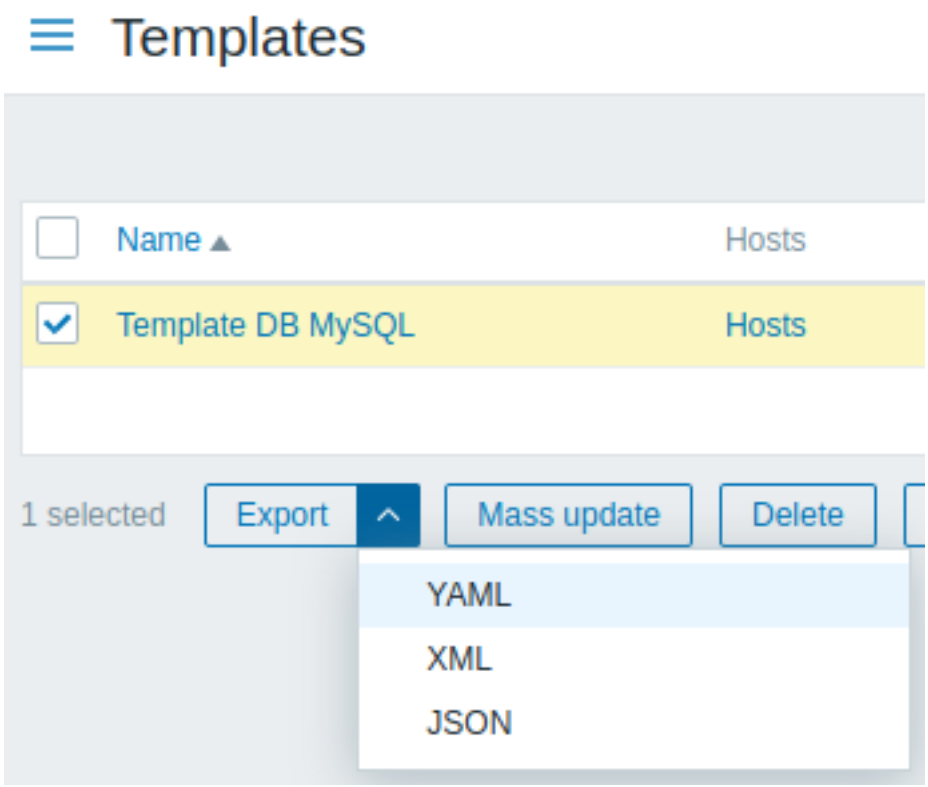
- Verknüpfte Vorlagengruppen
- Verknüpfte Host-Gruppen (falls in der Konfiguration des Host-Prototyps verwendet)
- Vorlagendaten
- Verknüpfungen zu anderen Vorlagen
- Verknüpfungen zu Vorlagengruppen

- Direkt verknüpfte Datenpunkte
- Direkt verknüpfte Auslöser
- Direkt verknüpfte Diagramme
- Direkt verknüpfte Dashboards
- Direkt verknüpfte Erkennungsregeln mit allen Prototypen
- Direkt verknüpfte Webszenarien
- Wertzuordnungen

Exportieren

Gehen Sie zum Exportieren von Vorlagen wie folgt vor:

1. Wechseln Sie zu *Datenerfassung* > *Vorlagen*.
2. Markieren Sie die Kontrollkästchen der zu exportierenden Vorlagen.
3. Klicken Sie unterhalb der Liste auf *Export*, um im standardmäßigen YAML-Format zu exportieren, oder klicken Sie auf den Pfeil, um ein anderes Exportformat auszuwählen.



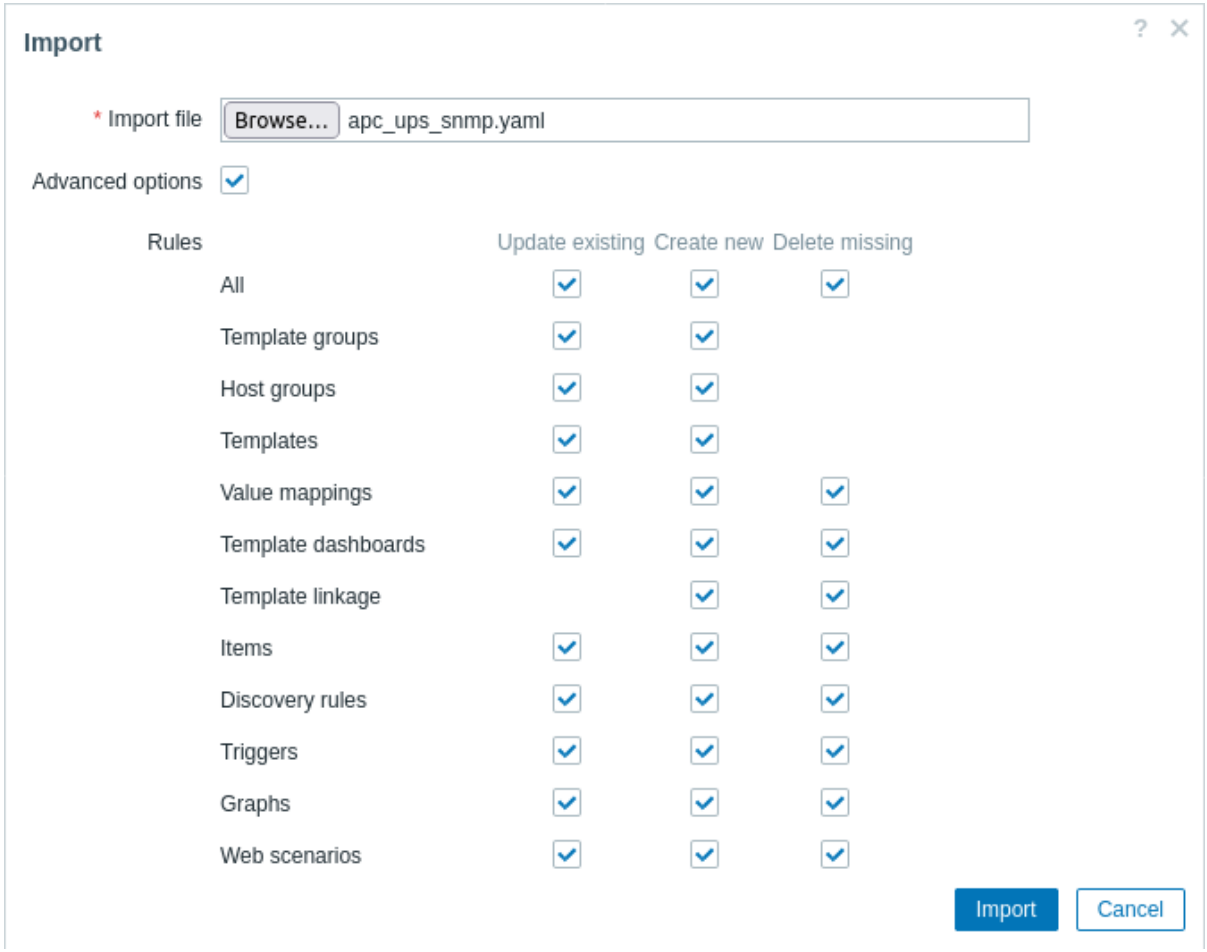
Abhängig vom ausgewählten Format werden Vorlagen in eine lokale Datei mit einem Standardnamen exportiert:

- `zabbix_export_templates.yaml` - beim YAML-Export;
- `zabbix_export_templates.xml` - beim XML-Export;
- `zabbix_export_templates.json` - beim JSON-Export.

Importieren

Gehen Sie zum Importieren von Vorlagen wie folgt vor:

1. Gehen Sie zu *Datensammlung* > *Vorlagen*.
2. Klicken Sie oben rechts auf *Importieren*.
3. Wählen Sie die Importdatei aus.
4. Aktivieren Sie die Kontrollkästchen der Importregeln, die Sie anwenden möchten (siehe unten).
5. Klicken Sie unten rechts im Konfigurationsformular auf *Importieren*.



Wenn Sie das Kontrollkästchen *Erweiterte Optionen* aktivieren, wird eine detaillierte Liste aller importierbaren Elemente angezeigt. Aktivieren oder deaktivieren Sie die einzelnen Importregeln nach Bedarf.

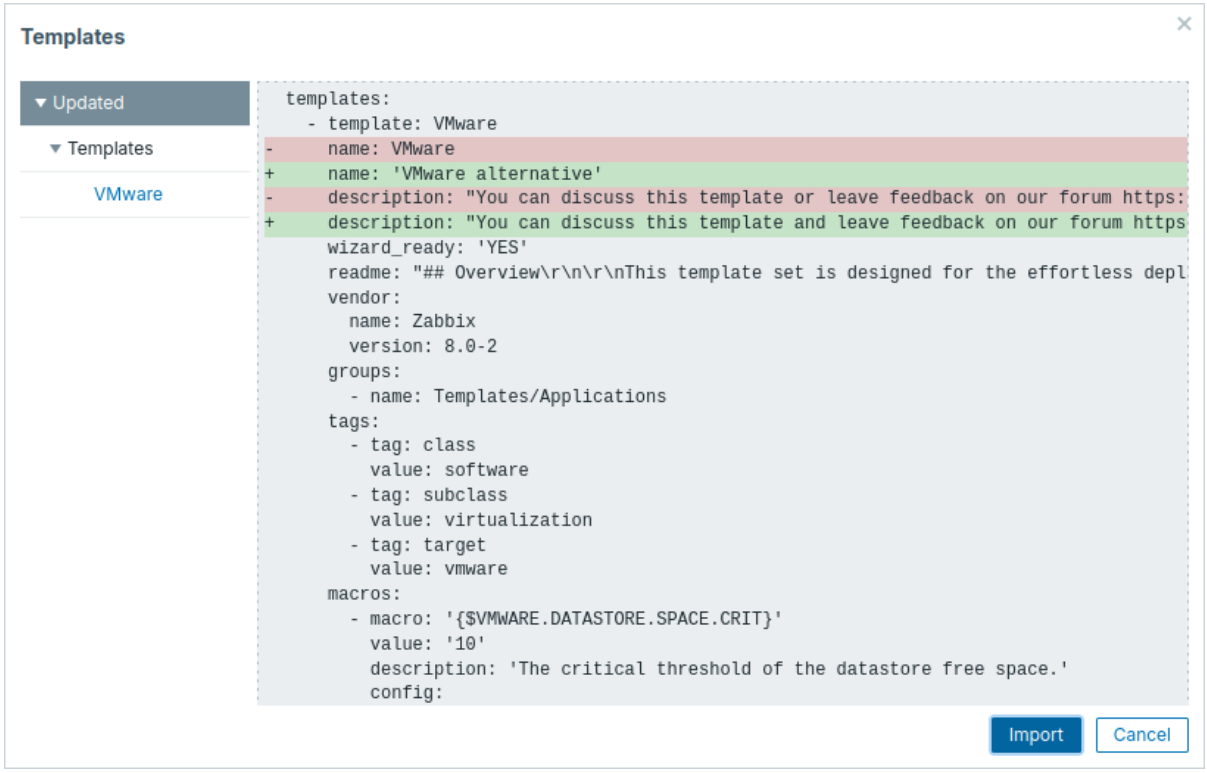
Wenn Sie das Kontrollkästchen in der Zeile *Alle* aktivieren, werden alle darunter aufgeführten Elemente aktiviert bzw. deaktiviert.

Importregeln:

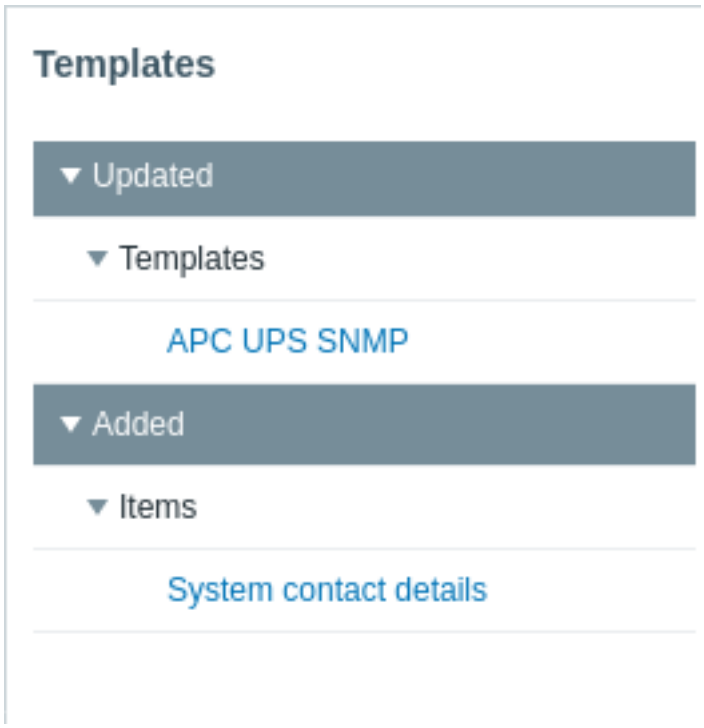
Regel	Beschreibung
Vorhandene aktualisieren	Vorhandene Elemente werden anhand der Daten aus der Importdatei aktualisiert. Andernfalls werden sie nicht aktualisiert.
Neue erstellen	Neue Elemente werden anhand der Daten aus der Importdatei erstellt. Andernfalls werden sie nicht erstellt.
Fehlende löschen	Vorhandene Elemente, die nicht in der Importdatei enthalten sind, werden entfernt. Andernfalls werden sie nicht entfernt. Wenn *Fehlende löschen* für *Vorlagenverknüpfung* aktiviert ist, werden aktuelle Vorlagenverknüpfungen, die nicht in der Importdatei enthalten sind, aufgehoben. Von den aufgehobenen Vorlagen geerbte Entitäten (Datenpunkte, Auslöser, Graphen usw.) werden nicht entfernt, es sei denn, die Option *Fehlende löschen* ist auch für jede dieser Entitäten ausgewählt.

Attention:
Wenn bereits Vorlagen mit denselben Namen vorhanden sind, sollten beim Import die Optionen *Fehlende löschen* aktiviert werden, um einen sauberen Import zu erzielen. Auf diese Weise werden alte Datenpunkte entfernt, die in der aktualisierten Vorlage nicht mehr vorhanden sind (beachten Sie, dass dadurch der Verlauf dieser alten Datenpunkte verloren geht).

Auf dem nächsten Bildschirm können Sie den Inhalt einer importierten Vorlage anzeigen. Wenn es sich um eine neue Vorlage handelt, werden alle Elemente grün aufgelistet. Beim Aktualisieren einer vorhandenen Vorlage werden neue Vorlagenelemente grün hervorgehoben, entfernte Vorlagenelemente rot hervorgehoben und unveränderte Elemente auf einem grauen Hintergrund aufgelistet.



Über das Menü auf der linken Seite können Sie durch die Liste der Änderungen navigieren. Der Abschnitt *Aktualisiert* hebt alle Änderungen an vorhandenen Vorlagenelementen hervor. Der Abschnitt *Hinzugefügt* listet neue Vorlagenelemente auf. Die Elemente in den einzelnen Abschnitten sind nach Elementtyp gruppiert. Klicken Sie auf den grauen Pfeil, um die Elementgruppe ein- oder auszublenden.



Überprüfen Sie die Änderungen an der Vorlage und klicken Sie anschließend auf *Importieren*, um den Vorlagenimport durchzuführen. Im Frontend wird eine Erfolgs- oder Fehlermeldung zum Import angezeigt.

Exportformat Wenn eine Vorlage exportiert wird, generiert Zabbix ein strukturiertes YAML-, JSON- oder XML-Format. Der Export umfasst Vorlagenelemente wie Vorlagenmetadaten, Datenpunkte, Makros, Auslöser, Dashboards und mehr.

Jedes Element erfüllt einen bestimmten Zweck und kann verschachtelte Elemente enthalten.

Die folgenden Abschnitte beschreiben jedes Element im Exportformat. Die Beispiele verwenden die Vorlage Linux by Zabbix agent.

Eine Ellipse (`...`) kennzeichnet aus Platzgründen ausgelassene Elemente. Der Hinweis (`siehe Tabelle unten`) wird verwendet, wenn das Element in späteren Abschnitten ausführlicher erläutert wird.

```
zabbix_export:
  version: '8.0'
  template_groups:
    - uuid: 846977d1dfed4968bc5f8bdb363285bc
      name: 'Templates/Operating systems'
  templates: (siehe Tabelle unten)
  triggers: (siehe Tabelle unten)
  graphs: (siehe Tabelle unten)
```

Vorlagenmetadaten

Element	Type	Beschreibung
version	string	(erforderlich) Exportversion.
template_groups		(erforderlich) Wurzelelement für Vorlagengruppen.
uuid	string	(erforderlich) Eindeutige Kennung für diese Vorlagengruppe.
name	string	(erforderlich) Name der Vorlagengruppe.
host_groups		Wurzelelement für Host-Gruppen, die von Host-Prototypen verwendet werden.
uuid	string	(erforderlich) Eindeutige Kennung für diese Host-Gruppe.
name	string	(erforderlich) Name der Host-Gruppe.
templates		Wurzelelement für Vorlagen.
triggers		Wurzelelement für Vorlagen-Auslöser-Elemente, die identisch mit denen für Vorlagen-Datenpunkt-Auslöser sind.
graphs		Wurzelelement für Vorlagen-Graph-Elemente, die identisch mit denen für Host-Graphen sind.

Vorlagen

```
zabbix_export:
  (...)
  templates:
    - uuid: f8f7908280354f2abeed07dc788c3747
      template: 'Linux by Zabbix Agent'
      name: 'Linux by Zabbix Agent'
      description: |
        Dies ist eine offizielle Linux-Vorlage. Sie erfordert Zabbix Agent 8.0 oder neuer. (...)
      wizard_ready: 'YES'
      readme: |
        ## Übersicht

        Dies ist eine offizielle Linux-Vorlage. Sie erfordert Zabbix Agent 8.0 oder neuer. (...)
      vendor:
        name: Zabbix
        version: 8.0-2
      groups:
        - name: 'Vorlagen/Betriebssysteme'
      items: (siehe Tabelle unten)
      discovery_rules: (siehe Tabelle unten)
      tags:
        - tag: class
          value: os
        - tag: target
          value: linux
      macros:
        - macro: '{$AGENT.TIMEOUT}'
          value: 3m
          description: 'Zeitüberschreitung, nach der der Agent als nicht verfügbar gilt. Funktioniert nur
          config: (siehe Tabelle unten)
        - macro: '{$CPU.UTIL.CRIT}'
          value: '90'
          description: 'Kritischer Schwellenwert der CPU-Auslastung in %.'
          config: (siehe Tabelle unten)
```

```
      (...)
    dashboards: (siehe Tabelle unten)
    valuemaps: (siehe Tabelle unten)
  (...)
```

Element	Type	Description
uuid	string	(erforderlich) Eindeutige Kennung für diese Vorlage.
template	string	(erforderlich) Eindeutiger Vorlagenname.
name	string	Angezeigter Vorlagenname.
description	text	Beschreibung der Vorlage.
wizard_ready	text	Gibt an, ob die Vorlage im Host-Assistenten zur Auswahl verfügbar ist. Mögliche Werte:[1] NO (0, Standard), YES (1). Siehe auch: Template object (`wizard_ready`).
readme	text	Vorlagenspezifische Konfigurationsanweisungen, die im Host-Assistenten angezeigt werden. Unterstützt Markdown-Formatierung.
vendor		Stammelement für den Vorlagenanbieter (vorhanden, wenn die exportierte Vorlage Anbieterdaten enthält).
name	string	(erforderlich) Name des Vorlagenanbieters.
version	string	(erforderlich) Vorlagenversion. Für Standardvorlagen wird die Version wie folgt angezeigt: Hauptversion von Zabbix, Trennzeichen (”-”), Revisionsnummer (wird mit jeder neuen Version der Vorlage erhöht und mit jeder Hauptversion von Zabbix zurückgesetzt). Zum Beispiel 7.0-0, 7.0-3, 8.0-0, 8.0-3.
templates		Stammelement für verknüpfte Vorlagen.
name	string	(erforderlich) Vorlagenname.
groups		Stammelement für Vorlagengruppen.
name	string	(erforderlich) Name der Vorlagengruppe.
items		Stammelement für Vorlagen-Datenpunkte.
discovery_rules		Stammelement für Low-Level-Discovery-Regeln der Vorlage.
httptests		Stammelement für Webszenarien der Vorlage.
tags		Stammelement für Vorlagen-Tags.
tag	string	(erforderlich) Tag-Name.
value	string	Tag-Wert.
macros		Stammelement für benutzerdefinierte Vorlagenmakros.
macro	string	(erforderlich) Name des benutzerdefinierten Makros.
type	string	Typ des benutzerdefinierten Makros. Mögliche Werte:[1] TEXT (0, Standard), SECRET_TEXT (1), VAULT (2). Siehe auch: User macro object (`type`).
value	string	Wert des benutzerdefinierten Makros.
description	string	Beschreibung des benutzerdefinierten Makros.
config		Stammelement für die Konfiguration von Vorlagenmakros, die dafür verantwortlich ist, wie das Makro im Host-Assistenten angezeigt wird.
dashboards		Stammelement für Vorlagen-Dashboards.
valuemaps		Stammelement für Vorlagen-Wertzuordnungen.

Datenpunkte der Vorlage

```
zabbix_export:
  (...)
  templates:
    (...)
      items:
        - uuid: f94f9f4699e94c369e6c98b2a2f485ce
          name: 'Zabbix agent ping'
          key: agent.ping
          description: 'Der Agent gibt für diesen Datenpunkt immer "1" zurück. Kann in Kombination mit `no
          valuemap:
            name: 'Zabbix agent ping status'
          tags:
            - tag: component
              value: system
        (...)
```

```
      - uuid: 58818005e76d46dda14d6592f601ab00
        name: 'Anzahl der installierten Pakete'
        key: system.sw.packages.get
        delay: 1h
        preprocessing: (siehe Tabelle unten)
        tags:
          - tag: component
            value: os
        triggers: (siehe Tabelle unten)
      - uuid: 403cebed115441369e94d35d070ca7b8
        name: 'Speicherauslastung'
        type: DEPENDENT
        key: vm.memory.utilization
        value_type: FLOAT
        units: '%'
        description: 'Der Prozentsatz des verwendeten Speichers wird als `100-pavailable` berechnet.'
        preprocessing: (siehe Tabelle unten)
        master_item:
          key: 'vm.memory.size[pavailable]'
        tags:
          - tag: component
            value: memory
        triggers: (siehe Tabelle unten)
      (...)
```

Element	Typ	Beschreibung
uuid	string	(erforderlich) Eindeutige Kennung für diesen Datenpunkt.
name	string	(erforderlich) Name des Datenpunkts.
type	string	Datenpunkttyp. Mögliche Werte:[1] ZABBIX_PASSIVE (0, Standard), TRAP (2), SIMPLE (3), INTERNAL (5), ZABBIX_ACTIVE (7), EXTERNAL (10), ODBC (11), IPMI (12), SSH (13), TELNET (14), CALCULATED (15), JMX (16), SNMP_TRAP (17), DEPENDENT (18), HTTP_AGENT (19), SNMP_AGENT (20), ITEM_TYPE_SCRIPT (21), ITEM_TYPE_BROWSER (22), TELEMETRY_QUERY (24).
snmp_oid	string	(erforderlich für SNMP_AGENT-Datenpunkte) SNMP-Objekt-ID.
key	string	(erforderlich) Datenpunktschlüssel.
delay	string	Aktualisierungsintervall des Datenpunkts. Standard: 1m. Der Wert ist für TRAP-Datenpunkte immer 0.
history	string	Zeitraum (unter Verwendung eines Zeitsuffixes, Benutzermakros oder LLD-Makros), für den Verlaufsdaten gespeichert werden sollen. Standard: 31d.
trends	string	Zeitraum (unter Verwendung eines Zeitsuffixes, Benutzermakros oder LLD-Makros), für den Trenddaten gespeichert werden sollen. Standard: 365d.
status	string	Datenpunktstatus. Mögliche Werte:[1] ENABLED (0, Standard), DISABLED (1).
value_type	string	Typ des empfangenen Werts. Mögliche Werte:[1] FLOAT (0), CHAR (1), LOG (2), UNSIGNED (3, Standard), TEXT (4), BINARY (5), JSON (6).
allowed_hosts	string	Kommagetrennte Liste der IP-Adressen von Hosts, die Daten für den Datenpunkt senden dürfen. Unterstützt für TRAP- und HTTP_AGENT-Datenpunkte.
units	string	Einheiten des empfangenen Werts (bps, B usw.).
params	text	Zusätzliche Parameter abhängig vom Datenpunkttyp (ausgeführtes Skript für SSH- und TELNET-Datenpunkte; SQL-Abfrage für ODBC-Datenpunkte; Formel für CALCULATED-Datenpunkte; Skript für ITEM_TYPE_SCRIPT- und ITEM_TYPE_BROWSER-Datenpunkte).
ipmi_sensor	string	IPMI-Sensor. Unterstützt für IPMI-Datenpunkte.

Element	Typ	Beschreibung
authtype	string	Authentifizierungstyp. Unterstützt für SSH- und HTTP_AGENT-Datenpunkte. Mögliche Werte für SSH-Datenpunkte:[1] PASSWORD (0, Standard), PUBLIC_KEY (1). Mögliche Werte für HTTP_AGENT-Datenpunkte:[1] NONE (0, Standard), BASIC (1), NTLM (2), Kerberos (3) oder Digest (4).
username	string	(erforderlich für SSH- und TELNET-Datenpunkte) Benutzername für die Authentifizierung. Unterstützt für SIMPLE-, ODBC-, JMX- und HTTP_AGENT-Datenpunkte. Bei Verwendung für JMX-Datenpunkte muss auch `password` (siehe unten) angegeben werden, oder beide Elemente müssen leer bleiben.
password	string	(erforderlich für SSH- und TELNET-Datenpunkte) Passwort für die Authentifizierung. Unterstützt für SIMPLE-, ODBC-, JMX- und HTTP_AGENT-Datenpunkte. Bei Verwendung für JMX-Datenpunkte muss auch `username` (siehe oben) angegeben werden, oder beide Elemente müssen leer bleiben.
publickey	string	(erforderlich für SSH-Datenpunkte) Name der Datei mit dem öffentlichen Schlüssel.
privatekey	string	(erforderlich für SSH-Datenpunkte) Name der Datei mit dem privaten Schlüssel.
description	text	Beschreibung des Datenpunkts.
inventory_link	string	Host-Inventarfeld, das durch den Datenpunkt befüllt wird. Mögliche Werte:[1] NONE (0), ALIAS (4) usw. (unter Host-Inventar finden Sie die unterstützten Felder).
valuemap		Stammelement für Wertzuordnungen des Datenpunkts.
name	string	(erforderlich) Name der für den Datenpunkt zu verwendenden Wertzuordnung.
logtimefmt	string	Format der Zeit in Protokolleinträgen. Unterstützt für Datenpunkte des Werttyps LOG.
preprocessing		Stammelement für die Wertvorverarbeitung des Datenpunkts.
step		Stammelement für Wertvorverarbeitungsschritte von Datenpunkten der Vorlage.
jmx_endpoint	string	JMX-Endpunkt. Unterstützt für JMX-Datenpunkte.
master_item		(erforderlich für DEPENDENT-Datenpunkte) Stammelement für den Master-Datenpunkt des abhängigen Datenpunkts.
key	string	(erforderlich) Schlüssel des Master-Datenpunkts des abhängigen Datenpunkts.
timeout	string	Zeitüberschreitung für die Abfrage von Daten des Datenpunkts. Unterstützt für die Liste der Datenpunkttypen unter Zeitüberschreitungen.
url	string	(erforderlich für HTTP_AGENT-Datenpunkte) URL-Zeichenfolge.
query_fields		Stammelement für Abfrageparameter. Unterstützt für HTTP_AGENT-Datenpunkte.
name	string	(erforderlich für HTTP_AGENT-Datenpunkte) Name des Abfrageparameters.
value	string	Wert des Abfrageparameters. Unterstützt für HTTP_AGENT-Datenpunkte.
parameters		Stammelement für benutzerdefinierte Parameter. Unterstützt für ITEM_TYPE_SCRIPT- und ITEM_TYPE_BROWSER-Datenpunkte.
name	string	(erforderlich für ITEM_TYPE_SCRIPT- und ITEM_TYPE_BROWSER-Datenpunkte) Name des benutzerdefinierten Parameters.
value	string	Wert des benutzerdefinierten Parameters. Unterstützt für ITEM_TYPE_SCRIPT- und ITEM_TYPE_BROWSER-Datenpunkte.
posts	string	Daten des HTTP(S)-Anforderungstexts. Unterstützt für HTTP_AGENT-Datenpunkte.
status_codes	string	Durch Kommas getrennte Bereiche der erforderlichen HTTP-Statuscodes. Unterstützt für HTTP_AGENT-Datenpunkte.
follow_redirects	string	Beim Abrufen von Daten Weiterleitungen der Antwort verfolgen. Unterstützt für HTTP_AGENT-Datenpunkte. Mögliche Werte:[1] NO (0, Standard), YES (1).
post_type	string	Typ des Textkörpers der POST-Daten. Unterstützt für HTTP_AGENT-Datenpunkte. Mögliche Werte:[1] RAW (0, Standard), JSON (2), XML (3).
http_proxy	string	Verbindungszeichenfolge des HTTP(S)-Proxy. Unterstützt für HTTP_AGENT-Datenpunkte.
headers		Stammelement für HTTP(S)-Anforderungsheader. Unterstützt für HTTP_AGENT-Datenpunkte.
name	string	(erforderlich für HTTP_AGENT-Datenpunkte) Headername.
value	string	(erforderlich für HTTP_AGENT-Datenpunkte) Headerwert.

Element	Typ	Beschreibung
retrieve_mode	string	Welcher Teil der Antwort gespeichert werden soll. Unterstützt für HTTP_AGENT-Datenpunkte. Mögliche Werte:[1] BODY (0, Standard), HEADERS (1), BOTH (2).
request_method	string	Typ der Anforderungsmethode. Unterstützt für HTTP_AGENT-Datenpunkte. Mögliche Werte:[1] GET (0, Standard), POST (1), PUT (2), HEAD (3).
output_format	string	Verarbeitung der Antwort. Unterstützt für HTTP_AGENT-Datenpunkte. Mögliche Werte:[1] RAW (0, Standard), JSON (1).
allow_traps	string	Erlauben, dass der Wert ähnlich wie beim Trapper-Datenpunkt befüllt wird. Unterstützt für HTTP_AGENT-Datenpunkte. Mögliche Werte:[1] NO (0, Standard), YES (1).
ssl_cert_file	string	Dateipfad des öffentlichen SSL-Schlüssels. Unterstützt für HTTP_AGENT-Datenpunkte.
ssl_key_file	string	Dateipfad des privaten SSL-Schlüssels. Unterstützt für HTTP_AGENT-Datenpunkte.
ssl_key_password	string	Passwort für die SSL-Schlüsseldatei. Unterstützt für HTTP_AGENT-Datenpunkte.
verify_peer	string	Gibt an, ob überprüft werden soll, dass das Zertifikat des Hosts authentisch ist. Unterstützt für HTTP_AGENT-Datenpunkte. Mögliche Werte:[1] NO (0, Standard), YES (1).
verify_host	string	Gibt an, ob überprüft werden soll, dass der Hostname der Verbindung mit dem im Zertifikat des Hosts übereinstimmt. Unterstützt für HTTP_AGENT-Datenpunkte. Mögliche Werte:[1] NO (0, Standard), YES (1).
time_shift	string	Wie weit zurück (vor der aktuellen Zeit) der Zeitraum endet, den der Datenpunkt abfragt. Unterstützt für TELEMETRY_QUERY-Datenpunkte.
lookback_limit	string	Wie weit zurück (vor der aktuellen Zeit) der Datenpunkt Daten abfragt, wenn keine vorherige Abfrage vorhanden ist, mit der fortgefahren werden kann (z. B. beim ersten Lauf des Datenpunkts), oder wenn die vorherige Abfrage älter als `delay` ist (z. B. weil der Zabbix-Server ausgefallen war). Unterstützt für TELEMETRY_QUERY-Datenpunkte.
granularity	string	Länge eines Abschnitts, in den der vollständige Zeitraum der Abfrage aufgeteilt wird. Unterstützt für TELEMETRY_QUERY-Datenpunkte.
query		(erforderlich für TELEMETRY_QUERY-Datenpunkte) Stammelement für die Telemetrieabfrage. Definiert, welche Daten der Datenpunkt aus der Datenquelle abfragt, wie er die Daten filtert und wie er das Ergebnis aggregiert.
signal_type	string	(erforderlich) Signal-Typ, den der Datenpunkt abfragt. Unterstützt für TELEMETRY_QUERY-Datenpunkte. Mögliche Werte:[1] TRACES (0), METRICS (1), LOGS (2).
metric_point_type	string	(erforderlich für den Signaltyp METRICS) Metrikpunkt-Typ, den der Datenpunkt abfragt. Unterstützt für TELEMETRY_QUERY-Datenpunkte. Mögliche Werte:[1] SUM (0), GAUGE (1), HISTOGRAM (2), EXPONENTIAL_HISTOGRAM (3).
columns		(erforderlich) Stammelement für die Spalten der Telemetriedaten, die der Datenpunkt abfragt. Dieses Element enthält die folgenden Elemente: `column (string)` - (erforderlich) Name der Telemetriedatenspalte, die der Datenpunkt abfragt. `attribute_key (string)` - (erforderlich für Spalten mit Schlüssel-Wert-Paaren) Name des Schlüssels, den der Datenpunkt innerhalb der Spalte mit Schlüssel-Wert-Paaren abfragt.

Element	Typ	Beschreibung
aggregated_columns		(erforderlich) Stammelement für aggregierte Spalten der Telemetriedaten, die der Datenpunkt abfragt. Dieses Element enthält die folgenden Elemente: `column` (`string`) - (erforderlich für die Funktionen MIN, MAX, AVG, SUM und PERCENTILE) Name der Telemetriedatenspalte, die der Datenpunkt abfragt. `function` (`string`) - (erforderlich) Funktion, die der Datenpunkt auf die Spalteneinträge anwendet. Mögliche Werte:[1] MIN (1), MAX (2), AVG (3), COUNT (4), SUM (5), PERCENTILE (8). `parameters` (`array`) - (erforderlich für PERCENTILE-Funktionen) Perzentil, das der Datenpunkt aus den Spalteneinträgen berechnet. `alias` (`string`) - (erforderlich) Eindeutiger Name für den berechneten Wert, der zur Identifizierung in dem Datenpunktwert verwendet wird.
filter		(erforderlich) Stammelement für den Filter der Telemetrieabfrage für Spalteneinträge in der Abfrage. Dieses Element enthält die folgenden Elemente: `evaltype` (`string`) - (erforderlich) Auswertungsmethode für Filterbedingungen. Mögliche Werte:[1] AND_OR (0), AND (1), OR (2), FORMULA (3). `formula` (`string`) - (erforderlich für die Auswertungsmethoden FORMULA) Benutzerdefinierter Ausdruck zur Auswertung der Bedingungen von Filtern mit einem benutzerdefinierten Ausdruck. `conditions` (`string`) - (erforderlich) Stammelement für die Filterbedingungen der Telemetriedaten, die ein Eintrag in einer Telemetriedatenspalte erfüllen muss. Das Element `conditions` enthält die folgenden Elemente: `column` (`string`) - (erforderlich) Name der Telemetriedatenspalte, auf die die Filterbedingungen angewendet werden. `attribute_key` (`string`) - (erforderlich für Spalten mit Schlüssel-Wert-Paaren) Name des Schlüssels, auf den die Bedingung innerhalb der Spalte mit Schlüssel-Wert-Paaren angewendet wird. `operator` (`string`) - (erforderlich) Operator der Filterbedingung. Mögliche Werte für Spalten mit Einzelwerten:[1] EQUAL (0), NOT_EQUAL (1), LIKE (2), NOT_LIKE (3). Mögliche Werte für Spalten mit Schlüssel-Wert-Paaren:[1] EQUAL (0), NOT_EQUAL (1), EXISTS (12). `formulaid` (`string`) - (erforderlich für die Auswertungsmethoden FORMULA) Beliebige eindeutige ID, die verwendet wird, um in einem benutzerdefinierten Ausdruck auf die Bedingung zu verweisen. `value` (`string`) - (erforderlich für die Operationen EQUAL, NOT_EQUAL, LIKE und NOT LIKE) Wert, mit dem verglichen werden soll.
tags		Stammelement für Tags des Datenpunkts.
tag	string	(erforderlich) Tagname.
value	string	Tagwert.
triggers		Stammelement für Auslöser von Datenpunkten der Vorlage.

Note:
Siehe auch: Datenpunktobjekt (siehe die relevante Eigenschaft mit übereinstimmendem Namen).

Vorverarbeitungsschritte für den Wert eines Vorlagen-Datenpunkts

```
zabbix_export:
  (...)
  templates:
    (...)
      items:
        (...)
        - uuid: 58818005e76d46dda14d6592f601ab00
          (...)
          preprocessing:
            - type: JSONPATH
              parameters:
                - $.length()
```

```
            - type: DISCARD_UNCHANGED_HEARTBEAT
              parameters:
                - 12h
          (...)
```

Element	Type	Beschreibung
type	string	(erforderlich) Typ des Vorverarbeitungsschritts für den Wert des Datenpunkts. Mögliche Werte:[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, berechnet: empfangener Wert - vorheriger Wert), CHANGE_PER_SECOND (10, berechnet: (empfangener Wert - vorheriger Wert)/(aktuelle Zeit - Zeitpunkt der letzten Prüfung)), 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		(erforderlich) Wurzelelement für Parameter des Vorverarbeitungsschritts für den Wert des Datenpunkts.
parameter	string	Einzelner Parameter des Vorverarbeitungsschritts für den Wert des Datenpunkts.
error_handler	string	Aktionstyp, der im Falle eines Fehlers beim Vorverarbeitungsschritt verwendet wird. Mögliche Werte:[1] ORIGINAL_ERROR (0, Standard), DISCARD_VALUE (1), CUSTOM_VALUE (2), CUSTOM_ERROR (3).
error_handler_params	string	Parameter des Fehlerbehandlers.

Note:
Siehe auch: Objekt zur Vorverarbeitung von Datenpunkten (verweisen Sie auf die entsprechende Eigenschaft mit demselben Namen).

Vorlage-Item-Auslöser

```
zabbix_export:
  (...)
  templates:
    (...)
      items:
        (...)
        - uuid: 58818005e76d46dda14d6592f601ab00
          (...)
          triggers:
            - uuid: b950c306394f4b3c902060a8273cbcde
              expression: 'change(/Linux by Zabbix agent/system.sw.packages.get)<>0'
              name: 'Linux: Anzahl der installierten Pakete wurde geändert'
              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)>{$MEMORY.UTIL.MAX}'
              name: 'Linux: Hohe Speicherauslastung'
              event_name: 'Linux: Hohe Speicherauslastung (>{$MEMORY.UTIL.MAX}% für 5m)'
              priority: AVERAGE
              description: 'Dem System geht der freie Speicher aus.'
              dependencies:
                - name: 'Linux: Nicht genügend verfügbarer Speicher'
```

```
                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	(erforderlich) Eindeutige Kennung für diesen Auslöser.
expression	string	(erforderlich) Auslöserausdruck.
recovery_mode	string	Grundlage für die Generierung von OK-Ereignissen. Mögliche Werte:[1] EXPRESSION (0, Standard), RECOVERY_EXPRESSION (1), NONE (2).
recovery_expression	string	Wiederherstellungsausdruck des Auslösers.
correlation_mode	string	Korrelationsmodus (keine Ereigniskorrelation oder Ereigniskorrelation nach Tag). Mögliche Werte:[1] DISABLED (0, Standard), TAG_VALUE (1).
correlation_tag	string	Der Tag-Name, der für die Ereigniskorrelation verwendet werden soll.
name	string	(erforderlich) Name des Auslösers.
event_name	string	Ereignisname.
opdata	string	Betriebsdaten.
url_name	string	Bezeichnung für die dem Auslöser zugeordnete URL.
url	string	Dem Auslöser zugeordnete URL.
status	string	Status des Auslösers. Mögliche Werte:[1] ENABLED (0, Standard), DISABLED (1).
priority	string	Schweregrad des Auslösers. Mögliche Werte:[1] NOT_CLASSIFIED (0, Standard), INFO (1), WARNING (2), AVERAGE (3), HIGH (4), DISASTER (5).
description	text	Beschreibung des Auslösers.
type	string	Typ der Ereignisgenerierung (ein einzelnes Problemereignis oder mehrere Problemereignisse). Mögliche Werte:[1] SINGLE (0, Standard), MULTIPLE (1).
manual_close	string	Manuelles Schließen von Problemereignissen. Mögliche Werte:[1] NO (0, Standard), YES (1).
dependencies		Stammelement für Abhängigkeiten.
name	string	(erforderlich) Name des abhängigen Auslösers.
expression	string	(erforderlich) Ausdruck des abhängigen Auslösers.
recovery_expression	string	Wiederherstellungsausdruck des abhängigen Auslösers.
tags		Stammelement für Auslöser-Tags.
tag	string	(erforderlich) Tag-Name.
value	string	Tag-Wert.

Note:
Siehe auch: Trigger object (verweisen Sie auf die entsprechende Eigenschaft mit demselben Namen).

Vorlage Low-Level-Discovery-Regeln

```
zabbix_export:
  (...)
  templates:
    (...)
      discovery_rules:
        - uuid: acfdea9c46ef48c68e6636f43b8f96a2
          name: 'Network interface discovery'
          key: net.if.discovery
          delay: 1h
          filter: (see table below)
          description: 'The discovery of network interfaces.'
          item_prototypes: (see table below)
          trigger_prototypes: (see table below)
          graph_prototypes: (see table below)
        (...)
```

Attention:
Die meisten Elemente von Low-Level-Discovery-Regeln in Vorlagen sind dieselben wie bei Vorlagen-Datenpunkten. Die folgende Tabelle beschreibt die Elemente, die sich von Vorlagen-Datenpunkten unterscheiden.

Element	Type	Description
type	string	Datenpunkt-Typ. Mögliche Werte:[1] ZABBIX_PASSIVE (0, default), 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	(erforderlich) Der Schlüssel der Low-Level-Discovery-Regel.
filter		Stammelement für Filter von Low-Level-Discovery-Regeln in Vorlagen.
lifetime	string	Zeitraum (unter Verwendung von Sekunden, Zeitsuffix oder Benutzermakro), nach dem nicht mehr entdeckte Ressourcen gelöscht werden. Standard: 7d.
lifetime_type	string	Szenario zum Löschen verlorener LLD-Ressourcen. Mögliche Werte: DELETE_NEVER, DELETE_IMMEDIATELY, DELETE_AFTER.
enabled_lifetime	string	Zeitraum (unter Verwendung von Sekunden, Zeitsuffix oder Benutzermakro), nach dem nicht mehr entdeckte Ressourcen deaktiviert werden.
enabled_lifetime_type	string	Szenario zum Deaktivieren verlorener LLD-Ressourcen. Mögliche Werte: DISABLE_NEVER, DISABLE_IMMEDIATELY, DISABLE_AFTER.
item_prototypes		Stammelement für Vorlagen-Datenpunktprototypen, die dieselben sind wie bei Vorlagen-Datenpunkten.
trigger_prototypes		Stammelement für Vorlagen-Auslöserprototypen, die dieselben sind wie bei Vorlagen-Datenpunktauslösern.
graph_prototypes		Stammelement für Vorlagen-Grafikprototypen, die dieselben sind wie bei Host-Grafiken.
host_prototypes		Stammelement für Vorlagen-Hostprototypen, die dieselben sind wie bei Hosts.
parent_discovery_rule		Stammelement für die übergeordnete Low-Level-Discovery-Regel (oder Regelvorlage) der Low-Level-Discovery-Regelvorlage. Diese Eigenschaft kennzeichnet, dass es sich um eine LLD-Regelvorlage handelt, die direktes Kind der referenzierten Regel/Regelvorlage ist.
key	string	(erforderlich) Der Schlüssel der übergeordneten Low-Level-Discovery-Regel (oder Regelvorlage).
master_item	string	(erforderlich für DEPENDENT-Regeln) Stammelement für den Master-Datenpunkt der abhängigen Regel.
lld_macro_paths		Stammelement für Makropfade der Low-Level-Discovery-Regel.
lld_macro	string	(erforderlich) Name des Makros der Low-Level-Discovery-Regel.
path	string	(erforderlich) Selektor für den Wert, der dem entsprechenden Makro zugewiesen wird.
preprocessing		Stammelement für die Vorverarbeitung des Werts der Low-Level-Discovery-Regel.
step		Stammelement für Elemente der Vorverarbeitungsschritte des Werts der Low-Level-Discovery-Regel, die dieselben sind wie bei Vorverarbeitungsschritten von Vorlagen-Datenpunkten, jedoch mit weniger möglichen Werten. Siehe auch: LLD-Regel-Vorverarbeitungsobjekt.
overrides		Stammelement für Überschreibungsregeln der Low-Level-Discovery-Regel.
name	string	(erforderlich) Eindeutiger Name der Überschreibung.
step	string	(erforderlich) Eindeutige Ordnungsnummer der Überschreibung.
stop	string	Die Verarbeitung der nächsten Überschreibungen beenden, wenn zutreffend.
filter		Stammelement für Filterelemente von Überschreibungsregeln für Low-Level-Discovery-Regeln in Vorlagen, die dieselben sind wie bei Filtern von Low-Level-Discovery-Regeln in Vorlagen.
operations		Stammelement für Überschreibungsoperationen von Low-Level-Discovery-Regeln in Vorlagen.

Note:
Siehe auch: LLD-Regelobjekt (verweisen Sie auf die entsprechende Eigenschaft mit passendem Namen).

Filter für Low-Level-Discovery-Regeln der Vorlage

```
zabbix_export:
  (...)
```

```
templates:
  (...)
    discovery_rules:
      - uuid: acfdea9c46ef48c68e6636f43b8f96a2
        (...)
        filter:
          evaltype: AND
          conditions:
            - macro: '{#IFNAME}'
              value: '{$NET.IF.IFNAME.MATCHES}'
            - macro: '{#IFNAME}'
              value: '{$NET.IF.IFNAME.NOT_MATCHES}'
              operator: NOT_MATCHES_REGEX
        (...)
      - uuid: 2bbdc79f082d4c618e01bec625e9c90a
        (...)
        filter:
          evaltype: AND
          conditions:
            - macro: '{#DEVNAME}'
              value: '{$VFS.DEV.DEVNAME.MATCHES}'
            - macro: '{#DEVNAME}'
              value: '{$VFS.DEV.DEVNAME.NOT_MATCHES}'
              operator: NOT_MATCHES_REGEX
            - macro: '{#DEVTYPE}'
              value: disk
        (...)
```

Element	Type	Beschreibung
evaltype	string	Methode zur Auswertung der Filterbedingungen überschreiben. Mögliche Werte:[1] AND_OR (0, Standard), AND (1), OR (2), FORMULA (3).
formula	string	Benutzerdefinierte Berechnungsformel für Filterbedingungen.
conditions		Stammelement für Filterbedingungen.
macro	string	(erforderlich) Der Name des Low-Level-Discovery-Makros, auf das die Prüfung angewendet wird.
value	string	Wert, mit dem verglichen werden soll.
operator	string	Bedingungsoperator. Mögliche Werte:[1] MATCHES_REGEX (8, Standard), NOT_MATCHES_REGEX (9).
formulaid	string	(erforderlich) Beliebige eindeutige ID, die verwendet wird, um in der benutzerdefinierten Ausdrucksformel auf eine Bedingung zu verweisen. Darf nur Großbuchstaben enthalten. Die ID muss vom Benutzer beim Ändern der Filterbedingungen festgelegt werden, wird jedoch bei einer späteren Abfrage neu generiert.

Note:
Siehe auch: LLD rule filter object (siehe die entsprechende Eigenschaft mit demselben Namen).

Überschreibungsoperationen für Low-Level-Discovery-Regeln von Vorlagen

Element	Type	Description
operationobject	string	Objekt, auf das die Operation angewendet wird. Mögliche Werte:[1] ITEM_PROTOTYPE (0), TRIGGER_PROTOTYPE (1), GRAPH_PROTOTYPE (2), HOST_PROTOTYPE (3).
operator	string	Operator der Überschreibungsbedingung. Mögliche Werte:[1] EQUAL (1), NOT_EQUAL (2), LIKE (3), NOT_LIKE (4), REGEXP (5), NOT_REGEXP (6).
value	string	Ein regulärer Ausdruck oder eine Zeichenfolge für den Operator der Überschreibungsbedingung.
status	string	Status des Objekts nach der Überschreibungsoperation.
discover	string	Ob das Objekt als Ergebnis der Erkennung hinzugefügt wird.

Element	Type	Description
delay	string	Aktualisierungsintervall, das für den Datenpunkt-Prototyp nach der Überschreibungsoperation festgelegt wird.
history	string	Speicherzeitraum für Historie, der für den Datenpunkt-Prototyp nach der Überschreibungsoperation festgelegt wird.
trends	string	Speicherzeitraum für Trends, der für den Datenpunkt-Prototyp nach der Überschreibungsoperation festgelegt wird.
severity	string	Schweregrad des Auslöser-Prototyps, der nach der Überschreibungsoperation festgelegt wird.
tags		Stammelement für die Tags, die für das Objekt nach der Überschreibungsoperation festgelegt werden.
tag	string	(erforderlich) Tag-Name.
value	string	Tag-Wert.
templates		Stammelement für die mit dem Host-Prototyp verknüpften Vorlagen nach der Überschreibungsoperation.
name	string	(erforderlich) Vorlagenname.
inventory_mode	string	Inventarmodus des Host-Prototyps, der nach der Überschreibungsoperation festgelegt wird.

Note:
Siehe auch: LLD rule override operation object (verweisen Sie auf die entsprechende Eigenschaft mit dem passenden Namen).

Vorlage-Webszenarien

Element	Type	Description
uuid	string	(erforderlich) Eindeutige Kennung für dieses Webszenario.
name	string	(erforderlich) Name des Webszenarios.
delay	string	Intervall (in Sekunden, Zeitsuffix oder Benutzermakro) für die Ausführung des Webszenarios. Standard: 1m.
attempts	integer	Anzahl der Versuche für die Ausführung der Webszenario-Schritte. Mögliche Werte:[1] 1-10 (Standard: 1).
agent	string	Client-Agent. Zabbix gibt vor, der ausgewählte Browser zu sein. Dies ist nützlich, wenn eine Website für verschiedene Browser unterschiedliche Inhalte zurückgibt. Standard: Zabbix.
http_proxy	string	Proxy, der vom Webszenario verwendet wird, angegeben als: `http://[username[:password]@]proxy.example.com[:port]`
variables		Wurzelelement für Webszenario-Variablen, die in Szenario-Schritten verwendet werden können.
name	string	(erforderlich) Variablenname.
value	text	(erforderlich) Variablenwert.
headers		Wurzelelement für HTTP-Header, die beim Ausführen einer Anfrage gesendet werden. Header sollten mit derselben Syntax aufgelistet werden, wie sie im HTTP-Protokoll erscheinen würden.
name	string	(erforderlich) Headername.
value	text	(erforderlich) Headerwert.
status	string	Status des Webszenarios. Mögliche Werte:[1] ENABLED (0, Standard), DISABLED (1).
authentication	string	Authentifizierungsmethode. Mögliche Werte:[1] NONE (0, Standard), BASIC (1), NTLM (2), Kerberos (3) oder Digest (4).
http_user	string	Benutzername, der für die BASIC-(HTTP)-, NTLM-, Kerberos- oder Digest-Authentifizierung verwendet wird.
http_password	string	Passwort, das für die BASIC-(HTTP)-, NTLM-, Kerberos- oder Digest-Authentifizierung verwendet wird.
verify_peer	string	SSL-Zertifikat des Webservers überprüfen. Mögliche Werte:[1] NO (0, Standard), YES (1).

Element	Type	Description
verify_host	string	Überprüfen, ob das Feld *Common Name* oder das Feld *Subject Alternate Name* des Webserver-Zertifikats übereinstimmt. Mögliche Werte:[1] NO (0, Standard), YES (1).
ssl_cert_file	string	Name der SSL-Zertifikatsdatei, die für die Client-Authentifizierung verwendet wird (muss im PEM-Format vorliegen).
ssl_key_file	string	Name der SSL-Privatschlüsseldatei, die für die Client-Authentifizierung verwendet wird (muss im PEM-Format vorliegen).
ssl_key_password	string	Passwort der SSL-Privatschlüsseldatei.
steps		(erforderlich) Wurzelelement für Vorlage-Webszenario-Schritte.
tags		Wurzelelement für Webszenario-Tags.
tag	string	(erforderlich) Tag-Name.
value	string	Tag-Wert.

Note:
Siehe auch: Webszenario-Objekt (verweisen Sie auf die entsprechende Eigenschaft mit demselben Namen).

Vorlage web scenario steps

Element	Type	Description
name	string	(required) Name des Web-Szenario-Schritts.
url	string	(required) URL zur Überwachung.
query_fields		Stammelement für Abfrageparameter (ein Array von HTTP-Feldern, die der URL beim Ausführen einer Anfrage hinzugefügt werden).
name	string	(required) Name des Abfrageparameters.
value	string	Wert des Abfrageparameters.
posts		Stammelement für HTTP-POST-Variablen (ein String (Raw-Post-Daten) oder ein Array von HTTP-Feldern (Formulardaten)).
name	string	(required) Name des Post-Felds.
value	string	(required) Wert des Post-Felds.
variables		Stammelement für Variablen auf Schritt-Ebene (Makros), die nach diesem Schritt angewendet werden sollen. Wenn der Variablenwert ein Präfix 'regex:' hat, wird sein Wert aus den von diesem Schritt zurückgegebenen Daten gemäß dem regulären Ausdruck extrahiert, der auf das Präfix 'regex:' folgt
name	string	(required) Variablenname.
value	text	(required) Variablenwert.
headers		Stammelement für HTTP-Header, die beim Ausführen einer Anfrage gesendet werden.
name	string	(required) Name des Headers.
value	text	(required) Wert des Headers.
follow_redirects	string	HTTP-Weiterleitungen folgen. Mögliche Werte:[1] NO (0), YES (1, Standard).
retrieve_mode	string	Abrufmodus der HTTP-Antwort. Mögliche Werte:[1] BODY (0, Standard), HEADERS (1), BOTH (2).
timeout	string	Zeitüberschreitung (in Sekunden, Zeitsuffix oder Benutzermakro) bei der Ausführung des Schritts. Standard: 15s.
required	string	Text, der in der Antwort vorhanden sein muss (wird ignoriert, wenn leer).
status_codes	string	Eine durch Kommas getrennte Liste akzeptierter HTTP-Statuscodes (z. B. 200-201,210-299; wird ignoriert, wenn leer).

Note:
Siehe auch: Objekt "Web-Szenario-Schritt" (beziehen Sie sich auf die entsprechende Eigenschaft mit dem passenden Namen).

Konfiguration von Vorlagenmakros

```
zabbix_export:
  (...)
```

```yaml
templates:
  - uuid: f8f7908280354f2abeed07dc788c3747
    (...)
    macros:
      - macro: '{$AGENT.TIMEOUT}'
        (...)
        config:
          type: TEXT
          priority: '1'
          label: 'Sekunden seit dem letzten gesehenen Zabbix Agent'
          description: 'Zeitüberschreitung, nach der der Agent als nicht verfügbar betrachtet wird.'
      - macro: '{$CPU.UTIL.CRIT}'
        (...)
        config:
          type: TEXT
          priority: '2'
          section_name: Thresholds
          label: 'Ausgedrückter Schwellenwert der CPU-Auslastung'
          description: 'Kritischer Schwellenwert der CPU-Auslastung in %. Im Bereich von 0 bis einschlie
          regex: '^-?([0-9]+|(([0-9]+)\.([0-9]+)))$'
      (...)
      - macro: '{$IFCONTROL}'
        (...)
        config:
          type: CHECKBOX
          priority: '19'
          label: 'Schnittstellensteuerung'
          description: 'Löst einen Auslöser aus, wenn sich der Betriebsstatus der Schnittstelle in "Link
          options:
            - checked: '1'
              unchecked: '0'
      (...)
```

Element	Type	Description
type	string	(erforderlich) Typ des Makro-Eingabefelds. Mögliche Werte:[1] NOCONF (0), TEXT (1), LIST (2), CHECKBOX (3).
priority	string	Position des Makros in der Makroliste.
section_name	string	Bezeichnung des einklappbaren Abschnitts, in dem das Makro gruppiert ist.
label	string	(erforderlich für TEXT-, LIST- und CHECKBOX-Makros) Bezeichnung des Makros.
description	text	Hilfetext für das Makro. Unterstützt Markdown-Formatierung.
required	string	Kennzeichnet das Makro als obligatorisch. Mögliche Werte:[1] NO (0), YES (1). Unterstützt für TEXT- und LIST-Makros.
regex	string	Regulärer Ausdruck zur Validierung der Benutzereingabe in einem Textfeld. Unterstützt für TEXT-Makros.
options		Stammelement für LIST-Elemente oder CHECKBOX-Werte.
value	string	(erforderlich für LIST-Makros) Wert des LIST-Elements.
text	string	(erforderlich für LIST-Makros) Text des LIST-Elements.
checked	string	(erforderlich für CHECKBOX-Makros) Wert, der einen aktivierten Zustand darstellt. Mögliche Werte:[1] 0 (false), 1 (true). Unterstützt für CHECKBOX-Makros.
unchecked	string	(erforderlich für CHECKBOX-Makros) Wert, der einen deaktivierten Zustand darstellt. Mögliche Werte:[1] 0 (false), 1 (true). Unterstützt für CHECKBOX-Makros.

Note:
Siehe auch: Makrokonfigurationsobjekt (verweisen Sie auf die entsprechende Eigenschaft mit passendem Namen).

Vorlage-Dashboards

```
zabbix_export:
  (...)
  templates:
    (...)
      dashboards:
        - uuid: c689ad3115fd46a4b927d1f70ee2e5a4
          name: Filesystems
          pages:
            - name: Overview
              widgets: (see table below)
        (...)
```

Element	Type	Description
uuid	string	(erforderlich) Eindeutiger Bezeichner für dieses Dashboard.
name	string	(erforderlich) Name des Vorlage-Dashboards.
display pe- riod	integer	Anzeigedauer der Dashboard-Seiten.
auto_start	string	Automatischer Start der Diashow. Mögliche Werte:[1] NO (0), YES (1, Standard).
pages		Stammelement für Seiten des Vorlage-Dashboards.
name	string	Seitenname.
display period	integer	Intervall der Diashow.
sortorder	integer	Sortierreihenfolge der Seite.
widgets		Stammelement für Widgets des Vorlage-Dashboards.

Note:
Siehe auch: Objekt "Template dashboard" (beziehen Sie sich auf die entsprechende Eigenschaft mit passendem Namen).

Vorlage-Dashboard-Widgets

```
zabbix_export:
  (...)
  templates:
    (...)
      dashboards:
        - uuid: c689ad3115fd46a4b927d1f70ee2e5a4
          name: Filesystems
          pages:
            - name: Overview
              widgets:
                - type: graphprototype
                  width: '72'
                  height: '5'
                  fields:
                    - type: INTEGER
                      name: columns
                      value: '3'
                    - type: GRAPH_PROTOTYPE
                      name: graphid.0
                      value:
                        host: 'Linux by Zabbix agent'
                        name: 'FS [{#FSTYPE}({#FSNAME})]: Diagramm zur Speicherauslastung (relativ zum Ges
                    - 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})]: Diagramm zur Speichernutzung in % (relativ zum m
                  - type: STRING
                    name: reference
                    value: AAAEC
        (...)
```

Element	Type	Description
type	string	(erforderlich) Widget-Typ.
name	string	Widget-Name.
x	integer	Horizontale Position von der linken Seite des Vorlage-Dashboards aus. Mögliche Werte:[1] 0-71.
y	integer	Vertikale Position von der oberen Seite des Vorlage-Dashboards aus. Mögliche Werte:[1] 0-63.
width	integer	Widget-Breite. Mögliche Werte:[1] 1-72.
height	integer	Widget-Höhe. Mögliche Werte:[1] 1-64.
hide_header	string	Widget-Kopfzeile ausblenden. Mögliche Werte:[1] NO (0, Standard), YES (1).
fields		Stammelement für die Felder des Vorlage-Dashboard-Widgets.
type	string	(erforderlich) Typ des Widget-Felds. Mögliche Werte:[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	(erforderlich) Name des Widget-Felds.
value	mixed	(erforderlich) Wert des Widget-Felds, abhängig vom Feldtyp.

Note:
Siehe auch: Objekt "Vorlage-Dashboard-Widget" (siehe die entsprechende Eigenschaft mit passendem Namen).

Vorlagen-Wertzuordnungen

```
zabbix_export:
  (...)
  templates:
    (...)
      valuemaps:
        - uuid: 8c048c6cca8248f2860c208e8db0f59e
          name: ifOperStatus
          mappings:
            - value: '0'
              newvalue: unknown
            - value: '1'
              newvalue: notpresent
            - value: '2'
              newvalue: down
            - value: '3'
              newvalue: lowerlayerdown
            - value: '4'
              newvalue: testing
            - value: '5'
              newvalue: dormant
            - value: '6'
```

```
            newvalue: up
        (...)
```

Element	Type	Beschreibung
uuid	string	(erforderlich) Eindeutige Kennung für diese Wertzuordnung.
name	string	(erforderlich) Name der Wertzuordnung.
mapping		Wurzelelement für Zuordnungen.
type	string	Typ des Zuordnungsabgleichs. Mögliche Werte:[1] EQUAL (0, Standard), GREATER_OR_EQUAL (2), LESS_OR_EQUAL (3), IN_RANGE (4), REGEXP (5), DEFAULT (6).
value	string	Ursprünglicher Wert.
newvalue	string	(erforderlich) Wert, dem der ursprüngliche Wert zugeordnet wird.

Note:
Siehe auch: Value map object (verweisen Sie auf die entsprechende Eigenschaft mit passendem Namen).

Fußnoten [1] API-Ganzzahlwerte in Klammern, zum Beispiel ENABLED (0), werden nur zur Referenz angegeben. Weitere Informationen finden Sie auf der verlinkten API-Objektseite im Tabelleneintrag oder am Ende jedes Abschnitts.

Hosts

Übersicht

Hosts werden mit vielen zugehörigen Objekten und Objektbeziehungen exportiert.

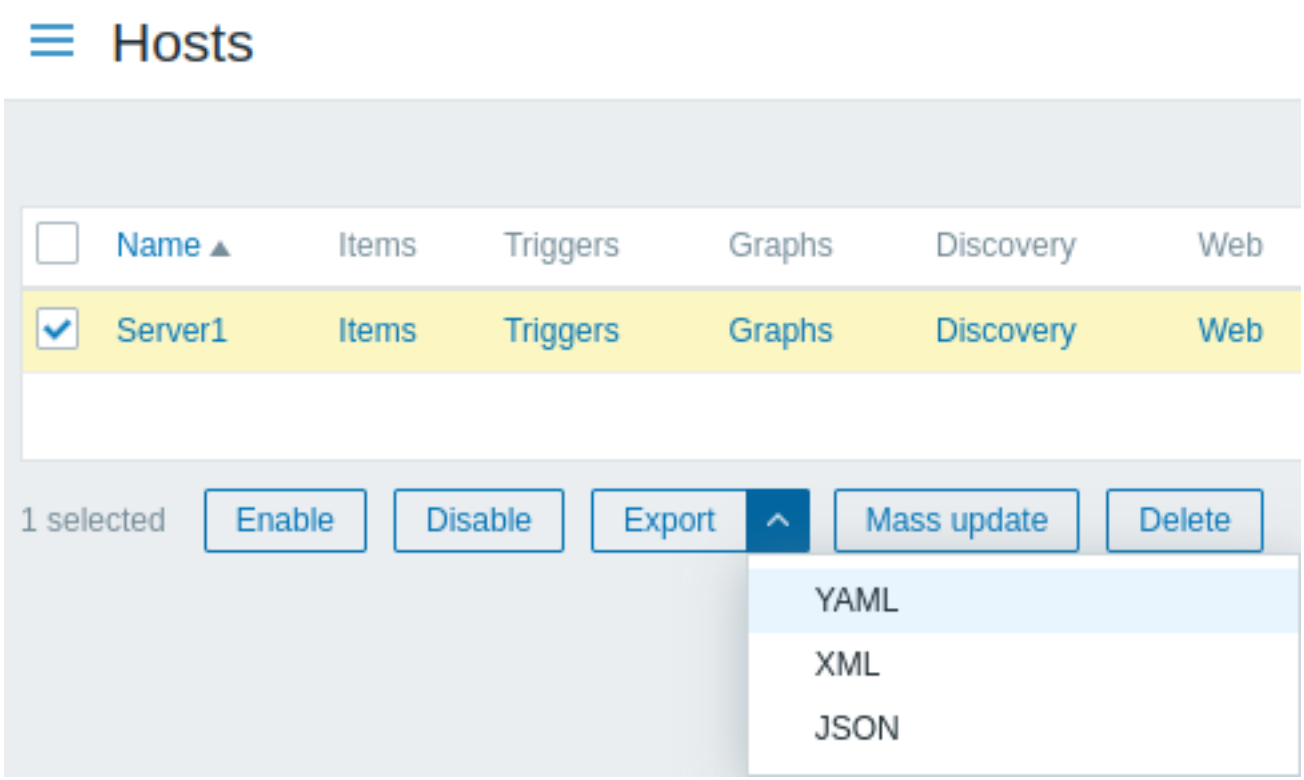
Der Host-Export enthält:

- Verknüpfte Hostgruppen
- Host-Daten
- Vorlagenverknüpfung
- Hostgruppenverknüpfung
- Host-Schnittstellen
- Direkt verknüpfte Datenpunkte
- Direkt verknüpfte Auslöser
- Direkt verknüpfte Erkennungsregeln mit allen Prototypen
- Direkt verknüpfte Webszenarien
- Host-Makros
- Host-Inventardaten
- Wertzuordnungen
- Verknüpfte Diagramme

Exportieren

Gehen Sie zum Exportieren von Hosts wie folgt vor:

1. Wechseln Sie zu *Datenerfassung* > *Hosts*.
2. Markieren Sie die Kontrollkästchen der Hosts, die exportiert werden sollen.
3. Klicken Sie unterhalb der Liste auf *Export*, um im standardmäßigen YAML-Format zu exportieren, oder klicken Sie auf den Pfeil, um ein anderes Exportformat auszuwählen.



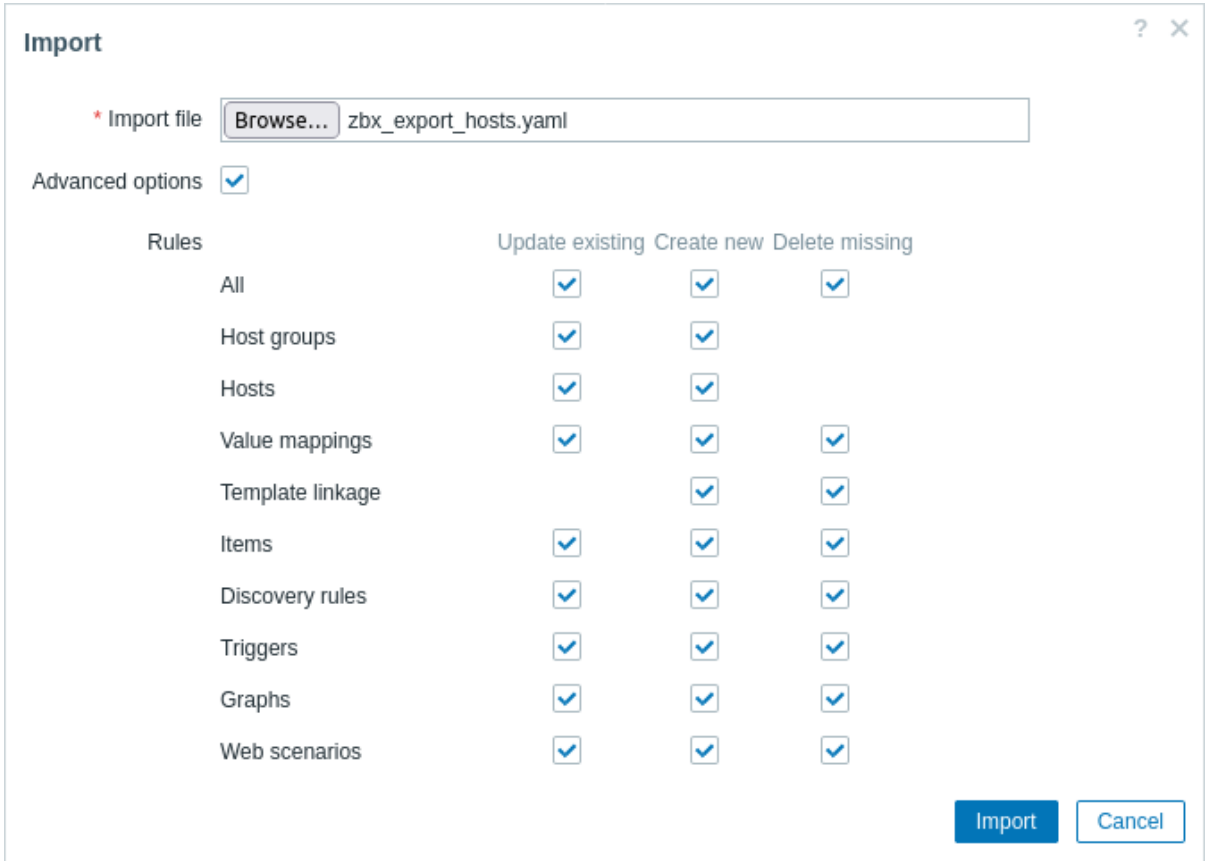
Je nach ausgewähltem Format werden Hosts in eine lokale Datei mit einem Standardnamen exportiert:

- `zabbix_export_hosts.yaml` - beim YAML-Export;
- `zabbix_export_hosts.xml` - beim XML-Export;
- `zabbix_export_hosts.json` - beim JSON-Export.

Importieren

Gehen Sie wie folgt vor, um Hosts zu importieren:

1. Gehen Sie zu *Datensammlung* > *Hosts*.
2. Klicken Sie oben rechts auf *Importieren*.
3. Wählen Sie die Importdatei aus.
4. Aktivieren Sie die Kontrollkästchen der Importregeln, die Sie anwenden möchten (siehe unten).
5. Klicken Sie unten rechts im Konfigurationsformular auf *Importieren*.



Wenn Sie das Kontrollkästchen *Erweiterte Optionen* aktivieren, wird eine detaillierte Liste aller importierbaren Elemente angezeigt. Aktivieren oder deaktivieren Sie jede Importregel nach Bedarf.

Wenn Sie das Kontrollkästchen in der Zeile *Alle* aktivieren, werden alle darunter aufgeführten Elemente aktiviert bzw. deaktiviert.

Importregeln:

Regel	Beschreibung
Vorhandene aktualisieren	Vorhandene Elemente werden anhand der Daten aus der Importdatei aktualisiert. Andernfalls werden sie nicht aktualisiert.
Neue erstellen	Neue Elemente werden anhand der Daten aus der Importdatei erstellt. Andernfalls werden sie nicht erstellt.
Fehlende löschen	Vorhandene Elemente, die nicht in der Importdatei enthalten sind, werden entfernt. Andernfalls werden sie nicht entfernt. Wenn *Fehlende löschen* für *Vorlagenverknüpfung* aktiviert ist, werden aktuelle Vorlagenverknüpfungen, die nicht in der Importdatei enthalten sind, aufgehoben. Von den aufgehobenen Vorlagen geerbte Entitäten (Datenpunkte, Auslöser, Graphen usw.) werden nicht entfernt, es sei denn, die Option *Fehlende löschen* ist ebenfalls für jede Entität ausgewählt.

Sie können Änderungen an Hosts während des Imports nicht überprüfen. Eine Erfolgs- oder Fehlermeldung zum Import wird im Frontend angezeigt.

Exportformat

Wenn ein Host exportiert wird, generiert Zabbix ein strukturiertes YAML-, JSON- oder XML-Format. Der Export umfasst Host-Elemente wie Host-Metadaten, Datenpunkte, Makros, Auslöser und mehr.

Jedes Element dient einem bestimmten Zweck und kann verschachtelte Elemente enthalten.

Die folgenden Abschnitte beschreiben jedes Element im Exportformat. Die Beispiele verwenden einen Host mit der Vorlage MySQL by Zabbix agent 2. Zu Demonstrationszwecken für die Elemente wird die Vorlage nach dem Erstellen des Hosts getrennt. Beim Trennen bleiben alle Elemente in der Konfiguration erhalten (im Gegensatz zu Trennen und Löschen, wodurch sie entfernt werden).

Eine Auslassung (`...`) kennzeichnet aus Platzgründen weggelassene Elemente. Der Hinweis (`siehe Tabelle unten`) wird verwendet, wenn das Element in späteren Abschnitten ausführlicher erläutert wird.

```
zabbix_export:
  version: '8.0'
  host_groups:
    - uuid: 748ad4d098d447d492bb935c907f652f
      name: Databases
  hosts: (siehe Tabelle unten)
  graphs: (siehe Tabelle unten)
```

Host-Metadaten

Element		Type	Description
version		string	(erforderlich) Zabbix-Version.
host_groups			(erforderlich) Wurzelelement für Host-Gruppen.
	uuid	string	(erforderlich) Eindeutige Kennung für diese Host-Gruppe.
	name	string	(erforderlich) Name der Host-Gruppe.
hosts			Wurzelelement für Hosts.
graphs			Wurzelelement für Host-Diagramme.

Hosts

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      name: 'MySQL server'
      groups:
        - name: Databases
      interfaces: (see table below)
      items: (see table below)
      discovery_rules: (see table below)
      tags:
        - tag: location
          value: Riga
      macros:
        (...)
        - macro: '{$MYSQL.DSN}'
          value: 192.0.2.1
        (...)
        - macro: '{$MYSQL.PASSWORD}'
          type: SECRET_TEXT
          description: 'MySQL user password.'
        (...)
      valuemaps: (see table below)
  (...)
```

Element	Type	Description
host	string	(erforderlich) Eindeutiger Hostname.
name	string	Sichtbarer Hostname.
description	text	Hostbeschreibung.
monitored_by	string	Wie der Host überwacht wird. Mögliche Werte:[1] SERVER (0, Standard), PROXY (1) oder PROXY_GROUP (2).
proxy		Stammelement für Proxy.
name	string	(erforderlich) Name des Proxys (falls vorhanden), der den Host überwacht.
proxy_group		Stammelement für Proxy-Gruppe.
name	string	(erforderlich) Name der Proxy-Gruppe (falls vorhanden), die zur Überwachung des Hosts verwendet wird.
status	string	Hoststatus. Mögliche Werte:[1] ENABLED (0, Standard), DISABLED (1).
ipmi_authtype	string	Authentifizierungstyp der IPMI-Sitzung. Mögliche Werte:[1] DEFAULT (-1, Standard), NONE (0), MD2 (1), MD5 (2), STRAIGHT (4), OEM (5), RMCP_PLUS (6).

Element	Type	Description
ipmi_privilege	string	Berechtigungsstufe der IPMI-Sitzung. Mögliche Werte:[1] CALLBACK (1), USER (2, Standard), OPERATOR (3), ADMIN (4), OEM (5).
ipmi_username	string	Benutzername für IPMI-Prüfungen.
ipmi_password	string	Passwort für IPMI-Prüfungen.
templates		Stammelement für verknüpfte Vorlagen.
name	string	(erforderlich) Vorlagenname.
groups		Stammelement für Host-Gruppen, zu denen der Host gehört.
name	string	(erforderlich) Name der Host-Gruppe.
interfaces		Stammelement für Host-Schnittstellen.
items		Stammelement für Host-Datenpunkte.
discovery_rules		Stammelement für Low-Level-Discovery-Regeln des Hosts.
http_tests		Stammelement für Webszenarien des Hosts.
tags		Stammelement für Host-Tags.
tag	string	(erforderlich) Tag-Name.
value	string	Tag-Wert.
macros		Stammelement für Host-Makros.
macro	string	(erforderlich) Name des Benutzermakros.
type	string	Typ des Benutzermakros. Mögliche Werte:[1] TEXT (0, Standard), SECRET_TEXT (1), VAULT (2).
value	string	Wert des Benutzermakros.
description	string	Beschreibung des Benutzermakros.
inventory		Stammelement für das Host-Inventar.
<inventory_property>	string	Inventareigenschaft. Alle Eigenschaften haben ihr jeweiliges Element (`type`, `name`, `os` usw.; siehe beispielsweise Exportformat).
inventory_mode	string	Inventarmodus. Mögliche Werte:[1] DISABLED (-1), MANUAL (0, Standard), AUTOMATIC (1).
valuemaps		Stammelement für Host-Wertabbildungen.

Note:
Siehe auch: Host object (verweisen Sie auf die entsprechende Eigenschaft mit passendem Namen).

Host-Schnittstellen

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      interfaces:
        - ip: 192.0.2.1
          interface_ref: if1
      (...)
```

Element	Type	Beschreibung
default	string	Gibt an, ob dies die primäre Host-Schnittstelle ist. Beachten Sie, dass es auf einem Host nur eine primäre Schnittstelle eines Typs geben kann. Mögliche Werte:[1] NO (0), YES (1, Standard).
type	string	Schnittstellentyp. Mögliche Werte:[1] ZABBIX (1, Standard), SNMP (2), IPMI (3), JMX (4).
useip	string	Gibt an, ob IP als Schnittstelle für die Verbindung zum Host verwendet werden soll (andernfalls wird DNS verwendet). Mögliche Werte:[1] NO (0), YES (1, Standard).
ip	string	(erforderlich für IP-Verbindungen) IP-Adresse (IPv4 oder IPv6).
dns	string	(erforderlich für DNS-Verbindungen) DNS-Name.
port	string	Portnummer.
details		Stammelement für Schnittstellendetails.
version	string	Diese SNMP-Version verwenden. Mögliche Werte:[1] SNMPV1 (1), SNMP_V2C (2, Standard), SNMP_V3 (3).
community	string	(erforderlich für SNMPv1- und SNMPv2-Datenpunkte) SNMP-Community.

Element	Type	Beschreibung
max_repetitions	string	Maximaler Wiederholungswert für native SNMP-Bulk-Anfragen (GetBulkRequest-PDUs). Unterstützt für SNMPv2- und SNMPv3-Datenpunkte (`discovery[]`- und `walk[]`-Datenpunkte). Standard: 10.
contextname	string	SNMPv3-Kontextname. Unterstützt für SNMPv3-Datenpunkte.
securityname	string	SNMPv3-Sicherheitsname. Unterstützt für SNMPv3-Datenpunkte.
securitylevel	string	SNMPv3-Sicherheitsstufe. Unterstützt für SNMPv3-Datenpunkte. Mögliche Werte:[1] NOAUTHNOPRIV (0, Standard), AUTHNOPRIV (1), AUTHPRIV (2).
authprotocol	string	SNMPv3-Authentifizierungsprotokoll. Unterstützt für SNMPv3-Datenpunkte. Mögliche Werte:[1] MD5 (0, Standard), SHA1 (1), SHA224 (2), SHA256 (3), SHA384 (4), SHA512 (5).
authpassphrase	string	SNMPv3-Authentifizierungs-Passphrase. Unterstützt für SNMPv3-Datenpunkte.
privprotocol	string	SNMPv3-Datenschutzprotokoll. Unterstützt für SNMPv3-Datenpunkte. Mögliche Werte:[1] DES (0, Standard), AES128 (1), AES192 (2), AES256 (3), AES192C (4), AES256C (5).
privpassphrase	string	SNMPv3-Datenschutz-Passphrase. Unterstützt für SNMPv3-Datenpunkte.
bulk	string	Bulk-Anfragen für SNMP verwenden. Mögliche Werte:[1] NO (0), YES (1, Standard).
interface_ref	string	Referenzname der Schnittstelle, der in Datenpunkten verwendet werden soll (Format: `if<N>`).

Note:
Siehe auch: Host interface object (verweisen Sie auf die entsprechende Eigenschaft mit demselben Namen).

Host-Datenpunkte

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      items:
        (...)
        - name: 'Binlog cache disk use'
          type: DEPENDENT
          key: mysql.binlog_cache_disk_use
          value_type: FLOAT
          description: 'Anzahl der Transaktionen, die einen temporären Festplatten-Cache verwendet haben,
          preprocessing: (siehe Tabelle unten)
          master_item:
            key: 'mysql.get_status_variables["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
          tags:
            - tag: component
              value: cache
        (...)
        - name: 'Buffer pool utilization'
          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: 'Verhältnis der verwendeten Seiten zur Gesamtzahl der Seiten im Buffer Pool.'
      tags:
        - tag: component
          value: memory
      triggers: (siehe Tabelle unten)
    (...)
    - name: Uptime
      type: DEPENDENT
      key: mysql.uptime
      units: uptime
      description: 'Anzahl der Sekunden, die der Server aktiv war.'
      preprocessing: (siehe Tabelle unten)
      master_item:
        key: 'mysql.get_status_variables["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
      tags:
        - tag: component
          value: application
      triggers: (siehe Tabelle unten)
    (...)
```

Element	Typ	Beschreibung
name	string	(erforderlich) Name des Datenpunkts.
type	string	Datenpunkttyp. Mögliche Werte:[1] ZABBIX_PASSIVE (0, Standard), TRAP (2), SIMPLE (3), INTERNAL (5), ZABBIX_ACTIVE (7), EXTERNAL (10), ODBC (11), IPMI (12), SSH (13), TELNET (14), CALCULATED (15), JMX (16), SNMP_TRAP (17), DEPENDENT (18), HTTP_AGENT (19), SNMP_AGENT (20), ITEM_TYPE_SCRIPT (21), ITEM_TYPE_BROWSER (22), ITEM_TYPE_NESTED (23), TELEMETRY_QUERY (24).
snmp_oid	string	(erforderlich für SNMP_AGENT-Datenpunkte) SNMP-Objekt-ID.
key	string	(erforderlich) Schlüssel des Datenpunkts.
delay	string	Aktualisierungsintervall des Datenpunkts. Standard: 1m. Der Wert ist für TRAP-Datenpunkte immer 0.
history	string	Zeitraum (unter Verwendung eines Zeitsuffixes, Benutzermakros oder LLD-Makros), für den Verlaufsdaten gespeichert werden sollen. Standard: 31d.
trends	string	Zeitraum (unter Verwendung eines Zeitsuffixes, Benutzermakros oder LLD-Makros), für den Trenddaten gespeichert werden sollen. Standard: 365d.
status	string	Status des Datenpunkts. Mögliche Werte:[1] ENABLED (0, Standard), DISABLED (1).
value_type	string	Typ des empfangenen Werts. Mögliche Werte:[1] FLOAT (0), CHAR (1), LOG (2), UNSIGNED (3, Standard), TEXT (4), BINARY (5), JSON (6).
allowed_hosts	string	Kommagetrennte Liste der IP-Adressen von Hosts, die Daten für den Datenpunkt senden dürfen. Unterstützt für TRAP- und HTTP_AGENT-Datenpunkte.
units	string	Einheiten des empfangenen Werts (bps, B usw.).
params	text	Zusätzliche Parameter abhängig vom Typ des Datenpunkts (ausgeführtes Skript für SSH- und TELNET-Datenpunkte; SQL-Abfrage für ODBC-Datenpunkte; Formel für CALCULATED-Datenpunkte; Skript für ITEM_TYPE_SCRIPT- und ITEM_TYPE_BROWSER-Datenpunkte).
ipmi_sensor	string	IPMI-Sensor. Unterstützt für IPMI-Datenpunkte.
authtype	string	Authentifizierungstyp. Unterstützt für SSH- und HTTP_AGENT-Datenpunkte. Mögliche Werte für SSH-Datenpunkte:[1] PASSWORD (0, Standard), PUBLIC_KEY (1). Mögliche Werte für HTTP_AGENT-Datenpunkte:[1] NONE (0, Standard), BASIC (1), NTLM (2), Kerberos (3) oder Digest (4).

Element	Typ	Beschreibung
username	string	(erforderlich für SSH- und TELNET-Datenpunkte) Benutzername für die Authentifizierung. Unterstützt für SIMPLE-, ODBC-, JMX- und HTTP_AGENT-Datenpunkte. Bei Verwendung für JMX-Datenpunkte muss auch `password` (siehe unten) angegeben werden, oder beide Elemente müssen leer bleiben.
password	string	(erforderlich für SSH- und TELNET-Datenpunkte) Passwort für die Authentifizierung. Unterstützt für SIMPLE-, ODBC-, JMX- und HTTP_AGENT-Datenpunkte. Bei Verwendung für JMX-Datenpunkte muss auch `username` (siehe oben) angegeben werden, oder beide Elemente müssen leer bleiben.
publickey	string	(erforderlich für SSH-Datenpunkte) Name der Datei mit dem öffentlichen Schlüssel.
privatekey	string	(erforderlich für SSH-Datenpunkte) Name der Datei mit dem privaten Schlüssel.
description	text	Beschreibung des Datenpunkts.
inventory_link	string	Inventarfeld des Hosts, das durch den Datenpunkt befüllt wird. Mögliche Werte:[1] NONE (0), ALIAS (4) usw. (unter Host-Inventar finden Sie die unterstützten Felder).
valuemap		Stammelement für Wertzuordnungen des Datenpunkts.
name	string	(erforderlich) Name der für den Datenpunkt zu verwendenden Wertzuordnung.
logtimefmt	string	Format der Zeit in Logeinträgen. Unterstützt für Datenpunkte des Werttyps LOG.
preprocessing		Stammelement für die Wertvorverarbeitung des Datenpunkts.
step		Stammelement für Vorverarbeitungsschritte von Host-Datenpunktwerten.
interface_ref	string	Referenz auf die Host-Schnittstelle (Format: `if<N>`).
jmx_endpoint	string	JMX-Endpunkt. Unterstützt für JMX-Datenpunkte.
master_item		(erforderlich für DEPENDENT-Datenpunkte) Stammelement für den Master-Datenpunkt des abhängigen Datenpunkts.
key	string	(erforderlich) Schlüssel des Master-Datenpunkts des abhängigen Datenpunkts.
timeout	string	Zeitüberschreitung für die Abfrage von Daten des Datenpunkts. Unterstützt für die Liste der Datenpunkttypen unter Timeouts.
url	string	(erforderlich für HTTP_AGENT-Datenpunkte) URL-Zeichenfolge.
query_fields		Stammelement für Abfrageparameter. Unterstützt für HTTP_AGENT-Datenpunkte.
name	string	(erforderlich für HTTP_AGENT-Datenpunkte) Name des Abfrageparameters.
value	string	Wert des Abfrageparameters. Unterstützt für HTTP_AGENT-Datenpunkte.
parameters		Stammelement für benutzerdefinierte Parameter. Unterstützt für ITEM_TYPE_SCRIPT- und ITEM_TYPE_BROWSER-Datenpunkte.
name	string	(erforderlich für ITEM_TYPE_SCRIPT- und ITEM_TYPE_BROWSER-Datenpunkte) Name des benutzerdefinierten Parameters.
value	string	Wert des benutzerdefinierten Parameters. Unterstützt für ITEM_TYPE_SCRIPT- und ITEM_TYPE_BROWSER-Datenpunkte.
posts	string	Daten des HTTP(S)-Anforderungstexts. Unterstützt für HTTP_AGENT-Datenpunkte.
status_codes	string	Durch Kommas getrennte Bereiche erforderlicher HTTP-Statuscodes. Unterstützt für HTTP_AGENT-Datenpunkte.
follow_redirects	string	Weiterleitungen der Antwort während der Datenabfrage verfolgen. Unterstützt für HTTP_AGENT-Datenpunkte. Mögliche Werte:[1] NO (0), YES (1, Standard).
post_type	string	Typ des Textkörpers der POST-Daten. Unterstützt für HTTP_AGENT-Datenpunkte. Mögliche Werte:[1] RAW (0, Standard), JSON (2), XML (3).
http_proxy	string	HTTP(S)-Proxy-Verbindungszeichenfolge. Unterstützt für HTTP_AGENT-Datenpunkte.
headers		Stammelement für HTTP(S)-Anforderungsheader. Unterstützt für HTTP_AGENT-Datenpunkte.
name	string	(erforderlich für HTTP_AGENT-Datenpunkte) Name des Headers.
value	string	(erforderlich für HTTP_AGENT-Datenpunkte) Wert des Headers.
retrieve_mode	string	Welcher Teil der Antwort gespeichert werden soll. Unterstützt für HTTP_AGENT-Datenpunkte. Mögliche Werte:[1] BODY (0, Standard), HEADERS (1), BOTH (2).

Element	Typ	Beschreibung
request_method	string	Typ der Anforderungsmethode. Unterstützt für HTTP_AGENT-Datenpunkte. Mögliche Werte:[1] GET (0, Standard), POST (1), PUT (2), HEAD (3).
output_format	string	Verarbeitung der Antwort. Unterstützt für HTTP_AGENT-Datenpunkte. Mögliche Werte:[1] RAW (0, Standard), JSON (1).
allow_traps	string	Erlauben, dass der Wert auf dieselbe Weise wie bei einem Trapper-Datenpunkt befüllt wird. Unterstützt für HTTP_AGENT-Datenpunkte. Mögliche Werte:[1] NO (0, Standard), YES (1).
ssl_cert_file	string	Pfad zur Datei mit dem öffentlichen SSL-Schlüssel. Unterstützt für HTTP_AGENT-Datenpunkte.
ssl_key_file	string	Pfad zur Datei mit dem privaten SSL-Schlüssel. Unterstützt für HTTP_AGENT-Datenpunkte.
ssl_key_password	string	Passwort für die SSL-Schlüsseldatei. Unterstützt für HTTP_AGENT-Datenpunkte.
verify_peer	string	Gibt an, ob überprüft werden soll, dass das Zertifikat des Hosts authentisch ist. Unterstützt für HTTP_AGENT-Datenpunkte. Mögliche Werte:[1] NO (0, Standard), YES (1).
verify_host	string	Gibt an, ob überprüft werden soll, dass der Hostname der Verbindung mit dem im Zertifikat des Hosts übereinstimmt. Unterstützt für HTTP_AGENT-Datenpunkte. Mögliche Werte:[1] NO (0, Standard), YES (1).
time_shift	string	Wie weit zurück (vor der aktuellen Zeit) der Zeitraum endet, den der Datenpunkt abfragt. Unterstützt für TELEMETRY_QUERY-Datenpunkte.
lookback_limit	string	Wie weit zurück (vor der aktuellen Zeit) der Datenpunkt Daten abfragt, wenn keine vorherige Abfrage vorhanden ist, von der aus fortgefahren werden kann (z. B. beim ersten Lauf des Datenpunkts), oder wenn die vorherige Abfrage älter als `delay` ist (z. B. weil der Zabbix-Server ausgefallen war). Unterstützt für TELEMETRY_QUERY-Datenpunkte.
granularity	string	Länge eines Abschnitts, in den der gesamte Zeitraum der Abfrage aufgeteilt wird. Unterstützt für TELEMETRY_QUERY-Datenpunkte.
query		(erforderlich für TELEMETRY_QUERY-Datenpunkte) Stammelement für die Telemetrieabfrage. Definiert, was der Datenpunkt aus der Datenquelle abfragt, wie die Daten gefiltert werden und wie das Ergebnis aggregiert wird.
signal_type	string	(erforderlich) Typ des Signals, das der Datenpunkt abfragt. Unterstützt für TELEMETRY_QUERY-Datenpunkte. Mögliche Werte:[1] TRACES (0), METRICS (1), LOGS (2).
metric_point_type	string	(erforderlich für den Signaltyp METRICS) Typ des Metrikpunkts, den der Datenpunkt abfragt. Unterstützt für TELEMETRY_QUERY-Datenpunkte. Mögliche Werte:[1] SUM (0), GAUGE (1), HISTOGRAM (2), EXPONENTIAL_HISTOGRAM (3).
columns		(erforderlich) Stammelement für Telemetriedatenspalten, die der Datenpunkt abfragt. Dieses Element enthält die folgenden Elemente: `column (string)` - (erforderlich) Name der Telemetriedatenspalte, die der Datenpunkt abfragt. `attribute_key (string)` - (erforderlich für Spalten mit Schlüssel-Wert-Paaren) Name des Schlüssels, den der Datenpunkt innerhalb der Spalte mit Schlüssel-Wert-Paaren abfragt.

Element	Typ	Beschreibung
aggregated_columns		(erforderlich) Stammelement für aggregierte Telemetriedatenspalten, die der Datenpunkt abfragt. Dieses Element enthält die folgenden Elemente: `column` (`string`) - (erforderlich für die Funktionen MIN, MAX, AVG, SUM und PERCENTILE) Name der Telemetriedatenspalte, die der Datenpunkt abfragt. `function` (`string`) - (erforderlich) Funktion, die der Datenpunkt auf die Datensätze der Spalte anwendet. Mögliche Werte:[1] MIN (1), MAX (2), AVG (3), COUNT (4), SUM (5), PERCENTILE (8). `parameters` (`array`) - (erforderlich für PERCENTILE-Funktionen) Perzentil, das der Datenpunkt aus den Datensätzen der Spalte berechnet. `alias` (`string`) - (erforderlich) Eindeutiger Name für den berechneten Wert, der zur Identifizierung in dem Datenpunktwert verwendet wird.
filter		(erforderlich) Stammelement für den Filter der Telemetrieabfrage auf Spaltendatensätze in der Abfrage. Dieses Element enthält die folgenden Elemente: `evaltype` (`string`) - (erforderlich) Auswertungsmethode für Filterbedingungen. Mögliche Werte:[1] AND_OR (0), AND (1), OR (2), FORMULA (3). `formula` (`string`) - (erforderlich für die Auswertungsmethode FORMULA) Benutzerdefinierter Ausdruck zur Auswertung der Bedingungen von Filtern mit einem benutzerdefinierten Ausdruck. `conditions` (`string`) - (erforderlich) Stammelement für die Filterbedingungen der Telemetriedaten, die ein Datensatz in einer Telemetriedatenspalte erfüllen muss. Das Element `conditions` enthält die folgenden Elemente: `column` (`string`) - (erforderlich) Name der Telemetriedatenspalte, auf die die Filterbedingungen angewendet werden. `attribute_key` (`string`) - (erforderlich für Spalten mit Schlüssel-Wert-Paaren) Name des Schlüssels, auf den die Bedingung innerhalb der Spalte mit Schlüssel-Wert-Paaren angewendet wird. `operator` (`string`) - (erforderlich) Operator der Filterbedingung. Mögliche Werte für Spalten mit Einzelwerten:[1] EQUAL (0), NOT_EQUAL (1), LIKE (2), NOT_LIKE (3). Mögliche Werte für Spalten mit Schlüssel-Wert-Paaren:[1] EQUAL (0), NOT_EQUAL (1), EXISTS (12). `formulaid` (`string`) - (erforderlich für die Auswertungsmethode FORMULA) Beliebige eindeutige ID, die verwendet wird, um in einem benutzerdefinierten Ausdruck auf die Bedingung zu verweisen. `value` (`string`) - (erforderlich für die Operationen EQUAL, NOT_EQUAL, LIKE und NOT LIKE) Wert, mit dem verglichen werden soll.
tags		Stammelement für Tags des Datenpunkts.
tag	string	(erforderlich) Tagname.
value	string	Tagwert.
triggers		Stammelement für Auslöser von Host-Datenpunkten.

Note:
Siehe auch: Item object (siehe die relevante Eigenschaft mit übereinstimmendem Namen).

Host-Datenpunkt-Werte-Vorverarbeitungsschritte

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      items:
        (...)
        - name: 'Binlog cache disk use'
          (...)
          preprocessing:
            - type: JSONPATH
              parameters:
```

```
              - $.Binlog_cache_disk_use
          - type: DISCARD_UNCHANGED_HEARTBEAT
            parameters:
              - 6h
        (...)
```

Element	Type	Beschreibung
type	string	(erforderlich) Der Typ des Vorverarbeitungsschritts für den Datenpunktwert. Mögliche Werte:[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, berechnet: empfangener Wert - vorheriger Wert), CHANGE_PER_SECOND (10, berechnet: (empfangener Wert - vorheriger Wert)/(aktuelle Zeit - Zeitpunkt der letzten Prüfung)), 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		(erforderlich) Wurzelelement für Parameter des Vorverarbeitungsschritts für den Datenpunktwert.
parameter	string	Einzelner Parameter des Vorverarbeitungsschritts für den Datenpunktwert.
error_handler	string	Aktionstyp, der im Falle eines Fehlers beim Vorverarbeitungsschritt verwendet wird. Mögliche Werte:[1] ORIGINAL_ERROR (0, Standard), DISCARD_VALUE (1), CUSTOM_VALUE (2), CUSTOM_ERROR (3).
error_handler_params	string	Parameter des Fehlerbehandlers.

Note:
Siehe auch: Objekt zur Vorverarbeitung von Datenpunkten (verweisen Sie auf die entsprechende Eigenschaft mit demselben Namen).

Host-Auslöser

```
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 hat in den letzten 30 Minuten keine Daten für Datenpunkte erhalten.'
              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: 'Die MySQL-Laufzeit beträgt weniger als 10 Minuten.'
              tags:
                - tag: scope
```

```
                value: notice
        (...)
```

Element	Type	Description
uuid	string	(erforderlich) Eindeutige Kennung für diesen Auslöser.
expression	string	(erforderlich) Ausdrück des Auslösers.
recovery_mode	string	Grundlage für die Generierung von OK-Ereignissen. Mögliche Werte:[1] EXPRESSION (0, Standard), RECOVERY_EXPRESSION (1), NONE (2).
recovery_expression	string	Wiederherstellungsausdruck des Auslösers.
correlation_mode	string	Korrelationsmodus (keine Ereigniskorrelation oder Ereigniskorrelation nach Tag). Mögliche Werte:[1] DISABLED (0, Standard), TAG_VALUE (1).
correlation_tag	string	Der Tag-Name, der für die Ereigniskorrelation verwendet werden soll.
name	string	(erforderlich) Name des Auslösers.
event_name	string	Ereignisname.
opdata	string	Betriebsdaten.
url_name	string	Bezeichnung für die dem Auslöser zugeordnete URL.
url	string	Dem Auslöser zugeordnete URL.
status	string	Status des Auslösers. Mögliche Werte:[1] ENABLED (0, Standard), DISABLED (1).
priority	string	Schweregrad des Auslösers. Mögliche Werte:[1] NOT_CLASSIFIED (0, Standard), INFO (1), WARNING (2), AVERAGE (3), HIGH (4), DISASTER (5).
description	text	Beschreibung des Auslösers.
type	string	Typ der Ereignisgenerierung (ein einzelnes Problemereignis oder mehrere Problemereignisse). Mögliche Werte:[1] SINGLE (0, Standard), MULTIPLE (1).
manual_close	string	Manuelles Schließen von Problemereignissen. Mögliche Werte:[1] NO (0, Standard), YES (1).
dependencies		Stammelement für Abhängigkeiten.
name	string	(erforderlich) Name des Abhängigkeits-Auslösers.
expression	string	(erforderlich) Ausdruck des Abhängigkeits-Auslösers.
recovery_expression	string	Wiederherstellungsausdruck des Abhängigkeits-Auslösers.
tags		Stammelement für Auslöser-Tags.
tag	string	(erforderlich) Tag-Name.
value	string	Tag-Wert.

Note:
Siehe auch: Trigger object (siehe die entsprechende Eigenschaft mit demselben Namen).

Host Low-Level-Discovery-Regeln

```
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:
Die meisten Elemente von Host Low-Level-Discovery-Regeln sind dieselben wie bei Host-Datenpunkten. Die folgende Tabelle beschreibt die Elemente, die sich von Host-Datenpunkten unterscheiden.

Element	Type	Beschreibung
type	string	Datenpunkttyp. Mögliche Werte:[1] ZABBIX_PASSIVE (0, Standard), 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	(erforderlich) Der Schlüssel der Low-Level-Discovery-Regel.
filter		Stammelement für Filter von Host Low-Level-Discovery-Regeln.
lifetime	string	Zeitraum (unter Verwendung von Sekunden, Zeitsuffix oder Benutzermakro), nach dem nicht mehr erkannte Ressourcen gelöscht werden. Standard: 7d.
lifetime_type	string	Szenario zum Löschen verlorener LLD-Ressourcen. Mögliche Werte: DELETE_NEVER, DELETE_IMMEDIATELY, DELETE_AFTER.
enabled_lifetime	string	Zeitraum (unter Verwendung von Sekunden, Zeitsuffix oder Benutzermakro), nach dem nicht mehr erkannte Ressourcen deaktiviert werden.
enabled_lifetime_type	string	Szenario zum Deaktivieren verlorener LLD-Ressourcen. Mögliche Werte: DISABLE_NEVER, DISABLE_IMMEDIATELY, DISABLE_AFTER.
item_prototypes		Stammelement für Host-Datenpunktprototyp-Elemente, die dieselben sind wie bei Host-Datenpunkten.
trigger_prototypes		Stammelement für Host-Auslöserprototyp-Elemente, die dieselben sind wie bei Host-Datenpunkt-Auslösern.
graph_prototypes		Stammelement für Host-Diagrammprototypen, die dieselben sind wie bei Host-Diagrammen.
host_prototypes		Stammelement für Hostprototypen, die dieselben sind wie bei Hosts.
parent_discovery_rule		Stammelement für die übergeordnete Low-Level-Discovery-Regel (oder Regelvorlage) des Low-Level-Discovery-Regelprototyps. Diese Eigenschaft kennzeichnet, dass es sich um einen LLD-Regelprototyp handelt, der direktes Kind der referenzierten Regel/Regelvorlage ist.
key	string	(erforderlich) Der Schlüssel der übergeordneten Low-Level-Discovery-Regel (oder Regelvorlage).
master_item	string	(erforderlich für DEPENDENT-Regeln) Stammelement für den Master-Datenpunkt der abhängigen Regel.
lld_macro_paths		Stammelement für Makropfade von Low-Level-Discovery-Regeln.
lld_macro	string	(erforderlich) Name des Makros der Low-Level-Discovery-Regel.
path	string	(erforderlich) Selektor für den Wert, der dem entsprechenden Makro zugewiesen wird.
preprocessing		Stammelement für die Vorverarbeitung von Werten der Low-Level-Discovery-Regel.
step		Stammelement für Elemente von Vorverarbeitungsschritten der Low-Level-Discovery-Regel, die dieselben sind wie bei Vorverarbeitungsschritten von Host-Datenpunktwerten, jedoch mit weniger möglichen Werten. Siehe auch: LLD-Regel-Vorverarbeitungsobjekt.

Element	Type	Beschreibung
overrides		Stammelement für Überschreibungsregeln von Low-Level-Discovery-Regeln.
name	string	(erforderlich) Eindeutiger Name der Überschreibung.
step	string	(erforderlich) Eindeutige Ordnungsnummer der Überschreibung.
stop	string	Die Verarbeitung weiterer Überschreibungen beenden, wenn zutreffend.
filter		Stammelement für Filterelemente von Überschreibungsregeln für Low-Level-Discovery-Regeln, die dieselben sind wie bei Filtern von Host Low-Level-Discovery-Regeln.
operations		Stammelement für Überschreibungsoperationen von Host Low-Level-Discovery-Regeln.

Note:
Siehe auch: LLD-Regelobjekt (verweisen Sie auf die entsprechende Eigenschaft mit übereinstimmendem Namen).

Host Low-Level-Discovery-Regelfilter

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      discovery_rules:
        - name: 'Database discovery'
          (...)
          filter:
            evaltype: AND
            conditions:
              - macro: '{#DATABASE}'
                value: '{$MYSQL.DBNAME.MATCHES}'
              - macro: '{#DATABASE}'
                value: '{$MYSQL.DBNAME.NOT_MATCHES}'
                operator: NOT_MATCHES_REGEX
          (...)
```

Element	Type	Beschreibung
evaltype	string	Überschreibt die Auswertungsmethode der Filterbedingungen. Mögliche Werte:[1] AND_OR (0, Standard), AND (1), OR (2), FORMULA (3).
formula	string	Benutzerdefinierte Berechnungsformel für Filterbedingungen.
conditions		Wurzelelement für Filterbedingungen.
macro	string	(erforderlich) Der Name des Low-Level-Discovery-Makros, auf das die Prüfung angewendet wird.
value	string	Wert, mit dem verglichen wird.
operator	string	Bedingungsoperator. Mögliche Werte:[1] MATCHES_REGEX (8, Standard), NOT_MATCHES_REGEX (9).
formulaid	string	(erforderlich) Beliebige eindeutige ID, die zum Verweisen auf eine Bedingung aus dem benutzerdefinierten Ausdruck verwendet wird. Darf nur Großbuchstaben enthalten. Die ID muss vom Benutzer beim Ändern der Filterbedingungen angegeben werden, wird jedoch bei einer späteren Abfrage neu generiert.

Note:
Siehe auch: LLD-Regelfilterobjekt (verweist auf die entsprechende Eigenschaft mit demselben Namen).

Host Low-Level-Discovery-Regel: Überschreibungsoperationen

Element	Type	Description
operationobject	string	Objekt, auf das die Operation angewendet wird. Mögliche Werte:[1] ITEM_PROTOTYPE (0), TRIGGER_PROTOTYPE (1), GRAPH_PROTOTYPE (2), HOST_PROTOTYPE (3), DISCOVERY_PROTOTYPE (4).

Element	Type	Description
operator	string	Operator der Überschreibungsbedingung. Mögliche Werte:[1] EQUAL (1), NOT_EQUAL (2), LIKE (3), NOT_LIKE (4), REGEXP (5), NOT_REGEXP (6).
value	string	Ein regulärer Ausdruck oder eine Zeichenfolge für den Operator der Überschreibungsbedingung.
status	string	Status des Objekts nach der Überschreibungsoperation.
discover	string	Gibt an, ob das Objekt als Ergebnis der Erkennung hinzugefügt wird.
delay	string	Aktualisierungsintervall, das für die Datenpunkt-Prototypen nach der Überschreibungsoperation festgelegt wird.
history	string	Speicherzeitraum für den Verlauf, der für die Datenpunkt-Prototypen nach der Überschreibungsoperation festgelegt wird.
trends	string	Speicherzeitraum für Trends, der für die Datenpunkt-Prototypen nach der Überschreibungsoperation festgelegt wird.
severity	string	Schweregrad des Auslöser-Prototyps, der nach der Überschreibungsoperation festgelegt wird.
tags		Wurzelelement für die Tags, die für das Objekt nach der Überschreibungsoperation festgelegt werden.
tag	string	(erforderlich) Tag-Name.
value	string	Tag-Wert.
templates		Wurzelelement für die Vorlagen, die mit dem Host-Prototyp nach der Überschreibungsoperation verknüpft werden.
name	string	(erforderlich) Vorlagenname.
inventory_mode	string	Inventarmodus des Host-Prototyps, der nach der Überschreibungsoperation festgelegt wird.

Note:
Siehe auch: LLD rule override operation object (verweist auf die entsprechende Eigenschaft mit demselben Namen).

Host-Webszenarien

Element	Type	Description
uuid	string	(erforderlich) Eindeutige Kennung für dieses Webszenario.
name	string	(erforderlich) Name des Webszenarios.
delay	string	Häufigkeit der Ausführung des Webszenarios (in Sekunden, Zeitsuffix oder Benutzermakro). Standard: 1m.
attempts	integer	Anzahl der Versuche für die Ausführung der Webszenario-Schritte. Mögliche Werte: 1-10 (Standard: 1).
agent	string	Client-Agent. Zabbix gibt vor, der ausgewählte Browser zu sein. Dies ist nützlich, wenn eine Website für verschiedene Browser unterschiedliche Inhalte zurückgibt. Standard: Zabbix.
http_proxy	string	Proxy, der vom Webszenario verwendet wird, angegeben als: `http://[username[:password]@]proxy.example.com[:port]`
variables		Stammelement für Webszenario-Variablen, die in Szenarioschritten verwendet werden können.
name	string	(erforderlich) Variablenname.
value	text	(erforderlich) Variablenwert.
headers		Stammelement für HTTP-Header, die bei einer Anfrage gesendet werden. Die Header sollten mit derselben Syntax aufgeführt werden, wie sie im HTTP-Protokoll erscheinen würden.
name	string	(erforderlich) Headername.
value	text	(erforderlich) Headerwert.
status	string	Status des Webszenarios. Mögliche Werte:[1] ENABLED (0, Standard), DISABLED (1).
authentication	string	Authentifizierungsmethode. Mögliche Werte:[1] NONE (0, Standard), BASIC (1), NTLM (2), Kerberos (3) oder Digest (4).
http_user	string	Benutzername, der für die BASIC-(HTTP)-, NTLM-, Kerberos- oder Digest-Authentifizierung verwendet wird.

Element	Type	Description
http_password	string	Passwort, das für die BASIC-(HTTP)-, NTLM-, Kerberos- oder Digest-Authentifizierung verwendet wird.
verify_peer	string	SSL-Zertifikat des Webservers überprüfen. Mögliche Werte:[1] NO (0, Standard), YES (1).
verify_host	string	Prüfen, ob das Feld *Common Name* oder das Feld *Subject Alternate Name* des Webserver-Zertifikats übereinstimmt. Mögliche Werte:[1] NO (0, Standard), YES (1).
ssl_cert_file	string	Name der SSL-Zertifikatsdatei, die für die Client-Authentifizierung verwendet wird (muss im PEM-Format vorliegen).
ssl_key_file	string	Name der SSL-Privatschlüsseldatei, die für die Client-Authentifizierung verwendet wird (muss im PEM-Format vorliegen).
ssl_key_password	string	Passwort der SSL-Privatschlüsseldatei.
steps		(erforderlich) Stammelement für Host-Webszenario-Schritte.
tags		Stammelement für Webszenario-Tags.
tag	string	(erforderlich) Tag-Name.
value	string	Tag-Wert.

Note:
Siehe auch: Webszenario-Objekt (verweisen Sie auf die entsprechende Eigenschaft mit passendem Namen).

Host Web-Szenario-Schritte

Element	Type	Description
name	string	(erforderlich) Name des Web-Szenario-Schritts.
url	string	(erforderlich) URL zur Überwachung.
query_fields		Stammelement für Abfrageparameter (ein Array von HTTP-Feldern, die der URL beim Ausführen einer Anfrage hinzugefügt werden).
name	string	(erforderlich) Name des Abfrageparameters.
value	string	Wert des Abfrageparameters.
posts		Stammelement für HTTP-POST-Variablen (ein String (Raw-Post-Daten) oder ein Array von HTTP-Feldern (Formulardaten)).
name	string	(erforderlich) Name des Post-Felds.
value	string	(erforderlich) Wert des Post-Felds.
variables		Stammelement für Variablen auf Schritt-Ebene (Makros), die nach diesem Schritt angewendet werden sollen. Wenn der Variablenwert ein Präfix 'regex:' hat, wird sein Wert aus den von diesem Schritt zurückgegebenen Daten gemäß dem regulären Ausdrucksmuster extrahiert, das auf das Präfix 'regex:' folgt
name	string	(erforderlich) Variablenname.
value	text	(erforderlich) Variablenwert.
headers		Stammelement für HTTP-Header, die beim Ausführen einer Anfrage gesendet werden.
name	string	(erforderlich) Headername.
value	text	(erforderlich) Headerwert.
follow_redirects	string	HTTP-Weiterleitungen folgen. Mögliche Werte:[1] NO (0), YES (1, Standard).
retrieve_mode	string	Abrufmodus der HTTP-Antwort. Mögliche Werte:[1] BODY (0, Standard), HEADERS (1), BOTH (2).
timeout	string	Zeitüberschreitung (in Sekunden, Zeitsuffix oder Benutzermakro) für die Ausführung des Schritts. Standard: 15s.
required	string	Text, der in der Antwort vorhanden sein muss (wird ignoriert, wenn leer).
status_codes	string	Eine durch Kommas getrennte Liste akzeptierter HTTP-Statuscodes (z. B. 200-201,210-299; wird ignoriert, wenn leer).

Note:
Siehe auch: Objekt "Web-Szenario-Schritt" (verweisen Sie auf die entsprechende Eigenschaft mit demselben Namen).

Host-Diagramme

```
zabbix_export:
  (...)
  graphs:
    - name: 'MySQL: Bandwidth'
      graph_items: (siehe Tabelle unten)
    (...)
```

Element	Typ	Beschreibung
uuid	string	Eindeutige Kennung des Diagramms.
name	string	(erforderlich) Name des Diagramms.
width	integer	Breite des Diagramms in Pixeln. Wird für die Vorschau und für Kreis-/aufgesplittete Diagramme verwendet. Mögliche Werte:[1] 20-65535 (Standard: 900).
height	integer	Höhe des Diagramms in Pixeln. Wird für die Vorschau und für Kreis-/aufgesplittete Diagramme verwendet. Mögliche Werte:[1] 20-65535 (Standard: 900).
yaxismin	double	Minimaler Wert der Y-Achse. Unterstützt für den FIXED-Minimalwert der Y-Achse. Standard: 0.
yaxismax	double	Maximaler Wert der Y-Achse. Unterstützt für den FIXED-Minimalwert der X-Achse. Standard: 0.
show_work_period	string	Nicht-Arbeitszeiten hervorheben. Unterstützt für NORMAL- und STACKED-Diagramme. Mögliche Werte:[1] NO (0), YES (1, Standard).
show_triggers	string	Einfache Auslöserwerte als Linie anzeigen. Unterstützt für NORMAL- und STACKED-Diagramme. Mögliche Werte:[1] NO (0), YES (1, Standard).
type	string	Diagrammtyp. Mögliche Werte:[1] NORMAL (0, Standard), STACKED (1), PIE (2), EXPLODED (3).
show_legend	string	Diagrammlegende anzeigen. Mögliche Werte:[1] NO (0), YES (1, Standard).
show_3d	string	3D-Stil aktivieren. Unterstützt für NORMAL- und STACKED-Diagramme. Mögliche Werte:[1] NO (0, Standard), YES (1).
percent_left	double	Perzentillinie für die linke Achse anzeigen. Unterstützt für NORMAL-Diagramme. Standard: 0.
percent_right	double	Perzentillinie für die rechte Achse anzeigen. Unterstützt für NORMAL-Diagramme. Standard: 0.
ymin_type_1	string	Minimalwert der Y-Achse. Unterstützt für NORMAL- und STACKED-Diagramme. Mögliche Werte:[1] CALCULATED (0, Standard), FIXED (1), ITEM (2).
ymin_item_1		(erforderlich, wenn `ymin_type_1` auf `ITEM` gesetzt ist) Wurzelelement für Details einzelner Datenpunkte.
host	string	(erforderlich) Host des Datenpunkts.
key	string	(erforderlich) Schlüssel des Datenpunkts.
ymax_type_1	string	Maximalwert der Y-Achse. Unterstützt für NORMAL- und STACKED-Diagramme. Mögliche Werte:[1] CALCULATED (0, Standard), FIXED (1), ITEM (2).
ymax_item_1		(erforderlich, wenn `ymax_type_1` auf `ITEM` gesetzt ist) Wurzelelement für Details einzelner Datenpunkte.
host	string	(erforderlich) Host des Datenpunkts.
key	string	(erforderlich) Schlüssel des Datenpunkts.
graph_items		(erforderlich) Wurzelelement für Host-Diagrammelemente.

Note:
Siehe auch: Graph object (verweisen Sie auf die entsprechende Eigenschaft mit demselben Namen).

Host-Graph-Elemente

```
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
    (...)
```

Element	Type	Beschreibung
sortorder	integer	Zeichenreihenfolge. Der kleinere Wert wird zuerst gezeichnet. Kann verwendet werden, um Linien oder Bereiche hinter (oder vor) einem anderen zu zeichnen.
drawtype	string	Zeichenstil des Graph-Elements. Unterstützt für NORMAL-Graphs. Mögliche Werte:[1] SINGLE_LINE (0, Standard), FILLED_REGION (1), BOLD_LINE (2), DOTTED_LINE (3), DASHED_LINE (4), GRADIENT_LINE (5).
color	string	Farbe des Elements (6 Zeichen, Hex).
yaxisside	string	Seite des Graphen, auf der die Y-Skala des Graph-Elements gezeichnet wird. Unterstützt für NORMAL- und STACKED-Graphs. Mögliche Werte:[1] LEFT (0, Standard), RIGHT (1).
calc_fnc	string	Daten, die gezeichnet werden, wenn für einen Datenpunkt mehr als ein Wert vorhanden ist. Mögliche Werte:[1] MIN (1), AVG (2, Standard), MAX (4), ALL (7; Minimum, Durchschnitt und Maximum; unterstützt für einfache Graphen), LAST (9, unterstützt für Kreis-/aufgeschnittene Kreisdiagramme).
type	string	Graph-Elementtyp. Mögliche Werte:[1] SIMPLE (0, Standard), GRAPH_SUM (2; der Wert des Datenpunkts stellt den gesamten Kreis dar; unterstützt für Kreis-/aufgeschnittene Kreisdiagramme).
item		(erforderlich) Einzelner Datenpunkt.
host	string	(erforderlich) Host des Datenpunkts.
key	string	(erforderlich) Schlüssel des Datenpunkts.

Note:
Siehe auch: Graph item object (siehe die entsprechende Eigenschaft mit passendem Namen).

Host-Wertzuordnungen

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      valuemaps:
        - name: Example value map
          mappings:
            - value: '1'
              newvalue: Example value
  (...)
```

Element	Type	Description
uuid	string	(erforderlich) Eindeutige Kennung für diese Wertzuordnung.

Element	Type	Description
name	string	(erforderlich) Name der Wertzuordnung.
mapping		Wurzelelement für Zuordnungen.
type	string	Typ der Zuordnungsübereinstimmung. Mögliche Werte:[1] EQUAL (0, Standard), GREATER_OR_EQUAL (2), LESS_OR_EQUAL (3), IN_RANGE (4), REGEXP (5), DEFAULT (6).
value	string	Ursprünglicher Wert.
newvalue	string	(erforderlich) Wert, dem der ursprüngliche Wert zugeordnet wird.

Note:
Siehe auch: Value map object (siehe die entsprechende Eigenschaft mit passendem Namen).

Fußnoten [1] API-Integerwerte in Klammern, zum Beispiel ENABLED (0), werden nur als Referenz angegeben. Weitere Informationen finden Sie auf der verlinkten API-Objektseite im Tabelleneintrag oder am Ende jedes Abschnitts.

Netzwerkkarten

Übersicht

Die Netzwerkkarte export enthält:

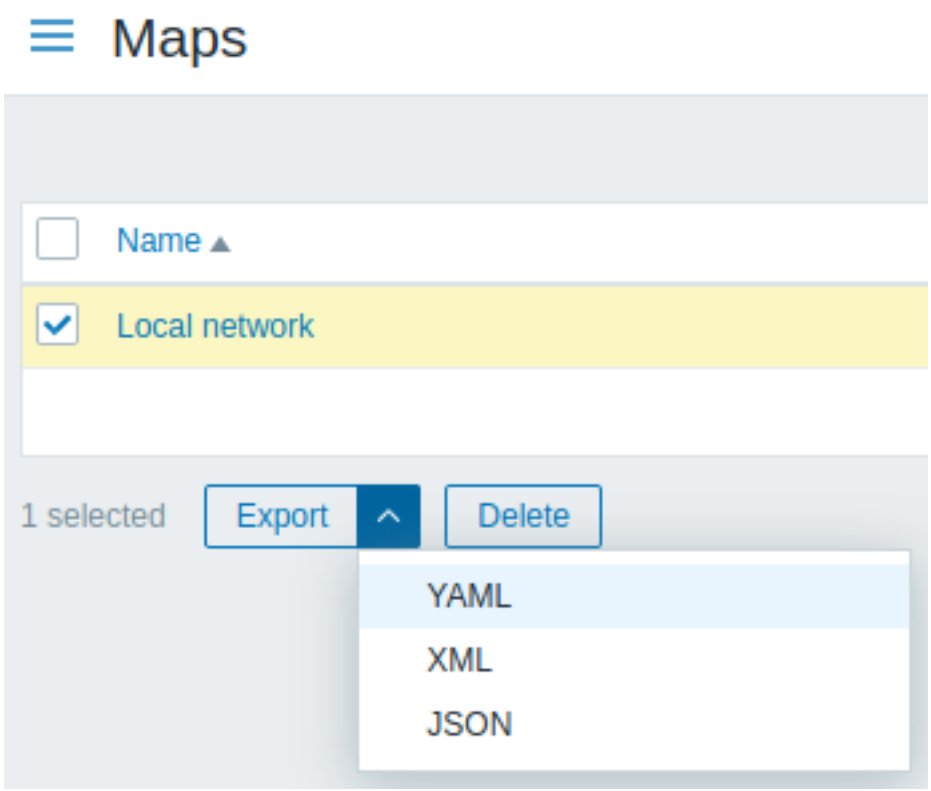
- Alle zugehörigen Bilder
- Die Kartenstruktur (alle Karteneinstellungen, alle enthaltenen Elemente mit ihren Einstellungen, Kartenverknüpfungen und Statusanzeigen der Kartenverknüpfungen)

Warning:
Alle Hostgruppen, Hosts, Auslöser, anderen Karten oder anderen Elemente, die mit der exportierten Karte verknüpft sein können, werden nicht exportiert. Wenn also mindestens eines der Elemente fehlt, auf die sich die Karte bezieht, schlägt der Import fehl.

Exportieren

Gehen Sie zum Exportieren von Netzwerkkarten wie folgt vor:

1. Gehen Sie zu *Monitoring* > *Maps*.
2. Markieren Sie die Kontrollkästchen der zu exportierenden Netzwerkkarten.
3. Klicken Sie unterhalb der Liste auf *Export*, um im standardmäßigen YAML-Format zu exportieren, oder klicken Sie auf den Pfeil, um ein anderes Exportformat auszuwählen.



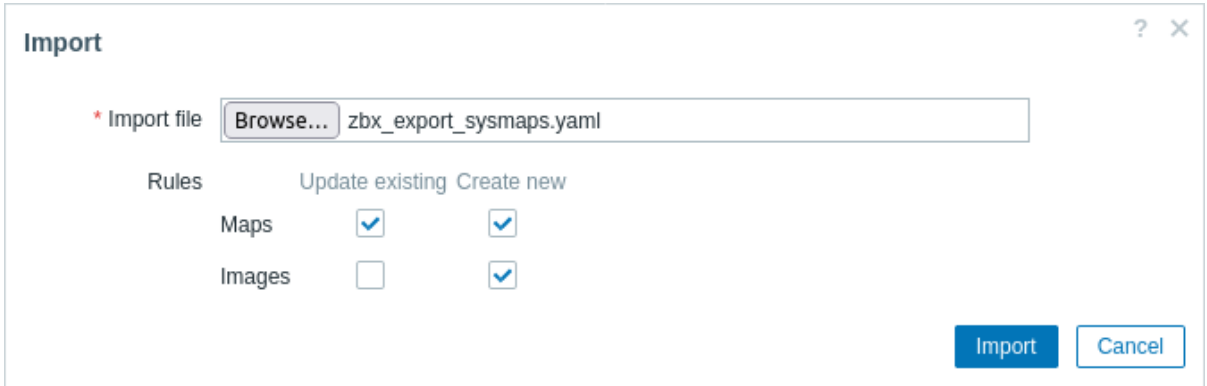
Abhängig vom ausgewählten Format werden die Karten in eine lokale Datei mit einem Standardnamen exportiert:

- `zabbix_export_maps.yaml` - beim YAML-Export;
- `zabbix_export_maps.xml` - beim XML-Export;
- `zabbix_export_maps.json` - beim JSON-Export.

Importieren

Gehen Sie zum Importieren von Netzwerkkarten wie folgt vor:

1. Gehen Sie zu *Monitoring* > *Maps*.
2. Klicken Sie oben rechts auf *Import*.
3. Wählen Sie die Importdatei aus.
4. Aktivieren Sie die Kontrollkästchen der Importregeln, die Sie anwenden möchten (siehe unten).
5. Klicken Sie unten rechts im Konfigurationsformular auf *Import*.



Importregeln:

Regel	Beschreibung
Update existing	Vorhandene Netzwerkkarten werden anhand der Daten aus der Importdatei aktualisiert. Andernfalls werden sie nicht aktualisiert.
Create new	Neue Netzwerkkarten werden anhand der Daten aus der Importdatei erstellt. Andernfalls werden sie nicht erstellt.

Wenn Sie beide Optionen für Netzwerkkarten deaktivieren und die entsprechenden Optionen für Bilder aktivieren, werden nur Bilder importiert. Der Bildimport ist nur für Benutzer mit der Rolle *Super admin* verfügbar.

Sie können Änderungen an Netzwerkkarten während des Imports nicht überprüfen. Eine Erfolgs- oder Fehlermeldung des Imports wird im Frontend angezeigt.

Warning:
Wenn ein vorhandenes Bild ersetzt wird, wirkt sich dies auf alle Netzwerkkarten aus, die dieses Bild verwenden.

Exportformat

Export nach YAML:

```yaml
zabbix_export:
  version: '8.0'
  images:
    - name: Zabbix_server_3D_(128)
      imagetype: '1'
      encodedImage: iVBOR...5CYII=
  maps:
    - name: 'Lokales Netzwerk'
      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: {  }
```

Exportierte Elemente

Exportierte Elemente werden in der folgenden Tabelle erläutert.

Element	Type	Description
images		Stammelement für Bilder.
name	string	Eindeutiger Bildname.
imagetype	integer	Bildtyp. Mögliche Werte: 1 - Bild; 2 - Hintergrund.
encodedImage	string	Base64-kodiertes Bild.
maps		Stammelement für maps.

Maps

Element	Type	Beschreibung
name	string	Eindeutiger Kartenname.
width	integer	Kartenbreite in Pixeln.
height	integer	Kartenhöhe in Pixeln.
label_type	integer	Beschriftungstyp des Kartenelements. Mögliche Werte: 0 - Beschriftung; 1 - Host-IP-Adresse; 2 - Elementname; 3 - Nur Status; 4 - Nichts.
label_location	integer	Standardmäßige Position der Beschriftung des Kartenelements. Mögliche Werte: 0 - Unten; 1 - Links; 2 - Rechts; 3 - Oben.

Element	Type	Beschreibung
highlight	integer	Symbolhervorhebung für aktive Auslöser und Host-Status aktivieren. Mögliche Werte: 0 - Nein; 1 - Ja.
expandproblem	integer	Problem-Auslöser für Elemente mit einem einzelnen Problem anzeigen. Mögliche Werte: 0 - Nein; 1 - Ja.
markelements	integer	Kartenelemente hervorheben, deren Status sich kürzlich geändert hat. Mögliche Werte: 0 - Nein; 1 - Ja.
show_unack	integer	Problemanzeige. Mögliche Werte: 0 - Anzahl aller Probleme; 1 - Anzahl der nicht bestätigten Probleme; 2 - Anzahl der bestätigten und nicht bestätigten Probleme getrennt.
severity_min	integer	Minimale Auslöser-Schweregradstufe, die standardmäßig auf der Karte angezeigt wird. Mögliche Werte: 0 - Nicht klassifiziert; 1 - Information; 2 - Warnung; 3 - Durchschnittlich; 4 - Hoch; 5 - Desaster.
show_suppressed	integer	Probleme anzeigen, die andernfalls aufgrund von Host-Wartung unterdrückt (nicht angezeigt) würden. Mögliche Werte: 0 - Nein; 1 - Ja.
grid_size	integer	Zellgröße eines Kartenrasters in Pixeln. Wird unterstützt, wenn `grid_show` auf 0 gesetzt ist. Mögliche Werte: 20, 40, 50, 75 oder 100.
grid_show	integer	Raster in der Kartenkonfiguration anzeigen. Mögliche Werte: 0 - Ja; 1 - Nein.
grid_align	integer	Symbole in der Kartenkonfiguration automatisch ausrichten. Mögliche Werte: 0 - Ja; 1 - Nein.
label_format	integer	Erweiterte Beschriftungskonfiguration verwenden. Mögliche Werte: 0 - Nein; 1 - Ja.
label_type_host	integer	Beschriftung als Host-Beschriftung anzeigen. Wird unterstützt, wenn `label_format` auf 1 gesetzt ist. Mögliche Werte: 0 - Beschriftung; 1 - Host-IP-Adresse; 2 - Elementname; 3 - Nur Status; 4 - Nichts; 5 - Benutzerdefinierte Beschriftung.
label_type_hostgroup	integer	Beschriftung als Hostgruppe-Beschriftung anzeigen. Wird unterstützt, wenn `label_format` auf 1 gesetzt ist. Mögliche Werte: 0 - Beschriftung; 2 - Elementname; 3 - Nur Status; 4 - Nichts; 5 - Benutzerdefinierte Beschriftung.

Element	Type	Beschreibung
label_type_trigger	integer	Beschriftung als Auslöser-Beschriftung anzeigen. Wird unterstützt, wenn `label_format` auf 1 gesetzt ist. Mögliche Werte: 0 - Beschriftung; 2 - Elementname; 3 - Nur Status; 4 - Nichts; 5 - Benutzerdefinierte Beschriftung.
label_type_map	integer	Beschriftung als Karten-Beschriftung anzeigen. Wird unterstützt, wenn `label_format` auf 1 gesetzt ist. Mögliche Werte: 0 - Beschriftung; 2 - Elementname; 3 - Nur Status; 4 - Nichts; 5 - Benutzerdefinierte Beschriftung.
label_type_image	integer	Beschriftung als Bild-Beschriftung anzeigen. Wird unterstützt, wenn `label_format` auf 1 gesetzt ist. Mögliche Werte: 0 - Beschriftung; 2 - Elementname; 4 - Nichts; 5 - Benutzerdefinierte Beschriftung.
label_string_host	string	Benutzerdefinierte Beschriftung für Host-Elemente. Wird unterstützt, wenn `label_type_host` auf 5 gesetzt ist.
label_string_hostgroup	string	Benutzerdefinierte Beschriftung für Hostgruppen-Elemente. Wird unterstützt, wenn `label_type_hostgroup` auf 5 gesetzt ist.
label_string_trigger	string	Benutzerdefinierte Beschriftung für Auslöser-Elemente. Wird unterstützt, wenn `label_type_trigger` auf 5 gesetzt ist.
label_string_map	string	Benutzerdefinierte Beschriftung für Karten-Elemente. Wird unterstützt, wenn `label_type_map` auf 5 gesetzt ist.
label_string_image	string	Benutzerdefinierte Beschriftung für Bildelemente. Wird unterstützt, wenn `label_type_image` auf 5 gesetzt ist.
expand_macros	integer	Makros in Beschriftungen in der Kartenkonfiguration erweitern. Mögliche Werte: 0 - Nein; 1 - Ja.
background		Wurzelelement für das Hintergrundbild (falls vorhanden). Wird unterstützt, wenn `imagetype` auf 2 gesetzt ist.
name	string	Name des Hintergrundbilds.
iconmap		Wurzelelement für die Symbolzuordnung (falls vorhanden).
name	string	Name der Symbolzuordnung.
urls		Wurzelelement für URLs, die von Karten oder einzelnen Kartenelementen verwendet werden.
name	string	Linkname.
url	string	Link-URL.
elementtype	integer	Typ des Kartenelements, zu dem der Link gehört. Mögliche Werte: 0 - Host; 1 - Karte; 2 - Auslöser; 3 - Hostgruppe; 4 - Bild.
selements		Wurzelelement für Kartenselemente.
shapes		Wurzelelement für Kartenformen.
type	integer	Formtyp. Mögliche Werte: 0 - Rechteck; 1 - Ellipse.
x	integer	X-Koordinate der Form in Pixeln.
y	integer	Y-Koordinate der Form in Pixeln.

Element		Type	Beschreibung
	width	integer	Breite der Form.
	height	integer	Höhe der Form.
	text	string	Text innerhalb der Form.
	font	integer	Schriftart des Textes. Mögliche Werte: 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	Schriftgröße in Pixeln.
	font_color	string	Schriftfarbe in hexadezimalem Code dargestellt.
	text_halign	integer	Horizontale Ausrichtung des Textes. Mögliche Werte: 0 - Zentriert; 1 - Links; 2 - Rechts.
	text_valign	integer	Vertikale Ausrichtung des Textes. Mögliche Werte: 0 - Mitte; 1 - Oben; 2 - Unten.
	border_type	integer	Rahmentyp der Form. Mögliche Werte: 0 - Kein; 1 - Dicke Linie; 2 - Gepunktete Linie; 3 - Gestrichelte Linie.
	border_width	integer	Rahmenbreite in Pixeln.
	border_color	string	Rahmenfarbe in hexadezimalem Code dargestellt.
	background_color	string	Hintergrundfarbe (Füllfarbe) in hexadezimalem Code dargestellt.
	zindex	integer	Wert zur Reihenfolge aller Formen und Linien (z-index).
lines			Wurzelelement für Kartenlinien.
	x1	integer	X-Koordinate von Punkt 1 der Linie in Pixeln.
	y1	integer	Y-Koordinate von Punkt 1 der Linie in Pixeln.
	x2	integer	X-Koordinate von Punkt 2 der Linie in Pixeln.
	y2	integer	Y-Koordinate von Punkt 2 der Linie in Pixeln.
	line_type	integer	Linientyp. Mögliche Werte: 0 - Kein; 1 - Dicke Linie; 2 - Gepunktete Linie; 3 - Gestrichelte Linie.
	line_width	integer	Linienbreite in Pixeln.
	line_color	string	Linienfarbe in hexadezimalem Code dargestellt.
	zindex	integer	Wert zur Reihenfolge aller Formen und Linien (z-index).
links			Wurzelelement für Verknüpfungen zwischen Kartenelementen.
	drawtype	integer	Link-Stil. Mögliche Werte: 0 - Linie; 2 - Dicke Linie; 3 - Gepunktete Linie; 4 - Gestrichelte Linie.

Element	Type	Beschreibung
color	string	Linkfarbe (6 Zeichen, hexadezimal).
label	string	Link-Beschriftung.
selementid1	id	ID eines zu verbindenden Elements.
selementid2	id	ID des anderen zu verbindenden Elements.
linktriggers		Wurzelelement für Link-Statusindikatoren.

Note:
Siehe auch: Map object (siehe die entsprechende Eigenschaft mit passendem Namen).

Map-Elemente

Element	Type	Description
elementtype	integer	Typ des Map-Elements. Mögliche Werte: 0 - Host; 1 - Map; 2 - Auslöser; 3 - Host group; 4 - Image.
elements		Wurzelelement für Zabbix-Entitäten (host, host group, map usw.), die auf der Map dargestellt werden. Alle Entitäten haben ihr jeweiliges Element (`host` usw.; siehe beispielsweise Export format).
label	string	Symbolbeschriftung.
label_location	integer	Position der Beschriftung. Mögliche Werte: -1 - Standard der Map verwenden; 0 - Unten; 1 - Links; 2 - Rechts; 3 - Oben.
x	integer	Position auf der X-Achse.
y	integer	Position auf der Y-Achse.
elementsubtype	integer	Elementuntertyp. Unterstützt, wenn `elementtype` auf 3 gesetzt ist. Mögliche Werte: 0 - Einzelne Host group; 1 - Alle Host groups.
areatype	integer	Bereichsgröße. Unterstützt, wenn `elementtype` auf 1 gesetzt ist. Mögliche Werte: 0 - Wie die gesamte Map; 1 - Benutzerdefinierte Größe.
width	integer	Breite des Bereichs. Unterstützt, wenn `areatype` auf 1 gesetzt ist.
height	integer	Höhe des Bereichs. Unterstützt, wenn `areatype` auf 1 gesetzt ist.
viewtype	integer	Algorithmus zur Platzierung des Bereichs. Unterstützt, wenn `elementsubtype` auf 1 gesetzt ist. Mögliche Werte: 0 - Gleichmäßig im Bereich platzieren.
use_iconmap	integer	Icon-Zuordnung für dieses Element verwenden. Relevant nur, wenn die Icon-Zuordnung auf Map-Ebene aktiviert ist. Mögliche Werte: 0 - Nein; 1 - Ja.
selementid	id	Eindeutige Datensatz-ID des Elements.
icon_off		Wurzelelement für das Bild, das verwendet wird, wenn sich das Element im Zustand `OK` befindet.

Element	Type	Description
name	string	Eindeutiger Bildname.
icon_on		Wurzelelement für das Bild, das verwendet wird, wenn sich das Element im Zustand `Problem` befindet.
name	string	Eindeutiger Bildname.
icon_disabled		Wurzelelement für das Bild, das verwendet wird, wenn das Element deaktiviert ist.
name	string	Eindeutiger Bildname.
icon_maintenance		Wurzelelement für das Bild, das verwendet wird, wenn sich das Element in Wartung befindet.
name	string	Eindeutiger Bildname.
urls		Wurzelelement für URLs, die von Maps oder einzelnen Map-Elementen verwendet werden.
name	string	Name des Links.
url	string	URL des Links.
evaltype	integer	Auswertungstyp für Tags.
tags		Wurzelelement für Problem-Tags (für Host- und Host group-Elemente). Wenn Tags angegeben sind, werden auf der Map nur Probleme mit diesen Tags angezeigt.
tag	string	Tag-Name.
value	string	Tag-Wert.
operator	integer	Operator.

Note:
Siehe auch: Map element object (verweisen Sie auf die entsprechende Eigenschaft mit passendem Namen).

Statusindikatoren für Kartenverknüpfungen

Element	Type	Description
drawtype	integer	Linienstil, wenn sich der Auslöser im Status 'Problem' befindet. Mögliche Werte: 0 - Linie; 2 - Fette Linie; 3 - Gepunktete Linie; 4 - Gestrichelte Linie.
color	string	Farbe der Verknüpfung (6 Zeichen, hexadezimal), wenn sich der Auslöser im Status 'Problem' befindet.
trigger		Wurzelelement für den Auslöser, der zur Anzeige des Verknüpfungsstatus verwendet wird.
description	string	Name des Auslösers.
expression	string	Ausdruck des Auslösers.
recovery_expression	string	Wiederherstellungsausdruck des Auslösers.

Note:
Siehe auch: Map link trigger object (siehe die entsprechende Eigenschaft mit demselben Namen).

Medientypen

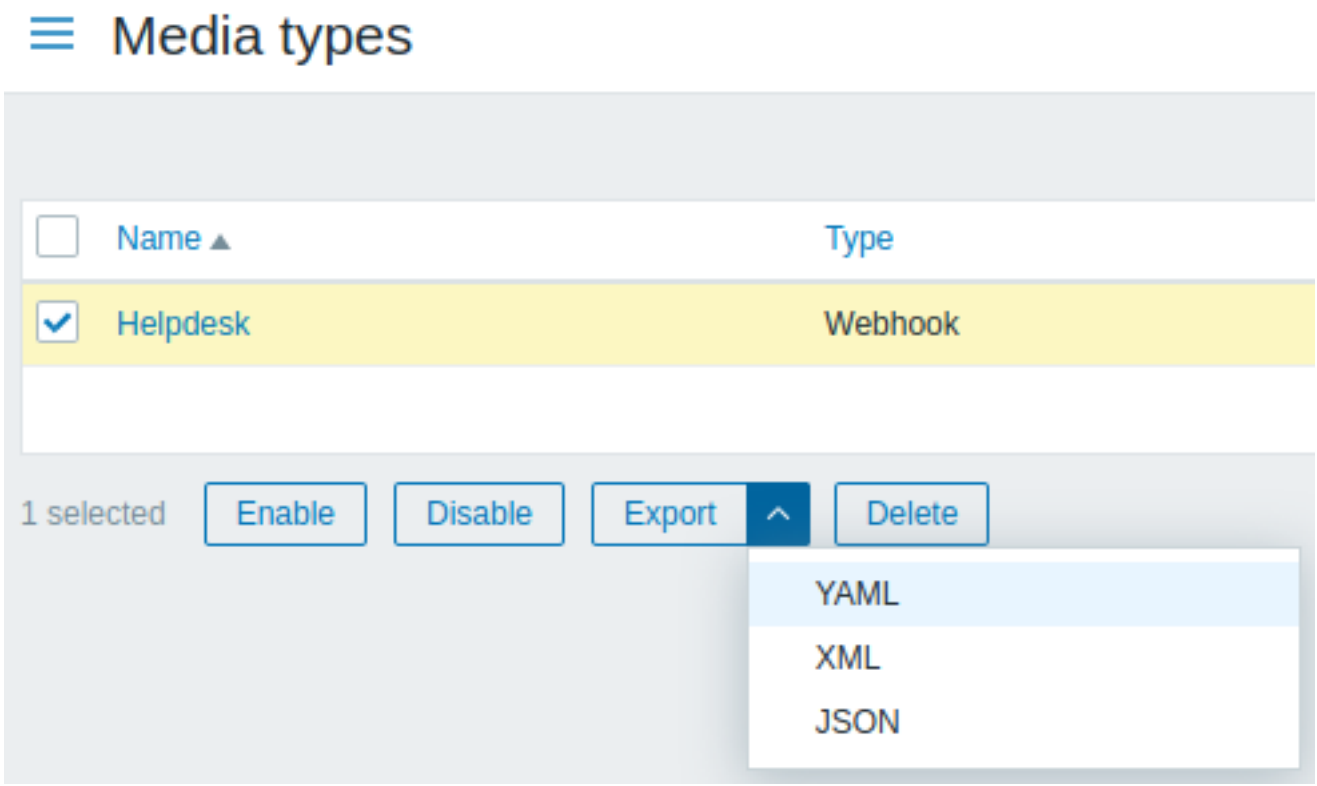
Übersicht

Medientypen werden zusammen mit allen zugehörigen Objekten und Objektbeziehungen exportiert.

Exportieren

Gehen Sie zum Exportieren von Medientypen wie folgt vor:

1. Gehen Sie zu *Alerts* > *Media types*.
2. Markieren Sie die Kontrollkästchen der zu exportierenden Medientypen.
3. Klicken Sie unterhalb der Liste auf *Export*, um im standardmäßigen YAML-Format zu exportieren, oder klicken Sie auf den Pfeil, um ein anderes Exportformat auszuwählen.



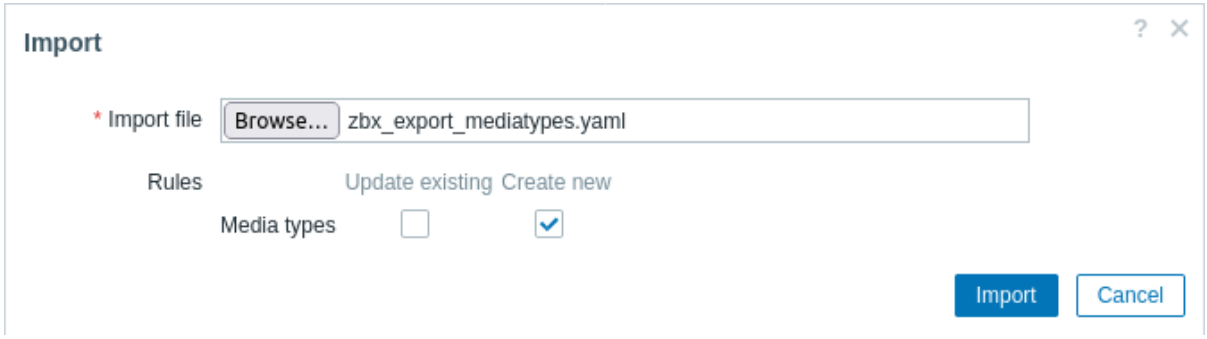
Je nach ausgewähltem Format werden Medientypen in eine lokale Datei mit einem Standardnamen exportiert:

- `zabbix_export_mediatypes.yaml` - beim YAML-Export;
- `zabbix_export_mediatypes.xml` - beim XML-Export;
- `zabbix_export_mediatypes.json` - beim JSON-Export.

Importieren

Gehen Sie zum Importieren von Medientypen wie folgt vor:

1. Gehen Sie zu *Alerts* > *Media types*.
2. Klicken Sie oben rechts auf *Import*.
3. Wählen Sie die Importdatei aus.
4. Aktivieren Sie die Kontrollkästchen der Importregeln, die Sie anwenden möchten (siehe unten).
5. Klicken Sie unten rechts im Konfigurationsformular auf *Import*.



Importregeln:

Regel	Beschreibung
Update existing	Aktivieren Sie dieses Kontrollkästchen, um vorhandene Medientypen anhand der Daten aus der Importdatei zu aktualisieren.
Create new	Aktivieren Sie dieses Kontrollkästchen, um neue Medientypen anhand der Daten aus der Importdatei zu erstellen. Wenn die Importdatei einen bereits vorhandenen Medientyp enthält (z. B. *Email*), wird kein neuer erstellt und der vorhandene Medientyp wird ebenfalls nicht aktualisiert.

Während des Imports können Sie Änderungen an Medientypen nicht überprüfen. Eine Erfolgs- oder Fehlermeldung des Imports wird im Frontend angezeigt.

Exportformat

Export nach YAML:

```yaml
zabbix_export:
  version: '8.0'
  media_types:
    - name: Pushover
      type: WEBHOOK
      parameters:
        - name: endpoint
          value: 'https://api.pushover.net/1/messages.json'
        - name: eventid
          value: '{EVENT.ID}'
        - name: event_nseverity
          value: '{EVENT.NSEVERITY}'
        - name: event_source
          value: '{EVENT.SOURCE}'
        - name: event_value
          value: '{EVENT.VALUE}'
        - name: expire
          value: '1200'
        - name: message
          value: '{ALERT.MESSAGE}'
        - name: priority_average
          value: '0'
        - name: priority_default
          value: '0'
        - name: priority_disaster
          value: '0'
        - name: priority_high
          value: '0'
        - name: priority_information
          value: '0'
        - name: priority_not_classified
          value: '0'
        - name: priority_warning
          value: '0'
        - name: retry
          value: '60'
        - name: title
          value: '{ALERT.SUBJECT}'
        - name: token
          value: '<PUSHOVER TOKEN HERE>'
        - name: triggerid
          value: '{TRIGGER.ID}'
        - name: url
          value: '{$ZABBIX.URL}'
        - name: url_title
          value: Zabbix
        - name: user
          value: '{ALERT.SENDTO}'
      status: DISABLED
      max_sessions: '0'
      script: |
        try {
            var params = JSON.parse(value),
                request = new HttpRequest(),
                data,
                response,
                severities = [
                    {name: 'not_classified', color: '#97AAB3'},
                    {name: 'information', color: '#7499FF'},
                    {name: 'warning', color: '#FFC859'},
```

```
            {name: 'average', color: '#FFA059'},
            {name: 'high', color: '#E97659'},
            {name: 'disaster', color: '#E45959'},
            {name: 'resolved', color: '#009900'},
            {name: 'default', color: '#000000'}
        ],
        priority;

    if (typeof params.HTTPProxy === 'string' && params.HTTPProxy.trim() !== '') {
        request.setProxy(params.HTTPProxy);
    }

    if ([0, 1, 2, 3].indexOf(parseInt(params.event_source)) === -1) {
        throw 'Falscher Parameter "event_source" angegeben: "' + params.event_source + '".\nMuss 0
    }

    if (params.event_value !== '0' && params.event_value !== '1'
        && (params.event_source === '0' || params.event_source === '3')) {
        throw 'Falscher Parameter "event_value" angegeben: ' + params.event_value + '\nMuss 0 oder
    }

    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" sollte -2..2 sein';
    }

    if (params.event_source === '0' && isNaN(params.triggerid)) {
        throw 'Feld "triggerid" ist keine Zahl';
    }

    if (isNaN(params.eventid)) {
        throw 'Feld "eventid" ist keine Zahl';
    }

    if (typeof params.message !== 'string' || params.message.trim() === '') {
        throw 'Feld "message" darf nicht leer sein';
    }

    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 'Feld "retry" sollte eine Zahl mit einem Wert von mindestens 30 sein, wenn "pric
```

```
            }

            if (isNaN(params.expire) || params.expire > 10800) {
                throw 'Feld "expire" sollte eine Zahl mit einem Wert von höchstens 10800 sein, wenn "p
            }

            data.retry = params.retry;
            data.expire = params.expire;
        }

        data = JSON.stringify(data);
        Zabbix.log(4, '[ Pushover Webhook ] Sending request: ' + params.endpoint + '\n' + data);

        request.addHeader('Content-Type: application/json');
        response = request.post(params.endpoint, data);

        Zabbix.log(4, '[ Pushover Webhook ] Received response with status code ' + request.getStatus()

        if (response !== null) {
            try {
                response = JSON.parse(response);
            }
            catch (error) {
                Zabbix.log(4, '[ Pushover Webhook ] Failed to parse response received from 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 'Unknown error. Check debug log for more information.';
            }
        }

        return 'OK';
    }
    catch (error) {
        Zabbix.log(4, '[ Pushover Webhook ] Pushover notification failed: ' + error);
        throw 'Pushover notification failed: ' + error;
    }
description: |
    Bitte beachten Sie die Einrichtungsanleitung hier: https://git.zabbix.com/projects/ZBX/repos/zabbi

    Setzen Sie den Parameter token mit Ihrem Pushover-Anwendungsschlüssel.
    Wenn Sie Pushover-Medien einem Zabbix-Benutzer zuweisen, fügen Sie den Benutzer-Schlüssel in das P
message_templates:
    - event_source: TRIGGERS
      operation_mode: PROBLEM
      subject: 'Problem: {EVENT.NAME}'
      message: |
        Problem started at {EVENT.TIME} on {EVENT.DATE}
        Problem name: {EVENT.NAME}
        Host: {HOST.NAME}
        Severity: {EVENT.SEVERITY}
        Operational data: {EVENT.OPDATA}
        Original problem ID: {EVENT.ID}
        {TRIGGER.URL}
    - event_source: TRIGGERS
```

```
      operation_mode: RECOVERY
      subject: 'Resolved in {EVENT.DURATION}: {EVENT.NAME}'
      message: |
        Problem has been resolved at {EVENT.RECOVERY.TIME} on {EVENT.RECOVERY.DATE}
        Problem name: {EVENT.NAME}
        Problem duration: {EVENT.DURATION}
        Host: {HOST.NAME}
        Severity: {EVENT.SEVERITY}
        Original problem ID: {EVENT.ID}
        {TRIGGER.URL}
    - event_source: TRIGGERS
      operation_mode: UPDATE
      subject: 'Updated problem in {EVENT.AGE}: {EVENT.NAME}'
      message: |
        {USER.FULLNAME} {EVENT.UPDATE.ACTION} problem at {EVENT.UPDATE.DATE} {EVENT.UPDATE.TIME}.
        {EVENT.UPDATE.MESSAGE}

        Current problem state is {EVENT.STATUS}, age is {EVENT.AGE}, acknowledged: {EVENT.ACK.STATUS}.
    - event_source: DISCOVERY
      operation_mode: PROBLEM
      subject: 'Discovery: {DISCOVERY.DEVICE.STATUS} {DISCOVERY.DEVICE.IPADDRESS}'
      message: |
        Discovery rule: {DISCOVERY.RULE.NAME}

        Device IP: {DISCOVERY.DEVICE.IPADDRESS}
        Device DNS: {DISCOVERY.DEVICE.DNS}
        Device status: {DISCOVERY.DEVICE.STATUS}
        Device uptime: {DISCOVERY.DEVICE.UPTIME}

        Device service name: {DISCOVERY.SERVICE.NAME}
        Device service port: {DISCOVERY.SERVICE.PORT}
        Device service status: {DISCOVERY.SERVICE.STATUS}
        Device service uptime: {DISCOVERY.SERVICE.UPTIME}
    - event_source: AUTOREGISTRATION
      operation_mode: PROBLEM
      subject: 'Autoregistration: {HOST.HOST}'
      message: |
        Host name: {HOST.HOST}
        Host IP: {HOST.IP}
        Agent port: {HOST.PORT}
```

Exportierte Elemente

Die exportierten Elemente werden in der folgenden Tabelle erläutert.

Element	Type	Beschreibung
name	string	(erforderlich) Name des Medientyps.
type	string	(erforderlich) Vom Medientyp verwendeter Transport. Mögliche Werte:[1] EMAIL (0), SMS (1), SCRIPT (2), WEBHOOK (4).
status	string	Ob der Medientyp aktiviert ist. Mögliche Werte:[1] ENABLED (0, Standard), DISABLED (1)
max_sessions	integer	Die maximale Anzahl von Alarmen, die parallel verarbeitet werden können. Mögliche Werte für SMS:[1] 1 (Standard). Mögliche Werte für andere Medientypen:[1] 0-100 (wobei 0 - unbegrenzt).
attempts	integer	Die maximale Anzahl von Versuchen, einen Alarm zu senden. Mögliche Werte:[1] 1-10 (Standard: 3).
attempt_interval	string	Das Intervall (in Sekunden oder mit Zeit-Suffix) zwischen Wiederholungsversuchen. Mögliche Werte:[1] 0-60s (Standard: 10s).
description	string	Beschreibung des Medientyps.
message_templates		Stammelement für Nachrichtenvorlagen des Medientyps.
event_source	string	(erforderlich) Ereignisquelle. Mögliche Werte:[1] TRIGGERS (0), DISCOVERY (1), AUTOREGISTRATION (2), INTERNAL (3), SERVICE (4).

Element	Type	Beschreibung
operation_mode	string	Betriebsmodus. Mögliche Werte:[1] PROBLEM (0), RECOVERY (1), UPDATE (2).
subject	string	Betreff der Nachricht.
message	string	Nachrichtentext.

Note:
Siehe auch: Medientyp-Objekt (verweisen Sie auf die entsprechende Eigenschaft mit dem passenden Namen).

Email

Die folgenden zusätzlichen Elemente werden nur für den Medientyp *Email* exportiert.

Element	Type	Description
provider	string	Email-Anbieter.
smtp_server	string	SMTP-Server.
smtp_port	integer	Port des SMTP-Servers, zu dem eine Verbindung hergestellt werden soll. Standard: 25.
smtp_helo	string	SMTP-HELO.
smtp_email	string	E-Mail-Adresse, von der aus Benachrichtigungen gesendet werden.
smtp_security	string	Zu verwendende Sicherheitsstufe der SMTP-Verbindung. Mögliche Werte:[1] NONE (0, Standard), STARTTLS (1), SSL_OR_TLS (2).
smtp_verify_host	string	SSL-Hostprüfung für SMTP. Mögliche Werte:[1] NO (0, Standard), YES (1).
smtp_verify_peer	string	SSL-Peerprüfung für SMTP. Mögliche Werte:[1] NO (0, Standard), YES (1).
smtp_authentication	string	Zu verwendende SMTP-Authentifizierungsmethode. Mögliche Werte:[1] NONE (0, Standard), PASSWORD (1), OAUTH (2).
username	string	Benutzername.
password	string	Authentifizierungskennwort.
redirection_url	string	Zabbix-Frontend-URL, zu der die OAuth-Autorisierung zurückgeleitet wird.
client_id	string	Die im OAuth-Autorisierungsserver registrierte Client-ID.
authorization_url	string	OAuth-URL mit Parametern zum Abrufen von Zugriffs- und Aktualisierungstokens.
token_url	string	OAuth-URL zum Austausch des Autorisierungstokens gegen Zugriffs- und Aktualisierungstokens.
message_format	string	Nachrichtenformat. Mögliche Werte:[1] TEXT (0), HTML (1, Standard).

Note:
Siehe auch: Media type object (verweisen Sie auf die entsprechende Eigenschaft mit passendem Namen).

SMS

Die folgenden zusätzlichen Elemente werden nur für den Medientyp *SMS* exportiert.

Element	Type	Description
gsm_modem	string	(erforderlich) Name des seriellen Geräts des GSM-Modems.

Note:
Siehe auch: Media type object (beziehen Sie sich auf die entsprechende Eigenschaft mit demselben Namen).

Script

Die folgenden zusätzlichen Elemente werden nur für den Medientyp *Script* exportiert.

Element	Type	Description
script	string	(erforderlich) Skriptname.
name		
parameters		Stammelement für Skriptparameter.
sortorder	string	(erforderlich) Reihenfolge der Skriptparameter, die als Befehlszeilenargumente an das Skript übergeben werden.
value	string	Wert des Skriptparameters.

Note:
Siehe auch: Media type object (verweisen Sie auf die entsprechende Eigenschaft mit demselben Namen).

webhook

Die folgenden zusätzlichen Elemente werden nur für den Medientyp *webhook* exportiert.

Element	Type	Description
script	string	Script.
timeout	string	Zeitlimit für das HTTP-Anfrageintervall des Javascript-Skripts. Mögliche Werte:[1] 1-60s (Standard: 30s).
process_tags	string	Gibt an, ob zurückgegebene Tags verarbeitet werden sollen. Mögliche Werte:[1] NO (0, Standard), YES (1).
show_event_menu	string	Gibt an, ob ein Eintrag im Ereignismenü vorhanden ist, wenn das Makro {EVENT.TAGS.*} in den Feldern `event_menu_url` und `event_menu_name` erfolgreich aufgelöst wurde. Mögliche Werte:[1] NO (0, Standard), YES (1).
event_menu_url	string	URL des Eintrags im Ereignismenü. Unterstützt das Makro {EVENT.TAGS.*}.
event_menu_name	string	Name des Eintrags im Ereignismenü. Unterstützt das Makro {EVENT.TAGS.*}.
parameters		Stammelement für Parameter des webhook-Medientyps.
name	string	(erforderlich) Name des webhook-Parameters.
value	string	Wert des webhook-Parameters.

Note:
Siehe auch: Media type object (verweisen Sie auf die entsprechende Eigenschaft mit demselben Namen).

Fußnoten

[1] API-Integerwerte in Klammern, zum Beispiel ENABLED (0), werden nur als Referenz angegeben. Weitere Informationen finden Sie auf der verlinkten API-Objektseite im Tabelleneintrag oder am Ende jedes Abschnitts.

Dashboards

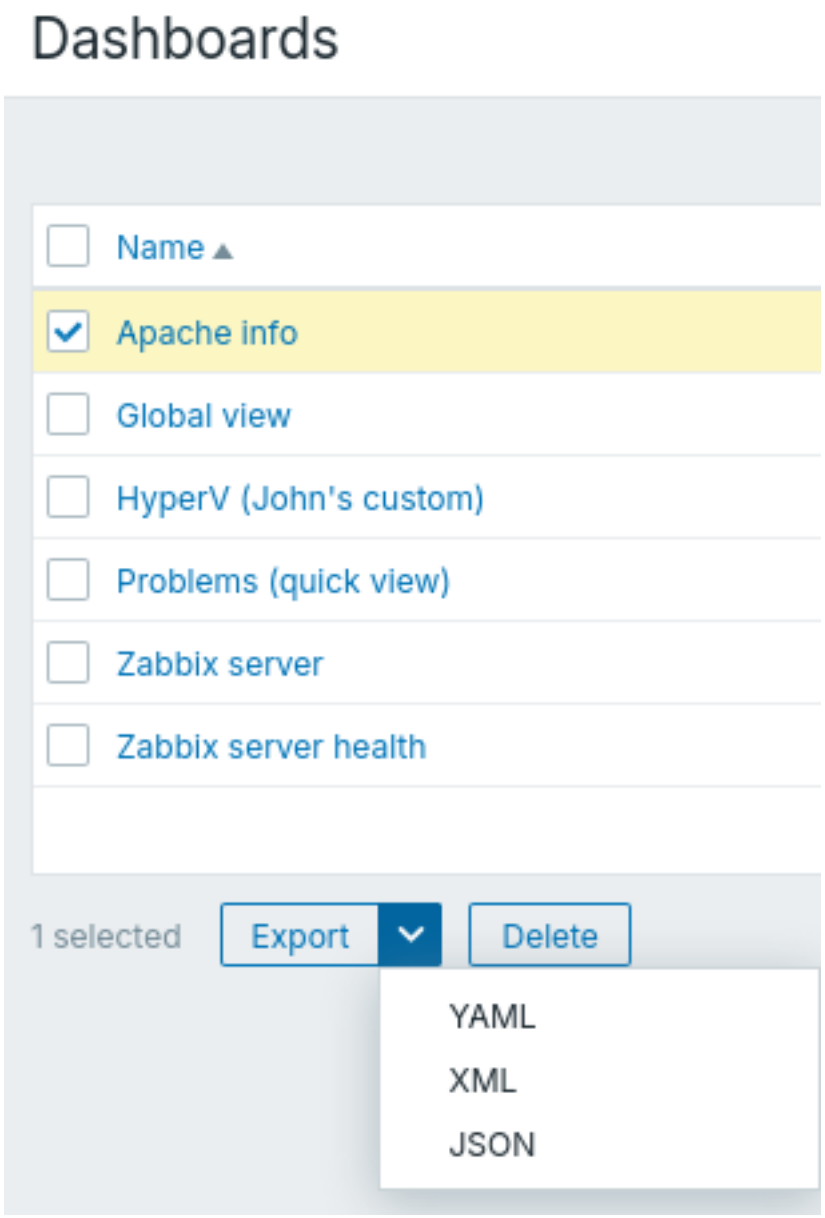
Übersicht

Dashboards werden zusammen mit allen zugehörigen Objekten und Objektbeziehungen exportiert.

Exportieren

Gehen Sie zum Exportieren von Dashboards wie folgt vor:

1. Gehen Sie zu *Dashboards* > *Alle Dashboards*.
2. Markieren Sie die Kontrollkästchen der zu exportierenden Dashboards.
3. Klicken Sie unterhalb der Liste auf *Export*, um im standardmäßigen YAML-Format zu exportieren, oder klicken Sie auf den Pfeil, um ein anderes Exportformat auszuwählen.



Je nach ausgewähltem Format werden Dashboards in eine lokale Datei mit einem Standardnamen exportiert:

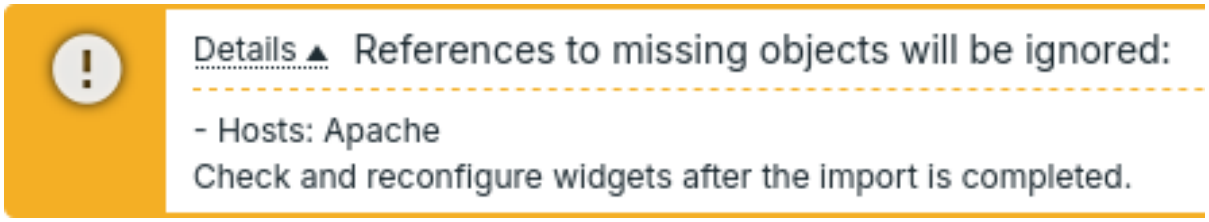
- `zbx_export_dashboards.yaml` - beim YAML-Export;
- `zbx_export_dashboards.xml` - beim XML-Export;
- `zbx_export_dashboards.json` - beim JSON-Export.

Importieren

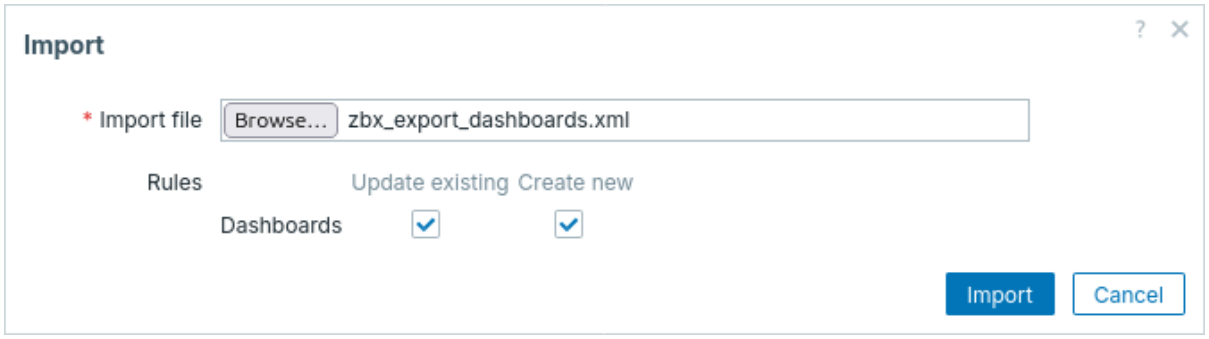
Gehen Sie zum Importieren von Dashboards wie folgt vor:

1. Gehen Sie zu *Dashboards* > *Alle Dashboards*.
2. Klicken Sie oben rechts auf *Importieren*. Die Option ist deaktiviert, wenn der Benutzer nicht über die Berechtigung zum *Erstellen und Bearbeiten von Dashboards* verfügt.
3. Wählen Sie die Importdatei aus.
4. Aktivieren Sie die Kontrollkästchen der Importregeln, die Sie anwenden möchten (siehe unten).
5. Klicken Sie unten rechts im Konfigurationsformular auf *Importieren*.

Wenn die Importdatei Objekte referenziert, die auf dem Zielsystem nicht vorhanden sind, wird eine Warnmeldung angezeigt. Dies ist bei Widgets zu erwarten, die sich auf bestimmte Hosts, Datenpunkte oder andere Objekte beziehen. In solchen Fällen ermöglicht Zabbix den Import des Dashboards, entfernt die fehlenden Referenzen und markiert das Widget entweder als nicht vollständig konfiguriert, wenn die Daten erforderlich sind, oder entfernt die Filterkriterien, wenn die Felder nur zum Filtern verwendet werden.



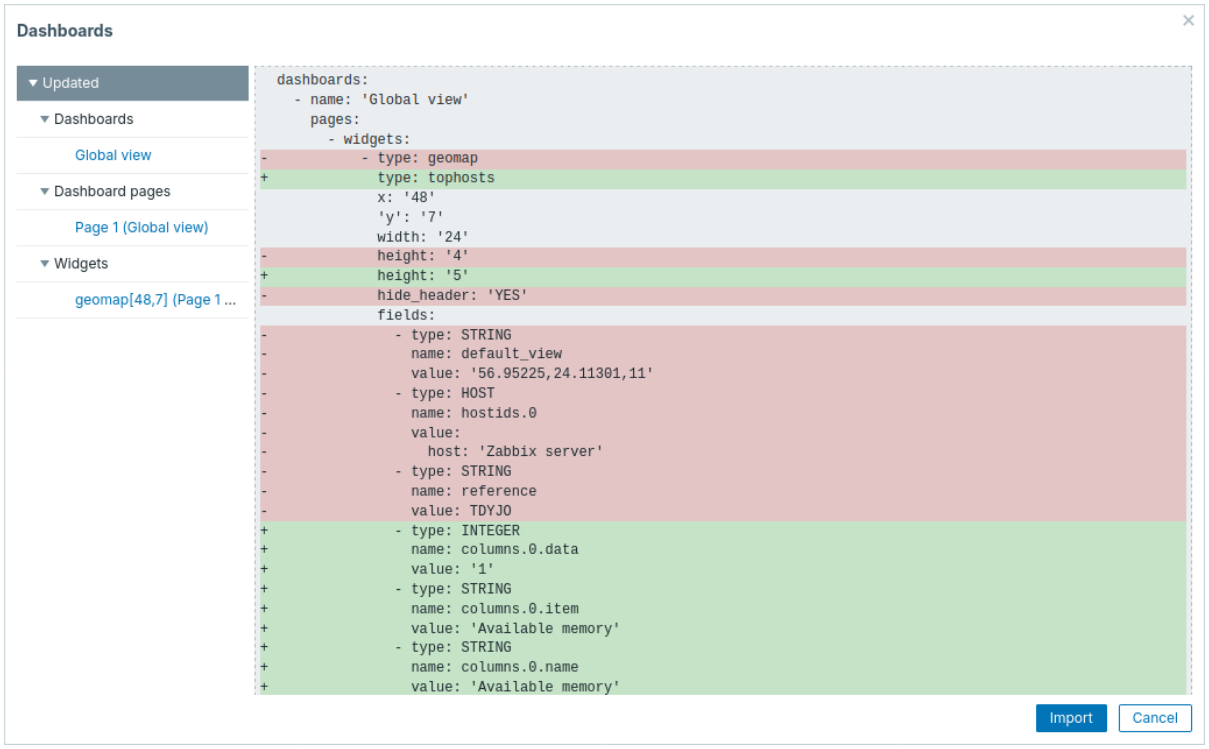
Durch erneutes Klicken auf die Schaltfläche *Importieren* wird das Dashboard hinzugefügt und/oder aktualisiert.



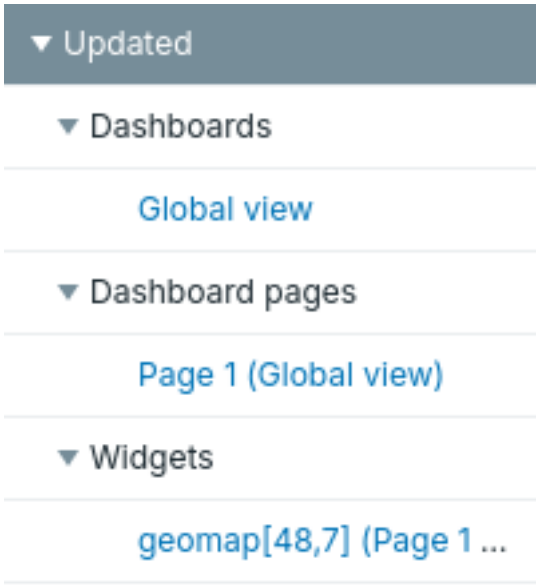
Importregeln:

Regel	Beschreibung
Vorhandene aktualisieren	Vorhandene Elemente werden anhand der Daten aus der Importdatei aktualisiert. Andernfalls werden sie nicht aktualisiert.
Neue erstellen	Neue Elemente werden anhand der Daten aus der Importdatei erstellt. Andernfalls werden sie nicht erstellt.

Auf dem nächsten Bildschirm können Sie den Inhalt der zu importierenden Dashboards anzeigen. Wenn es sich um ein neues Dashboard handelt, werden alle Elemente grün aufgelistet. Beim Aktualisieren eines vorhandenen Dashboards werden neue Dashboard-Elemente grün hervorgehoben, entfernte Dashboard-Elemente rot hervorgehoben und unveränderte Elemente auf einem grauen Hintergrund aufgelistet.



Über das Menü auf der linken Seite können Sie durch die Liste der Änderungen navigieren. Der Abschnitt *Aktualisiert* hebt alle an vorhandenen Dashboard-Elementen vorgenommenen Änderungen hervor. Der Abschnitt *Hinzugefügt* listet neue Dashboard-Elemente auf. Die Elemente in den einzelnen Abschnitten sind nach Elementtyp gruppiert. Klicken Sie auf den grauen Pfeil, um die Elementgruppe ein- oder auszublenden.



Überprüfen Sie die Dashboard-Änderungen und klicken Sie anschließend auf *Importieren*, um den Dashboard-Import durchzuführen. Im Frontend wird eine Erfolgs- oder Fehlermeldung zum Import angezeigt.

Exportformat

Export nach YAML:

```yaml
zabbix_export:
  version: '8.0'
  dashboards:
    - name: 'Global view'
      pages:
        - widgets:
            - type: tophosts
              name: 'Top hosts by CPU utilization'
              width: '25'
              height: '4'
              fields:
                - type: INTEGER
                  name: column
                  value: '1'
                - type: INTEGER
                  name: columns.0.data
                  value: '2'
                - type: STRING
                  name: columns.0.item
                  value: 'System name'
                - type: STRING
                  name: columns.0.name
                  value: 'Host name'
                - type: STRING
                  name: columns.1.base_color
                  value: 4CAF50
                - type: INTEGER
                  name: columns.1.data
                  value: '1'
                - type: INTEGER
                  name: columns.1.display
                  value: '3'
                - type: STRING
                  name: columns.1.item
                  value: 'CPU utilization'
```

```
      - type: STRING
        name: columns.1.max
        value: '100'
      - type: STRING
        name: columns.1.min
        value: '0'
      - type: STRING
        name: columns.1.name
        value: Utilization
      - type: INTEGER
        name: columns.2.data
        value: '1'
      - type: STRING
        name: columns.2.item
        value: 'Load average (1m avg)'
      - type: STRING
        name: columns.2.name
        value: '1m avg'
      - type: INTEGER
        name: columns.3.data
        value: '1'
      - type: STRING
        name: columns.3.item
        value: 'Load average (5m avg)'
      - type: STRING
        name: columns.3.name
        value: '5m avg'
      - type: INTEGER
        name: columns.4.data
        value: '1'
      - type: STRING
        name: columns.4.item
        value: 'Load average (15m avg)'
      - type: STRING
        name: columns.4.name
        value: '15m avg'
      - type: INTEGER
        name: columns.5.data
        value: '1'
      - type: INTEGER
        name: columns.5.decimal_places
        value: '0'
      - type: STRING
        name: columns.5.item
        value: 'Number of processes'
      - type: STRING
        name: columns.5.name
        value: Processes
      - type: STRING
        name: columnsthresholds.1.color.0
        value: FFFF00
      - type: STRING
        name: columnsthresholds.1.color.1
        value: FF8000
      - type: STRING
        name: columnsthresholds.1.color.2
        value: FF4000
      - type: STRING
        name: columnsthresholds.1.threshold.0
        value: '50'
      - type: STRING
        name: columnsthresholds.1.threshold.1
```

```
            value: '80'
          - type: STRING
            name: columnsthresholds.1.threshold.2
            value: '90'
          - type: STRING
            name: reference
            value: NLVMO
        - type: hostavail
          'y': '4'
          width: '25'
          fields:
            - type: INTEGER
              name: interface_type.0
              value: '1'
        - type: problems
          name: 'Current problems'
          'y': '6'
          width: '48'
          height: '5'
          fields:
            - type: STRING
              name: reference
              value: BRUQE
            - type: INTEGER
              name: show
              value: '3'
            - type: INTEGER
              name: show_tags
              value: '3'
        - type: systeminfo
          x: '25'
          width: '23'
          height: '4'
        - type: problemsbysv
          name: 'Problems by severity'
          x: '25'
          'y': '4'
          width: '23'
          fields:
            - type: STRING
              name: reference
              value: LBSVB
            - type: INTEGER
              name: show_type
              value: '1'
        - type: gauge
          name: 'Zabbix server'
          x: '48'
          width: '12'
          height: '4'
          hide_header: 'YES'
          fields:
            - type: INTEGER
              name: angle
              value: '270'
            - type: INTEGER
              name: desc_size
              value: '8'
            - type: ITEM
              name: itemid.0
              value:
                host: 'Zabbix server'
```

```
      key: vm.memory.utilization
  - type: STRING
    name: max
    value: '100'
  - type: STRING
    name: min
    value: '0'
  - type: INTEGER
    name: scale_size
    value: '11'
  - type: INTEGER
    name: show.0
    value: '1'
  - type: INTEGER
    name: show.1
    value: '2'
  - type: INTEGER
    name: show.2
    value: '3'
  - type: INTEGER
    name: show.3
    value: '4'
  - type: INTEGER
    name: show.4
    value: '5'
  - type: STRING
    name: thresholds.0.color
    value: 42A5F5
  - type: STRING
    name: thresholds.0.threshold
    value: '0'
  - type: STRING
    name: thresholds.1.color
    value: 9CCC65
  - type: STRING
    name: thresholds.1.threshold
    value: '20'
  - type: STRING
    name: thresholds.2.color
    value: FFEE58
  - type: STRING
    name: thresholds.2.threshold
    value: '40'
  - type: STRING
    name: thresholds.3.color
    value: FFA726
  - type: STRING
    name: thresholds.3.threshold
    value: '60'
  - type: STRING
    name: thresholds.4.color
    value: FF7043
  - type: STRING
    name: thresholds.4.threshold
    value: '80'
  - type: INTEGER
    name: th_arc_size
    value: '9'
  - type: INTEGER
    name: th_show_arc
    value: '1'
  - type: INTEGER
```

```
            name: th_show_labels
            value: '1'
          - type: INTEGER
            name: units_bold
            value: '1'
          - type: INTEGER
            name: units_size
            value: '24'
          - type: INTEGER
            name: value_arc_size
            value: '30'
          - type: INTEGER
            name: value_bold
            value: '1'
          - type: INTEGER
            name: value_size
            value: '24'
      - type: svggraph
        x: '48'
        'y': '4'
        width: '24'
        height: '3'
        hide_header: 'YES'
        fields:
          - type: INTEGER
            name: axisx
            value: '0'
          - type: STRING
            name: ds.0.color
            value: AB47BC
          - type: INTEGER
            name: ds.0.fill
            value: '0'
          - type: STRING
            name: ds.0.hosts.0
            value: 'Zabbix server'
          - type: STRING
            name: ds.0.items.0
            value: 'Available memory'
          - type: INTEGER
            name: ds.0.transparency
            value: '2'
          - type: INTEGER
            name: ds.0.type
            value: '2'
          - type: INTEGER
            name: ds.0.width
            value: '2'
          - type: INTEGER
            name: ds.1.axisy
            value: '1'
          - type: STRING
            name: ds.1.color
            value: 42A5F5
          - type: INTEGER
            name: ds.1.fill
            value: '0'
          - type: STRING
            name: ds.1.hosts.0
            value: 'Zabbix server'
          - type: STRING
            name: ds.1.items.0
```

```
            value: 'CPU utilization'
          - type: INTEGER
            name: ds.1.transparency
            value: '0'
          - type: INTEGER
            name: ds.1.type
            value: '2'
          - type: INTEGER
            name: ds.1.width
            value: '2'
          - type: STRING
            name: reference
            value: TOYGS
      - type: geomap
        x: '48'
        'y': '7'
        width: '24'
        height: '4'
        hide_header: 'YES'
        fields:
          - type: STRING
            name: default_view
            value: '56.95225,24.11301,11'
          - type: HOST
            name: hostids.0
            value:
              host: 'Zabbix server'
          - type: STRING
            name: reference
            value: TDYJO
      - type: clock
        name: 'Local time'
        x: '60'
        width: '12'
        hide_header: 'YES'
        fields:
          - type: INTEGER
            name: clock_type
            value: '1'
          - type: INTEGER
            name: show.0
            value: '1'
          - type: INTEGER
            name: show.1
            value: '2'
          - type: INTEGER
            name: show.2
            value: '3'
          - type: INTEGER
            name: time_sec
            value: '0'
          - type: INTEGER
            name: tzone_format
            value: '1'
      - type: item
        name: 'Server performance'
        x: '60'
        'y': '2'
        width: '12'
        hide_header: 'YES'
        fields:
          - type: INTEGER
```

```
      name: aggregate_function
      value: '2'
    - type: INTEGER
      name: decimal_size
      value: '32'
    - type: STRING
      name: description
      value: 'Values per second'
    - type: INTEGER
      name: desc_h_pos
      value: '0'
    - type: INTEGER
      name: desc_v_pos
      value: '0'
    - type: ITEM
      name: itemid.0
      value:
        host: 'Zabbix server'
        key: 'zabbix[wcache,values]'
    - type: INTEGER
      name: show.0
      value: '1'
    - type: INTEGER
      name: show.1
      value: '2'
    - type: INTEGER
      name: show.2
      value: '5'
    - type: INTEGER
      name: sparkline.fill
      value: '2'
    - type: INTEGER
      name: sparkline.width
      value: '2'
    - type: STRING
      name: thresholds.0.color
      value: DCEDC8
    - type: STRING
      name: thresholds.0.threshold
      value: '20'
    - type: STRING
      name: thresholds.1.color
      value: FFF9C4
    - type: STRING
      name: thresholds.1.threshold
      value: '40'
    - type: STRING
      name: thresholds.2.color
      value: FFCCBC
    - type: STRING
      name: thresholds.2.threshold
      value: '60'
    - type: INTEGER
      name: units_bold
      value: '0'
    - type: INTEGER
      name: units_show
      value: '0'
    - type: INTEGER
      name: value_h_pos
      value: '0'
    - type: INTEGER
```

```
                name: value_size
                value: '32'
```

Exportierte Elemente

Exportierte Elemente werden in der folgenden Tabelle erläutert.

Element	Type	Description
name	string	(required) Dashboard-Name.
pages		Stammelement für Dashboard-Seiten
widgets		Stammelement für Dashboard-Seiten

Note:
Siehe auch: Dashboard object (verweisen Sie auf die entsprechende Eigenschaft mit demselben Namen).

Dashboard-Widget

Element	Type	Description
type	string	(erforderlich) Typ des Dashboard-Widgets. Mögliche Werte: actionlog - Aktionsprotokoll; clock - Uhr; discovery - Erkennungsstatus; favgraphs - Favoriten-Diagramme; favmaps - Favoriten-Karten; gauge - Messanzeige; geomap - Geomap; graph - Diagramm (klassisch); graphprototype - Diagrammvorlage; honeycomb - Honeycomb; hostavail - Host-Verfügbarkeit; hostcard - Host-Karte; hostnavigator - Host-Navigator; itemcard - Datenpunkt-Karte; itemhistory - Datenpunktverlauf; itemnavigator - Datenpunkt-Navigator; item - Datenpunktwert; map - Karte; navtree - Karten-Navigationstree; piechart - Kreisdiagramm; problemhosts - Hosts mit Problemen; problems - Probleme; problemsbysv - Probleme nach Schweregrad; scatterplot - Streudiagramm; slareport - SLA-Bericht; svggraph - Diagramm; systeminfo - Systeminformationen; tophosts - Top-Hosts; topitems - Top-Datenpunkte; toptriggers - Top-Auslöser; trigover - Auslöserübersicht; url - URL; web - Webüberwachung.
x	integer	Eine horizontale Position von der linken Seite des Dashboards aus. Mögliche Werte:[1] 0-71.
y	integer	Eine vertikale Position von der Oberseite des Dashboards aus. Mögliche Werte:[1] 0-63.
name	string	Benutzerdefinierter Name des Widgets.
width	integer	Die Breite des Widgets. Mögliche Werte:[1] 1-72.

Element	Type	Description
height	integer	Die Höhe des Widgets. Mögliche Werte:[1] 1-64.
hide_header	string	Widget-Kopfzeile ausblenden. Mögliche Werte:[1] NO (0, Standard), YES (1).
fields		Stammelement für die Felder des Dashboard-Widgets.

Note:
Siehe auch: Dashboard widget (verweist auf die entsprechende Eigenschaft mit passendem Namen).

Dashboard-Widget-Feld

Element	Type	Description
type	string	(erforderlich) Typ des Widget-Felds. Mögliche Werte:[1] INTEGER (0), STRING (1), HOST_GROUP (2), HOST (3), ITEM (4), ITEM_PROTOTYPE (5), GRAPH (6), GRAPH_PROTOTYPE (7), MAP (8), SERVICE (9), SLA (10), USER (11), ACTION (12), MEDIA_TYPE (13).
name	string	(erforderlich) Name des Widget-Felds.
value	mixed	(erforderlich) Wert des Widget-Felds, abhängig vom Feldtyp.
host	string	Host des Datenpunkts.
key	string	Schlüssel des Datenpunkts.

Note:
Siehe auch: Dashboard widget field (verweisen Sie auf die entsprechende Eigenschaft mit dem passenden Namen).

Fußnoten

[1] API-Integerwerte in Klammern, zum Beispiel ENABLED (0), werden nur als Referenz angegeben. Weitere Informationen finden Sie auf der verlinkten API-Objektseite im Tabelleneintrag oder am Ende jedes Abschnitts.

Reguläre Ausdrücke

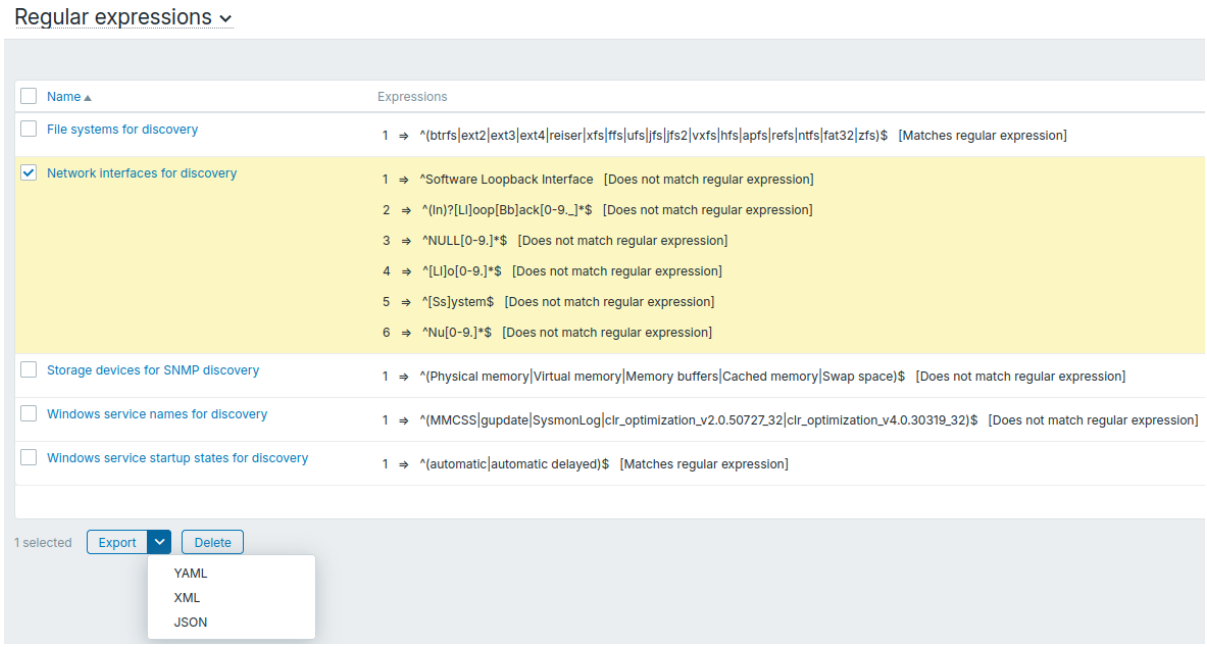
Übersicht

Reguläre Ausdrücke werden zusammen mit allen zugehörigen Objekten und Objektbeziehungen exportiert.

Exportieren

Um reguläre Ausdrücke zu exportieren, gehen Sie wie folgt vor:

1. Gehen Sie zu *Administration* > *Allgemein* > *Reguläre Ausdrücke*.
2. Markieren Sie die Kontrollkästchen der zu exportierenden regulären Ausdrücke.
3. Klicken Sie unterhalb der Liste auf *Export*, um im standardmäßigen YAML-Format zu exportieren, oder klicken Sie auf den Pfeil, um ein anderes Exportformat auszuwählen.



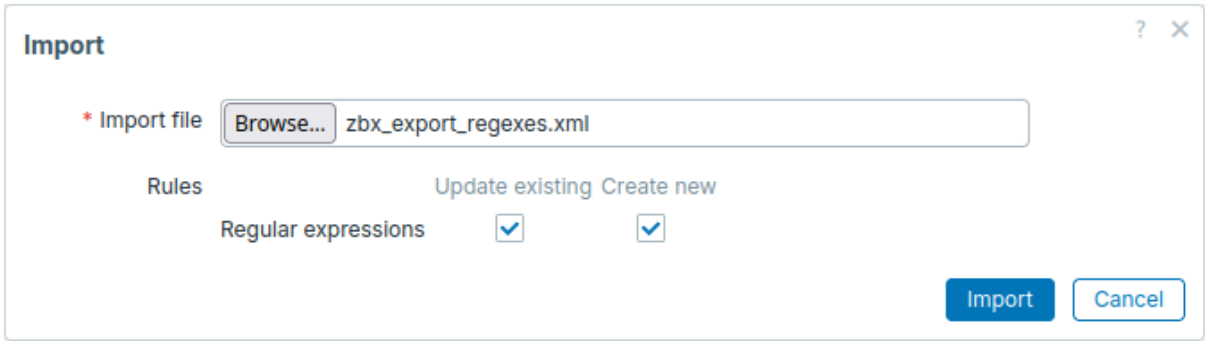
Je nach ausgewähltem Format werden reguläre Ausdrücke in eine lokale Datei mit einem Standardnamen exportiert:

- `zbx_export_regexes.yaml` - beim YAML-Export;
- `zbx_export_regexes.xml` - beim XML-Export;
- `zbx_export_regexes.json` - beim JSON-Export.

Importieren

Gehen Sie wie folgt vor, um reguläre Ausdrücke zu importieren:

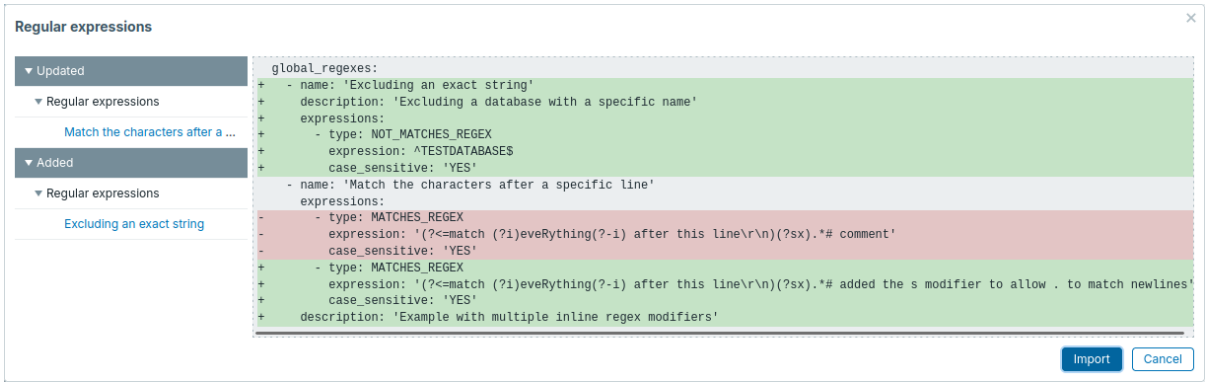
1. Gehen Sie zu *Administration* > *Allgemein* > *Reguläre Ausdrücke*.
2. Klicken Sie oben rechts auf *Importieren*.
3. Wählen Sie die Importdatei aus.
4. Aktivieren Sie die Kontrollkästchen der Importregeln, die Sie anwenden möchten (siehe unten).
5. Klicken Sie unten rechts im Konfigurationsformular auf *Importieren*.



Importregeln:

Regel	Beschreibung
Vorhandene aktualisieren	Vorhandene reguläre Ausdrücke werden anhand der Daten aus der Importdatei aktualisiert. Dabei werden vorhandene Ausdrücke mit demselben Namen vollständig überschrieben. Andernfalls werden sie nicht aktualisiert.
Neue erstellen	Neue reguläre Ausdrücke werden anhand der Daten aus der Importdatei erstellt. Andernfalls werden sie nicht erstellt.

Auf dem nächsten Bildschirm können Sie den Inhalt der importierten regulären Ausdrücke anzeigen. Handelt es sich um einen neuen regulären Ausdruck, werden alle Daten grün aufgelistet. Beim Aktualisieren eines vorhandenen regulären Ausdrucks werden neue Daten des regulären Ausdrucks grün hervorgehoben, entfernte Daten rot und unveränderte Daten auf grauem Hintergrund angezeigt.



Über das Menü auf der linken Seite können Sie durch die Liste der Änderungen navigieren. Der Abschnitt *Aktualisiert* hebt alle an vorhandenen regulären Ausdrücken vorgenommenen Änderungen hervor. Der Abschnitt *Hinzugefügt* listet neue reguläre Ausdrücke auf. Klicken Sie auf den Pfeil, um eine Gruppe regulärer Ausdrücke auf- oder zuzuklappen.

Überprüfen Sie die Änderungen an den regulären Ausdrücken und klicken Sie anschließend auf *Importieren*, um den Import der regulären Ausdrücke durchzuführen. Im Frontend wird eine Erfolgs- oder Fehlermeldung zum Import angezeigt.

Exportformat

Export nach YAML:

```yaml
zabbix_export:
  version: '8.0'
  global_regexes:
    - name: 'Dateisysteme für die Erkennung'
      description: 'Globale reguläre Ausdrucksregel, die verwendet wird, um Dateisysteme auf dem überwacht
      expressions:
        - type: MATCHES_REGEX
          expression: ^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs
    - name: 'Speichergeräte für die SNMP-Erkennung'
      expressions:
        - type: NOT_MATCHES_REGEX
          expression: '^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$'
          case_sensitive: 'YES'
```

Exportierte Elemente

Exportierte Elemente von regulären Ausdrücken:

Element	Type	Description
global_regexes	object	Root-Element für reguläre Ausdrücke.
name	string	(erforderlich) Name des regulären Ausdrucks.
description	text	Beschreibung des regulären Ausdrucks.
expressions	object	Root-Element für reguläre Ausdrücke.
type	string	Typ des regulären Ausdrucks.
expression	string	Regulärer Ausdruck.
exp_delimiter	string	Trennzeichen des regulären Ausdrucks; vorhanden, wenn der Typ *Contains any substring from list* ist.
case_sensitive	string	Option für Groß-/Kleinschreibung des regulären Ausdrucks.

Note:
Siehe auch: Regular expression object (verweisen Sie auf die entsprechende Eigenschaft mit dem passenden Namen).

Reguläre Ausdrücke

Übersicht Perl Compatible Regular Expressions (PCRE, PCRE2) werden in Zabbix unterstützt.

Es gibt zwei Möglichkeiten, reguläre Ausdrücke in Zabbix zu verwenden:

- Manuelle Eingabe eines regulären Ausdrucks
- Verwendung eines in Zabbix erstellten globalen regulären Ausdrucks

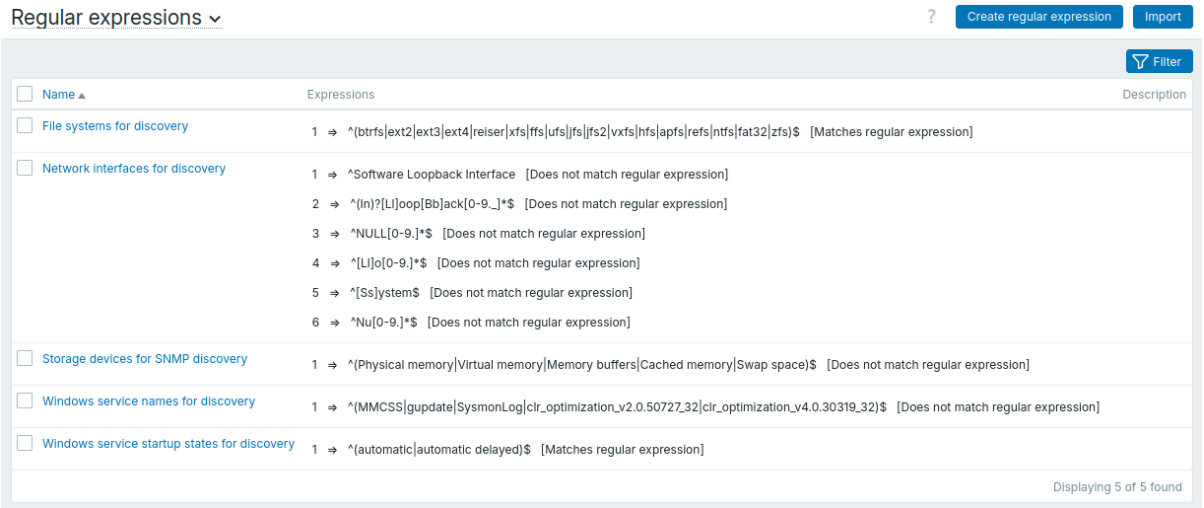
Reguläre Ausdrücke Sie können einen regulären Ausdruck manuell in unterstützten Bereichen eingeben.
 Beachten Sie, dass der Ausdruck nicht mit @ beginnen darf, da dieses Symbol in Zabbix verwendet wird, um auf globale reguläre Ausdrücke zu verweisen.

Attention:
Bei der Verwendung regulärer Ausdrücke kann der Stack-Speicher erschöpft werden. Weitere Informationen finden Sie unter PCRE PERFORMANCE.

Beachten Sie, dass bei der mehrzeiligen Übereinstimmung die Anker ^ und $ jeweils am Anfang und Ende jeder Zeile und nicht am Anfang und Ende des gesamten Strings übereinstimmen.

Siehe auch Beispiele für das korrekte Escaping in verschiedenen Kontexten.

Globale reguläre Ausdrücke Zabbix stellt im Frontend einen erweiterten Editor zum Erstellen und Testen komplexer regulärer Ausdrücke bereit.
 Um darauf zuzugreifen, gehen Sie zu *Administration* > *Allgemein* > *Reguläre Ausdrücke*.
 Zabbix enthält in seinem Standarddatensatz eine Reihe von vordefinierten globalen regulären Ausdrücken.

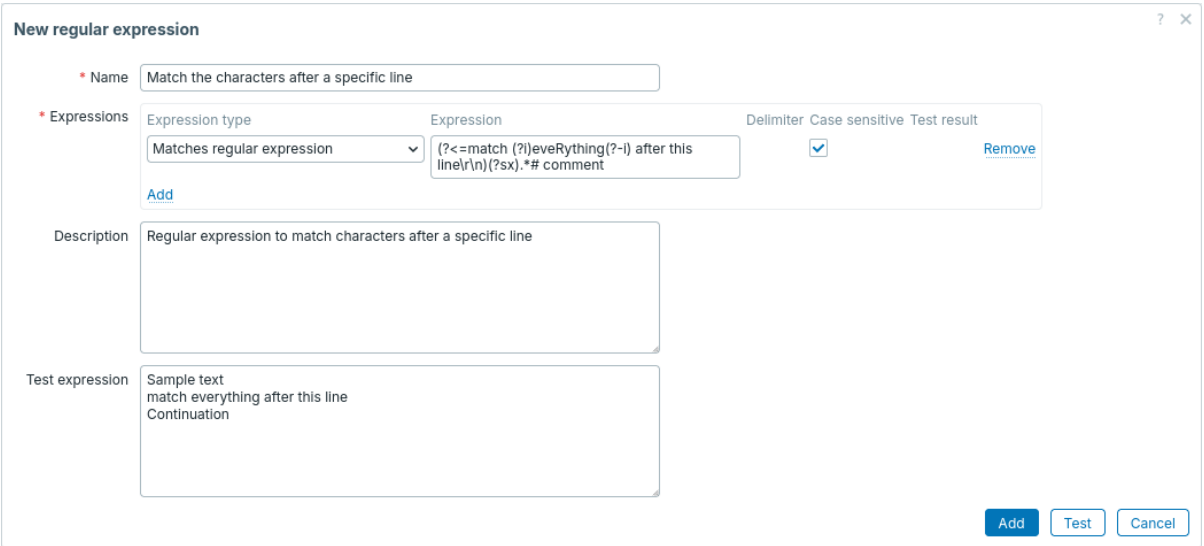


Sie können die Liste der regulären Ausdrücke nach Name und Beschreibung filtern, um die benötigten Ausdrücke zu finden.

Konfiguration

So erstellen Sie einen globalen regulären Ausdruck:

1. Klicken Sie auf *Regulären Ausdruck erstellen*.
2. Geben Sie die Parameter des regulären Ausdrucks im Formular ein.



Alle Pflichtfelder sind mit einem roten Sternchen gekennzeichnet.

Parameter	Beschreibung
Name	Legen Sie den Namen des regulären Ausdrucks fest. Alle Unicode-Zeichen sind zulässig.
Ausdrücke	Klicken Sie im Abschnitt *Ausdrücke* auf *Hinzufügen*, um einen neuen Unterausdruck hinzuzufügen.
Ausdruckstyp	Wählen Sie den Ausdruckstyp aus: **Enthält Zeichenfolge** – stimmt mit der Teilzeichenfolge überein. **Enthält eine beliebige Teilzeichenfolge aus der Liste** – stimmt mit einer beliebigen Teilzeichenfolge aus einer durch Trennzeichen getrennten Liste überein. Die Liste kann durch ein Komma `,`, einen Punkt `.` oder einen Schrägstrich `/` getrennt sein. **Enthält keine Zeichenfolge** – stimmt mit jeder Zeichenfolge außer der Teilzeichenfolge überein. **Entspricht regulärem Ausdruck** – stimmt mit dem regulären Ausdruck überein. **Entspricht nicht dem regulären Ausdruck** – stimmt nicht mit dem regulären Ausdruck überein.
Ausdruck	Geben Sie eine Teilzeichenfolge bzw. einen regulären Ausdruck ein. Maximale Länge: 2048 Zeichen.
Trennzeichen	Ein Komma `,`, ein Punkt `.` oder ein Schrägstrich `/`, um Textzeichenfolgen in einem regulären Ausdruck zu trennen. Dieser Parameter ist nur aktiv, wenn der Ausdruckstyp *Enthält eine beliebige Teilzeichenfolge aus der Liste* ausgewählt ist.
Groß-/Kleinschreibung beachten	Ein Kontrollkästchen, mit dem festgelegt wird, ob beim regulären Ausdruck die Groß-/Kleinschreibung beachtet wird.
Beschreibung	Geben Sie eine Beschreibung für den regulären Ausdruck ein.
Ausdruck testen	Ein Wert, der zum Testen des regulären Ausdrucks verwendet wird.

Ein Schrägstrich `/` im Ausdruck wird wörtlich und nicht als Trennzeichen behandelt. Dadurch können Ausdrücke mit einem Schrägstrich fehlerfrei gespeichert werden.

Attention:
Ein benutzerdefinierter Name für einen regulären Ausdruck in Zabbix darf Kommas, Leerzeichen usw. enthalten. Wenn dies bei der Referenzierung zu Fehlinterpretationen führen kann (beispielsweise ein Komma im Parameter eines Datenpunktschlüssels), kann die gesamte Referenz in Anführungszeichen eingeschlossen werden, zum Beispiel: *"@My custom regexp for purpose1, purpose2"*.
 Namen regulärer Ausdrücke dürfen an anderen Stellen nicht in Anführungszeichen gesetzt werden (beispielsweise in den Eigenschaften einer LLD-Regel).

Nachdem ein regulärer Ausdruck erstellt wurde, kann er an mehreren Stellen im Frontend verwendet werden, indem auf seinen Namen mit dem Präfix `@` verwiesen wird, zum Beispiel `@mycustomregexp`.

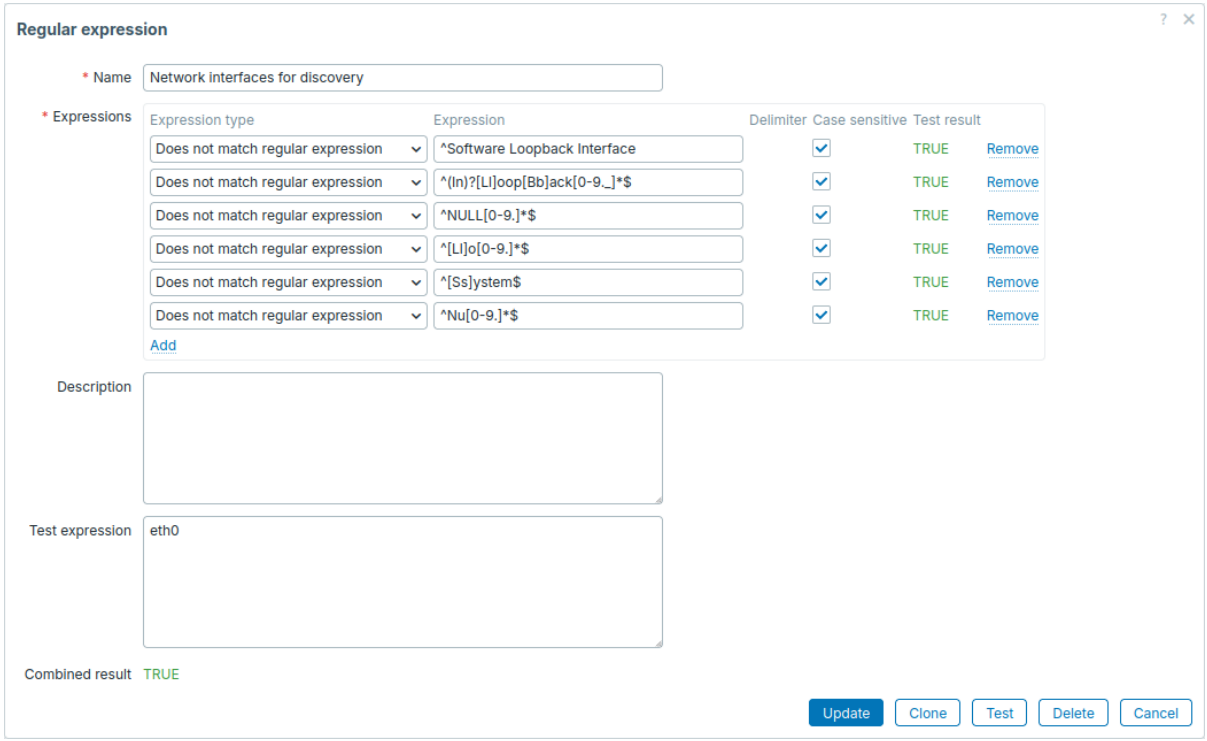
Formularschaltflächen

Die Schaltflächen am unteren Rand des Formulars ermöglichen verschiedene Aktionen:

Schaltfläche	Beschreibung
Add	Einen regulären Ausdruck hinzufügen. Diese Schaltfläche ist nur für neue reguläre Ausdrücke verfügbar.
Update	Die Eigenschaften eines regulären Ausdrucks aktualisieren.
Clone	Einen weiteren regulären Ausdruck auf Grundlage der Eigenschaften des aktuellen regulären Ausdrucks erstellen.
Test	Einen regulären Ausdruck und seine Teilausdrücke anhand eines angegebenen Werts testen.
Delete	Den regulären Ausdruck löschen.
Cancel	Die Bearbeitung der Eigenschaften des regulären Ausdrucks abbrechen.

Testen eines regulären Ausdrucks

Sie können den regulären Ausdruck und seine Teil-Ausdrücke testen, indem Sie einen Wert in das Feld *Test expression* eingeben und auf die Schaltfläche *Test* button klicken.



In der Spalte *Test result* wird der Status jedes Teil-Ausdrucks angezeigt. Das Feld *Combined result* zeigt den Gesamtstatus des regulären Ausdrucks an. Wenn mehrere Teil-Ausdrücke definiert sind, verwendet Zabbix den logischen UND-Operator, um das *Combined result* zu berechnen; das heißt, wenn mindestens ein Ergebnis `False` ist, ist auch das *Combined result* `False`.

Standardmäßige globale reguläre Ausdrücke

Der Standarddatensatz enthält die folgenden vordefinierten globalen regulären Ausdrücke:

Name	Expression	Matches
Dateisysteme für die Erkennung	`^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs)$`	”btrfs” oder ”ext2” oder ”ext3” oder ”ext4” oder ”reiser” oder ”xfs” oder ”ffs” oder ”ufs” oder ”jfs” oder ”jfs2” oder ”vxfs” oder ”hfs” oder ”apfs” oder ”refs” oder ”ntfs” oder ”fat32” oder ”zfs”.
Netzwerkschnittstellen für die Erkennung	`^Software Loopback Interface`	Zeichenfolgen, die mit ”Software Loopback Interface” beginnen.
	`^lo$`	”lo”.
	`^(In)?[Ll]oop[Bb]ack[0-9._]*$`	Zeichenfolgen, die optional mit ”In” beginnen, gefolgt von ”L” oder ”l”, dann ”oop”, dann ”B” oder ”b”, dann ”ack” und optional gefolgt von beliebig vielen Ziffern, Punkten oder Unterstrichen.
	`^NULL[0-9.]*$`	Zeichenfolgen, die mit ”NULL” beginnen und optional von beliebig vielen Ziffern oder Punkten gefolgt werden.
	`^[Ll]o[0-9.]*$`	Zeichenfolgen, die mit ”Lo” oder ”lo” beginnen und optional von beliebig vielen Ziffern oder Punkten gefolgt werden.
	`^[Ss]ystem$`	”System” oder ”system”.
	`^Nu[0-9.]*$`	Zeichenfolgen, die mit ”Nu” beginnen und optional von beliebig vielen Ziffern oder Punkten gefolgt werden.
Speichergeräte für die SNMP-Erkennung	`^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$`	”Physical memory” oder ”Virtual memory” oder ”Memory buffers” oder ”Cached memory” oder ”Swap space”.

Name	Expression	Matches
Windows-Dienstnamen für die Erkennung	`^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.30319_32)$`	"MMCSS", "gupdate", "SysmonLog", "clr_optimization_v2.0.50727_32" oder "clr_optimization_v4.0.30319_32"
Windows-Dienststartzustände für die Erkennung	`^(automatic|automatic delayed)$`	"automatic" oder "automatic delayed".

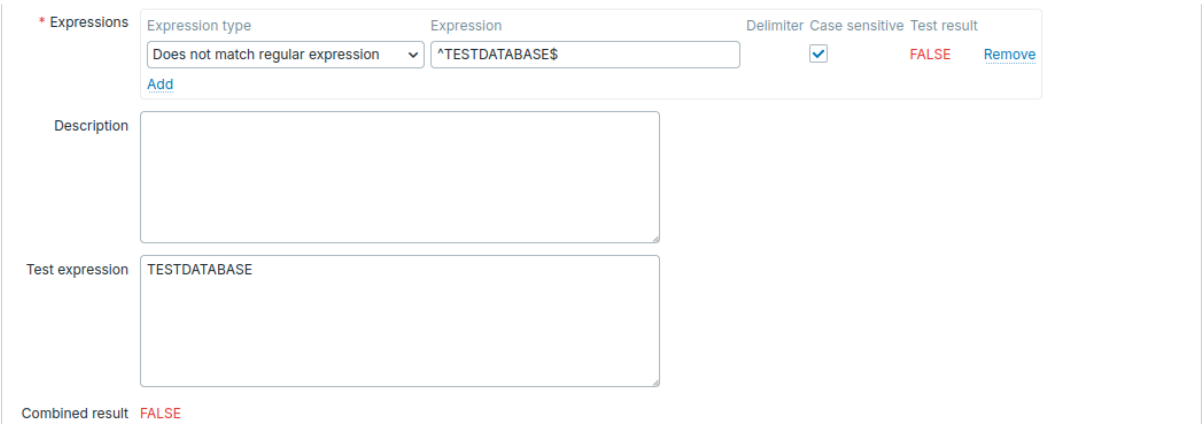
Importieren/Exportieren eines regulären Ausdrucks

Sie können globale reguläre Ausdrücke importieren oder exportieren, wie auf der Konfigurationsseite für den Export/Import von regulären Ausdrücken beschrieben.

Beispiele Beispiel für den Ausschluss eines exakten Strings

Verwenden Sie den folgenden Ausdruck in der Low-Level-Discovery, um Datenbanken zu entdecken, außer einer Datenbank mit einem bestimmten Namen:

```
^TESTDATABASE$
```

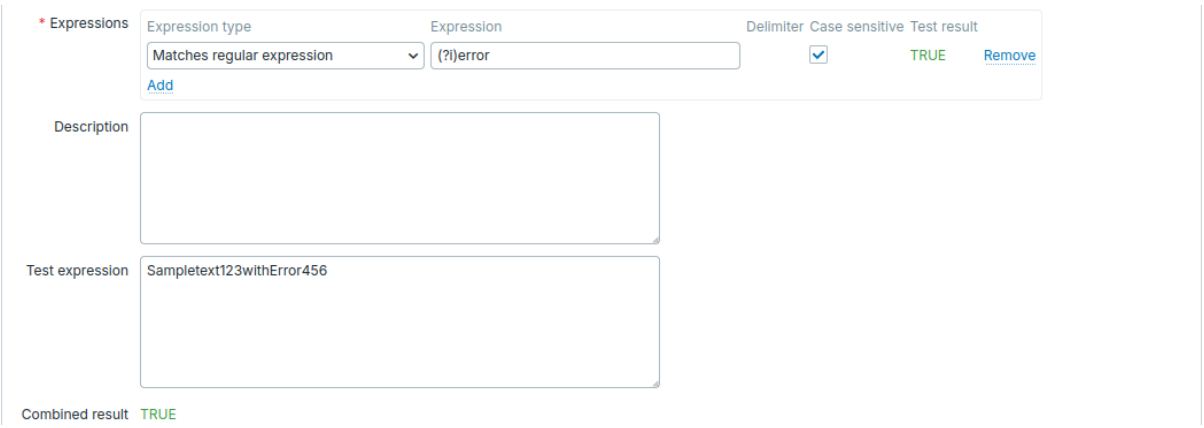


Sowohl *Test result* als auch *Combined result* zeigen `FALSE` an: Der Name stimmt mit "TESTDATABASE" überein, daher ist die Bedingung *Does not match regular expression* nicht erfüllt.

Beispiel mit einem Inline-Regex-Modifikator für einen nicht zwischen Groß- und Kleinschreibung unterscheidenden Abgleich

Verwenden Sie den folgenden regulären Ausdruck, der den Inline-Modifikator `(?i)` enthält, um einen nicht zwischen Groß- und Kleinschreibung unterscheidenden Abgleich der Zeichenfolge "error" durchzuführen:

```
(?i)error
```



Sowohl *Test result* als auch *Combined result* zeigen `TRUE` an: Die Zeichenfolge "error" wird auch dann abgeglichen, wenn *Case sensitive* ausgewählt ist, da der Inline-Modifikator `(?i)` den Abgleich unabhängig von Groß- und Kleinschreibung macht.

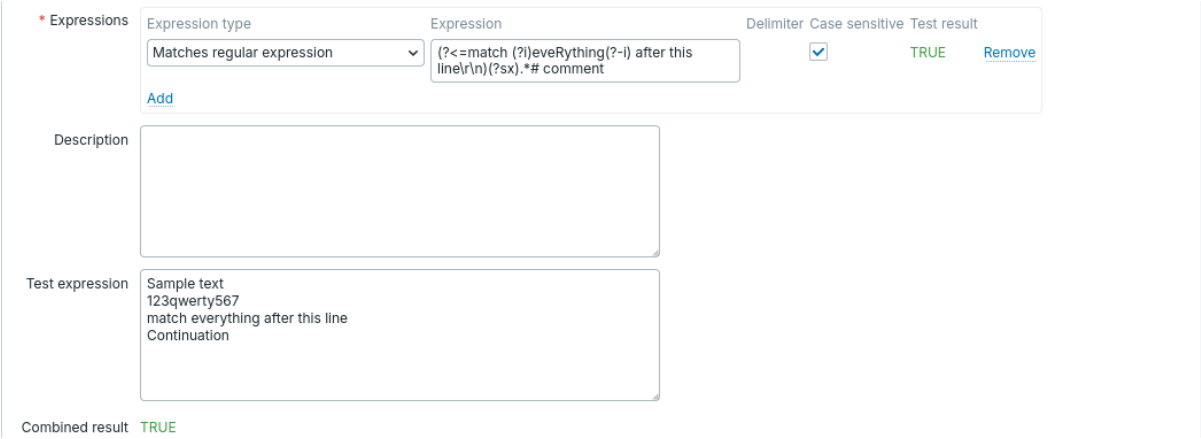
Beispiel mit mehreren Inline-Regex-Modifikatoren

Verwenden Sie den folgenden regulären Ausdruck, der mehrere Inline-Modifikatoren enthält, um die Zeichen nach einer bestimmten Zeile abzugleichen:

```
(?<=match (?i)eveRything(?-i) after this line\r\n)(?sx).*# comment
```

wobei:

- `(?i)` - aktiviert die Groß-/Kleinschreibung ignorierende Übereinstimmung;
- `(?-i)` - deaktiviert die Groß-/Kleinschreibung ignorierende Übereinstimmung;
- `(?s)` - aktiviert den Dotall-Modus, sodass `.` mit Zeilenumbruchzeichen übereinstimmen kann;
- `(?x)` - aktiviert den erweiterten Modus, sodass Leerzeichen und Kommentare im regulären Ausdruck ignoriert werden.



Sowohl *Test result* als auch *Combined result* zeigen `TRUE`: Die Zeichen nach der spezifischen Zeile werden abgeglichen.

Attention:
Der *g*-Modifikator kann nicht inline angegeben werden. Die Liste der verfügbaren Modifikatoren finden Sie auf der pcresyntax man page. Weitere Informationen zur PCRE-Syntax finden Sie in der PCRE HTML documentation.

Unterstützung regulärer Ausdrücke nach Ort

Ort		Regulärer Ausdruck	Globaler regulärer Ausdruck	Mehrzeilenabgleich	Kommentare
Agent items					
	eventlog[]	Ja	Ja	Ja	Parameter `regexp`, `severity`, `source`, `eventid`
	eventlog.count[]				Parameter `regexp`, `severity`, `source`, `eventid`
	log[]				Parameter `regexp`
	log.count[]				
	logrt[]		Ja/Nein		Parameter `regexp` unterstützt beide, der Parameter `file_regexp` unterstützt nur nicht-globale Ausdrücke
	logrt.count[]				
	proc.cpu.util[]		Nein	Nein	Parameter `cmdline`
	proc.get[]				
	proc.mem[]				
	proc.num[]				
	sensor[]				Parameter `device` und `sensor` unter Linux 2.4
	system.hw.macaddr[]				Parameter `interface`
	system.sw.packages[]				Parameter `regexp`
	system.sw.packages.get[]				Parameter `regexp`
	vfs.dir.count[]				Parameter `regex_incl`, `regex_excl`, `regex_excl_dir`

Ort		Regulärer Ausdruck	Globaler regulärer Ausdruck	Mehrzeilenabgleich	Kommentare
	vfs.dir.get[]				Parameter `regex_incl`, `regex_excl`, `regex_excl_dir`
	vfs.dir.size[]				Parameter `regex_incl`, `regex_excl`, `regex_excl_dir`
	vfs.file.regexp[] vfs.file.regmatch[] web.page.regexp[]			Ja	Parameter `regexp`
SNMP traps					
	snmptrap[]	Ja	Ja	Nein	Parameter `regexp`
Item value preprocessing		Ja	Nein	Nein	Parameter `pattern`
Functions for triggers/calculated items					
	count()	Ja	Ja	Ja	Parameter `pattern`, wenn der Parameter `operator` *regexp* oder *iregexp* ist
	countunique()	Ja	Ja		
	find()	Ja	Ja		
	logeventid() logsource()	Ja	Ja	Nein	Parameter `pattern`
Low-level discovery					
	Filter	Ja	Ja	Nein	Feld *Regulärer Ausdruck*
	Überschreibungen	Ja	Nein		In den Optionen *entspricht* und *entspricht nicht* für Bedingungen *Operation*
Action conditions		Ja	Nein	Nein	In den Optionen *entspricht* und *entspricht nicht* für die Autoregistrierungsbedingungen *Host name* und *Host metadata*
Scripts		Ja	Ja	Nein	Feld *Eingabevalidierungsregel*
Web monitoring		Ja	Nein	Ja	*Variablen* mit Präfix **regex:** Feld *Erforderliche Zeichenfolge*
User macro context		Ja	Nein	Nein	Im Makrokontext mit Präfix **regex:**
Macro functions					
	regsub() iregsub()	Ja	Nein	Nein	Parameter `pattern`
Link indicators in maps		Ja	Nein	Nein	Feld *Muster* (für Textelemente)
Icon mapping		Ja	Ja	Nein	Feld *Ausdruck*
Value mapping		Ja	Nein	Nein	Feld *Wert*, wenn der Zuordnungstyp `regexp` ist

Ort		Regulärer Ausdruck	Globaler regulärer Ausdruck	Mehrzeilenabgleich	Kommentare
Agent configuration parameters					
	AllowKeyRegexp DenyKeyRegexp	Ja	Nein	Nein	Muster

Ereignisse und Auflösung

Übersicht In Zabbix werden mehrere Arten von Ereignissen erzeugt:

- Auslöserereignisse - immer dann, wenn ein Auslöser seinen Status ändert (*OK→PROBLEM→OK*)
- Serviceereignisse - immer dann, wenn ein Service seinen Status ändert (*OK→PROBLEM→OK*)
- Erkennungsereignisse - wenn Hosts oder Services erkannt werden
- Autoregistrierungsereignisse - wenn aktive Agenten vom Server automatisch registriert werden
- interne Ereignisse - wenn ein Datenpunkt/eine Low-Level-Discovery-Regel nicht mehr unterstützt wird oder ein Auslöser in einen unbekannten Zustand wechselt

Ereignisse sind mit Zeitstempeln versehen und können die Grundlage für Aktionen wie das Senden von Benachrichtigungs-E-Mails, das Erstellen von Tickets oder das Ausführen von Remote-Befehlen sein.

Um Ereignisdetails im Frontend anzuzeigen, gehen Sie zu *Monitoring* > *Problems*. Dort können Sie auf Datum und Uhrzeit des Ereignisses klicken, um die Seite mit den Ereignisdetails zu öffnen. Dort sehen Sie zusätzliche Informationen zum Ereignis, einschließlich der Information, ob es aufgrund von Wartung unterdrückt oder von einem Benutzer manuell unterdrückt wurde.

Problembehebung Problemereignisse, die von Auslösern erstellt werden, werden automatisch behoben, wenn die Bedingung, die das Problem verursacht hat, nicht mehr besteht.

Um die Verwaltung von Problemereignissen flexibler und komfortabler zu gestalten, stehen die folgenden Optionen zur Verfügung:

- Ereigniskorrelation
- Ereignisbestätigung und Unterdrückung
- Manuelles Schließen von Problemen
- Wartungszeiträume - definieren Zeiträume, in denen Probleme unterdrückt werden
- Tagging - Probleme nach gemeinsamen Merkmalen taggen

Auslöser-Ereignisse Änderungen des Auslöserzustands sind die häufigste und wichtigste Quelle von Ereignissen. Jedes Mal, wenn der Auslöser seinen Zustand ändert, wird ein Ereignis erzeugt. Das Ereignis enthält Details zur Änderung des Auslöserzustands - wann sie aufgetreten ist und wie der neue Zustand lautet.

Von Auslösern werden zwei Arten von Ereignissen erstellt - `Problem` und `OK`.

Problemereignisse

Ein Problemereignis wird erstellt:

- wenn ein Auslöserausdruck zu TRUE ausgewertet wird, wenn sich der Auslöser im OK-Zustand befindet;
- jedes Mal, wenn ein Auslöserausdruck zu TRUE ausgewertet wird, wenn die Generierung mehrerer Problemereignisse für den Auslöser aktiviert ist.

OK-Ereignisse

Ein OK-Ereignis schließt die zugehörigen Problemereignisse und kann von 3 Komponenten erstellt werden:

- Auslöser - basierend auf den Einstellungen ”OK event generation” und ”OK event closes”;
- Ereigniskorrelation
- Task-Manager - wenn ein Ereignis manuell geschlossen wird

Auslöser verfügen über die Einstellung „OK-Ereignisgenerierung“, die steuert, wie OK-Ereignisse generiert werden:

- *Expression* - Ein OK-Ereignis wird für einen Auslöser im Problemzustand generiert, wenn sein Ausdruck zu FALSE ausgewertet wird. Dies ist die einfachste Einstellung und standardmäßig aktiviert.

- *Recovery expression* - Ein OK-Ereignis wird für einen Auslöser im Problemzustand generiert, wenn sein Ausdruck zu FALSE ausgewertet wird und der Wiederherstellungsausdruck zu TRUE ausgewertet wird. Dies kann verwendet werden, wenn die Wiederherstellungskriterien des Auslösers von den Problemkriterien abweichen.
- *None* - Ein OK-Ereignis wird niemals generiert. Dies kann in Verbindung mit der Generierung mehrerer Problemereignisse verwendet werden, um einfach eine Benachrichtigung zu senden, wenn etwas passiert.

Zusätzlich verfügen Auslöser über die Einstellung „OK-Ereignis schließt“, die steuert, welche Problemereignisse geschlossen werden:

- *All problems* - Ein OK-Ereignis schließt alle offenen Probleme, die vom Auslöser erstellt wurden
- *All problems if tag values match* - Ein OK-Ereignis schließt offene Probleme, die vom Auslöser erstellt wurden und mindestens einen übereinstimmenden Tag-Wert haben. Der Tag wird durch die Auslöser-Einstellung „Tag for matching“ definiert. Wenn keine Problemereignisse zum Schließen vorhanden sind, wird kein OK-Ereignis generiert. Dies wird oft als Ereigniskorrelation auf Auslöser-Ebene bezeichnet.

Ereigniskorrelation

Die Ereigniskorrelation (auch als globale Ereigniskorrelation bezeichnet) ermöglicht die Einrichtung benutzerdefinierter Regeln zum Schließen von Ereignissen (wodurch ein OK-Ereignis erzeugt wird).

Die Regeln legen fest, wie neue Problemereignisse mit vorhandenen Problemereignissen verknüpft werden. Dadurch können das neue Ereignis oder die übereinstimmenden Ereignisse durch die Erzeugung entsprechender OK-Ereignisse geschlossen werden.

Die Ereigniskorrelation muss jedoch sehr sorgfältig konfiguriert werden, da sie die Leistung der Ereignisverarbeitung negativ beeinflussen oder bei einer fehlerhaften Konfiguration mehr Ereignisse als beabsichtigt schließen kann (im schlimmsten Fall könnten sogar alle Problemereignisse geschlossen werden). Einige Konfigurationstipps:

1. den Korrelationsbereich immer durch das Festlegen eines eindeutigen Tags für das Steuerereignis (das Ereignis, das mit alten Ereignissen verknüpft wird) einschränken und die Korrelationsbedingung „new event tag“ verwenden
2. bei Verwendung der Operation „close old event“ nicht vergessen, eine auf dem alten Ereignis basierende Bedingung hinzuzufügen, da andernfalls alle vorhandenen Probleme geschlossen werden könnten
3. die Verwendung allgemeiner Tagnamen vermeiden, die von verschiedenen Korrelationskonfigurationen verwendet werden

Task-Manager

Wenn die Einstellung ’Allow manual close’ für den Auslöser aktiviert ist, können Problemereignisse, die vom Auslöser generiert wurden, manuell geschlossen werden. Dies geschieht im Frontend beim Aktualisieren eines Problems. Das Ereignis wird nicht direkt geschlossen - stattdessen wird eine Aufgabe ’close event’ erstellt, die vom Task-Manager kurz darauf verarbeitet wird. Der Task-Manager erzeugt ein entsprechendes OK-Ereignis und das Problemereignis wird geschlossen.

Service-Ereignisse Service-Ereignisse werden nur generiert, wenn Service-Aktionen für diese Ereignisse aktiviert sind. In diesem Fall erzeugt jede Änderung des Service-Status ein neues Ereignis:

- Problem-Ereignis - wenn der Service-Status von `OK` zu `Problem` geändert wird
- OK-Ereignis - wenn der Service-Status von `Problem` zu `OK` geändert wird

Das Ereignis enthält Details zur Änderung des Service-Status - wann sie aufgetreten ist und wie der neue Status lautet.

Discovery-Ereignisse Zabbix durchsucht regelmäßig die in den Regeln zur Netzwerkerkennung definierten IP-Bereiche. Die Häufigkeit der Prüfung kann für jede Regel individuell konfiguriert werden. Sobald ein Host oder ein Dienst erkannt wird, wird ein Discovery-Ereignis (oder mehrere Ereignisse) generiert.

Zabbix generiert die folgenden Ereignisse:

Ereignis	Wann generiert
Service Up	Jedes Mal, wenn Zabbix einen aktiven Dienst erkennt.
Service Down	Jedes Mal, wenn Zabbix den Dienst nicht erkennen kann.
Host Up	Wenn mindestens einer der Dienste für die IP UP ist.
Host Down	Wenn alle Dienste nicht antworten.
Service Discovered	Wenn der Dienst nach einer Ausfallzeit wieder verfügbar ist oder zum ersten Mal erkannt wird.
Service Lost	Wenn der Dienst nach zuvor erfolgreichem Betrieb verloren geht.
Host Discovered	Wenn der Host nach einer Ausfallzeit wieder verfügbar ist oder zum ersten Mal erkannt wird.
Host Lost	Wenn der Host nach zuvor erfolgreichem Betrieb verloren geht.

Ereignisse der automatischen Registrierung aktiver Agenten Die automatische Registrierung aktiver Agenten erstellt Ereignisse in Zabbix.

Wenn sie konfiguriert ist, wird ein Ereignis der automatischen Registrierung aktiver Agenten erstellt, wenn ein zuvor unbekannter aktiver Agent nach Prüfungen fragt oder wenn sich die Host-Metadaten geändert haben. Der Server fügt einen neu automatisch registrierten Host hinzu und verwendet dabei die empfangene IP-Adresse und den Port des Agenten.

Weitere Informationen finden Sie auf der Seite zur automatischen Registrierung aktiver Agenten.

Interne Ereignisse Interne Ereignisse treten auf, wenn:

- ein Datenpunkt seinen Status von 'normal' zu 'unsupported' ändert
- ein Datenpunkt seinen Status von 'unsupported' zu 'normal' ändert
- eine Low-Level-Discovery-Regel ihren Status von 'normal' zu 'unsupported' ändert
- eine Low-Level-Discovery-Regel ihren Status von 'unsupported' zu 'normal' ändert
- ein Auslöser seinen Status von 'normal' zu 'unknown' ändert
- ein Auslöser seinen Status von 'unknown' zu 'normal' ändert

Ziel der Einführung interner Ereignisse ist es, Benutzer zu benachrichtigen, wenn ein internes Ereignis eintritt, zum Beispiel wenn ein Datenpunkt nicht mehr unterstützt wird und keine Daten mehr erfasst.

Interne Ereignisse werden nur erstellt, wenn interne Aktionen für diese Ereignisse aktiviert sind. Um die Generierung interner Ereignisse zu stoppen (zum Beispiel, wenn Datenpunkte nicht mehr unterstützt werden), deaktivieren Sie alle Aktionen für interne Ereignisse unter *Alerts* > *Actions* > *Internal actions*.

> **Note:**
> Wenn interne Aktionen deaktiviert sind, wird für dieses Objekt dennoch ein Wiederherstellungsereignis erstellt, während sich das Objekt im Status 'unsupported' befindet.
> Wenn interne Aktionen aktiviert sind, wird für dieses Objekt ein Wiederherstellungsereignis erstellt, während sich das Objekt im Status 'unsupported' befindet, auch wenn für das Objekt kein 'problem event' erstellt wurde.

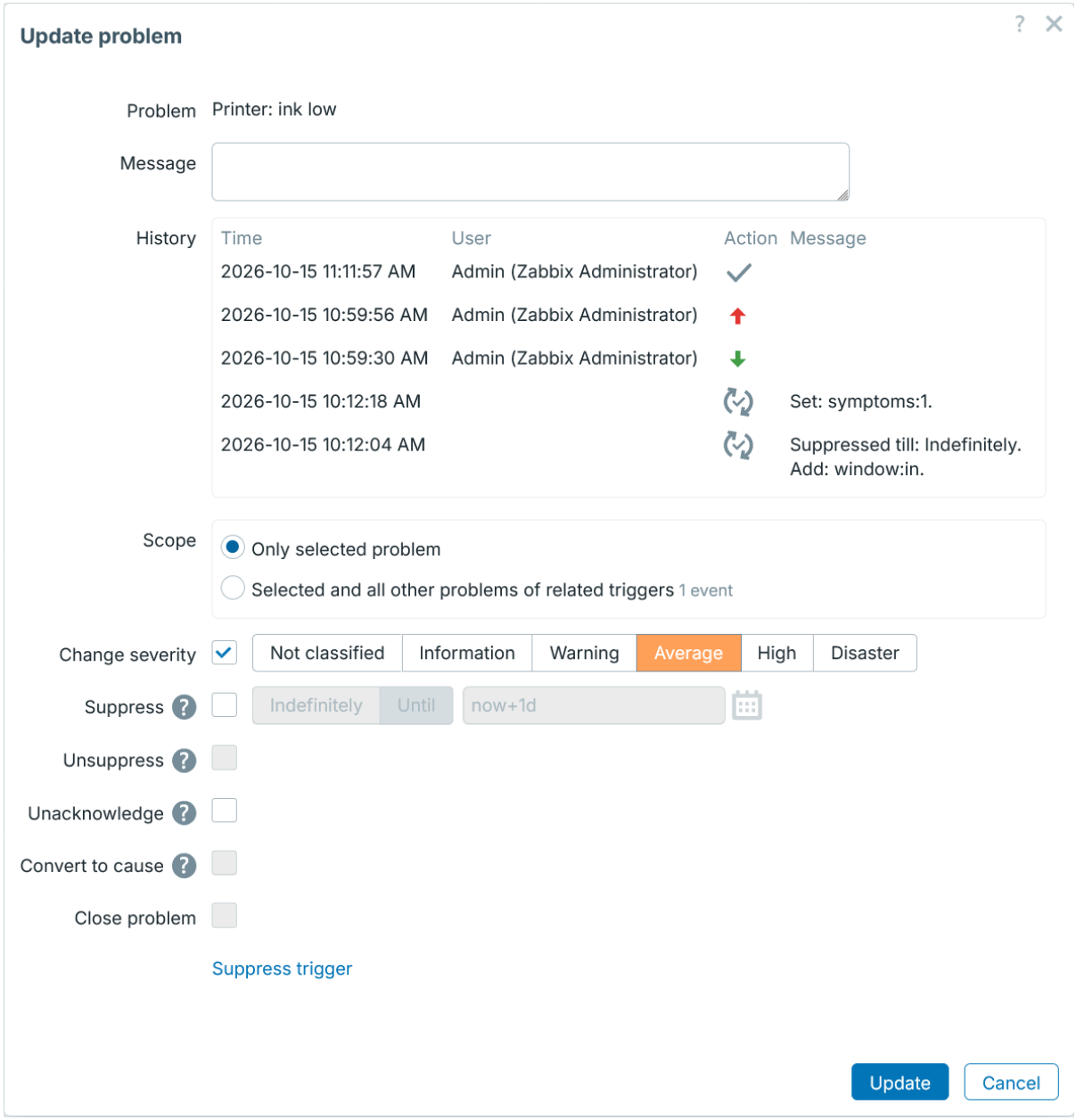
Siehe auch: Benachrichtigung bei nicht unterstützten Datenpunkten empfangen

Problem-Bestätigung und Unterdrückung

Problemereignisse in Zabbix können von Benutzern bestätigt oder unterdrückt werden.

Probleme aktualisieren Das Popup zur Problemaktualisierung ermöglicht Ihnen Folgendes:

- Das Problem kommentieren.
- Bisherige Kommentare und Aktionen anzeigen.
- Den Schweregrad des Problems ändern.
- Das Problem unterdrücken bzw. die Unterdrückung aufheben.
- Das Problem bestätigen bzw. die Bestätigung aufheben.
- Ein Symptomproblem in ein Ursprungsproblem umwandeln.
- Das Problem manuell schließen.
- Eine Ad-hoc-Wartung für den Auslöser erstellen.



Alle Pflichtfelder sind mit einem roten Sternchen gekennzeichnet.

Parameter	Beschreibung
Problem	Wenn nur ein Problem ausgewählt ist, wird der Problemname angezeigt. Wenn mehrere Probleme ausgewählt sind, wird *N problems selected* angezeigt.
Message	Geben Sie Text ein, um das Problem zu kommentieren (maximal 2048 Zeichen).
History	Bisherige Aktivitäten und Kommentare zum Problem werden zusammen mit Zeit- und Benutzerdetails aufgelistet. Die Bedeutung der Symbole für Benutzeraktionen finden Sie auf der Seite Ereignisdetails. Beachten Sie, dass der Verlauf nur angezeigt wird, wenn für die Aktualisierung ein einziges Problem ausgewählt ist.
Scope	Legen Sie den Geltungsbereich von Aktionen wie dem Ändern des Schweregrads, dem Bestätigen oder dem manuellen Schließen von Problemen fest: **Only selected problem** - betrifft nur dieses Ereignis **Selected and all other problems of related triggers** - beim Bestätigen oder Schließen eines Problems betrifft dies dieses Ereignis und alle anderen Probleme, die bisher nicht bestätigt bzw. geschlossen wurden. Wenn der Geltungsbereich bereits bestätigte oder geschlossene Probleme enthält, werden diese Probleme nicht erneut bestätigt bzw. geschlossen. Die Anzahl der Vorgänge zum Hinzufügen von Nachrichten und Ändern des Schweregrads ist dagegen nicht begrenzt.

Parameter	Beschreibung
Change severity	Aktivieren Sie das Kontrollkästchen und klicken Sie auf die Schaltfläche für den Schweregrad, um den Schweregrad des Problems zu aktualisieren. Das Kontrollkästchen zum Ändern des Schweregrads ist verfügbar, wenn für mindestens eines der ausgewählten Probleme Lese-/Schreibberechtigungen vorhanden sind. Beim Klicken auf *Update* werden nur Probleme aktualisiert, für die Lese-/Schreibberechtigungen bestehen. Wenn für keinen der ausgewählten Auslöser Lese-/Schreibberechtigungen vorhanden sind, ist das Kontrollkästchen deaktiviert.
Suppress	Aktivieren Sie das Kontrollkästchen, um das Problem zu unterdrücken: **Indefinitely** - auf unbestimmte Zeit unterdrücken **Until** - bis zu einem bestimmten Zeitpunkt unterdrücken. Sowohl absolute als auch relative Zeitformate werden unterstützt, zum Beispiel: `now+1d` - für einen Tag ab jetzt (Standard) `now/w` - bis zum Ende der aktuellen Woche `2022-05-28 12:00:00` - bis zum absoluten Datum/Zeitpunkt Beachten Sie, dass ein einfacher Zeitraum (z. B. `1d`, `1w`) nicht unterstützt wird. Die Verfügbarkeit dieser Option hängt von den Einstellungen der Benutzerrolle "Suppress problems" ab. Siehe auch: Problemunterdrückung
Unsuppress	Aktivieren Sie das Kontrollkästchen, um die Unterdrückung des Problems aufzuheben. Dieses Kontrollkästchen ist nur aktiv, wenn mindestens eines der ausgewählten Probleme unterdrückt wird. Die Verfügbarkeit dieser Option hängt von den Einstellungen der Benutzerrolle "Suppress problems" ab.
Acknowledge	Aktivieren Sie das Kontrollkästchen, um das Problem zu bestätigen. Dieses Kontrollkästchen ist verfügbar, wenn unter den ausgewählten Problemen mindestens ein unbestätigtes Problem vorhanden ist. Für ein bereits bestätigtes Problem kann keine weitere Bestätigung hinzugefügt werden (ein weiterer Kommentar kann jedoch hinzugefügt werden).
Unacknowledge	Aktivieren Sie das Kontrollkästchen, um die Bestätigung des Problems aufzuheben. Dieses Kontrollkästchen ist verfügbar, wenn unter den ausgewählten Problemen mindestens ein bestätigtes Problem vorhanden ist.
Convert to cause	Aktivieren Sie das Kontrollkästchen, um die Symptomprobleme in Ursprungsprobleme umzuwandeln.
Close problem	Aktivieren Sie das Kontrollkästchen, um die ausgewählten Probleme manuell zu schließen. Das Kontrollkästchen zum Schließen eines Problems ist verfügbar, wenn in der Auslöserkonfiguration für mindestens eines der ausgewählten Probleme die Option *Allow manual close* aktiviert ist. Beim Klicken auf *Update* werden nur Probleme geschlossen, die geschlossen werden dürfen. Wenn kein Problem manuell geschlossen werden kann, ist das Kontrollkästchen deaktiviert. Bereits geschlossene Probleme werden nicht erneut geschlossen.
Suppress trigger	Klicken Sie hier, um eine Ad-hoc-Wartung für den Auslöser zu starten. Ein Konfigurationsformular für einen Wartungszeitraum wird geöffnet. Darin ist ein einmaliger Wartungszeitraum von 1 h (konfigurierbar in den Benutzereinstellungen und im Profil) ab jetzt vorausgewählt. Der entsprechende Auslöser ist ebenfalls vorausgewählt. Der Name der Wartung wird im Format `Ad-hoc: <event name>` vorausgefüllt.

Problem acknowledgment Wenn ein Benutzer über ein Problemereignis benachrichtigt wird, kann er zum Zabbix Frontend gehen, das Popup-Fenster zur Problemaktualisierung für dieses Problem über eine der unten aufgeführten Möglichkeiten öffnen und das Problem bestätigen. Bei der Bestätigung kann er einen Kommentar dazu eingeben, etwa dass er daran arbeitet oder was auch immer er dazu sagen möchte.

Auf diese Weise sieht ein anderer Systembenutzer, der dasselbe Problem bemerkt, sofort, ob es bereits bestätigt wurde und welche Kommentare bisher vorliegen.

So kann der Workflow zur Behebung von Problemen mit mehr als einem Systembenutzer koordiniert ablaufen.

Der Bestätigungsstatus wird auch bei der Definition von action operations verwendet. Sie können zum Beispiel festlegen, dass eine Benachrichtigung an einen Manager auf höherer Ebene nur gesendet wird, wenn ein Ereignis für eine gewisse Zeit nicht bestätigt wurde.

Um Ereignisse zu bestätigen und Kommentare dazu zu hinterlassen, muss ein Benutzer mindestens Leseberechtigungen für die entsprechenden Auslöser haben. Um die Schwere eines Problems zu ändern oder ein Problem zu schließen, muss ein Benutzer Lese- und Schreibberechtigungen für die entsprechenden Auslöser haben.

Es gibt **mehrere** Möglichkeiten, auf das Popup-Fenster zur Problemaktualisierung zuzugreifen, mit dem ein Problem bestätigt werden kann.

- Sie können Probleme unter *Monitoring* > *Problems* auswählen und dann unterhalb der Liste auf *Mass update* klicken
- Sie können in der Spalte *Update* eines Problems auf *Update* klicken in:
  - *Dashboards* (*Problems* und *Problems by severity* Widgets)
  - *Monitoring* > *Problems*
  - *Monitoring* > *Problems* > *Event details*
- Sie können auf eine Zelle mit einem ungelösten Problem in:
  - *Dashboards* (*Trigger overview* Widget)

Das Popup-Menü enthält die Option *Update*, die Sie zum Fenster zur Problemaktualisierung führt.

Anhand von Bestätigungsinformationen kann konfiguriert werden, wie die Anzahl der Probleme im Dashboard oder in Karten angezeigt wird. Dazu müssen Sie in der Option *Problemanzeige* eine Auswahl treffen, die sowohl in der Kart enkonfiguration als auch im Dashboard-Widget „Probleme nach Schweregrad“ verfügbar ist. Es ist möglich, die Gesamtzahl der Probleme, die Anzahl der unbestätigten Probleme getrennt von der Gesamtzahl oder nur die Anzahl der unbestätigten Probleme anzuzeigen.

Anhand von Informationen zu Problemaktualisierungen (Bestätigung usw.) können Aktualisierungsvorgänge konfiguriert werden – eine Nachricht senden oder Remote-Befehle ausführen.

Unterdrückung von Problemen Die Unterdrückung von Problemen bietet eine Möglichkeit, ein Problem vorübergehend auszublenden und sich später darum zu kümmern. Dies ist nützlich, um die Problemliste zu bereinigen und den dringendsten Problemen die höchste Priorität zu geben. Beispielsweise kann am Wochenende ein Problem auftreten, das nicht dringend genug ist, um sofort bearbeitet zu werden. In diesem Fall kann es bis Montagmorgen „zurückgestellt“ werden.

Mit der Unterdrückung von Problemen kann ein *einzelnes* Problem ausgeblendet werden. Dies steht im Gegensatz zur Unterdrückung durch die Wartung eines Hosts, bei der alle Probleme des Wartungshosts ausgeblendet werden.

Operationen für Auslöseraktionen werden für unterdrückte Probleme genauso pausiert wie bei einer Host-Wartung.

Ein Problem kann über das Fenster **Problemaktualisierung** unterdrückt werden, wobei Unterdrückung eine der Optionen zur Problemaktualisierung ist, zusammen mit Kommentieren, Ändern des Schweregrads, Bestätigen usw.

Ein Problem kann auch über dasselbe Fenster zur Problemaktualisierung wieder aufgehoben werden.

Anzeige

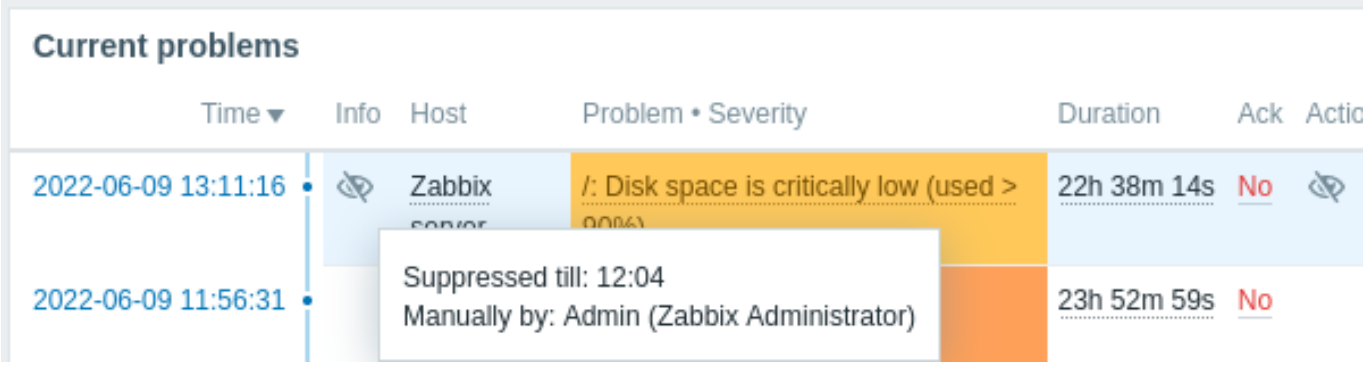
Sobald ein Problem unterdrückt wurde, wird es durch ein blinkendes  Unterdrückungssymbol in der Spalte *Info* markiert, bevor es ausgeblendet wird.

Wenn die Unterdrückung aufgrund einer Host-Wartung erfolgt, wird das Unterdrückungssymbol in der Spalte *Info* orange angezeigt. Wenn sowohl Wartung als auch manuelle Unterdrückung zutreffen, hat die manuelle Unterdrückung bei der Symbolfarbe Vorrang.

Das Unterdrückungssymbol blinkt, solange sich die Unterdrückungsaufgabe in der Warteliste befindet. Sobald der Aufgabenmanager das Problem unterdrückt hat, hört das Symbol auf zu blinken. Wenn das Unterdrückungssymbol über längere Zeit weiter blinkt, kann dies auf ein Serverproblem hinweisen, zum Beispiel wenn der Server ausgefallen ist und der Aufgabenmanager die Aufgabe nicht abschließen kann. Die gleiche Logik gilt für die Aufhebung der Unterdrückung. In der kurzen Zeit nach dem Absenden der Aufgabe und bevor der Server sie abgeschlossen hat, blinkt das Symbol zur Aufhebung der Unterdrückung.

Ein unterdrücktes Problem kann je nach Einstellungen des Problemfilters bzw. Widgets sowohl ausgeblendet als auch angezeigt werden.

Wenn es in der Problemliste angezeigt wird, ist ein unterdrücktes Problem mit dem Unterdrückungssymbol markiert, und die Unterdrückungsdetails werden beim Überfahren mit der Maus angezeigt:



Unterdrückungsdetails werden auch in einem Popup angezeigt, wenn der Mauszeiger auf das Unterdrückungssymbol in der Spalte *Aktionen* gesetzt wird.

Manuelles Schließen von Problemen

Übersicht Während Problemereignisse im Allgemeinen automatisch behoben werden, wenn der Auslöserstatus von `Problem` zu `OK` wechselt, kann es Fälle geben, in denen es schwierig ist festzustellen, ob ein Problem mithilfe eines Auslöserausdrucks behoben wurde. In solchen Fällen muss das Problem manuell behoben werden.

Beispielsweise kann *syslog* melden, dass einige Kernel-Parameter für eine optimale Leistung angepasst werden müssen. In diesem Fall wird das Problem an die Linux-Administratoren gemeldet, diese beheben es und schließen das Problem anschließend manuell.

Probleme können nur für Auslöser manuell geschlossen werden, bei denen die Option *Manuelles Schließen erlauben* aktiviert ist.

Wenn ein Problem "manuell geschlossen" wird, erzeugt Zabbix eine neue interne Aufgabe für den Zabbix Server. Anschließend führt der Prozess *task manager* diese Aufgabe aus und erzeugt ein OK-Ereignis, wodurch das Problemereignis geschlossen wird.

Das erzeugte *OK*-Ereignis enthält den vollständigen Satz der für dieses Ereignis aufgelösten Ereignistags (einschließlich der von Vorlagen, Hosts und Auslösern geerbten Tags). Diese Tags sind in Benachrichtigungen und in Makros wie `{EVENT.RECOVERY.TAGS}` und `{EVENT.RECOVERY.TAGSJSON}` verfügbar.

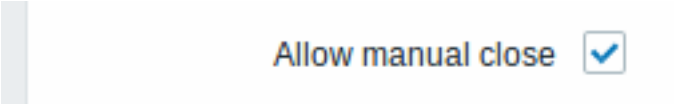
Ein manuell geschlossenes Problem bedeutet nicht, dass der zugrunde liegende Auslöser niemals wieder in den Status *Problem* wechseln wird. Der Auslöserausdruck wird erneut ausgewertet und kann zu einem Problem führen:

- Wenn neue Daten für einen beliebigen in den Auslöserausdruck einbezogenen Datenpunkt eintreffen (beachten Sie, dass die durch einen Drosselungs-Preprocessing-Schritt verworfenen Werte nicht als empfangen gelten und keine erneute Auswertung des Auslöserausdrucks auslösen);
- Wenn im Ausdruck Funktionen für Datum und Uhrzeit und/oder nodata() verwendet werden.

Konfiguration Zum manuellen Schließen eines Problems sind zwei Schritte erforderlich.

Auslöser-Konfiguration

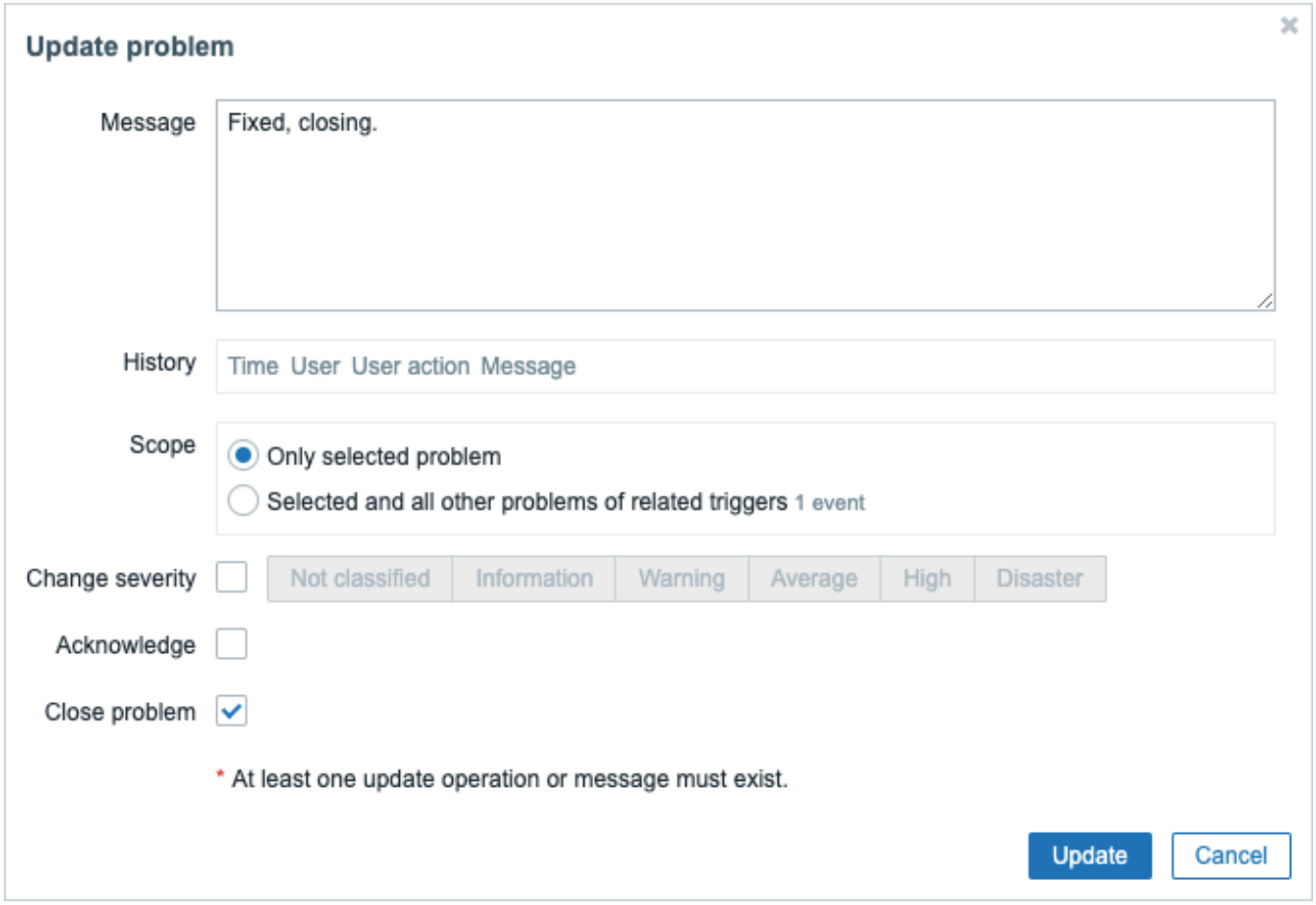
Aktivieren Sie in der Auslöser-Konfiguration die Option *Manuelles Schließen erlauben*.



Problemaktualisierungsfenster

Wenn bei einem Auslöser mit dem Flag *Manuelles Schließen* ein Problem auftritt, können Sie das Popup-Fenster Problemaktualisierung dieses Problems öffnen und das Problem manuell schließen.

Um das Problem zu schließen, aktivieren Sie im Formular die Option *Problem schließen* und klicken Sie auf *Aktualisieren*.



Alle Pflichtfelder sind mit einem roten Sternchen markiert.

Die Anfrage wird vom Zabbix Server verarbeitet. Normalerweise dauert es einige Sekunden, bis das Problem geschlossen wird. Während dieses Vorgangs wird in *Monitoring* > *Probleme* der Status des Problems als *CLOSING* angezeigt.

Verifizierung Es kann überprüft werden, dass ein Problem manuell geschlossen wurde:

- in den Ereignisdetails, verfügbar über *Monitoring* > *Probleme*;
- durch Verwendung des Makros `{EVENT.UPDATE.HISTORY}` in Benachrichtigungsnachrichten, das diese Information bereitstellt.

Wartungszeiträume

Übersicht

Mit Wartungen können Probleme während vordefinierter Zeiträume unterdrückt werden.

Sie können Wartungszeiträume für folgende Entitäten definieren:

- Hostgruppen
- Hosts
- Auslöser
- Ereignisnamen
- Tags

Wenn eine Wartung für bestimmte Tags oder Ereignisnamen definiert ist, wird die Wartung nur für die entsprechenden Auslöser aktiviert; alle anderen Auslöser des Hosts oder der Hostgruppe befinden sich nicht in Wartung.

Es gibt zwei Wartungstypen: *Mit Datensammlung* und *Keine Datensammlung*.

Während einer Wartung vom Typ *Mit Datensammlung* werden Auslöser wie gewohnt verarbeitet und bei Bedarf Ereignisse erstellt. Wenn jedoch die Option *Vorgänge für unterdrückte Probleme pausieren* in der Aktionskonfiguration aktiviert ist, werden Problemeskalationen für Entitäten in Wartung pausiert. In diesem Fall werden Eskalationsschritte, die das Senden von Benachrichtigungen oder Remote-Befehlen umfassen können, für die Dauer der Wartung ignoriert. Beachten Sie, dass die Vorgänge zur

Problembehebung und Aktualisierung während der Wartung nicht unterdrückt werden, sondern nur Eskalationen. Benachrichtigungen zur Problembehebung werden nicht gesendet, wenn das Problem während der Wartung begonnen hat.

Wenn beispielsweise Eskalationsschritte 0, 30 und 60 Minuten nach Beginn eines Problems geplant sind und ein halbstündiger Wartungszeitraum von 10 bis 40 Minuten nach dem Auftreten eines tatsächlichen Problems stattfindet, werden die Schritte zwei und drei eine halbe Stunde später ausgeführt, also nach 60 bzw. 90 Minuten (vorausgesetzt, das Problem besteht weiterhin). Wenn ein Problem während der Wartung auftritt, beginnt die Eskalation ebenfalls erst nach Ende der Wartung.

Um Problembenachrichtigungen während der Wartung normal, also ohne Verzögerung, zu erhalten, müssen Sie die Option *Vorgänge für unterdrückte Probleme pausieren* in der Aktionskonfiguration deaktivieren.

> **Note:**
> Wenn sich mindestens ein Host (der im Ausdruck des Auslösers verwendet wird) nicht im Wartungsmodus befindet, sendet Zabbix eine Problembenachrichtigung.

Der Zabbix Server muss während der Wartung ausgeführt werden. Wartungen werden jede Minute neu berechnet oder sobald der Konfigurations-Cache neu geladen wird, wenn Änderungen am Wartungszeitraum vorgenommen wurden.

Timer-Prozesse prüfen bei 0 Sekunden jeder Minute, ob der Status eines Hosts in den Wartungsmodus oder aus dem Wartungsmodus geändert werden muss. Zusätzlich prüft der Timer-Prozess jede Sekunde, ob Wartungen gestartet oder beendet werden müssen, basierend darauf, ob sich nach der Konfigurationsaktualisierung Änderungen an den [Wartungszeiträumen] ergeben haben. Daher hängt die Geschwindigkeit, mit der Wartungszeiträume gestartet oder beendet werden, vom Konfigurations-Aktualisierungsintervall ab (standardmäßig 10 Sekunden). Beachten Sie, dass Änderungen an Wartungszeiträumen die Einstellungen *Aktiv seit/Aktiv bis* nicht umfassen. Wenn außerdem eine Entität (Hostgruppe/Host/Auslöser usw.) zu einem bestehenden aktiven Wartungszeitraum hinzugefügt wird, werden die Änderungen erst zu Beginn der nächsten Minute durch den Timer-Prozess aktiviert.

Beachten Sie, dass die Timer-Prozesse des Zabbix Servers beim Eintritt eines Hosts in die Wartung alle offenen Probleme lesen, um zu prüfen, ob diese unterdrückt werden müssen. Dies kann sich auf die Leistung auswirken, wenn viele offene Probleme vorhanden sind. Der Zabbix Server liest beim Start ebenfalls alle offenen Probleme, selbst wenn zu diesem Zeitpunkt keine Wartungen konfiguriert sind.

Beachten Sie, dass der Zabbix Server (oder Proxy) unabhängig vom Wartungstyp immer Daten sammelt (einschließlich Wartungen vom Typ *Keine Datensammlung*). Die Daten werden vom Server später ignoriert, wenn *Keine Datensammlung* festgelegt ist.

Wenn eine Wartung vom Typ *Keine Datensammlung* endet, werden Auslöser, die die Funktion `nodata()` verwenden, erst bei der nächsten Prüfung innerhalb des von ihnen überprüften Zeitraums ausgelöst.

Wenn ein Log-Datenpunkt hinzugefügt wird, während sich ein Host in Wartung befindet, und die Wartung endet, werden nur neue Logdateieinträge seit dem Ende der Wartung erfasst.

Wenn ein Zeitstempelwert für einen Host gesendet wird, der sich in einer Wartung vom Typ *Keine Datensammlung* befindet (z. B. mit Zabbix sender), wird dieser Wert verworfen. Es ist jedoch möglich, einen Zeitstempelwert für einen abgelaufenen Wartungszeitraum zu senden; dieser wird akzeptiert.

Wenn der Benutzer den Wartungszeitraum oder die Wartungsoption (Hostgruppe, Host, Auslöser usw.) ändert, werden die Änderungen erst nach der Synchronisierung des Konfigurations-Caches wirksam.

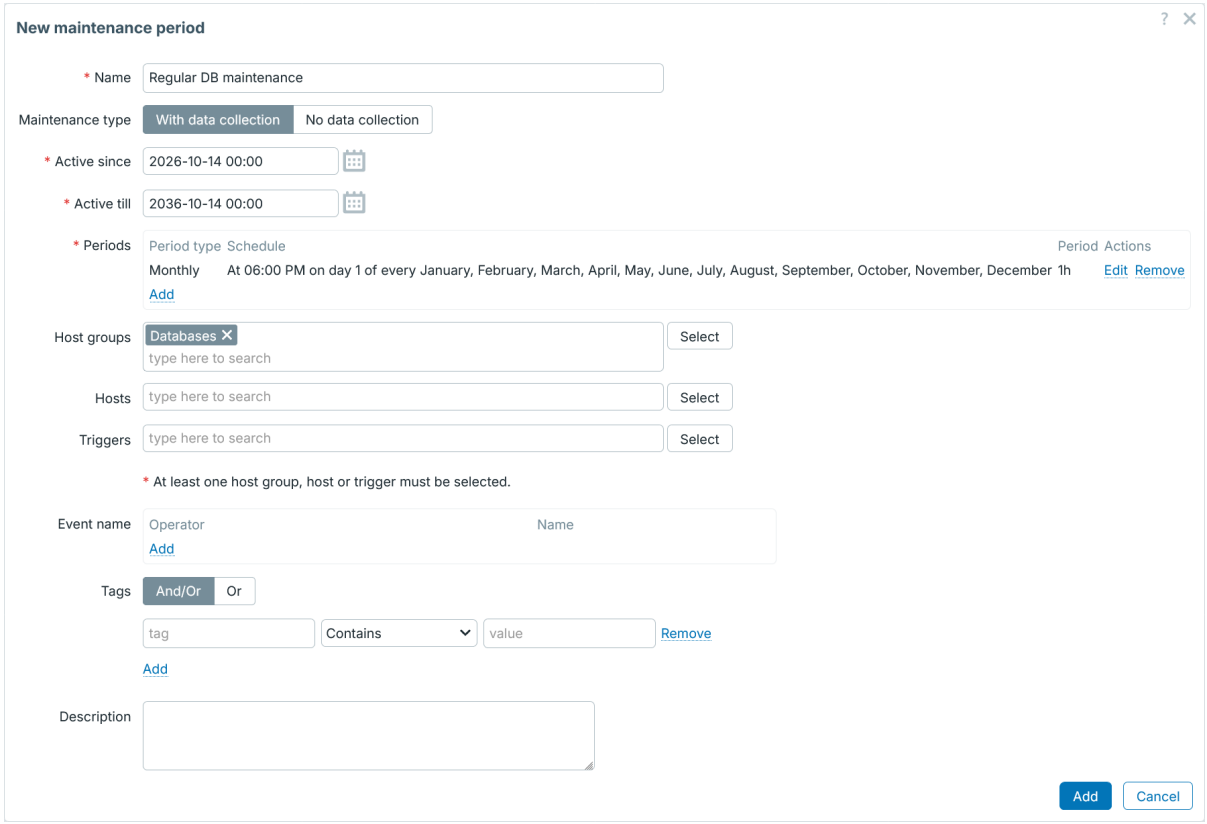
Konfiguration

So starten Sie einen Wartungszeitraum sofort:

1. Gehen Sie zu *Monitoring* > *Probleme*.
2. Klicken Sie auf den Namen des Problems.
3. Wählen Sie im Ereignismenü die Wartungsoption aus (Host, Auslöser, Ereignisname, Tags).
4. Ändern Sie die Wartungsparameter im Formular.

So konfigurieren Sie einen geplanten Wartungszeitraum:

1. Gehen Sie zu *Datensammlung* > *Wartung*.
2. Klicken Sie auf *Wartungszeitraum erstellen* (oder auf den Namen eines vorhandenen Wartungszeitraums).
3. Geben Sie die Wartungsparameter im Formular ein.



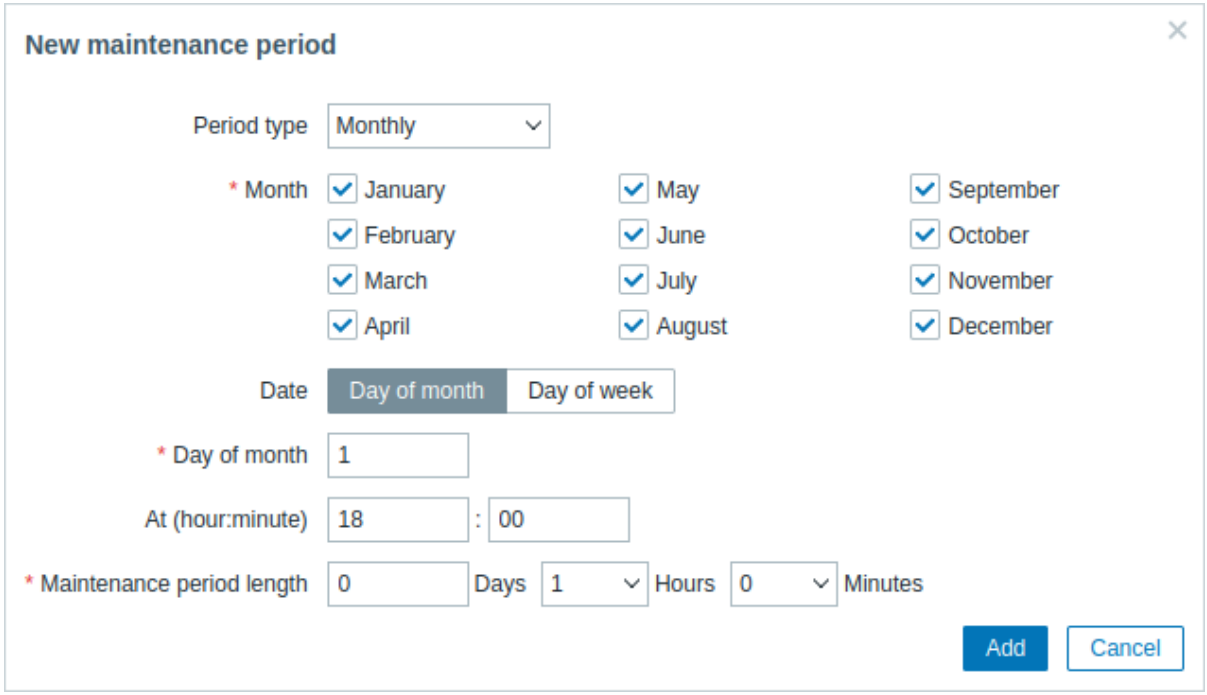
Alle Pflichtfelder sind mit einem roten Sternchen gekennzeichnet.

Parameter	Beschreibung
Name	Name des Wartungszeitraums.
Wartungstyp	Es können zwei Wartungstypen festgelegt werden: **Mit Datensammlung** - Während der Wartung werden Daten vom Server gesammelt und Auslöser verarbeitet; **Keine Datensammlung** - Daten können weiterhin gesammelt werden, werden während der Wartung jedoch nicht in der Datenbank gespeichert, und Auslöser (einschließlich der Funktion `nodata()`) werden nicht ausgelöst. Weitere Informationen dazu, wie sich die einzelnen Typen auf den Verfügbarkeitsbericht auswirken, finden Sie unter Auswirkungen von Wartungszeiträumen.
Aktiv seit	Datum und Uhrzeit, ab denen die Ausführung von Wartungszeiträumen aktiv wird. *Hinweis:* Durch das alleinige Festlegen dieses Zeitpunkts wird ein Wartungszeitraum nicht aktiviert; Wartungszeiträume müssen unter *Zeiträume* konfiguriert werden (siehe unten).
Aktiv bis	Datum und Uhrzeit, bis zu denen die Ausführung von Wartungszeiträumen aktiv bleibt.
Zeiträume	In diesem Block können Sie die genauen Tage und Uhrzeiten definieren, an denen die Wartung stattfindet. Durch Klicken auf Add wird ein Popup-Fenster mit einem flexiblen Formular für den *Wartungszeitraum* geöffnet, in dem Sie den Wartungsplan definieren können. Eine ausführliche Beschreibung finden Sie unter Wartungszeiträume.
Hostgruppen	Wählen Sie die Hostgruppen aus, für die die Wartung aktiviert wird. Die Wartung wird für alle Hosts aus den angegebenen Hostgruppen aktiviert. Dieses Feld unterstützt die automatische Vervollständigung. Wenn Sie mit der Eingabe beginnen, wird eine Dropdown-Liste mit allen verfügbaren Hostgruppen angezeigt. Durch Angabe einer übergeordneten Hostgruppe werden implizit alle verschachtelten Hostgruppen ausgewählt. Daher wird die Wartung auch für Hosts aus verschachtelten Gruppen aktiviert.
Hosts	Wählen Sie die Hosts aus, für die die Wartung aktiviert wird. Dieses Feld unterstützt die automatische Vervollständigung. Wenn Sie mit der Eingabe beginnen, wird eine Dropdown-Liste mit allen verfügbaren Hosts angezeigt.
Auslöser	Wählen Sie die Auslöser aus, für die die Wartung aktiviert wird. Dieses Feld unterstützt die automatische Vervollständigung. Wenn Sie mit der Eingabe beginnen, wird eine Dropdown-Liste mit allen verfügbaren Auslösern angezeigt.

Parameter	Beschreibung
Ereignisname	Geben Sie die Ereignisnamen an, für die die Wartung aktiviert wird. Es können mehrere Bedingungen festgelegt werden. Für jede Bedingung stehen zwei Operatoren zur Verfügung: **Enthält** - schließt die angegebenen Tag-Namen ein, wenn die Tag-Werte die eingegebene Zeichenfolge enthalten (Teilstringvergleich, ohne Beachtung der Groß-/Kleinschreibung) **Enthält nicht** - schließt die angegebenen Tag-Namen aus, wenn die Tag-Werte die eingegebene Zeichenfolge enthalten (Teilstringvergleich, ohne Beachtung der Groß-/Kleinschreibung)
Tags	Geben Sie die Tags an, für die die Wartung aktiviert wird. Es können mehrere Bedingungen festgelegt werden. Beim Abgleich von Tag-Namen wird immer zwischen Groß- und Kleinschreibung unterschieden. Für jede Bedingung stehen mehrere Operatoren zur Verfügung: **Enthält** - schließt die angegebenen Tag-Namen ein, wenn die Tag-Werte die eingegebene Zeichenfolge enthalten (Teilstringvergleich, unter Beachtung der Groß-/Kleinschreibung); **Ist gleich** - schließt die angegebenen Tag-Namen und -Werte ein (unter Beachtung der Groß-/Kleinschreibung). **Ist nicht gleich** - schließt die angegebenen Tag-Namen und -Werte aus (unter Beachtung der Groß-/Kleinschreibung) **Enthält nicht** - schließt die angegebenen Tag-Namen aus, wenn die Tag-Werte die eingegebene Zeichenfolge enthalten (Teilstringvergleich, ohne Beachtung der Groß-/Kleinschreibung) Für Bedingungen stehen zwei Berechnungstypen zur Verfügung: **Und/Oder** - alle Bedingungen müssen erfüllt sein; Bedingungen mit demselben Tag-Namen werden mit der Oder-Bedingung gruppiert; **Oder** - es genügt, wenn eine Bedingung erfüllt ist. Tags können nur angegeben werden, wenn der Wartungstyp *Mit Datensammlung* ausgewählt ist.
Beschreibung	Beschreibung des Wartungszeitraums.

Wartungszeiträume

Das Fenster für den Wartungszeitraum dient zur Planung einer wiederkehrenden oder einmaligen Wartung. Das Formular ist dynamisch; die verfügbaren Felder ändern sich je nach ausgewähltem *Period type*.



Period type	Beschreibung
One time only	Konfigurieren Sie einen einmaligen Wartungszeitraum: *Date* - Datum und Uhrzeit des Wartungszeitraums; *Maintenance period length* - wie lange die Wartung aktiv sein soll.

Period type	Beschreibung
Daily	Konfigurieren Sie einen täglichen Wartungszeitraum: *Every day(s)* - Wartungshäufigkeit (1 - *(Standard)* jeden Tag, 2 - alle zwei Tage usw.); *At (hour:minute)* - Uhrzeit des Tages, zu der die Wartung beginnt; *Maintenance period length* - wie lange die Wartung aktiv sein soll. Wenn der Parameter *Every day(s)* größer als ”1” ist, ist der Starttag der Tag, auf den die Zeit *Active since* fällt. Beispiele: - wenn *Active since* auf ”2021-01-01 12:00” gesetzt ist, *Every day(s)* auf ”2” und *At (hour:minute)* auf ”23:00”, dann beginnt der erste Wartungszeitraum am 1. Januar um 23:00 Uhr, während der zweite Wartungszeitraum am 3. Januar um 23:00 Uhr beginnt; - wenn *Active since* auf ”2021-01-01 12:00” gesetzt ist, *Every day(s)* auf ”2” und *At (hour:minute)* auf ”01:00”, dann beginnt der erste Wartungszeitraum am 3. Januar um 01:00 Uhr, während der zweite Wartungszeitraum am 5. Januar um 01:00 Uhr beginnt.
Weekly	Konfigurieren Sie einen wöchentlichen Wartungszeitraum: *Every week(s)* - Wartungshäufigkeit (1 - *(Standard)* jede Woche, 2 - alle zwei Wochen usw.); *Day of week* - an welchem Wochentag die Wartung stattfinden soll; *At (hour:minute)* - Uhrzeit des Tages, zu der die Wartung beginnt; *Maintenance period length* - wie lange die Wartung aktiv sein soll. Wenn der Parameter *Every week(s)* größer als ”1” ist, ist die Startwoche die Woche, auf die die Zeit *Active since* fällt. Beispiele finden Sie in der Beschreibung des Parameters *Daily* oben.
Monthly	Konfigurieren Sie einen monatlichen Wartungszeitraum: *Month* - wählen Sie alle Monate aus, in denen die regelmäßige Wartung durchgeführt wird; *Date:* ***Day of month*** - wählen Sie diese Option, wenn die Wartung an jedem Monat am selben Datum stattfinden soll (zum Beispiel am 1. Tag jedes Monats), und wählen Sie dann im angezeigten Feld *Day of month* den gewünschten Tag aus; *Date:* ***Day of week*** - wählen Sie diese Option, wenn die Wartung nur an bestimmten Tagen stattfinden soll (zum Beispiel jeden ersten Montag im Monat), wählen Sie dann in der Dropdown-Liste die gewünschte Woche des Monats aus (erste, zweite, dritte, vierte oder letzte) und markieren Sie anschließend die Kontrollkästchen für den/die Wartungstag(e); *At (hour:minute)* - Uhrzeit des Tages, zu der die Wartung beginnt; *Maintenance period length* - wie lange die Wartung aktiv sein soll.

Attention:
Beim Erstellen eines Wartungszeitraums wird die Zeitzone des Benutzers verwendet, der ihn erstellt. Wenn jedoch wiederkehrende Wartungszeiträume (*Täglich*, *Wöchentlich*, *Monatlich*) geplant werden, wird die Zeitzone des Zabbix-Servers verwendet. Um ein vorhersehbares Verhalten wiederkehrender Wartungszeiträume sicherzustellen, ist es erforderlich, für alle Teile von Zabbix eine gemeinsame Zeitzone zu verwenden.

Wenn Sie fertig sind, klicken Sie auf *Add*, um den Wartungszeitraum zum Abschnitt *Periods* hinzuzufügen.

Beachten Sie, dass Änderungen der Sommerzeit (DST) keinen Einfluss darauf haben, wie lange die Wartung dauert. Angenommen, wir haben eine auf zwei Stunden konfigurierte Wartung, die normalerweise um 01:00 beginnt und um 03:00 endet:

- wenn nach einer Stunde Wartung (um 02:00) eine DST-Änderung eintritt und sich die aktuelle Zeit von 02:00 auf 03:00 ändert, läuft die Wartung noch eine weitere Stunde weiter (bis 04:00);
- wenn nach zwei Stunden Wartung (um 03:00) eine DST-Änderung eintritt und sich die aktuelle Zeit von 03:00 auf 02:00 ändert, wird die Wartung beendet, da zwei Stunden vergangen sind;
- wenn ein Wartungszeitraum während der Stunde beginnt, die durch eine DST-Änderung übersprungen wird, dann beginnt die Wartung nicht.

Wenn ein Wartungszeitraum auf ”1 day” gesetzt ist (der tatsächliche Wartungszeitraum beträgt 24 Stunden, da Zabbix Tage in Stunden berechnet), um 00:00 beginnt und am nächsten Tag um 00:00 endet:

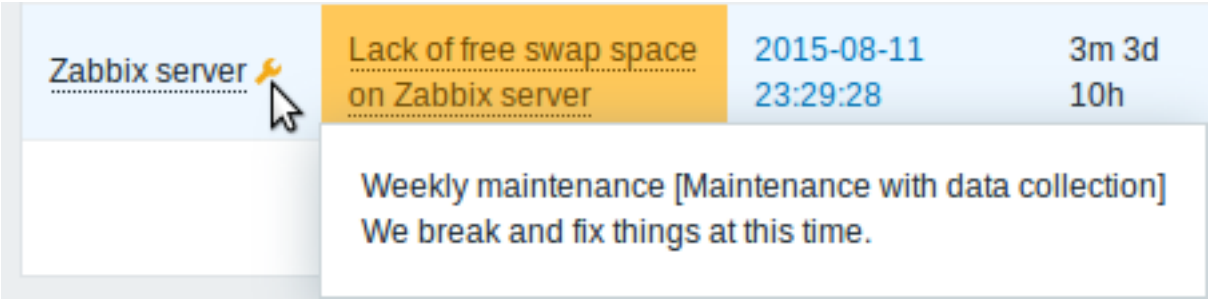
- die Wartung endet am nächsten Tag um 01:00, wenn sich die aktuelle Zeit um eine Stunde nach vorne verschiebt;
- die Wartung endet an diesem Tag um 23:00, wenn sich die aktuelle Zeit um eine Stunde zurück verschiebt.

Anzeige

Anzeige von Hosts in Wartung

Ein orangefarbenes Schraubenschlüssel-Symbol neben dem Hostnamen zeigt an, dass sich dieser Host in Wartung befindet in:

- *Dashboards*
- *Monitoring* > *Probleme*
- *Inventory* > *Hosts* > *Host inventory details*
- *Data collection* > *Hosts* (siehe Spalte *Status*)



Details zur Wartung werden angezeigt, wenn der Mauszeiger über dem Symbol positioniert wird.

Zusätzlich erhalten Hosts in Wartung einen orangefarbenen Hintergrund in *Monitoring* > *Maps*.

Angezeigte unterdrückte Probleme

Normalerweise werden Probleme für Hosts in Wartung unterdrückt, das heißt, sie werden im Frontend nicht angezeigt.

Unterdrückte Probleme können auch ausgeblendet sein, weil sie manuell unterdrückt wurden.

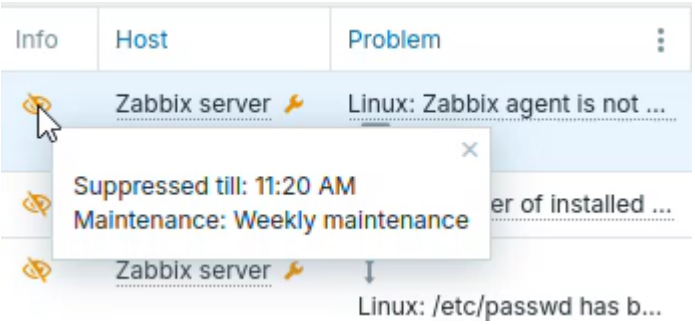
Es ist möglich, unterdrückte Probleme anzuzeigen, indem Sie die Option *Unterdrückte Probleme anzeigen* an diesen Stellen auswählen:

- *Dashboards* (in der Widget-Konfiguration *Problem-Hosts*, *Probleme*, *Probleme nach Schweregrad*, *Trigger-Übersicht*)
- *Überwachung* > *Probleme* (im Filter)
- *Überwachung* > *Karten* (in der Kartenkonfiguration)
- Globale Benachrichtigungen (in der Konfiguration des Benutzerprofils)

Wenn unterdrückte Probleme angezeigt werden, wird das folgende Symbol angezeigt:  . Wenn die Unterdrückung auf Host-Wartung zurückzuführen ist, werden die folgenden Symbole angezeigt:

-  - Problem wurde aufgrund von Host-Wartung unterdrückt.
-  - Problem wurde nach der Host-Wartung nicht mehr unterdrückt.

Wenn Sie mit der Maus über das Symbol fahren, werden weitere Details angezeigt. In den Ereignisdetails können Sie prüfen, ob das Ereignis aufgrund von Wartung oder manuell unterdrückt wurde. Wenn sowohl Wartung als auch manuelle Unterdrückung zutreffen, hat die manuelle Unterdrückung bei der Symbolfarbe Vorrang.



Berechnung von Warteschlangen während der Wartung

Die in der Zabbix Frontend (Administration > Queue) angezeigten Warteschlangen werden vom Zabbix Server berechnet. Sie enthalten keine Datenpunkte in der Wartung *No data collection*, und die Warteschlangenlänge ist für diese Datenpunkte immer null, selbst wenn ihre Werte verzögert sind. Verzögerte Datenpunkte in der Wartung *With data collection* werden weiterhin in der Warteschlange gezählt.

Der Zabbix Proxy ist sich der Wartungszeiträume nicht bewusst, da keine Synchronisierung der Wartungskonfiguration zwischen dem Zabbix Server und dem Proxy erfolgt. Interne Prüfungen, die auf Zabbix Proxys berechnet werden (zum Beispiel `zabbix[queue,,]` und `zabbix[stats,,,queue,,]`), melden verzögerte Datenpunkte unabhängig vom Wartungsstatus auf dem Zabbix Server.

Infolgedessen können für dieselben Datenpunkte in der Wartung *No data collection* unterschiedliche Warteschlangenlängen vom Zabbix Frontend und von internen Prüfungen auf Zabbix Proxys gemeldet werden.

Tagging

Tags haben mehrere Zwecke, insbesondere zum Markieren von events. Wenn Entitäten mit Tags versehen werden, erbt jedes neue Ereignis, das mit einer getaggten Entität zusammenhängt, deren Tags. Zum Beispiel:

- bei getaggten Vorlagen - jedes Host-Problem (das von Auslösern aus der Vorlage erstellt wird) erbt die Vorlagen-Tags.
- bei getaggten Hosts - jedes Host-Problem erbt die Host-Tags.
- bei getaggten Datenpunkten/Web-Szenarien - jedes Problem eines Datenpunkts/Web-Szenarios erbt die Tags des Datenpunkts/Web-Szenarios.
- bei getaggten Auslösern - jedes vom Auslöser erstellte Problem erbt die Auslöser-Tags.

Ein Problemereignis erbt alle Tags aus der gesamten Kette von Entitäten - Vorlagen, Hosts, Datenpunkten/Web-Szenarien, Auslösern. Identische `tag:value`-Kombinationen (nach Auflösung der Makros) werden zu einer zusammengeführt, wodurch Duplikate vermieden werden.

Wiederherstellungsereignisse, die durch ein manuelles Schließen erzeugt werden, enthalten ebenfalls die aufgelösten Ereignis-Tags, die von Vorlagen, Hosts, Datenpunkten/Web-Szenarien und Auslösern geerbt wurden. Diese Tags sind in Benachrichtigungen und über Makros wie `{EVENT.RECOVERY.TAGS}` und `{EVENT.RECOVERY.TAGSJSON}` verfügbar.

Benutzerdefinierte Ereignis-Tags bieten mehr Flexibilität. Zum Beispiel:

- Ereigniskorrelation kann auf Basis von Ereignis-Tags konfiguriert werden.
- Aktionsbedingungen können auf Basis von Ereignis-Tags konfiguriert werden.
- Probleme von Datenpunkten können auf Basis von Ereignis-Tags gruppiert werden.
- Problem-Tags können verwendet werden, um Probleme services zuzuordnen.

Entitäten können mit demselben Tag-Namen, aber unterschiedlichen Tag-Werten versehen werden (zum Beispiel `component:memory` und `component:storage`). Ebenso kann eine Entität einen Tag ohne Wert und denselben Tag mit einem Wert haben (zum Beispiel `database` und `database:postgresql`). Solche Tags gelten nicht als Duplikate.

Tags bestehen aus einem Tag-Namen und einem Tag-Wert. Beim Taggen von Entitäten können Sie nur den Namen verwenden oder ihn mit einem Wert kombinieren (zum Beispiel `mysql`, `jira`, `target:mysql`, `service:jira` usw.).

Tags können für verschiedene Entitäten definiert werden:

- Vorlagen
- Hosts
- Datenpunkte
- Web-Szenarien
- Auslöser
- Services
- Vorlagen-Datenpunkte und -Auslöser
- Host-, Datenpunkt- und Auslöser-Prototypen

In den Listen *Vorlagen*, *Hosts*, *Datenpunkte*, *Auslöser*, *Web-Szenarien* und ihren Prototypen zeigt die Spalte *Tags* sowohl reguläre als auch geerbte Tags an. Wenn Sie mit der Maus über einen *geerbten* Tag fahren oder darauf klicken, erscheint ein JavaScript-Tooltip mit dem Text "Inherited tag". Wenn ein geerbter Tag auch als regulärer Tag vorhanden ist, wird je nach Liste, in der er angezeigt wird, ein anderer Tooltip-Text angezeigt (zum Beispiel "Inherited and template tag" in der Liste Vorlagen oder "Inherited and host tag" in der Liste Hosts). An anderen Stellen des Frontends werden reguläre und geerbte Tags zusammen angezeigt (Duplikate werden entfernt), ohne Symbole und ohne zusätzlichen Tooltip-Text.

> **Note:**
> Beachten Sie die offiziellen Zabbix-Richtlinien für allgemeine Empfehlungen zur Definition von Tags sowie die spezifischen Hinweise für Vorlagen, Datenpunkte, Auslöser und Low-Level-Discovery-Regeln.

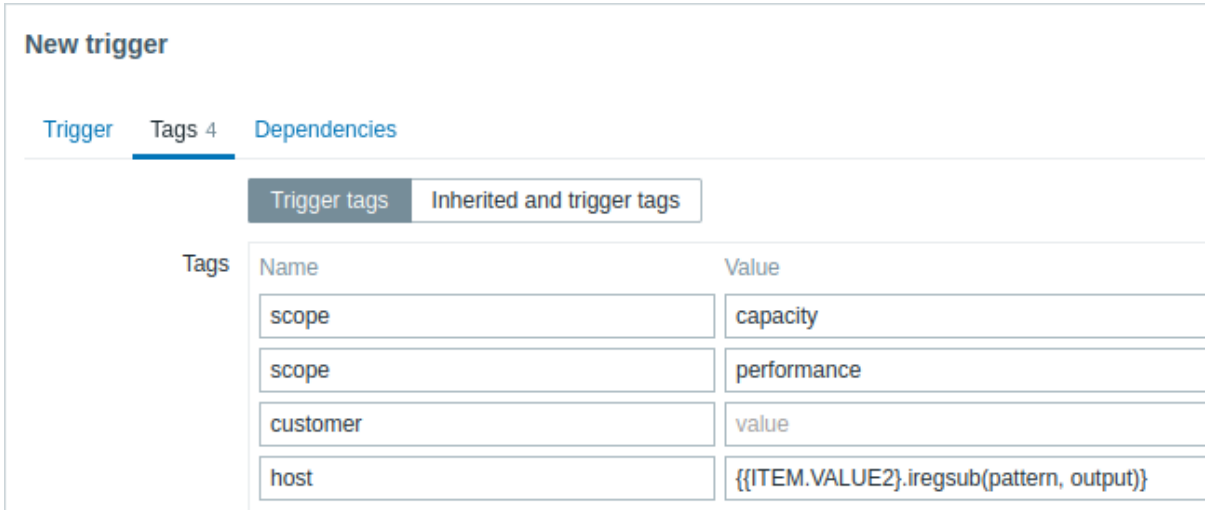
Anwendungsfälle Einige häufige Anwendungsfälle für Tagging sind wie folgt:

1. Trigger-Ereignisse markieren:
   - Definieren Sie einen Trigger-Tag (zum Beispiel `scope:performance`).
   - Von diesem Trigger erzeugte Probleme erhalten den Trigger-Tag.
2. Von Vorlagen geerbte Probleme markieren:
   - Definieren Sie einen Vorlage-Tag (zum Beispiel `target:mysql`).
   - Von Triggern aus dieser Vorlage erzeugte Probleme erhalten den Vorlage-Tag.
3. Host-Probleme markieren:
   - Definieren Sie einen Host-Tag (zum Beispiel `service:jira`).

- Von Triggern dieses Hosts erzeugte Probleme erhalten den Host-Tag.

4. Zugehörige Datenpunkte filtern:
   - Definieren Sie einen Datenpunkt-Tag (zum Beispiel `component:cpu`).
   - In *Monitoring* > *Latest data* können Datenpunkte nach dem Tag `component:cpu` gefiltert werden.
5. Aus dem Datenpunktwert extrahierte Informationen als Tag-Wert verwenden:
   - Definieren Sie einen Tag mit einem Makro als Tag-Wert (zum Beispiel `tag-name:{{ITEM.VALUE<N>}.regsub()}`).
   - In *Monitoring* > *Problems* enthalten Probleme den Tag-Wert, der in die aus dem Datenpunktwert extrahierten Daten aufgelöst wird.
6. Probleme in einer Protokolldatei identifizieren und separat schließen:
   - Definieren Sie einen Trigger-Tag für den Trigger des Log-Monitoring-Datenpunkts, der Werte aus dem Datenpunktwert mithilfe eines Makros extrahiert (zum Beispiel `service:{{ITEM.VALUE<N>}.regsub()}`).
   - Richten Sie in der Trigger-Konfiguration Ereigniskorrelation ein:
     - setzen Sie den *PROBLEM event generation mode* auf "Multiple";
     - setzen Sie *OK event closes* auf "All problems if tag values match";
     - setzen Sie den Tag für die Übereinstimmung.
   - Von dem Log-Monitoring-Datenpunkt-Trigger erzeugte Probleme erhalten den Trigger-Tag und werden einzeln geschlossen.
7. Benachrichtigungen filtern:
   - Definieren Sie Trigger-Tags (zum Beispiel `scope:security` für trigger1 und `scope:availability` für trigger2).
   - Verwenden Sie Tag-Filter in Aktion-Bedingungen, um Benachrichtigungen nur für die Ereignisse zu erhalten, die den Tag-Daten entsprechen.
8. Probleme in Benachrichtigungen identifizieren:
   - Definieren Sie Trigger-Tags.
   - Verwenden Sie das Makro {EVENT.TAGS} in der Problembenachrichtigung.
   - Die Problembenachrichtigung enthält die Trigger-Tags, wodurch leichter erkennbar ist, zu welcher Anwendung bzw. welchem Dienst die Benachrichtigung gehört.
9. Konfigurationsaufgaben durch Verwendung von Vorlage-Tags vereinfachen:
   - Definieren Sie einen Vorlage-Trigger-Tag.
   - Trigger, die aus diesem Vorlage-Trigger erstellt werden, erhalten dessen Tag.
10. Trigger mit Tags aus Low-Level-Discovery (LLD) erstellen:
    - Definieren Sie einen Trigger-Prototyp-Tag mit einem LLD-Makro im Tag-Namen oder -Wert (zum Beispiel `scope:{#FSNAME}`).
    - Trigger, die aus dem Trigger-Prototyp erstellt werden, erhalten dessen Tag.
11. Dienste mithilfe von Service-Tags zuordnen:
    - Definieren Sie Service-Tags.
    - Konfigurieren Sie Service-Aktionen für Dienste mit übereinstimmenden Tags.
    - Verwenden Sie außerdem Service-Tags, um einen Dienst für SLA-Berechnungen mit einer SLA zu verknüpfen.
12. Dienste mithilfe von Service-Problem-Tags mit Problemen verknüpfen:
    - Definieren Sie einen Problem-Tag in der Service-Konfiguration (zum Beispiel `target:mysql`).
    - Probleme mit einem übereinstimmenden Tag werden automatisch dem Dienst zugeordnet, und der Dienststatus ändert sich basierend auf den konfigurierten Regeln zur Berechnung des Dienststatus.
13. Probleme unterdrücken, wenn sich ein Host im Wartungsmodus befindet:
    - Definieren Sie Tags in der Konfiguration des Wartungszeitraums.
    - Probleme mit den definierten Tags werden unterdrückt.
14. Zugriff für Benutzergruppen gewähren:
    - Definieren Sie Tags in der Konfiguration der Benutzergruppe.
    - Benutzer in der Benutzergruppe können nur Probleme mit den definierten Tags anzeigen.

Konfiguration Tags können in einem eigenen Tab definiert werden, zum Beispiel in der Auslöserkonfiguration:



Makrounterstützung Integrierte Makros und Benutzermakros in Tags werden zum Zeitpunkt des Ereignisses aufgelöst. Bis das Ereignis eingetreten ist, werden diese Makros im Zabbix Frontend als nicht aufgelöst angezeigt.

Makros der Low-Level-Discovery werden während des Discovery-Prozesses aufgelöst.

Die folgenden Makros können in Trigger-Tag-Namen und -Werten verwendet werden:

- {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} und {HOST.ID} integrierte Makros
- {INVENTORY.*} integrierte Makros (zum Verweisen auf Host-Inventarwerte von einem oder mehreren Hosts in einem Trigger-Ausdruck)
- Benutzermakros und Benutzermakros mit Kontext (der Kontext kann Makros der Low-Level-Discovery enthalten)
- Makros der Low-Level-Discovery (nur in Trigger-Vorlagen-Tags)

Die folgenden Makros können in Tag-Namen und -Werten von Vorlage, Host und Datenpunkt/Web-Szenario verwendet werden:

- {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} und {HOST.ID} integrierte Makros
- {INVENTORY.*} integrierte Makros
- Benutzermakros
- Makros der Low-Level-Discovery (nur in Host- und Datenpunkt-Vorlagen-Tags)

Die folgenden Makros können in triggerbasierten Benachrichtigungen verwendet werden:

- {EVENT.TAGS} und {EVENT.RECOVERY.TAGS} integrierte Makros (diese Makros werden in eine durch Kommas getrennte Liste von Ereignis-Tags oder Wiederherstellungs-Ereignis-Tags aufgelöst)
- {EVENT.TAGSJSON} und {EVENT.RECOVERY.TAGSJSON} integrierte Makros (diese Makros werden in ein JSON-Array aufgelöst, das Ereignis-Tag-Objekte oder Wiederherstellungs-Ereignis-Tag-Objekte enthält)

Teilzeichenfolgenextraktion in Auslöser-Tags

Die Extraktion von Teilzeichenfolgen wird unterstützt, um den Tag-Namen oder den Tag-Wert mithilfe einer Makro-Funktion zu füllen. Die Funktion wendet einen regulären Ausdruck auf den Wert an, der vom unterstützten Makro zurückgegeben wird. Zum Beispiel:

```
{{ITEM.VALUE}.regsub(pattern, output)}
{{ITEM.VALUE}.iregsub(pattern, output)}

{{#LLDMACRO}.regsub(pattern, output)}
{{#LLDMACRO}.iregsub(pattern, output)}
```

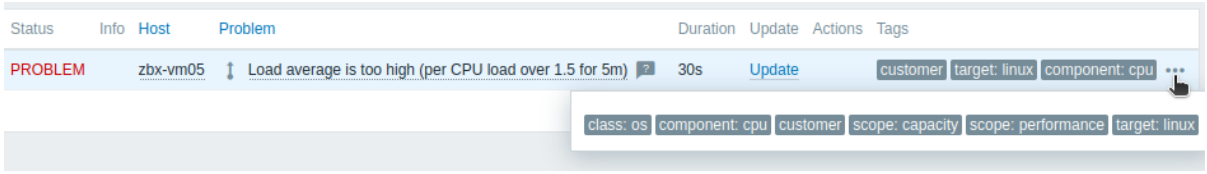
Wenn der Tag-Name oder -Wert nach der Makroauflösung mehr als 255 Zeichen umfasst, wird er auf 255 Zeichen gekürzt.

Siehe auch: Verwendung von Makro-Funktionen in Low-Level-Discovery-Makros für das Event-Tagging.

Anzeigen von Ereignis-Tags Tags, sofern definiert, können bei neuen Ereignissen angezeigt werden in:

- *Monitoring* > *Probleme*
- *Monitoring* > *Probleme* > *Ereignisdetails*
- *Dashboards* > *Probleme*-Widget

Die Reihenfolge und Anzahl der angezeigten Tags wird durch die Filteroptionen *Tag-Anzeigepriorität* und *Tags anzeigen* in *Monitoring > Probleme* oder im *Probleme*-Dashboard-Widget bestimmt. Beachten Sie, dass maximal drei Tags angezeigt werden können; wenn es mehr Tags gibt, werden beim Überfahren der drei Punkte alle Tags in einem Popup-Fenster angezeigt.



Complex event processing

Overview Complex event processing (CEP) performs rule-based processing for Zabbix events.

For example, you might want to:

- group related events and mark first event in a series as cause (the following as symptoms)
- specify a time-to-live period for events
- specify a maximum number for certain events
- be notified about similar events by location
- be notified if some event happens too often (or not at all) within a time period

In addition to single event processing, you may define **time windows** for event analysis. Within a time window, all matching events form a single context and can be analyzed as a whole.

A sliding time window moves forward with the time. Older events leave the window (are ”evicted”), while new events may enter.

A tumbling time window does not move forward. When the window expires, a new window with the same duration is started. *Cause and symptom grouping* windows are a tumbling time window.

The time window concept presents logical points for executing operations. For example, an event can be transformed or ”enriched” when it enters the window - severity can be raised or lowered, tags can be added or adjusted, suppression can be activated, etc.

Similar operations can be executed when the event leaves the window or when a pattern is found.

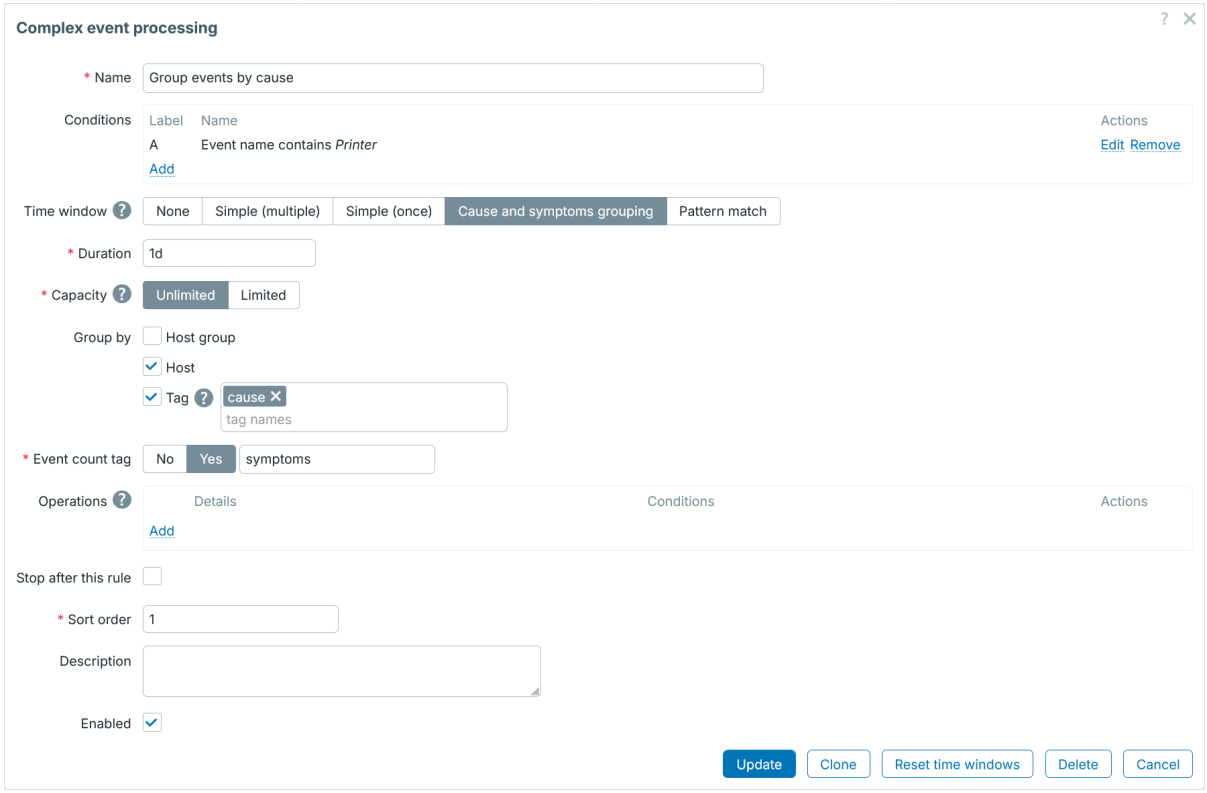
Event stream analysis is defined by CEP rules. A CEP rule consists of a time window (optional), conditions and operations. Operations are tied to an execution point.

Use cases

- Group related events, with the first such event automatically marked as cause (all following as symptoms).
- Enrich event with higher severity/specific tag if too many 401 events happen within a minute (possible bot activity).

Configuration To create a CEP rule in Zabbix frontend, do the following:

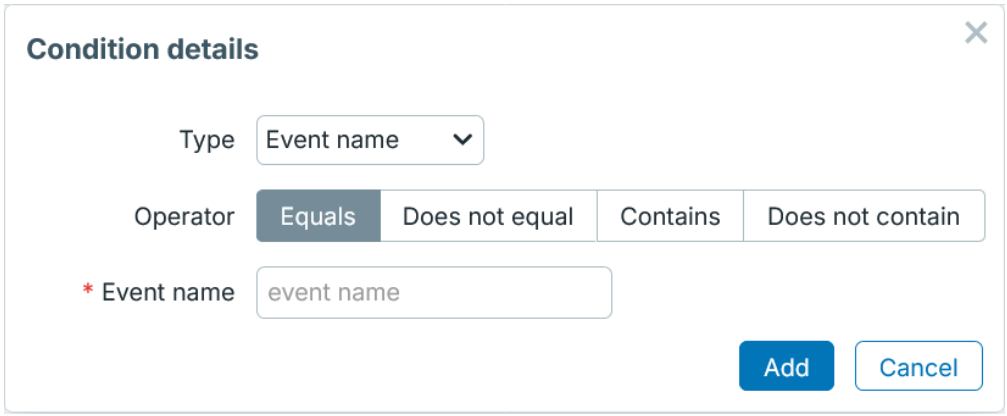
1. Go to *Data collection > Event processing*.
2. Click *Create complex event processing* in the upper-right corner of the screen.
3. Enter parameters of the rule in the form.



Parameter	Description
Name	Specify a unique CEP rule name.
Conditions	Specify conditions for associating events with this CEP rule.
Time window	Select the time window type: **None** - process single events with no time window (default) **Simple (multiple)** - process events in a sliding time window (event can be added to multiple windows of this type) **Simple (once)** - process events in a sliding time window (event can be added only to one window of this type) **Cause and symptom grouping** - process first event in time window as cause, consecutive events as symptoms **Pattern match** - analyze sliding time window events based on a JavaScript script. The window is examined every second; events older than the duration are evicted, and the script is then run over the events that remain in it. When the script reports a match, the pattern match operations are executed for every event the window holds; the events stay in the window and are examined again a second later, until they age out. A window can be forcibly closed by the ”Close window” operation.
Duration	Limit the time window duration (in seconds, 365 days max). The window duration is started by the first matching event. In *Simple* and *Pattern match* windows, the window is moving so older events will leave the window (are evicted) after the duration passes. In *Cause and symptom grouping* windows, window duration is a fixed timeframe for setting cause and symptom relationship. The window is started when the first event matches. When the duration expires, the window is closed and a new one is started. No event is evicted when the duration expires; periods run back to back. Time suffixes are supported (`10m`, `1h`, etc). User macros are supported.
Capacity	Specify the event capacity within the time window: **Unlimited** - event count is not limited **Limited** - event count is limited by the specified number Once the limit is surpassed, the latest event is evicted. User macros are supported.
Group by	Group events by host group, host, and/or event tag. This option is available for CEP rules with a time window. If specified, the rule will create multiple windows based on specified criteria; otherwise events will be grouped into a single window group.

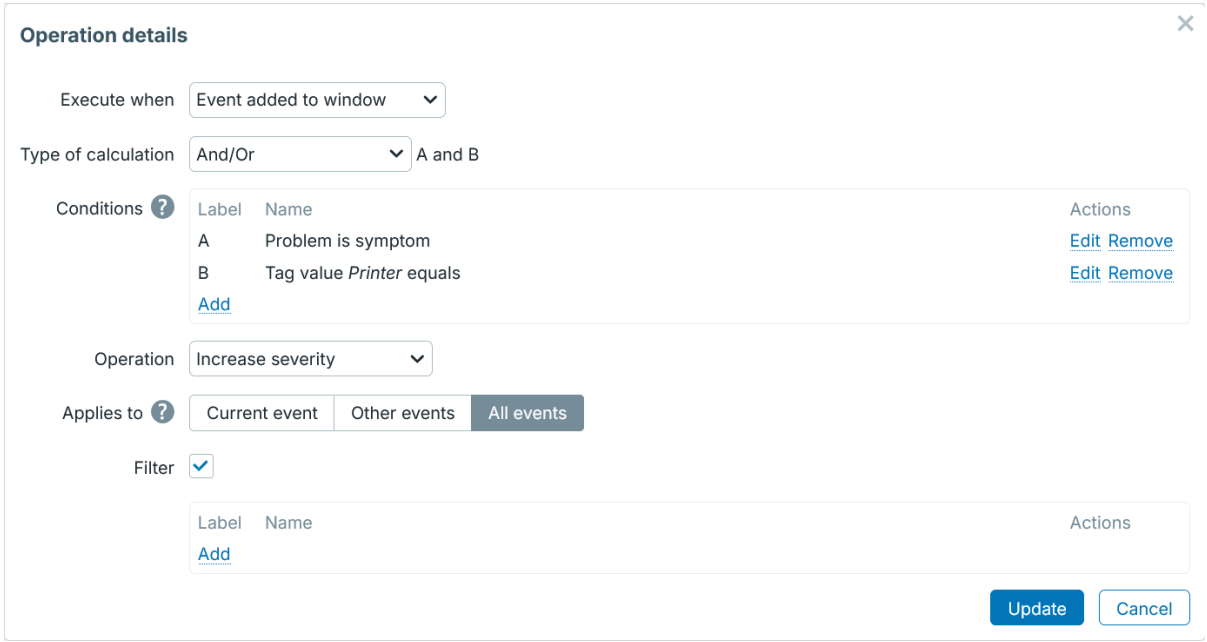
Parameter	Description
Event count tag	Specify name for a tag displaying symptom event count. Symptom event count will be displayed as `<tag name>:<event count>` (for example: `symptoms:2`). This field is only available if *Cause and symptom grouping* is selected as time window type.
Script	Specify the script for identifying event patterns. This field is only available if *Pattern match* is selected as time window type.
Operations	Click on *Add* to specify operations.
Stop after this rule	Mark the checkbox if no further processing is required after this rule. Otherwise, continue with the next CEP rule according to the sort order.
Sort order	Specify position of this CEP rule in the processing order. If the order priority is the same, the rules will be executed alphabetically.
Description	Add the CEP rule description.

CEP rule condition



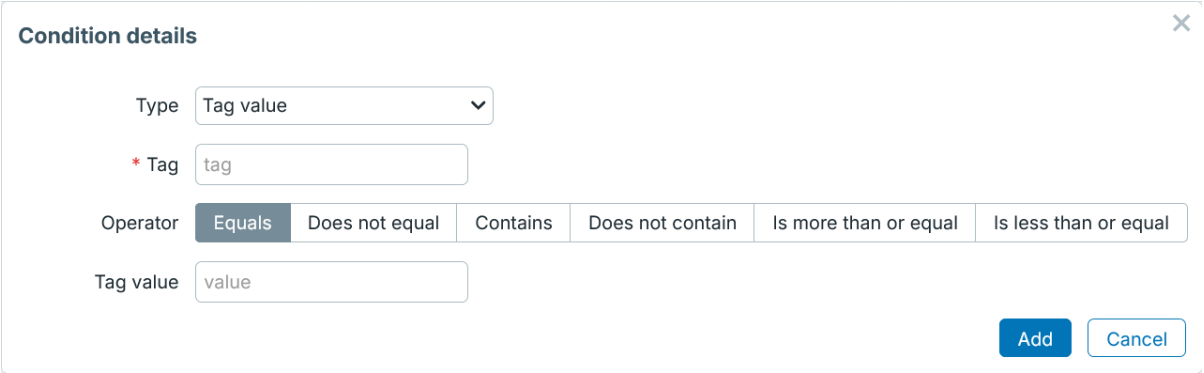
Parameter	Description
Type	Select condition type. Event filtering can be based on event name, tag name, tag value, severity, host group, host (visible name), or time period.

CEP rule operation



Parameter	Description
Execute when	Select execution point: **Event occurred** - when event occurs. This is the only stage where *Discard* operation is supported, since event is not actually created yet and can be discarded. Window-related operations are not available at this point. Changes made to the event in *Event occurred* phase (except closing) are visible to the next CEP rules trying to match this event. For example, if rule A added tag X and rule B has condition ”Tag X exists”, then it will match rule A and B. Of course, rule A should not be configured to stop after matching. **Event added to window** - when event is added to time window. Operations are executed after event is added to matched CEP-rule windows. If window is at its capacity, then *Event evicted* operations are executed instead. **Event evicted** - when event leaves the window. Operations are executed before event is evicted from a window because of duration or capacity. **Event closed** - when event is closed. **Window closed** - when time window is closed. Operations are executed on all window events when window is closed because of *Close window* operation or as expired cause-symptom window. **Pattern matched** - when pattern is matched. Operations are executed on all window events when the pattern match JavaScript returns `true`. Note that the *Event occurred*, *Event added to window* and *Event evicted* have fixed execution order, while *Window closed* and *Pattern matched* operate independently.
Conditions	Select additional conditions for the operation.
Operation	Select operation. **Events**: *Set name* - set event name *Close* - close event *Discard* - discard event. Only available at *Event occurred* stage. *Set severity* - set event severity *Increase severity* - increase event severity *Decrease severity* - decrease event severity *Suppress* - suppress event *Unsuppress* - unsuppress event *Clone first* - clone first event. Useful for producing a summary event based on the first event. Only available at *Event evicted*, *Window closed*, or *Pattern matched* stages. *Clone last* - clone last event. Useful for producing a summary event based on the last event. Only available at *Event evicted*, *Window closed*, or *Pattern matched* stages. For event cloning to work either the source (first/last) event must be resolved or its trigger must have multi-event generation enabled. **Tags**: *Add* - add tag *Set* - set tag and value *Set value* - set tag value *Increase value* - increase tag value *Decrease value* - decrease tag value *Rename* - rename tag *Remove* - remove tag **Window**: *Close (window)* - close event window
Applies to	Select operation scope: **Current event** - apply operation to the current event **Other events** - apply operation to all events except the current one **All events** - apply operation to all events
Filter	Select additional filter for the operation scope. This field is available if *Applies to* is set ”Other events” or ”All events”.

CEP rule operation condition



Parameter	Description
Type	Select condition type. Event filtering can be based on tag name, tag value, problem is opened/closed, problem is symptom/not symptom, first event in time window, last event in time window, problem is suppressed/not suppressed, event is cloned/not cloned.

Ereigniskorrelation

Übersicht

Die Ereigniskorrelation ermöglicht es, Problemergebnisse sehr präzise und flexibel mit ihrer Lösung zu korrelieren.

Die Ereigniskorrelation kann definiert werden:

- Auf Auslöser-Ebene - ein Auslöser kann verwendet werden, um separate Probleme mit ihrer Lösung zu verknüpfen;
- Global - Probleme können mithilfe globaler Korrelationsregeln mit ihrer Lösung aus einem anderen Auslöser/Abfragemethode korreliert werden.

Trigger-basierte Ereigniskorrelation

Übersicht

Die auslöserbasierte Ereigniskorrelation ermöglicht die Korrelation separater von einem Auslöser gemeldeter Probleme.

Im Allgemeinen kann ein OK-Ereignis alle von einem Auslöser erstellten Problemereignisse schließen. Es gibt jedoch Fälle, in denen ein detaillierterer Ansatz erforderlich ist. Wenn Sie beispielsweise Protokolldateien überwachen, möchten Sie möglicherweise bestimmte Probleme in einer Protokolldatei erkennen und einzeln statt alle zusammen schließen.

Dies ist bei Auslösern der Fall, deren Parameter *PROBLEM event generation mode* auf *Multiple* gesetzt ist. Solche Auslöser werden normalerweise zur Protokollüberwachung, Trap-Verarbeitung usw. verwendet.

In Zabbix ist es möglich, Problemereignisse anhand von Tags miteinander in Beziehung zu setzen. Tags werden verwendet, um Werte zu extrahieren und eine Identifikation für Problemereignisse zu erstellen. Dadurch können Probleme auch einzeln anhand übereinstimmender Tags geschlossen werden.

Mit anderen Worten: Derselbe Auslöser kann separate Ereignisse erstellen, die durch das Ereignis-Tag identifiziert werden. Problemereignisse können daher einzeln identifiziert und anhand der Identifikation durch das Ereignis-Tag separat geschlossen werden.

Funktionsweise

Bei der Protokollüberwachung können Ihnen Zeilen wie die folgenden begegnen:

```
Zeile1: Dienst 1 gestoppt
Zeile2: Dienst 2 gestoppt
Zeile3: Dienst 1 wurde neu gestartet
Zeile4: Dienst 2 wurde neu gestartet
```

Die Idee der Ereigniskorrelation besteht darin, das Problemereignis aus Zeile1 mit der Behebung aus Zeile3 und das Problemereignis aus Zeile2 mit der Behebung aus Zeile4 zu verknüpfen und diese Probleme nacheinander zu schließen:

```
Zeile1: Dienst 1 gestoppt
Zeile3: Dienst 1 wurde neu gestartet #Problem aus Zeile 1 geschlossen
```

```
Zeile2: Dienst 2 gestoppt
Zeile4: Dienst 2 wurde neu gestartet #Problem aus Zeile 2 geschlossen
```

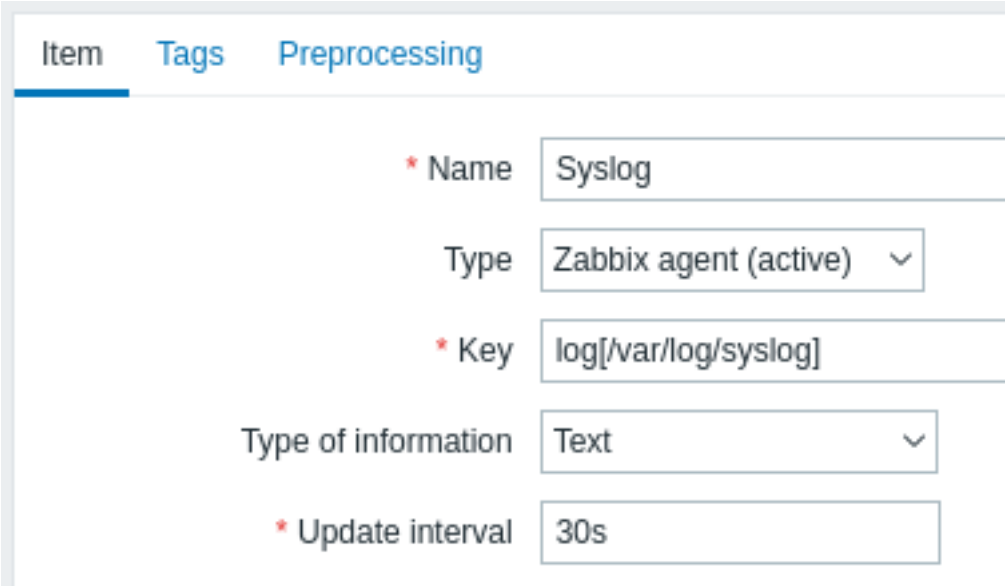
Dazu müssen Sie diese zusammengehörigen Ereignisse beispielsweise mit ”Dienst 1” und ”Dienst 2” kennzeichnen. Dies kann durch Anwenden eines regulären Ausdrucks auf die Protokollzeile erfolgen, um den Tag-Wert zu extrahieren. Wenn dann Ereignisse erstellt werden, werden sie entsprechend mit ”Dienst 1” und ”Dienst 2” getaggt, und das Problem kann der Behebung zugeordnet werden.

Konfiguration

Datenpunkt

Zunächst möchten Sie möglicherweise einen Datenpunkt einrichten, der eine Protokolldatei überwacht, zum Beispiel:

```
log[/var/log/syslog]
```



Nachdem der Datenpunkt eingerichtet ist, warten Sie eine Minute, bis die Konfigurationsänderungen übernommen wurden, und gehen Sie dann zu Latest data, um sicherzustellen, dass der Datenpunkt mit der Datenerfassung begonnen hat.

Auslöser

Wenn der Datenpunkt funktioniert, müssen Sie den Auslöser konfigurieren. Es ist wichtig zu entscheiden, welche Einträge in der Protokolldatei beachtet werden sollen. Das folgende Auslöserausdrucksbeispiel sucht beispielsweise nach einer Zeichenfolge wie ’Stopping’, um potenzielle Probleme zu signalisieren:

```
find(/My host/log[/var/log/syslog],,"regexp","Stopping")=1
```

Attention:
Damit jede Zeile, die die Zeichenfolge ”Stopping” enthält, als Problem betrachtet wird, setzen Sie außerdem den *Problem event generation mode* in der Auslöserkonfiguration auf ’Multiple’.

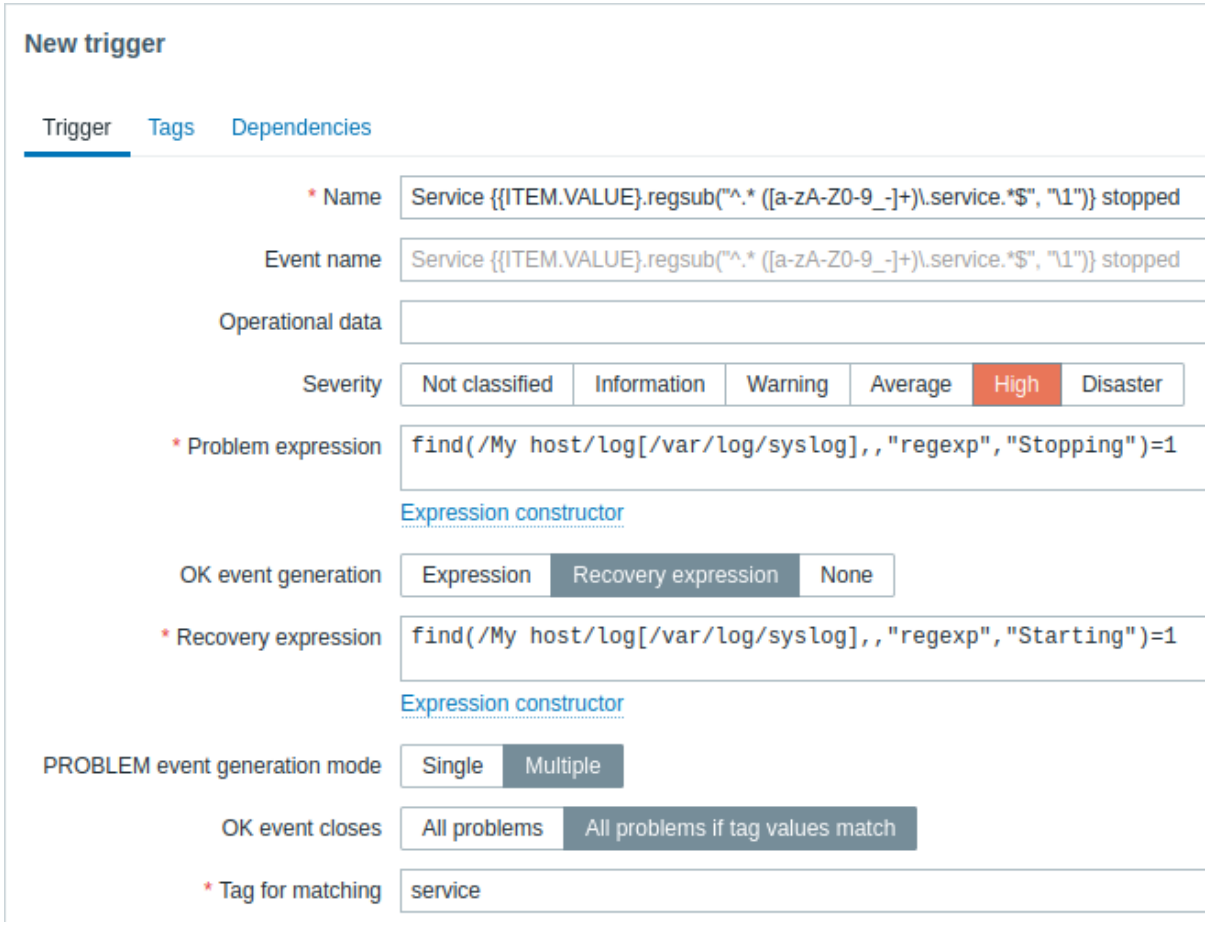
Definieren Sie dann einen Wiederherstellungsausdruck. Der folgende Wiederherstellungsausdruck löst alle Probleme auf, wenn eine Protokollzeile gefunden wird, die die Zeichenfolge ”Starting” enthält:

```
find(/My host/log[/var/log/syslog],,"regexp","Starting")=1
```

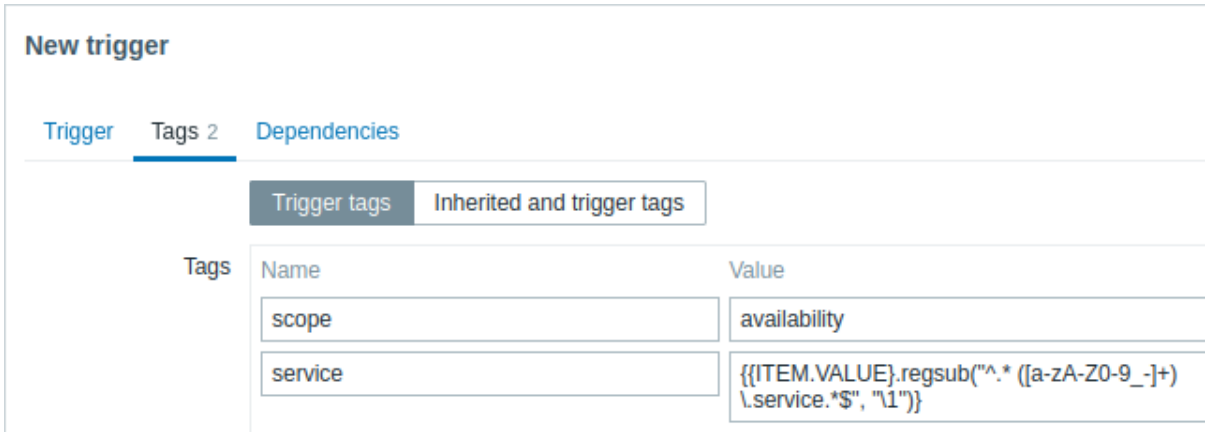
Da wir das nicht möchten, ist es wichtig, irgendwie sicherzustellen, dass die entsprechenden Root-Probleme geschlossen werden und nicht nur alle Probleme. Hier kann Tagging helfen.

Probleme und Wiederherstellungen können durch Angabe eines Tags in der Auslöserkonfiguration miteinander abgeglichen werden. Die folgenden Einstellungen müssen vorgenommen werden:

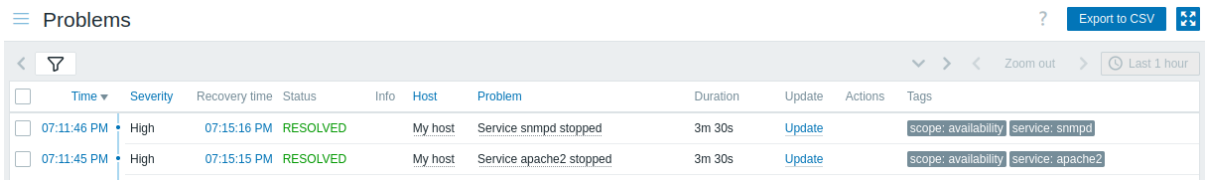
- *Problem event generation mode*: Multiple
- *OK event closes*: All problems if tag values match
- Geben Sie den Namen des Tags für den Ereignisabgleich ein



- konfigurieren Sie die Tags, um Tag-Werte aus Protokollzeilen zu extrahieren



Wenn die Konfiguration erfolgreich ist, können Sie unter *Monitoring* > *Problems* Problemereignisse sehen, die nach Anwendung getaggt und ihrer Wiederherstellung zugeordnet sind.



Warning:
Da Fehlkonfigurationen möglich sind und ähnliche Ereignis-Tags für **nicht zusammenhängende** Probleme erstellt werden können, prüfen Sie bitte die unten aufgeführten Fälle!

- Indizierte Makros beziehen sich immer auf das Feld *Expression* der Auslöserkonfiguration, nicht auf die *Recovery expression*. Zum Beispiel wird in einem Wiederherstellungsereignis **{ITEM.VALUE1}** zum neuesten Wert des ersten Datenpunkts **im Problem-Ausdruck** zum Zeitpunkt der Wiederherstellung aufgelöst. Wenn der Wiederherstellungsausdruck auf einem anderen Datenpunkt basiert und sich der Wert des Datenpunkts im Problem-Ausdruck bis zur Wiederherstellung ändert,

erhalten die Ereignisse unterschiedliche Tags und werden nicht korreliert.

- Wenn zwei Anwendungen Fehler- und Wiederherstellungsnachrichten in dieselbe Protokolldatei schreiben, kann ein Benutzer zwei *service*-Tags im selben Auslöser mit unterschiedlichen Tag-Werten verwenden, indem separate reguläre Ausdrücke in den Tag-Werten genutzt werden, um die Namen von beispielsweise Dienst A und Dienst B aus dem Makro {ITEM.VALUE} zu extrahieren (z. B. wenn sich die Nachrichtenformate unterscheiden). Dies funktioniert jedoch möglicherweise nicht wie geplant, wenn keine Übereinstimmung mit den regulären Ausdrücken vorliegt. Nicht übereinstimmende reguläre Ausdrücke führen zu leeren Tag-Werten, und ein einzelner leerer Tag-Wert sowohl in Problem- als auch in OK-Ereignissen reicht aus, um sie zu korrelieren. Daher kann eine Wiederherstellungsnachricht von Dienst A versehentlich eine Fehlermeldung von Dienst B schließen.
- Tatsächliche Tags und Tag-Werte werden erst sichtbar, wenn ein Auslöser auslöst. Wenn der verwendete reguläre Ausdruck ungültig ist, wird er stillschweigend durch eine Zeichenfolge *UNKNOWN* ersetzt. Wenn das erste Problemereignis mit einem *UNKNOWN*-Tag-Wert verpasst wird, können nachfolgende OK-Ereignisse mit demselben *UNKNOWN*-Tag-Wert erscheinen, die Problemereignisse schließen, die sie nicht hätten schließen sollen.
- Wenn ein Benutzer das Makro {ITEM.VALUE} ohne Makrofunktionen als Tag-Wert verwendet, gilt die Begrenzung auf 255 Zeichen. Wenn Protokollmeldungen lang sind und die ersten 255 Zeichen nicht spezifisch sind, kann dies ebenfalls zu ähnlichen Ereignis-Tags für nicht zusammenhängende Probleme führen.

Globale Ereigniskorrelation

Übersicht

Die globale Ereigniskorrelation ermöglicht es, alle von Zabbix überwachten Metriken einzubeziehen und Korrelationen zu erstellen.

Es ist möglich, Ereignisse zu korrelieren, die von völlig unterschiedlichen Auslösern erstellt wurden, und auf alle dieselben Aktionen anzuwenden. Durch das Erstellen intelligenter Korrelationsregeln können Sie sich tatsächlich Tausende sich wiederholender Benachrichtigungen ersparen und sich auf die Grundursachen eines Problems konzentrieren!

Die globale Ereigniskorrelation ist ein leistungsfähiger Mechanismus, mit dem Sie sich von einer auf einem einzelnen Auslöser basierenden Logik für Probleme und deren Behebung lösen können. Bisher wurde ein einzelnes Problemereignis von einem Auslöser erstellt, und wir waren für die Behebung dieses Problems von demselben Auslöser abhängig. Ein von einem Auslöser erstelltes Problem konnte nicht mit einem anderen Auslöser behoben werden. Mit einer auf Ereignis-Tags basierenden Ereigniskorrelation ist dies jedoch möglich.

Beispielsweise kann ein Protokoll-Auslöser Anwendungsprobleme melden, während ein Abfrage-Auslöser melden kann, dass die Anwendung ordnungsgemäß ausgeführt wird. Mithilfe von Ereignis-Tags können Sie den Protokoll-Auslöser mit *status:down* und den Abfrage-Auslöser mit *status:up* kennzeichnen. Anschließend können Sie diese Auslöser in einer globalen Korrelationsregel miteinander in Beziehung setzen und dieser Korrelation eine geeignete Aktion zuweisen, z. B. das Schließen der alten Ereignisse.

In einem anderen Anwendungsfall kann die globale Korrelation ähnliche Auslöser identifizieren und dieselbe Aktion auf sie anwenden. Was wäre, wenn wir pro Problem mit einem Netzwerkport nur einen einzigen Problembericht erhalten könnten? Es wäre nicht nötig, alle zu melden. Auch das ist mit der globalen Ereigniskorrelation möglich.

Die globale Ereigniskorrelation wird in **Korrelationsregeln** konfiguriert. Eine Korrelationsregel definiert, wie neue Problemereignisse mit vorhandenen Problemereignissen verknüpft werden und was im Fall einer Übereinstimmung geschehen soll (das neue Ereignis schließen oder übereinstimmende alte Ereignisse durch die Generierung entsprechender OK-Ereignisse schließen). Wenn ein Problem durch die globale Korrelation geschlossen wird, wird dies in der Spalte *Info* unter *Überwachung* > *Probleme* angezeigt.

Die Konfiguration globaler Korrelationsregeln ist nur Benutzern mit der Berechtigungsstufe „Super Admin“ möglich.

> **Attention:**
> Die Ereigniskorrelation muss sehr sorgfältig konfiguriert werden, da sie die Leistung der Ereignisverarbeitung negativ beeinflussen oder bei einer fehlerhaften Konfiguration mehr Ereignisse als beabsichtigt schließen kann (im schlimmsten Fall könnten sogar alle Problemereignisse geschlossen werden).

Um globale Korrelationsregeln **sicher** zu konfigurieren, beachten Sie die folgenden wichtigen Hinweise:

- Den Korrelationsbereich einschränken. Legen Sie immer ein eindeutiges Tag für das neue Ereignis fest, das mit alten Ereignissen verknüpft wird, und verwenden Sie die Korrelationsbedingung *Name des Tags des neuen Ereignisses*.
- Beim Verwenden von **Alte Ereignisse schließen** eine explizite Bedingung für alte Ereignisse hinzufügen. Fügen Sie bei der Auswahl von **Alte Ereignisse schließen** immer mindestens eine Bedingung für *Altes Ereignis* hinzu (z. B. *Name des Tags des alten Ereignisses*, *Wert des Tags des alten Ereignisses* oder *Ereignis-Tag-Paar*). Andernfalls kann die Regel übereinstimmen und nicht zugehörige vorhandene Problemereignisse schließen (im schlimmsten Fall alle Probleme). Verwenden

Sie für den Abgleich von Laufzeitwerten (Host:Port, Sitzungs-ID usw.) bevorzugt *Ereignis-Tag-Paar* und schränken Sie die Übereinstimmung nach Möglichkeit zusätzlich durch den Host oder die Hostgruppe ein.

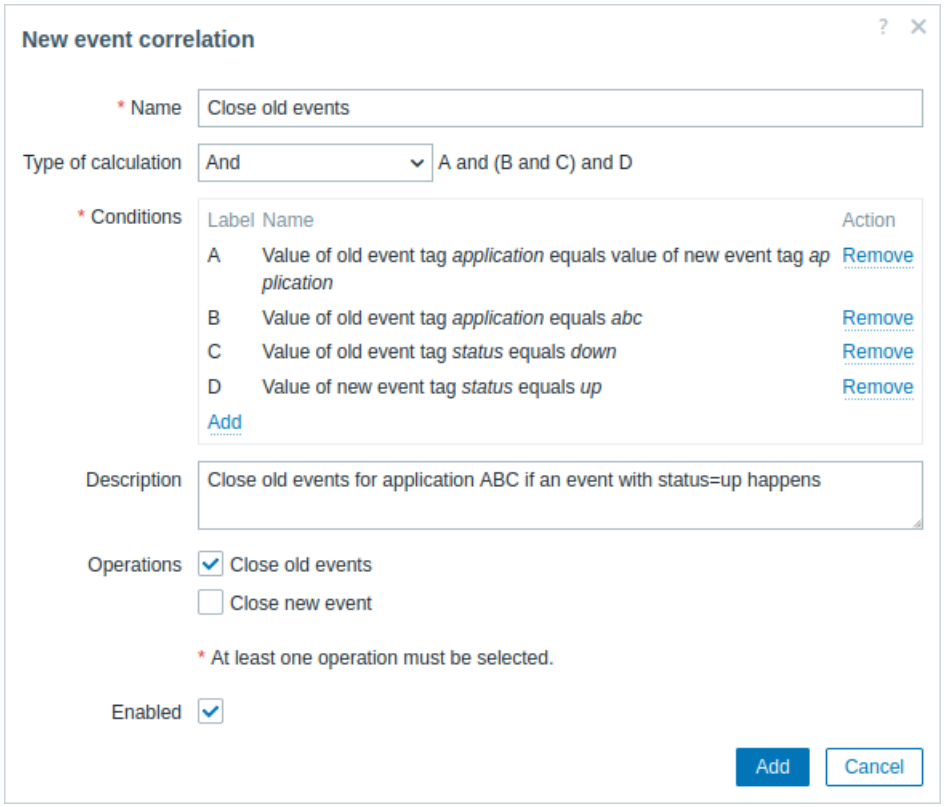
- Vermeiden Sie die Verwendung allgemeiner Tagnamen, die möglicherweise in verschiedenen Korrelationskonfigurationen verwendet werden.
- Beschränken Sie die Anzahl der Korrelationsregeln auf die Regeln, die Sie tatsächlich benötigen.

Siehe auch: bekannte Probleme.

Konfiguration

So konfigurieren Sie globale Regeln für die Ereigniskorrelation:

1. Gehen Sie zu *Data collection* > *Event correlation*.
2. Klicken Sie rechts auf *Create event correlation* (oder auf den Korrelationsnamen, um eine vorhandene Regel zu bearbeiten).
3. Geben Sie die Parameter der Korrelationsregel im Formular ein.

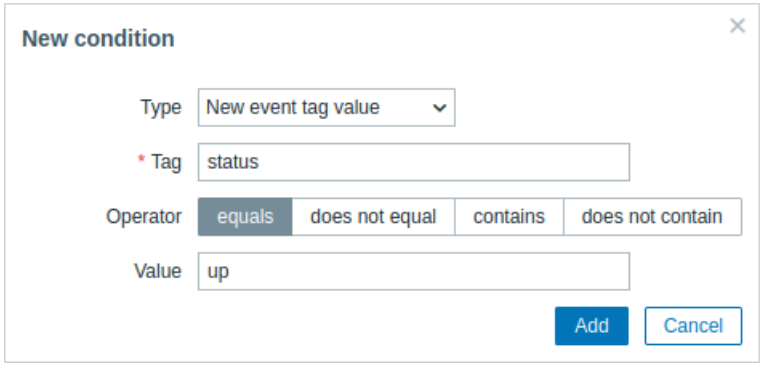


Alle Pflichtfelder sind mit einem roten Sternchen markiert.

Parameter	Beschreibung
Name	Eindeutiger Name der Korrelationsregel.
Type of calculation	Die folgenden Optionen zur Berechnung von Bedingungen sind verfügbar: **And** - alle Bedingungen müssen erfüllt sein **Or** - es reicht aus, wenn eine Bedingung erfüllt ist **And/Or** - AND bei unterschiedlichen Bedingungstypen und OR bei demselben Bedingungstyp **Custom expression** - eine benutzerdefinierte Berechnungsformel zur Auswertung von Aktionsbedingungen. Sie muss alle Bedingungen enthalten (dargestellt durch Großbuchstaben A, B, C, ...) und darf Leerzeichen, Tabulatoren, Klammern ( ), **and** (Groß-/Kleinschreibung beachten), **or** (Groß-/Kleinschreibung beachten), **not** (Groß-/Kleinschreibung beachten) enthalten.
Conditions	Liste der Bedingungen. Weitere Details zur Konfiguration einer Bedingung siehe unten.
Description	Beschreibung der Korrelationsregel.

Parameter	Beschreibung
Operations	Aktivieren Sie das Kontrollkästchen der Operation, die ausgeführt werden soll, wenn ein Ereignis korreliert wird. Die folgenden Operationen sind verfügbar: **Close old events** - schließt alte Ereignisse, wenn ein neues Ereignis auftritt. Fügen Sie bei Verwendung der Operation *Close old events* immer eine Bedingung auf Basis des alten Ereignisses hinzu, da sonst alle vorhandenen Probleme geschlossen werden könnten. **Close new event** - schließt das neue Ereignis, wenn es auftritt. **Warnung!** Lassen Sie bei Verwendung von *Close old events*/*Close new event* die Bedingungen für alte/neue Ereignisse nicht leer. Wenn Sie die Operation *Close old events* ohne eine Bedingung auswählen, die dem alten Ereignis entspricht, kann Zabbix alle vorhandenen alten Ereignisse zuordnen und schließen. Fügen Sie bei Verwendung von *Close old events* immer eine explizite Bedingung für das alte Ereignis hinzu (z. B. *Old event tag name* oder *Event tag pair*). Eine Regel, die nur eine *New event condition* und die Operation *Close old events* verwendet, ordnet beispielsweise alle alten Ereignisse zu, die die (fehlenden) Kriterien für alte Ereignisse erfüllen - und schließt damit effektiv alte Probleme.
Enabled	Wenn Sie dieses Kontrollkästchen aktivieren, wird die Korrelationsregel aktiviert.

Um Details zu einer neuen Bedingung zu konfigurieren, klicken Sie im Block Conditions auf Add. Es öffnet sich ein Popup-Fenster, in dem Sie die Details der Bedingung bearbeiten können.



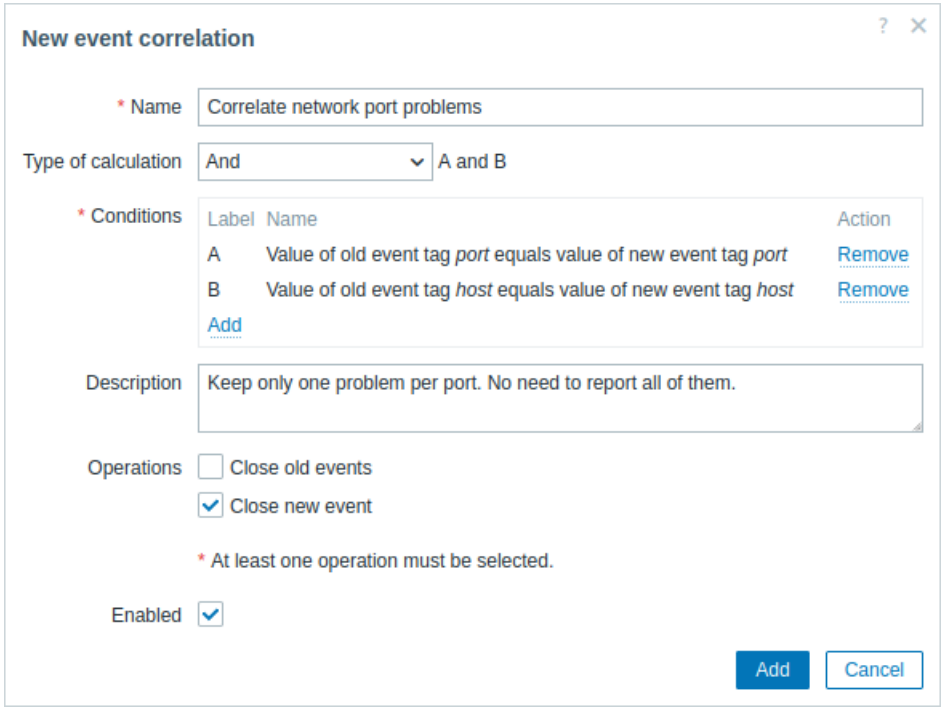
Parameter	Beschreibung
New condition	Wählen Sie eine Bedingung für die Korrelation von Ereignissen aus. *Hinweis*: Wenn keine Bedingung für das alte Ereignis angegeben ist, können alle alten Ereignisse zugeordnet und geschlossen werden. Ebenso können alle neuen Ereignisse zugeordnet und geschlossen werden, wenn keine Bedingung für das neue Ereignis angegeben ist. Die folgenden Bedingungen sind verfügbar: **Old event tag name** - geben Sie den Tag-Namen des alten Ereignisses für die Zuordnung an. **New event tag name** - geben Sie den Tag-Namen des neuen Ereignisses für die Zuordnung an. **New event host group** - geben Sie die Hostgruppe des neuen Ereignisses für die Zuordnung an. **Event tag pair** - geben Sie den Tag-Namen des neuen Ereignisses und den Tag-Namen des alten Ereignisses für die Zuordnung an. In diesem Fall liegt eine Übereinstimmung vor, wenn die **Werte** der Tags in beiden Ereignissen übereinstimmen. Die Tag-*Namen* müssen nicht übereinstimmen. Diese Option ist nützlich für die Zuordnung von Laufzeitwerten, die zum Zeitpunkt der Konfiguration möglicherweise noch nicht bekannt sind (siehe auch Beispiel). **Old event tag value** - geben Sie den Tag-Namen und den Wert des alten Ereignisses für die Zuordnung an, unter Verwendung der folgenden Operatoren: *equals* - hat den Tag-Wert des alten Ereignisses *does not equal* - hat den Tag-Wert des alten Ereignisses nicht *contains* - enthält die Zeichenfolge im Tag-Wert des alten Ereignisses *does not contain* - enthält die Zeichenfolge im Tag-Wert des alten Ereignisses nicht **New event tag value** - geben Sie den Tag-Namen und den Wert des neuen Ereignisses für die Zuordnung an, unter Verwendung der folgenden Operatoren: *equals* - hat den Tag-Wert des neuen Ereignisses *does not equal* - hat den Tag-Wert des neuen Ereignisses nicht *contains* - enthält die Zeichenfolge im Tag-Wert des neuen Ereignisses *does not contain* - enthält die Zeichenfolge im Tag-Wert des neuen Ereignisses nicht

Warning:
Da Fehlkonfigurationen möglich sind, prüfen Sie bitte die unten beschriebenen Fälle, wenn ähnliche Ereignis-Tags für **nicht zusammenhängende** Probleme erstellt werden können!

- Tatsächliche Tags und Tag-Werte werden erst sichtbar, wenn ein Auslöser auslöst. Wenn der verwendete reguläre Ausdruck ungültig ist, wird er stillschweigend durch eine *UNKNOWN*-Zeichenfolge ersetzt. Wenn das erste Problemereignis mit einem *UNKNOWN*-Tag-Wert übersehen wird, können anschließend OK-Ereignisse mit demselben *UNKNOWN*-Tag-Wert auftreten, die Problemereignisse schließen, die sie nicht hätten schließen sollen.
- Wenn ein Benutzer das Makro {ITEM.VALUE} ohne Makrofunktionen als Tag-Wert verwendet, gilt die Begrenzung auf 255 Zeichen. Wenn Protokollmeldungen lang sind und die ersten 255 Zeichen nicht spezifisch sind, kann dies ebenfalls zu ähnlichen Ereignis-Tags für nicht zusammenhängende Probleme führen.

Beispiel

Verhindern Sie wiederholte Problemereignisse vom selben Netzwerkport.



Diese globale Korrelationsregel korreliert Probleme, wenn auf dem Auslöser die Tag-Werte *host* und *port* vorhanden sind und sie im ursprünglichen Ereignis und im neuen Ereignis identisch sind.

Die Operation schließt neue Problemereignisse am selben Netzwerkport und lässt nur das ursprüngliche Problem offen.

Erkennung

Übersicht Die Discovery bietet automatische Möglichkeiten, Hosts hinzuzufügen und das Monitoring ohne manuelle Einrichtung zu konfigurieren. Zabbix unterstützt mehrere Discovery-Methoden, um sich an unterschiedliche Szenarien anzupassen.

Erkennungsmethoden

- Netzwerkerkennung - automatisch Hosts und Dienste erkennen, die in Ihrem Netzwerk verfügbar sind.
- Aktive Agent-Autoregistrierung - aktiven Agents erlauben, sich automatisch beim Server zu registrieren.
- Low-Level-Erkennung - automatisch Datenpunkte, Auslöser, Diagramme und Hosts für erkannte Entitäten wie Dateisysteme oder Netzwerkschnittstellen mithilfe von Prototypen und verschachtelten Erkennungsregeln erstellen.

Netzwerkerkennung

Überblick

Zabbix bietet eine automatische Netzwerkerkennung, die effektiv und sehr flexibel ist.

Bei korrekt eingerichteter Netzwerkerkennung können Sie:

- die Bereitstellung von Zabbix beschleunigen.
- die Administration vereinfachen.
- Zabbix in sich schnell ändernden Umgebungen ohne übermäßigen Verwaltungsaufwand einsetzen.

Die Zabbix-Netzwerkerkennung basiert auf den folgenden Informationen:

- IP-Bereiche
- Verfügbarkeit externer Dienste (FTP, SSH, WEB, POP3, IMAP, TCP usw.)
- Informationen, die vom Zabbix Agent empfangen werden (nur unverschlüsselter Modus wird unterstützt)
- Informationen, die vom SNMP-Agent empfangen werden

Sie bietet KEINE Erkennung der Netzwerktopologie.

Die Netzwerkerkennung besteht im Wesentlichen aus zwei Phasen: Erkennung und Aktionen.

Discovery

Zabbix durchsucht regelmäßig die in den Netzwerk-Discovery-Regeln definierten IP-Bereiche. Die Häufigkeit der Prüfung ist für jede Regel individuell konfigurierbar.

Jede Regel verfügt über einen Satz definierter Dienstprüfungen, die für den IP-Bereich ausgeführt werden.

Discovery-Regeln werden vom Discovery-Manager verarbeitet. Der Discovery-Manager erstellt für jede Regel einen Job mit einer Liste von Aufgaben (Netzwerkprüfungen). Netzwerkprüfungen werden parallel von den verfügbaren Discovery-Workern ausgeführt (die Anzahl ist im Frontend pro Regel konfigurierbar). Die Ausnahme sind SNMPv3-Prüfungen, die von einem einzelnen Worker verarbeitet werden. Nur Prüfungen mit derselben IP und demselben Port werden sequenziell geplant, da einige Geräte keine parallelen Verbindungen über denselben Port akzeptieren.

Die Größe der Warteschlange für Netzwerkprüfungen ist auf 2000000 oder ungefähr 4 GB Speicher begrenzt. Wenn die Warteschlange voll wird, wird die Discovery-Regel übersprungen und eine Warnmeldung im Log ausgegeben. Sie können das interne Datenpunkt `zabbix[discovery_queue]` verwenden, um die Anzahl der Discovery-Prüfungen in der Warteschlange zu überwachen.

Discovery-Prüfungen werden unabhängig von den anderen Prüfungen verarbeitet. Wenn bei einer Prüfung kein Dienst gefunden wird (oder sie fehlschlägt), werden die anderen Prüfungen dennoch verarbeitet.

Note:
Wenn eine Discovery-Regel während der Ausführung geändert wird, wird die aktuelle Discovery-Ausführung abgebrochen.

Jede Prüfung eines Dienstes und eines Hosts (IP), die vom Netzwerk-Discovery-Modul durchgeführt wird, erzeugt ein Discovery-Ereignis.

Ereignis	Ergebnis der Dienstprüfung
Dienst entdeckt	Der Dienst ist 'up', nachdem er 'down' war, oder wurde zum ersten Mal entdeckt.
Dienst up	Der Dienst ist 'up', nachdem er bereits 'up' war.
Dienst verloren	Der Dienst ist 'down', nachdem er 'up' war.
Dienst down	Der Dienst ist 'down', nachdem er bereits 'down' war.
Host entdeckt	Mindestens ein Dienst eines Hosts ist 'up', nachdem alle Dienste dieses Hosts 'down' waren, oder ein Dienst wird entdeckt, der zu einem nicht registrierten Host gehört.
Host up	Mindestens ein Dienst eines Hosts ist 'up', nachdem mindestens ein Dienst bereits 'up' war.
Host verloren	Alle Dienste eines Hosts sind 'down', nachdem mindestens einer 'up' war.
Host down	Alle Dienste eines Hosts sind 'down', nachdem sie bereits 'down' waren.

Aktionen

Discovery-Ereignisse können die Grundlage für relevante Aktionen sein, wie zum Beispiel:

- Benachrichtigungen senden
- Hosts hinzufügen/entfernen
- Hosts aktivieren/deaktivieren
- Hosts zu einer Gruppe hinzufügen
- Hosts aus einer Gruppe entfernen
- Tags zu einem Host hinzufügen
- Tags von einem Host entfernen

- Eine Vorlage mit Hosts verknüpfen/eine Vorlage von Hosts trennen
- Remote-Skripte ausführen

Diese Aktionen können in Bezug auf Gerätetyp, IP, Status, Uptime/Downtime usw. konfiguriert werden. Vollständige Details zur Konfiguration von Aktionen für auf Netzwerk-Discovery basierende Ereignisse finden Sie auf den Seiten zu operation und conditions von Aktionen.

Da Aktionen der Netzwerk-Discovery ereignisbasiert sind, werden sie sowohl ausgelöst, wenn ein entdeckter Host online ist, als auch wenn er offline ist. Es wird dringend empfohlen, eine Aktions-condition *Discovery status: up* hinzuzufügen, um zu vermeiden, dass Aktionen wie *Add host* bei Ereignissen vom Typ *Service Lost*/*Service Down* ausgelöst werden. Andernfalls erzeugt ein entdeckter Host, wenn er manuell entfernt wird, weiterhin *Service Lost*/*Service Down*-Ereignisse und wird während des nächsten Discovery-Zyklus erneut erstellt.

Note:
Das Verknüpfen von Vorlagen mit einem entdeckten Host schlägt insgesamt fehl, wenn eine der verknüpfbaren Vorlagen eine eindeutige Entität (z. B. einen Datenpunkt-Schlüssel) besitzt, die mit einer eindeutigen Entität (z. B. einem Datenpunkt-Schlüssel) identisch ist, die bereits auf dem Host oder auf einer anderen der verknüpfbaren Vorlagen vorhanden ist.

Host-Erstellung

Ein Host wird hinzugefügt, wenn die Operation *Host hinzufügen* ausgewählt ist. Ein Host wird auch dann hinzugefügt, wenn die Operation *Host hinzufügen* fehlt, sofern Sie Operationen auswählen, die Aktionen auf einem Host auslösen. Solche Operationen sind:

- Host aktivieren.
- Host deaktivieren.
- Host zu einer Hostgruppe hinzufügen.
- Vorlage mit einem Host verknüpfen.

Erstellte Hosts werden der Gruppe *Entdeckte Hosts* hinzugefügt (standardmäßig, konfigurierbar unter *Administration* > *Allgemein* > *Andere*). Wenn Sie möchten, dass Hosts einer anderen Gruppe hinzugefügt werden, fügen Sie eine Operation *Aus Hostgruppen entfernen* hinzu (mit der Angabe ”Entdeckte Hosts”) und fügen Sie außerdem eine Operation *Zu Hostgruppen hinzufügen* hinzu (mit der Angabe einer anderen Hostgruppe), da ein Host zu einer Hostgruppe gehören muss.

Die IP-Adresse des entdeckten Geräts wird zusammen mit der Entdeckungsquelle (Zabbix Server, Zabbix Proxy oder Proxy-Gruppe) und dem Schnittstellentyp als Kriterium verwendet, um einen Host im System zu finden. Wenn bereits ein Host mit derselben IP-Adresse, demselben Schnittstellentyp und derselben Entdeckungsquelle vorhanden ist, wird dieser Host als Ziel für die Ausführung der Operationen verwendet. Wenn sich die Entdeckungsquelle unterscheidet, wird das entdeckte Objekt als anderer Host behandelt, und es kann ein neuer Host erstellt werden.

Wenn die IP-Adresse des entdeckten Hosts geändert wird oder die Schnittstelle gelöscht wird, wird bei der nächsten Entdeckung ein neuer Host erstellt.

Benennung von Hosts

Beim Hinzufügen von Hosts ist ein Host-Name das Ergebnis einer Reverse-DNS-Abfrage oder die IP-Adresse, falls die Reverse-Abfrage fehlschlägt. Die Abfrage wird vom Zabbix Server oder Zabbix Proxy durchgeführt, je nachdem, welcher die Discovery ausführt. Wenn die Abfrage auf dem Proxy fehlschlägt, wird sie nicht auf dem Server erneut versucht. Wenn bereits ein Host mit einem solchen Namen existiert, wird an den nächsten Host **_2** an den Namen angehängt, dann **_3** und so weiter.

Es ist auch möglich, die DNS-/IP-Abfrage zu überschreiben und stattdessen einen Datenpunktwert für den Host-Namen zu verwenden, zum Beispiel:

- Sie können mehrere Server mit laufendem Zabbix Agent mithilfe eines Zabbix-Agent-Datenpunkts für die Discovery erkennen und ihnen automatisch passende Namen zuweisen, basierend auf dem von diesem Datenpunkt zurückgegebenen Zeichenfolgenwert
- Sie können mehrere SNMP-Netzwerkgeräte mithilfe eines SNMP-Agent-Datenpunkts für die Discovery erkennen und ihnen automatisch passende Namen zuweisen, basierend auf dem von diesem Datenpunkt zurückgegebenen Zeichenfolgenwert

Wenn der Host-Name mithilfe eines Datenpunktwerts festgelegt wurde, wird er bei den folgenden Discovery-Prüfungen nicht aktualisiert. Wenn es nicht möglich ist, den Host-Namen mithilfe eines Datenpunktwerts festzulegen, wird der Standardwert (DNS-Name) verwendet.

Wenn bereits ein Host mit der erkannten IP-Adresse existiert und sich die Discovery-Quelle (Zabbix Server, Proxy oder Proxy-Gruppe) nicht geändert hat, wird kein neuer Host erstellt. Wenn sich die Discovery-Quelle unterscheidet, wird die erkannte Entität als eigenständig behandelt und es kann ein neuer Host erstellt werden. Wenn die Discovery-Aktion jedoch Operationen enthält (Vorlage verknüpfen, zu Host-Gruppe hinzufügen usw.), werden diese auf dem vorhandenen Host ausgeführt, der anhand von IP-Adresse, Schnittstellentyp und Discovery-Quelle übereinstimmt.

Entfernen von Hosts

Hosts, die durch eine Netzwerkerkennungsregel erkannt wurden, werden automatisch aus *Monitoring* > *Discovery* entfernt, wenn sich eine erkannte Entität nicht mehr im IP-Bereich der Regel befindet. Hosts werden sofort entfernt.

Interface-Erstellung beim Hinzufügen von Hosts

Wenn Hosts infolge einer Netzwerkerkennung hinzugefügt werden, werden für sie gemäß diesen Regeln Interfaces erstellt:

- Die erkannten Dienste - zum Beispiel, wenn eine SNMP-Prüfung erfolgreich war, wird ein SNMP-Interface erstellt.
- Wenn ein Host sowohl auf Zabbix-Agent- als auch auf SNMP-Anfragen reagiert hat, werden beide Interface-Typen erstellt.
- Wenn die Eindeutigkeitskriterien Zabbix-Agent- oder SNMP-zurückgegebene Daten sind, wird das erste für einen Host gefundene Interface als Standard-Interface erstellt. Andere IP-Adressen werden ignoriert, es sei denn, das Standard-Interface fällt aus. Falls das Standard-Interface ausgefallen ist, wird die kleinste der anderen IP-Adressen als Ersatz für das Standard-Interface verwendet. Bedingungen der Aktion (z. B. Host IP) haben keinen Einfluss auf das Hinzufügen von Interfaces. *Hinweis*: Dies funktioniert, wenn alle Interfaces durch dieselbe Erkennungsregel entdeckt werden. Wenn eine andere Erkennungsregel ein anderes Interface desselben Hosts entdeckt, wird ein zusätzlicher Host hinzugefügt.
- Wenn ein Host nur auf Agent-Prüfungen reagiert hat, wird er nur mit einem Agent-Interface erstellt. Wenn er später auf SNMP reagieren sollte, würden zusätzliche SNMP-Interfaces hinzugefügt.
- Wenn ursprünglich 3 separate Hosts erstellt wurden, die anhand der Eindeutigkeitskriterien ”IP” entdeckt wurden, und die Erkennungsregel dann so geändert wird, dass die Hosts A, B und C identische Ergebnisse für die Eindeutigkeitskriterien haben, werden B und C als zusätzliche Interfaces für A, den ersten Host, erstellt. Die einzelnen Hosts B und C bleiben bestehen. Unter *Monitoring* > *Erkennung* werden die hinzugefügten Interfaces in der Spalte ”Erkanntes Gerät” in schwarzer Schrift und eingerückt angezeigt, während die Spalte ”Überwachter Host” nur A, den zuerst erstellten Host, anzeigt. ”Uptime/Downtime” wird für IPs, die als zusätzliche Interfaces betrachtet werden, nicht gemessen.

Ändern der Proxy-Einstellung

Die von verschiedenen Proxys erkannten Hosts werden nicht immer als unterschiedliche Hosts behandelt. Erkennung und Eindeutigkeitsprüfungen hängen von der Struktur der Proxy-Gruppe ab: Wenn ein Proxy eine Erkennungsregel ausführt und einen Host erstellt, wird dieser Host der übergeordneten Proxy-Gruppe des Proxys hinzugefügt und nicht dem Proxy selbst zugewiesen. Wenn Zabbix während der Erkennung die IP-Eindeutigkeit bewertet, prüft es Hosts, die von der übergeordneten Proxy-Gruppe überwacht werden. Hosts, die von einzelnen Proxys innerhalb dieser Gruppe überwacht werden, einschließlich des Proxys, der die Erkennung ausgeführt hat, werden bei der Eindeutigkeitsprüfung ignoriert, was zu doppelten Hosts führen kann, wenn mehrere Proxys überlappende Subnetze überwachen.

Während dieses Verhalten es ermöglicht, dass die Erkennung über überlappende IP-Bereiche hinweg funktioniert, die von verschiedenen Subnetzen verwendet werden, ist das Ändern des einem bereits überwachten Subnetz zugewiesenen Proxys komplizierter, da Proxy-Änderungen konsistent auf erkannte Hosts und auf die Mitgliedschaft der übergeordneten Proxy-Gruppe angewendet werden müssen, um Duplikate zu vermeiden.

Beispiel: Schritte zum Ersetzen des Proxys in einer Erkennungsregel:

1. Erkennungsregel deaktivieren.
2. Proxy-Konfiguration synchronisieren.
3. Proxy in der Erkennungsregel ersetzen.
4. Proxy für alle von dieser Regel erkannten Hosts ersetzen (stellen Sie sicher, dass Hosts in der übergeordneten Proxy-Gruppe und alle von einzelnen Proxys in dieser Gruppe überwachten Hosts aktualisiert werden, um Duplikate zu vermeiden).
5. Erkennungsregel aktivieren.

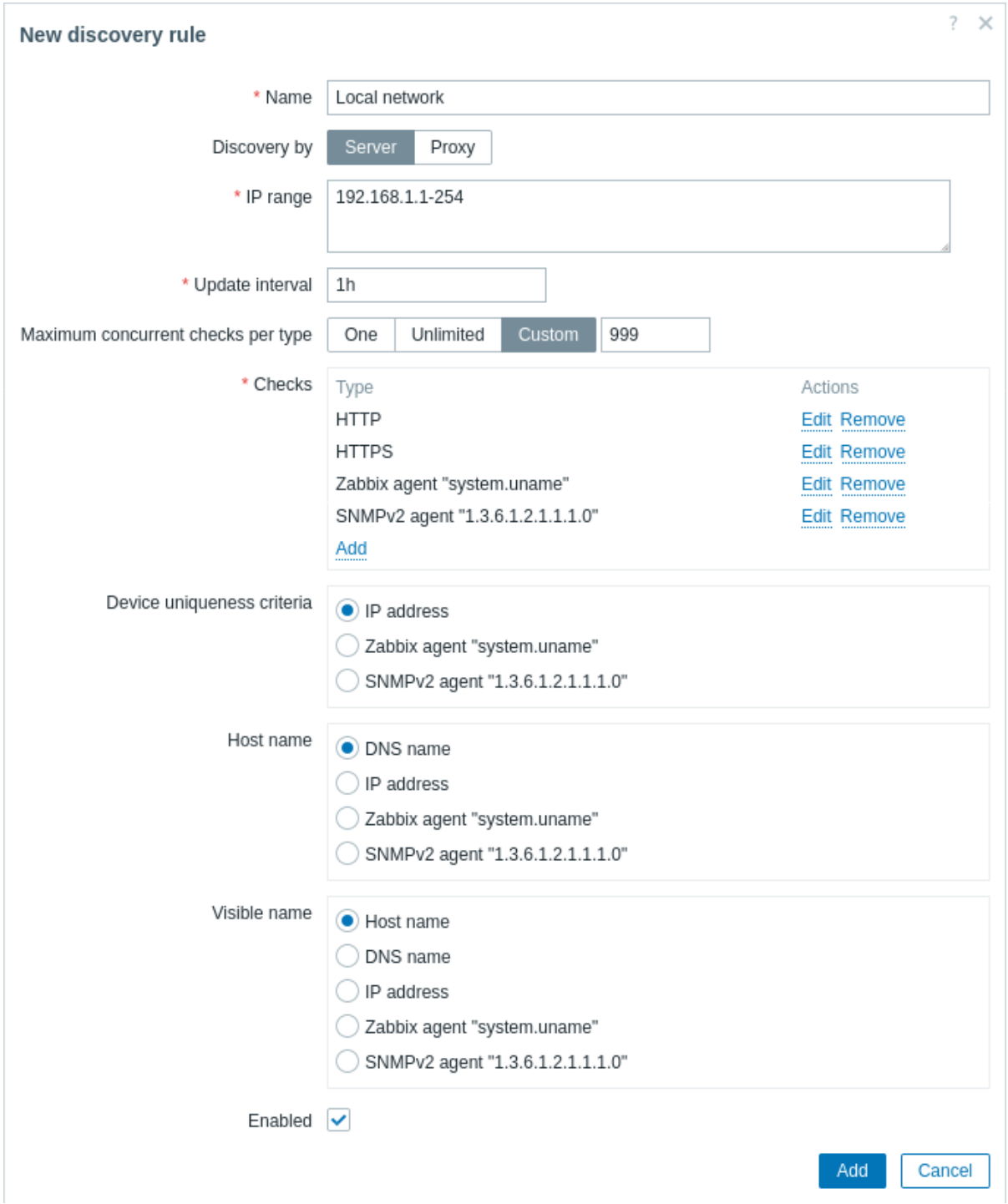
Konfigurieren einer Netzwerkerkennungsregel

Übersicht

Um eine von Zabbix verwendete Netzwerkerkennungsregel zum Erkennen von Hosts und Diensten zu konfigurieren:

1. Gehen Sie zu *Data collection* > *Discovery*.
2. Klicken Sie auf *Create discovery rule* (oder auf den Namen der Regel, um eine vorhandene zu bearbeiten).
3. Bearbeiten Sie die Attribute der Erkennungsregel.

Regelattribute



Alle obligatorischen Eingabefelder sind mit einem roten Sternchen markiert.

Parameter	Beschreibung
Name	Eindeutiger Name der Regel. Zum Beispiel ”Local network”.
Discovery by	Die Erkennung wird durchgeführt durch: **Server** - durch den Zabbix Server **Proxy** - durch den Zabbix Proxy (ausgewählt im Feld für den Proxy-Namen)
IP range	Der Bereich der IP-Adressen für die Erkennung. Er kann folgende Formate haben: Einzelne IP: 192.168.1.33 Bereich von IP-Adressen: 192.168.1-10.1-255. Der Bereich ist durch die Gesamtzahl der abgedeckten Adressen begrenzt (weniger als 64K). IP-Maske: 192.168.4.0/24 unterstützte IP-Masken: /16 - /30 für IPv4-Adressen /112 - /128 für IPv6-Adressen Liste: 192.168.1.1-255, 192.168.2.1-100, 192.168.2.200, 192.168.4.0/24 Dieses Feld unterstützt Leerzeichen, Tabulatoren und mehrere Zeilen.

Parameter	Beschreibung
Update interval	Dieser Parameter definiert, wie oft Zabbix die Regel ausführt. Das Intervall wird nach dem Ende der vorherigen Erkennungsinstanz gemessen, sodass es keine Überschneidung gibt. Zeitsuffixe werden unterstützt, z. B. 30s, 1m, 2h, 1d. Benutzer-Makros werden unterstützt. *Hinweis*: Wenn ein Benutzer-Makro verwendet wird und sein Wert geändert wird (z. B. 1w → 1h), wird die nächste Prüfung gemäß dem vorherigen Wert ausgeführt (im Beispiel also weit in der Zukunft).
Maximum concurrent checks per type	Legen Sie die maximale Anzahl von Erkennungs-Threads (Worker) pro Dienstprüfung für die parallele Verarbeitung von Erkennungsprüfungen fest: **One** - ein Thread **Unlimited** - unbegrenzte Anzahl von Threads (jedoch nicht mehr als im Parameter StartDiscoverers) **Custom** - eine benutzerdefinierte Anzahl von Threads festlegen (0-999). Asynchrone SNMPv1- und SNMPv2-Dienstprüfungen werden von den verfügbaren Workern parallel verarbeitet. Alle Erkennungsregeln mit asynchronen SNMPv3-Dienstprüfungen werden aufgrund der Besonderheiten der libsnmp-Implementierung von einem Worker verarbeitet; daher erhöht eine Erhöhung der Worker-Anzahl die Erkennungsgeschwindigkeit nicht.
Checks	Zabbix verwendet diese Liste von Prüfungen für die Erkennung. Klicken Sie auf Add, um eine neue Prüfung in einem Popup-Fenster zu konfigurieren. Unterstützte Prüfungen: SSH, LDAP, SMTP, FTP, HTTP, HTTPS, POP, NNTP, IMAP, TCP, Telnet, Zabbix Agent, SNMPv1 Agent, SNMPv2 Agent, SNMPv3 Agent, ICMP ping. Eine protokollbasierte Erkennung verwendet die Funktion **net.tcp.service[]**, um jeden Host zu testen, außer bei SNMP, das eine SNMP-OID abfragt. Der Zabbix Agent wird durch Abfrage eines Datenpunkts im unverschlüsselten Modus getestet. Weitere Details finden Sie unter Agent-Datenpunkte. Der Parameter 'Ports' kann einer der folgenden Werte sein: Einzelner Port: 22 Bereich von Ports: 22-45 Liste: 22-45,55,60-70 Seit Zabbix 7.0 werden alle Dienstprüfungen asynchron ausgeführt, außer LDAP-Prüfungen. Seit Zabbix 7.0 erfolgt die HTTP/HTTPS-Prüfung über libcurl. Wenn Zabbix Server/Proxy ohne libcurl kompiliert wurde, funktionieren HTTP-Prüfungen wie in früheren Versionen (d. h. als TCP-Prüfungen), HTTPS-Prüfungen funktionieren jedoch nicht.
Device uniqueness criteria	Die Kriterien für die Eindeutigkeit können sein: **IP address** - keine mehrfachen Geräte mit einzelner IP verarbeiten. Wenn bereits ein Gerät mit derselben IP vorhanden ist, wird es als bereits erkannt betrachtet und ein neuer Host wird nicht hinzugefügt. **<discovery check>** - entweder Zabbix Agent- oder SNMP-Agent-Prüfung. Beachten Sie, dass die bei der Erkennung verwendeten Eindeutigkeitskriterien nicht mit der Host-Identifizierung im System bei der Ausführung von Aktionen übereinstimmen. Die Eindeutigkeitskriterien bei der Erkennung legen fest, ob zwei oder mehr erkannte Geräte identisch (oder unterschiedlich) sind, während in Zabbix für die Host-Identifizierung nur die IP-Adresse als Kriterium verwendet wird (siehe Host creation).
Host name	Legen Sie den technischen Hostnamen eines erstellten Hosts fest mit: **DNS name** - DNS-Name (Standard) **IP address** - IP-Adresse **<discovery check>** - empfangener Zeichenfolgenwert der Erkennungsprüfung (z. B. Zabbix Agent-, SNMP-Agent-Prüfung) Siehe auch: Host naming.
Visible name	Legen Sie den sichtbaren Hostnamen eines erstellten Hosts fest mit: **Host name** - technischer Hostname (Standard) **DNS name** - DNS-Name **IP address** - IP-Adresse **<discovery check>** - empfangener Zeichenfolgenwert der Erkennungsprüfung (z. B. Zabbix Agent-, SNMP-Agent-Prüfung) Siehe auch: Host naming.
Enabled	Wenn das Kontrollkästchen aktiviert ist, ist die Regel aktiv und wird vom Zabbix Server ausgeführt. Wenn es nicht aktiviert ist, ist die Regel nicht aktiv. Sie wird nicht ausgeführt.

Überschreitung des Dateideskriptor-Limits

Bei einer großen Anzahl gleichzeitiger Prüfungen kann das Dateideskriptor-Limit für den Discovery-Manager erschöpft werden.

Die für die Erkennung erforderliche Anzahl an Dateideskriptoren entspricht der Anzahl der Discovery-Worker * 1000. Standardmäßig gibt es 5 Discovery-Worker, während das Soft-Limit des Systems bei etwa 1024 liegt.

Wenn sich dieses Limit nähert, reduziert Zabbix die Standardanzahl gleichzeitiger Prüfungen pro Typ für jeden Worker und schreibt eine Warnung in die Protokolldatei. Wenn der Benutzer jedoch einen höheren Wert für *Maximum concurrent checks per type* festgelegt hat als den von Zabbix berechneten Wert, verwendet Zabbix den benutzerdefinierten Wert für einen Worker.

Ein praxisnahes Szenario

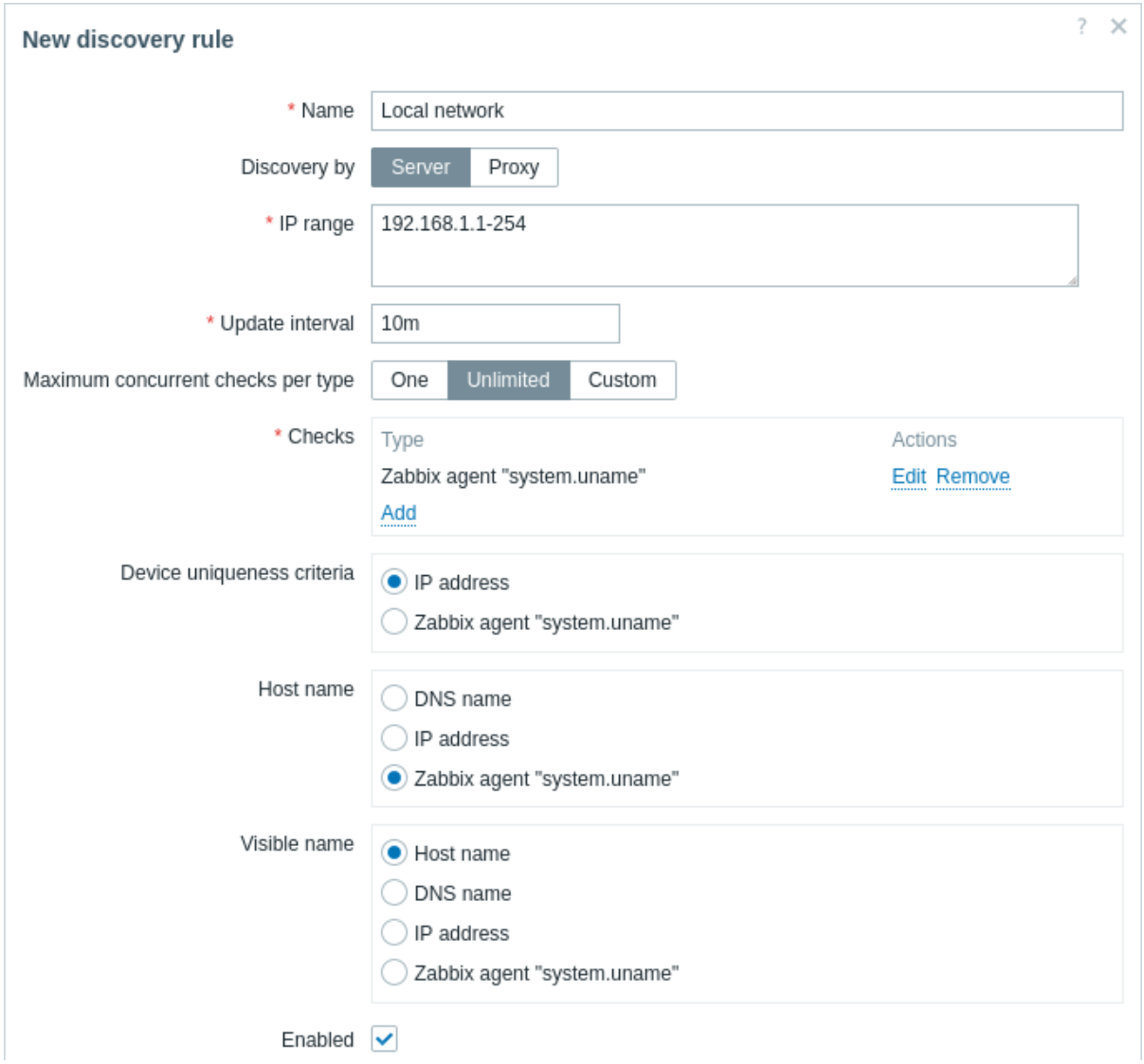
In diesem Beispiel möchten wir die Netzwerk-Erkennung für das lokale Netzwerk mit einem IP-Bereich von 192.168.1.1-192.168.1.254 einrichten.

In unserem Szenario möchten wir:

- diejenigen Hosts erkennen, auf denen der Zabbix Agent läuft
- die Erkennung alle 10 Minuten ausführen
- einen Host zur Überwachung hinzufügen, wenn die Host-Uptime mehr als 1 Stunde beträgt
- Hosts entfernen, wenn die Host-Downtime mehr als 24 Stunden beträgt
- Linux-Hosts zur Gruppe ”Linux servers” hinzufügen
- Windows-Hosts zur Gruppe ”Windows servers” hinzufügen
- die Vorlage *Linux* für Linux-Hosts verwenden
- die Vorlage *Windows* für Windows-Hosts verwenden

Schritt 1

Definieren einer Netzwerkerkennungsregel für unseren IP-Bereich.



Zabbix wird versuchen, Hosts im IP-Bereich 192.168.1.1-192.168.1.254 zu erkennen, indem eine Verbindung zu Zabbix Agents

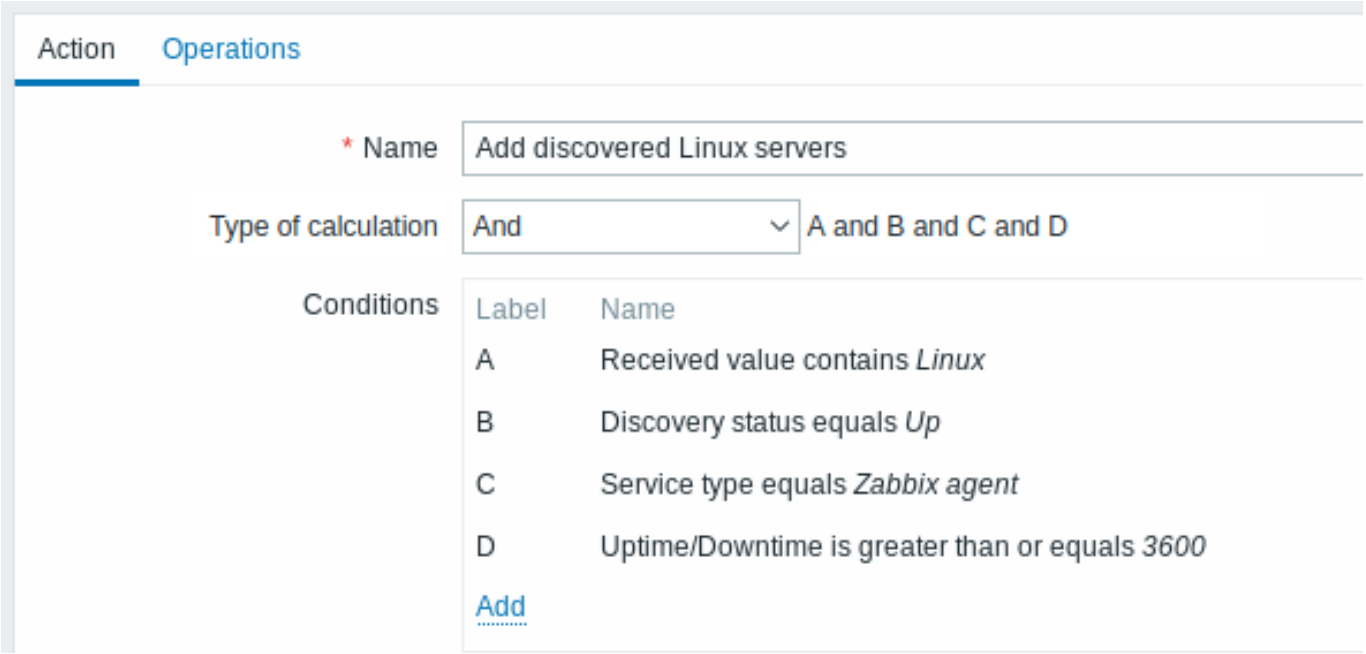
hergestellt und der Wert des Schlüssels **system.uname** abgerufen wird. Der vom Agent empfangene Wert kann verwendet werden, um die Hosts zu benennen und auch, um unterschiedliche Aktionen für verschiedene Betriebssysteme anzuwenden. Zum Beispiel Windows-Server mit der Vorlage *Windows* verknüpfen, Linux-Server mit der Vorlage *Linux*.

Die Regel wird alle 10 Minuten ausgeführt.

Wenn diese Regel hinzugefügt wird, startet Zabbix automatisch die Erkennung und die Erzeugung der auf der Erkennung basierenden Ereignisse zur weiteren Verarbeitung.

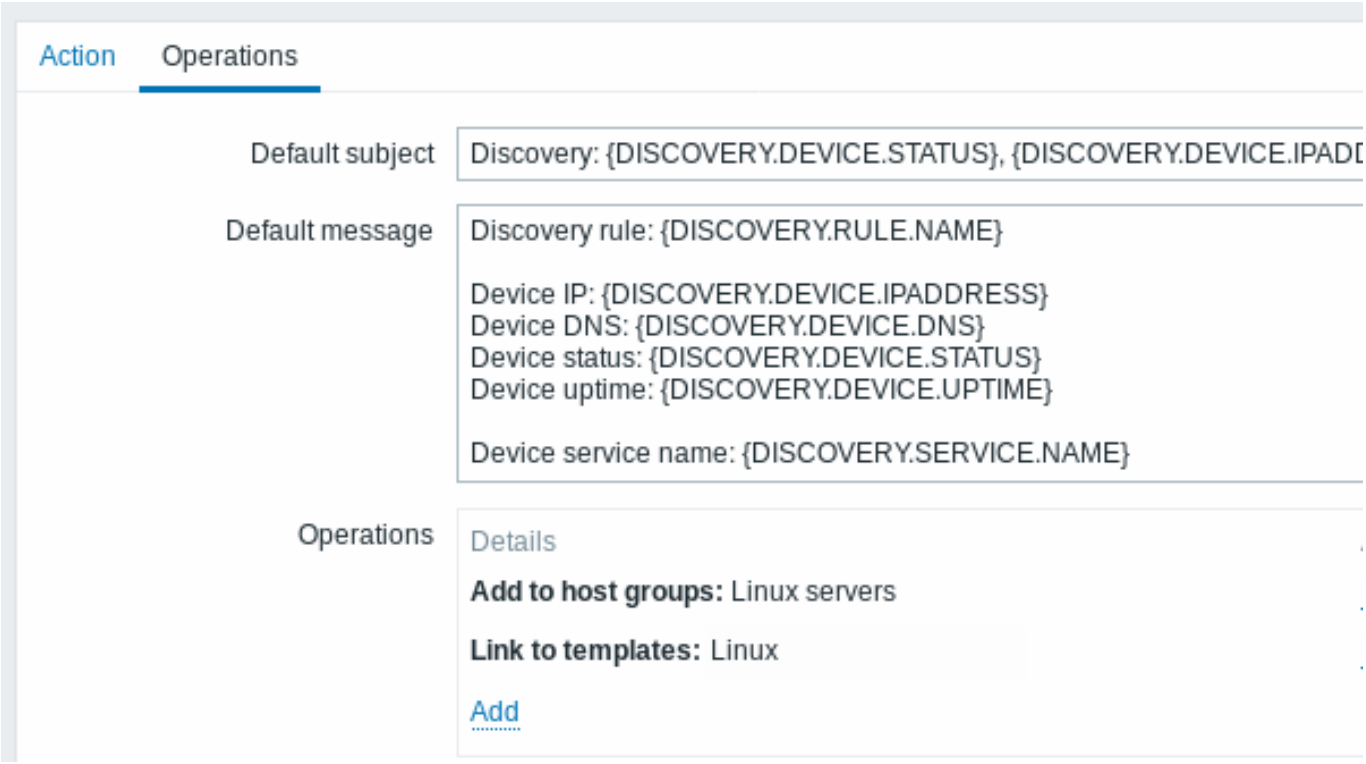
Schritt 2

Definieren einer Discovery-Aktion zum Hinzufügen der erkannten Linux-Server zur jeweiligen Gruppe/Vorlage.



Die Aktion wird aktiviert, wenn:

- der Dienst „Zabbix agent“ „up“ ist
- der Wert von system.uname (der Zabbix-Agent-Schlüssel, den wir in der Regeldefinition verwendet haben) „Linux“ enthält
- die Uptime 1 Stunde (3600 Sekunden) oder mehr beträgt

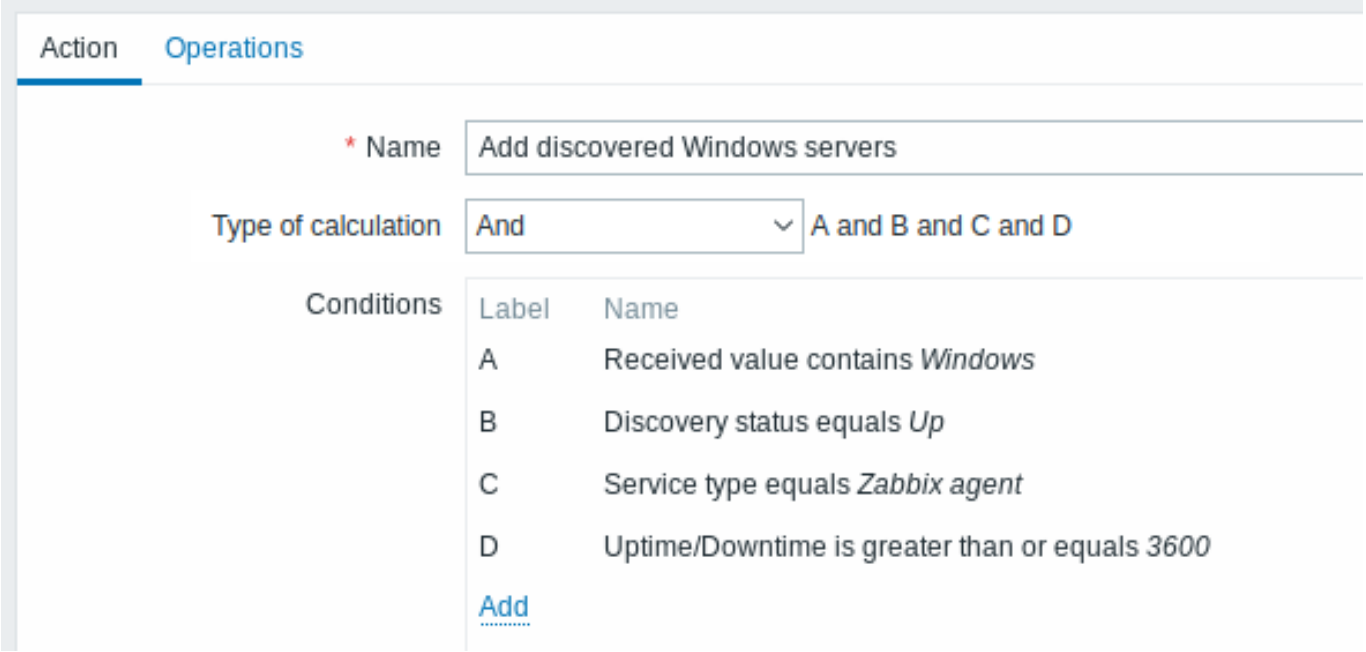


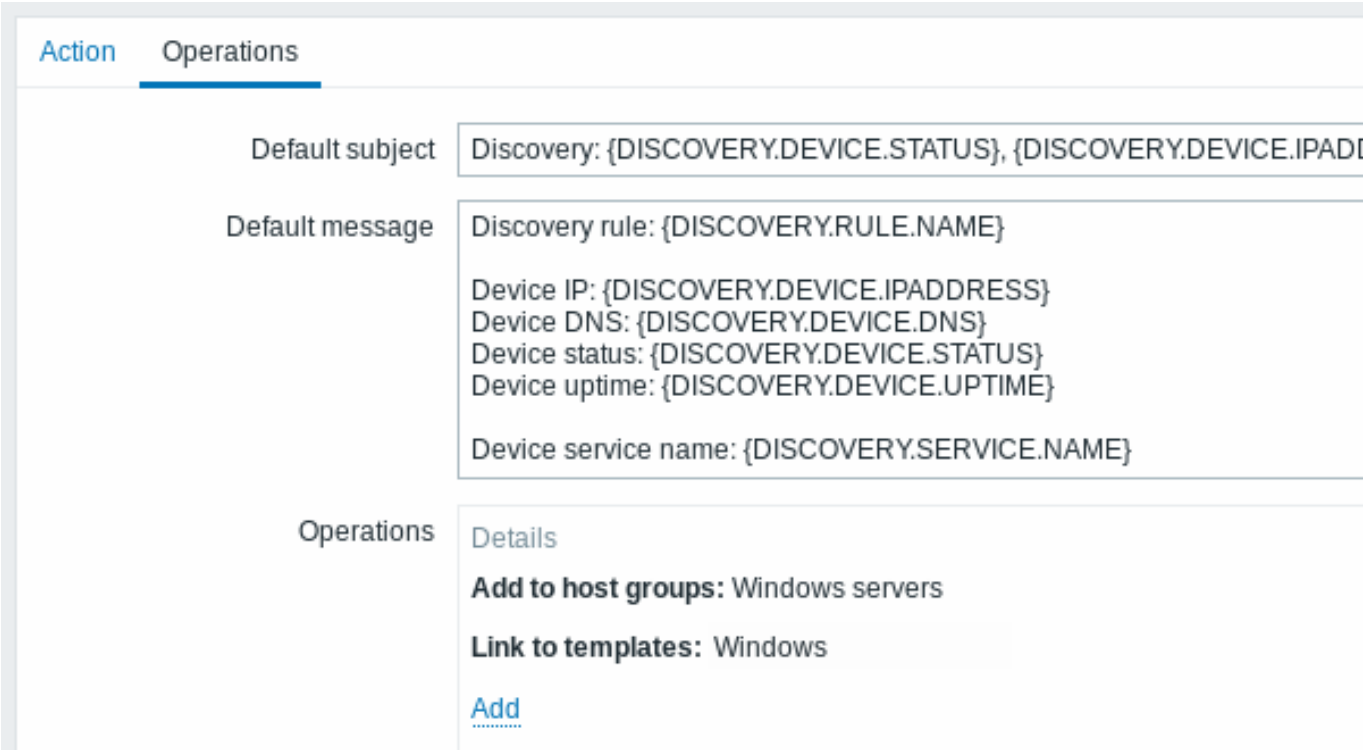
Die Aktion führt die folgenden Operationen aus:

- den erkannten Host zur Gruppe „Linux servers“ hinzufügen (und den Host auch hinzufügen, falls er zuvor nicht hinzugefügt wurde)
- den Host mit der Vorlage *Linux* verknüpfen. Zabbix beginnt automatisch mit der Überwachung des Hosts unter Verwendung der Datenpunkte und Auslöser aus der Vorlage „Linux“.

Schritt 3

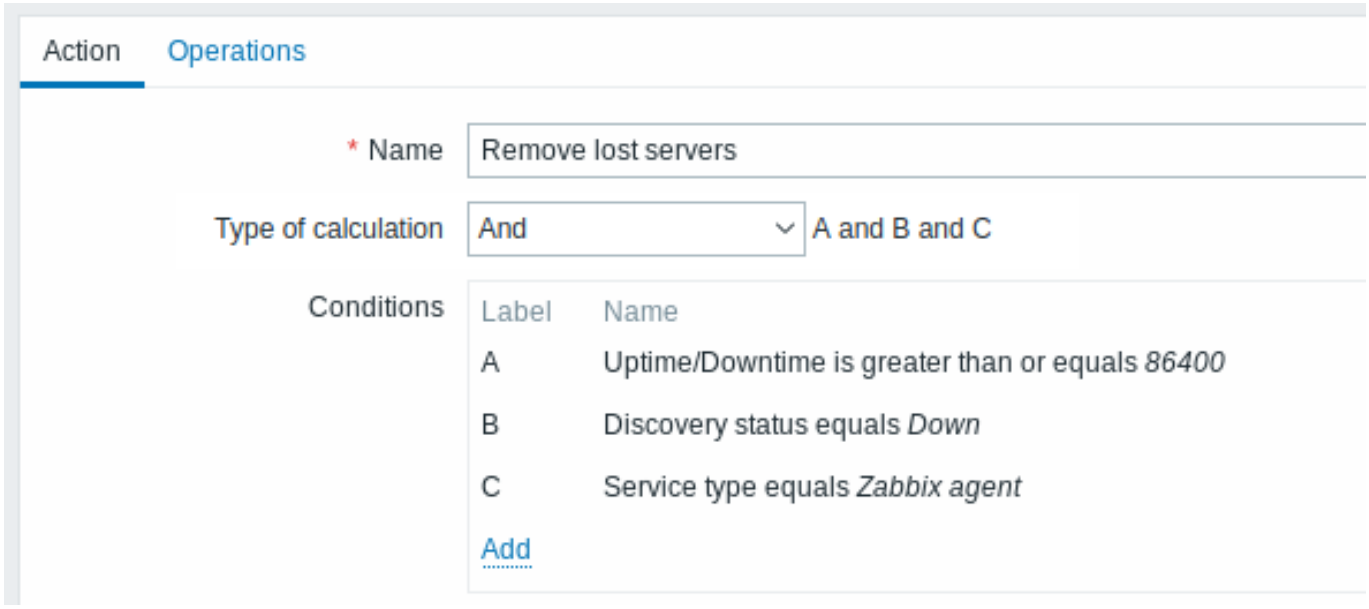
Definieren einer Discovery-Aktion, um die erkannten Windows-Server zur jeweiligen Gruppe/Vorlage hinzuzufügen.

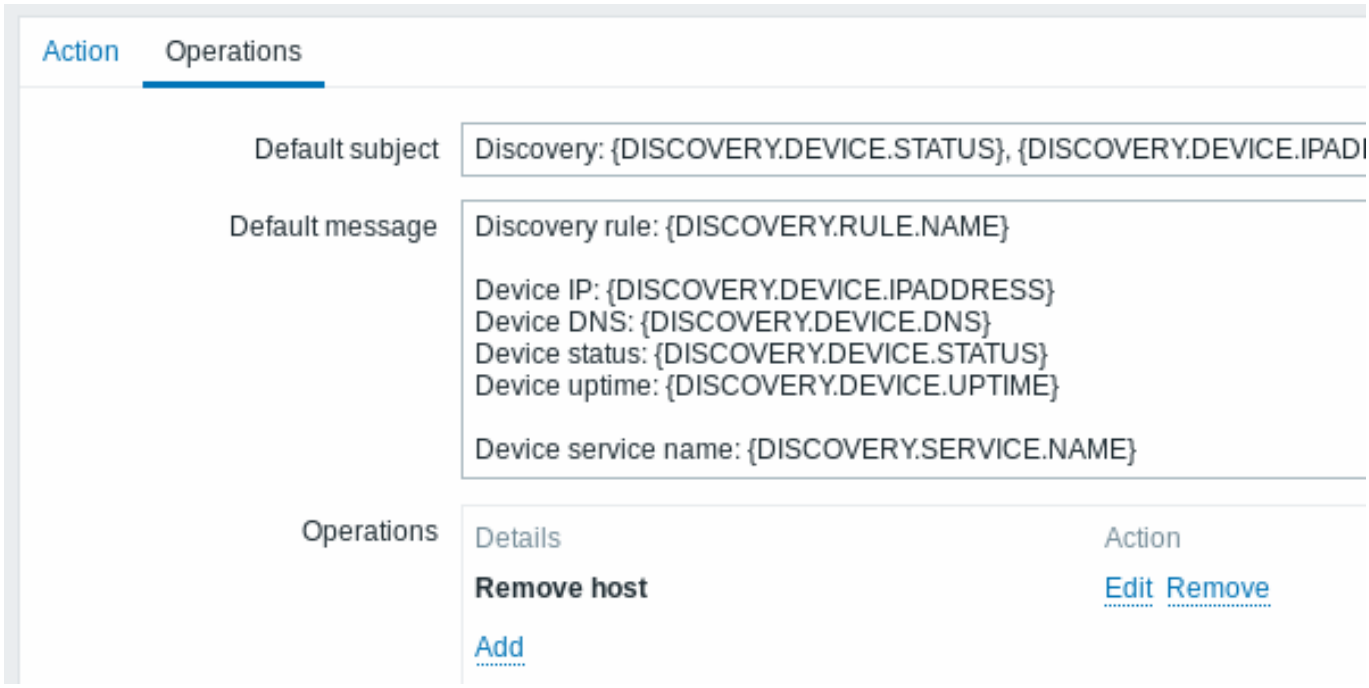




Schritt 4

Definieren einer Discovery-Aktion zum Entfernen verlorener Server.





Ein Server wird entfernt, wenn der Dienst „Zabbix Agent“ länger als 24 Stunden (86400 Sekunden) „down“ ist.

Aktive Agent-Autoregistrierung

Übersicht

Es ist möglich, die automatische Registrierung aktiver Zabbix Agenten zuzulassen, woraufhin der Server mit deren Überwachung beginnen kann. Auf diese Weise können neue Hosts zur Überwachung hinzugefügt werden, ohne sie manuell auf dem Server zu konfigurieren.

Die automatische Registrierung kann erfolgen, wenn ein zuvor unbekannter aktiver Agent nach Prüfungen fragt.

Diese Funktion kann für die automatische Überwachung neuer Cloud-Knoten sehr nützlich sein. Sobald Sie einen neuen Knoten in der Zabbix Cloud haben, beginnt Zabbix automatisch mit der Erfassung von Leistungs- und Verfügbarkeitsdaten des Hosts.

Die automatische Registrierung aktiver Agenten unterstützt auch die Überwachung hinzugefügter Hosts mit passiven Prüfungen. Wenn der aktive Agent nach Prüfungen fragt und die Konfigurationsparameter `ListenIP` oder `ListenPort` in der Konfigurationsdatei definiert sind, werden diese an den Server übermittelt. Wenn mehrere IP-Adressen angegeben sind, wird die erste an den Server gesendet.

Der Server verwendet beim Hinzufügen des neu automatisch registrierten Hosts die empfangene IP-Adresse und den Port, um den Agenten zu konfigurieren. Wenn kein IP-Adresswert empfangen wird, wird die für die eingehende Verbindung verwendete Adresse

genutzt. Wenn kein Portwert empfangen wird, wird 10050 verwendet.

Es ist möglich anzugeben, dass der Host mit einem DNS-Namen als standardmäßiger Agent-Schnittstelle automatisch registriert werden soll.

Die automatische Registrierung wird erneut ausgeführt:

- wenn sich die Host-Metadaten ändern:
  - aufgrund einer Änderung von HostMetadata und eines Neustarts des Agenten
  - aufgrund einer Änderung des von HostMetadataItem zurückgegebenen Werts
- für manuell erstellte Hosts, bei denen Metadaten fehlen
- wenn ein Host manuell so geändert wird, dass er von einem anderen Zabbix Proxy überwacht wird
- wenn die automatische Registrierung für denselben Host von einem neuen Zabbix Proxy kommt

Der Heartbeat für die automatische Registrierung aktiver Agenten für Zabbix Server und Zabbix Proxy beträgt 120 Sekunden. Falls also ein erkannter Host gelöscht wird, wird die automatische Registrierung nach 120 Sekunden erneut ausgeführt.

Konfiguration

Server angeben

Stellen Sie sicher, dass der Zabbix Server in der Agent-Konfigurationsdatei (`zabbix_agentd.conf`) angegeben ist:

```
ServerActive=10.0.0.1
```

Sofern Sie in `zabbix_agentd.conf` nicht ausdrücklich einen *Hostname* definieren, verwendet der Server den System-Hostnamen des Standorts des Agent für die Benennung des Hosts. Der System-Hostname unter Linux kann durch Ausführen des Befehls `hostname` ermittelt werden.

Wenn *Hostname* in der Zabbix-Agent-Konfiguration als durch Kommas getrennte Liste von Hosts definiert ist, werden für alle aufgeführten Hostnamen Hosts erstellt.

Starten Sie den Agent nach Änderungen an der Konfigurationsdatei neu.

Aktion für die automatische Registrierung aktiver Agenten

Wenn der Server eine Anfrage zur automatischen Registrierung von einem Agenten erhält, ruft er eine Aktion auf. Für die automatische Registrierung von Agenten muss eine Aktion mit der Ereignisquelle „Autoregistrierung“ konfiguriert werden.

> **Note:**
> Das Einrichten der Netzwerkerkennung ist nicht erforderlich, damit sich aktive Agenten automatisch registrieren.

Im Zabbix Frontend:

1. Gehen Sie zu *Alerts* > *Actions*.
2. Wählen Sie *Autoregistration actions*.
3. Klicken Sie auf *Create action*.
4. Geben Sie auf der Registerkarte *Action* einen Namen für Ihre Aktion ein.
5. Geben Sie optional Bedingungen an. In den Bedingungen für Hostnamen/Host-Metadaten können Sie eine Teilzeichenfolgen- oder reguläre Übereinstimmung verwenden. Wenn Sie den Bedingungstyp *Host metadata* verwenden möchten, lesen Sie Host-Metadaten verwenden.
6. Fügen Sie auf der Registerkarte *Operations* relevante Vorgänge hinzu, z. B. *Add host*, *Add to host group* (beispielsweise *Discovered hosts*), *Link templates* usw.

> **Note:**
> Wenn die Hosts, die automatisch registriert werden, voraussichtlich nur für die aktive Überwachung unterstützt werden (z. B. Hosts, die durch eine Firewall von Ihrem Zabbix Server getrennt sind), sollten Sie möglicherweise eine spezielle Vorlage wie *Template_Linux-active* erstellen, mit der sie verknüpft werden.

Erstellte Hosts werden der Gruppe *Discovered hosts* hinzugefügt (standardmäßig konfigurierbar unter *Administration* > *General* > *Other*). Wenn Hosts einer anderen Gruppe hinzugefügt werden sollen, fügen Sie einen Vorgang *Remove from host group* (mit Angabe von „Discovered hosts“) und zusätzlich einen Vorgang *Add to host group* (mit Angabe einer anderen Hostgruppe) hinzu, da ein Host einer Hostgruppe angehören muss.

Sichere automatische Registrierung

Eine sichere automatische Registrierung ist möglich, indem eine PSK-basierte Authentifizierung mit verschlüsselten Verbindungen konfiguriert wird.

Die Verschlüsselungsstufe wird global unter *Administration* > *Allgemein* > *Automatische Registrierung* konfiguriert. Es kann keine Verschlüsselung, TLS-Verschlüsselung mit PSK-Authentifizierung oder beides ausgewählt werden (sodass sich einige Hosts ohne Verschlüsselung registrieren können, während andere eine verschlüsselte Verbindung verwenden).

Die PSK-Authentifizierung wird vom Zabbix Server überprüft, bevor ein Host hinzugefügt wird. Bei erfolgreicher Überprüfung wird der Host hinzugefügt und *Verbindungen vom/zum Host* werden ausschließlich auf „PSK" gesetzt, wobei Identität und vorab gemeinsam genutzter Schlüssel mit den Werten der globalen Einstellung für die automatische Registrierung übereinstimmen.

Attention:
Um die Sicherheit der automatischen Registrierung in Installationen mit Proxys zu gewährleisten, sollte die Verschlüsselung zwischen Zabbix Server und Proxy aktiviert werden.

DNS als Standardschnittstelle verwenden

HostInterface und HostInterfaceItem Konfigurationsparameter ermöglichen es Ihnen, während der automatischen Registrierung einen benutzerdefinierten Wert für die Host-Schnittstelle anzugeben.

Genauer gesagt sind sie nützlich, wenn der Host mit einem DNS-Namen als Standardschnittstelle des Agent anstelle seiner IP-Adresse automatisch registriert werden soll. In diesem Fall sollte der DNS-Name als Wert des Parameters HostInterface oder HostInterfaceItem angegeben oder zurückgegeben werden. Wenn sich der Wert eines dieser Parameter ändert – beispielsweise von einer IP-Adresse zu einem DNS-Namen oder umgekehrt –, wird die Standardschnittstelle des automatisch registrierten Hosts entsprechend aktualisiert. Diese Aktualisierung wird auf den vorhandenen Host angewendet, anstatt einen neuen zu erstellen. Um den neuen Wert zu senden, muss der Agent neu gestartet werden, damit er den Prozess der automatischen Registrierung erneut initiiert.

Note:
Wenn die Parameter HostInterface oder HostInterfaceItem nicht konfiguriert sind, wird stattdessen der Parameter listen_dns verwendet. Dieser Wert wird durch eine umgekehrte DNS-Suche der IP-Adresse des Agent ermittelt. Wenn Reverse DNS nicht ordnungsgemäß konfiguriert ist oder einen ungültigen Namen zurückgibt, kann dies aufgrund eines ungültigen Schnittstellenwerts zu einer falschen oder fehlgeschlagenen automatischen Registrierung führen.

Host-Metadaten verwenden

Wenn der Agent eine Anfrage zur automatischen Registrierung an den Server sendet, übermittelt er seinen Hostnamen. In einigen Fällen (z. B. bei Amazon-Cloud-Knoten) reicht ein Hostname nicht aus, damit der Zabbix-Server erkannte Hosts unterscheiden kann. Host-Metadaten können optional verwendet werden, um weitere Informationen von einem Agent an den Server zu senden.

Host-Metadaten werden in der Agent-Konfigurationsdatei (`zabbix_agentd.conf`) konfiguriert. Es gibt zwei Möglichkeiten, Host-Metadaten in der Konfigurationsdatei anzugeben:

```
HostMetadata
HostMetadataItem
```

Eine Beschreibung der Optionen finden Sie unter dem obigen Link.

Der Parameter HostMetadataItem kann bis zu 65535 UTF-8-Codepoints zurückgeben. Ein längerer Wert wird abgeschnitten.

Beachten Sie, dass bei MySQL die effektive maximale Länge in Zeichen geringer ist, wenn der zurückgegebene Wert Multibyte-Zeichen enthält. Beispielsweise ist ein Wert, der ausschließlich 3-Byte-Zeichen enthält, insgesamt auf 21844 Zeichen begrenzt, während ein Wert, der ausschließlich 4-Byte-Zeichen enthält, auf 16383 Symbole begrenzt ist.

Attention:
Ein Versuch zur automatischen Registrierung erfolgt jedes Mal, wenn ein aktiver Agent eine Anfrage zur Aktualisierung der aktiven Prüfungen an den Server sendet. Die Verzögerung zwischen den Anfragen wird im Agent-Parameter RefreshActiveChecks festgelegt. Die erste Anfrage wird unmittelbar nach dem Neustart des Agent gesendet.

Beispiele

Automatische Registrierung nach Betriebssystem mit HostMetadata

Angenommen, Sie möchten, dass die Hosts durch den Zabbix Server automatisch registriert werden. In Ihrem Netzwerk sind aktive Zabbix Agents vorhanden (siehe Abschnitt „Konfiguration" oben). In Ihrem Netzwerk gibt es Windows-Hosts und Linux-Hosts, und in Ihrem Zabbix Frontend sind die Vorlagen „Linux by Zabbix agent" und „Windows by Zabbix agent" verfügbar. Bei der Host-Registrierung soll also die passende Linux-/Windows-Vorlage auf den registrierten Host angewendet werden. Standardmäßig wird bei der automatischen Registrierung nur der Hostname an den Server gesendet, was möglicherweise nicht ausreicht. Um sicherzustellen, dass die richtige Vorlage auf den Host angewendet wird, sollten Sie Host-Metadaten verwenden.

Frontend-Konfiguration

Als Erstes ist das Frontend zu konfigurieren. Erstellen Sie 2 Aktionen. Die erste Aktion:

- Name: Linux-Host-Autoregistrierung
- Bedingungen: Host-Metadaten enthalten *Linux*
- Operationen: Vorlagen verknüpfen: Linux by Zabbix Agent

> **Note:**
> In diesem Fall können Sie die Operation „Host hinzufügen" überspringen. Zum Verknüpfen einer Vorlage mit einem Host muss der Host zuerst hinzugefügt werden, daher übernimmt der Server dies automatisch.

Die zweite Aktion:

- Name: Windows-Host-Autoregistrierung
- Bedingungen: Host-Metadaten enthalten *Windows*
- Operationen: Vorlagen verknüpfen: Windows by Zabbix Agent

Agent-Konfiguration

Nun müssen Sie die Agenten konfigurieren. Fügen Sie die folgende Zeile zu den Agent-Konfigurationsdateien hinzu:

```
HostMetadataItem=system.uname
```

Auf diese Weise stellen Sie sicher, dass die Host-Metadaten je nach Host, auf dem ein Agent ausgeführt wird, „Linux" oder „Windows" enthalten. Ein Beispiel für Host-Metadaten in diesem Fall:

```
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
```

Vergessen Sie nicht, den Agent nach Änderungen an der Konfigurationsdatei neu zu starten.

Verwendung von HostMetadata zur Steuerung der automatischen Registrierung und von Vorlagen

Schritt 1 - Registrierung mit HostMetadata schützen

Verwenden Sie Host-Metadaten, um einen grundlegenden Schutz gegen die Registrierung unerwünschter Hosts zu ermöglichen.

Frontend-Konfiguration

Erstellen Sie im Frontend eine Aktion und verwenden Sie dabei einen schwer zu erratenden geheimen Code, um unerwünschte Hosts auszuschließen:

- Name: Autoregistrierungsaktion Linux
- Bedingungen:
  - Berechnungstyp: UND
  - Bedingung (A): Host-Metadaten enthalten //Linux//
  - Bedingung (B): Host-Metadaten enthalten //21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae//
- Operationen:
  - Nachricht an Benutzer senden: Admin über alle Medien
  - Zu Host-Gruppen hinzufügen: Linux-Server
  - Vorlagen verknüpfen: Linux by Zabbix agent

Bitte beachten Sie, dass diese Methode allein keinen starken Schutz bietet, da Daten im Klartext übertragen werden. Damit Änderungen sofort wirksam werden, ist ein Neuladen des Konfigurationscaches erforderlich.

Agent-Konfiguration

Fügen Sie die folgende Zeile zur Agent-Konfigurationsdatei hinzu:

```
HostMetadata=Linux    21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
```

wobei „Linux" eine Plattform ist und der Rest der Zeichenfolge der schwer zu erratende geheime Text ist.

Vergessen Sie nicht, den Agent nach Änderungen an der Konfigurationsdatei neu zu starten.

Schritt 2 - Vorlage zu registriertem Host hinzufügen

Es ist möglich, zusätzliche Vorlagen für einen bereits registrierten Host hinzuzufügen. In diesem Fall wird die Vorlage MySQL by Zabbix agent nur mit Hosts verknüpft, deren HostMetadata das Token `MySQL` enthält.

Frontend-Konfiguration

Aktualisieren Sie die Aktion im Frontend:

- Name: Autoregistrierungsaktion Linux
- Bedingungen:

    - Berechnungstyp: UND
    - Bedingung (A): Host-Metadaten enthalten Linux
    - Bedingung (B): Host-Metadaten enthalten 21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
    - Bedingung (C): Host-Metadaten enthalten MySQL
- Operationen:
    - Nachricht an Benutzer senden: Admin über alle Medien
    - Zu Host-Gruppen hinzufügen: Linux-Server
    - Vorlagen verknüpfen: Linux by Zabbix agent
    - Vorlagen verknüpfen: MySQL by Zabbix Agent

Agent-Konfiguration

Aktualisieren Sie die folgende Zeile in der Agent-Konfigurationsdatei:

```
HostMetadata=MySQL on Linux 21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
```

Vergessen Sie nicht, den Agent nach Änderungen an der Konfigurationsdatei neu zu starten.

Low-level-Erkennung

Übersicht Low-Level Discovery (LLD) erkennt automatisch Entitäten (z. B. Dateisysteme, Netzwerkschnittstellen) auf einem Host und erstellt die entsprechenden Datenpunkte, Auslöser und Diagramme ohne manuelle Einrichtung für jede einzelne Entität.

Um LLD zu verwenden, erstellen Sie eine **Erkennungsregel**, die JSON-Daten mit einer Beschreibung der Entitäten sammelt, sowie **Prototypen** (Datenpunkt, Auslöser und Diagramm) für jede Entität.

Das von der Regel zurückgegebene JSON muss ein Array von Objekten sein, wobei jedes Objekt eine erkannte Entität mithilfe von Schlüssel-Wert-Paaren darstellt. Beispielsweise kann eine Erkennungsregel mit `net.if.discovery` Folgendes zurückgeben:

```
[
  {"{#IFNAME}": "lo"},
  {"{#IFNAME}": "eth0"}
]
```

Ein entsprechender Satz von Datenpunkten, Auslösern und Diagrammen wird dann aus den Prototypen erstellt - jeweils einer pro Schnittstelle (`lo` und `eth0`).

Wenn Objekte Schlüssel im Format `{#MACRO}` enthalten, werden diese Makros direkt in Prototypen verwendet. Wenn nicht oder wenn zusätzliche/benutzerdefinierte Makros erforderlich sind, können Makros manuell definiert und mithilfe von JSONPath JSON-Feldern zugeordnet werden.

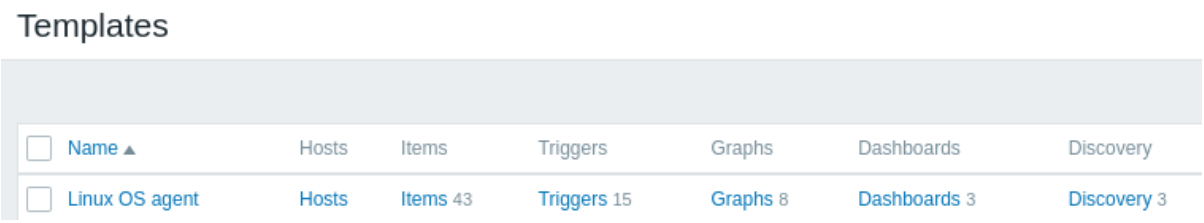
LLD kann auch Hosts erstellen (z. B. für auf einem Hypervisor erkannte virtuelle Maschinen) und unterstützt verschachtelte Erkennungsregeln für mehrstufige Erkennung.

Entitäten, die nicht mehr erkannt werden, können automatisch deaktiviert oder gelöscht werden.

Konfigurieren der Low-Level-Discovery Wir veranschaulichen die Low-Level-Discovery anhand eines Beispiels für die Erkennung von Dateisystemen.

Gehen Sie wie folgt vor, um die Discovery zu konfigurieren:

1. Gehen Sie zu *Datenerfassung* > *Vorlagen* oder *Hosts*.
2. Klicken Sie in der Zeile der entsprechenden Vorlage/des entsprechenden Hosts auf *Discovery*.



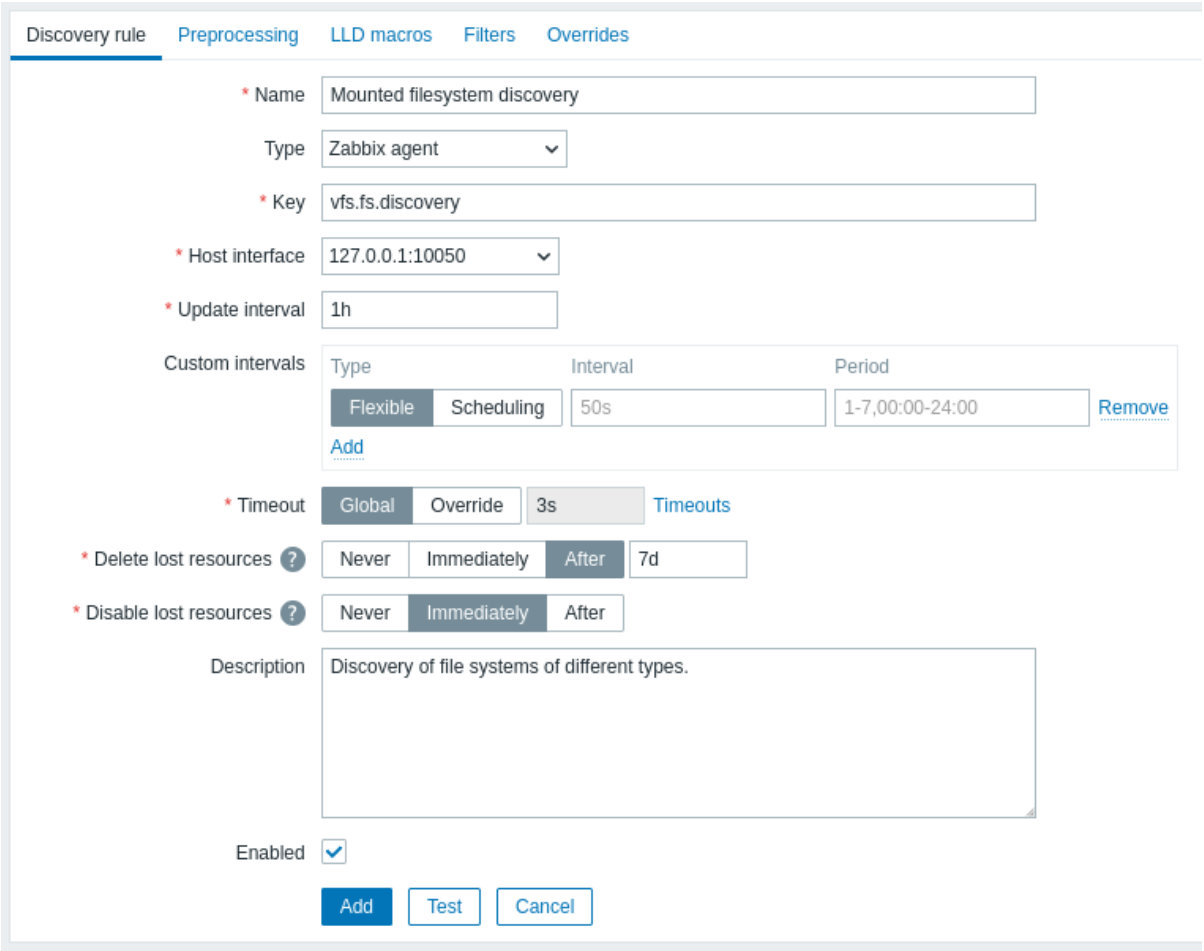
3. Klicken Sie oben rechts auf dem Bildschirm auf *Discovery-Regel erstellen*.
4. Füllen Sie das Formular für die Discovery-Regel mit den erforderlichen Details aus.

Discovery-Regel

Das Formular für Discovery-Regeln enthält fünf Registerkarten, die von links nach rechts den Datenfluss während der Discovery darstellen:

- *Discovery-Regel* – Geben Sie hier vor allem den integrierten Datenpunkt oder das benutzerdefinierte Skript an, mit dem die Discovery-Daten abgerufen werden.
- *Vorverarbeitung* – Wenden Sie eine Vorverarbeitung auf die ermittelten Daten an.
- *LLD-Makros* – Ermöglichen Sie das Extrahieren von Makrowerten zur Verwendung in ermittelten Datenpunkten, Auslösern usw.
- *Filter* – Ermöglichen Sie das Filtern ermittelter Werte.
- *Überschreibungen* – Ermöglichen Sie die Änderung von Datenpunkten, Auslösern, Graphen oder Host-Prototypen bei der Anwendung auf bestimmte ermittelte Objekte.

Die Registerkarte **Discovery-Regel** enthält den für die Discovery zu verwendenden Datenpunktschlüssel sowie einige allgemeine Attribute der Discovery-Regel:



Alle Pflichtfelder sind mit einem roten Sternchen gekennzeichnet.

Parameter	Beschreibung
Name	Name der Discovery-Regel.
Typ	Der Typ der Prüfung, mit der die Discovery durchgeführt wird. In diesem Beispiel verwenden wir einen Datenpunkt vom Typ *Zabbix agent*. Die Discovery-Regel kann auch ein abhängiger Datenpunkt sein, der von einem regulären Datenpunkt abhängt. Sie kann nicht von einer anderen Discovery-Regel abhängen. Wählen Sie für einen abhängigen Datenpunkt den entsprechenden Typ (*Dependent item*) aus und geben Sie den Master-Datenpunkt im Feld „Master item“ an. Der Master-Datenpunkt muss vorhanden sein.
Key	Geben Sie den Schlüssel des Discovery-Datenpunkts ein (bis zu 2048 Zeichen). Sie können beispielsweise den integrierten Datenpunktschlüssel „vfs.fs.discovery“ verwenden, um eine JSON-Zeichenfolge mit der Liste der auf dem Computer vorhandenen Dateisysteme, deren Typen und Einhängeoptionen zurückzugeben. Beachten Sie, dass eine weitere Möglichkeit zur Dateisystem-Discovery darin besteht, Discovery-Ergebnisse über den Agent-Schlüssel „vfs.fs.get“ zu verwenden (siehe Beispiel).

Parameter	Beschreibung
Update interval	Dieses Feld gibt an, wie häufig Zabbix die Discovery durchführt. Zu Beginn, wenn Sie die Dateisystem-Discovery gerade einrichten, möchten Sie möglicherweise ein kurzes Intervall festlegen. Sobald Sie wissen, dass sie funktioniert, können Sie 30 Minuten oder mehr einstellen, da sich Dateisysteme normalerweise nicht sehr häufig ändern. Zeit-Suffixe werden unterstützt, z. B. 30s, 1m, 2h, 1d. Benutzermakros werden unterstützt. *Hinweis*: Das Aktualisierungsintervall kann nur auf „0“ gesetzt werden, wenn benutzerdefinierte Intervalle mit einem Wert ungleich null vorhanden sind. Wenn es auf „0“ gesetzt wird und ein benutzerdefiniertes Intervall (flexibel oder geplant) mit einem Wert ungleich null vorhanden ist, wird der Datenpunkt während der Dauer des benutzerdefinierten Intervalls abgefragt. Neue Discovery-Regeln werden innerhalb von 60 Sekunden nach ihrer Erstellung geprüft, sofern sie nicht über ein geplantes oder flexibles Aktualisierungsintervall verfügen und *Update interval* auf 0 gesetzt ist. *Hinweis*: Bei einer vorhandenen Discovery-Regel kann die Discovery sofort durchgeführt werden, indem Sie auf die Schaltfläche *Execute now* Schaltfläche klicken.
Custom intervals	Sie können benutzerdefinierte Regeln für die Prüfung des Datenpunkts erstellen: **Flexible** – Erstellt eine Ausnahme für *Update interval* (ein Intervall mit einer anderen Häufigkeit). **Scheduling** – Erstellt einen benutzerdefinierten Abfragezeitplan. Detaillierte Informationen finden Sie unter Benutzerdefinierte Intervalle.
Timeout	Legen Sie das Timeout für die Discovery-Prüfung fest. Wählen Sie die Timeout-Option aus: **Global** – Das Proxy-/globale Timeout wird verwendet (im ausgegrauten Feld *Timeout* angezeigt). **Override** – Ein benutzerdefiniertes Timeout wird verwendet (im Feld *Timeout* festgelegt; zulässiger Bereich: 1–600 s). Zeit-Suffixe, z. B. 30s und 1m, sowie Benutzermakros werden unterstützt. Über den Link *Timeouts* können Sie Proxy-Timeouts oder globale Timeouts konfigurieren (wenn kein Proxy verwendet wird). Beachten Sie, dass der Link *Timeouts* nur Benutzern vom Typ *Super admin* angezeigt wird, die über Berechtigungen für die Frontend-Bereiche *Administration* > *General* oder *Administration* > *Proxies* verfügen.
Delete lost resources	Geben Sie an, wie schnell die ermittelte Entität gelöscht wird, nachdem ihr Discovery-Status zu „Nicht mehr ermittelt“ geworden ist: *Never* – Sie wird nicht gelöscht. *Immediately* – Sie wird sofort gelöscht. *After* – Sie wird nach dem angegebenen Zeitraum gelöscht. Der Wert muss größer sein als der Wert von *Disable lost resources*. Zeit-Suffixe werden unterstützt, z. B. 2h und 1d. Benutzermakros werden unterstützt. *Hinweis*: Die Verwendung von „Immediately“ wird nicht empfohlen, da bereits eine fehlerhafte Bearbeitung des Filters dazu führen kann, dass die Entität zusammen mit allen historischen Daten gelöscht wird. Beachten Sie, dass manuell deaktivierte Ressourcen nicht durch Low-Level-Discovery gelöscht werden.
Disable lost resources	Geben Sie an, wie schnell die ermittelte Entität deaktiviert wird, nachdem ihr Discovery-Status zu „Nicht mehr ermittelt“ geworden ist: *Never* – Sie wird nicht deaktiviert. *Immediately* – Sie wird sofort deaktiviert. *After* – Sie wird nach dem angegebenen Zeitraum deaktiviert. Der Wert sollte größer sein als das Aktualisierungsintervall der Discovery-Regel. Beachten Sie, dass automatisch deaktivierte Ressourcen wieder aktiviert werden, wenn sie durch die Low-Level-Discovery erneut ermittelt werden. Manuell deaktivierte Ressourcen werden bei einer erneuten Ermittlung nicht wieder aktiviert. Dieses Feld wird nicht angezeigt, wenn *Delete lost resources* auf „Immediately“ gesetzt ist. Zeit-Suffixe werden unterstützt, z. B. 2h und 1d. Benutzermakros werden unterstützt.
Description	Geben Sie eine Beschreibung ein.
Enabled	Wenn diese Option aktiviert ist, wird die Regel verarbeitet.

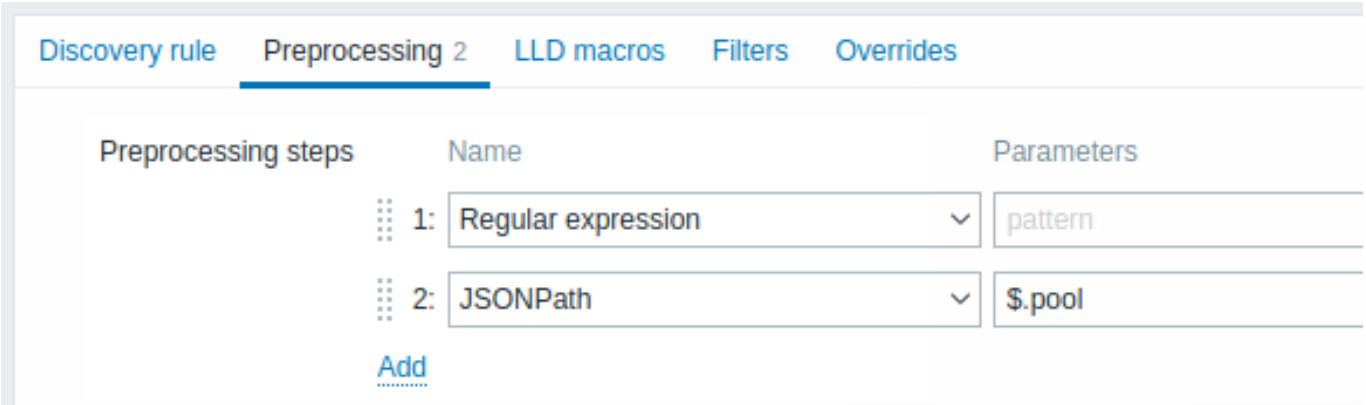
Note:
Der Verlauf der Discovery-Regel wird nicht gespeichert.

Vorverarbeitung

Auf der Registerkarte **Vorverarbeitung** können Sie Transformationsregeln definieren, die auf das Ergebnis der Erkennung angewendet werden sollen. In diesem Schritt sind eine oder mehrere Transformationen möglich. Transformationen werden in der Reihenfolge ausgeführt, in der sie definiert wurden. Die gesamte Vorverarbeitung wird vom Zabbix Server durchgeführt.

Siehe auch:

- Details zur Vorverarbeitung
- Testen der Vorverarbeitung



Typ		
	Transformation	Beschreibung
Text		
	Regulärer Ausdruck	Den empfangenen Wert mit dem regulären Ausdruck <pattern> abgleichen und den Wert durch die extrahierte Ausgabe <output> ersetzen. Der reguläre Ausdruck unterstützt die Extraktion von maximal 10 erfassten Gruppen mit der Sequenz \N. Parameter: **pattern** - regulärer Ausdruck **output** - Vorlage für die Ausgabeformatierung. Eine Escape-Sequenz \N (wobei N=1…9) wird durch die N-te übereinstimmende Gruppe ersetzt. Eine Escape-Sequenz \0 wird durch den übereinstimmenden Text ersetzt. Wenn Sie das Kontrollkästchen *Bei Fehler benutzerdefiniert* aktivieren, können Sie benutzerdefinierte Optionen zur Fehlerbehandlung festlegen: den Wert verwerfen, einen bestimmten Wert festlegen oder eine bestimmte Fehlermeldung festlegen.
	Ersetzen	Die Suchzeichenfolge suchen und durch eine andere Zeichenfolge (oder nichts) ersetzen. Alle Vorkommen der Suchzeichenfolge werden ersetzt. Parameter: **search string** - die zu suchende und zu ersetzende Zeichenfolge, unter Beachtung der Groß-/Kleinschreibung (erforderlich) **replacement** - die Zeichenfolge, durch die die Suchzeichenfolge ersetzt werden soll. Die Ersetzungszeichenfolge kann auch leer sein, wodurch die Suchzeichenfolge bei ihrem Auffinden gelöscht werden kann. Es ist möglich, Escape-Sequenzen zu verwenden, um Zeilenumbrüche, Wagenrückläufe, Tabulatoren und Leerzeichen zu suchen oder zu ersetzen: ”\n \r \t \s”; der umgekehrte Schrägstrich kann als ”\\” maskiert werden und Escape-Sequenzen können als ”\\n” maskiert werden. Das Maskieren von Zeilenumbrüchen, Wagenrückläufen und Tabulatoren erfolgt während der Low-Level-Erkennung automatisch.
Strukturierte Daten		
	JSONPath	Einen Wert oder ein Fragment aus JSON-Daten mithilfe der JSONPath-Funktionalität extrahieren. Wenn Sie das Kontrollkästchen *Bei Fehler benutzerdefiniert* aktivieren, können Sie benutzerdefinierte Optionen zur Fehlerbehandlung festlegen: den Wert verwerfen, einen bestimmten Wert festlegen oder eine bestimmte Fehlermeldung festlegen.

Typ	

XML XPath	Einen Wert oder ein Fragment aus XML-Daten mithilfe der XPath-Funktionalität extrahieren.
	Damit diese Option funktioniert, muss der Zabbix Server mit libxml-Unterstützung kompiliert worden sein.
	Beispiele:
	`number(/document/item/value)` extrahiert 10 aus `<document><item><value>10</value></item></document>`
	`number(/document/item/@attribute)` extrahiert 10 aus `<document><item attribute="10"></item></document>`
	`/document/item` extrahiert `<item><value>10</value></item>` aus `<document><item><value>10</value></item></document>`
	Beachten Sie, dass Namespaces nicht unterstützt werden.
	Wenn Sie das Kontrollkästchen *Bei Fehler benutzerdefiniert* aktivieren, können Sie benutzerdefinierte Optionen zur Fehlerbehandlung festlegen: den Wert verwerfen, einen bestimmten Wert festlegen oder eine bestimmte Fehlermeldung festlegen.

CSV in JSON	Daten einer CSV-Datei in das JSON-Format konvertieren. Weitere Informationen finden Sie unter: CSV-zu-JSON-Vorverarbeitung.
XML in JSON	Daten im XML-Format in JSON konvertieren. Weitere Informationen finden Sie unter: Serialisierungsregeln. Wenn Sie das Kontrollkästchen *Bei Fehler benutzerdefiniert* aktivieren, können Sie benutzerdefinierte Optionen zur Fehlerbehandlung festlegen: den Wert verwerfen, einen bestimmten Wert festlegen oder eine bestimmte Fehlermeldung festlegen.
SNMP	
SNMP-Walk-Wert	Wert anhand des angegebenen OID-/MIB-Namens extrahieren und Formatierungsoptionen anwenden: **Unverändert** - Hex-STRING als nicht maskierte Hex-Zeichenfolge zurückgeben (*Hinweis*: Anzeigehinweise werden weiterhin angewendet); **UTF-8 aus Hex-STRING** - Hex-STRING in eine UTF-8-Zeichenfolge konvertieren; **MAC aus Hex-STRING** - Hex-STRING in eine MAC-Adresszeichenfolge konvertieren (wobei `' '` durch `':'` ersetzt wird); **Integer aus BITS** - die ersten 8 Bytes einer Bitzeichenfolge, die als Folge von Hexadezimalzeichen ausgedrückt wird (z. B. ”1A 2B 3C 4D”), in eine vorzeichenlose 64-Bit-Ganzzahl konvertieren. Bei Bitzeichenfolgen mit mehr als 8 Bytes werden die nachfolgenden Bytes ignoriert. Wenn Sie das Kontrollkästchen *Bei Fehler benutzerdefiniert* aktivieren, können Sie benutzerdefinierte Optionen zur Fehlerbehandlung festlegen: den Wert verwerfen, einen bestimmten Wert festlegen oder eine bestimmte Fehlermeldung festlegen.
SNMP-Walk in JSON	SNMP-Werte in JSON konvertieren. Geben Sie einen Feldnamen im JSON und den entsprechenden SNMP-OID-Pfad an. Feldwerte werden mit den Werten aus dem angegebenen SNMP-OID-Pfad befüllt. Sie können diesen Vorverarbeitungsschritt für die SNMP-OID-Erkennung verwenden. Ähnliche Optionen zur Wertformatierung wie im Schritt *SNMP-Walk-Wert* sind verfügbar. Wenn Sie das Kontrollkästchen *Bei Fehler benutzerdefiniert* aktivieren, können Sie benutzerdefinierte Optionen zur Fehlerbehandlung festlegen: den Wert verwerfen, einen bestimmten Wert festlegen oder eine bestimmte Fehlermeldung festlegen.
SNMP-Get-Wert	Formatierungsoptionen auf den SNMP-Get-Wert anwenden: **UTF-8 aus Hex-STRING** - Hex-STRING in eine UTF-8-Zeichenfolge konvertieren; **MAC aus Hex-STRING** - Hex-STRING in eine MAC-Adresszeichenfolge konvertieren (wobei `' '` durch `':'` ersetzt wird); **Integer aus BITS** - die ersten 8 Bytes einer Bitzeichenfolge, die als Folge von Hexadezimalzeichen ausgedrückt wird (z. B. ”1A 2B 3C 4D”), in eine vorzeichenlose 64-Bit-Ganzzahl konvertieren. Bei Bitzeichenfolgen mit mehr als 8 Bytes werden die nachfolgenden Bytes ignoriert. Wenn Sie das Kontrollkästchen *Bei Fehler benutzerdefiniert* aktivieren, können Sie benutzerdefinierte Optionen zur Fehlerbehandlung festlegen: den Wert verwerfen, einen bestimmten Wert festlegen oder eine bestimmte Fehlermeldung festlegen.
Benutzerdefinierte Skripte	
JavaScript	Geben Sie JavaScript-Code im modalen Editor ein, der geöffnet wird, wenn Sie in das Parameterfeld oder auf das Stiftsymbol daneben klicken. Beachten Sie, dass die verfügbare JavaScript-Länge von der verwendeten Datenbank abhängt. Weitere Informationen finden Sie unter: JavaScript-Vorverarbeitung

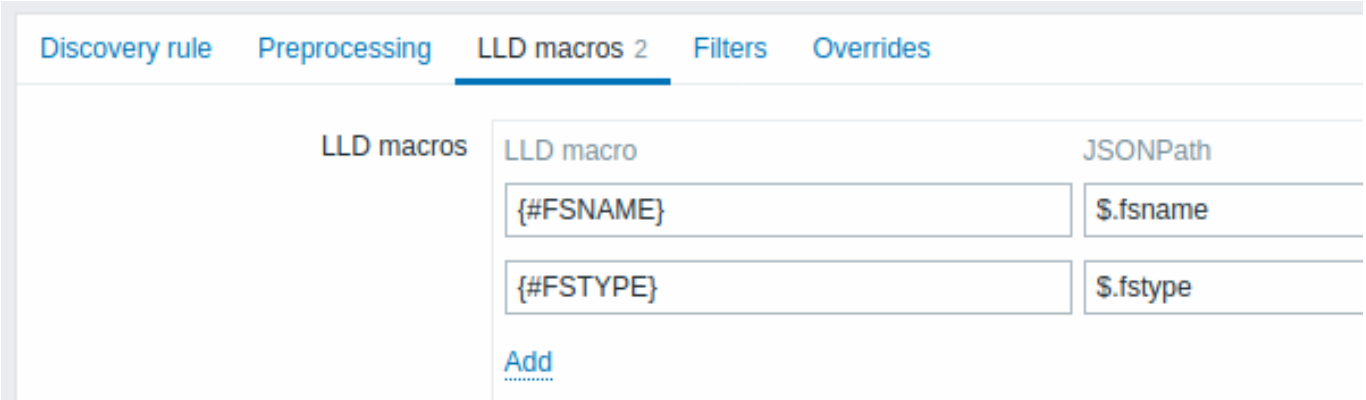
Typ	
Validierung	
Entspricht nicht dem regulären Ausdruck	Geben Sie einen regulären Ausdruck an, dem ein Wert nicht entsprechen darf. Beispiel: `Error:(.*?)\.` Wenn Sie das Kontrollkästchen *Bei Fehler benutzerdefiniert* aktivieren, können Sie benutzerdefinierte Optionen zur Fehlerbehandlung festlegen: den Wert verwerfen, einen bestimmten Wert festlegen oder eine bestimmte Fehlermeldung festlegen.
Auf Fehler in JSON prüfen	Auf eine Meldung auf Anwendungsebene prüfen, die sich am JSONPath befindet. Die Verarbeitung beenden, wenn die Prüfung erfolgreich ist und die Meldung nicht leer ist; andernfalls die Verarbeitung mit dem Wert fortsetzen, der vor diesem Vorverarbeitungsschritt vorhanden war. Beachten Sie, dass diese Fehler externer Dienste dem Benutzer unverändert gemeldet werden, ohne Informationen zum Vorverarbeitungsschritt hinzuzufügen. Beispiel: `$.errors`. Wenn ein JSON wie `{"errors":"e1"}` empfangen wird, wird der nächste Vorverarbeitungsschritt nicht ausgeführt. Wenn Sie das Kontrollkästchen *Bei Fehler benutzerdefiniert* aktivieren, können Sie benutzerdefinierte Optionen zur Fehlerbehandlung festlegen: den Wert verwerfen, einen bestimmten Wert festlegen oder eine bestimmte Fehlermeldung festlegen.
Auf Fehler in XML prüfen	Auf eine Meldung auf Anwendungsebene prüfen, die sich am XPath befindet. Die Verarbeitung beenden, wenn die Prüfung erfolgreich ist und die Meldung nicht leer ist; andernfalls die Verarbeitung mit dem Wert fortsetzen, der vor diesem Vorverarbeitungsschritt vorhanden war. Beachten Sie, dass diese Fehler externer Dienste dem Benutzer unverändert gemeldet werden, ohne Informationen zum Vorverarbeitungsschritt hinzuzufügen. Bei einem Fehler beim Parsen ungültiger XML-Daten wird kein Fehler gemeldet. Wenn Sie das Kontrollkästchen *Bei Fehler benutzerdefiniert* aktivieren, können Sie benutzerdefinierte Optionen zur Fehlerbehandlung festlegen: den Wert verwerfen, einen bestimmten Wert festlegen oder eine bestimmte Fehlermeldung festlegen.
Entspricht dem regulären Ausdruck	Geben Sie einen regulären Ausdruck an, dem ein Wert entsprechen muss. Wenn Sie das Kontrollkästchen *Bei Fehler benutzerdefiniert* aktivieren, können Sie benutzerdefinierte Optionen zur Fehlerbehandlung festlegen: den Wert verwerfen, einen bestimmten Wert festlegen oder eine bestimmte Fehlermeldung festlegen.
Drosselung	
Unveränderte Werte mit Heartbeat verwerfen	Einen Wert verwerfen, wenn er innerhalb des definierten Zeitraums (in Sekunden) nicht geändert wurde. Zur Angabe der Sekunden werden positive Ganzzahlen unterstützt (Minimum - 1 Sekunde). In diesem Feld können Zeitsuffixe verwendet werden (z. B. `30s`, `1m`, `2h`, `1d`). Benutzer-Makros und Low-Level-Erkennungs-Makros können in diesem Feld verwendet werden. Für einen Erkennungs-Datenpunkt kann nur eine Drosselungsoption angegeben werden. Beispiel: `1m`. Wenn innerhalb von 60 Sekunden zweimal identischer Text an diese Regel übergeben wird, wird er verworfen. *Hinweis*: Das Ändern von Datenpunkt-Prototypen setzt die Drosselung nicht zurück. Die Drosselung wird nur zurückgesetzt, wenn Vorverarbeitungsschritte geändert werden.
Prometheus	
Prometheus in JSON	Erforderliche Prometheus-Metriken in JSON konvertieren. Weitere Informationen finden Sie unter Prometheus-Prüfungen.

Beachten Sie, dass der Inhalt dieser Registerkarte schreibgeschützt ist, wenn die Erkennungsregel über eine Vorlage auf den Host angewendet wurde.

Benutzerdefinierte Makros

Auf der Registerkarte **LLD-Makros** können Sie benutzerdefinierte LLD-Makros angeben, wenn das zurückgegebene JSON noch keine Schlüssel im Format `{#MACRO}` enthält.

Beispielsweise gibt der Agent-Datenpunkt-Schlüssel `vfs.fs.get` JSON-Dateisystemdaten ohne vordefinierte LLD-Makros zurück.
Hier definieren Sie die Makros manuell und ordnen sie mithilfe von JSONPath-Ausdrücken Werten im JSON zu:



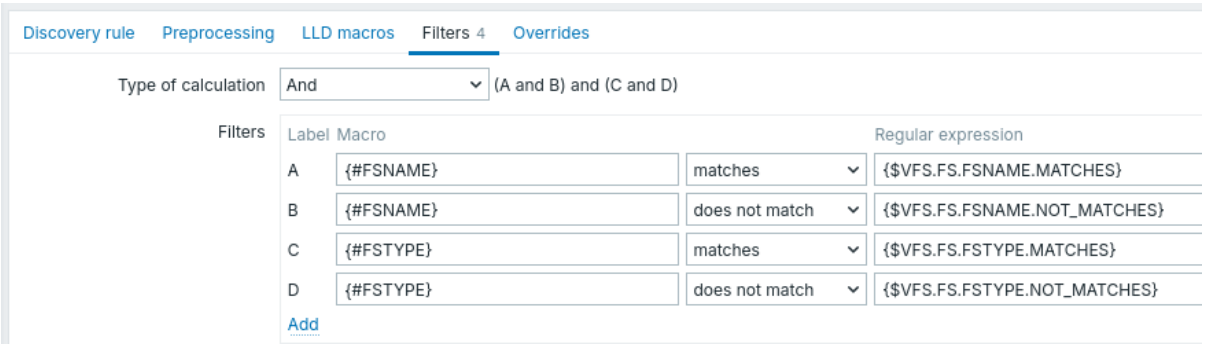
Parameter	Beschreibung
LLD macro	Name des LLD-Makros im Format `{#MACRO}`.
JSONPath	Ausdruck, der festlegt, welcher Wert aus jeder erkannten Entität im JSON-Array extrahiert werden soll. Beispielsweise extrahiert `$.foo` `"bar"` und `"baz"` aus `[{"foo":"bar"}, {"foo":"baz"}]`. JSONPath unterstützt sowohl Punkt- als auch Klammernotation. Die Klammernotation muss für Schlüssel mit Sonderzeichen oder Unicode verwendet werden, z. B. `$['unicode + special chars #1']['unicode + special chars #2']`.

Die Erkennungsregel muss ein JSON-Array auf der Wurzelebene (`$.`) zurückgeben.
Aus Gründen der Abwärtskompatibilität wird auch ein JSON-Objekt akzeptiert, das ein einzelnes `"data"`-Array enthält. In diesem Fall wird das `"data"`-Array automatisch entpackt:

```
{"data":[{"foo":"bar"}, {"foo":"baz"}]}
```

Filter

Ein Filter kann verwendet werden, um reale Datenpunkte, Auslöser und Graphen nur für Entitäten zu generieren, die den Kriterien entsprechen. Die Registerkarte **Filter** enthält Filterdefinitionen für Discovery-Regeln, mit denen Sie Discovery-Werte filtern können:



Parameter	Beschreibung
Berechnungstyp	Für die Berechnung von Filtern stehen folgende Optionen zur Verfügung: **Und** – alle Filter müssen erfüllt sein; **Oder** – es reicht aus, wenn ein Filter erfüllt ist; **Und/Oder** – verwendet *Und* mit unterschiedlichen Makronamen und *Oder* mit demselben Makronamen; **Benutzerdefinierter Ausdruck** – ermöglicht die Definition einer benutzerdefinierten Filterberechnung. Die Formel muss alle Filter in der Liste enthalten. Begrenzt auf 255 Zeichen.

Parameter	Beschreibung
Filter	Für Filterbedingungen stehen folgende Operatoren zur Verfügung: *stimmt überein*, *stimmt nicht überein*, *ist vorhanden*, *ist nicht vorhanden*. Die Operatoren *stimmt überein* und *stimmt nicht überein* erwarten einen Perl-kompatiblen regulären Ausdruck (PCRE). Sie können im Feld *Regulärer Ausdruck* einen regulären Ausdruck eingeben oder auf einen globalen regulären Ausdruck verweisen. Mit den Operatoren *ist vorhanden* und *ist nicht vorhanden* können Entitäten anhand des Vorhandenseins oder Fehlens des angegebenen LLD-Makros in der Antwort gefiltert werden. Beachten Sie, dass eine gefundene Entität ignoriert wird, wenn ein Makro aus dem Filter in der Antwort fehlt, es sei denn, für dieses Makro ist die Bedingung „ist nicht vorhanden“ angegeben. Eine Warnung wird angezeigt, wenn das Fehlen eines Makros das Ergebnis des Ausdrucks beeinflusst. Wenn beispielsweise `{#B}` in folgendem Ausdruck fehlt: `{#A}` stimmt mit 1 überein und `{#B}` stimmt mit 2 überein – es wird eine Warnung angezeigt `{#A}` stimmt mit 1 überein oder `{#B}` stimmt mit 2 überein – keine Warnung.

Wenn Sie beispielsweise nur an den Dateisystemen C:, D: und E: interessiert sind, können Sie `{#FSNAME}` in das Feld *Makro* und `^C|^D|^E` in das Feld *Regulärer Ausdruck* eingeben.
 Eine Filterung nach Dateisystemtypen ist ebenfalls mithilfe des Makros `{#FSTYPE}` möglich (z. B. `^ext|^reiserfs`) sowie nach Laufwerkstypen (nur vom Windows Agent unterstützt) mithilfe des Makros `{#FSDRIVETYPE}` (z. B. `fixed`).

Um einen regulären Ausdruck zu testen, können Sie beispielsweise `grep -E` verwenden:

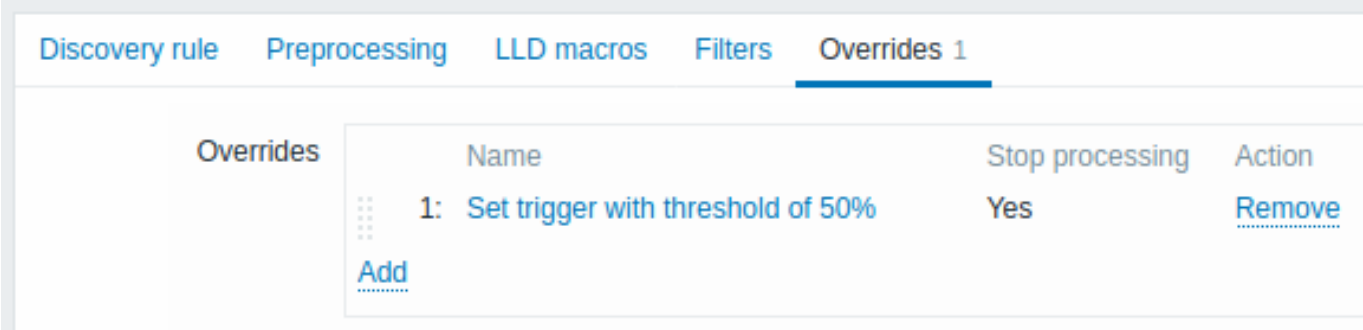
```
for f in ext2 nfs reiserfs smbfs;
do echo $f | grep -E '^ext|^reiserfs' || echo "SKIP: $f";
done
```

Warning:
Ein Fehler oder Tippfehler im regulären Ausdruck, der in der LLD-Regel verwendet wird (beispielsweise ein falscher regulärer Ausdruck für „Dateisysteme für Discovery“), kann zur Löschung von Tausenden Konfigurationselementen, historischen Werten und Ereignissen für viele Hosts führen.

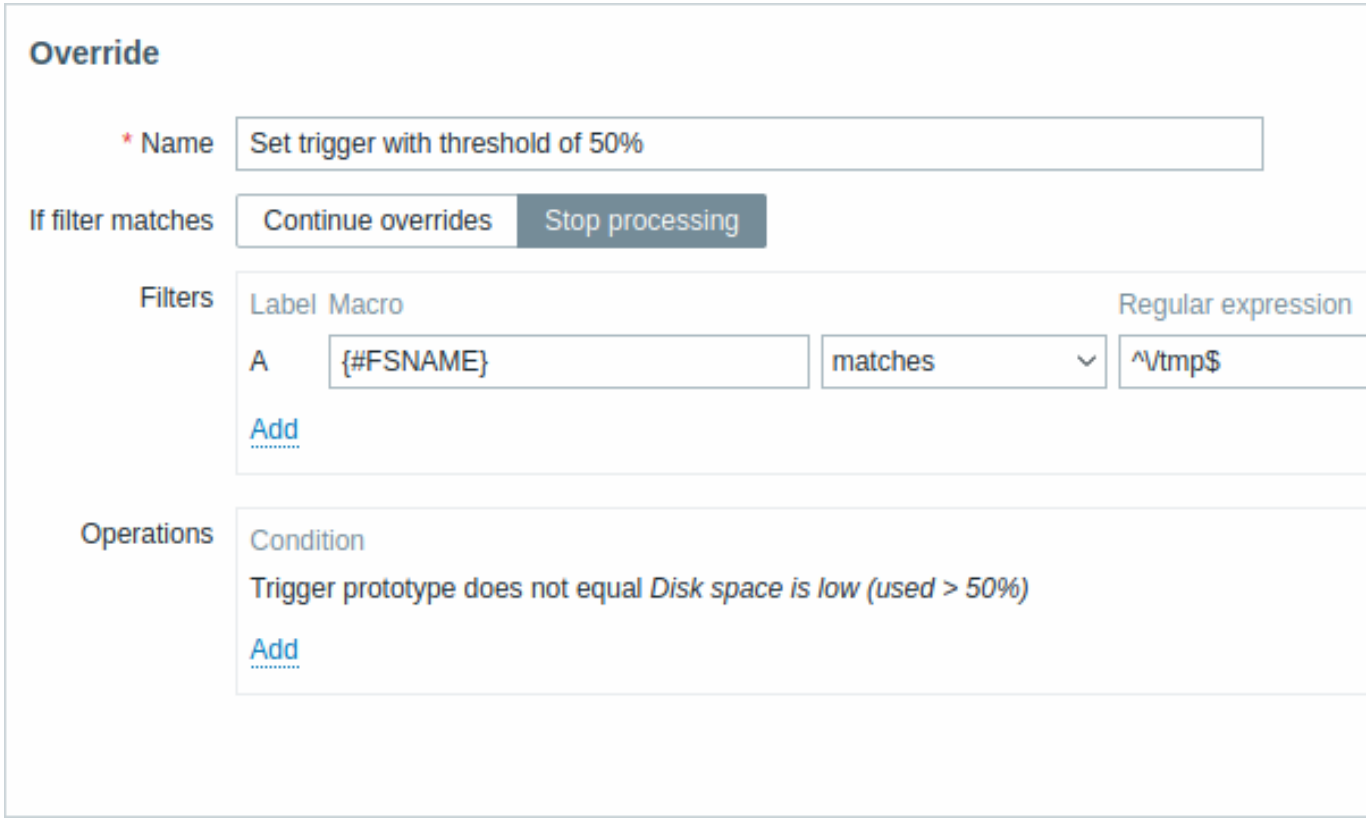
Attention:
Die Zabbix-Datenbank in MySQL muss mit Beachtung der Groß- und Kleinschreibung erstellt werden, damit Dateisystemnamen, die sich nur durch die Groß- und Kleinschreibung unterscheiden, korrekt erkannt werden können.

Überschreibungen

Auf der Registerkarte **Überschreibungen** können Regeln festgelegt werden, um die Liste der Datenpunkt-, Auslöser-, Graph-, Host- und Discovery-Prototypen oder deren Attribute für erkannte Objekte zu ändern, die bestimmte Kriterien erfüllen.



Überschreibungen (falls vorhanden) werden in einer per Drag-and-drop neu sortierbaren Liste angezeigt und in der Reihenfolge ausgeführt, in der sie definiert sind. Um die Details einer neuen Überschreibung zu konfigurieren, klicken Sie im Block *Überschreibungen* auf Add. Um eine vorhandene Überschreibung zu bearbeiten, klicken Sie auf den Namen der Überschreibung. Daraufhin wird ein Popup-Fenster geöffnet, in dem Sie die Details der Überschreibungsregel bearbeiten können.

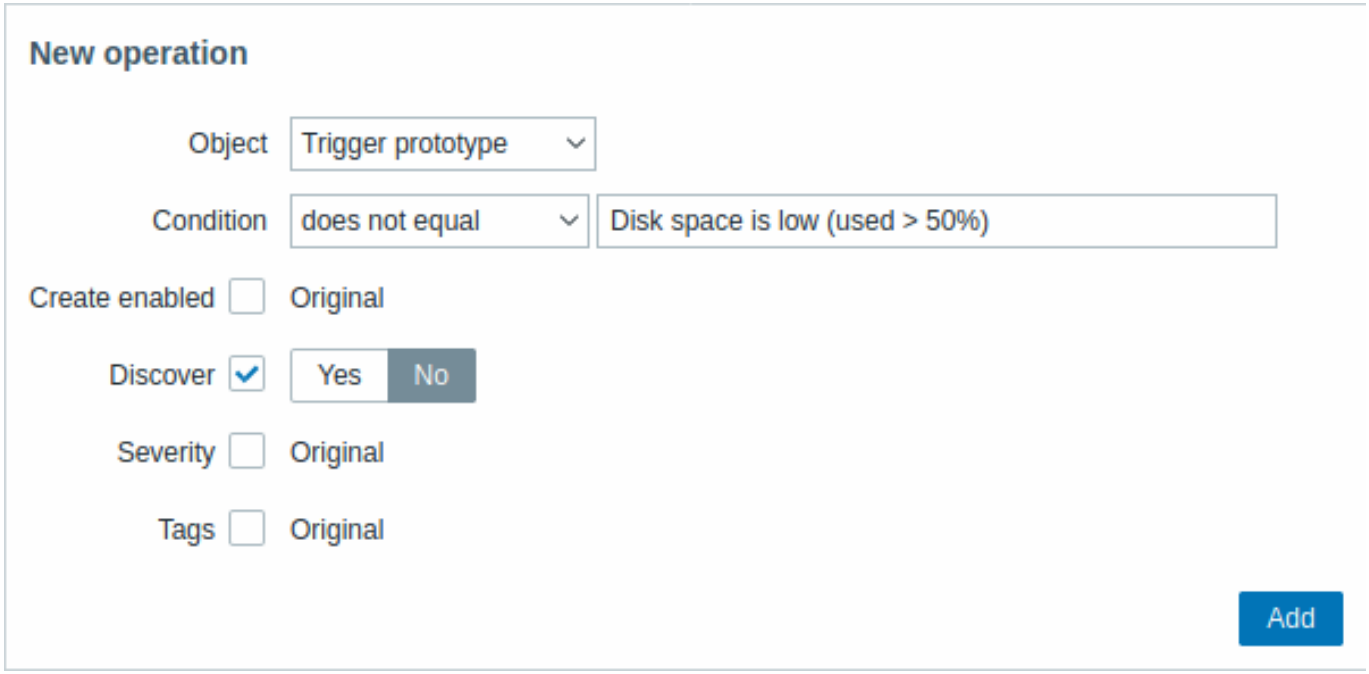


Alle Pflichtparameter sind mit roten Sternchen gekennzeichnet.

Parameter	Beschreibung
Name	Ein eindeutiger Name der Überschreibung (pro LLD-Regel).
Wenn Filter übereinstimmt	Legt fest, ob nachfolgende Überschreibungen verarbeitet werden sollen, wenn die Filterbedingungen erfüllt sind: **Überschreibungen fortsetzen** – nachfolgende Überschreibungen werden verarbeitet. **Verarbeitung beenden** – die Operationen der vorherigen Überschreibungen (falls vorhanden) und dieser Überschreibung werden ausgeführt; nachfolgende Überschreibungen werden für übereinstimmende LLD-Zeilen ignoriert.
Filter	Bestimmt, auf welche erkannten Entitäten die Überschreibung angewendet werden soll. Überschreibungsfilter werden nach den Filtern der Discovery-Regel verarbeitet und haben dieselbe Funktionalität.
Operationen	Überschreibungsoperationen werden mit folgenden Details angezeigt: **Bedingung** – ein Objekttyp und eine Bedingung, die für den Objektnamen erfüllt sein muss; zum Beispiel: Der Auslöser-Prototyp ist nicht gleich *Disk space is low (used > 50%)*. **Aktionen** – Links zum Bearbeiten und Entfernen einer Operation werden angezeigt.

Konfigurieren einer Operation

Um die Details einer neuen Operation zu konfigurieren, klicken Sie im Block „Operationen“ auf Add. Um eine vorhandene Operation zu bearbeiten, klicken Sie neben der Operation auf Edit. Daraufhin wird ein Popup-Fenster geöffnet, in dem Sie die Details der Operation bearbeiten können.



Parameter	Beschreibung
Objekt	Fünf Objekttypen sind verfügbar: Datenpunkt-Prototyp Auslöser-Prototyp Graph-Prototyp Host-Prototyp Discovery-Prototyp
Bedingung	Ermöglicht das Filtern der Entitäten, auf die die Operation angewendet werden soll.
Operator	Unterstützte Operatoren: **ist gleich** – auf diesen Prototyp anwenden **ist nicht gleich** – auf alle Prototypen außer diesen anwenden **enthält** – anwenden, wenn der Prototypname diese Zeichenfolge enthält **enthält nicht** – anwenden, wenn der Prototypname diese Zeichenfolge nicht enthält **entspricht** – anwenden, wenn der Prototypname dem regulären Ausdruck entspricht **entspricht nicht** – anwenden, wenn der Prototypname nicht dem regulären Ausdruck entspricht
Muster	Ein regulärer Ausdruck oder eine zu suchende Zeichenfolge.
Objekt: *Datenpunkt-Prototyp*	
Aktiviert erstellen	Wenn das Kontrollkästchen aktiviert ist, werden Schaltflächen angezeigt, mit denen Sie die ursprünglichen Einstellungen des Datenpunkt-Prototyps überschreiben können: *Ja* – der Datenpunkt wird in aktiviertem Zustand hinzugefügt. *Nein* – der Datenpunkt wird einer erkannten Entität hinzugefügt, jedoch in deaktiviertem Zustand.
Erkennen	Wenn das Kontrollkästchen aktiviert ist, werden Schaltflächen angezeigt, mit denen Sie die ursprünglichen Einstellungen des Datenpunkt-Prototyps überschreiben können: *Ja* – der Datenpunkt wird hinzugefügt. *Nein* – der Datenpunkt wird nicht hinzugefügt.
Aktualisierungsintervall	Wenn das Kontrollkästchen aktiviert ist, werden zwei Optionen angezeigt, mit denen Sie ein anderes Intervall für den Datenpunkt festlegen können: *Verzögerung* – Aktualisierungsintervall des Datenpunkts. Benutzermakros, LLD-Makros und Zeitsuffixe (z. B. 30s, 1m, 2h, 1d) werden unterstützt. Muss auf 0 gesetzt werden, wenn *Benutzerdefiniertes Intervall* verwendet wird. *Benutzerdefiniertes Intervall* – klicken Sie auf Add, um flexible/geplante Intervalle festzulegen. Weitere Informationen finden Sie unter Benutzerdefinierte Intervalle.

Parameter	Beschreibung
Historie	Wenn das Kontrollkästchen aktiviert ist, werden Schaltflächen angezeigt, mit denen Sie unterschiedliche Speicherzeiträume für die Historie des Datenpunkts festlegen können: *Nicht speichern* – wenn ausgewählt, wird die Historie nicht gespeichert. *Speichern bis* – wenn ausgewählt, wird rechts ein Eingabefeld zur Angabe des Speicherzeitraums angezeigt. Benutzermakros und LLD-Makros werden unterstützt.
Trends	Wenn das Kontrollkästchen aktiviert ist, werden Schaltflächen angezeigt, mit denen Sie unterschiedliche Speicherzeiträume für die Trends des Datenpunkts festlegen können: *Nicht speichern* – wenn ausgewählt, werden die Trends nicht gespeichert. *Speichern bis* – wenn ausgewählt, wird rechts ein Eingabefeld zur Angabe des Speicherzeitraums angezeigt. Benutzermakros und LLD-Makros werden unterstützt.
Tags	Wenn das Kontrollkästchen aktiviert ist, wird ein neuer Block angezeigt, in dem Sie Tag-Wert-Paare angeben können. Benutzermakros und LLD-Makros werden unterstützt. Diese Tags werden an die im Datenpunkt-Prototyp angegebenen Tags angehängt, auch wenn die Tagnamen übereinstimmen.
Objekt: *Auslöser-Prototyp*	
Aktiviert erstellen	Wenn das Kontrollkästchen aktiviert ist, werden Schaltflächen angezeigt, mit denen Sie die ursprünglichen Einstellungen des Auslöser-Prototyps überschreiben können: *Ja* – der Auslöser wird in aktiviertem Zustand hinzugefügt. *Nein* – der Auslöser wird einer erkannten Entität hinzugefügt, jedoch in deaktiviertem Zustand.
Erkennen	Wenn das Kontrollkästchen aktiviert ist, werden Schaltflächen angezeigt, mit denen Sie die ursprünglichen Einstellungen des Auslöser-Prototyps überschreiben können: *Ja* – der Auslöser wird hinzugefügt. *Nein* – der Auslöser wird nicht hinzugefügt.
Schweregrad	Wenn das Kontrollkästchen aktiviert ist, werden Schaltflächen für den Schweregrad des Auslösers angezeigt, mit denen Sie den Schweregrad des Auslösers ändern können.
Tags	Wenn das Kontrollkästchen aktiviert ist, wird ein neuer Block angezeigt, in dem Sie Tag-Wert-Paare angeben können. Benutzermakros und LLD-Makros werden unterstützt. Diese Tags werden an die im Auslöser-Prototyp angegebenen Tags angehängt, auch wenn die Tagnamen übereinstimmen.
Objekt: *Graph-Prototyp*	
Erkennen	Wenn das Kontrollkästchen aktiviert ist, werden Schaltflächen angezeigt, mit denen Sie die ursprünglichen Einstellungen des Graph-Prototyps überschreiben können: *Ja* – der Graph wird hinzugefügt. *Nein* – der Graph wird nicht hinzugefügt.
Objekt: *Host-Prototyp*	
Aktiviert erstellen	Wenn das Kontrollkästchen aktiviert ist, werden Schaltflächen angezeigt, mit denen Sie die ursprünglichen Einstellungen des Host-Prototyps überschreiben können: *Ja* – der Host wird in aktiviertem Zustand erstellt. *Nein* – der Host wird in deaktiviertem Zustand erstellt.
Erkennen	Wenn das Kontrollkästchen aktiviert ist, werden Schaltflächen angezeigt, mit denen Sie die ursprünglichen Einstellungen des Host-Prototyps überschreiben können: *Ja* – der Host wird erkannt. *Nein* – der Host wird nicht erkannt.
Vorlagen verknüpfen	Wenn das Kontrollkästchen aktiviert ist, wird ein Eingabefeld zur Angabe von Vorlagen angezeigt. Beginnen Sie mit der Eingabe des Vorlagennamens oder klicken Sie neben dem Feld auf *Auswählen* und wählen Sie Vorlagen aus der Liste in einem Popup-Fenster aus. Vorlagen aus dieser Überschreibung werden zu allen bereits mit dem Host-Prototyp verknüpften Vorlagen hinzugefügt.
Tags	Wenn das Kontrollkästchen aktiviert ist, wird ein neuer Block angezeigt, in dem Sie Tag-Wert-Paare angeben können. Benutzermakros und LLD-Makros werden unterstützt. Diese Tags werden an die im Host-Prototyp angegebenen Tags angehängt, auch wenn die Tagnamen übereinstimmen.

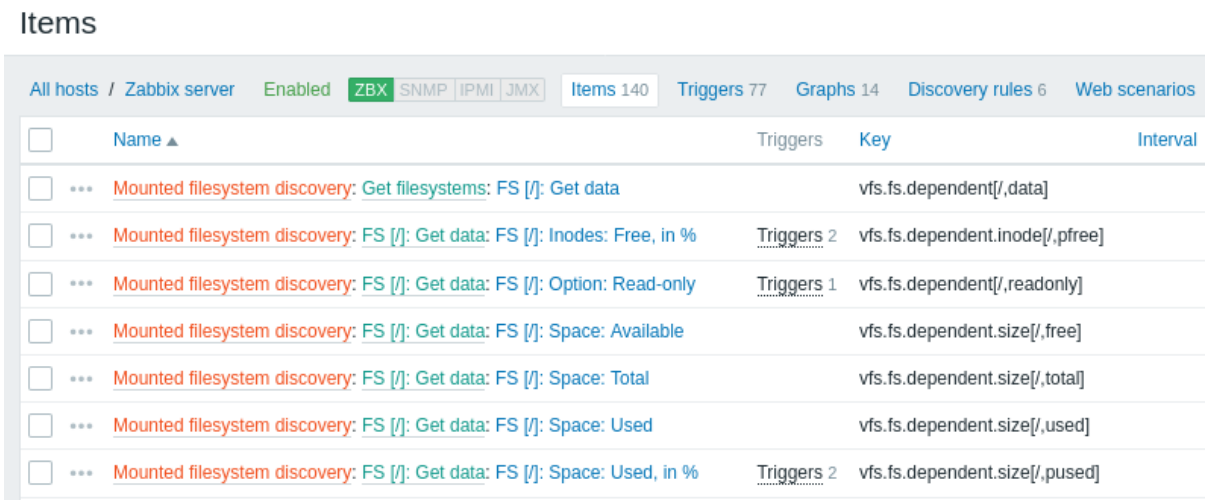
Parameter	Beschreibung
Hostinventar	Wenn das Kontrollkästchen aktiviert ist, werden Schaltflächen angezeigt, mit denen Sie einen anderen Modus für das Hostinventar des Host-Prototyps auswählen können: *Deaktiviert* – Hostinventar nicht ausfüllen *Manuell* – Details manuell angeben *Automatisiert* – Hostinventardaten automatisch anhand der erfassten Metriken ausfüllen.
Objekt: *Discovery-Prototyp*	
Aktiviert erstellen	Wenn das Kontrollkästchen aktiviert ist, werden Schaltflächen angezeigt, mit denen Sie die ursprünglichen Einstellungen des Discovery-Prototyps überschreiben können: *Ja* – die Discovery-Regel wird in aktiviertem Zustand hinzugefügt. *Nein* – die Discovery-Regel wird in deaktiviertem Zustand hinzugefügt.
Erkennen	Wenn das Kontrollkästchen aktiviert ist, werden Schaltflächen angezeigt, mit denen Sie die ursprünglichen Einstellungen des Discovery-Prototyps überschreiben können: *Ja* – die Discovery-Regel wird hinzugefügt. *Nein* – die Discovery-Regel wird nicht hinzugefügt.
Aktualisierungsintervall	Wenn das Kontrollkästchen aktiviert ist, werden zwei Optionen angezeigt, mit denen Sie ein anderes Intervall für die Discovery-Regel festlegen können: *Verzögerung* – Aktualisierungsintervall der Regel. Benutzermakros, LLD-Makros und Zeitsuffixe (z. B. 30s, 1m, 2h, 1d) werden unterstützt. Muss auf 0 gesetzt werden, wenn *Benutzerdefiniertes Intervall* verwendet wird. *Benutzerdefiniertes Intervall* – klicken Sie auf Add, um flexible/geplante Intervalle festzulegen. Weitere Informationen finden Sie unter Benutzerdefinierte Intervalle.

Formularschaltflächen

Mit den Schaltflächen am unteren Rand des Formulars können Sie mehrere Vorgänge ausführen.

Add	Eine Discovery-Regel hinzufügen. Diese Schaltfläche ist nur für neue Discovery-Regeln verfügbar.
Update	Die Eigenschaften einer Discovery-Regel aktualisieren. Diese Schaltfläche ist nur für vorhandene Discovery-Regeln verfügbar.
Clone	Eine weitere Discovery-Regel basierend auf den Eigenschaften der aktuellen Discovery-Regel erstellen.
Execute now	Die Discovery basierend auf der Discovery-Regel sofort durchführen. Die Discovery-Regel muss bereits vorhanden sein. Weitere Informationen finden Sie hier. *Hinweis*: Bei einer sofortigen Discovery wird der Konfigurations-Cache nicht aktualisiert. Daher spiegelt das Ergebnis sehr aktuelle Änderungen an der Konfiguration der Discovery-Regel möglicherweise nicht wider.
Test	Die Konfiguration der Discovery-Regel testen. Verwenden Sie diese Schaltfläche, um die Konfigurationseinstellungen (z. B. Konnektivität und Korrektheit der Parameter) zu überprüfen, ohne Änderungen dauerhaft anzuwenden.
Delete	Die Discovery-Regel löschen.
Cancel	Die Bearbeitung der Eigenschaften der Discovery-Regel abbrechen.

Erkannte Entitäten Die folgenden Screenshots veranschaulichen, wie erkannte Datenpunkte, Auslöser und Diagramme in der Konfiguration des Hosts aussehen. Erkannte Entitäten sind mit einem orangefarbenen Link zu der Discovery-Regel versehen, aus der sie stammen.



Beachten Sie, dass erkannte Entitäten nicht erstellt werden, wenn bereits vorhandene Entitäten mit denselben Eindeutigkeitskriterien existieren, zum Beispiel ein Datenpunkt mit demselben Schlüssel oder ein Diagramm mit demselben Namen. In diesem Fall wird im Frontend eine Fehlermeldung angezeigt, dass die Low-Level-Discovery-Regel bestimmte Entitäten nicht erstellen konnte. Die Discovery-Regel selbst wird jedoch nicht auf „nicht unterstützt“ gesetzt, nur weil eine Entität nicht erstellt werden konnte und übersprungen werden musste. Die Discovery-Regel fährt mit dem Erstellen/Aktualisieren anderer Entitäten fort.

Wenn eine erkannte Entität (Host, Dateisystem, Schnittstelle usw.) nicht mehr erkannt wird (oder den Filter nicht mehr erfüllt), können die auf ihrer Grundlage erstellten Entitäten automatisch deaktiviert und schließlich gelöscht werden.

Verlorene Ressourcen können basierend auf dem Wert des Parameters *Disable lost resources* automatisch deaktiviert werden. Dies betrifft verlorene Hosts, Datenpunkte und Auslöser.

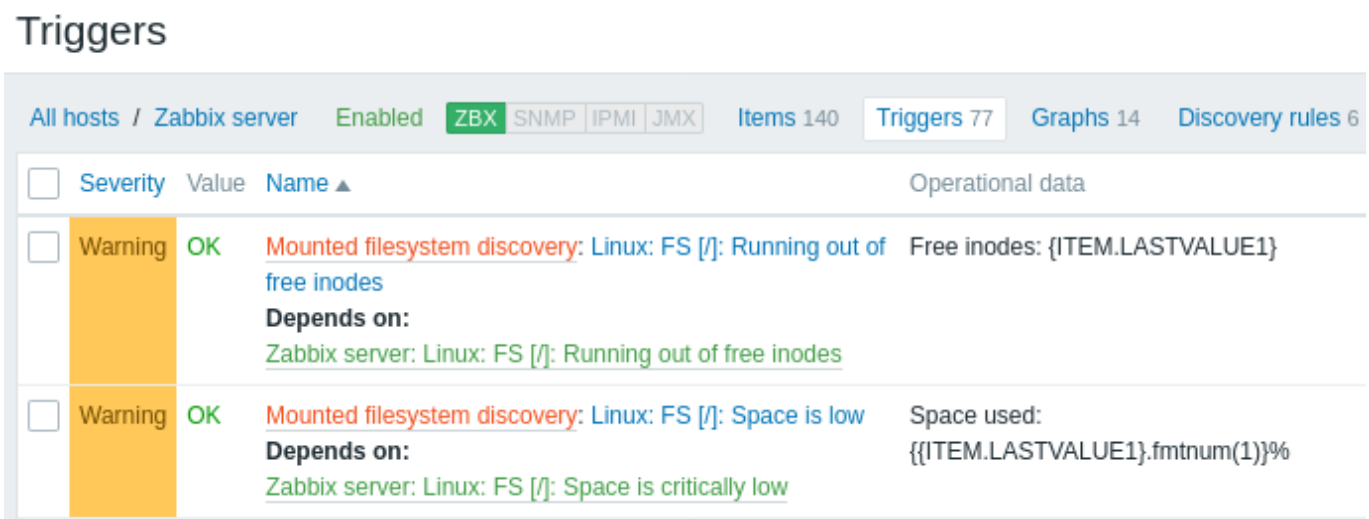
Verlorene Ressourcen können basierend auf dem Wert des Parameters *Delete lost resources* automatisch gelöscht werden. Dies betrifft verlorene Hosts, Hostgruppen, Datenpunkte, Auslöser und Diagramme.

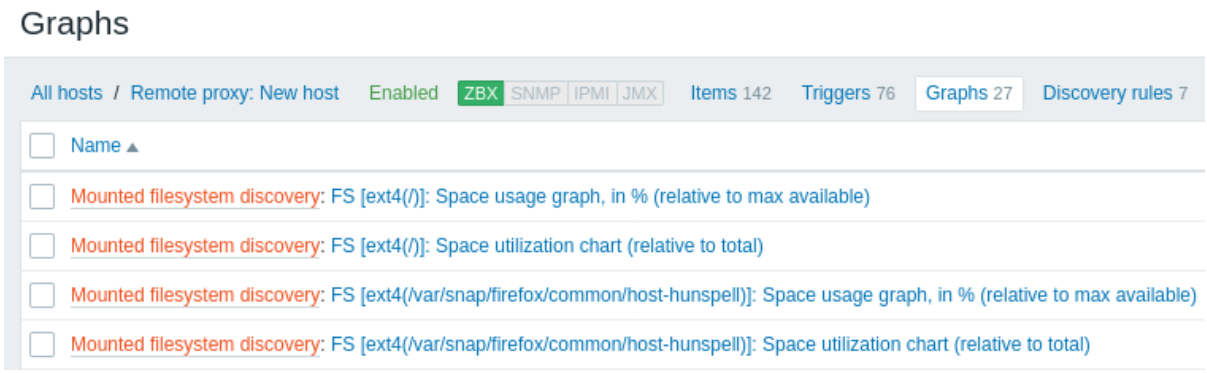
Wenn erkannte Entitäten den Status „Nicht mehr erkannt“ erhalten, wird in der Entitätenliste ein Lebensdauerindikator angezeigt. Bewegen Sie den Mauszeiger darüber, und es wird eine Meldung mit Details zu ihrem Status angezeigt.



Wenn Entitäten zum Löschen markiert wurden, aber nicht zum erwarteten Zeitpunkt gelöscht wurden (deaktivierte Discovery-Regel oder Datenpunkt-Host), werden sie beim nächsten Verarbeiten der Discovery-Regel gelöscht.

Entitäten, die andere Entitäten enthalten und zum Löschen markiert sind, werden nicht aktualisiert, wenn sie auf Ebene der Discovery-Regel geändert werden. Zum Beispiel werden LLD-basierte Auslöser nicht aktualisiert, wenn sie Datenpunkte enthalten, die zum Löschen markiert sind.





Andere Arten der Discovery Weitere Details und Anleitungen zu anderen Arten der sofort einsatzbereiten Discovery finden Sie in den folgenden Abschnitten:

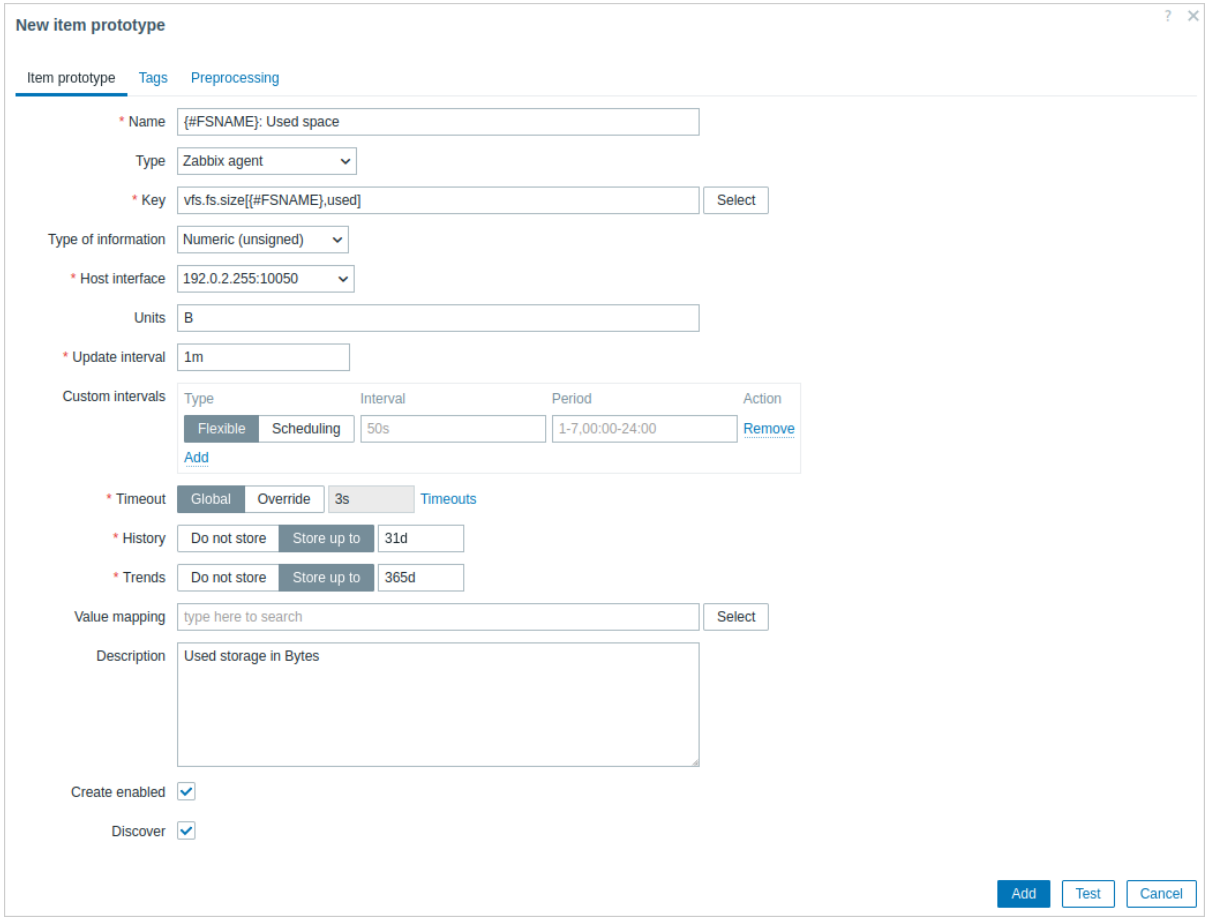
- Discovery von Netzwerkschnittstellen
- Discovery von CPUs und CPU-Kernen
- Discovery von SNMP-OIDs
- Discovery von JMX-Objekten;
- Discovery mit ODBC-SQL-Abfragen
- Discovery von Windows-Diensten
- Discovery von Host-Schnittstellen in Zabbix

Weitere Details zum JSON-Format für Discovery-Datenpunkte sowie ein Beispiel dafür, wie Sie Ihren eigenen Dateisystem-Discoverer als Perl-Skript implementieren, finden Sie unter Erstellen benutzerdefinierter LLD-Regeln.

Datenpunkt-Prototypen

Nachdem eine Regel erstellt wurde, wechseln Sie zu den Datenpunkten dieser Regel und klicken Sie auf *Datenpunkt-Prototyp erstellen*, um einen Datenpunkt-Prototyp zu erstellen.

Beachten Sie, wie das Makro `{#FSNAME}` verwendet wird, wenn ein Dateisystemname erforderlich ist. Die Verwendung eines Makros für die Low-Level-Erkennung im Datenpunktschlüssel ist obligatorisch, um sicherzustellen, dass die Erkennung korrekt verarbeitet wird. Bei der Verarbeitung der Erkennungsregel wird dieses Makro durch das erkannte Dateisystem ersetzt.



Note:
Wenn keine Parameter für den Datenpunktschlüssel verwendet werden, platzieren Sie das LLD-Makro innerhalb der Datenpunktschlüsselklammern `[...]` (zum Beispiel `v_[{#MACRO}]`).

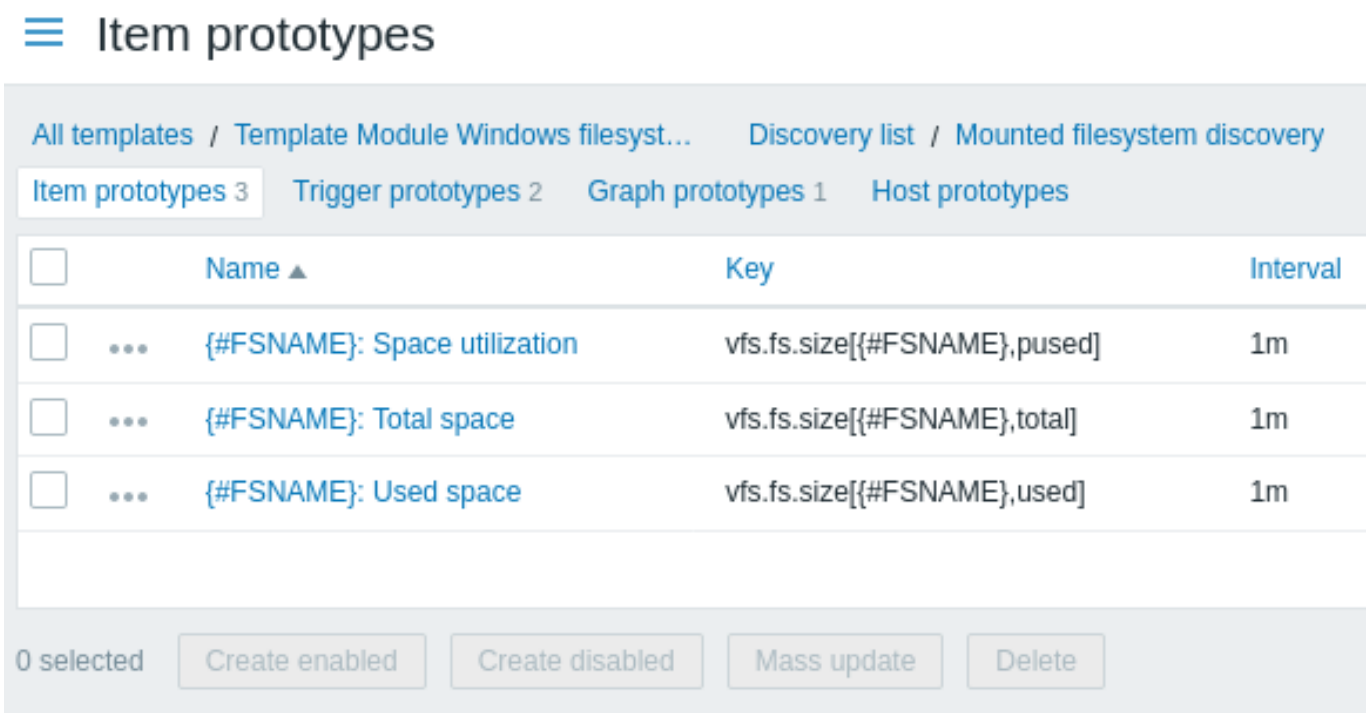
Makros für die Low-Level-Erkennung macros und Benutzer-Makros werden in der Konfiguration von Datenpunkt-Prototypen und in den Parametern der Vorverarbeitung von Datenpunktwerten unterstützt. Beachten Sie, dass bei der Verwendung in Aktualisierungsintervallen ein einzelnes Makro das gesamte Feld ausfüllen muss. Mehrere Makros in einem Feld oder Makros zusammen mit Text werden nicht unterstützt.

Note:
Eine kontextspezifische Maskierung von Makros für die Low-Level-Erkennung wird für die sichere Verwendung in Parametern der Vorverarbeitung mit regulären Ausdrücken und XPath durchgeführt.

Attribute, die spezifisch für Datenpunkt-Prototypen sind:

Parameter	Beschreibung
Erstellung aktiviert	Wenn aktiviert, wird der Datenpunkt im aktivierten Zustand hinzugefügt. Wenn deaktiviert, wird der Datenpunkt einer erkannten Entität hinzugefügt, jedoch im deaktivierten Zustand.
Erkennen	Wenn aktiviert (Standard), wird der Datenpunkt einer erkannten Entität hinzugefügt. Wenn deaktiviert, wird der Datenpunkt keiner erkannten Entität hinzugefügt, es sei denn, diese Einstellung wird in der Erkennungsregel überschrieben.

Wir können für jede Dateisystemmetrik, die uns interessiert, mehrere Datenpunkt-Prototypen erstellen:



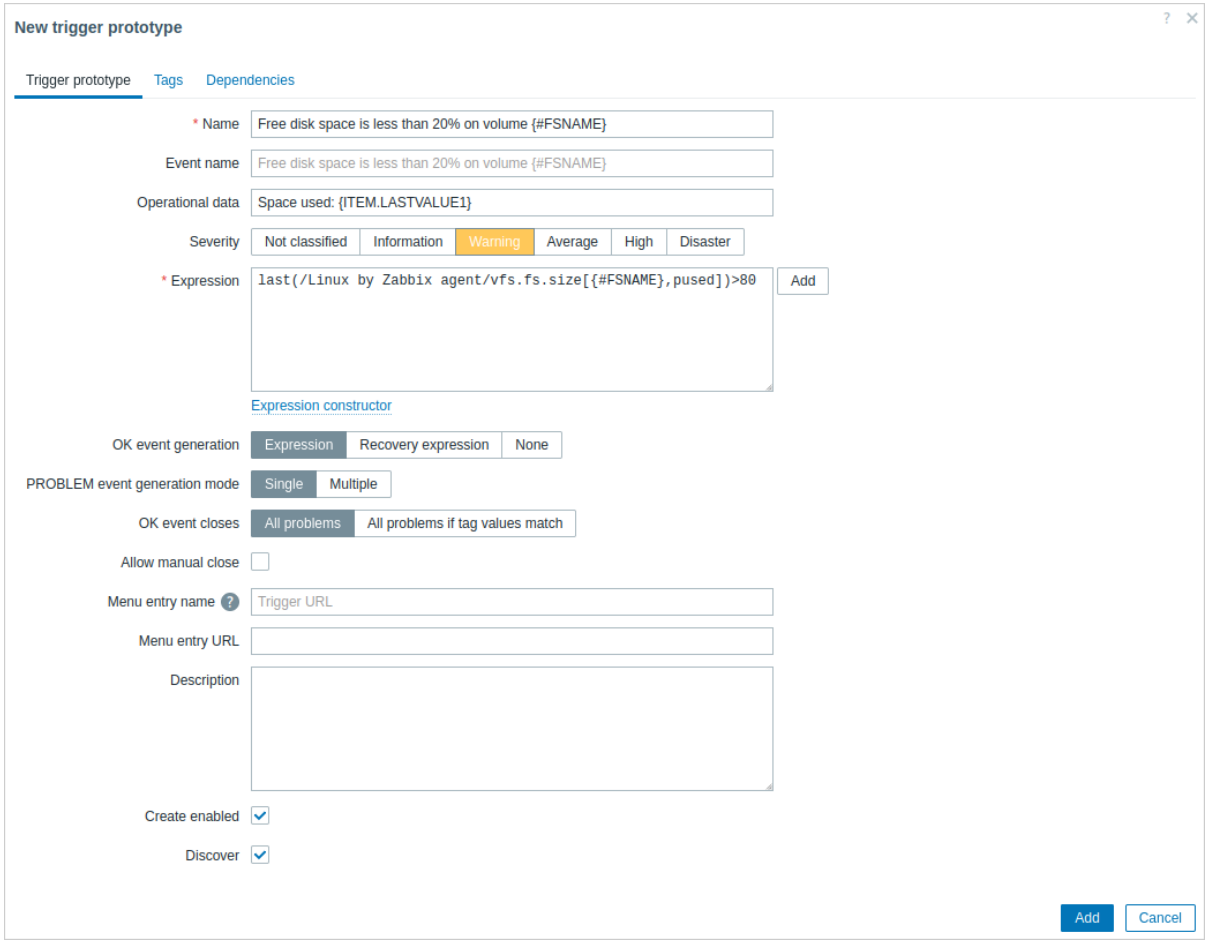
Klicken Sie auf das Symbol mit den drei Punkten, um das Menü für den jeweiligen Datenpunkt-Prototyp mit den folgenden Optionen zu öffnen:

- *Auslöser-Prototyp erstellen* - einen Auslöser-Prototyp basierend auf diesem Datenpunkt-Prototyp erstellen
- *Auslöser-Prototypen* - klicken, um eine Liste mit Links zu den bereits konfigurierten Auslöser-Prototypen dieses Datenpunkt-Prototyps anzuzeigen
- *Abhängigen Datenpunkt erstellen* - einen abhängigen Datenpunkt für diesen Datenpunkt-Prototyp erstellen

Die Option *Massenaktualisierung* ist verfügbar, wenn Sie die Eigenschaften mehrerer Datenpunkt-Prototypen gleichzeitig aktualisieren möchten.

Auslöser-Prototypen

Wir erstellen Auslöser-Prototypen auf ähnliche Weise wie Datenpunkt-Prototypen:

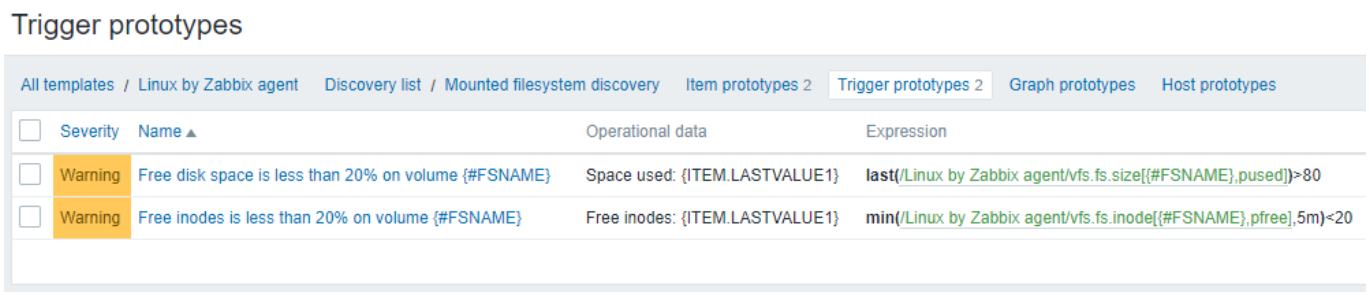


Attribute, die speziell für Auslöser-Prototypen gelten:

Parameter	Beschreibung
Create enabled	Wenn diese Option aktiviert ist, wird der Auslöser im aktivierten Zustand hinzugefügt. Wenn sie deaktiviert ist, wird der Auslöser zu einem erkannten Objekt hinzugefügt, jedoch im deaktivierten Zustand.
Discover	Wenn diese Option aktiviert ist (Standard), wird der Auslöser zu einem erkannten Objekt hinzugefügt. Wenn sie deaktiviert ist, wird der Auslöser nicht zu einem erkannten Objekt hinzugefügt, es sei denn, diese Einstellung wird in der Erkennungsregel überschrieben.

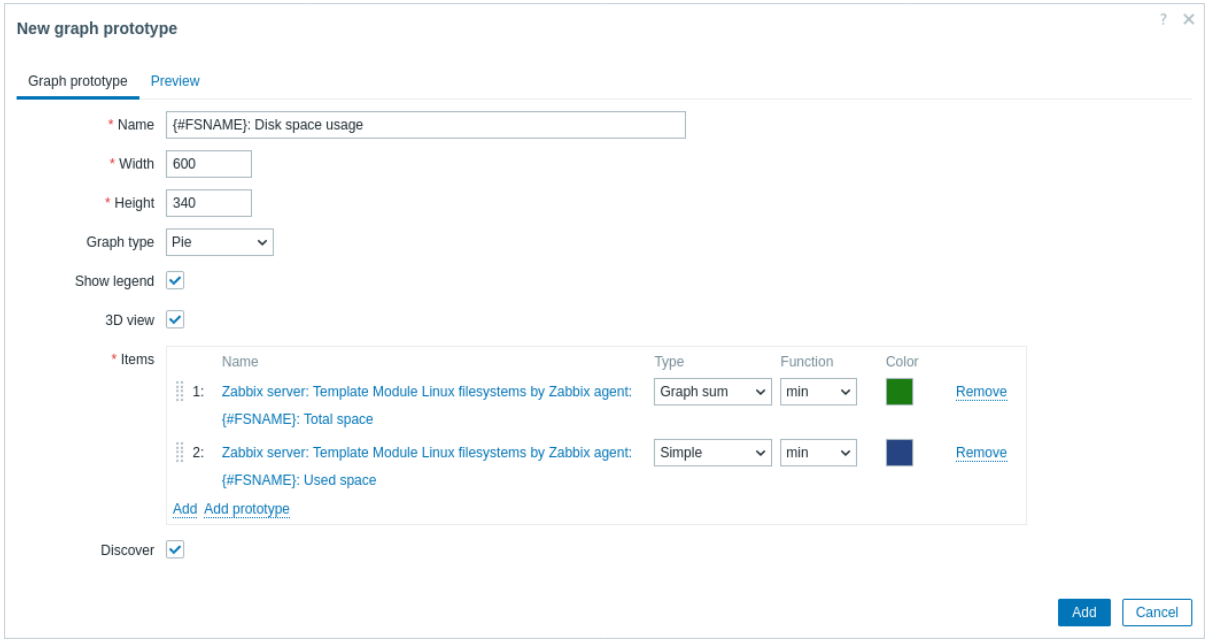
Wenn aus den Prototypen echte Auslöser erstellt werden, kann es erforderlich sein, flexibel zu sein, welcher konstante Wert ('20' in unserem Beispiel) für den Vergleich in der Ausdruck verwendet wird. Sehen Sie, wie Benutzer-Makros mit Kontext nützlich sein können, um diese Flexibilität zu erreichen.

Sie können Abhängigkeiten zwischen Auslöser-Prototypen definieren. Gehen Sie dazu auf die Registerkarte *Abhängigkeiten*. Ein Auslöser-Prototyp kann von einem anderen Auslöser-Prototyp aus derselben Low-Level-Discovery-(LLD)-Regel oder von einem regulären Auslöser abhängen. Ein Auslöser-Prototyp darf nicht von einem Auslöser-Prototyp aus einer anderen LLD-Regel oder von einem aus einem Auslöser-Prototyp erstellten Auslöser abhängen. Ein Host-Auslöser-Prototyp kann nicht von einem Auslöser aus einer Vorlage abhängen.



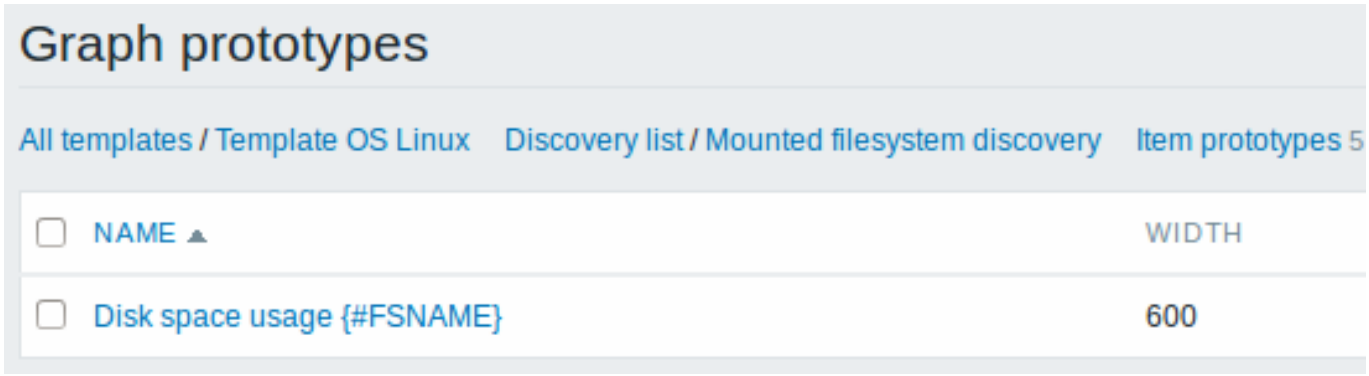
Diagrammvorlagen

Wir können auch Diagrammvorlagen erstellen:

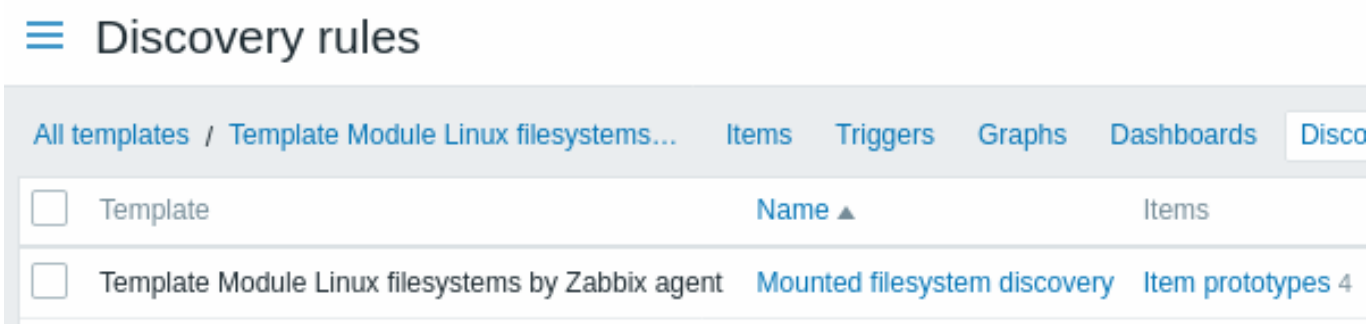


Attribute, die speziell für Diagrammvorlagen gelten:

Parameter	Description
Discover	Wenn aktiviert (Standard), wird das Diagramm zu einer erkannten Entität hinzugefügt. Wenn deaktiviert, wird das Diagramm nicht zu einer erkannten Entität hinzugefügt, es sei denn, diese Einstellung wird in der Erkennungsregel überschrieben.



Schließlich haben wir eine Erkennungsregel erstellt, die wie unten dargestellt aussieht. Sie hat fünf Datenpunktvorlagen, zwei Auslöservorlagen und eine Diagrammvorlage.



Host-Prototypen

Host-Prototypen sind Vorlagen für die Erstellung von Hosts durch Regeln der Low-Level-Discovery. Bevor sie als Hosts erkannt werden, können diese Prototypen keine Datenpunkte und Auslöser haben, außer denjenigen, die aus Vorlagen verknüpft sind.

Konfiguration

Host-Vorlagen werden unter Low-Level-Discovery-Regeln konfiguriert.

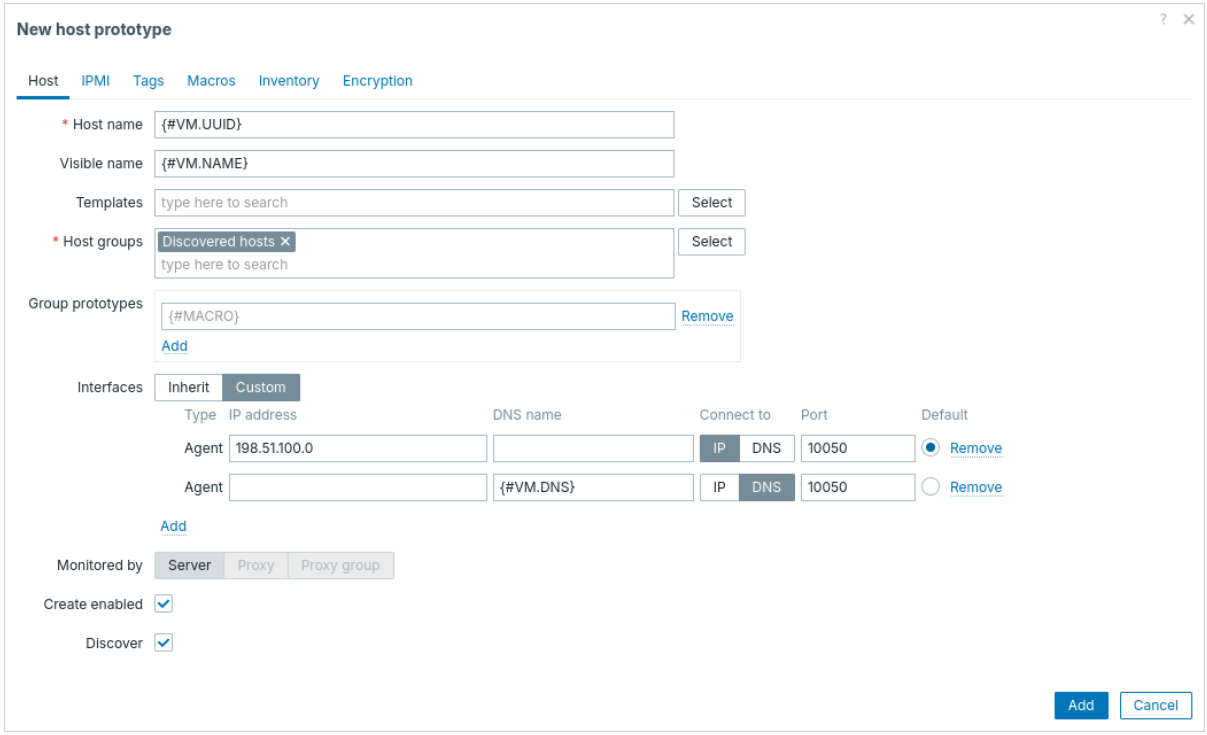
Um eine Host-Vorlage zu erstellen:

1. Gehen Sie zu *Datenerfassung > Hosts*.

2. Klicken Sie für den gewünschten Host auf *Discovery*, um zur Liste der für diesen Host konfigurierten Low-Level-Discovery-Regeln zu wechseln.

3. Klicken Sie für die gewünschte Discovery-Regel auf *Host-Vorlagen*.

4. Klicken Sie oben rechts auf die Schaltfläche *Host-Vorlage erstellen* und füllen Sie das Konfigurationsformular aus.



Alle Pflichtfelder sind mit einem roten Sternchen markiert.

Parameter	Beschreibung
Host name	Dieser Parameter muss mindestens ein Low-Level-Discovery-Makro enthalten, um eindeutige Host-Namen für erstellte Hosts sicherzustellen. Low-Level-Discovery-Makros werden unterstützt.
Visible name	
Group prototypes	Ermöglicht die Angabe von Hostgruppen-Vorlagen mithilfe von Low-Level-Discovery-Makros. Auf Grundlage der angegebenen Gruppen-Vorlagen werden Hostgruppen entdeckt, erstellt und mit den erstellten Hosts verknüpft; entdeckte Gruppen, die bereits von anderen Low-Level-Discovery-Regeln erstellt wurden, werden ebenfalls mit den erstellten Hosts verknüpft. Entdeckte Hostgruppen, die mit manuell erstellten Hostgruppen übereinstimmen, werden jedoch nicht mit den erstellten Hosts verknüpft.
Interfaces	Legen Sie fest, ob entdeckte Hosts die IP vom Host erben, zu dem die Discovery-Regel gehört (Standard), oder benutzerdefinierte Schnittstellen erhalten. Low-Level-Discovery-Makros und Benutzermakros werden unterstützt.
Monitored by	Dieses Feld wird automatisch mit dem entsprechenden Feld in der Host-Konfiguration synchronisiert.
Create enabled	Legen Sie den Status der entdeckten Hosts fest; wenn diese Option nicht aktiviert ist, werden Hosts als deaktiviert erstellt.
Discover	Legen Sie fest, ob Hosts aus der Host-Vorlage erstellt werden; wenn diese Option nicht aktiviert ist, werden keine Hosts aus der Host-Vorlage erstellt (es sei denn, diese Einstellung wird in der Low-Level-Discovery-Regel überschrieben).

Note:
Low-Level-Discovery-Makros werden auch für Tag-Werte und Benutzermakro-Werte von Host-Vorlagen unterstützt.
Value maps werden für Host-Vorlagen nicht unterstützt.

Ein Beispiel zur Konfiguration einer Host-Vorlage finden Sie unter *Überwachung virtueller Maschinen*.

Host-Schnittstellen

Um benutzerdefinierte Schnittstellen hinzuzufügen, wechseln Sie den Selektor *Schnittstellen* von „Vererben“ zu „Benutzerdefiniert“. Klicken Sie auf Add und wählen Sie den Schnittstellentyp aus – Zabbix Agent, SNMP, JMX, IPMI.

Note:
Wenn *Benutzerdefiniert* ausgewählt ist, aber keine Schnittstellen festgelegt wurden, werden die Hosts ohne Schnittstellen erstellt.
 Wenn *Vererben* ausgewählt ist und der Host-Prototyp zu einer Vorlage gehört, übernehmen alle erkannten Hosts die Host-Schnittstelle von dem Host, mit dem die Vorlage verknüpft ist.

Wenn mehrere benutzerdefinierte Schnittstellen angegeben sind, kann die primäre Schnittstelle in der Spalte *Standard* festgelegt werden.

Ein Beispiel für die Konfiguration einer benutzerdefinierten Host-Schnittstelle finden Sie unter *Beispiel für die Einrichtung der VMware-Überwachung*.

Warning:
Ein Host wird nur erstellt, wenn eine Host-Schnittstelle korrekte Daten enthält.

Entdeckte Hosts

In der Host-Liste werden entdeckte Hosts mit dem Namen der Discovery-Regel, die sie erstellt hat, als Präfix versehen.

Entdeckte Hosts übernehmen die meisten Parameter von Host-Prototypen als *schreibgeschützt*. Nur die folgenden Parameter entdeckter Hosts können konfiguriert werden:

- *Vorlagen* - neue Vorlagen verknüpfen oder manuell hinzugefügte Verknüpfungen aufheben. Von einem Host-Prototyp geerbte Vorlagen können nicht getrennt werden.
- *Beschreibung* - die Host-Beschreibung hinzufügen oder bearbeiten.
- *Status* - den Host aktivieren oder deaktivieren.
- *Tags* - neue Tags hinzufügen oder manuell hinzugefügte entfernen. Von einem Host-Prototyp geerbte Tags können nicht entfernt werden. Beachten Sie, dass Tags keine Duplikate haben können (Tags mit demselben Namen und Wert). Wenn dem Host-Prototyp ein neues Tag hinzugefügt wird und es mit einem manuell hinzugefügten Tag auf einem entdeckten Host übereinstimmt, wird das manuell hinzugefügte Tag bei der Discovery durch das geerbte Tag ersetzt.
- *Makros* - neue Host-Makros hinzufügen oder manuell hinzugefügte sowie von einem Host-Prototyp geerbte entfernen; Makrowerte und Typen ändern.

Note:
Entdeckte Hosts übernehmen außerdem Benutzermakros von dem Host, auf dem die Discovery-Regel konfiguriert ist. Diese geerbten Makros können entfernt sowie ihre Werte und Typen geändert werden.

Entdeckte Hosts können auch andere Hosts entdecken. Wenn beispielsweise ein Host-Prototyp (der zum Entdecken von Hypervisoren verwendet wird) mit einer Vorlage verknüpft ist, die eine Low-Level-Discovery-Regel mit einem eigenen Host-Prototyp enthält (der zum Entdecken virtueller Maschinen verwendet wird), entdeckt Zabbix sowohl Hypervisoren als auch deren virtuelle Maschinen. Um zusätzlich Container auf diesen VMs zu entdecken, können Sie Low-Level-Discovery-Regeln erstellen oder eine neue Vorlage mit Host-Prototypen mit den entdeckten VMs verknüpfen oder den VM-Host-Prototyp vorab konfigurieren, indem Sie eine Vorlage verknüpfen, die selbst Host-Prototypen enthält.

Entdeckte Hosts können manuell gelöscht werden. Beachten Sie jedoch, dass sie erneut entdeckt werden, wenn die Discovery für sie aktiviert ist.

Hosts, die nicht mehr entdeckt werden, können:

- automatisch deaktiviert werden (basierend auf dem Wert *Verlorene Ressourcen deaktivieren* der Discovery-Regel)
- automatisch gelöscht werden (basierend auf dem Wert *Verlorene Ressourcen löschen* der Discovery-Regel).

Discovery-Prototypen

Übersicht

Discovery-Prototypen sind **verschachtelte Low-Level-Discovery-Regeln** innerhalb einer übergeordneten Discovery-Regel. Sie ermöglichen die mehrstufige Discovery von Objekten mit eigenen Datenpunkten, Auslösern usw. Beispielsweise können Sie alle Datenbankinstanzen auf einem Datenbankserver ermitteln, anschließend für jede Instanz Tablespaces und danach für jeden Tablespace Tabellen ermitteln.

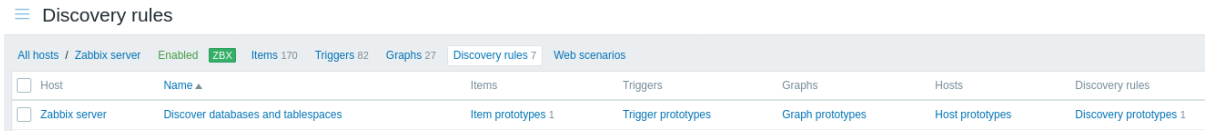
Discovery-Prototypen verfügen über eigene Datenpunkt-, Auslöser-, Graph-, Host- und Discovery-Prototypen. Ein verschachtelter Discovery-Prototyp verwendet denselben JSON-Wert wie die übergeordnete Regel, wenn Sie den Typ *Nested* angeben.

Die Verschachtelungsebenen für Discovery-Prototypen sind unbegrenzt.

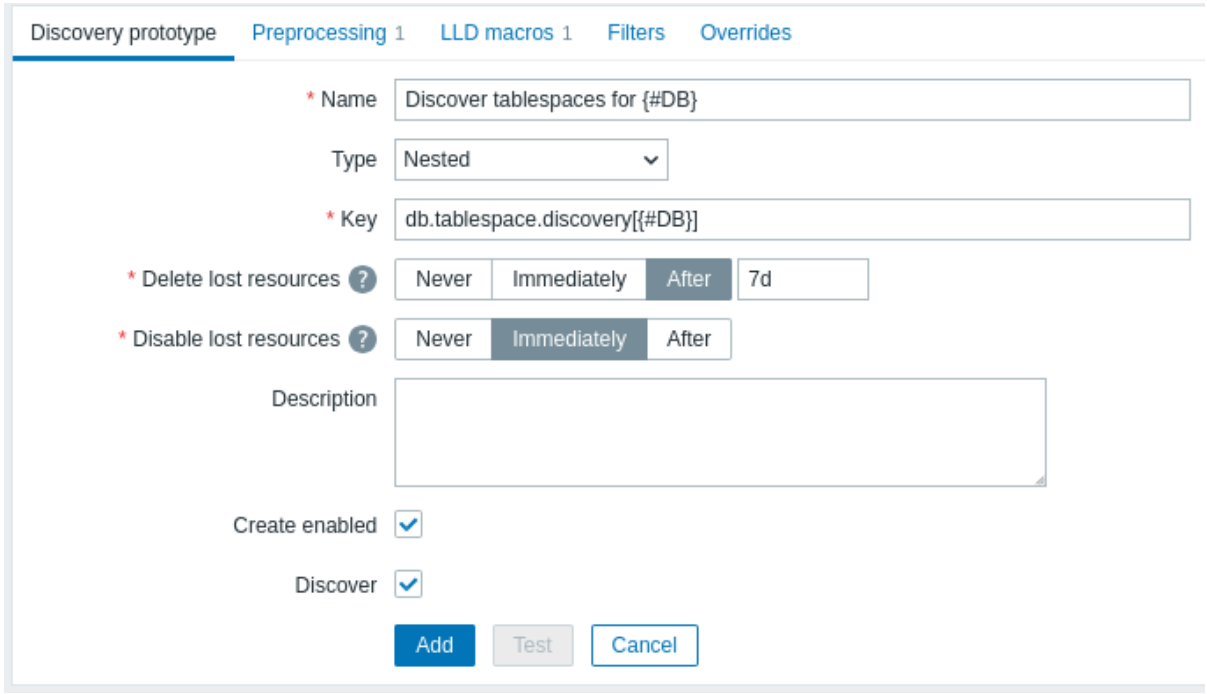
Konfiguration

Um eine Erkennungsvorlage zu erstellen:

- Klicken Sie in der Zeile einer vorhandenen Erkennungsregel auf *Erkennungsvorlagen*



- Klicken Sie auf *Erkennungsvorlage erstellen*



Die Konfigurationsfelder dieses Formulars werden mit der regulären Low-Level-Erkennung gemeinsam genutzt.

Wenn Sie im geöffneten Formular der Erkennungsvorlage *Verschachtelt* als *Typ* auswählen, werden Erkennungsregeln (aus der Erkennungsvorlage) auf Grundlage eines JSON-Objekts aus demselben JSON-Wert wie die übergeordnete Erkennungsregel generiert. Wenn das ursprüngliche JSON beispielsweise [<object A>, <object B>] ist und es eine verschachtelte Erkennungsvorlage gibt, dann werden zwei Erkennungsregeln generiert, die jeweils auf den Daten von Objekt A bzw. Objekt B basieren.

In diesem Fall wird die Erkennungsvorlage gleichzeitig mit der übergeordneten Regel aktiviert. Die verschachtelte Regel kann daher Vorverarbeitung verwenden, um mit einem anderen "Ausschnitt" derselben Daten zu arbeiten, die bereits von der übergeordneten Regel erfasst wurden.

LLD-Makros aus der übergeordneten LLD-Regel sind für verschachtelte Erkennungsregeln verfügbar.

Verschachtelte LLD-Regeln auf erkannten Hosts

Eine *verschachtelte* Low-Level-Discovery-Regel kann auf einer Host-Vorlage verwendet werden, die einem Host-Prototyp zugewiesen ist. Wenn auf einem erkannten Host eine *verschachtelte* Discovery-Regel vorhanden ist, wird das JSON-Objekt, das zur Erkennung des Hosts verwendet wurde, auch an alle LLD-Regeln des verschachtelten Typs auf diesem Host gesendet. Weitere Details finden Sie im Beispiel.

LLD-Makros aus der Discovery-Regel, die den Host erstellt hat, sind für verschachtelte Discovery-Regeln verfügbar.

Beispiel

Veranschaulichen wir die mögliche Anwendung von Discovery-Prototypen anhand des Empfangs des folgenden Beispiels für mehrstufiges JSON.

```
[
  {
    "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" }
    ]
  }
]
```

Fall 1

Erkennung von Datenbankinstanzen auf einem Datenbankserver und anschließende Erkennung der Tablespaces für jede Instanz.

1. Sie haben mindestens einen Host, der mit der Erkennung des Datenbankservers verknüpft ist.
2. Erstellen Sie für diesen Host eine LLD-Regel mit dem Namen *Datenbanken und Tablespaces erkennen*.
3. Wechseln Sie in dieser Regel auf den Reiter *LLD-Makros* und fügen Sie das Makro `{#DB}=$.database` hinzu.
4. Fügen Sie für diese Regel einen Datenpunkt-Prototyp mit dem Namen *Aktive Verbindungen zu {#DB}* hinzu (Typ: Agent, Schlüssel: `db.connections[{#DB}]`).
5. Die Datenpunkte für jede Datenbank werden erkannt:

```
Active connections to db1, Active connections to db2, Active connections to db3.
```

6. Erstellen Sie für diese Regel einen Discovery-Prototyp mit dem Namen *Tablespaces für {#DB} erkennen* (Typ: Nested, Schlüssel: `db.tablespace.discovery[{#DB}]`).
7. Wechseln Sie auf den Reiter *Preprocessing* dieses Discovery-Prototyps und fügen Sie den Schritt `JSONPath=$.tablespaces` hinzu.
8. Wechseln Sie auf den Reiter *LLD-Makros* dieses Discovery-Prototyps und fügen Sie das Makro `{#TSNAME}=$.name` hinzu.
9. Erstellen Sie für diesen Discovery-Prototyp einen Datenpunkt-Prototyp mit dem Namen *Größe des Tablespace {#TSNAME} für {#DB}* (Typ: Agent, Schlüssel: `db.ts.size[{#DB}, {#TSNAME}]`).
10. Die Datenpunkte für jeden Tablespace jeder Datenbank werden erkannt:

```
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.
```

mit den Schlüsseln `db.ts.size[db1,ts1]`, `db.ts.size[db1,ts2]`, ... `db.ts.size[db3,ts3]`.

Fall 2

Erkennung von Datenbankinstanzen auf dem Datenbank-Server, indem sie als erkannte Hosts dargestellt werden, und anschließende Erkennung der Tablespaces für jede Instanz.

1. Sie haben mindestens einen Host (Root-Host), der mit der Erkennung des Datenbank-Servers verknüpft ist.
2. Erstellen Sie eine Vorlage, um die Tablespaces für jede Datenbank zu erkennen.
3. Erstellen Sie in dieser Vorlage einen Datenpunkt mit dem Namen *Aktive Verbindungen zu {#DB}* (Typ: Agent, Schlüssel: `db.connections[{#DB}]`).
4. Erstellen Sie für diese Vorlage eine LLD-Regel mit dem Namen *Tablespaces erkennen* (Typ: Verschachtelt).
5. Wechseln Sie zur Registerkarte *Preprocessing* dieser Regel und fügen Sie den Schritt `JSONPath=$.tablespaces` hinzu.
6. Wechseln Sie zur Registerkarte *LLD-Makros* dieser Regel und fügen Sie das Makro `{#TSNAME}=$.name` hinzu.
7. Erstellen Sie für diese Regel einen Datenpunkt-Prototyp mit dem Namen *Größe des Tablespace {#TSNAME} für {#DB}* (Typ: Agent, Schlüssel: `db.ts.size[{#DB}, {#TSNAME}]`).
8. Erstellen Sie zurück auf dem Root-Host eine LLD-Regel für diesen Host mit dem Namen *Datenbanken und Tablespaces erkennen*.
9. Wechseln Sie bei dieser Regel zur Registerkarte *LLD-Makros* und fügen Sie das Makro `{#DB}=$.database` hinzu.
10. Fügen Sie für diese Regel einen Host-Prototyp mit dem Namen *Host für Datenbank {#DB}* hinzu.
11. Wechseln Sie bei diesem Host-Prototyp zur Registerkarte *Makros* und fügen Sie das Makro `{$DB}={#DB}` hinzu (für den Namen und den Schlüssel des Datenpunkts aus Schritt 3).
12. Verknüpfen Sie die Vorlage aus Schritt 2 mit diesem Host-Prototyp.
13. Die erkannten Hosts enthalten die erkannten Datenpunkte für jede Datenbank und ihre Tablespaces:

Host	Datenpunkte
Host für Datenbank db1	Aktive Verbindungen zu db1
	Größe des Tablespace ts1 für db1
	Größe des Tablespace ts2 für db1
	Größe des Tablespace ts3 für db1
Host für Datenbank db2	Aktive Verbindungen zu db2
	Größe des Tablespace ts1 für db2
	Größe des Tablespace ts2 für db2
	Größe des Tablespace ts3 für db2
Host für Datenbank db3	Aktive Verbindungen zu db3
	Größe des Tablespace ts1 für db3
	Größe des Tablespace ts2 für db3
	Größe des Tablespace ts3 für db3

Hinweise zur Low-Level-Erkennung

Verwendung von LLD-Makros in Benutzer-Makrokontexten

LLD-Makros können innerhalb des Benutzer-Makrokontexts verwendet werden, zum Beispiel in Auslöser- Prototypen.

Mehrere LLD-Regeln für denselben Datenpunkt

Es ist möglich, mehrere Low-Level-Discovery-Regeln mit demselben Discovery-Datenpunkt zu definieren.

Dazu müssen Sie den Agent-Parameter Alias definieren. Dadurch können Sie in verschiedenen Discovery-Regeln geänderte Schlüssel für den Discovery-Datenpunkt verwenden, zum Beispiel `vfs.fs.discovery[foo]`, `vfs.fs.discovery[bar]` usw.

Datenlimits für Rückgabewerte

Für JSON-Daten von Low-Level-Discovery-Regeln gibt es kein Limit, wenn sie direkt vom Zabbix Server empfangen werden. Der Grund dafür ist, dass die Rückgabewerte verarbeitet werden, ohne in einer Datenbank gespeichert zu werden.

Auch für benutzerdefinierte Low-Level-Discovery-Regeln gibt es kein Limit. Wenn Daten benutzerdefinierter Low-Level-Discovery-Regeln jedoch über einen UserParameter abgerufen werden, gilt das Rückgabewert-Limit für UserParameter.

Wenn Daten über einen Zabbix Proxy laufen müssen, muss dieser diese Daten in der Datenbank speichern. In einem solchen Fall gelten die Datenbanklimits.

Beispiele für Discovery-Regeln

Dieser Abschnitt bietet praktische Beispiele für die Konfiguration von Discovery-Regeln für verschiedene Arten von Entitäten, wie Dateisysteme, Netzwerkschnittstellen, Dienste, Sensoren, SNMP-Objekte, Java-Anwendungen und Datenbanken.

Jedes Beispiel veranschaulicht, wie eine Discovery-Regel aufgebaut sein kann, einschließlich des verwendeten Item-Schlüssels oder der Abfrage, der JSON-Ausgabe mit LLD-Makros und der Art und Weise, wie diese Ergebnisse zum Erstellen von Item-, Auslöser- und Diagrammvorlagen verwendet werden.

- Discovery von Blockgeräten
- Discovery von CPUs und CPU-Kernen
- Discovery von Host-Schnittstellen in Zabbix
- Discovery von IPMI-Sensoren
- Discovery von JMX-Objekten
- Discovery von eingehängten Dateisystemen
- Discovery von Netzwerkschnittstellen
- Discovery von SNMP-OIDs
- Discovery von SNMP-OIDs (legacy)
- Discovery von systemd-Diensten
- Discovery von Windows-Performance-Counter-Instanzen
- Discovery von Windows-Diensten
- Discovery mithilfe von WMI-Abfragen
- Discovery mithilfe von ODBC-SQL-Abfragen
- Discovery mithilfe von Prometheus-Daten

Benutzerdefinierte LLD-Regeln

Übersicht

Es ist auch möglich, eine vollständig benutzerdefinierte LLD-Regel zu erstellen, die beliebige Arten von Entitäten erkennt – zum Beispiel Datenbanken auf einem Datenbank-Server.

Dazu sollte ein benutzerdefinierter Datenpunkt erstellt werden, der eine JSON-Zeichenfolge zurückgibt und die gefundenen Objekte sowie optional einige ihrer Eigenschaften angibt. Die Anzahl der Makros pro Entität ist nicht begrenzt – während die integrierten Discovery-Regeln entweder ein oder zwei Makros zurückgeben (zum Beispiel zwei für die Dateisystem-Erkennung), ist es möglich, mehr zurückzugeben.

Beispiel

Das erforderliche JSON-String-Format lässt sich am besten anhand eines Beispiels veranschaulichen. Angenommen, Sie verwenden einen alten Zabbix 1.8 Agent (einen, der den Schlüssel `vfs.fs.discovery` nicht unterstützt), müssen aber dennoch Dateisysteme erkennen. Hier ist ein einfaches Perl-Skript für Linux, das eingehängte Dateisysteme erkennt und JSON ausgibt, das sowohl den Namen als auch den Typ des Dateisystems enthält. Eine Möglichkeit zur Verwendung wäre als UserParameter mit dem Schlüssel `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:
Zulässige Symbole für LLD-Makronamen sind **0-9** , **A-Z** , **_** , **.**. Kleinbuchstaben werden in den Namen nicht unterstützt.

Ein Beispiel für die Ausgabe (zur besseren Übersichtlichkeit neu formatiert) ist unten dargestellt. Das JSON für benutzerdefinierte Discovery-Prüfungen muss demselben Format folgen.

```
[
    { "{#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"  }
]
```

Im vorherigen Beispiel ist es erforderlich, dass die Schlüssel mit den in Prototypen verwendeten LLD-Makronamen übereinstimmen. Alternativ können LLD-Makrowerte mit JSONPath extrahiert werden: `{#FSNAME}` → `$.fsname` und `{#FSTYPE}` → `$.fstype`, wodurch ein solches Skript möglich wird:

```
####!/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";
```

Ein Beispiel für die Ausgabe (zur besseren Übersichtlichkeit neu formatiert) ist unten dargestellt.

```
[
    { "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"  }
]
```

Anschließend können Sie im Feld *Filter* der Discovery-Regel `{#FSTYPE}` als Makro und `rootfs|ext3` als regulären Ausdruck angeben.

Note:
Sie müssen bei benutzerdefinierten LLD-Regeln keine Makronamen wie `FSNAME`/`FSTYPE` verwenden; Sie können beliebige Namen Ihrer Wahl verwenden. Wenn JSONPath verwendet wird, ist die LLD-Zeile ein Array-Element, das ein Objekt sein kann, aber auch ein anderes Array oder ein Wert sein kann.

Beachten Sie, dass der Rückgabewert bei Verwendung eines Benutzerparameters auf 16 MB begrenzt ist. Weitere Details finden Sie unter Datenlimits für LLD-Rückgabewerte.

Überwachung virtueller Maschinen

Überblick Zabbix kann Regeln für die Low-Level-Discovery verwenden, um VMware-Hypervisoren und virtuelle Maschinen automatisch zu erkennen und Hosts zu erstellen, um sie auf Grundlage vordefinierter Host-Prototypen zu überwachen.

Zabbix enthält außerdem einsatzbereite Vorlagen für die Überwachung von VMware vCenter- oder ESXi-Hypervisoren.

Die mindestens erforderliche VMware vCenter- oder vSphere-Version ist 5.1.

Datenerfassung Die Überwachung virtueller Maschinen besteht aus zwei Schritten:

1. Zabbix-*vmware collector*-Prozesse sammeln Daten virtueller Maschinen – die Prozesse beziehen die erforderlichen Informationen über das SOAP-Protokoll von VMware-Webservices, verarbeiten sie vor und speichern sie im Shared Memory des Zabbix Server.
2. Zabbix-*poller*-Prozesse rufen Daten mithilfe des einfachen Zabbix-Checks VMware monitoring item keys ab.

Zabbix unterteilt die gesammelten Daten in VMware-Konfigurationsdaten und VMware-Performance-Counter-Daten.
Beide Datentypen werden unabhängig voneinander von den *vmware collector*-Prozessen erfasst.

Die folgenden Statistiken sind auf Grundlage der Informationen aus den VMware-Performance-Countern verfügbar:

- Datenspeicher
- Festplattengerät
- CPU
- Stromversorgung
- Netzwerkschnittstelle
- Benutzerdefinierte Performance-Counter-Datenpunkte

Die vollständige Liste der Datenpunkte, die Daten aus VMware-Performance-Countern beziehen, finden Sie unter *VMware monitoring item keys*.

Beachten Sie, dass die Häufigkeit des Abrufs von VMware-Ereignissen vom Abfrageintervall von vmware.eventlog abhängt, jedoch nicht unter 5 Sekunden liegen kann.

Konfiguration Wenn der Zabbix Server aus Quellen kompiliert wird, muss er mit den Konfigurationsoptionen `--with-libcurl --with-libxml2` kompiliert werden, um die Überwachung virtueller Maschinen zu aktivieren. Zabbix-Pakete werden bereits mit diesen aktivierten Optionen kompiliert.

Die folgenden Parameter der Zabbix Server-Konfigurationsdatei können für die Überwachung virtueller Maschinen angepasst werden:

- `StartVMwareCollectors`

Note:
Es wird empfohlen, mehr Collector zu aktivieren als die Anzahl der überwachten VMware-Dienste; andernfalls kann sich das Abrufen der VMware-Performance-Counter-Statistiken durch das Abrufen der VMware-Konfigurationsdaten verzögern (was bei großen Installationen einige Zeit in Anspruch nimmt).

 Im Allgemeinen sollte der Wert von `StartVMwareCollectors` nicht unter 2 liegen und nicht mehr als das Doppelte der Anzahl der überwachten VMware-Dienste betragen: Anzahl der Dienste < `StartVMwareCollectors` < (Anzahl der Dienste * 2). Wenn Sie beispielsweise einen VMware-Dienst überwachen, setzen Sie `StartVMwareCollectors` auf 2; wenn Sie drei Dienste überwachen, setzen Sie `StartVMwareCollectors` auf 5.

 Beachten Sie, dass die erforderliche Anzahl an Collector auch vom Umfang der VMware-Umgebung sowie von den Konfigurationsparametern `VMwareFrequency` und `VMwarePerfFrequency` abhängt.

- `VMwareCacheSize`

- VMwareFrequency
- VMwarePerfFrequency
- VMwareTimeout

Attention:
Um Metriken zur Datenspeicherkapazität zu unterstützen, stellen Sie sicher, dass der Wert des VMware-Schlüssels `vpxd.stats.maxQueryMetrics` auf mindestens 64 gesetzt ist. Weitere Informationen finden Sie im Artikel der *VMware Knowledge Base*.

Discovery

Zabbix kann Low-Level-Discovery-Regeln (zum Beispiel `vmware.hv.discovery[{$VMWARE.URL}]`) verwenden, um VMware-Hypervisoren und virtuelle Maschinen automatisch zu entdecken. Darüber hinaus kann Zabbix Host-Prototypen verwenden, um automatisch echte Hosts für die erkannten Entitäten zu erzeugen. Weitere Informationen finden Sie unter *Host-Prototypen*.

Konfigurationsbeispiele

Ein grundlegendes Beispiel dafür, wie Zabbix für die Überwachung von VMware mit der Vorlage *VMware FQDN* eingerichtet wird, finden Sie unter *VMware mit Zabbix überwachen*.

Ein ausführlicheres Beispiel dafür, wie ein Host, eine Low-Level-Discovery-Regel und ein Host-Prototyp für die Überwachung von VMware erstellt werden, finden Sie unter *Einrichtungsbeispiel*.

Erweiterte Protokollierung Die von den *vmware collector*-Prozessen erfassten Daten können zur detaillierten Fehlersuche mit Debug-Level 5 protokolliert werden. Das Debug-Level kann in den Konfigurationsdateien von Server und Proxy oder mithilfe der Laufzeitsteuerungsoption `-R log_level_increase="vmware collector,N"` konfiguriert werden, wobei `N` die Prozessnummer ist.

Um beispielsweise das Debug-Level für alle *vmware collector*-Prozesse von 4 auf 5 zu erhöhen, führen Sie den folgenden Befehl aus:

```
zabbix_server -R log_level_increase="vmware collector"
```

Um das Debug-Level für den zweiten *vmware collector*-Prozess von 4 auf 5 zu erhöhen, führen Sie den folgenden Befehl aus:

```
zabbix_server -R log_level_increase="vmware collector,2"
```

Wenn die erweiterte Protokollierung der von VMware Collector erfassten Daten nicht mehr erforderlich ist, wird empfohlen, das Debug-Level durch Ausführen des Befehls `-R log_level_decrease` auf den Standardwert (3) zu verringern.

Fehlerbehebung

- Falls Metriken nicht verfügbar sind, stellen Sie bitte sicher, dass sie in neueren VMware vSphere-Versionen nicht standardmäßig deaktiviert oder abgeschaltet wurden oder dass keine Einschränkungen für Datenbankabfragen zu Leistungsmetriken festgelegt sind. Weitere Informationen finden Sie unter ZBX-12094.
- Wenn `config.vpxd.stats.maxQueryMetrics` ungültig ist oder der Fehler auftritt, dass die maximal zulässige Anzahl von Zeichen überschritten wird, fügen Sie den vCenter-Server-Einstellungen einen Parameter `config.vpxd.stats.maxQueryMetrics` hinzu. Der Wert dieses Parameters sollte mit dem Wert von `maxQuerysize` in der Datei `web.xml` von VMware übereinstimmen. Weitere Informationen finden Sie im Artikel *VMware Knowledge Base*.
- Wenn Sie vermuten, dass Ihre Zabbix-Installation zu viel Speicher verwendet, siehe Profiling excessive memory usage with tcmalloc.

VMware-Überwachungs-Datenpunkt-Schlüssel

Übersicht Diese Seite enthält Details zu den einfachen Prüfungen, die zur Überwachung von VMware-Umgebungen verwendet werden können.
Die Metriken sind nach dem Überwachungsziel gruppiert.

Unterstützte Datenpunktschlüssel Die Datenpunktschlüssel werden ohne Parameter und zusätzliche Informationen aufgeführt. Klicken Sie auf den Datenpunktschlüssel, um die vollständigen Details anzuzeigen.

Datenpunktschlüssel	Beschreibung	Datenpunktgruppe
vmware.eventlog	Das VMware-Ereignisprotokoll.	Allgemeiner Dienst

Datenpunktschlüssel	Beschreibung	Datenpunktgruppe
vmware.fullname	Der vollständige Name des VMware-Dienstes.	
vmware.version	Die Version des VMware-Dienstes.	
vmware.cl.perfcounter	Die Metriken des Leistungszählers des VMware-Clusters.	Cluster
vmware.cluster.alarms.get	Die Alarmdaten des VMware-Clusters.	
vmware.cluster.discovery	Die Erkennung von VMware-Clustern.	
vmware.cluster.property	Die Eigenschaft des VMware-Clusters.	
vmware.cluster.status	Der Status des VMware-Clusters.	
vmware.cluster.tags.get	Das Tag-Array des VMware-Clusters.	
vmware.datastore.alarms.get	Die Alarmdaten des VMware-Datenspeichers.	Datenspeicher
vmware.datastore.discovery	Die Erkennung von VMware-Datenspeichern.	
vmware.datastore.hv.list	Die Liste der Datenspeicher-Hypervisoren.	
vmware.datastore.perfcounter	Der Wert des Leistungszählers des VMware-Datenspeichers.	
vmware.datastore.property	Die Eigenschaft des VMware-Datenspeichers.	
vmware.datastore.read	Die Zeitdauer für einen Lesevorgang aus dem Datenspeicher.	
vmware.datastore.size	Der Speicherplatz des VMware-Datenspeichers in Bytes oder als Prozentsatz des Gesamtspeichers.	
vmware.datastore.tags.get	Das Tag-Array des VMware-Datenspeichers.	
vmware.datastore.write	Die Zeitdauer für einen Schreibvorgang in den Datenspeicher.	
vmware.dc.alarms.get	Die Alarmdaten des VMware-Datenzentrums.	Datenzentrum
vmware.dc.discovery	Die Erkennung von VMware-Datenzentren.	
vmware.dc.tags.get	Das Tag-Array des VMware-Datenzentrums.	
vmware.dvswitch.discovery	Die Erkennung von VMware vSphere Distributed Switches.	vSphere Distributed Switch
vmware.dvswitch.fetchportdata	Die Portdaten des VMware vSphere Distributed Switch.	
vmware.hv.alarms.get	Die Alarmdaten des VMware-Hypervisors.	Hypervisor
vmware.hv.cluster.name	Der Clustername des VMware-Hypervisors.	
vmware.hv.connectionstate	Der Verbindungsstatus des VMware-Hypervisors.	
vmware.hv.cpu.usage	Die Prozessorauslastung des VMware-Hypervisors (Hz).	
vmware.hv.cpu.usage.perf	Die Prozessorauslastung des VMware-Hypervisors als Prozentsatz während des Intervalls.	
vmware.hv.cpu.utilization	Die Prozessorauslastung des VMware-Hypervisors als Prozentsatz während des Intervalls, abhängig von der Energieverwaltung oder HT.	
vmware.hv.datacenter.name	Der Name des Datenzentrums des VMware-Hypervisors.	
vmware.hv.datastore.discovery	Die Erkennung von Datenspeichern des VMware-Hypervisors.	
vmware.hv.datastore.list	Die Liste der Datenspeicher des VMware-Hypervisors.	
vmware.hv.datastore.multipath	Die Anzahl der verfügbaren Datenspeicherpfade.	
vmware.hv.datastore.read	Die durchschnittliche Zeitdauer für einen Lesevorgang aus dem Datenspeicher.	
vmware.hv.datastore.size	Der Speicherplatz des VMware-Datenspeichers in Bytes oder als Prozentsatz des Gesamtspeichers.	
vmware.hv.datastore.write	Die durchschnittliche Zeitdauer für einen Schreibvorgang in den Datenspeicher.	
vmware.hv.discovery	Die Erkennung von VMware-Hypervisors.	
vmware.hv.diskinfo.get	Die Festplattendaten des VMware-Hypervisors.	
vmware.hv.fullname	Der vollständige Produktname einschließlich Versionsinformationen.	
vmware.hv.hw.cpu.freq	Die Prozessorfrequenz des VMware-Hypervisors.	
vmware.hv.hw.cpu.model	Das Prozessormodell des VMware-Hypervisors.	
vmware.hv.hw.cpu.num	Die Anzahl der Prozessorkerne auf dem VMware-Hypervisor.	
vmware.hv.hw.cpu.threads	Die Anzahl der Prozessor-Threads auf dem VMware-Hypervisor.	
vmware.hv.hw.memory	Die gesamte Speicherkapazität des VMware-Hypervisors.	
vmware.hv.hw.model	Das Modell des VMware-Hypervisors.	
vmware.hv.hw.sensors.get	Die Werte der Hardwaresensoren des VMware-Hypervisors.	
vmware.hv.hw.serialnumber	Die Seriennummer des VMware-Hypervisors.	
vmware.hv.hw.uuid	Die BIOS-UUID des VMware-Hypervisors.	
vmware.hv.hw.vendor	Der Herstellername des VMware-Hypervisors.	
vmware.hv.maintenance	Der Wartungsstatus des VMware-Hypervisors.	
vmware.hv.memory.size.ballooned	Die Balloon-Speichergröße des VMware-Hypervisors.	
vmware.hv.memory.used	Die verwendete Speichermenge des VMware-Hypervisors.	
vmware.hv.net.if.discovery	Die Erkennung von Netzwerkschnittstellen des VMware-Hypervisors.	
vmware.hv.network.in	Die eingehenden Netzwerkstatistiken des VMware-Hypervisors.	
vmware.hv.network.linkspeed	Die Geschwindigkeit der Netzwerkschnittstelle des VMware-Hypervisors.	
vmware.hv.network.out	Die ausgehenden Netzwerkstatistiken des VMware-Hypervisors.	

Datenpunktschlüssel	Beschreibung	Datenpunktgruppe
vmware.hv.perfcounter	Der Wert des Leistungszählers des VMware-Hypervisors.	
vmware.hv.property	Die Eigenschaft des VMware-Hypervisors.	
vmware.hv.power	Der Stromverbrauch des VMware-Hypervisors.	
vmware.hv.sensor.health.state	Der zusammengefasste Gesundheitszustandssensor des VMware-Hypervisors.	
vmware.hv.sensors.get	Die Zustandsensoren des HW-Herstellers des VMware-Hypervisors.	
vmware.hv.status	Der Status des VMware-Hypervisors.	
vmware.hv.tags.get	Das Tag-Array des VMware-Hypervisors.	
vmware.hv.uptime	Die Betriebszeit des VMware-Hypervisors.	
vmware.hv.version	Die Version des VMware-Hypervisors.	
vmware.hv.vm.num	Die Anzahl der virtuellen Maschinen auf dem VMware-Hypervisor.	
vmware.rp.cpu.usage	Die CPU-Auslastung in Hertz während des Intervalls im VMware Resource Pool.	Ressourcenpool
vmware.rp.memory	Die Speicherkennzahlen des VMware-Ressourcenpools.	
vmware.alarms.get	Die Alarmdaten des VMware Virtual Center.	Virtual Center
vmware.vm.alarms.get	Die Alarmdaten der VMware-virtuellen Maschine.	Virtuelle Maschine
vmware.vm.attribute	Der Wert eines benutzerdefinierten Attributs der VMware-virtuellen Maschine.	
vmware.vm.cluster.name	Der Name der VMware-virtuellen Maschine.	
vmware.vm.consolidationneeded	Die Festplatte der VMware-virtuellen Maschine muss konsolidiert werden.	
vmware.vm.cpu.latency	Der Prozentsatz der Zeit, in der die virtuelle Maschine nicht ausgeführt werden kann, weil sie um den Zugriff auf die physischen CPU(s) konkurriert.	
vmware.vm.cpu.num	Die Anzahl der Prozessoren auf der VMware-virtuellen Maschine.	
vmware.vm.cpu.readiness	Der Prozentsatz der Zeit, in der die virtuelle Maschine bereit war, aber nicht zur Ausführung auf der physischen CPU eingeplant werden konnte.	
vmware.vm.cpu.ready	Die Zeit, in der die virtuelle Maschine bereit war, aber nicht zur Ausführung auf der physischen CPU eingeplant werden konnte.	
vmware.vm.cpu.swapwait	Der Prozentsatz der CPU-Zeit, die auf Swap-In gewartet wurde.	
vmware.vm.cpu.usage	Die Prozessorauslastung der VMware-virtuellen Maschine (Hz).	
vmware.vm.cpu.usage.perf	Die Prozessorauslastung der VMware-virtuellen Maschine als Prozentsatz während des Intervalls.	
vmware.vm.datacenter.name	Der Name des Datenzentrums der VMware-virtuellen Maschine.	
vmware.vm.discovery	Die Erkennung von VMware-virtuellen Maschinen.	
vmware.vm.guest.memory.swapped	Die Menge des physischen Gastspeichers, die in den Swap-Bereich ausgelagert wurde.	
vmware.vm.guest.osuptime	Die insgesamt verstrichene Zeit seit dem letzten Start des Betriebssystems.	
vmware.vm.hv.maintenance	Der Wartungsstatus des Hypervisors der VMware-virtuellen Maschine.	
vmware.vm.hv.name	Der Name des Hypervisors der VMware-virtuellen Maschine.	
vmware.vm.memory.size	Die gesamte Speicherkapazität der VMware-virtuellen Maschine.	
vmware.vm.memory.size.ballooned	Die Balloon-Speichergröße der VMware-virtuellen Maschine.	
vmware.vm.memory.size.compressed	Die komprimierte Speichergröße der VMware-virtuellen Maschine.	
vmware.vm.memory.size.consumed	Die Menge an physischem Host-Speicher, die für die Sicherung der physischen Gastspeicherseiten verbraucht wird.	
vmware.vm.memory.size.private	Die private Speichergröße der VMware-virtuellen Maschine.	
vmware.vm.memory.size.shared	Die gemeinsam genutzte Speichergröße der VMware-virtuellen Maschine.	
vmware.vm.memory.size.swapped	Die ausgelagerte Speichergröße der VMware-virtuellen Maschine.	
vmware.vm.memory.size.usage.guest	Die Speichernutzung des Gastsystems der VMware-virtuellen Maschine.	
vmware.vm.memory.size.usage.host	Die Speichernutzung des Hosts der VMware-virtuellen Maschine.	
vmware.vm.memory.usage	Der Prozentsatz des physischen Host-Speichers, der verbraucht wurde.	
vmware.vm.net.if.discovery	Die Erkennung von Netzwerkschnittstellen der VMware-virtuellen Maschine.	
vmware.vm.net.if.in	Die eingehenden Statistiken der Netzwerkschnittstelle der VMware-virtuellen Maschine.	
vmware.vm.net.if.out	Die ausgehenden Statistiken der Netzwerkschnittstelle der VMware-virtuellen Maschine.	
vmware.vm.net.if.usage	Die Netzwerkauslastung der VMware-virtuellen Maschine während des Intervalls.	
vmware.vm.perfcounter	Der Wert des Leistungszählers der VMware-virtuellen Maschine.	
vmware.vm.powerstate	Der Stromversorgungszustand der VMware-virtuellen Maschine.	
vmware.vm.property	Die Eigenschaft der VMware-virtuellen Maschine.	
vmware.vm.snapshot.get	Der Snapshot-Status der VMware-virtuellen Maschine.	
vmware.vm.state	Der Zustand der VMware-virtuellen Maschine.	
vmware.vm.storage.committed	Der zugesicherte Speicherplatz der VMware-virtuellen Maschine.	
vmware.vm.storage.readoio	Die durchschnittliche Anzahl ausstehender Leseanforderungen an die virtuelle Festplatte während des Erfassungsintervalls.	

Datenpunktschlüssel	Beschreibung	Datenpunktgruppe
vmware.vm.storage.totalreadlatency	Die durchschnittliche Zeit, die ein Lesevorgang von der virtuellen Festplatte benötigt.	
vmware.vm.storage.totalwritelatency	Die durchschnittliche Zeit, die ein Schreibvorgang auf die virtuelle Festplatte benötigt.	
vmware.vm.storage.uncommitted	Der nicht zugesicherte Speicherplatz der VMware-virtuellen Maschine.	
vmware.vm.storage.unshared	Der nicht gemeinsam genutzte Speicherplatz der VMware-virtuellen Maschine.	
vmware.vm.storage.writeoio	Die durchschnittliche Anzahl ausstehender Schreibanforderungen an die virtuelle Festplatte während des Erfassungsintervalls.	
vmware.vm.tags.get	Das Tag-Array der VMware-virtuellen Maschine.	
vmware.vm.tools	Der Status oder die Version der Gast-Tools der VMware-virtuellen Maschine.	
vmware.vm.uptime	Die Betriebszeit der VMware-virtuellen Maschine.	
vmware.vm.vfs.dev.discovery	Die Erkennung von Festplattengeräten der VMware-virtuellen Maschine.	
vmware.vm.vfs.dev.read	Die Lese-Statistiken der Festplattengeräte der VMware-virtuellen Maschine.	
vmware.vm.vfs.dev.write	Die Schreib-Statistiken der Festplattengeräte der VMware-virtuellen Maschine.	
vmware.vm.vfs.fs.discovery	Die Erkennung von Dateisystemen der VMware-virtuellen Maschine.	
vmware.vm.vfs.fs.size	Die Dateisystemstatistiken der VMware-virtuellen Maschine.	

Details zum Datenpunktschlüssel Parameter ohne spitze Klammern sind obligatorisch. Parameter, die mit spitzen Klammern **< >** gekennzeichnet sind, sind optional.

vmware.eventlog[url,<mode>,<severity>]

 Das VMware-Ereignisprotokoll.
 Rückgabewert: *Log*.

Parameter:

- **url** - die VMware-Service-URL;
- **mode** - *all* (Standard) oder *skip* - Verarbeitung älterer Daten überspringen;
- **severity** - nach Schweregrad filtern: *error*, *warning*, *info* oder *user*. Dieser Parameter muss in Anführungszeichen gesetzt werden, wenn mehr als ein Schweregrad in einer kommagetrennten Liste angegeben wird (z. B. `"error,warning,info,user"`). Standardmäßig deaktiviert.

Kommentare:

- Pro URL darf es nur einen `vmware.eventlog`-Datenpunktschlüssel geben;
- Siehe auch Beispiel für die Filterung von VMware-Ereignisprotokolleinträgen.

vmware.fullnameurl

 Der vollständige Name des VMware-Dienstes.
 Rückgabewert: *String*.

Parameter:

- **url** - die URL des VMware-Dienstes.

vmware.versionurl

 Die Version des VMware-Dienstes.
 Rückgabewert: *String*.

Parameter:

- **url** - die URL des VMware-Dienstes.

vmware.cl.perfcounter[url,id,path,<instance>]

 Die Metriken des VMware-Cluster-Performance-Counters.
 Rückgabewert: *Integer*.

Parameter:

- **url** - die VMware-Service-URL;
- **id** - die VMware-Cluster-ID. `id` kann von `vmware.cluster.discovery[]` als `{#CLUSTER.ID}` abgerufen werden.
- **path** - der Pfad des Performance-Counters[1];
- **instance** - die Instanz des Performance-Counters.

vmware.cluster.alarms.get[url,id]

 Die VMware-Cluster-Alarmdaten.
 Rückgabewert: JSON object.

Parameter:

- **url** - die VMware-Dienst-URL;

- **id** - die VMware-Cluster-ID.

vmware.cluster.discoveryurl

 Die Discovery von VMware-Clustern.
 Rückgabewert: JSON-Objekt.

Parameter:

- **url** - die URL des VMware-Service.

vmware.cluster.property[url,id,prop]

 Die VMware-Cluster-Eigenschaft.
 Rückgabewert: *String*.

Parameter:

- **url** - die VMware-Service-URL;
- **id** - die VMware-Cluster-ID;
- **prop** - der Eigenschaftspfad, also der Name einer Eigenschaft des VM-Objekts, wie im VMware SDK definiert.

Beispiele:

```
vmware.vm.property[{$VMWARE.URL},{$VMWARE.VM.UUID},overallStatus]
vmware.vm.property[{$VMWARE.URL},{$VMWARE.VM.UUID},runtime.powerState]
```

vmware.cluster.status[url,name]

 Der VMware-Clusterstatus.
 Rückgabewert: *0* - grau; *1* - grün; *2* - gelb; *3* - rot.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **name** - der Name des VMware-Clusters.

vmware.cluster.tags.get[url,id]

 Das Array der VMware-Cluster-Tags.
 Rückgabewert: JSON object.

Parameter:

- **url** - die VMware-Dienst-URL;
- **id** - die VMware-Cluster-ID.

Dieser Datenpunkt funktioniert mit vSphere 6.5 und neuer.

vmware.datastore.alarms.get[url,uuid]

 Die VMware-Datastore-Alarmdaten.
 Rückgabewert: JSON object.

Parameter:

- **url** - die VMware-Dienst-URL;
- **uuid** - der globale eindeutige Bezeichner des VMware-Datastores.

vmware.datastore.discovery[url,<filter_uuid>]

 Die Discovery von VMware-Datastores.
 Rückgabewert: JSON-Objekt.

Parameter:

- **url** - die VMware-Service-URL;
- **filter_uuid** - eine einzelne eindeutige Kennung eines HV oder einer VM (Standard: leer). Falls angegeben, werden nur verknüpfte Datastores erkannt.

vmware.datastore.hv.list[url,datastore]

 Die Liste der Datastore-Hypervisoren.
 Rückgabewert: *String*.

Parameter:

- **url** - die VMware-Service-URL;
- **datastore** - die UUID oder der Name des Datastore.

Ausgabebeispiel:

```
esx7-01-host.zabbix.sandbox
esx7-02-host.zabbix.sandbox
```

vmware.datastore.perfcounter[url,uuid,path,<instance>]

 Der Wert des VMware-Datastore-Performance-Counters.
 Rückgabewert: *Integer* [2].

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung des VMware-Datastores;
- **path** - der Pfad des Performance-Counters[1];
- **instance** - die Instanz des Performance-Counters. Verwenden Sie eine leere Instanz für aggregierte Werte (Standard). `instance` kann von `vmware.datastore.discovery[]` als Teil des Arrays {#DATASTORE.EXTENT} empfangen werden.

vmware.datastore.property[url,uuid,prop]

 Die VMware-Datastore-Eigenschaft.
 Rückgabewert: *String*.

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung des VMware-Datastores;
- **prop** - der Eigenschaftspfad.

vmware.datastore.read[url,datastore,<mode>]

 Die Dauer eines Lesevorgangs aus dem Datenspeicher (Millisekunden).
 Rückgabewert: *Integer* [2].

Parameter:

- **url** - die VMware-Service-URL;
- **datastore** - die UUID oder der Name des Datenspeichers;
- **mode** - *latency* (Durchschnittswert, Standard) oder *maxlatency* (Maximalwert).

vmware.datastore.size[url,datastore,<mode>]

 Der Speicherplatz des VMware-Datastores in Byte oder als Prozentsatz der Gesamtkapazität.
 Rückgabewert: *Integer* - für Byte; *Float* - für Prozentwerte.

Parameter:

- **url** - die URL des VMware-Service;
- **datastore** - die UUID oder der Name des Datastores;
- **mode** - mögliche Werte: *total* (Standard), *free*, *pfree* (freier Prozentsatz), *uncommitted*.

vmware.datastore.tags.get[url,uuid]

 Das Array der VMware-Datastore-Tags.
 Rückgabewert: JSON object.

Parameter:

- **url** - die VMware-Dienst-URL;
- **uuid** - der global eindeutige Bezeichner des VMware-Datastores.

Dieser Datenpunkt funktioniert mit vSphere 6.5 und neuer.

vmware.datastore.write[url,datastore,<mode>]

 Die Dauer eines Schreibvorgangs in den Datenspeicher (Millisekunden).
 Rückgabewert: *Integer* [2].

Parameter:

- **url** - die VMware-Service-URL;
- **datastore** - die UUID oder der Name des Datenspeichers;
- **mode** - *latency* (Durchschnittswert, Standard) oder *maxlatency* (Maximalwert).

vmware.dc.alarms.get[url,id]

 Die VMware-Datacenter-Alarmdaten.
 Rückgabewert: JSON object.

Parameter:

- **url** - die VMware-Dienst-URL;
- **id** - die VMware-Datacenter-ID.

vmware.dc.discoveryurl

 Die Erkennung von VMware-Datacentern.
 Rückgabewert: JSON-Objekt.

Parameter:

- **url** - die URL des VMware-Dienstes.

vmware.dc.tags.get[url,id]

 Das Array der VMware-Datacenter-Tags.
 Rückgabewert: JSON object.

Parameter:

- **url** - die VMware-Dienst-URL;
- **id** - die VMware-Datacenter-ID.

Dieser Datenpunkt funktioniert mit vSphere 6.5 und neuer.

vmware.dvswitch.discoveryurl

 Die Discovery von VMware vSphere Distributed Switches.
 Rückgabewert: JSON-Objekt.

Parameter:

- **url** - die VMware-Service-URL.

vmware.dvswitch.fetchports.get[url,uuid,<filter>,<mode>]

 Die Portdaten des VMware vSphere Distributed Switch.
 Rückgabewert: JSON object.

Parameter:

- **url** - die VMware-Dienst-URL;
- **uuid** - die globale eindeutige Kennung des VMware DVSwitch.
- **filter** - eine einzelne Zeichenfolge mit durch Kommas getrennten Kriterien zur Auswahl von Ports;
- **mode** - *state* (alle XML ohne ”config”-XML-Knoten, Standard) oder *full*.

Der Parameter **filter** unterstützt die Kriterien, die im VMware-Datenobjekt DistributedVirtualSwitchPortCriteria verfügbar sind.

Beispiel:

```
vmware.dvswitch.fetchports.get[{$VMWARE.URL},{$VMWARE.DVS.UUID},"connected:true,active:true,uplinkPort:fal
```

vmware.hv.alarms.get[url,uuid]

 Die Alarmdaten des VMware-Hypervisors.
 Rückgabewert: JSON object.

Parameter:

- **url** - die VMware-Dienst-URL;
- **uuid** - die globale eindeutige Kennung des VMware-Hypervisors.

vmware.hv.cluster.name[url,uuid]

 Der Name des VMware-Hypervisor-Clusters.
 Rückgabewert: *String*.

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors.

vmware.hv.connectionstate[url,uuid]

 Der Verbindungsstatus des VMware-Hypervisors.
 Rückgabewert: *String*: *connected*, *disconnected* oder *notResponding*.

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors.

vmware.hv.cpu.usage[url,uuid]

 Die Prozessorauslastung des VMware-Hypervisors (Hz).
 Rückgabewert: *Integer*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors.

vmware.hv.cpu.usage.perf[url,uuid]

 Die Prozessorauslastung des VMware-Hypervisors als Prozentsatz während des Intervalls.
 Rückgabewert: *Float*.

Parameter:

- **url** - die URL des VMware-Dienstes;

- **uuid** - die global eindeutige Kennung des VMware-Hypervisors.

vmware.hv.cpu.utilization[url,uuid]

 Die Prozessorauslastung des VMware-Hypervisors als Prozentsatz während des Intervalls; hängt von der Energieverwaltung oder HT ab.
 Rückgabewert: *Float*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors.

vmware.hv.datacenter.name[url,uuid]

 Der Name des VMware-Hypervisor-Datacenters.
 Rückgabewert: *String*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors.

vmware.hv.datastore.discovery[url,uuid]

 Die Discovery von VMware-Hypervisor-Datenspeichern.
 Rückgabewert: JSON-Objekt.

Parameter:

- **url** - die URL des VMware-Service;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors.

vmware.hv.datastore.list[url,uuid]

 Die Liste der VMware-Hypervisor-Datenspeicher.
 Rückgabewert: *String*.

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors.

Ausgabebeispiel:

```
SSD-RAID1-VAULT1
SSD-RAID1-VAULT2
SSD-RAID10
```

vmware.hv.datastore.multipath[url,uuid,<datastore>,<partitionid>]

 Die Anzahl der verfügbaren Datastore-Pfade.
 Rückgabewert: *Integer*.

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors;
- **datastore** - die UUID oder der Name des Datastores;
- **partitionid** - die interne ID des physischen Geräts aus `vmware.hv.datastore.discovery`.

vmware.hv.datastore.read[url,uuid,datastore,<mode>]

 Die durchschnittliche Dauer eines Lesevorgangs aus dem Datenspeicher (Millisekunden).
 Rückgabewert: *Integer* [2].

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors;
- **datastore** - die UUID oder der Name des Datenspeichers;
- **mode** - *latency* (Standard).

vmware.hv.datastore.size[url,uuid,datastore,<mode>]

 Der Speicherplatz des VMware-Datastores in Byte oder als Prozentsatz des Gesamtwerts.
 Rückgabewert: *Integer* - für Byte; *Float* - für Prozentangaben.

Parameter:

- **url** - die URL des VMware-Service;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors;
- **datastore** - die UUID oder der Name des Datastores;
- **mode** - mögliche Werte: *total* (Standard), *free*, *pfree* (freier Prozentsatz), *uncommitted*.

vmware.hv.datastore.write[url,uuid,datastore,<mode>]

 Die durchschnittliche Dauer eines Schreibvorgangs in den Datenspeicher (Millisekunden).
 Rückgabewert: *Integer* [2].

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors;
- **datastore** - die UUID oder der Name des Datenspeichers;
- **mode** - *latency* (Standard).

vmware.hv.discoveryurl

 Die Erkennung von VMware-Hypervisoren.
 Rückgabewert: JSON-Objekt.

Parameter:

- **url** - die URL des VMware-Dienstes.

vmware.hv.diskinfo.get[url,uuid]

 Die VMware-Hypervisor-Diskdaten.
 Rückgabewert: JSON object.

Parameter:

- **url** - die VMware-Dienst-URL;
- **uuid** - die globale eindeutige Kennung des VMware-Hypervisors.

vmware.hv.fullname[url,uuid]

 Der vollständige Produktname einschließlich Versionsinformationen.
 Rückgabewert: *String*.

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors.

vmware.hv.hw.cpu.freq[url,uuid]

 Die Prozessorfrequenz des VMware-Hypervisors (Hz).
 Rückgabewert: *Integer*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors.

vmware.hv.hw.cpu.model[url,uuid]

 Das Prozessormodell des VMware-Hypervisors.
 Rückgabewert: *String*.

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors.

vmware.hv.hw.cpu.num[url,uuid]

 Die Anzahl der Prozessorkerne auf dem VMware-Hypervisor.
 Rückgabewert: *Integer*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors.

vmware.hv.hw.cpu.threads[url,uuid]

 Die Anzahl der Prozessor-Threads auf dem VMware-Hypervisor.
 Rückgabewert: *Integer*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors.

vmware.hv.hw.memory[url,uuid]

 Die gesamte Speichergröße des VMware-Hypervisors (Bytes).
 Rückgabewert: *Integer*.

Parameter:

- **url** - die URL des VMware-Service;

- **uuid** - die global eindeutige Kennung des VMware-Hypervisors.

vmware.hv.hw.model[url,uuid]

 Das VMware-Hypervisor-Modell.
 Rückgabewert: *String*.

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors.

vmware.hv.hw.sensors.get[url,uuid]

 Der Wert der Hardware-Sensoren des VMware-Hypervisors.
 Rückgabewert: JSON-Objekt.

Parameter:

- **url** - die VMware-Dienst-URL;
- **uuid** - die globale eindeutige Kennung des VMware-Hypervisors.

vmware.hv.hw.serialnumber[url,uuid]

 Die Seriennummer des VMware-Hypervisors.
 Rückgabewert: *String*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors.

Dieser Datenpunkt funktioniert mit der vSphere API 6.7 und neuer.

vmware.hv.hw.uuid[url,uuid]

 Die BIOS-UUID des VMware-Hypervisors.
 Rückgabewert: *String*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors.

Dieser Datenpunkt funktioniert mit der vSphere API 6.7 und neuer.

vmware.hv.hw.vendor[url,uuid]

 Der Herstellername des VMware-Hypervisors.
 Rückgabewert: *String*.

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors.

Dieser Datenpunkt funktioniert mit der vSphere-API 6.7 und neuer.

vmware.hv.maintenance[url,uuid]

 Der Wartungsstatus des VMware-Hypervisors.
 Rückgabewert: *0* - nicht im Wartungsmodus; *1* - im Wartungsmodus.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors.

vmware.hv.memory.size.ballooned[url,uuid]

 Die Größe des vom VMware-Hypervisor ballonierten Speichers (in Byte).
 Rückgabewert: *Integer*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors.

vmware.hv.memory.used[url,uuid]

 Die Größe des vom VMware-Hypervisor verwendeten Speichers (in Byte).
 Rückgabewert: *Integer*.

Parameter:

- **url** - die URL des VMware-Service;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors.

vmware.hv.net.if.discovery[url,uuid]

 Die Erkennung von Netzwerk-Interfaces des VMware-Hypervisors.
 Rückgabewert: *JSON-Objekt*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors.

vmware.hv.network.in[url,uuid,<mode>]

 Die Netzwerk-Eingangsstatistiken des VMware-Hypervisors (Bytes pro Sekunde).
 Rückgabewert: *Integer* [2].

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors;
- **mode** - *bps* (Standard), *packets*, *dropped*, *errors*, *broadcast*.

vmware.hv.network.linkspeed[url,uuid,ifname]

 Die Geschwindigkeit der Netzwerkschnittstelle des VMware-Hypervisors.
 Rückgabewert: *Integer*. Gibt *0* zurück, wenn die Netzwerkschnittstelle inaktiv ist, andernfalls den Geschwindigkeitswert der Schnittstelle.

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors;
- **ifname** - der Name der Schnittstelle.

vmware.hv.network.out[url,uuid,<mode>]

 Die Netzwerk-Ausgangsstatistiken des VMware-Hypervisors (Bytes pro Sekunde).
 Rückgabewert: *Integer* [2].

Parameter:

- **url** - die URL des VMware-Service;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors;
- **mode** - *bps* (Standard), *packets*, *dropped*, *errors*, *broadcast*.

vmware.hv.perfcounter[url,uuid,path,<instance>]

 Der Wert des VMware-Hypervisor-Performance-Counters.
 Rückgabewert: *Integer* [2].

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors;
- **path** - der Pfad des Performance-Counters [1];
- **instance** - die Instanz des Performance-Counters. Verwenden Sie eine leere Instanz für aggregierte Werte (Standard).

vmware.hv.property[url,uuid,prop]

 Die VMware-Hypervisor-Eigenschaft.
 Rückgabewert: *String*.

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors;
- **prop** - der Eigenschaftspfad.

vmware.hv.power[url,uuid,<max>]

 Der Stromverbrauch des VMware-Hypervisors (W).
 Rückgabewert: *Integer*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors;
- **max** - der maximal zulässige Stromverbrauch.

vmware.hv.sensor.health.state[url,uuid]

 Der Sensor für den zusammengefassten Integritätsstatus des VMware-Hypervisors.
 Rückgabewert: *Integer*: *0* - grau; *1* - grün; *2* - gelb; *3* - rot.

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors.

Beachten Sie, dass der Datenpunkt in VMware vSphere 6.5 und neuer möglicherweise nicht funktioniert, da VMware den Sensor *VMware Rollup Health State* als veraltet eingestuft hat.

vmware.hv.sensors.get[url,uuid]

 Die HW-Hersteller-Statussensoren des VMware-Hypervisors.
 Rückgabewert: JSON-Objekt.

Parameter:

- **url** - die VMware-Dienst-URL;
- **uuid** - die globale eindeutige Kennung des VMware-Hypervisors.

vmware.hv.status[url,uuid]

 Der Status des VMware-Hypervisors.
 Rückgabewert: *Integer*: *0* - grau; *1* - grün; *2* - gelb; *3* - rot.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors.

Dieser Datenpunkt verwendet die Eigenschaft für den Gesamtstatus des Host-Systems.

vmware.hv.tags.get[url,uuid]

 Das Array der VMware-Hypervisor-Tags.
 Rückgabewert: JSON object.

Parameter:

- **url** - die VMware-Dienst-URL;
- **uuid** - der global eindeutige Bezeichner des VMware-Hypervisors.

Dieser Datenpunkt funktioniert mit vSphere 6.5 und neuer.

vmware.hv.uptime[url,uuid]

 Die Betriebszeit des VMware-Hypervisors (in Sekunden).
 Rückgabewert: *Integer*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors.

Dieser Datenpunkt verwendet die Eigenschaft für den Gesamtstatus des Host-Systems.

vmware.hv.version[url,uuid]

 Die VMware-Hypervisor-Version.
 Rückgabewert: *String*.

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors.

vmware.hv.vm.num[url,uuid]

 Die Anzahl der virtuellen Maschinen auf dem VMware-Hypervisor.
 Rückgabewert: *Integer*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung des VMware-Hypervisors.

vmware.rp.cpu.usage[url,rpid]

 Die CPU-Auslastung in Hertz während des Intervalls im VMware-Ressourcenpool.
 Rückgabewert: *Integer*.

Parameter:

- **url** - die VMware-Service-URL;
- **rpid** - die ID des VMware-Ressourcenpools.

vmware.rp.memory[url,rpid,<mode>]

 Die Speichermetriken des VMware-Ressourcenpools.
 Rückgabewert: *Integer*.

Parameter:

- **url** - die VMware-Service-URL;
- **rpid** - die VMware-Ressourcenpool-ID;
- **mode** - mögliche Werte:
consumed (Standard) - die Menge an physischem Host-Speicher, die zur Absicherung der physischen Gastspeicherseiten verbraucht wird
 ballooned - die Menge an physischem Gastspeicher, die der virtuellen Maschine durch den Balloon-Treiber im Gast entzogen wurde
overhead - der physische Host-Speicher, der von ESXi-Datenstrukturen für den Betrieb der virtuellen Maschinen verbraucht wird

vmware.alarms.geturl

 Die Alarmdaten des VMware Virtual Centers.
 Rückgabewert: JSON object.

Parameter:

- **url** - die VMware-Dienst-URL.

vmware.vm.alarms.get[url,uuid]

 Die Alarmdaten der VMware-virtuellen Maschine.
 Rückgabewert: JSON object.

Parameter:

- **url** - die VMware-Dienst-URL;
- **uuid** - die globale eindeutige Kennung der VMware-virtuellen Maschine.

vmware.vm.attribute[url,uuid,name]

 Der Wert des benutzerdefinierten Attributs der virtuellen VMware-Maschine.
 Rückgabewert: *String*.

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine;
- **name** - der Name des benutzerdefinierten Attributs.

vmware.vm.cluster.name[url,uuid]

 Der Name der virtuellen VMware-Maschine.
 Rückgabewert: *String*.

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine.

vmware.vm.consolidationneeded[url,uuid]

 Die Festplatte der virtuellen VMware-Maschine erfordert eine Konsolidierung.
 Rückgabewert: *String*: *true* - Konsolidierung ist erforderlich; *false* - Konsolidierung ist nicht erforderlich.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine.

vmware.vm.cpu.latency[url,uuid]

 Der Prozentsatz der Zeit, in der die virtuelle Maschine nicht ausgeführt werden kann, weil sie um den Zugriff auf die physische(n) CPU(s) konkurriert.
 Rückgabewert: *Float*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine.

vmware.vm.cpu.num[url,uuid]

 Die Anzahl der Prozessoren auf der virtuellen VMware-Maschine.
 Rückgabewert: *Integer*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine.

vmware.vm.cpu.readiness[url,uuid,<instance>]

 Der prozentuale Anteil der Zeit, in der die virtuelle Maschine bereit war, aber nicht zur Ausführung auf der physischen CPU eingeplant werden konnte.
 Rückgabewert: *Float*.

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine;
- **instance** - die CPU-Instanz.

vmware.vm.cpu.ready[url,uuid]

 Die Zeit (in Millisekunden), in der die virtuelle Maschine bereit war, aber nicht für die Ausführung auf der physischen CPU eingeplant werden konnte. Die CPU-Ready-Zeit hängt von der Anzahl der virtuellen Maschinen auf dem Host und deren CPU-Last (%) ab.
 Rückgabewert: *Integer* [2].

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine.

vmware.vm.cpu.swapwait[url,uuid,<instance>]

 Der Prozentsatz der CPU-Zeit, der auf das Einlagern aus dem Swap-Speicher wartet.
 Rückgabewert: *Float*.

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine;
- **instance** - die CPU-Instanz.

vmware.vm.cpu.usage[url,uuid]

 Die Prozessorauslastung (Hz) der virtuellen VMware-Maschine.
 Rückgabewert: *Integer*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine.

vmware.vm.cpu.usage.perf[url,uuid]

 Die Prozessorauslastung der virtuellen VMware-Maschine als Prozentsatz während des Intervalls.
 Rückgabewert: *Float*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine.

vmware.vm.datacenter.name[url,uuid]

 Der Name des Datacenters der virtuellen VMware-Maschine.
 Rückgabewert: *String*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine.

vmware.vm.discoveryurl

 Die Discovery von virtuellen VMware-Maschinen.
 Rückgabewert: JSON-Objekt.

Parameter:

- **url** - die URL des VMware-Dienstes.

vmware.vm.guest.memory.size.swapped[url,uuid]

 Die Menge des physischen Speichers des Gasts, die in den Auslagerungsspeicher ausgelagert wird (KB).
 Rückgabewert: *Integer*.

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine.

vmware.vm.guest.osuptime[url,uuid]

 Die gesamte seit dem letzten Start des Betriebssystems verstrichene Zeit (in Sekunden).
 Rückgabewert: *Integer*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine.

vmware.vm.hv.maintenance[url,uuid]

 Der Wartungsstatus des Hypervisors der virtuellen VMware-Maschine.
 Rückgabewert: *0* - nicht in Wartung; *1* - in Wartung.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine.

vmware.vm.hv.name[url,uuid]

 Der Name des Hypervisors der virtuellen VMware-Maschine.
 Rückgabewert: *String*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine.

vmware.vm.memory.size[url,uuid]

 Die Gesamtgröße des Arbeitsspeichers der virtuellen VMware-Maschine (Bytes).
 Rückgabewert: *Integer*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine.

vmware.vm.memory.size.ballooned[url,uuid]

 Die Größe des aufgeblähten Speichers der virtuellen VMware-Maschine (in Byte).
 Rückgabewert: *Integer*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine.

vmware.vm.memory.size.compressed[url,uuid]

 Die Größe des komprimierten Speichers der virtuellen VMware-Maschine (in Byte).
 Rückgabewert: *Integer*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine.

vmware.vm.memory.size.consumed[url,uuid]

 Die Menge des physischen Speichers des Hosts, die zum Sichern der physischen Speicherseiten des Gasts verwendet wird (KB).
 Rückgabewert: *Integer*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine.

vmware.vm.memory.size.private[url,uuid]

 Die Größe des privaten Speichers der virtuellen VMware-Maschine (Bytes).
 Rückgabewert: *Integer*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine.

vmware.vm.memory.size.shared[url,uuid]

 Die Größe des gemeinsam genutzten Speichers der virtuellen VMware-Maschine (Bytes).
 Rückgabewert: *Integer*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine.

vmware.vm.memory.size.swapped[url,uuid]

 Die Größe des ausgelagerten Speichers der virtuellen VMware-Maschine (Bytes).
 Rückgabewert: *Integer*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine.

vmware.vm.memory.size.usage.guest[url,uuid]

 Die Speichernutzung (Bytes) des Gasts der virtuellen VMware-Maschine.
 Rückgabewert: *Integer*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine.

vmware.vm.memory.size.usage.host[url,uuid]

 Die Speichernutzung des VMware-Hosts der virtuellen Maschine (Bytes).
 Rückgabewert: *Integer*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der VMware-virtuellen Maschine.

vmware.vm.memory.usage[url,uuid]

 Der Prozentsatz des physischen Speichers des Hosts, der verbraucht wurde.
 Rückgabewert: *Float*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine.

vmware.vm.net.if.discovery[url,uuid]

 Die Discovery von VMware-Netzwerkschnittstellen virtueller Maschinen.
 Rückgabewert: JSON-Objekt.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine.

vmware.vm.net.if.in[url,uuid,instance,<mode>]

 Die Eingangsstatistiken der Netzwerkschnittstelle der virtuellen VMware-Maschine (Bytes/Pakete pro Sekunde).
 Rückgabewert: *Integer* [2].

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine;
- **instance** - die Netzwerkschnittstelleninstanz;
- **mode** - *bps* (Standard) oder *pps* - Bytes oder Pakete pro Sekunde.

vmware.vm.net.if.out[url,uuid,instance,<mode>]

 Die Ausgabestatistik der Netzwerkschnittstelle der virtuellen VMware-Maschine (Bytes/Pakete pro Sekunde).
 Rückgabewert: *Integer* [2].

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine;
- **instance** - die Instanz der Netzwerkschnittstelle;
- **mode** - *bps* (Standard) oder *pps* - Bytes oder Pakete pro Sekunde.

vmware.vm.net.if.usage[url,uuid,<instance>]

 Die Netzwerkauslastung der virtuellen VMware-Maschine (kombinierte Sende- und Empfangsraten) während des Intervalls (KBps).
 Rückgabewert: *Integer*.

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine;
- **instance** - die Instanz der Netzwerkschnittstelle.

vmware.vm.perfcounter[url,uuid,path,<instance>]

 Der Wert des Leistungszählers der virtuellen VMware-Maschine.
 Rückgabewert: *Integer* [2].

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine;
- **path** - der Pfad des Leistungszählers [1];
- **instance** - die Instanz des Leistungszählers. Verwenden Sie eine leere Instanz für aggregierte Werte (Standard).

vmware.vm.powerstate[url,uuid]

 Der Energiezustand der virtuellen VMware-Maschine.
 Rückgabewert: *0* - poweredOff; *1* - poweredOn; *2* - suspended.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine.

vmware.vm.property[url,uuid,prop]

 Die VMware-Eigenschaft der virtuellen Maschine.
 Rückgabewert: *String*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine;
- **prop** - der Eigenschaftspfad.

vmware.vm.snapshot.get[url,uuid]

 Der Snapshot-Status der VMware-virtuellen Maschine.
 Rückgabewert: JSON-Objekt.

Parameter:

- **url** - die VMware-Dienst-URL;
- **uuid** - die globale eindeutige Kennung der VMware-virtuellen Maschine.

vmware.vm.state[url,uuid]

 Der Status der virtuellen VMware-Maschine.
 Rückgabewert: *String*: *notRunning*, *resetting*, *running*, *shuttingDown*, *standby* oder *unknown*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine.

vmware.vm.storage.committed[url,uuid]

 Der von der virtuellen VMware-Maschine belegte Speicherplatz (Bytes).
 Rückgabewert: *Integer*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine.

vmware.vm.storage.readoio[url,uuid,instance]

 Die durchschnittliche Anzahl ausstehender Leseanforderungen an die virtuelle Festplatte während des Erfassungsintervalls.
 Rückgabewert: *Integer*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine;
- **instance** - die Instanz des Festplattengeräts.

vmware.vm.storage.totalreadlatency[url,uuid,instance]

 Die durchschnittliche Zeit, die ein Lesevorgang von der virtuellen Festplatte benötigt (Millisekunden).
 Rückgabewert: *Integer*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine;
- **instance** - die Instanz des Festplattengeräts.

vmware.vm.storage.totalwritelatency[url,uuid,instance]

 Die durchschnittliche Zeit, die ein Schreibvorgang auf die virtuelle Festplatte benötigt (Millisekunden).
 Rückgabewert: *Integer*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine;
- **instance** - die Instanz des Festplattengeräts.

vmware.vm.storage.uncommitted[url,uuid]

 Der nicht zugewiesene Speicherplatz der virtuellen VMware-Maschine (in Byte).
 Rückgabewert: *Integer*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine.

vmware.vm.storage.unshared[url,uuid]

 Der nicht gemeinsam genutzte Speicherplatz der virtuellen VMware-Maschine (Bytes).
 Rückgabewert: *Integer*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine.

vmware.vm.storage.writeoio[url,uuid,instance]

 Die durchschnittliche Anzahl ausstehender Schreibanforderungen an die virtuelle Festplatte während des Erfassungsintervalls.
 Rückgabewert: *Integer*.

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine;
- **instance** - die Instanz des Festplattengeräts.

vmware.vm.tags.get[url,uuid]

 Das Array der VMware-Tag-Zuordnungen für virtuelle Maschinen.
 Rückgabewert: JSON object.

Parameter:

- **url** - die VMware-Dienst-URL;
- **uuid** - der global eindeutige Bezeichner der VMware-virtuellen Maschine.

Dieser Datenpunkt funktioniert mit vSphere 6.5 und neuer.

vmware.vm.tools[url,uuid,<mode>]

 Der Status der Gast-Tools der virtuellen VMware-Maschine.
 Rückgabewert: *String*.
 Im Modus *status*: *guestToolsExecutingScripts* - VMware Tools wird gestartet; *guestToolsNotRunning* - VMware Tools wird nicht ausgeführt; *guestToolsRunning* - VMware Tools wird ausgeführt.
 Im Modus *version*: Version.

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine;
- **mode** - *version*, *status* (Standard).

vmware.vm.uptime[url,uuid]

 Die Betriebszeit der virtuellen VMware-Maschine (Sekunden).
 Rückgabewert: *Integer*.

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine.

vmware.vm.vfs.dev.discovery[url,uuid]

 Die Discovery von VMware-Datenträgergeräten virtueller Maschinen.
 Rückgabewert: JSON-Objekt.

Parameter:

- **url** - die URL des VMware-Dienstes;

- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine.

vmware.vm.vfs.dev.read[url,uuid,instance,<mode>]

 Die Lesestatistiken des Festplattengeräts einer virtuellen VMware-Maschine (Bytes/Operationen pro Sekunde).
 Rückgabewert: *Integer* [2].

Parameter:

- **url** - die URL des VMware-Dienstes;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine;
- **instance** - die Instanz des Festplattengeräts;
- **mode** - *bps* (Standard) oder *ops* - Bytes bzw. Operationen pro Sekunde.

vmware.vm.vfs.dev.write[url,uuid,instance,<mode>]

 Die Schreibstatistik des Festplattengeräts der virtuellen VMware-Maschine (Bytes/Operationen pro Sekunde).
 Rückgabewert: *Integer* [2].

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine;
- **instance** - die Instanz des Festplattengeräts;
- **mode** - *bps* (Standard) oder *ops* - Bytes oder Operationen pro Sekunde.

vmware.vm.vfs.fs.discovery[url,uuid]

 Die Erkennung von Dateisystemen virtueller VMware-Maschinen.
 Rückgabewert: JSON-Objekt.

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine.

VMware Tools müssen auf der virtuellen Gastmaschine installiert sein, damit dieser Datenpunkt funktioniert.

vmware.vm.vfs.fs.size[url,uuid,fsname,<mode>]

 Die Dateisystemstatistiken der virtuellen VMware-Maschine (Bytes/Prozentwerte).
 Rückgabewert: *Integer* - für Bytes; *Float* - für Prozentwerte.

Parameter:

- **url** - die VMware-Service-URL;
- **uuid** - die global eindeutige Kennung der virtuellen VMware-Maschine;
- **fsname** - der Dateisystemname;
- **mode** - *total*, *free*, *used*, *pfree* oder *pused*.

Damit dieser Datenpunkt funktioniert, müssen VMware Tools auf der virtuellen Gastmaschine installiert sein.

Fußnoten

[1] Siehe Benutzerdefinierte Namen für Leistungsindikatoren für VMware erstellen.

[2] Der Wert dieser Datenpunkte wird aus VMware-Leistungsindikatoren ermittelt, und der Parameter VMwarePerfFrequency parameter wird verwendet, um ihre Daten im Zabbix-VMware-Cache zu aktualisieren:

- 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

Weitere Informationen

Siehe Überwachung virtueller Maschinen für detaillierte Informationen dazu, wie Zabbix zur Überwachung von VMware-Umgebungen konfiguriert wird.

Felder des Erkennungs-Schlüssels für virtuelle Maschinen

Die folgende Tabelle listet die Felder auf, die von Erkennungs-Schlüsseln im Zusammenhang mit virtuellen Maschinen zurückgegeben werden.

Item key		
Description	**Field**	**Retrieved content**
vmware.cluster.discovery		
Führt Cluster-Erkennung durch.	{#CLUSTER.ID}	Cluster-Identifikator.
	{#CLUSTER.NAME}	Clustername.

	”resource_pool”	Ein Array mit Resource-Pool-Daten, einschließlich Resource-Group-ID, Tags-Array, Resource-Pool-Pfad und Anzahl der virtuellen Maschinen.
		
		Array-Struktur:
		`[{`
		`"rpid":"resource group id",`
		`"tags":[{}],`
		`"rpath":"resource group path",`
		`"vm_count":0`
		`}]`

	”tags”	Zur Array-Struktur von `"tags"` siehe das Feld ”tags”.
		Ein Array mit Tags, einschließlich Tag-Name, Beschreibung und Kategorie.
		
		Array-Struktur:
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

vmware.datastore.discovery		
Führt Datastore-Erkennung durch.	{#DATASTORE}	Datastore-Name.

	{#DATASTORE.EXTENT}	Ein Array mit der Partition-ID des Datastore-Extents und dem Instanznamen.
		
		Array-Struktur:
		`[{`
		`"partitionid":1,`
		`"instance":"name"`
		`}]`

	{#DATASTORE.TYPE}	Datastore-Typ. Beispielwerte: `VMFS`, `NFS`, `vsan` usw.
	{#DATASTORE.UUID}	Datastore-Identifikator.

	”tags”	Ein Array mit Tags, einschließlich Tag-Name, Beschreibung und Kategorie.
		
		Array-Struktur:
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

vmware.dc.discovery		

Item key		
Führt Datacenter-Erkennung durch.	{#DATACENTER}	Datacenter-Name.
	{#DATACENTERID}	Datacenter-Identifikator.

	”tags”	Ein Array mit Tags, einschließlich Tag-Name, Beschreibung und Kategorie.
		
		Array-Struktur:
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

vmware.dvswitch.discovery		
Führt die Erkennung verteilter vSphere-Switches durch.	{#DVS.NAME}	Switch-Name.
	{#DVS.UUID}	Switch-Identifikator.
vmware.hv.discovery		
Führt Hypervisor-Erkennung durch.	{#HV.UUID}	Eindeutiger Hypervisor-Identifikator.
	{#HV.ID}	Hypervisor-Identifikator (HostSystem-Managed-Object-Name).
	{#HV.NAME}	Hypervisor-Name.
	{#HV.NETNAME}	Netzwerk-Hostname des Hypervisors.
	{#HV.IP}	IP-Adresse des Hypervisors, kann leer sein. Bei einer HA-Konfiguration mit mehreren Netzwerkschnittstellen wird die folgende Priorität bei der Auswahl der Schnittstelle beachtet: - bevorzugt die IP, die dasselbe IP-Subnetz wie die vCenter-IP verwendet; - bevorzugt die IP aus dem IP-Subnetz mit Standardgateway; - bevorzugt die IP der Schnittstelle mit der niedrigsten ID.
	{#CLUSTER.NAME}	Clustername, kann leer sein.
	{#DATACENTER.NAME}	Datacenter-Name.
	{#PARENT.NAME}	Name des Containers, in dem der Hypervisor gespeichert ist.
	{#PARENT.TYPE}	Typ des Containers, in dem der Hypervisor gespeichert ist. Mögliche Werte sind `Datacenter`, `Folder`, `ClusterComputeResource`, `VMware`, wobei ”VMware” für einen unbekannten Containertyp steht.

	”resource_pool”	Ein Array mit Resource-Pool-Daten, einschließlich Resource-Group-ID, Tags-Array, Resource-Pool-Pfad und Anzahl der virtuellen Maschinen.
		
		Array-Struktur:
		`[{`
		`"rpid":"resource group id",`
		`"tags":[{}],`
		`"rpath":"resource group path",`
		`"vm_count":0`
		`}]`
		
		Zur Array-Struktur von `"tags"` siehe das Feld ”tags”.

	”tags”	Ein Array mit Tags, einschließlich Tag-Name, Beschreibung und Kategorie.
		
		Array-Struktur:
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

Item key		
vmware.hv.datastore.discovery		
Führt die Datastore-Erkennung für Hypervisoren durch. Beachten Sie, dass mehrere Hypervisoren denselben Datastore verwenden können.	{#DATASTORE}	Datastore-Name.
	{#DATASTORE.TYPE}	Datastore-Typ. Beispielwerte: `VMFS`, `NFS`, `vsan` usw.
	{#DATASTORE.UUID}	Datastore-Identifikator.
	{#MULTIPATH.COUNT}	Registrierte Anzahl der Datastore-Pfade.
	{#MULTIPATH.PARTITION.COUNT}	Anzahl der verfügbaren Festplattenpartitionen.

	"datastore_extent"	Ein Array mit dem Instanznamen und der Partition-ID des Datastore-Extents.
		
		Array-Struktur:
		`[{`
		`"partitionid":1,`
		`"instance":"name"`
		`}]`

	"tags"	Ein Array mit Tags, einschließlich Tag-Name, Beschreibung und Kategorie.
		
		Array-Struktur:
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

vmware.hv.net.if.discovery		
Führt die Erkennung der Netzwerkschnittstellen des Hypervisors durch.	{#IFNAME}	Schnittstellenname.
	{#IFDRIVER}	Schnittstellentreiber.
	{#IFDUPLEX}	Duplex-Einstellungen der Schnittstelle.
	{#IFSPEED}	Geschwindigkeit der Schnittstelle.
	{#IFMAC}	MAC-Adresse der Schnittstelle.
vmware.vm.discovery		
Führt die Erkennung virtueller Maschinen durch.	{#VM.UUID}	Eindeutiger Identifikator der virtuellen Maschine.
	{#VM.ID}	Identifikator der virtuellen Maschine (VirtualMachine-Managed-Object-Name).
	{#VM.NAME}	Name der virtuellen Maschine.
	{#HV.NAME}	Hypervisor-Name.
	{#HV.UUID}	Eindeutiger Hypervisor-Identifikator.
	{#HV.ID}	Hypervisor-Identifikator (HostSystem-Managed-Object-Name).
	{#CLUSTER.NAME}	Clustername, kann leer sein.
	{#DATACENTER.NAME}	Datacenter-Name.
	{#DATASTORE.NAME}	Datastore-Name.
	{#DATASTORE.UUID}	Datastore-Identifikator.
	{#VM.IP}	IP-Adresse der virtuellen Maschine, kann leer sein.
	{#VM.DNS}	DNS-Name der virtuellen Maschine, kann leer sein.
	{#VM.GUESTFAMILY}	OS-Familie des Gastbetriebssystems der virtuellen Maschine, kann leer sein.
	{#VM.GUESTFULLNAME}	Vollständiger Name des Gastbetriebssystems der virtuellen Maschine, kann leer sein.

Item key		
	{#VM.FOLDER}	Die Kette der übergeordneten Ordner der virtuellen Maschine kann als Wert für verschachtelte Gruppen verwendet werden; Ordnernamen werden mit ”/” verbunden. Kann leer sein.
	{#VM.TOOLS.STATUS}	Status der VMware-Tools der virtuellen Maschine.
	{#VM.POWERSTATE}	Energiezustand der VMware-virtuellen Maschine (`poweredOff`, `poweredOn` oder `suspended`).
	{#VM.RPOOL.ID}	Identifikator des Resource Pools.
	{#VM.RPOOL.PATH}	Vollständiger Resource-Pool-Pfad ohne den ”root”-Namen ”Resources”. Ordnernamen werden mit ”/” verbunden.
	{#VM.SNAPSHOT.COUNT}	Anzahl der VM-Snapshots.

	”tags”	Ein Array mit Tags, einschließlich Tag-Name, Beschreibung und Kategorie.
		
		Array-Struktur:
		`[{`
		`"tag":"tag name",`
		`"tag_description":"tag description",`
		`"category":"tag category"`
		`}]`

	”vm_customattribute”	Ein Array benutzerdefinierter Attribute der virtuellen Maschine (falls definiert).
		
		Array-Struktur:
		`[{`
		`"name":"custom field name",`
		`"value":"custom field value"`
		`}]`

	”net_if”	Ein Array der Netzwerkschnittstellen der virtuellen Maschine.
		
		Array-Struktur:
		`[{`
		`"ifname": "interface name",`
		`"ifdesc": "interface description",`
		`"ifmac": "00:00:00:00:00:00",`
		`"ifconnected": true,`
		`"iftype": "interface type",`
		`"ifbackingdevice": "interface backing device",`
		`"ifdvswitch_uuid": "interface switch uuid",`
		`"ifdvswitch_portgroup": "interface switch port group",`
		`"ifdvswitch_port": "interface switch port",`
		`"ifip": ["interface ip addresses"]`
		`}]`
		
		Zur Beschreibung der zurückgegebenen Daten siehe den Item key ”vmware.vm.net.if.discovery”.

vmware.vm.net.if.discovery		
Führt die Erkennung der Netzwerkschnittstellen virtueller Maschinen durch.	{#IFNAME}	Name der Netzwerkschnittstelle.
	{#IFDESC}	Beschreibung der Schnittstelle.
	{#IFMAC}	MAC-Adresse der Schnittstelle.
	{#IFCONNECTED}	Verbindungsstatus der Schnittstelle (`false` - getrennt; `true` - verbunden).
	{#IFTYPE}	Schnittstellentyp.
	{#IFBACKINGDEVICE}	Name des Backing-Geräts.
	{#IFDVSWITCH.UUID}	Eindeutiger Identifikator des vSphere Distributed Switch.
	{#IFDVSWITCH.PORTGROUP}	Distributed Port Group.
	{#IFDVSWITCH.PORT}	Port des vSphere Distributed Switch.

Item key		
	"ifip"	Ein Array von Schnittstellenadressen.
vmware.vm.vfs.dev.discovery		
Führt die Erkennung von Festplattengeräten virtueller Maschinen durch.	{#DISKNAME}	Name des Festplattengeräts.
vmware.vm.vfs.fs.discovery		
Führt die Erkennung von Dateisystemen virtueller Maschinen durch.	{#FSNAME}	Name des Dateisystems.

JSON-Beispiele für VMware-Datenpunkte

Übersicht Dieser Abschnitt enthält zusätzliche Informationen zu JSON-Objekten, die von verschiedenen VMware Datenpunkten zurückgegeben werden.

vmware.*.alarms.get Die Datenpunkte **vmware.alarms.get[]**, **vmware.cluster.alarms.get[]**, **vmware.datastore.alarms.get[]**, **vmware.dc.alarms.get[]**, **vmware.hv.alarms.get[]**, **vmware.vm.alarms.get[]** geben JSON-Objekte mit der folgenden Struktur zurück (die Werte werden als Beispiel angegeben):

```
{
    "alarms": [
        {
            "name": "Host connection and power state",
            "system_name": "alarm.HostConnectionStateAlarm",
            "description": "Default alarm to monitor host connection and power state",
            "enabled": true,
            "key": "alarm-1.host-2013",
            "time": "2022-06-27T05:27:38.759976Z",
            "overall_status": "red",
            "acknowledged": false,
            "entity": {
                "id": "host-2013",
                "uuid": "4201b707-f737-7cf8-881f-49215d1aad2a",
                "type": "HostSystem",
                "name": "esx-riga-01.prod.local"
            }
        },
        {
            "name": "Host memory usage",
            "system_name": "alarm.HostMemoryUsageAlarm",
            "description": "Default alarm to monitor host memory usage",
            "enabled": true,
            "key": "alarm-4.host-1004",
            "time": "2022-05-16T13:32:42.47863Z",
            "overall_status": "yellow",
            "acknowledged": false,
            "entity": {
                "id": "host-1004",
                "uuid": "5315c818-g848-8dg9-992g-50326e2bbe3a",
                "type": "HostSystem",
                "name": "esx-riga-02.prod.local",
            }
        },
        {
            // other alarms
        }
```

```
    ]
}
```

vmware.*.tags.get Die Datenpunkte **vmware.cluster.tags.get[]**, **vmware.datastore.tags.get[]**, **vmware.dc.tags.get[]**, **vmware.hv.tags.get[]**, **vmware.vm.tags.get[]** geben JSON-Objekte mit der folgenden Struktur zurück (die Werte dienen als Beispiel):

```
{
    "tags": [
        {
            "name": "Windows",
        "description": "Tag für die Kategorie Betriebssystemtyp",
        "category": "Betriebssystemtyp"
    },
    {
            "name": "SQL Server",
        "description": "Tag für die Kategorie Anwendungsname",
        "category": "Anwendungsname"
        },
    {
            // andere Tags
        }
    ]
}
```

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 Der Datenpunkt **vmware.dvswitch.fetchports.get[]** gibt JSON-Objekte mit der folgenden Struktur zurück (Werte dienen als Beispiel):

```
{
    "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"
            },
            {
                //weitere Schlüssel
            }
        ]
    }
}
```

vmware.hv.hw.sensors.get Der Datenpunkt **vmware.hv.hw.sensors.get[]** gibt JSON-Objekte mit der folgenden Struktur zurück (die Werte dienen als Beispiel):

```
{
    "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"
                 }
            },
            {
                    // andere hv hw-Sensoren
                }
        ]
        }
}
```

vmware.hv.sensors.get Der Datenpunkt **vmware.hv.sensors.get[]** gibt JSON-Objekte mit der folgenden Struktur zurück (die Werte dienen als Beispiel):

```
{
    "val":
        {
            "@type": "ArrayOfHostNumericSensorInfo", "HostNumericSensorInfo": [
                {
                    "@type": "HostNumericSensorInfo",
                    "name": "System Board 1 PwrMeter Output --- Normal",
                    "healthState":
                        {
                            "label": "Green",
                            "summary": "Sensor arbeitet unter normalen Bedingungen",
                    "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 arbeitet unter normalen Bedingungen",
                    "key": "green"
                },
                    "currentReading": "10000",
```

```
            "unitModifier": "-2",
            "baseUnits": "Watts",
            "sensorType": "power"
        },
            {
                    // andere hv-Sensoren
                }
        ]
        }
}
```

vmware.vm.snapshot.get Wenn Snapshots vorhanden sind, gibt der Datenpunkt **vmware.snapshot.get[]** ein JSON-Objekt mit der folgenden Struktur zurück (die Werte dienen als Beispiel):

```
{
  "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
}
```

Wenn kein Snapshot vorhanden ist, gibt der Datenpunkt **vmware.snapshot.get[]** ein JSON-Objekt mit leeren Werten zurück:

```
{
  "snapshot": [],
  "count": 0,
  "latestdate": null,
  "latestage": 0,
  "oldestdate": null,
  "oldestage": 0,
  "size": 0,
  "uniquesize": 0
}
```

Beispiel für die VMware-Überwachungseinrichtung

Überblick

Das folgende Beispiel beschreibt, wie Zabbix für die Überwachung von virtuellen VMware-Maschinen eingerichtet wird. Dies umfasst:

- das Erstellen eines Hosts, der Ihre VMware-Umgebung repräsentiert;
- das Erstellen einer Low-Level-Discovery-Regel, die virtuelle Maschinen in Ihrer VMware-Umgebung erkennt;
- das Erstellen eines Host-Prototyps, auf dessen Grundlage Zabbix echte Hosts für virtuelle Maschinen generiert, die von der Low-Level-Discovery-Regel erkannt wurden.

Voraussetzungen

Note:
Dieses Beispiel behandelt nicht die Konfiguration von VMware. Es wird davon ausgegangen, dass VMware bereits konfiguriert ist.

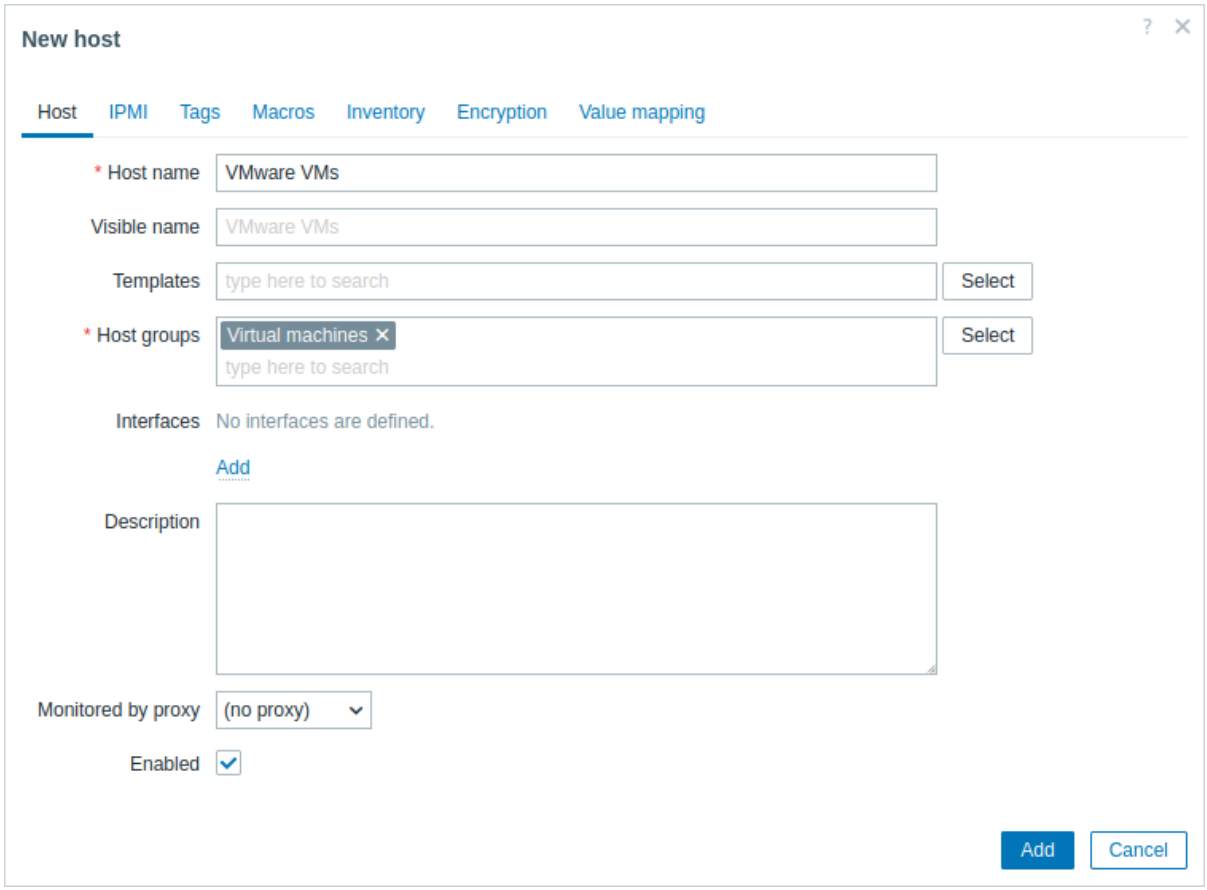
Bevor Sie fortfahren, setzen Sie den Parameter `StartVMwareCollectors` in der Konfigurationsdatei des Zabbix Server auf 2 oder mehr (der Standardwert ist 0).

Einen Host erstellen

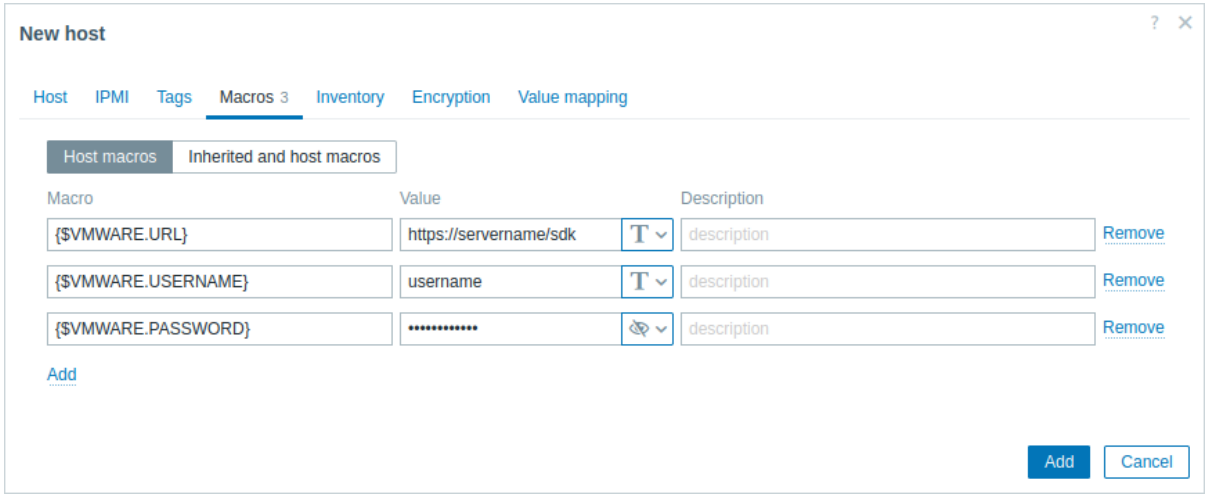
1. Gehen Sie zu *Datenerfassung* > *Hosts*.

2. Erstellen Sie einen Host:

- Geben Sie im Feld *Host name* einen Hostnamen ein (zum Beispiel "VMware VMs").
- Geben Sie im Feld *Host groups* eine Hostgruppe ein oder wählen Sie eine aus (zum Beispiel "Virtual machines").



- Legen Sie auf der Registerkarte *Macros* die folgenden Host-Makros fest:
  - {$VMWARE.URL} - SDK-URL des VMware-Dienstes (ESXi-Hypervisor) (https://servername/sdk)
  - {$VMWARE.USERNAME} - Benutzername des VMware-Dienstes
  - {$VMWARE.PASSWORD} - Passwort des Benutzers {$VMWARE.USERNAME} für den VMware-Dienst



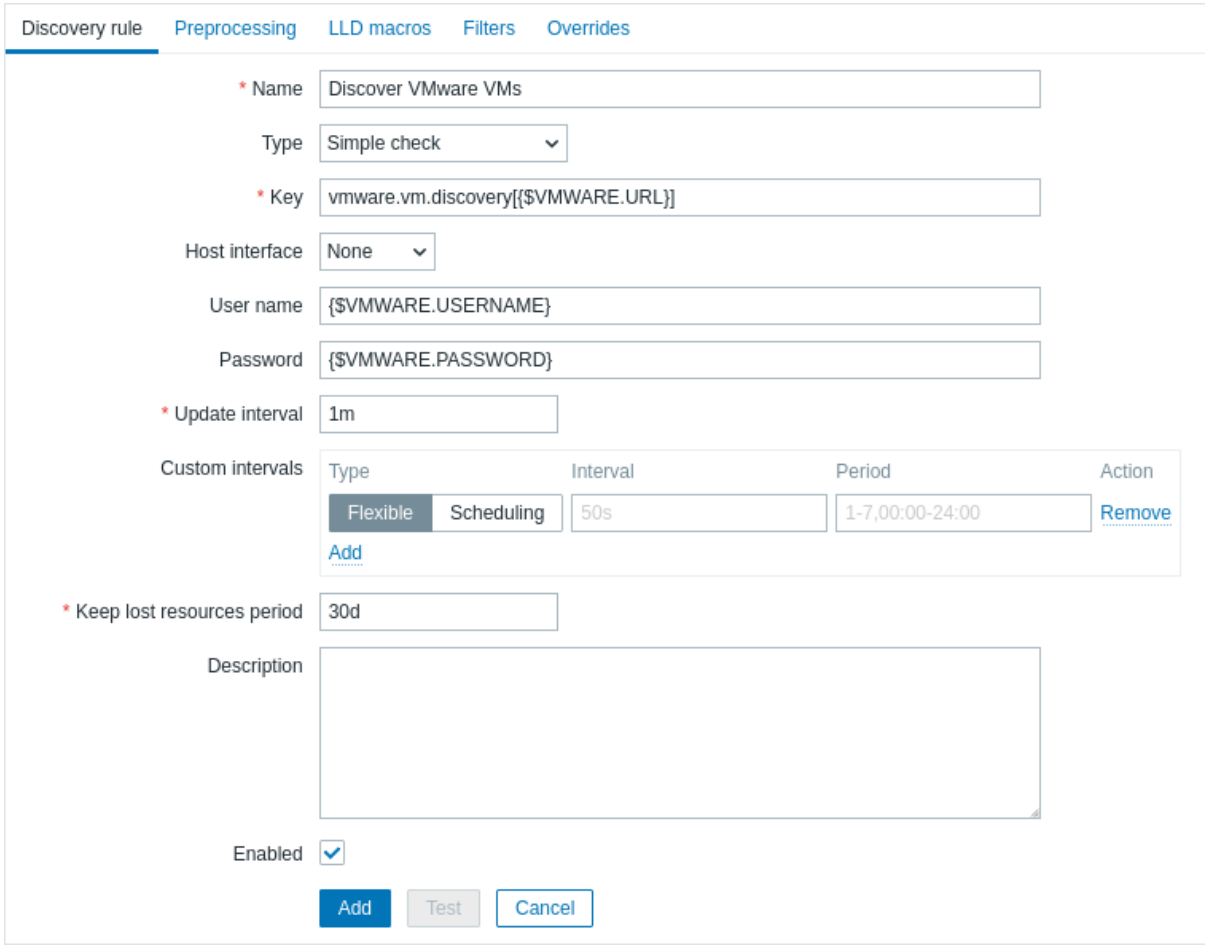
3. Klicken Sie auf die Schaltfläche *Add*, um den Host zu erstellen. Dieser Host repräsentiert Ihre VMware-Umgebung.

Eine Low-Level-Discovery-Regel erstellen

1. Klicken Sie beim erstellten Host auf *Discovery*, um zur Liste der Low-Level-Discovery-Regeln für diesen Host zu gelangen.

2. Erstellen Sie eine Low-Level-Discovery-Regel:

- Geben Sie im Feld *Name* einen Namen für die Low-Level-Discovery-Regel ein, zum Beispiel „VMware-VMs erkennen“.
- Wählen Sie im Feld *Type* „Simple check“ aus.
- Geben Sie im Feld *Key* den integrierten Datenpunkt-Schlüssel zur Erkennung von virtuellen VMware-Maschinen ein: `vmware.vm.discovery[{$VMWARE.URL}]`
- Geben Sie in den Feldern *User name* und *Password* die entsprechenden Makros ein, die zuvor auf dem Host konfiguriert wurden.



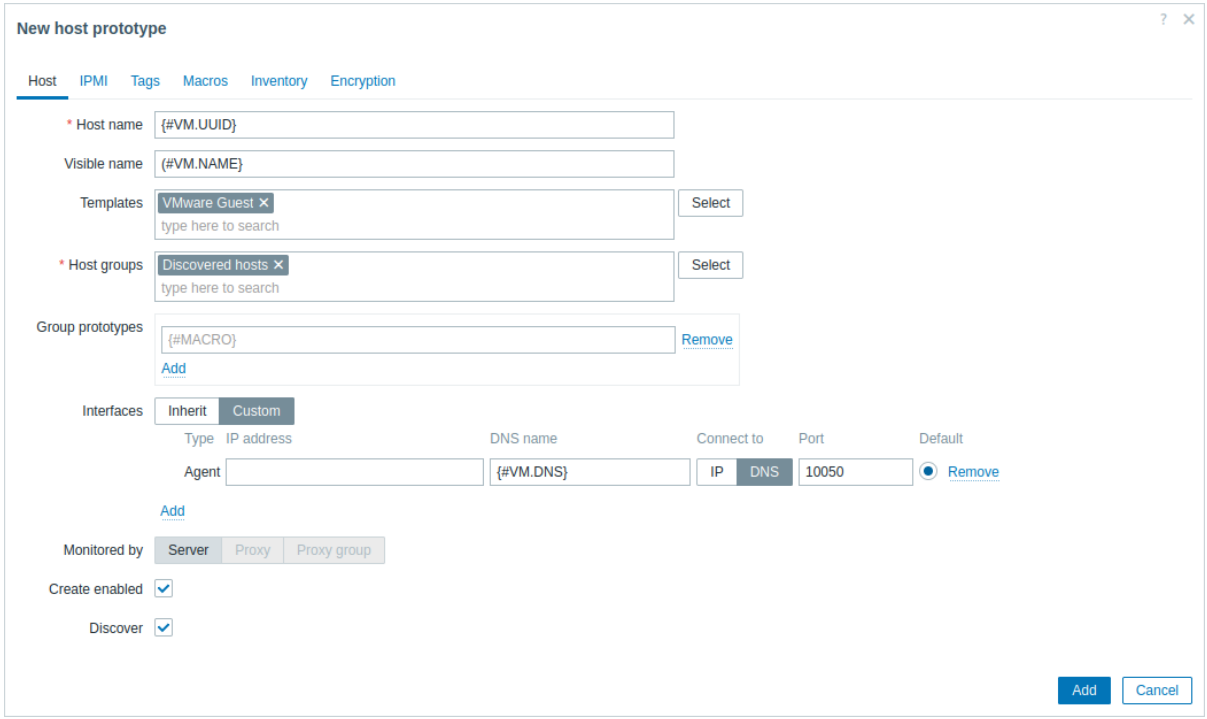
3. Klicken Sie auf die Schaltfläche *Add*, um die Low-Level-Discovery-Regel zu erstellen. Diese Discovery-Regel erkennt virtuelle Maschinen in Ihrer VMware-Umgebung.

Einen Host-Prototyp erstellen

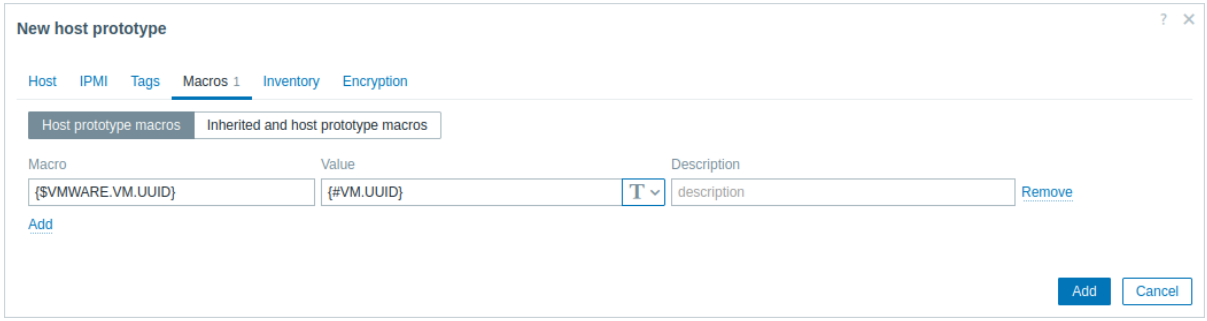
1. Klicken Sie in der Liste der Low-Level-Discovery-Regeln bei der zuvor erstellten Low-Level-Discovery-Regel auf *Host-Prototypen*.

2. Erstellen Sie einen Host-Prototyp. Da Host-Prototypen als Vorlagen für die Erstellung von Hosts über Low-Level-Discovery-Regeln dienen, enthalten die meisten Felder Low-Level-Discovery-Makros. Dadurch wird sichergestellt, dass die Hosts mit Eigenschaften erstellt werden, die auf dem abgerufenen Inhalt der zuvor erstellten Low-Level-Discovery-Regel basieren.

- Geben Sie im Feld *Host name* das Makro {#VM.UUID} ein.
- Geben Sie im Feld *Visible name* das Makro {#VM.NAME} ein.
- Geben Sie im Feld *Templates* die Vorlage "VMware Guest" ein oder wählen Sie sie aus. Diese Vorlage enthält VMware-Datenpunkte und Discovery-Regeln zur Überwachung des Energiezustands einer virtuellen Maschine, der CPU-Auslastung, der Speicherauslastung, von Netzwerkgeräten usw.
- Geben Sie im Feld *Host groups* eine Host-Gruppe ein oder wählen Sie eine aus (zum Beispiel "Discovered hosts").
- Fügen Sie im Feld *Interfaces* eine benutzerdefinierte Host-Schnittstelle hinzu. Geben Sie dann im Feld *DNS name* das Makro {#VM.DNS} ein oder im Feld *IP address* das Makro {#VM.IP}. Wenn die virtuellen Maschinen in Ihrer VMware-Umgebung alternativ mehrere Schnittstellen haben, fahren Sie mit dem Abschnitt *Erweiterte Host-Schnittstellenkonfiguration* fort. Die Konfiguration einer benutzerdefinierten Host-Schnittstelle ist für die korrekte Funktion der Vorlage *VMware Guest* erforderlich.



- Legen Sie auf der Registerkarte *Macros* das Makro {$VMWARE.VM.UUID} mit dem Wert {#VM.UUID} fest. Dies ist für die korrekte Funktion der Vorlage *VMware Guest* erforderlich, die dieses Makro als User-Makro auf Host-Ebene in Datenpunkt-Parametern verwendet (zum Beispiel *vmware.vm.net.if.discovery[{$VMWARE.URL}, {$VMWARE.VM.UUID}]*).

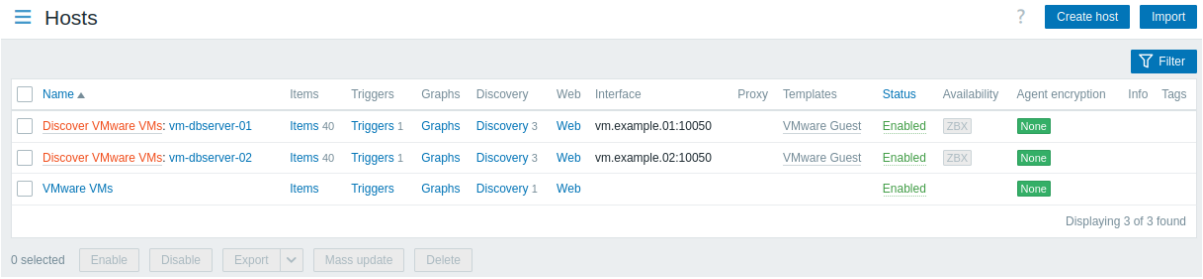


3. Klicken Sie auf die Schaltfläche *Add*, um den Host-Prototyp zu erstellen. Dieser Host-Prototyp wird verwendet, um Hosts für virtuelle Maschinen zu erstellen, die durch die zuvor erstellte Low-Level-Discovery-Regel erkannt wurden.

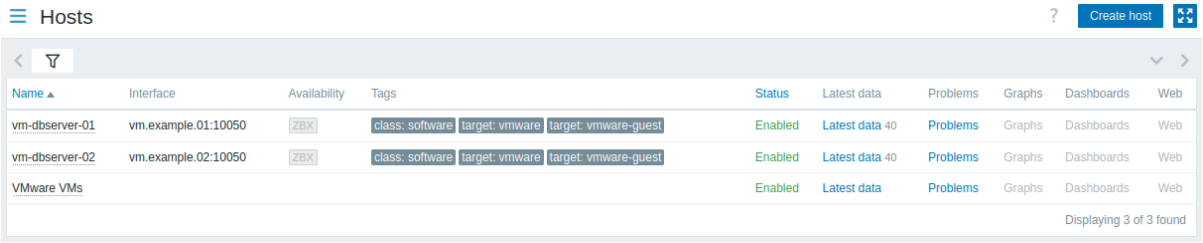
Hosts und Metriken anzeigen

Nachdem die Hostvorlage erstellt wurde, erstellt die Low-Level-Discovery-Regel Hosts für erkannte VMware-VMs, und Zabbix beginnt, diese zu überwachen. Beachten Sie, dass die Erkennung und Erstellung von Hosts bei Bedarf auch manuell ausgeführt werden kann.

Um die erstellten Hosts anzuzeigen, navigieren Sie zum Menübereich *Datenerfassung > Hosts*.



Um die erfassten Metriken anzuzeigen, navigieren Sie zum Menübereich *Überwachung > Hosts* und klicken Sie bei einem der Hosts auf *Neueste Daten*.



Erweiterte Konfiguration der Host-Schnittstelle

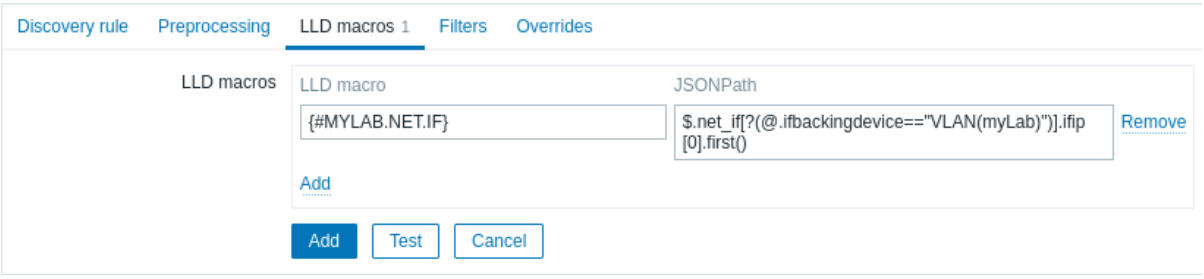
Der im Abschnitt *Eine Low-Level-Discovery-Regel erstellen* konfigurierte Datenpunktschlüssel `vmware.vm.discovery[{$VMWARE.URL}]` gibt Daten zu Netzwerkschnittstellen im Feld "net_if" zurück:

```
"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": []
      }
    ]
```

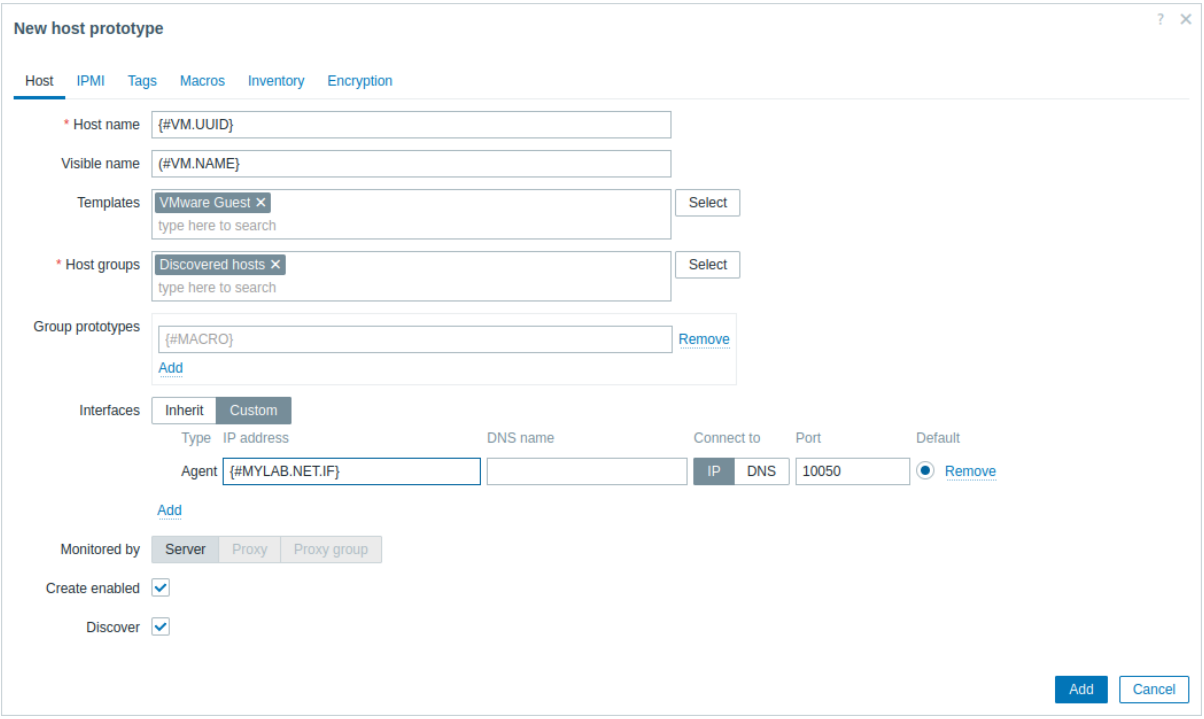
Diese Daten können verwendet werden, um eine benutzerdefinierte Host-Schnittstelle zu konfigurieren.

1. Konfigurieren Sie beim Erstellen einer Low-Level-Discovery-Regel zusätzlich ein Low-Level-Discovery-Makro. Erstellen Sie auf der Registerkarte *LLD-Makros* ein benutzerdefiniertes LLD-Makro mit einem JSONPath-Wert. Zum Beispiel:

- `{#MYLAB.NET.IF}` - `$.net_if[?(@.ifbackingdevice=="VLAN(myLab)")].ifip[0].first()`



2. Fügen Sie beim Erstellen eines Host-Prototyps eine benutzerdefinierte Host-Schnittstelle hinzu und geben Sie das LLD-Makro im Feld *DNS-Name* oder *IP address* ein.



Webüberwachung

Übersicht Mit Zabbix können Sie mehrere Verfügbarkeitsaspekte von Websites überprüfen.

Attention:
Für die Durchführung des Web-Monitorings muss der Zabbix Server zunächst mit cURL- Unterstützung (libcurl) konfiguriert werden.

Um das Web-Monitoring zu aktivieren, müssen Sie Web-Szenarien definieren. Ein Web-Szenario besteht aus einer oder mehreren HTTP-Anfragen oder "Schritten". Die Schritte werden vom Zabbix Server in einer vordefinierten Reihenfolge regelmäßig ausgeführt. Wird ein Host über einen Proxy überwacht, werden die Schritte vom Proxy ausgeführt.

Web-Szenarien werden Hosts/Vorlagen auf die gleiche Weise wie Datenpunkte, Auslöser usw. zugeordnet. Das bedeutet, dass Web-Szenarien auch auf Vorlagenebene erstellt und dann in einem Schritt auf mehrere Hosts angewendet werden können.

In jedem Web-Szenario werden die folgenden Informationen erfasst:

- durchschnittliche Download-Geschwindigkeit pro Sekunde für alle Schritte des gesamten Szenarios
- Nummer des fehlgeschlagenen Schritts
- letzte Fehlermeldung

In jedem Schritt eines Web-Szenarios werden die folgenden Informationen erfasst:

- Download-Geschwindigkeit pro Sekunde
- Antwortzeit
- Antwortcode

Weitere Informationen finden Sie unter Web-Monitoring-Datenpunkte.

Die beim Ausführen von Web-Szenarien erfassten Daten werden in der Datenbank gespeichert. Die Daten werden automatisch für Diagramme, Auslöser und Benachrichtigungen verwendet.

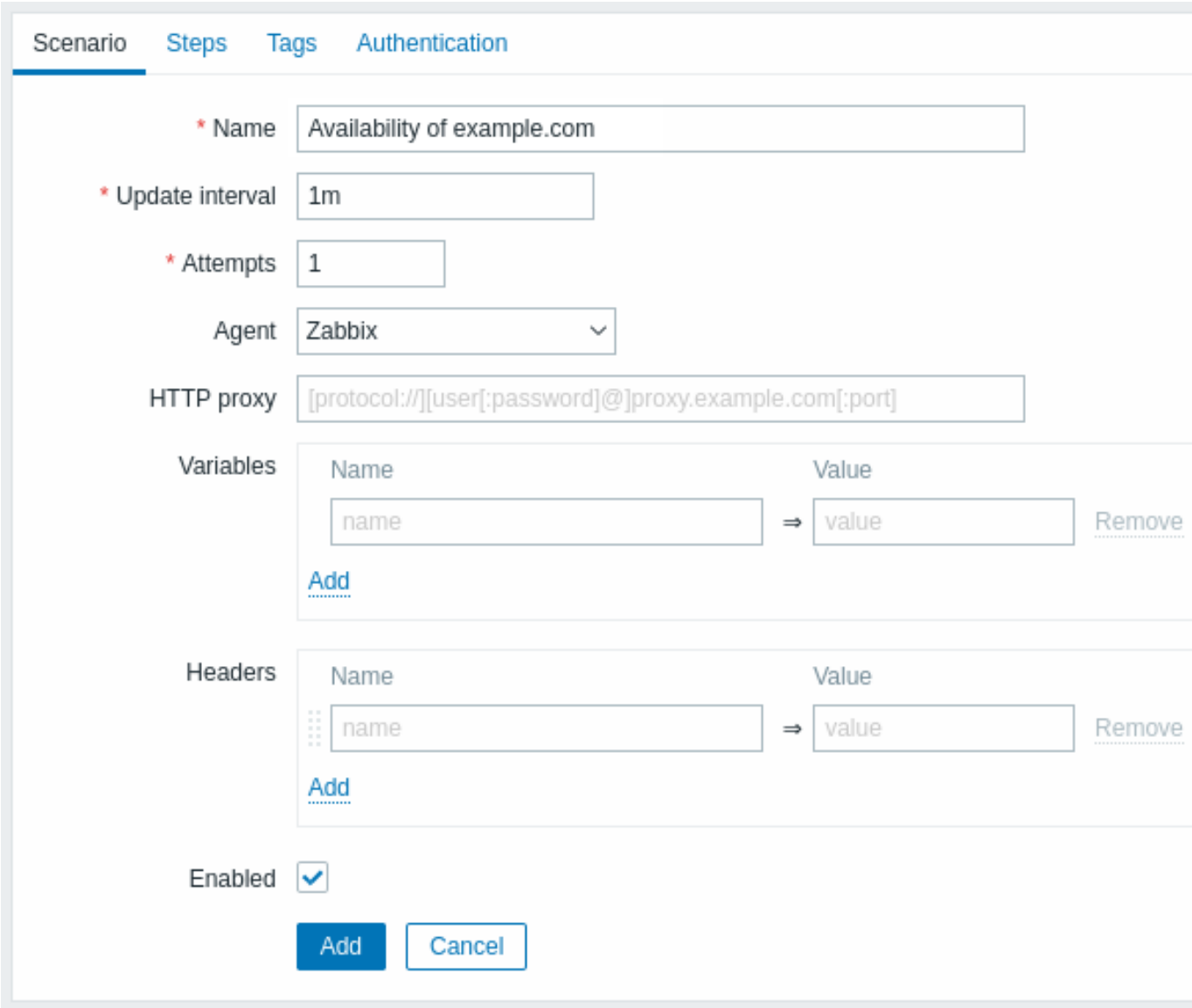
Zabbix kann außerdem prüfen, ob eine abgerufene HTML-Seite eine vordefinierte Zeichenfolge enthält. Es kann eine simulierte Anmeldung ausführen und einem Pfad simulierter Mausklicks auf der Seite folgen.

Das Zabbix Web-Monitoring unterstützt sowohl HTTP als auch HTTPS. Beim Ausführen eines Web-Szenarios folgt Zabbix optional Weiterleitungen (siehe unten die Option *Weiterleitungen folgen*). Die maximale Anzahl an Weiterleitungen ist fest auf 10 codiert (unter Verwendung der cURL-Option CURLOPT_MAXREDIRS). Alle Cookies bleiben während der Ausführung eines einzelnen Szenarios erhalten.

Konfigurieren eines Webszenarios So konfigurieren Sie ein Webszenario:

1. Gehen Sie zu: *Datensammlung* > *Hosts* (oder *Vorlagen*).
2. Klicken Sie in der Zeile des Hosts/der Vorlage auf *Web*.
3. Klicken Sie rechts auf *Webszenario erstellen* (oder auf den Namen des Szenarios, um ein vorhandenes Szenario zu bearbeiten).
4. Geben Sie die Parameter des Szenarios im Formular ein.

Auf der Registerkarte **Szenario** können Sie die allgemeinen Parameter eines Webszenarios konfigurieren.



Alle Pflichtfelder sind mit einem roten Sternchen gekennzeichnet.

Parameter des Szenarios:

Parameter	Beschreibung
Name	Eindeutiger Name des Szenarios. Benutzermakros werden unterstützt. *Beachten Sie*, dass diese Makros in den Namen der Webüberwachungs-Datenpunkte nicht aufgelöst werden, wenn Benutzermakros verwendet werden.
Aktualisierungsintervall	Gibt an, wie oft das Szenario ausgeführt wird. Zeitsuffixe werden unterstützt, z. B. 30s, 1m, 2h, 1d. Benutzermakros werden unterstützt. *Beachten Sie*, dass bei Verwendung eines Benutzermakros und einer Änderung seines Werts (z. B. 5m → 30s) die nächste Prüfung gemäß dem vorherigen Wert ausgeführt wird (bei den Beispielwerten also später). Neue Webszenarien werden innerhalb von 60 Sekunden nach ihrer Erstellung geprüft.
Versuche	Die Anzahl der Versuche zur Ausführung der Schritte des Webszenarios. Bei Netzwerkproblemen (Zeitüberschreitung, keine Verbindung usw.) kann Zabbix die Ausführung eines Schritts mehrmals wiederholen. Der festgelegte Wert gilt gleichermaßen für jeden Schritt des Szenarios. Es können bis zu 10 Versuche angegeben werden; der Standardwert ist 1. *Hinweis*: Zabbix wiederholt einen Schritt nicht aufgrund eines falschen Antwortcodes oder weil eine erforderliche Zeichenfolge nicht übereinstimmt.
Agent	Wählen Sie einen Client-Agent aus. Zabbix gibt sich als der ausgewählte Browser aus. Dies ist nützlich, wenn eine Website je nach Browser unterschiedliche Inhalte zurückgibt. In diesem Feld können Benutzermakros verwendet werden.
HTTP-Proxy	Sie können einen HTTP-Proxy im Format `[protocol://][username[:password]@]proxy.example.com[:port]` angeben. Dadurch wird die cURL-Option CURLOPT_PROXY gesetzt. Das optionale Präfix `protocol://` kann verwendet werden, um alternative Proxy-Protokolle anzugeben (die Unterstützung für das Protokollpräfix wurde in cURL 7.21.7 hinzugefügt). Wenn kein Protokoll angegeben wird, wird der Proxy als HTTP-Proxy behandelt. Standardmäßig wird Port 1080 verwendet. Wenn ein Proxy angegeben ist, überschreibt er proxybezogene Umgebungsvariablen wie http_proxy und HTTPS_PROXY. Wenn kein Proxy angegeben ist, überschreibt er proxybezogene Umgebungsvariablen nicht. Der eingegebene Wert wird unverändert übernommen; es findet keine Plausibilitätsprüfung statt. Sie können auch eine SOCKS-Proxyadresse eingeben. Wenn Sie das falsche Protokoll angeben, schlägt die Verbindung fehl und der Datenpunkt wird nicht unterstützt. *Beachten Sie*, dass HTTP-Proxy nur einfache Authentifizierung unterstützt. In diesem Feld können Benutzermakros verwendet werden.

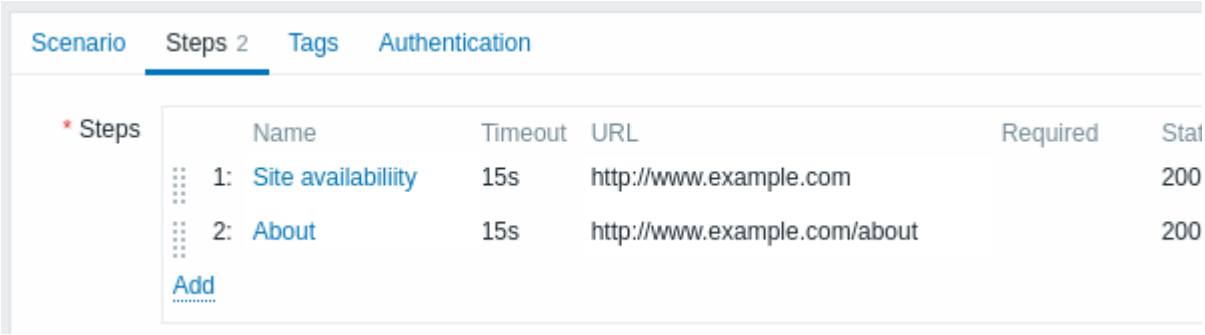
Parameter	Beschreibung
Variablen	Variablen, die in den Schritten des Szenarios verwendet werden können (URL, POST-Variablen). Sie haben das folgende Format: **{macro1}**=value1 **{macro2}**=value2 **{macro3}**=regex:<regular expression> **{macro4}**=jsonpath:<jsonpath> **{macro5}**=xmlxpath:<xmlxpath> **{macro6}**={{macro}.function()} (siehe Makrofunktionen) Beispiel: {username}=Alexei {password}=kj3h5kJ34bd {hostid}=regex:hostid is ([0-9]+) {url}=jsonpath:$.host_url {status}=xmlxpath://host/response/status {newvar}={{myvar}.btoa()} Die Makros können anschließend in den Schritten als {username}, {password}, {hostid} usw. referenziert werden. Zabbix ersetzt sie automatisch durch die tatsächlichen Werte. Beachten Sie, dass Variablen mit `regex:` einen Schritt benötigen, um den Wert des regulären Ausdrucks zu ermitteln; der extrahierte Wert kann daher erst auf den darauffolgenden Schritt angewendet werden. Wenn der Wertteil mit `regex:` beginnt, wird der darauf folgende Teil als regulärer Ausdruck behandelt, der die Webseite durchsucht und bei einem Treffer die Übereinstimmung in der Variablen speichert. Es muss mindestens eine Untergruppe vorhanden sein, damit der übereinstimmende Wert extrahiert werden kann. Benutzermakros und {HOST.*}-Makros werden unterstützt. Variablen werden automatisch URL-kodiert, wenn sie in Abfragefeldern oder Formulardaten für POST-Variablen verwendet werden. Bei Verwendung in rohen POST-Daten oder direkt in der URL müssen sie jedoch manuell URL-kodiert werden.
Header	HTTP-Header werden beim Ausführen einer Anfrage verwendet. Standardheader und benutzerdefinierte Header können verwendet werden. Header werden abhängig vom auf Szenarioebene aus einer Dropdown-Liste ausgewählten Agent-Typ mit Standardeinstellungen zugewiesen und auf alle Schritte angewendet, sofern sie nicht auf Schrittebene benutzerdefiniert definiert wurden. **Beachten Sie, dass die Definition des Headers auf Schrittebene automatisch alle zuvor definierten Header verwirft, mit Ausnahme eines Standardheaders, der durch Auswahl von „User-Agent“ aus einer Dropdown-Liste auf Szenarioebene zugewiesen wurde.** Der Standardheader „User-Agent“ kann jedoch ebenfalls überschrieben werden, indem er auf Schrittebene angegeben wird. Um den Header auf Szenarioebene zu deaktivieren, muss der Header auf Schrittebene benannt und ohne Wert angegeben werden. Header sollten mit derselben Syntax aufgelistet werden, wie sie im HTTP-Protokoll erscheinen würden; optional können einige zusätzliche Funktionen unterstützt durch die cURL-Option CURLOPT_HTTPHEADER verwendet werden. Beispiel: Accept-Charset=utf-8 Accept-Language=en-US Content-Type=application/xml; charset=utf-8 User macros und {HOST.*} Makros werden unterstützt.
Aktiviert	Das Szenario ist aktiv, wenn dieses Kontrollkästchen aktiviert ist; andernfalls ist es deaktiviert.

Beachten Sie, dass beim Bearbeiten eines vorhandenen Szenarios zwei zusätzliche Schaltflächen im Formular verfügbar sind:

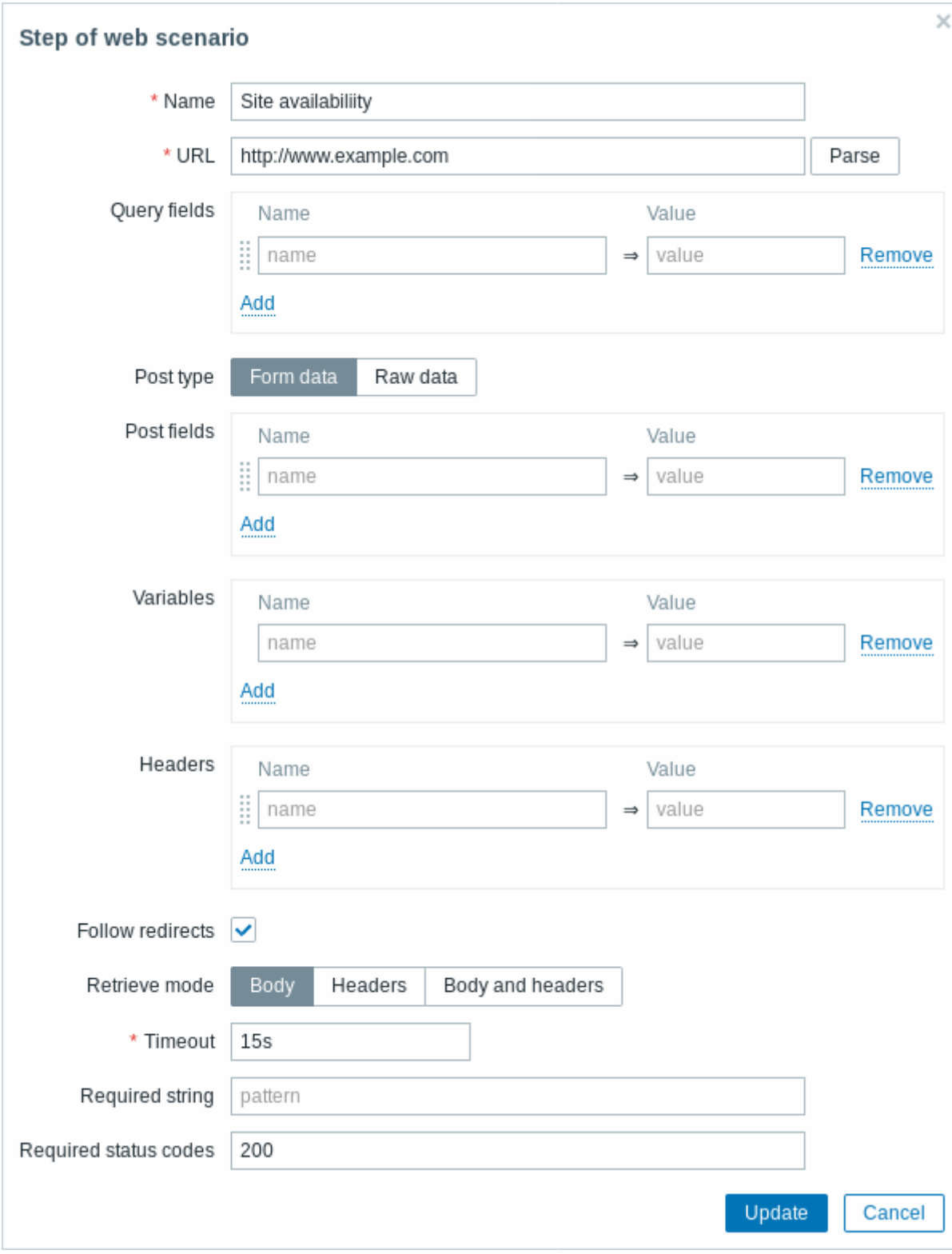
Clone	Erstellt ein weiteres Szenario auf Grundlage der Eigenschaften des vorhandenen Szenarios.
Clear history and trends	Löscht Verlaufs- und Trenddaten für das Szenario. Dadurch führt der Server das Szenario unmittelbar nach dem Löschen der Daten aus.

Note:
Wenn das Feld *HTTP-Proxy* leer bleibt, können Sie einen HTTP-Proxy auch über proxybezogene Umgebungsvariablen verwenden.
Für HTTP-Prüfungen setzen Sie die Umgebungsvariable `http_proxy` für den Zabbix-Serverbenutzer. Beispiel: `http_proxy=http://proxy_ip:proxy_port`.
Für HTTPS-Prüfungen setzen Sie die Umgebungsvariable `HTTPS_PROXY`. Beispiel: `HTTPS_PROXY=http://proxy_ip:proxy_port`.
Weitere Informationen erhalten Sie durch Ausführen des Shell-Befehls: `# man curl`.

Auf der Registerkarte **Schritte** können Sie die Schritte des Webszenarios konfigurieren. Um einen Schritt zu einem Webszenario hinzuzufügen, klicken Sie im Block *Schritte* auf *Hinzufügen*.



Note:
Geheime Benutzermakros dürfen nicht in URLs verwendet werden, da sie zu "******" aufgelöst werden.



Konfigurationsschritte

Schrittparameter:

Parameter	Beschreibung
Name	Eindeutiger Schrittname. Benutzermakros werden unterstützt. *Beachten Sie*, dass diese Makros in den Namen von Webüberwachungs-Datenpunkten nicht aufgelöst werden, wenn Benutzermakros verwendet werden.

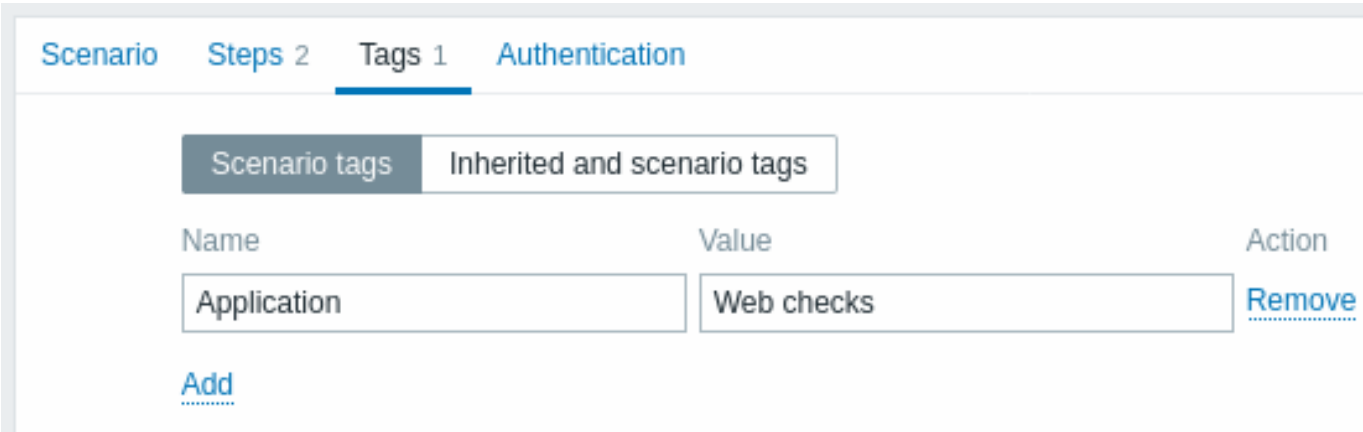
Parameter	Beschreibung
URL	URL, zu der eine Verbindung hergestellt und von der Daten abgerufen werden sollen. Zum Beispiel: https://www.example.com http://www.example.com/download Domänennamen können in Unicode-Zeichen angegeben werden. Sie werden bei der Ausführung des Webszenarioschritts automatisch in ASCII mittels Punycode konvertiert. Mit der Schaltfläche *Parse* können optionale Abfragefelder (wie `?key=value&limit=10`) von der URL getrennt werden. Dabei werden die Attribute und Werte zur automatischen URL-Kodierung in *Abfragefelder* verschoben. In der URL können Variablen mithilfe der Syntax {macro} verwendet werden. Variablen können manuell mithilfe der Syntax {{macro}.urlencode()} URL-kodiert werden. Benutzermakros und {HOST.*}-Makros werden unterstützt. Aus Sicherheitsgründen sollten tatsächliche Benutzernamen oder Passwörter nicht in URLs enthalten sein. Verwenden Sie stattdessen Benutzermakros, um die Offenlegung von Zugangsdaten zu vermeiden. Begrenzt auf 2048 Zeichen.
Abfragefelder	HTTP-GET-Variablen für die URL. Werden als Attribut-Wert-Paare angegeben. Werte werden automatisch URL-kodiert. Werte aus Szenariovariablen, Benutzermakros oder {HOST.*}-Makros werden aufgelöst und anschließend automatisch URL-kodiert. Bei Verwendung der Syntax {{macro}.urlencode()} werden sie doppelt URL-kodiert. Benutzermakros und {HOST.*}-Makros werden unterstützt.
Post	HTTP-POST-Variablen. Im Modus **Formulardaten** werden sie als Attribut-Wert-Paare angegeben. Werte werden automatisch URL-kodiert. Werte aus Szenariovariablen, Benutzermakros oder {HOST.*}-Makros werden aufgelöst und anschließend automatisch URL-kodiert. Im Modus **Rohdaten** werden Attribute und Werte in einer einzigen Zeile angezeigt und mit dem Symbol **&** verkettet. Rohwerte können manuell mithilfe der Syntax {{macro}.urlencode()} oder {{macro}.urldecode()} URL-kodiert bzw. dekodiert werden. Zum Beispiel: id=2345&userid={user} Wenn {user} als Variable des Webszenarios definiert ist, wird sie bei der Ausführung des Schritts durch ihren Wert ersetzt. Wenn Sie die Variable URL-kodieren möchten, ersetzen Sie {user} durch {{user}.urlencode()}. Benutzermakros und {HOST.*}-Makros werden unterstützt.
Variablen	Variablen auf Schrittebene, die für GET- und POST-Funktionen verwendet werden können. Werden als Attribut-Wert-Paare angegeben. Variablen auf Schrittebene überschreiben Variablen auf Szenarioebene oder Variablen aus dem vorherigen Schritt. Der Wert einer Variable auf Schrittebene wirkt sich jedoch erst auf den folgenden Schritt aus, nicht auf den aktuellen Schritt. Sie haben das folgende Format: **{macro}**=value **{macro}**=regex:<regular expression> Weitere Informationen finden Sie in der Variablenbeschreibung auf Szenarioebene. Variablen werden automatisch URL-kodiert, wenn sie in Abfragefeldern oder Formulardaten für POST-Variablen verwendet werden. Bei Verwendung in rohen POST-Daten oder direkt in der URL müssen sie jedoch manuell URL-kodiert werden.
Header	Benutzerdefinierte HTTP-Header, die beim Ausführen einer Anfrage gesendet werden. Werden als Attribut-Wert-Paare angegeben. Ein auf Schrittebene definierter Header wird für den jeweiligen Schritt verwendet. **Beachten Sie, dass die Definition des Headers auf Schrittebene automatisch alle zuvor definierten Header verwirft, mit Ausnahme eines Standardheaders, der durch Auswahl von „User-Agent“ in einer Dropdown-Liste auf Szenarioebene zugewiesen wurde.** Der Standardheader „User-Agent“ kann jedoch ebenfalls überschrieben werden, indem er auf Schrittebene angegeben wird. Wenn beispielsweise ein Headername zugewiesen, aber kein Wert festgelegt wird, wird der Standardheader auf Szenarioebene entfernt. Benutzermakros und {HOST.*}-Makros werden unterstützt. Dadurch wird die cURL-Option CURLOPT_HTTPHEADER festgelegt.
Weiterleitungen folgen	Aktivieren Sie das Kontrollkästchen, um HTTP-Weiterleitungen zu folgen. Dadurch wird die cURL-Option CURLOPT_FOLLOWLOCATION festgelegt.

Parameter	Beschreibung
Abrufmodus	Wählen Sie den Abrufmodus aus: **Body** – nur den Body aus der HTTP-Antwort abrufen **Header** – nur die Header aus der HTTP-Antwort abrufen **Body und Header** – Body und Header aus der HTTP-Antwort abrufen
Zeitüberschreitung	Zabbix verwendet für die Verarbeitung der URL nicht mehr als die festgelegte Zeitspanne (von einer Sekunde bis maximal 1 Stunde). Tatsächlich definiert dieser Parameter die maximale Zeit für den Verbindungsaufbau zur URL und die maximale Zeit für die Ausführung einer HTTP-Anfrage. Daher verwendet Zabbix für den Schritt nicht mehr als **2 x Zeitüberschreitung** Sekunden. Zeitsuffixe werden unterstützt, z. B. 30s, 1m, 1h. Benutzermakros werden unterstützt.
Erforderliche Zeichenfolge	Erforderliches Muster eines regulären Ausdrucks. Wenn der abgerufene Inhalt (HTML) nicht mit dem erforderlichen Muster übereinstimmt, schlägt der Schritt fehl. Wenn das Feld leer ist, wird keine Prüfung auf die erforderliche Zeichenfolge durchgeführt. Zum Beispiel: Homepage of Zabbix Welcome.*admin *Hinweis*: Die Referenzierung von im Zabbix-Frontend erstellten regulären Ausdrücken wird in diesem Feld nicht unterstützt. Benutzermakros und {HOST.*}-Makros werden unterstützt.
Erforderliche Statuscodes	Liste der erwarteten HTTP-Statuscodes. Wenn Zabbix einen Code erhält, der nicht in der Liste enthalten ist, schlägt der Schritt fehl. Wenn das Feld leer ist, wird keine Prüfung der Statuscodes durchgeführt. Zum Beispiel: 200,201,210-299 Benutzermakros werden unterstützt.

Note:
Änderungen an Webszenarioschritten werden erst gespeichert, wenn das gesamte Szenario gespeichert wird.

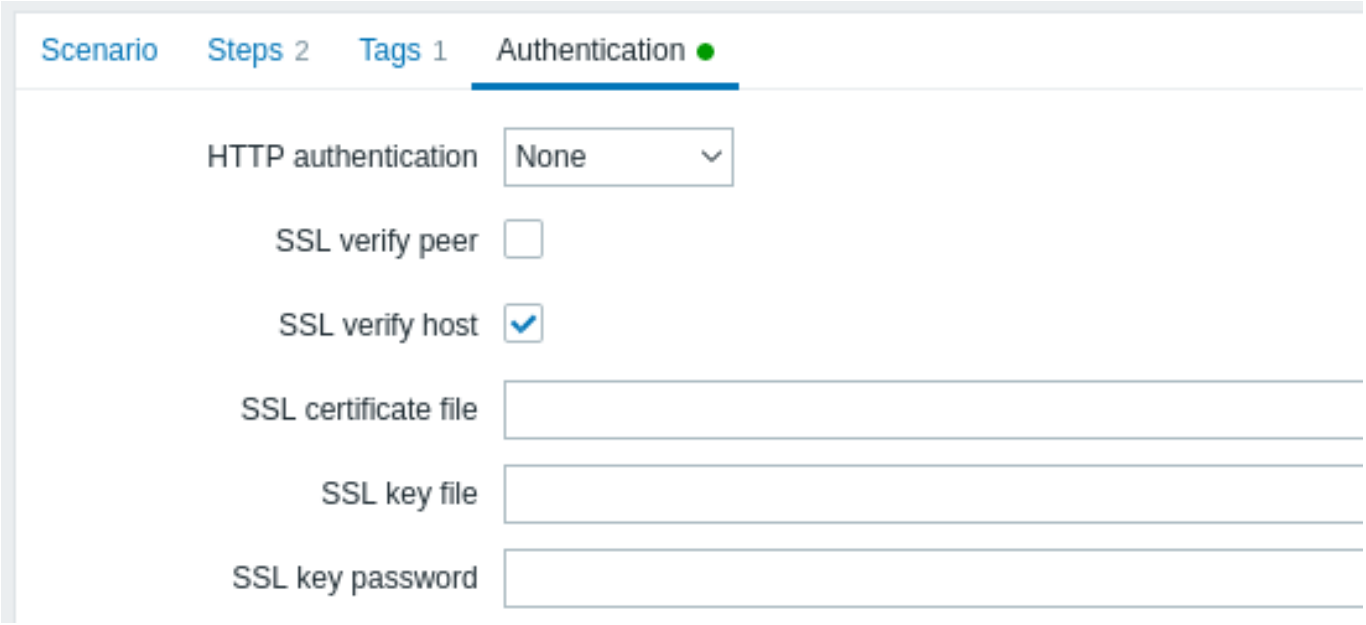
Siehe auch ein Praxisbeispiel, das zeigt, wie Webüberwachungsschritte konfiguriert werden können.

Tags konfigurieren Auf der Registerkarte **Tags** können Sie Tags auf Szenarioebene definieren (Tags).



Durch das Tagging können Sie Webszenarien und Datenpunkte zur Webüberwachung filtern.

Authentifizierung konfigurieren Auf der Registerkarte **Authentifizierung** können Sie die Authentifizierungsoptionen für das Szenario konfigurieren. Ein grüner Punkt neben dem Namen der Registerkarte zeigt an, dass eine HTTP-Authentifizierung aktiviert ist.



Authentifizierungsparameter:

Parameter	Beschreibung
HTTP-Authentifizierung	Wählen Sie die Authentifizierungsoption aus: **Keine** – keine Authentifizierung wird verwendet; **Basic** – die Basic-Authentifizierung wird verwendet; **NTLM** – die NTLM-Authentifizierung (Windows NT LAN Manager) wird verwendet; **Kerberos** – die Kerberos-Authentifizierung wird verwendet (siehe auch: Kerberos mit Zabbix konfigurieren); **Digest** – die Digest-Authentifizierung wird verwendet.
Benutzer	Geben Sie den Benutzernamen ein (bis zu 255 Zeichen). Dieses Feld ist verfügbar, wenn *HTTP-Authentifizierung* auf Basic, NTLM, Kerberos oder Digest gesetzt ist. Benutzermakros werden unterstützt.
Passwort	Geben Sie das Benutzerpasswort ein (bis zu 255 Zeichen). Dieses Feld ist verfügbar, wenn *HTTP-Authentifizierung* auf Basic, NTLM, Kerberos oder Digest gesetzt ist. Benutzermakros werden unterstützt.
SSL-Peer überprüfen	Aktivieren Sie das Kontrollkästchen, um das SSL-Zertifikat des Webservers zu überprüfen. Das Serverzertifikat wird anhand der Zertifikate der vertrauenswürdigen Zertifizierungsstellen (CA) am systemweiten CA-Speicherort überprüft. Sie können den Speicherort der CA-Dateien mit dem Zabbix-Server- oder Proxy-Konfigurationsparameter SSLCALocation überschreiben. Dadurch wird die cURL-Option CURLOPT_SSL_VERIFYPEER gesetzt.
SSL-Host überprüfen	Aktivieren Sie das Kontrollkästchen, um zu überprüfen, ob das Feld *Common Name* oder das Feld *Subject Alternate Name* des Webserverzertifikats übereinstimmt. Dadurch wird die cURL-Option CURLOPT_SSL_VERIFYHOST gesetzt.
SSL-Zertifikatsdatei	Name der SSL-Zertifikatsdatei, die für die Clientauthentifizierung verwendet wird. Die Zertifikatsdatei muss im PEM[1]-Format vorliegen. Wenn die Zertifikatsdatei auch den privaten Schlüssel enthält, lassen Sie das Feld *SSL-Schlüsseldatei* leer. Wenn der Schlüssel verschlüsselt ist, geben Sie das Passwort im Feld *SSL-Schlüsselpasswort* an. Das Verzeichnis, das diese Datei enthält, wird durch den Zabbix-Server- oder Proxy-Konfigurationsparameter SSLCertLocation festgelegt. In diesem Feld können `HOST.*`-Makros und Benutzermakros verwendet werden. Dadurch wird die cURL-Option CURLOPT_SSLCERT gesetzt.
SSL-Schlüsseldatei	Name der privaten SSL-Schlüsseldatei, die für die Clientauthentifizierung verwendet wird. Die Datei mit dem privaten Schlüssel muss im PEM[1]-Format vorliegen. Das Verzeichnis, das diese Datei enthält, wird durch den Zabbix-Server- oder Proxy-Konfigurationsparameter SSLKeyLocation festgelegt. In diesem Feld können `HOST.*`-Makros und Benutzermakros verwendet werden. Dadurch wird die cURL-Option CURLOPT_SSLKEY gesetzt.
SSL-Schlüsselpasswort	Passwort der privaten SSL-Schlüsseldatei. In diesem Feld können Benutzermakros verwendet werden. Dadurch wird die cURL-Option CURLOPT_KEYPASSWD gesetzt.

Attention:
[1] Zabbix unterstützt Zertifikats- und private Schlüsseldateien ausschließlich im PEM-Format. Wenn Ihre Zertifikats- und privaten Schlüsseldaten in einer Datei im PKCS #12-Format vorliegen (üblicherweise mit der Erweiterung *.p12 oder *.pfx), können Sie daraus mit den folgenden Befehlen eine PEM-Datei erzeugen:

```
openssl pkcs12 -in ssl-cert.p12 -clcerts -nokeys -out ssl-cert.pem
openssl pkcs12 -in ssl-cert.p12 -nocerts -nodes  -out ssl-cert.key
```

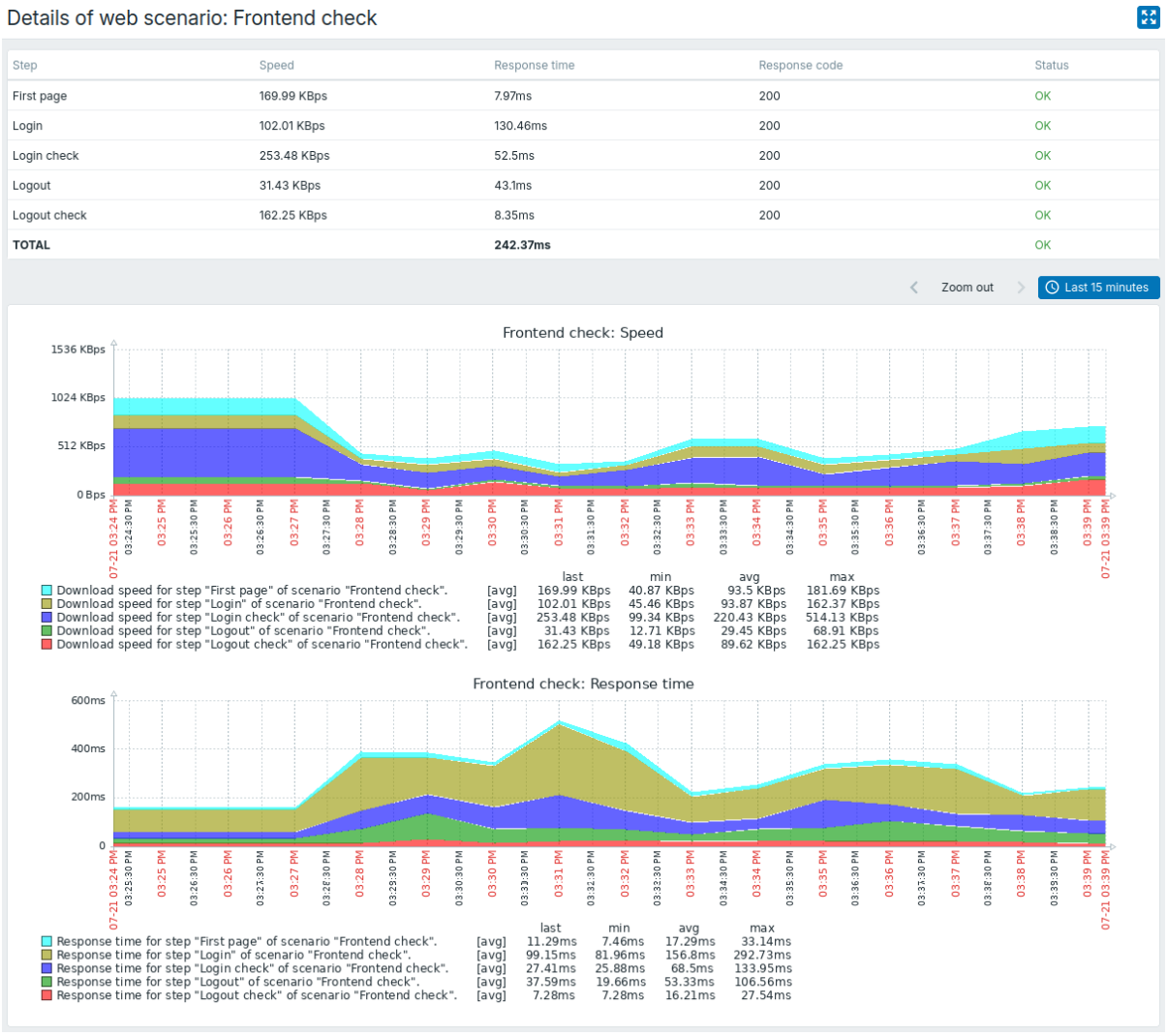
Note:
Der Zabbix-Server übernimmt Änderungen an Zertifikaten ohne Neustart.

Note:
Wenn sich Clientzertifikat und privater Schlüssel in einer einzigen Datei befinden, geben Sie diese einfach im Feld „SSL-Zertifikatsdatei“ an und lassen Sie das Feld „SSL-Schlüsseldatei“ leer. Zertifikat und Schlüssel müssen weiterhin im PEM-Format vorliegen. Das Zusammenführen von Zertifikat und Schlüssel ist einfach:

```
cat client.crt client.key > client.pem
```

Anzeige So zeigen Sie die für einen Host konfigurierten Webszenarien an:

1. Gehen Sie zu *Monitoring* > *Hosts*.
2. Suchen Sie den Host in der Liste und klicken Sie in der letzten Spalte auf den Hyperlink *Web*.
3. Klicken Sie auf den Namen des Szenarios, um detaillierte Informationen anzuzeigen.



Eine Übersicht der Webszenarien kann auch in *Dashboards* über das Widget zur Webüberwachung angezeigt werden.

Die aktuellen Ergebnisse der Ausführung des Webszenarios finden Sie im Abschnitt *Monitoring* > *Latest data*.

Erweiterte Überwachung Manchmal ist es notwendig, den empfangenen Inhalt einer HTML-Seite zu protokollieren. Dies ist besonders nützlich, wenn ein Schritt eines Web-Szenarios fehlschlägt.

Dafür dient die Debug-Stufe 5 (trace).
Diese Stufe kann in den Konfigurationsdateien Server und Proxy oder über eine Option zur Laufzeitsteuerung (`-R log_level_increase="http poller,N"`, wobei N die Prozessnummer ist) gesetzt werden.
Die folgenden Beispiele zeigen, wie die erweiterte Überwachung gestartet werden kann, sofern die Debug-Stufe 4 bereits gesetzt ist:

```
# Protokollstufe aller HTTP-Poller erhöhen:
zabbix_server -R log_level_increase="http poller"

# Protokollstufe des zweiten HTTP-Pollers erhöhen:
zabbix_server -R log_level_increase="http poller,2"
```

Wenn die erweiterte Web-Überwachung nicht erforderlich ist, kann sie mit der Option `-R log_level_decrease` gestoppt werden.

Web-Monitoring-Datenpunkte

Übersicht

Einige neue Datenpunkte werden beim Erstellen von Webszenarien automatisch zur Überwachung hinzugefügt.

Alle Datenpunkte übernehmen Tags aus dem Webszenario.

Datenpunkte des Szenarios

Sobald ein Szenario erstellt wird, fügt Zabbix automatisch die folgenden Datenpunkte zur Überwachung hinzu.

Datenpunkt	Beschreibung
Download-Geschwindigkeit für Szenario <Scenario>	Dieser Datenpunkt erfasst Informationen über die Download-Geschwindigkeit (Bytes pro Sekunde) des gesamten Szenarios, d. h. den Durchschnitt für alle Schritte. Datenpunktschlüssel: web.test.in[Scenario„bps] Typ: *Numeric(float)*
Fehlgeschlagener Schritt des Szenarios <Scenario>	Dieser Datenpunkt zeigt die Nummer des Schritts an, der im Szenario fehlgeschlagen ist. Wenn alle Schritte erfolgreich ausgeführt werden, wird 0 zurückgegeben. Datenpunktschlüssel: web.test.fail[Scenario] Typ: *Numeric(unsigned)*
Letzte Fehlermeldung des Szenarios <Scenario>	Dieser Datenpunkt gibt den Text der letzten Fehlermeldung des Szenarios zurück. Ein neuer Wert wird nur gespeichert, wenn im Szenario ein Schritt fehlgeschlagen ist. Wenn alle Schritte in Ordnung sind, wird kein neuer Wert erfasst. Datenpunktschlüssel: web.test.error[Scenario] Typ: *Character*

Anstelle von ”Scenario” wird der tatsächliche Szenarioname verwendet.

Note:
Wenn der Szenarioname Benutzermakros enthält, bleiben diese Makros in den Namen der Web-Monitoring-Datenpunkte unaufgelöst.

 Wenn der Szenarioname mit einem doppelten Anführungszeichen beginnt oder ein Komma oder eine eckige Klammer enthält, wird er in Datenpunktschlüsseln korrekt in Anführungszeichen gesetzt. In anderen Fällen werden keine zusätzlichen Anführungszeichen gesetzt.

Note:
Web-Monitoring-Datenpunkte werden mit einer Aufbewahrungsdauer von 30 Tagen für die Historie und 90 Tagen für Trends hinzugefügt.

Diese Datenpunkte können verwendet werden, um Auslöser zu erstellen und Benachrichtigungsbedingungen zu definieren.

Beispiel 1

Um einen Auslöser „Web-Szenario fehlgeschlagen“ zu erstellen, können Sie einen Auslöserausdruck definieren:

```
last(/host/web.test.fail[Scenario])<>0
```

Stellen Sie sicher, dass Sie „Scenario“ durch den tatsächlichen Namen Ihres Szenarios ersetzen.

Beispiel 2

Um einen Auslöser „Web-Szenario fehlgeschlagen“ mit einer nützlichen Problembeschreibung im Auslösernamen zu erstellen, können Sie einen Auslöser mit folgendem Namen definieren:

```
Web-Szenario „Scenario“ fehlgeschlagen: {ITEM.VALUE}
```

und folgender Auslöser-Expression:

```
length(last(/host/web.test.error[Scenario]))>0 and last(/host/web.test.fail[Scenario])>0
```

Stellen Sie sicher, „Scenario“ durch den tatsächlichen Namen Ihres Szenarios zu ersetzen.

Beispiel 3

Um einen Auslöser „Webanwendung ist langsam“ zu erstellen, können Sie einen Auslöserausdruck definieren:

```
last(/host/web.test.in[Scenario,,bps])<10000
```

Stellen Sie sicher, dass Sie „Scenario“ durch den tatsächlichen Namen Ihres Szenarios ersetzen.

Datenpunkte für Szenarioschritte

Sobald ein Schritt erstellt wird, fügt Zabbix automatisch die folgenden Datenpunkte zur Überwachung hinzu.

Datenpunkt	Beschreibung
Download-Geschwindigkeit für Schritt <Step> des Szenarios <Scenario>	Dieser Datenpunkt erfasst Informationen über die Download-Geschwindigkeit (Bytes pro Sekunde) des Schritts. Datenpunktschlüssel: web.test.in[Scenario,Step,bps] Typ: *Numeric(float)*
Antwortzeit für Schritt <Step> des Szenarios <Scenario>	Dieser Datenpunkt erfasst Informationen über die Antwortzeit des Schritts in Sekunden. Die Antwortzeit wird vom Beginn der Anfrage an gezählt, bis alle Informationen übertragen wurden. Datenpunktschlüssel: web.test.time[Scenario,Step,resp] Typ: *Numeric(float)*
Antwortcode für Schritt <Step> des Szenarios <Scenario>	Dieser Datenpunkt erfasst die Antwortcodes des Schritts. Datenpunktschlüssel: web.test.rspcode[Scenario,Step] Typ: *Numeric(unsigned)*

Anstelle von ”Scenario” und ”Step” werden jeweils die tatsächlichen Szenario- und Schrittnamen verwendet.

Note:
Datenpunkte der Webüberwachung werden mit einer Verlaufsaufbewahrung von 30 Tagen und einer Trendaufbewahrungsdauer von 90 Tagen hinzugefügt.

Note:
Wenn der Szenarioname mit einem doppelten Anführungszeichen beginnt oder ein Komma oder eine eckige Klammer enthält, wird er in Datenpunktschlüsseln korrekt in Anführungszeichen gesetzt. In anderen Fällen erfolgt keine zusätzliche Maskierung.

Diese Datenpunkte können verwendet werden, um Auslöser zu erstellen und Benachrichtigungsbedingungen zu definieren. Um beispielsweise einen Auslöser ”Zabbix GUI login is too slow” zu erstellen, können Sie den folgenden Auslöserausdruck definieren:

```
last(/zabbix/web.test.time[ZABBIX GUI,Login,resp])>3
```

Praxisbeispiel

Übersicht

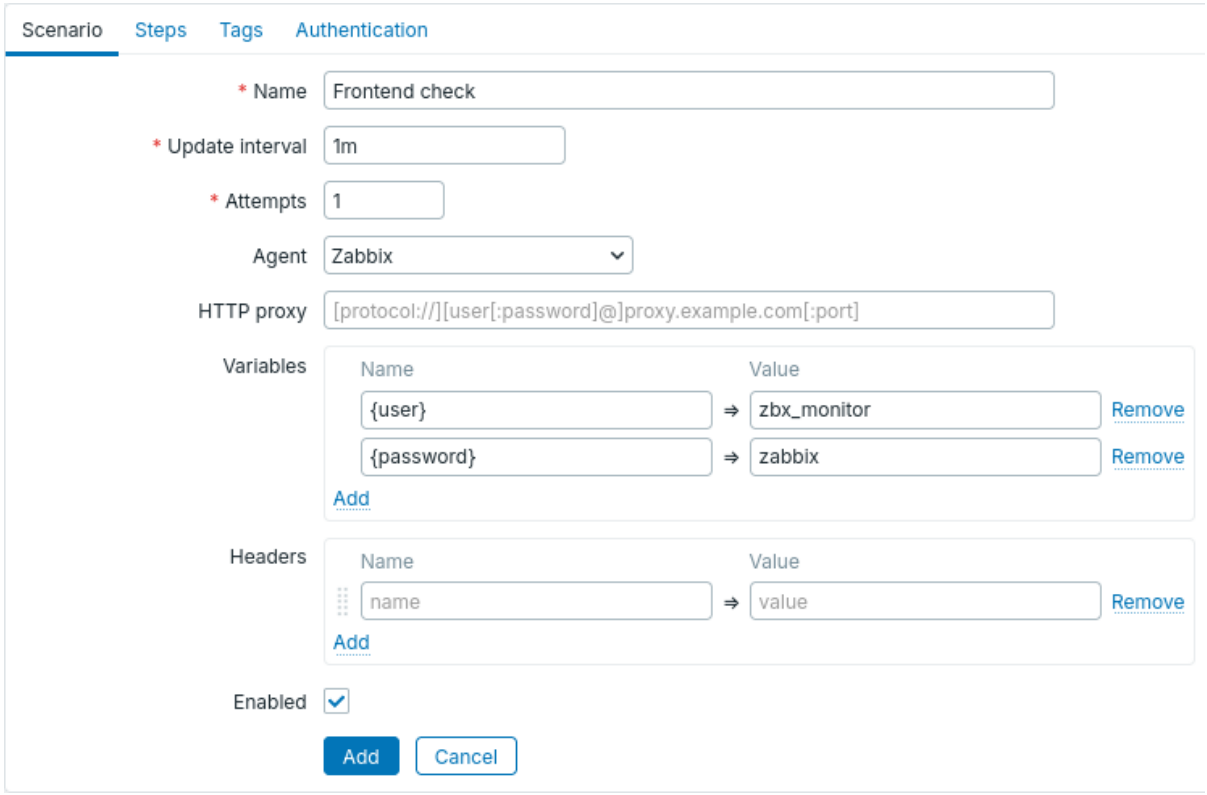
Dieser Abschnitt zeigt ein schrittweises, praxisnahes Beispiel dafür, wie Web-Monitoring verwendet werden kann.

Das Zabbix Web-Monitoring wird verwendet, um das Zabbix Frontend zu überwachen. Das Ziel ist festzustellen, ob es verfügbar ist, den richtigen Inhalt bereitstellt und wie schnell es arbeitet. Um dies zu erreichen, sind mehrere Schritte erforderlich, darunter das Prüfen der Verfügbarkeit der ersten Seite, das Anmelden mit Benutzername und Passwort, das Verifizieren des erfolgreichen Anmeldens, das Abmelden und das Bestätigen der Abmeldung.

Szenario

Ein neues web scenario hinzufügen

1. Gehen Sie zu *Data collection* > *Hosts*.
2. Wählen Sie einen Host aus und klicken Sie in der Zeile dieses Hosts auf *Web*.
3. Klicken Sie auf *Create web scenario*.



4. Füllen Sie im Formular für das neue scenario auf der Registerkarte *Scenario* die folgenden Felder aus:

    - *Name*: Frontend check.
    - *Update interval*: 1m.
    - *Attempts*: 1.
    - *Agent*: Zabbix.
 Im Abschnitt *Variables* fügen Sie die folgenden Name-Wert-Paare hinzu:
    - `{user}`: Ihre vorhandenen Zabbix-Benutzeranmeldedaten;
    - `{password}`: Ihre vorhandenen Zabbix-Benutzeranmeldedaten.

Note:
Aus Sicherheitsgründen wird empfohlen, einen separaten Benutzer mit minimalen Berechtigungen zu erstellen, der für Überwachungszwecke verwendet wird.

Optional können Sie zur Registerkarte *Tags* wechseln und Tags für das web scenario hinzufügen. Sie können Tags für web scenarios verwenden, um zugehörige Datenpunkte und Auslöser schnell zu identifizieren oder die gesammelten Daten zu durchsuchen. Geeignete Tags für dieses Tutorial sind beispielsweise `component: web-scenario` und/oder `target: frontend`.

Webszenario-Schritte konfigurieren

Wechseln Sie zur Registerkarte *Steps* und definieren Sie die Schritte für das Szenario.
 Nachdem Sie jeden Schritt konfiguriert haben, klicken Sie auf die Schaltfläche *Add*.

Attention:
Beachten Sie, dass das Zabbix Frontend beim Anmelden eine JavaScript-Weiterleitung verwendet. Daher muss die Anmeldung zuerst erfolgen, und angemeldete Funktionen können erst in den weiteren Schritten überprüft werden. Außerdem muss der Anmeldeschritt die vollständige URL zur Datei **index.php** verwenden.

Allgemeine Felder für alle Schritte

Füllen Sie zusätzlich zu den schrittspezifischen Feldern für jeden unten beschriebenen Schritt die folgenden allgemeinen Felder aus:

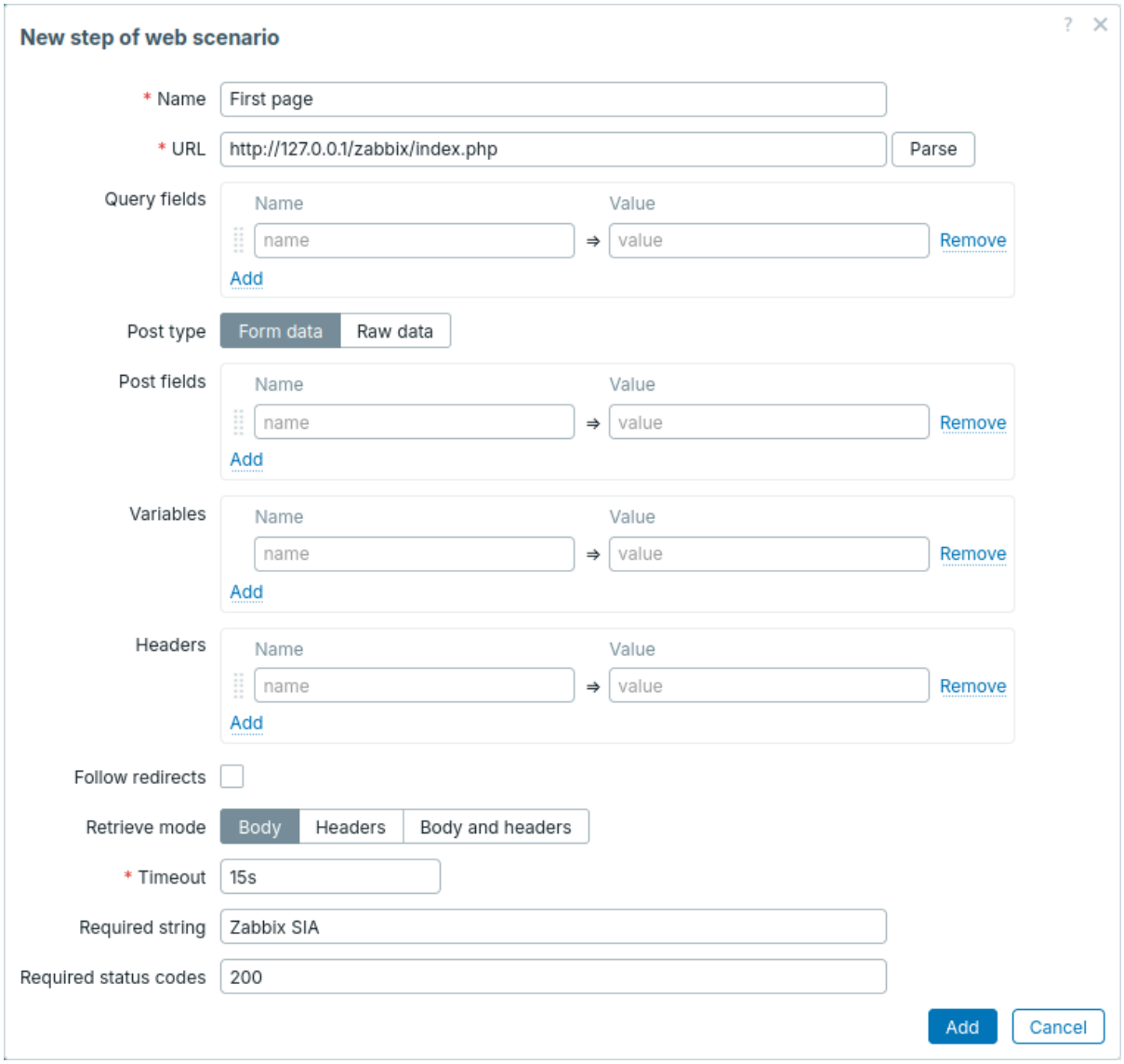
- *URL*: die URL des Zabbix Frontend (`http://<server_ip_or_name>/zabbix/index.php` (für Apache) oder `http://<server_ip_o`
  (für NGINX)). Verwenden Sie für Schritt 3 des Webszenarios eine andere URL, wie unten beschrieben.
- *Timeout*: 15s.

- *Erforderliche Statuscodes*: 200.

Web-Szenario-Schritt 1

Prüfen Sie, dass die erste Seite korrekt antwortet, den HTTP-Statuscode 200 zurückgibt und den Text *Zabbix SIA* enthält.
Füllen Sie die folgenden Felder aus:

- *Name*: Erste Seite.
- *Timeout*: 15s.
- *Required string*: Zabbix SIA.
- Die gemeinsamen Felder für alle Schritte.

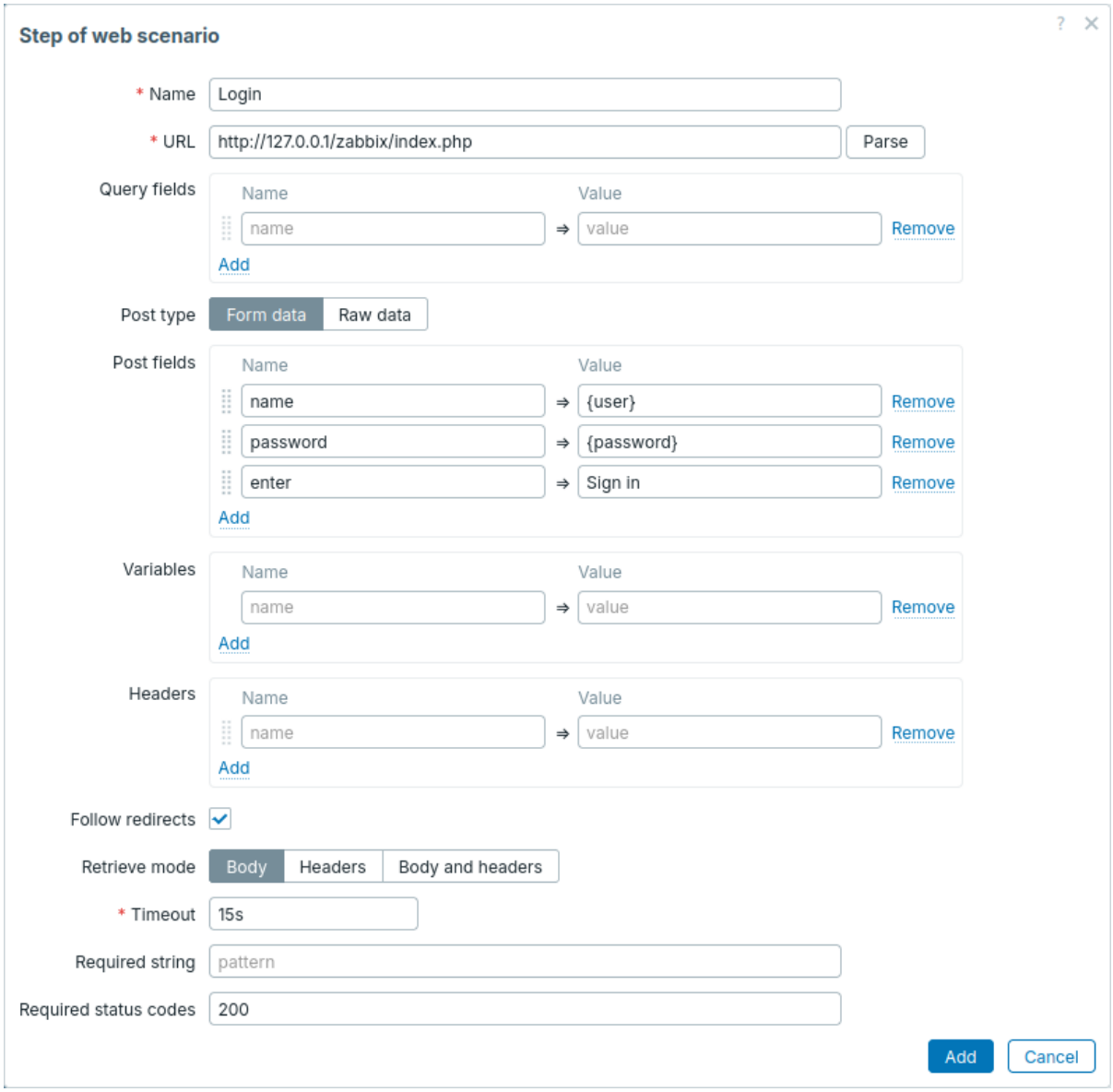


Web-Szenario-Schritt 2

Melden Sie sich im Zabbix-Frontend mit den im Tab *Szenario* definierten Makros (Variablen) an: *{user}* und *{password}*.
Füllen Sie die folgenden Felder aus:

- *Name*: Login.
- Im Abschnitt *Post fields* fügen Sie die folgenden Name-Wert-Paare hinzu:
  - name: `{user}`;
  - password: `{password}`;
  - enter: Sign in.
- Die gemeinsamen Felder für alle Schritte.

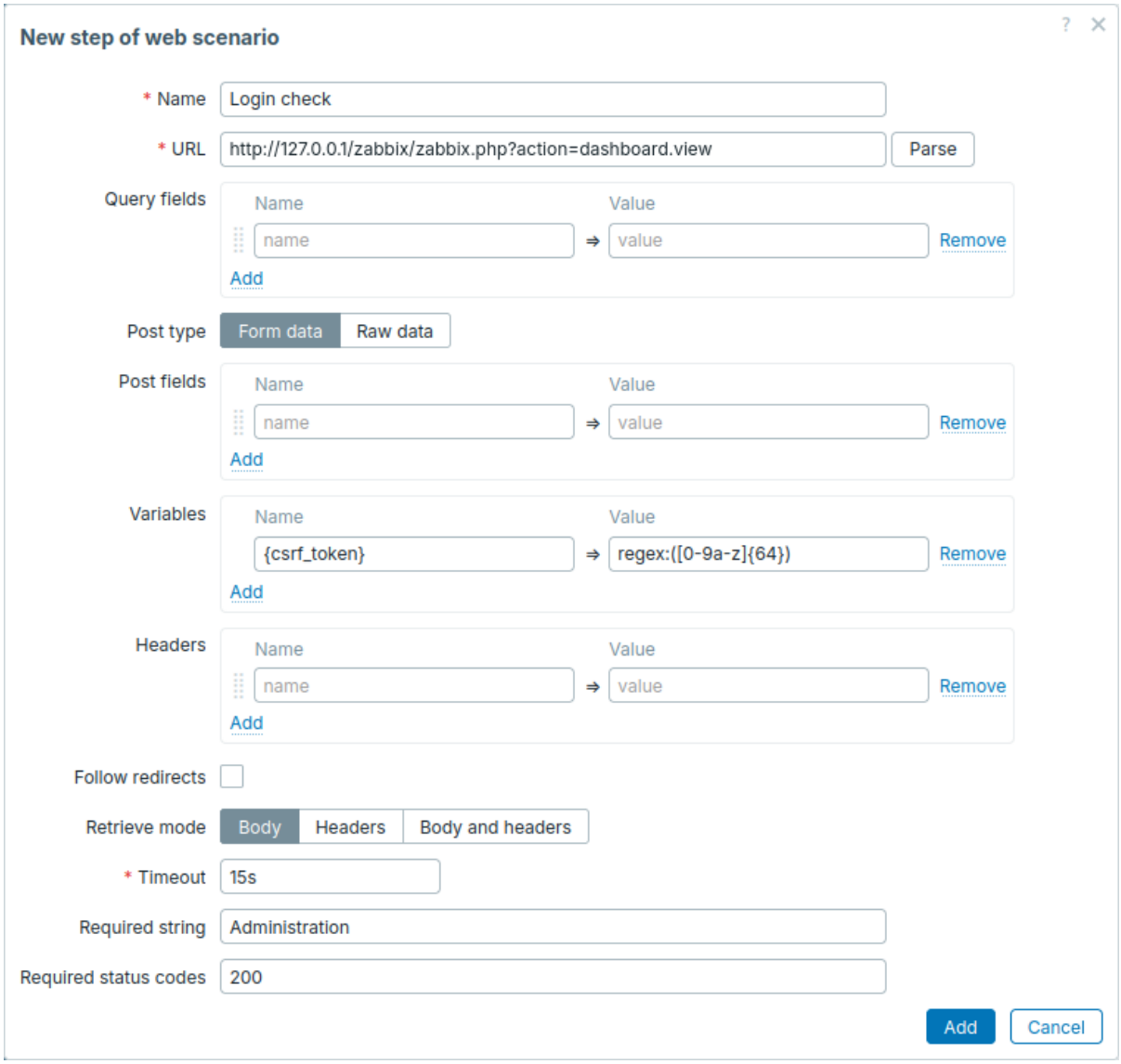
Aktivieren Sie das Kontrollkästchen *Follow redirects*.



Web-Szenario Schritt 3

Nach der Anmeldung überprüfen Sie den Erfolg, indem Sie nach einer Zeichenfolge suchen, die nur im angemeldeten Zustand sichtbar ist, zum Beispiel *Administration*.
 Füllen Sie die folgenden Felder aus:

- *Name*: Login check.
- *URL*: die URL, zu der der Benutzer nach der Anmeldung weitergeleitet wird (zum Beispiel http://127.0.0.1/zabbix/zabbix.php?action=dash
- Im Abschnitt *Variablen* fügen Sie das Name-Wert-Paar `{csrf_token}: regex:([0-9a-z]{64})` hinzu. Diese Variable erfasst den Wert des zugewiesenen CSRF-Tokens, um ihn im web scenario step 4 wiederzuverwenden.
- *Erforderliche Zeichenfolge*: Administration.
- Die gemeinsamen Felder für alle Schritte.



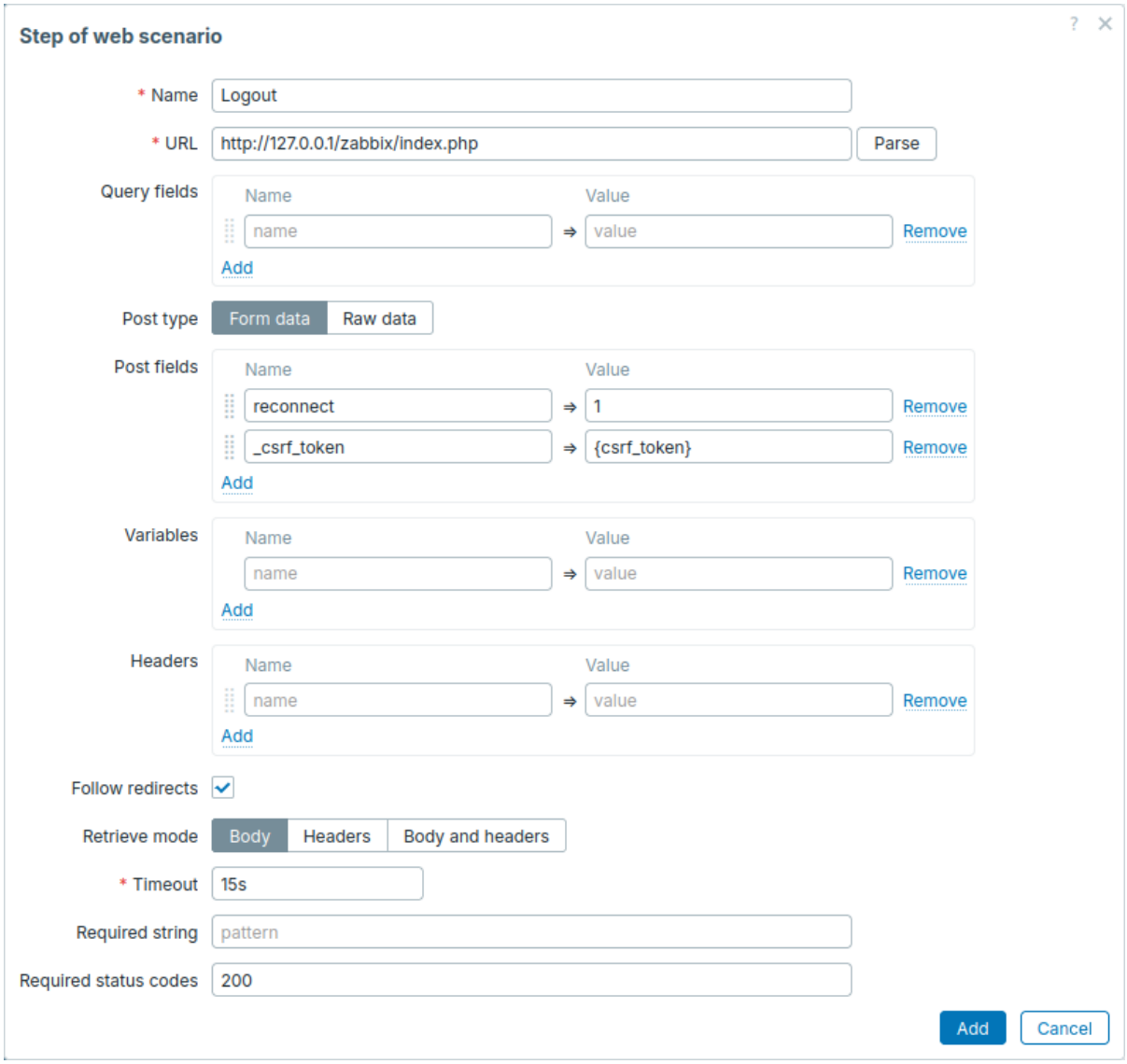
Web-Szenario-Schritt 4

Nachdem die Erreichbarkeit und Anmeldung am Frontend überprüft wurden, fügen Sie einen Abmeldeschritt hinzu. Andernfalls wird die Zabbix-Datenbank mit zahlreichen offenen Sitzungseinträgen überlastet.
 Füllen Sie die folgenden Felder aus:

- *Name*: Abmeldung.
- Fügen Sie im Abschnitt *Post fields* die folgenden Name-Wert-Paare hinzu:
  - reconnect: 1;
  - _csrf_token: `{csrf_token}`.
- Die allgemeinen Felder für alle Schritte.

Aktivieren Sie das Kontrollkästchen *Follow redirects*.

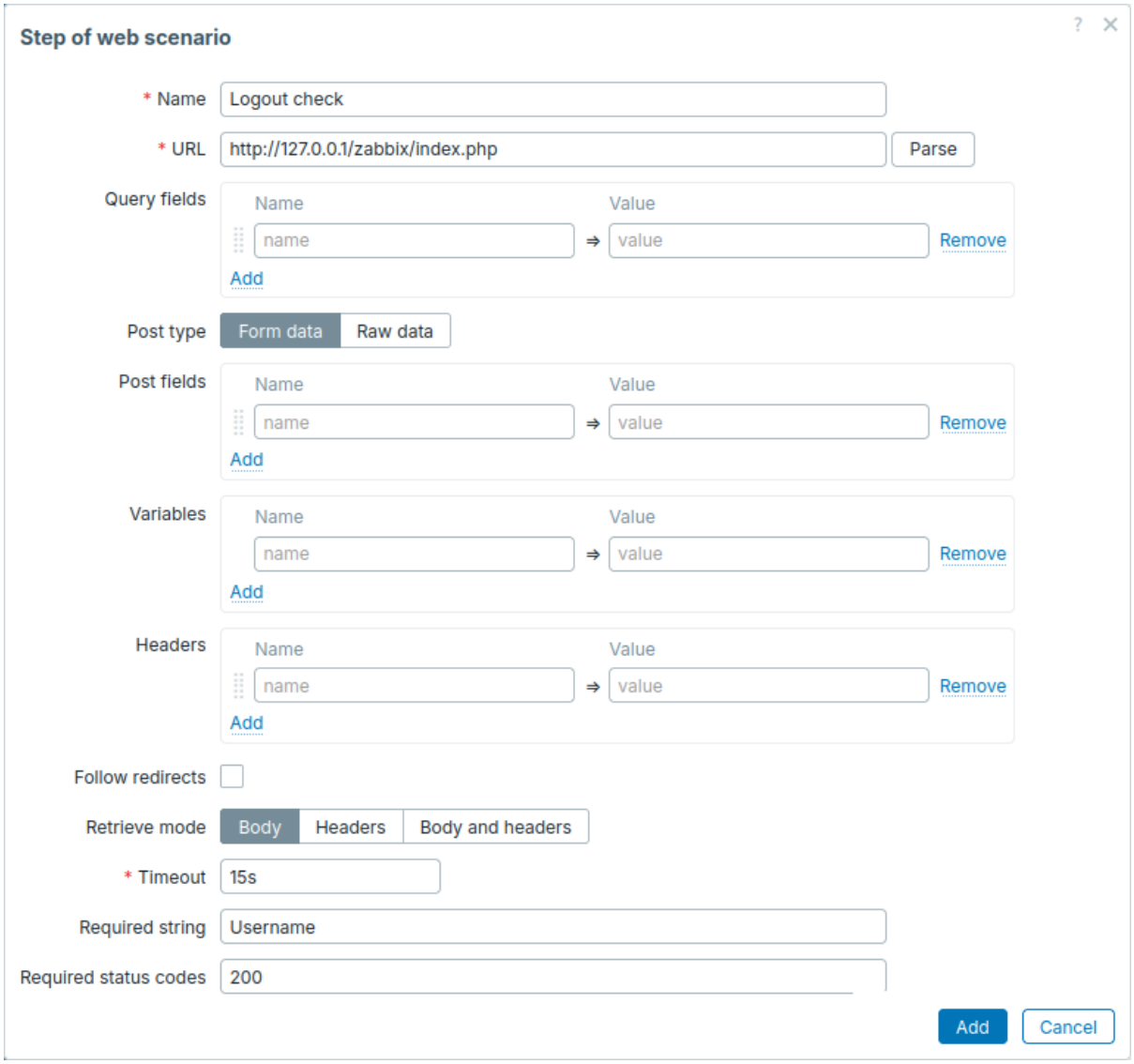
Dieser Schritt verwendet die Variable `{csrf_token}`, die im Web-Szenario-Schritt 3 abgerufen wurde.



Web-Szenario Schritt 5

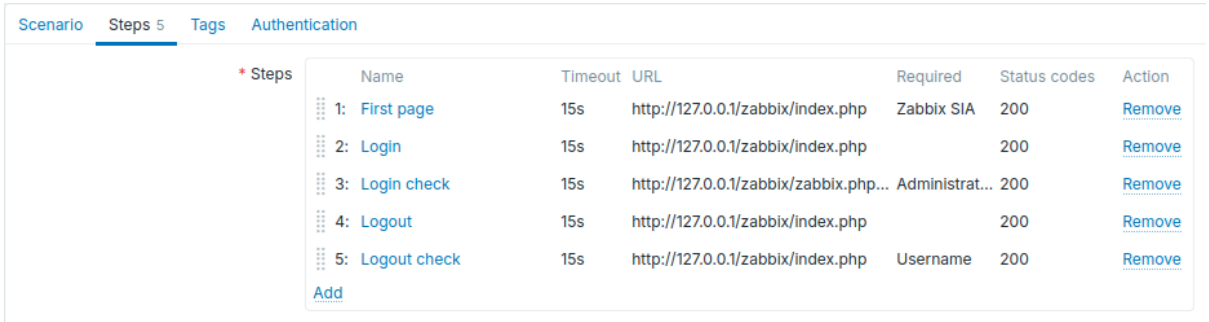
Um die Abmeldung zu bestätigen, prüfen Sie auf die Zeichenfolge *Username*.
 Füllen Sie die folgenden Felder aus:

- *Name*: Logout-Prüfung.
- *Erforderliche Zeichenfolge*: Username.
- Die gemeinsamen Felder für alle Schritte.



Vollständige Konfiguration von Schritten

Eine vollständige Konfiguration von Schritten eines Webszenarios sollte wie folgt aussehen:

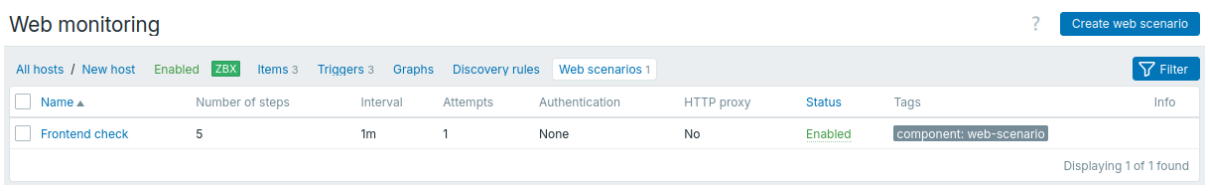


Prüfen Sie die Ergebnisse

Speichern Sie das fertige Webüberwachungsszenario.

Wenn das Webszenario erstellt wird, werden automatisch neue web monitoring items für die Überwachung hinzugefügt.

Das Szenario wird dem Host hinzugefügt. Um Informationen zum Webszenario anzuzeigen, gehen Sie zu *Monitoring* > *Hosts*, suchen Sie den Host in der Liste und klicken Sie in der letzten Spalte auf den Hyperlink *Web*.



Klicken Sie auf den Namen des Szenarios, um detaillierte Statistiken anzuzeigen:



Serviceüberwachung

Übersicht Die Serviceüberwachung ist eine Überwachung auf Geschäftsebene, mit der Sie sich einen Überblick über den gesamten Servicebaum der IT-Infrastruktur verschaffen, Schwachstellen der Infrastruktur identifizieren, die SLA verschiedener IT-Services berechnen und weitere Informationen auf einer höheren Ebene überprüfen können. Die Serviceüberwachung konzentriert sich auf die Gesamtverfügbarkeit eines Services statt auf Details auf niedriger Ebene, wie etwa fehlenden Speicherplatz, eine hohe Prozessorlast usw. Die Serviceüberwachung bietet außerdem Funktionen zur Ermittlung der Grundursache eines Problems, wenn ein Service nicht wie erwartet funktioniert.

Die Serviceüberwachung ermöglicht die Erstellung einer hierarchischen Darstellung der überwachten Daten.

Eine sehr einfache Servicestruktur kann beispielsweise so aussehen:

```
Service
|
|-Workstations
| |
| |-Workstation1
| |
| |-Workstation2
|
|-Server
```

Jeder Knoten der Struktur verfügt über ein Statusattribut. Der Status wird berechnet und gemäß dem ausgewählten Algorithmus an übergeordnete Ebenen weitergegeben. Der Status einzelner Knoten wird durch den Status der zugeordneten Probleme beeinflusst. Die Problemzuordnung erfolgt durch Tagging.

Zabbix kann Benachrichtigungen senden oder automatisch ein Skript auf dem Zabbix-Server ausführen, wenn eine Änderung des Servicestatus erkannt wird. Es ist möglich, flexible Regeln zu definieren, anhand derer festgelegt wird, ob ein übergeordneter Service basierend auf den Statuswerten der untergeordneten Services in den „Problemstatus“ wechseln soll. Die Problemdaten der Services können anschließend zur Berechnung der SLA und zum Senden von SLA-Berichten auf Grundlage eines flexiblen Satzes von Bedingungen verwendet werden.

Die Serviceüberwachung wird im Menü „Services“ konfiguriert, das aus den folgenden Abschnitten besteht:

- Services

Im Abschnitt „Services“ können Sie eine Hierarchie Ihrer überwachten Infrastruktur erstellen, indem Sie übergeordnete Services und anschließend untergeordnete Services zu den übergeordneten Services hinzufügen.

Neben der Konfiguration des Servicebaums bietet dieser Abschnitt einen Überblick über die gesamte Infrastruktur und ermöglicht es Ihnen, schnell die Probleme zu identifizieren, die zu einer Änderung des Servicestatus geführt haben.

- SLA

In diesem Abschnitt können Sie Service-Level-Agreements definieren und Service-Level-Ziele für bestimmte Services festlegen.

- SLA-Bericht

In diesem Abschnitt können Sie SLA-Berichte anzeigen.

Serviceaktionen

Sie können auch Service-Aktionen konfigurieren.

Serviceaktionen sind optional und ermöglichen Folgendes:

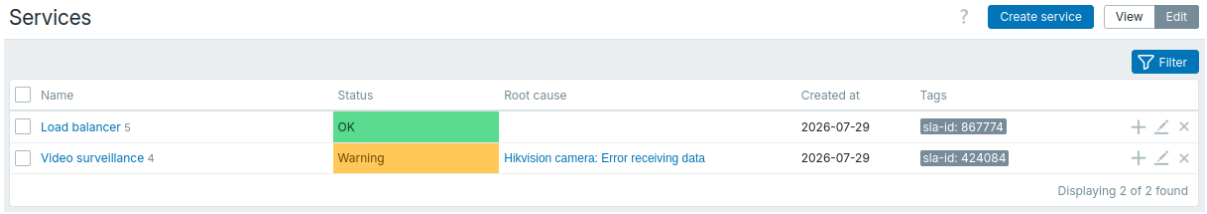
- eine Benachrichtigung zu senden, dass ein Service ausgefallen ist
- bei einer Änderung des Servicestatus einen Remote-Befehl auf dem Zabbix-Server auszuführen
- eine Wiederherstellungsbenachrichtigung zu senden, wenn ein Service wieder verfügbar ist.

Siehe auch:

- Beispiel für die Konfiguration der SLA-Überwachung
- Hinweise zum Upgrade von Services aus Zabbix-Versionen vor 6.0

Service tree

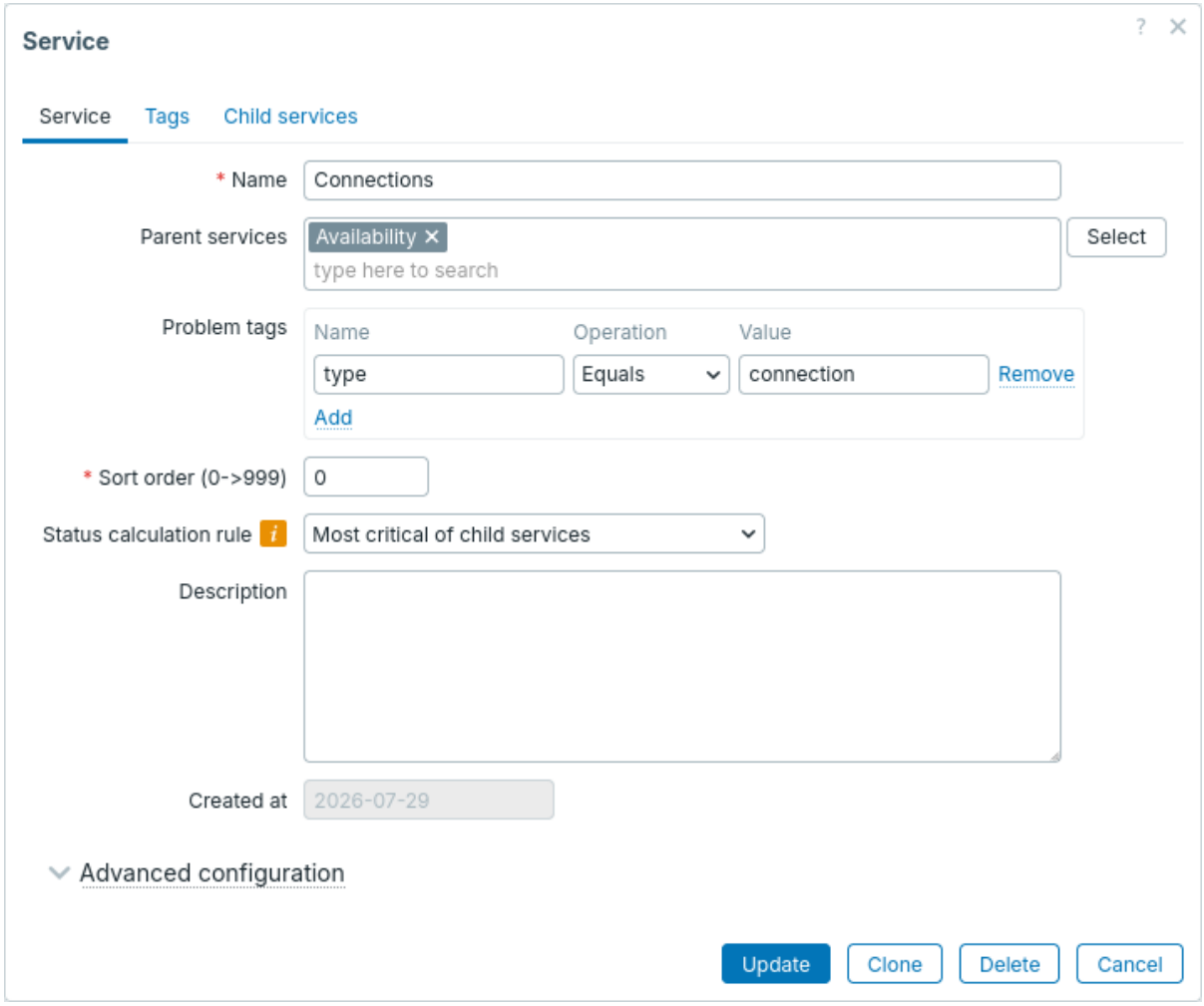
Der Servicebaum wird im Menübereich *Services* > *Services* konfiguriert. Wechseln Sie in der oberen rechten Ecke von der Ansicht in den Bearbeitungsmodus.



Um einen neuen Service zu konfigurieren, klicken Sie in der oberen rechten Ecke auf die Schaltfläche *Create service*.

Um schnell einen untergeordneten Service hinzuzufügen, können Sie alternativ auf das Plus-Symbol neben dem übergeordneten Service klicken. Dadurch wird dasselbe Konfigurationsformular für den Service geöffnet, wobei der Parameter *Parent services* bereits vorausgefüllt ist.

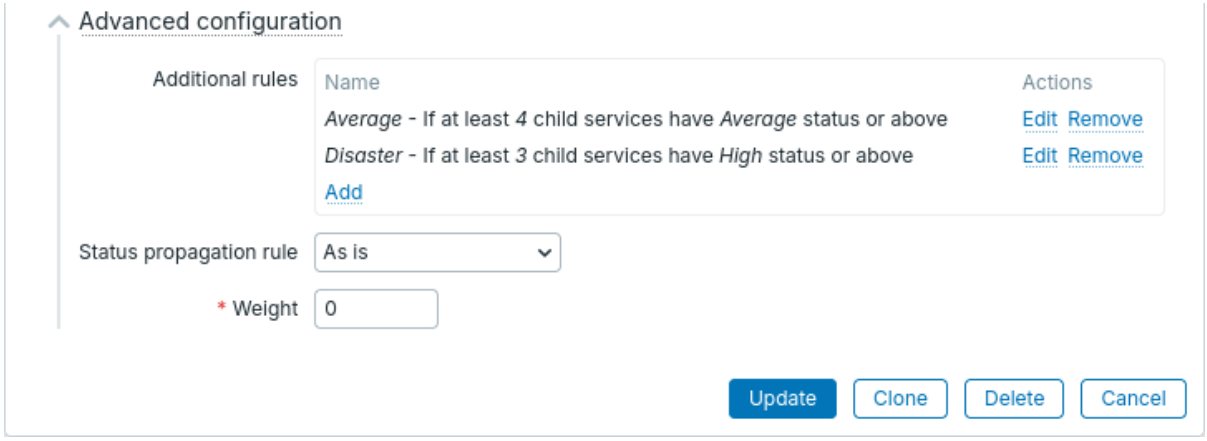
Service-Konfiguration Im Tab **Service** geben Sie die erforderlichen Service-Parameter an:



Alle obligatorischen Eingabefelder sind mit einem roten Sternchen markiert.

Parameter	Beschreibung
Name	Service-Name.
Parent services	Übergeordnete Services, zu denen der Service gehört. Lassen Sie dieses Feld leer, wenn Sie den Service der obersten Ebene hinzufügen. Ein Service kann mehrere übergeordnete Services haben. In diesem Fall wird er im Service-Baum unter jedem der übergeordneten Services angezeigt.
Problem tags	Geben Sie Tags an, um Problemdaten dem Service zuzuordnen. Es können mehrere Bedingungen festgelegt werden. Die Übereinstimmung des Tag-Namens ist immer groß-/kleinschreibungssensitiv. Für jede Bedingung stehen zwei Operatoren zur Verfügung: **Equals** - die angegebenen Tag-Namen und Werte einschließen (groß-/kleinschreibungssensitiv) **Contains** - die angegebenen Tag-Namen einschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, groß-/kleinschreibungssensitiv)
Sort order	Sortierreihenfolge für die Anzeige, die niedrigste kommt zuerst.
Status calculation rule	Regel zur Berechnung des Service-Status: **Most critical if all children have problems** - das kritischste Problem in den untergeordneten Services wird zur Farbgebung des Service-Status verwendet, wenn alle untergeordneten Services Probleme haben **Most critical of child services** - das kritischste Problem in den untergeordneten Services wird zur Farbgebung des Service-Status verwendet **Set status to OK** - den Service-Status nicht berechnen Zusätzliche Regeln zur Statusberechnung können in den Optionen der erweiterten Konfiguration konfiguriert werden.
Description	Service-Beschreibung.
Created at	Der Zeitpunkt, zu dem der Service erstellt wurde; wird beim Bearbeiten eines vorhandenen Services angezeigt.
Advanced configuration	Klicken Sie auf die Überschrift *Advanced configuration*, um die Optionen der erweiterten Konfiguration anzuzeigen.

Erweiterte Konfiguration



Parameter	Beschreibung
Zusätzliche Regeln	Klicken Sie auf *Add*, um zusätzliche Regeln zur Statusberechnung zu konfigurieren.
Status setzen auf	Setzt den Servicestatus im Falle einer passenden Bedingung auf *OK* (Standard), *Not classified*, *Information*, *Warning*, *Average*, *High* oder *Disaster*.
Bedingung	Wählen Sie die Bedingung für direkte untergeordnete Services aus: **wenn mindestens (N) untergeordnete Services den Status (Status) oder höher haben** **wenn mindestens (N%) der untergeordneten Services den Status (Status) oder höher haben** **wenn weniger als (N) untergeordnete Services den Status (Status) oder niedriger haben** **wenn weniger als (N%) der untergeordneten Services den Status (Status) oder niedriger haben** **wenn das Gewicht der untergeordneten Services mit dem Status (Status) oder höher mindestens (W) beträgt** **wenn das Gewicht der untergeordneten Services mit dem Status (Status) oder höher mindestens (N%) beträgt** **wenn das Gewicht der untergeordneten Services mit dem Status (Status) oder niedriger weniger als (W) beträgt** **wenn das Gewicht der untergeordneten Services mit dem Status (Status) oder niedriger weniger als (N%) beträgt** Wenn mehrere Bedingungen angegeben sind und die Situation mehr als einer Bedingung entspricht, wird die höchste Schwere festgelegt.
N (W)	Legen Sie den Wert von N oder W (1-100000) bzw. N% (1-100) in der Bedingung fest.
Status	Wählen Sie den Wert von *Status* in der Bedingung aus: *OK* (Standard), *Not classified*, *Information*, *Warning*, *Average*, *High* oder *Disaster*.
Regel zur Statusweitergabe	Regel zur Weitergabe des Servicestatus an den übergeordneten Service: **As is** - der Status wird unverändert weitergegeben **Increase by** - Sie können den weitergegebenen Status um 1 bis 5 Schweregrade erhöhen **Decrease by** - Sie können den weitergegebenen Status um 1 bis 5 Schweregrade verringern **Ignore this service** - der Status wird überhaupt nicht an den übergeordneten Service weitergegeben **Fixed status** - der Status wird statisch weitergegeben, d. h. immer derselbe
Gewicht	Gewicht des Services (Ganzzahl im Bereich von 0 (Standard) bis 1000000).

Note:
Zusätzliche Regeln zur Statusberechnung können nur verwendet werden, um den Schweregrad über den anhand des Hauptparameters *Status calculation rule* berechneten Wert zu erhöhen. Wenn der Status gemäß den zusätzlichen Regeln Warning sein sollte, der Status gemäß der *Status calculation rule* jedoch Disaster ist, hat der Service den Status Disaster.

Die Registerkarte *Tags* enthält Service-Level-Tags. Service-Level-Tags werden verwendet, um einen Service zu identifizieren. Tags dieses Typs werden nicht verwendet, um Probleme dem Service zuzuordnen (verwenden Sie dafür *Problem-Tags* aus der ersten Registerkarte).

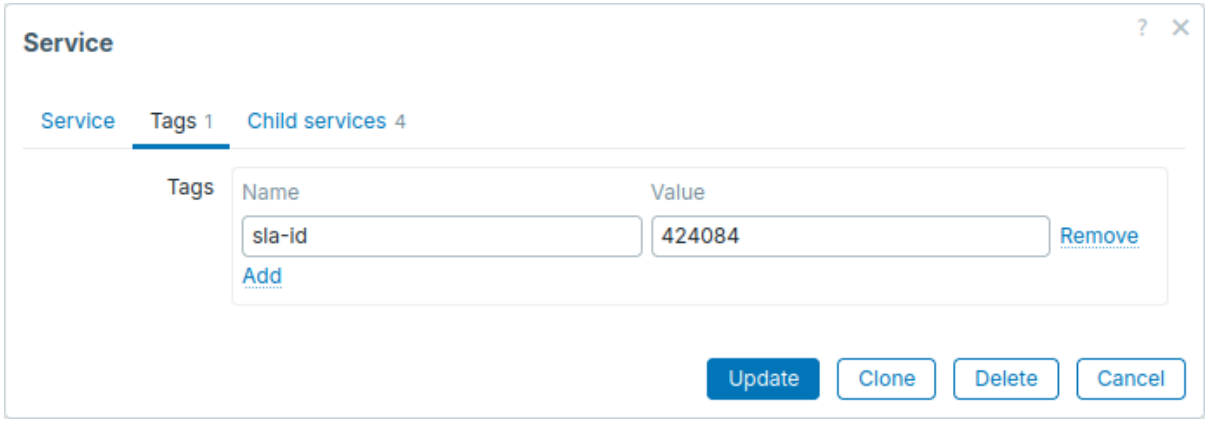
Auf der Registerkarte *Untergeordnete Services* können Sie abhängige Services angeben. Klicken Sie auf *Hinzufügen*, um einen Service aus der Liste der vorhandenen Services hinzuzufügen. Wenn Sie einen neuen untergeordneten Service hinzufügen möchten, speichern Sie diesen Service zuerst und klicken Sie anschließend auf ein Plus-Symbol neben dem gerade erstellten Service.

Tags Es gibt zwei verschiedene Arten von Tags in Services:

- Service-Tags
- Problem-Tags

Service-Tags

Service-Tags werden verwendet, um Services mit service actions und SLAs abzugleichen. Diese Tags werden im Konfigurations-Tab *Tags* des Services angegeben. Für die Zuordnung von SLAs wird die *OR*-Logik verwendet: Ein Service wird einem SLA zugeordnet, wenn er mindestens einen passenden Tag enthält. Bei service actions sind die Zuordnungsregeln konfigurierbar und können entweder die Logik *AND*, *OR* oder *AND/OR* verwenden.

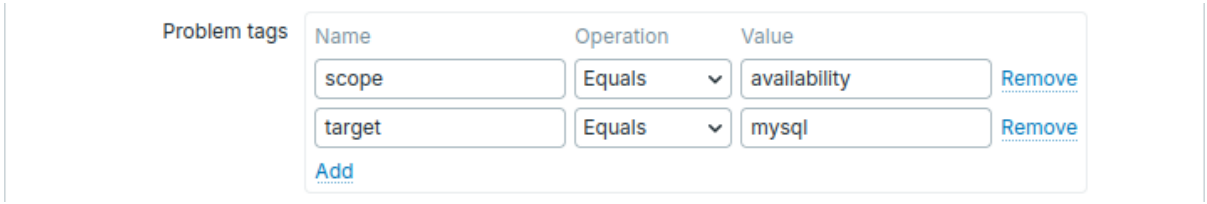


Problem-Tags

Problem-Tags werden verwendet, um Probleme und Services zuzuordnen. Diese Tags werden auf der Registerkarte der primären Service-Konfiguration angegeben.

Nur untergeordnete Services der niedrigsten Hierarchieebene dürfen definierte Problem-Tags haben und direkt Problemen zugeordnet werden. Wenn Problem-Tags übereinstimmen, ändert sich der Status des Services auf denselben Status wie das Problem. Bei mehreren Problemen hat ein Service den Status des schwerwiegendsten Problems. Der Status eines übergeordneten Services wird dann auf Grundlage der Status der untergeordneten Services gemäß den Regeln zur Statusberechnung ermittelt.

Wenn mehrere Tags angegeben sind, wird eine *UND*-Logik verwendet: Ein Problem muss alle in der Service-Konfiguration angegebenen Tags besitzen, um dem Service zugeordnet zu werden.

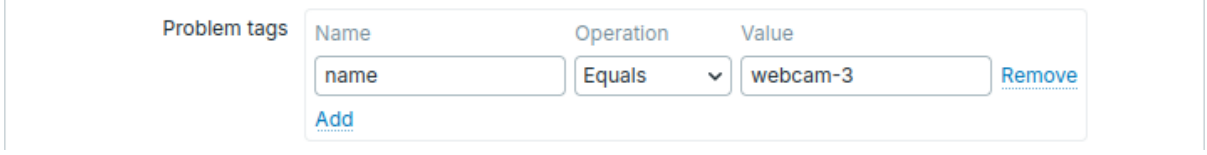


Note:
Ein Problem in Zabbix erbt Tags aus der gesamten Kette von Vorlagen, Hosts, Datenpunkten, Web-Szenarien und Auslösern. Jeder dieser Tags kann verwendet werden, um Probleme Services zuzuordnen.

Beispiel:

Das Problem *Web camera 3 is down* hat die Tags `type:video-surveillance`, `floor:1` und `name:webcam-3` sowie den Status *Warning*

Der Service **Web camera 3** hat nur den angegebenen Problem-Tag: `name:webcam-3`



Der Servicestatus wird von *OK* auf *Warning* geändert, wenn dieses Problem erkannt wird.

Wenn der Service **Web camera 3** die Problem-Tags `name:webcam-3` und `floor:2` hätte, würde sein Status bei Erkennung des Problems nicht geändert werden, da die Bedingungen nur teilweise erfüllt sind.

Note:
Die unten beschriebenen Schaltflächen sind nur sichtbar, wenn sich der Abschnitt *Services* im Bearbeitu

Vorhandene Services ändern

Um einen vorhandenen Service zu bearbeiten, klicken Sie auf das Stiftsymbol neben dem Service.

Um einen vorhandenen Service zu klonen, klicken Sie auf das Stiftsymbol, um seine Konfiguration zu öffnen, und klicken Sie dann auf die Schaltfläche *Clone*. Wenn ein Service geklont wird, bleiben seine übergeordneten Verknüpfungen erhalten, während die untergeordneten Verknüpfungen nicht übernommen werden.

Um einen Service zu löschen, klicken Sie auf das Symbol `x` daneben. Wenn Sie einen übergeordneten Service löschen, werden seine untergeordneten Services nicht gelöscht, sondern eine Ebene höher im Servicebaum verschoben (untergeordnete Services der 1. Ebene erhalten dieselbe Ebene wie der gelöschte übergeordnete Service).

Zwei Schaltflächen unter der Liste der Services bieten einige Optionen für Massenbearbeitungen:

- *Mass update* - Eigenschaften von Services massenhaft aktualisieren
- *Delete* - Services löschen

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Services und klicken Sie dann auf die gewünschte Schaltfläche.

SLA

Übersicht Sobald die Services erstellt wurden, können Sie damit beginnen zu überwachen, ob ihre Leistung den Vorgaben des Service-Level-Agreements (SLA) entspricht.

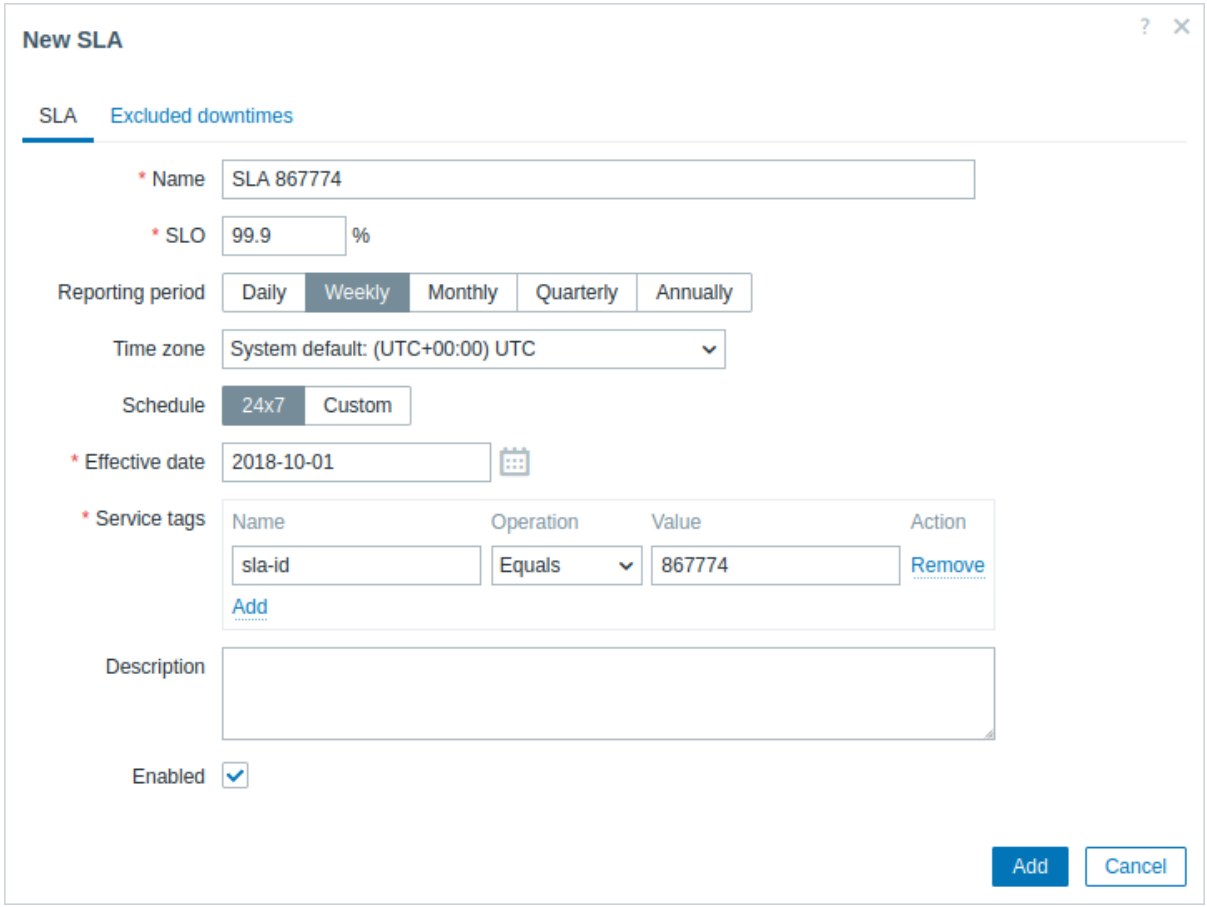
Im Menübereich *Services* > *SLA* können SLAs für verschiedene Services konfiguriert werden. Ein SLA in Zabbix definiert ein Service-Level-Ziel (SLO), den erwarteten Verfügbarkeitszeitplan und geplante Ausfallzeiten.

SLAs und Services werden anhand von Service-Tags einander zugeordnet. Dasselbe SLA kann auf mehrere Services angewendet werden – die Leistung wird für jeden passenden Service separat gemessen. Einem einzelnen Service können mehrere SLAs zugewiesen werden – die Daten für jedes SLA werden separat angezeigt.

In SLA-Berichten stellt Zabbix Daten zum Service-Level-Indikator (SLI) bereit, der die tatsächliche Serviceverfügbarkeit misst. Ob ein Service die SLA-Ziele erfüllt, wird durch den Vergleich des SLO (erwartete Verfügbarkeit in %) mit dem SLI (tatsächliche Verfügbarkeit in %) bestimmt.

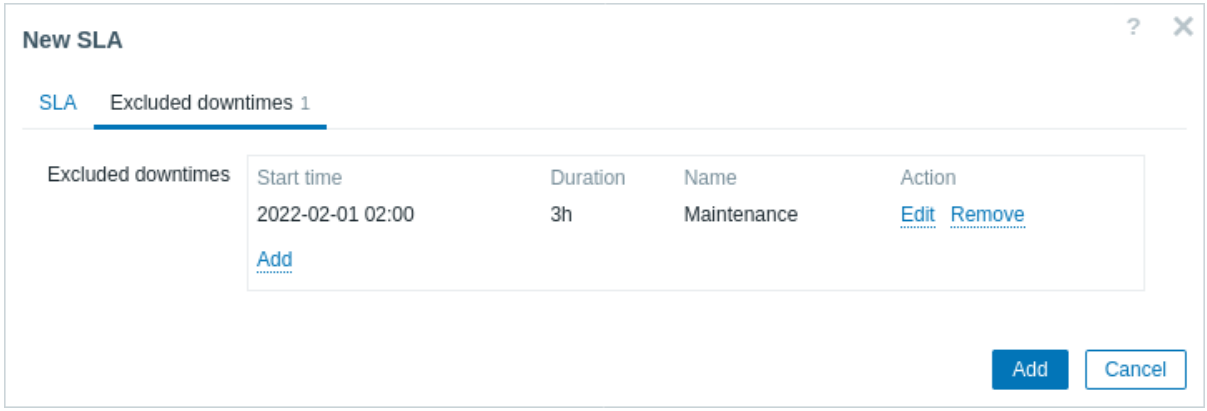
Konfiguration Um eine neue SLA zu erstellen, klicken Sie auf die Schaltfläche *SLA erstellen*.

Auf der Registerkarte *SLA* können Sie allgemeine SLA-Parameter festlegen.



Parameter	Beschreibung
Name	Geben Sie den Namen der SLA ein.
SLO	Geben Sie das Service-Level-Ziel (SLO) als Prozentsatz ein.
Berichtszeitraum	Wählen Sie den SLI-Berichtszeitraum aus, der zur Berechnung der SLI-Daten im SLA-Bericht verwendet wird: **Täglich** – jeder Tag von 00:00:00 bis 23:59:59. **Wöchentlich** – jede Woche von Sonntag 00:00:00 bis Samstag 23:59:59. **Monatlich** – jeder Monat vom ersten Tag 00:00:00 bis zum letzten Tag 23:59:59. **Vierteljährlich** – jedes Kalenderquartal: Q1 (Jan.–März), Q2 (Apr.–Juni), Q3 (Juli–Sept.), Q4 (Okt.–Dez.). **Jährlich** – jedes Kalenderjahr vom 1. Januar 00:00:00 bis zum 31. Dezember 23:59:59.
Zeitzone	Wählen Sie die Zeitzone der SLA aus.
Zeitplan	Wählen Sie den SLA-Zeitplan aus – 24x7 oder benutzerdefiniert.
Gültigkeitsdatum	Wählen Sie das Datum aus, an dem die SLA-Berechnung beginnen soll.
Service-Tags	Fügen Sie Service-Tags hinzu, um die Services zu identifizieren, auf die diese SLA angewendet werden soll. **Name** – Name des Service-Tags; muss exakt übereinstimmen und berücksichtigt die Groß-/Kleinschreibung. **Operation** – wählen Sie *Entspricht*, wenn der Tag-Wert exakt übereinstimmen muss (Groß-/Kleinschreibung wird berücksichtigt), oder *Enthält*, wenn ein Teil des Tag-Werts übereinstimmen muss (Groß-/Kleinschreibung wird nicht berücksichtigt). **Wert** – der Service-Tag-Wert, nach dem entsprechend der ausgewählten Operation gesucht wird. Die SLA wird auf einen Service angewendet, wenn mindestens ein Service-Tag übereinstimmt.
Beschreibung	Fügen Sie eine Beschreibung für die SLA hinzu.
Aktiviert	Aktivieren Sie das Kontrollkästchen, um die SLA-Berechnung zu aktivieren.

Auf der Registerkarte *Ausgeschlossene Ausfallzeiten* können Sie Ausfallzeiten festlegen, die von der SLA-Berechnung ausgeschlossen werden.



Klicken Sie auf *Hinzufügen*, um ausgeschlossene Ausfallzeiten zu konfigurieren, und geben Sie anschließend den Namen des Zeitraums, das Startdatum und die Dauer ein.

SLA-Berichte Wie ein Service im Vergleich zu einem SLA abschneidet, ist im SLA-Bericht sichtbar. SLA-Berichte können angezeigt werden:

- im Abschnitt *SLA*, indem Sie auf den Hyperlink zum SLA-Bericht klicken;
- im Abschnitt *Services*, indem Sie im Info-Tab auf den SLA-Namen klicken;
- im Dashboard-Widget *SLA report*.

Sobald ein SLA konfiguriert ist, zeigt der Tab *Info* im Abschnitt *Services* außerdem einige Informationen zur Service-Performance an.

Einrichtungsbeispiel

Übersicht Dieser Abschnitt beschreibt eine einfache Einrichtung zur Überwachung eines Zabbix-Hochverfügbarkeitsclusters als Dienst.

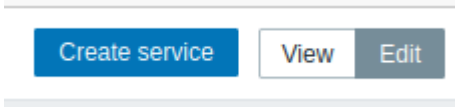
Voraussetzungen Bevor Sie die Service-Überwachung konfigurieren, müssen die Hosts konfiguriert sein:

- *HA-Knoten 1* mit mindestens einem Auslöser und einem Tag (vorzugsweise auf Auslöser-Ebene gesetzt) `component:ha-node-1`
- *HA-Knoten 2* mit mindestens einem Auslöser und einem Tag (vorzugsweise auf Auslöser-Ebene gesetzt) `component:ha-node-2`

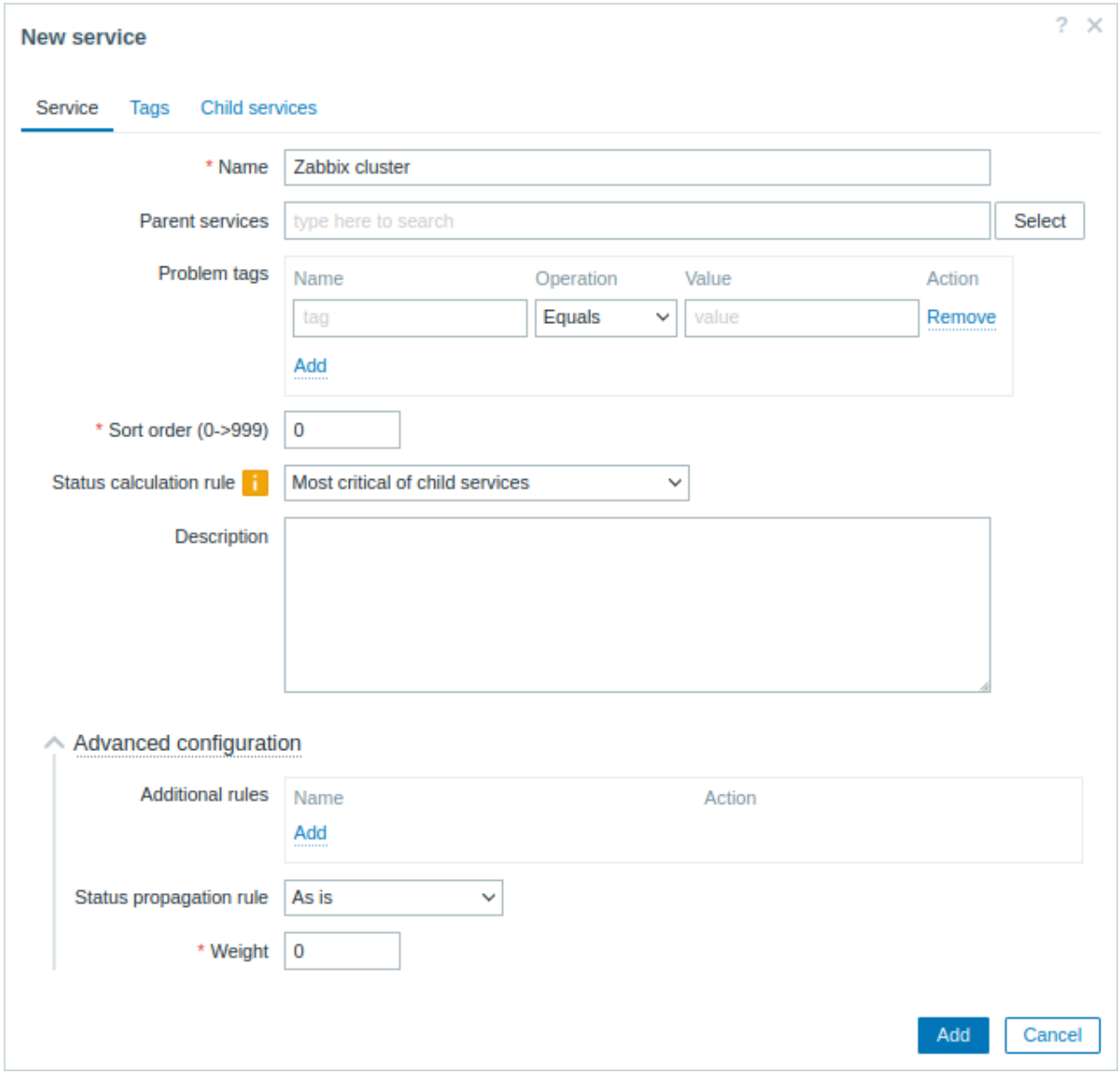
Servicetree Der nächste Schritt besteht darin, den Servicetree aufzubauen. In diesem Beispiel ist die Infrastruktur sehr einfach und besteht aus drei Services: *Zabbix cluster* (übergeordnet) und zwei untergeordnete Services *Zabbix server node 1* und *Zabbix server node 2*.

```
Zabbix cluster
|
|- Zabbix server node 1
|- Zabbix server node 2
```

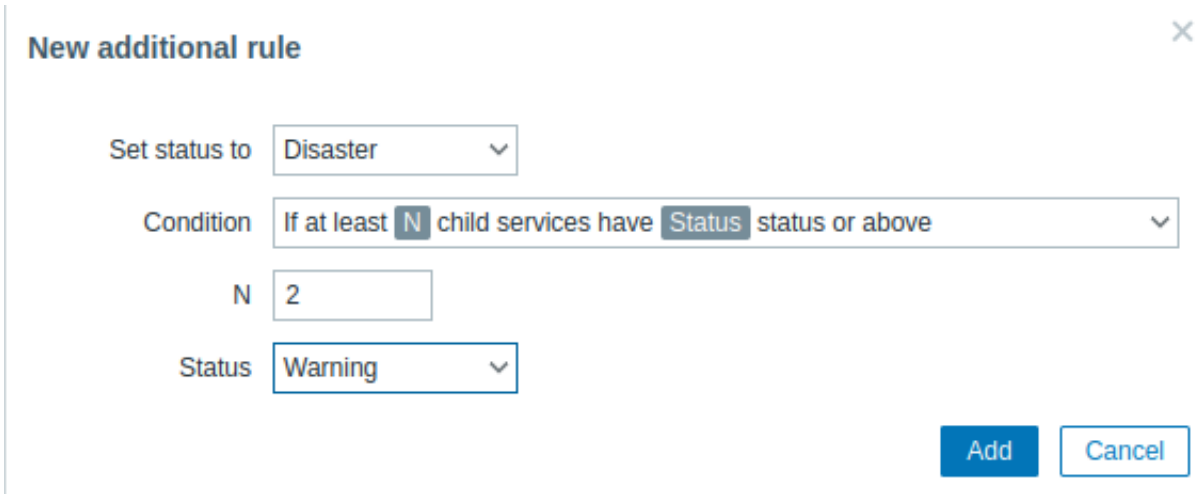
Auf der Seite *Services* aktivieren Sie den *Edit*-Modus und klicken auf *Create service*:



Geben Sie im Konfigurationsfenster für den Service den Namen *Zabbix cluster* ein und klicken Sie auf die Überschrift *Advanced configuration*, um die erweiterten Konfigurationsoptionen anzuzeigen.



Zusätzliche Regel konfigurieren:

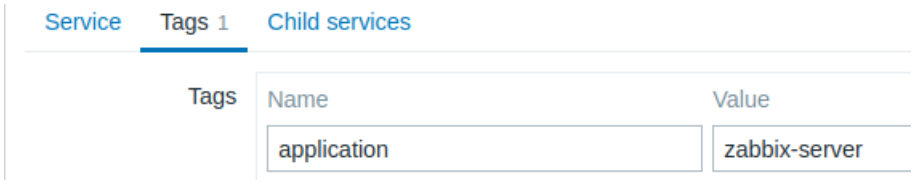


Zabbix cluster wird zwei untergeordnete Services haben - einen für jeden der HA-Knoten. Wenn bei beiden HA-Knoten Probleme mit mindestens dem Status *Warning* vorliegen, sollte der Status des übergeordneten Services auf *Disaster* gesetzt werden. Um dies zu erreichen, sollte die zusätzliche Regel wie folgt konfiguriert werden:

- Status setzen auf: Disaster
- Bedingung: Wenn mindestens `N` untergeordnete Services den Status `Status` oder höher haben
- N: 2
- Status: Warning

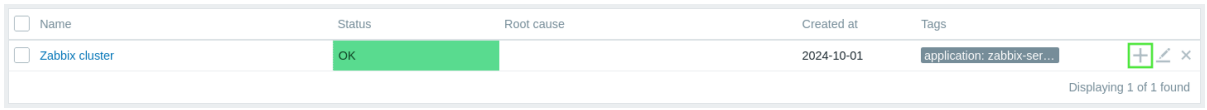
Wechseln Sie zur Registerkarte *Tags* und fügen Sie das Tag `application:zabbix-server` hinzu. Dieses Tag wird später für

Service-Aktionen und SLA-Berichte verwendet.



Speichern Sie den neuen Service.

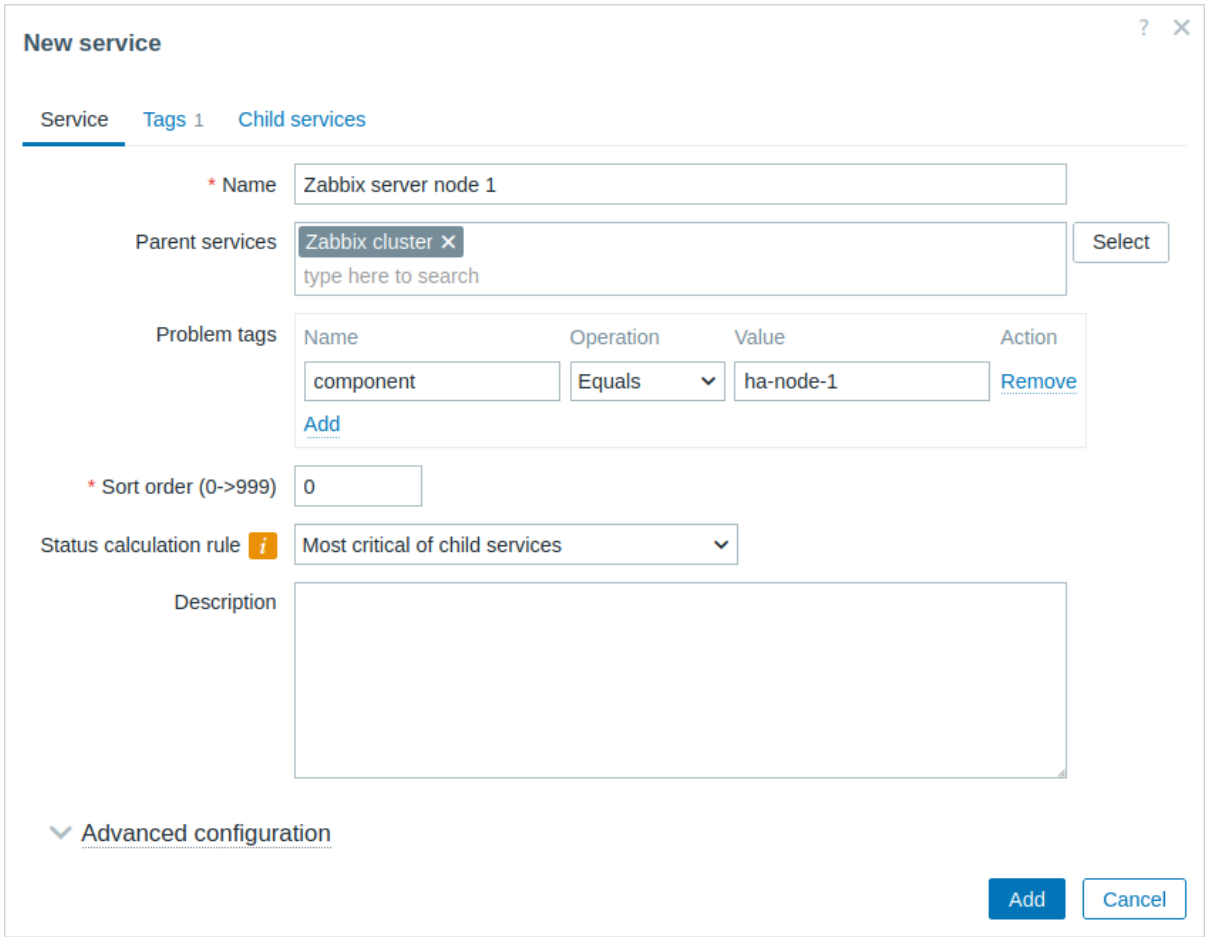
Um einen untergeordneten Service hinzuzufügen, klicken Sie auf das Plus-Symbol neben dem Zabbix-Cluster-Service (das Symbol ist nur im Modus *Bearbeiten* sichtbar).



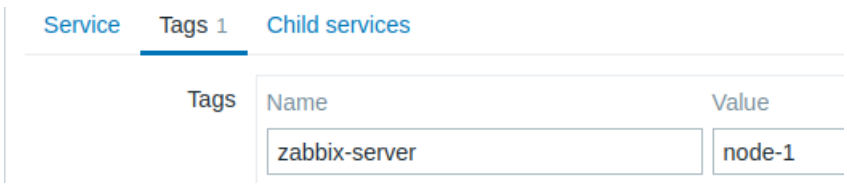
Geben Sie im Konfigurationsfenster des Services den Namen *Zabbix server node 1* ein. Beachten Sie, dass der Parameter *Übergeordnete Services* bereits mit *Zabbix cluster* vorausgefüllt ist.

Die Verfügbarkeit dieses Services wird durch Probleme auf dem Host *HA node 1* beeinflusst, die mit dem Problem-Tag `component:ha-node-1` markiert sind. Geben Sie im Parameter *Problem-Tags* Folgendes ein:

- Name: component
- Operation: Gleich
- Wert: ha-node-1



Wechseln Sie zur Registerkarte *Tags* und fügen Sie ein Service-Tag hinzu: `zabbix-server:node-1`. Dieses Tag wird später für Service-Aktionen und SLA-Berichte verwendet.



Speichern Sie den neuen Service.

Erstellen Sie einen weiteren untergeordneten Service des Zabbix-Clusters mit dem Namen „Zabbix server node 2“.

Legen Sie die Problem-Tags wie folgt fest:

- Name: component
- Operation: Equals
- Wert: ha-node-2

Wechseln Sie zur Registerkarte *Tags* und fügen Sie ein Service-Tag hinzu: `zabbix-server:node-2`.

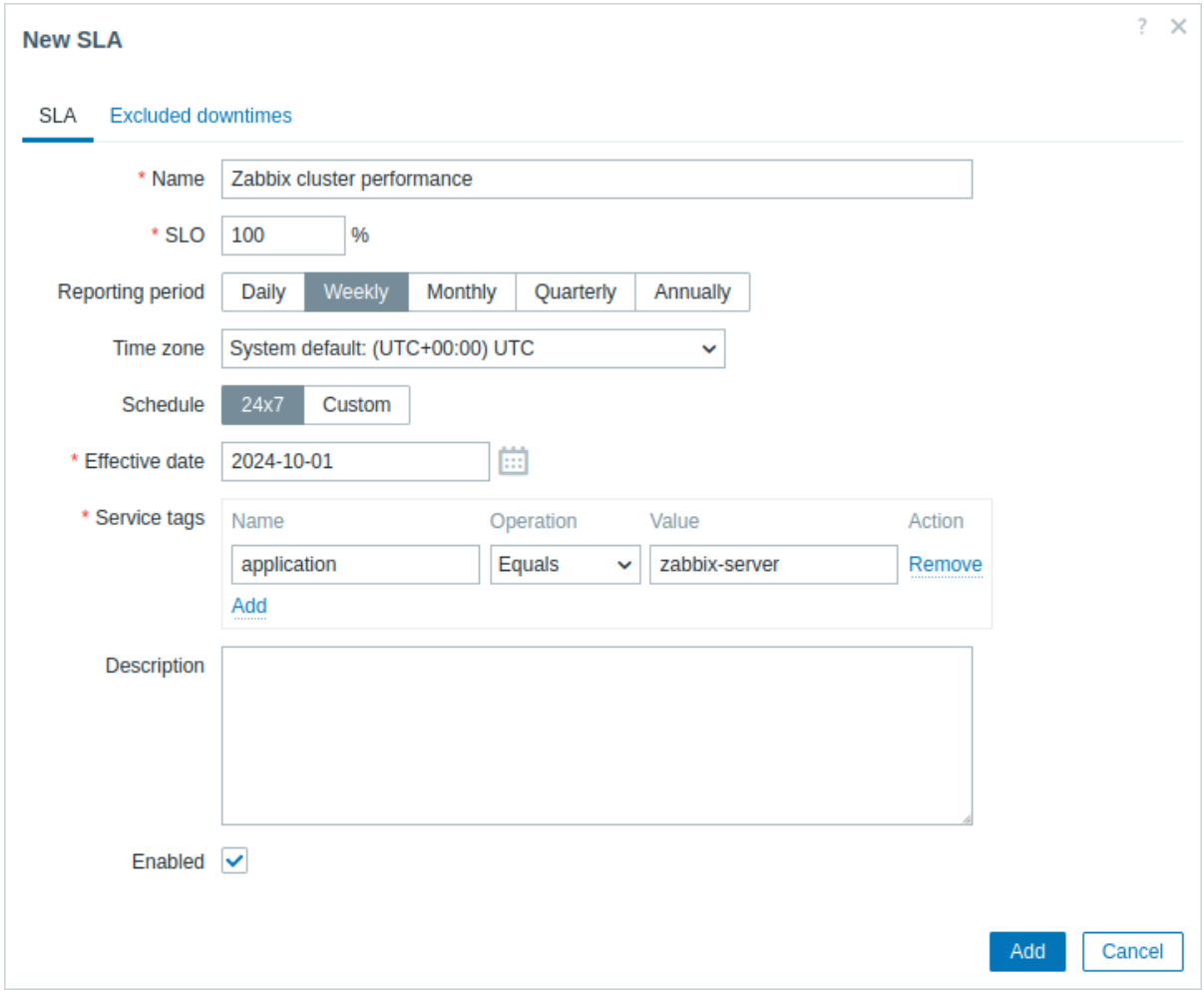
Speichern Sie den neuen Service.

SLA In diesem Beispiel beträgt die erwartete Leistung des Zabbix-Clusters 100 %, ausgenommen eine halbjährliche einstündige Wartungsperiode.

Zunächst müssen Sie eine neue Service-Level-Vereinbarung hinzufügen:

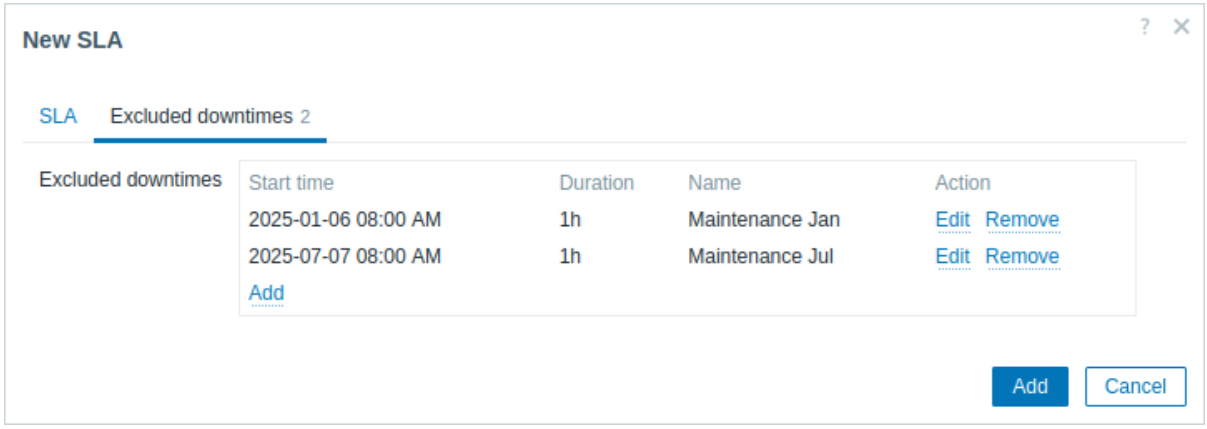
1. Gehen Sie zu *Services* > *SLA*.
2. Klicken Sie auf *Create SLA*.
3. Geben Sie den Namen *Zabbix cluster performance* ein und setzen Sie das SLO auf 100 %. Der Service Zabbix cluster verfügt über das Service-Tag `application:zabbix-server`.
4. Um diese SLA zur Messung der Leistung des Zabbix-Clusters zu verwenden, geben Sie im Parameter *Service tags* Folgendes an:

- Name: application
- Operation: Equals
- Value: zabbix-server



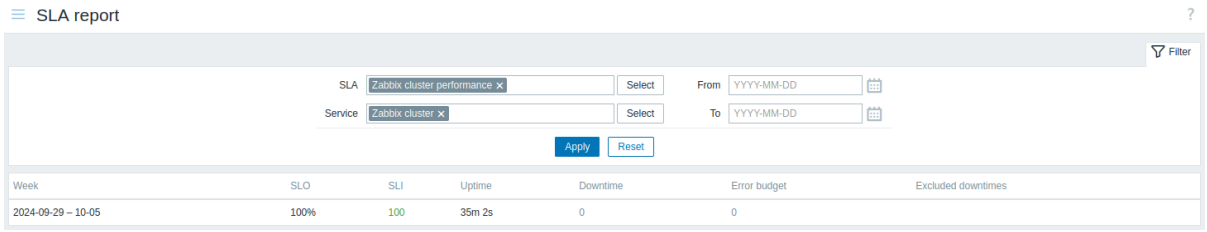
In einer realen Umgebung können Sie außerdem den gewünschten Berichtszeitraum, die Zeitzone und das Startdatum aktualisieren oder den Zeitplan von 24/7 auf einen benutzerdefinierten Zeitplan ändern. Für dieses Beispiel sind die Standardeinstellungen ausreichend.

5. Wechseln Sie zur Registerkarte *Excluded downtimes* und fügen Sie Ausfallzeiten für geplante Wartungszeiträume hinzu, um diese Zeiträume aus der SLA-Berechnung auszuschließen.
6. Klicken Sie im Abschnitt *Excluded downtimes* auf den Link *Add*, geben Sie den Namen der Ausfallzeit, die geplante Startzeit und die Dauer ein.

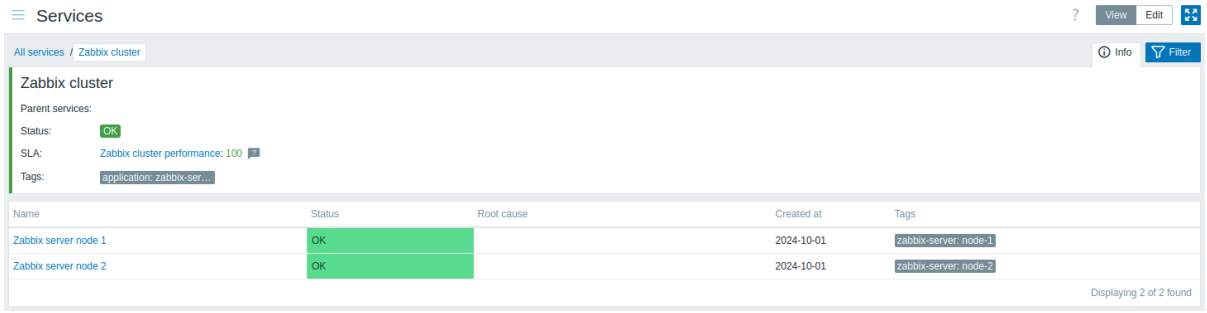


7. Klicken Sie auf *Add*, um die neue SLA zu speichern.

Wechseln Sie zum Abschnitt mit den SLA-Berichten, um den SLA-Bericht für den Zabbix-Cluster anzuzeigen.



Die SLA-Informationen können auch im Abschnitt *Services* überprüft werden.



Upgrade der Dienstüberwachung

Übersicht In Zabbix 6.0 wurde die Funktionalität der Serviceüberwachung erheblich überarbeitet (siehe Was ist neu in Zabbix 6.0.0 für die Liste der Änderungen).

Diese Seite beschreibt, wie sich Services und SLAs, die in früheren Zabbix-Versionen definiert wurden, während eines Upgrades auf Zabbix 6.0 oder neuer ändern.

Dienste In älteren Zabbix-Versionen hatten Dienste zwei Arten von Abhängigkeiten: weich und hart. Nach einem Upgrade werden alle Abhängigkeiten gleich behandelt.

Wenn ein Dienst ”Child service” zuvor über eine harte Abhängigkeit mit ”Parent service 1” und zusätzlich über eine weiche Abhängigkeit mit ”Parent service 2” verknüpft war, hat der Dienst ”Child service” nach einem Upgrade zwei übergeordnete Dienste: ”Parent service 1” und ”Parent service 2”.

Die triggerbasierte Zuordnung zwischen Problemen und Diensten wurde durch eine tagbasierte Zuordnung ersetzt. In Zabbix 6.0 und neuer enthält das Dienstkonfigurationsformular einen neuen Parameter *Problem tags*, mit dem ein oder mehrere Tag-Namen- und Wertpaare für den Problemabgleich angegeben werden können. Trigger, die mit einem Dienst verknüpft waren, erhalten ein neues Tag `ServiceLink : <trigger ID>:<trigger name>` (der Tag-Wert wird auf 32 Zeichen gekürzt). Verknüpfte Dienste erhalten `ServiceLink` Problem-Tag mit demselben Wert.

Regeln zur Statusberechnung

Der ”Status calculation algorithm” wird anhand der folgenden Regeln aktualisiert:

- Nicht berechnen → Status auf OK setzen

- Problem, wenn mindestens ein untergeordnetes Element ein Problem hat → Das kritischste der untergeordneten Services
- Problem, wenn alle untergeordneten Elemente Probleme haben → Das kritischste, wenn alle untergeordneten Elemente Probleme haben

SLAs Früher mussten SLA-Ziele für jeden Service separat definiert werden. Seit Zabbix 6.0 ist SLA zu einer eigenen Entität geworden, die Informationen über den Service-Zeitplan, das erwartete Service-Level-Objective (SLO) und Ausfallzeiten enthält, die von der Berechnung ausgeschlossen werden. Nach der Konfiguration kann ein SLA mehreren Services über Service-Tags zugewiesen werden.

Während eines Upgrades:

- Identische SLAs, die für jeden Service definiert wurden, werden gruppiert, und für jede Gruppe wird ein SLA erstellt.
- Jeder betroffene Service erhält ein spezielles Tag `SLA:<ID>`, und dasselbe Tag wird im Parameter *Service tags* des entsprechenden SLA angegeben.
- Die Service-Erstellungszeit, eine neue Kennzahl in SLA-Berichten, wird für vorhandene Services auf 01/01/2000 00:00 gesetzt.

Verschlüsselung und Sicherheit

Verschlüsselung Zabbix unterstützt verschlüsselte Kommunikation zwischen Zabbix-Komponenten mithilfe des Transport Layer Security (TLS)-Protokolls v.1.2 und 1.3 (je nach Kryptobibliothek). Zertifikatsbasierte und auf vorab gemeinsam genutzten Schlüsseln basierende Verschlüsselung werden unterstützt.

Die Verschlüsselung kann für Verbindungen konfiguriert werden:

- Zwischen Zabbix Server, Zabbix Proxy, Zabbix Agent, Zabbix Webdienst, zabbix_sender und zabbix_get-Dienstprogrammen
- Zur Zabbix-Datenbank vom Zabbix Frontend und Server/Proxy
- Zwischen Zabbix Frontend und Zabbix Server

Die Verschlüsselung ist optional und für einzelne Komponenten konfigurierbar:

- Einige Proxies und Agents können so konfiguriert werden, dass sie mit dem Server eine zertifikatsbasierte Verschlüsselung verwenden, während andere eine auf vorab gemeinsam genutzten Schlüsseln basierende Verschlüsselung nutzen und wieder andere weiterhin unverschlüsselt kommunizieren (wie zuvor).
- Server (Proxy) kann für verschiedene Hosts unterschiedliche Verschlüsselungskonfigurationen verwenden.

Zabbix-Daemonprogramme verwenden einen einzigen Listening-Port für verschlüsselte und unverschlüsselte eingehende Verbindungen. Das Hinzufügen von Verschlüsselung erfordert keine neuen Ports in Firewalls.

Einschränkungen

- Private Schlüssel werden in Dateien im Klartext gespeichert, die von Zabbix-Komponenten während des Starts gelesen werden können.
- Vorab gemeinsam genutzte Schlüssel werden im Zabbix Frontend eingegeben und in der Zabbix-Datenbank im Klartext gespeichert.
- Die integrierte Verschlüsselung schützt die Kommunikation zwischen dem Webserver, auf dem das Zabbix Frontend läuft, und dem Webbrowser des Benutzers nicht.
- Derzeit wird jede verschlüsselte Verbindung mit einem vollständigen TLS-Handshake geöffnet; Session-Caching und Tickets sind nicht implementiert.
- Das Hinzufügen von Verschlüsselung erhöht die Zeit für Datenpunkt-Prüfungen und Aktionen, abhängig von der Netzwerklatenz:
  - Wenn beispielsweise die Paketverzögerung 100 ms beträgt, dauert das Öffnen einer TCP-Verbindung und das Senden einer unverschlüsselten Anfrage etwa 200 ms. Mit Verschlüsselung werden etwa 1000 ms für den Aufbau der TLS-Verbindung hinzugefügt.
  - Möglicherweise müssen Timeouts erhöht werden, andernfalls können einige Datenpunkte und Aktionen, die Remote-Skripte auf Agenten ausführen, mit unverschlüsselten Verbindungen funktionieren, aber bei verschlüsselten Verbindungen aufgrund eines Timeouts fehlschlagen.
- Verschlüsselung wird von der Netzwerkerkennung nicht unterstützt. Zabbix-Agent-Prüfungen, die von der Netzwerkerkennung durchgeführt werden, sind unverschlüsselt, und wenn der Zabbix Agent so konfiguriert ist, dass er unverschlüsselte Verbindungen ablehnt, werden solche Prüfungen nicht erfolgreich sein.

Zabbix mit Verschlüsselungsunterstützung kompilieren Für die Unterstützung der Verschlüsselung muss Zabbix mit einer der unterstützten Kryptobibliotheken kompiliert und gelinkt werden:

- GnuTLS - ab Version 3.1.18

- OpenSSL - Versionen 1.0.1, 1.0.2, 1.1.0, 1.1.1, 3.0.x - 3.5.x
- LibreSSL - getestet mit den Versionen 2.7.4, 2.8.2:
  - LibreSSL 2.6.x wird nicht unterstützt
  - LibreSSL wird als kompatibler Ersatz für OpenSSL unterstützt; die neuen LibreSSL-spezifischen API-Funktionen `tls_*()` werden nicht verwendet. Mit LibreSSL kompilierte Zabbix-Komponenten können kein PSK verwenden, sondern nur Zertifikate.

Note:
Weitere Informationen zur Einrichtung von SSL für das Zabbix-Frontend finden Sie in diesen Best Practices.

Die Bibliothek wird angegeben, indem die entsprechende Option für das Skript „configure“ festgelegt wird:

- `--with-gnutls[=DIR]`
- `--with-openssl[=DIR]` (wird auch für LibreSSL verwendet)

Um beispielsweise die Quellen für Server und Agent mit *OpenSSL* zu konfigurieren, können Sie Folgendes verwenden:

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

Verschiedene Zabbix-Komponenten können mit unterschiedlichen Kryptobibliotheken kompiliert werden (z. B. ein Server mit *OpenSSL* und ein Agent mit *GnuTLS*).

Attention:
Wenn Sie vorab gemeinsam genutzte Schlüssel (PSK) verwenden möchten, sollten Sie in Zabbix-Komponenten, die PSKs verwenden, *GnuTLS* oder *OpenSSL 1.1.0* (oder neuer) einsetzen. Die Bibliotheken *GnuTLS* und *OpenSSL 1.1.0* unterstützen PSK-Verschlüsselungssammlungen mit Perfect Forward Secrecy. Ältere Versionen der *OpenSSL*-Bibliothek (1.0.1, 1.0.2c) unterstützen PSKs ebenfalls, aber die verfügbaren PSK-Verschlüsselungssammlungen bieten keine Perfect Forward Secrecy.

Verwaltung der Verbindungsverschlüsselung Verbindungen in Zabbix können Folgendes verwenden:

- keine Verschlüsselung (Standard)
- Verschlüsselung auf Basis von RSA-Zertifikaten
- PSK-basierte Verschlüsselung

Für die Angabe der Verschlüsselung zwischen Zabbix-Komponenten werden zwei wichtige Parameter verwendet:

- TLSConnect - gibt an, welche Verschlüsselung für ausgehende Verbindungen verwendet werden soll (unverschlüsselt, PSK oder Zertifikat)
- TLSAccept - gibt an, welche Verbindungstypen für eingehende Verbindungen zulässig sind (unverschlüsselt, PSK oder Zertifikat). Es können ein oder mehrere Werte angegeben werden.

`TLSConnect` wird in den Konfigurationsdateien für den Zabbix Proxy (im aktiven Modus werden nur Verbindungen zum Server angegeben) und den Zabbix Agent (für aktive Prüfungen) verwendet. Im Zabbix Frontend entspricht `TLSConnect` dem Feld *Verbindungen zum Host* unter *Datensammlung* > *Hosts* > <ein Host> > Registerkarte *Verschlüsselung* sowie dem Feld *Verbindungen zum Proxy* unter *Administration* > *Proxies* > <ein Proxy> > Registerkarte *Verschlüsselung*. Wenn der konfigurierte Verschlüsselungstyp für die Verbindung fehlschlägt, werden keine anderen Verschlüsselungstypen versucht.

`TLSAccept` wird in den Konfigurationsdateien für den Zabbix Proxy (im passiven Modus werden nur Verbindungen vom Server angegeben) und den Zabbix Agent (für passive Prüfungen) verwendet. Im Zabbix Frontend entspricht `TLSAccept` dem Feld *Verbindungen vom Host* unter *Datensammlung* > *Hosts* > <ein Host> > Registerkarte *Verschlüsselung* sowie dem Feld *Verbindungen vom Proxy* unter *Administration* > *Proxies* > <ein Proxy> > Registerkarte *Verschlüsselung*.

Normalerweise konfigurieren Sie nur einen Verschlüsselungstyp für eingehende Verbindungen. Sie können jedoch den Verschlüsselungstyp beispielsweise mit minimaler Ausfallzeit und der Möglichkeit eines Rollbacks von unverschlüsselt auf zertifikatsbasierte Verschlüsselung umstellen. Gehen Sie dazu wie folgt vor:

- Setzen Sie `TLSAccept=unencrypted,cert` in der Agent-Konfigurationsdatei und starten Sie den Zabbix Agent neu.
- Testen Sie die Verbindung mit `zabbix_get` zum Agent unter Verwendung eines Zertifikats. Wenn dies funktioniert, können Sie die Verschlüsselung für diesen Agent im Zabbix Frontend unter *Datensammlung* > *Hosts* > <ein Host> > Registerkarte *Verschlüsselung* neu konfigurieren, indem Sie *Verbindungen zum Host* auf *Zertifikat* setzen.
- Sobald der Server-Konfigurations-Cache aktualisiert wurde (und die Proxy-Konfiguration aktualisiert wurde, wenn der Host durch einen Proxy überwacht wird), werden die Verbindungen zu diesem Agent verschlüsselt.
- Wenn alles wie erwartet funktioniert, können Sie `TLSAccept=cert` in der Agent-Konfigurationsdatei setzen und den Zabbix Agent neu starten. Der Agent akzeptiert nun nur noch verschlüsselte, zertifikatsbasierte Verbindungen. Unverschlüsselte und PSK-basierte Verbindungen werden abgelehnt.

Auf ähnliche Weise funktioniert dies auf dem Server und dem Proxy. Wenn im Zabbix Frontend in der Host-Konfiguration *Verbindungen vom Host* auf „Zertifikat“ gesetzt ist, werden vom Agent (aktive Prüfungen) und von `zabbix\_sender` (Trapper-Datenpunkte) nur zertifikatsbasierte verschlüsselte Verbindungen akzeptiert.

Wahrscheinlich konfigurieren Sie eingehende und ausgehende Verbindungen so, dass sie denselben Verschlüsselungstyp oder überhaupt keine Verschlüsselung verwenden. Technisch ist es jedoch möglich, sie asymmetrisch zu konfigurieren, beispielsweise zertifikatsbasierte Verschlüsselung für eingehende und PSK-basierte Verschlüsselung für ausgehende Verbindungen.

Die Verschlüsselungskonfiguration für jeden Host wird im Zabbix Frontend unter *Datensammlung* > *Hosts* in der Spalte *Agent-Verschlüsselung* angezeigt. Beispiel:

Beispiel	Verbindungen zum Host	Zulässige Verbindungen vom Host	Abgelehnte Verbindungen vom Host
NONE	Unverschlüsselt	Unverschlüsselt	Verschlüsselt, zertifikatsbasiert und PSK-basiert verschlüsselt
CERT NONE PSK CERT	Verschlüsselt, zertifikatsbasiert	Verschlüsselt, zertifikatsbasiert	Unverschlüsselt und PSK-basiert verschlüsselt
PSK NONE PSK CERT	Verschlüsselt, PSK-basiert	Verschlüsselt, PSK-basiert	Unverschlüsselt und zertifikatsbasiert verschlüsselt
PSK NONE PSK CERT	Verschlüsselt, PSK-basiert	Unverschlüsselt und PSK-basiert verschlüsselt	Zertifikatsbasiert verschlüsselt
CERT NONE PSK CERT	Verschlüsselt, zertifikatsbasiert	Unverschlüsselt, PSK- oder zertifikatsbasiert verschlüsselt	-

Attention:
Verbindungen sind standardmäßig unverschlüsselt. Die Verschlüsselung muss für jeden Host und Proxy einzeln konfiguriert werden.

zabbix_get und zabbix_sender mit Verschlüsselung Siehe die Manpages zabbix_get und zabbix_sender zur Verwendung mit Verschlüsselung.

Ciphersuites Ciphersuites werden standardmäßig intern während des Zabbix-Starts konfiguriert.

Außerdem werden von Benutzern konfigurierte Ciphersuites für GnuTLS und OpenSSL unterstützt. Benutzer können konfigurieren Ciphersuites entsprechend ihren Sicherheitsrichtlinien. Die Verwendung dieser Funktion ist optional (die integrierten Standard-Ciphersuites funktionieren weiterhin).

Für Kryptobibliotheken, die mit den Standard-Einstellungen kompiliert wurden, führen die integrierten Zabbix-Regeln typischerweise zu den folgenden Ciphersuites (in Reihenfolge von höherer zu niedrigerer Priorität):

Library	Certificate ciphersuites	PSK ciphersuites
GnuTLS 3.1.18	TLS_ECDHE_RSA_AES_128_GCM_SHA256 TLS_ECDHE_RSA_AES_128_CBC_SHA256 TLS_ECDHE_RSA_AES_128_CBC_SHA1 TLS_RSA_AES_128_GCM_SHA256 TLS_RSA_AES_128_CBC_SHA256 TLS_RSA_AES_128_CBC_SHA1	TLS_ECDHE_PSK_AES_128_CBC_SHA256 TLS_ECDHE_PSK_AES_128_CBC_SHA1 TLS_PSK_AES_128_GCM_SHA256 TLS_PSK_AES_128_CBC_SHA256 TLS_PSK_AES_128_CBC_SHA1
OpenSSL 1.0.2c	ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA AES128-GCM-SHA256 AES128-SHA256 AES128-SHA	PSK-AES128-CBC-SHA
OpenSSL 1.1.0	ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA AES128-GCM-SHA256 AES128-CCM8 AES128-CCM AES128-SHA256 AES128-SHA	ECDHE-PSK-AES128-CBC-SHA256 ECDHE-PSK-AES128-CBC-SHA PSK-AES128-GCM-SHA256 PSK-AES128-CCM8 PSK-AES128-CCM PSK-AES128-CBC-SHA256 PSK-AES128-CBC-SHA

Library	Certificate ciphersuites	PSK ciphersuites
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

Vom Benutzer konfigurierte Ciphersuites Die integrierten Auswahlkriterien für Ciphersuites können mit vom Benutzer konfigurierten Ciphersuites überschrieben werden.

Attention:
Vom Benutzer konfigurierte Ciphersuites sind eine Funktion für fortgeschrittene Benutzer, die TLS-Ciphersuites, deren Sicherheit und die Folgen von Fehlern verstehen und sich mit der Fehlersuche bei TLS wohlfühlen.

Die integrierten Auswahlkriterien für Ciphersuites können mit den folgenden Parametern überschrieben werden:

Überschreibungsbereich	Parameter	Wert	Beschreibung
Ciphersuite-Auswahl für Zertifikate	TLSCipherCert13	Gültige OpenSSL 1.1.1 Cipher-Strings für das TLS-1.3-Protokoll (ihre Werte werden an die OpenSSL-Funktion `SSL_CTX_set_ciphersuites()` übergeben).	Zertifikatsbasierte Auswahlkriterien für Ciphersuites für TLS 1.3 Nur OpenSSL 1.1.1 oder neuer.
	TLSCipherCert	Gültige OpenSSL Cipher-Strings für TLS 1.2 oder gültige GnuTLS-Prioritätszeichenfolgen. Ihre Werte werden jeweils an die Funktionen `SSL_CTX_set_cipher_list()` bzw. `gnutls_priority_init()` übergeben.	Zertifikatsbasierte Auswahlkriterien für Ciphersuites für TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)
Ciphersuite-Auswahl für PSK	TLSCipherPSK13	Gültige OpenSSL 1.1.1 Cipher-Strings für das TLS-1.3-Protokoll (ihre Werte werden an die OpenSSL-Funktion `SSL_CTX_set_ciphersuites()` übergeben).	PSK-basierte Auswahlkriterien für Ciphersuites für TLS 1.3 Nur OpenSSL 1.1.1 oder neuer.
	TLSCipherPSK	Gültige OpenSSL Cipher-Strings für TLS 1.2 oder gültige GnuTLS-Prioritätszeichenfolgen. Ihre Werte werden jeweils an die Funktionen `SSL_CTX_set_cipher_list()` bzw. `gnutls_priority_init()` übergeben.	PSK-basierte Auswahlkriterien für Ciphersuites für TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)
Kombinierte Ciphersuite-Liste für Zertifikat und PSK	TLSCipherAll13	Gültige OpenSSL 1.1.1 Cipher-Strings für das TLS-1.3-Protokoll (ihre Werte werden an die OpenSSL-Funktion `SSL_CTX_set_ciphersuites()` übergeben).	Auswahlkriterien für Ciphersuites für TLS 1.3 Nur OpenSSL 1.1.1 oder neuer.
	TLSCipherAll	Gültige OpenSSL Cipher-Strings für TLS 1.2 oder gültige GnuTLS-Prioritätszeichenfolgen. Ihre Werte werden jeweils an die Funktionen `SSL_CTX_set_cipher_list()` bzw. `gnutls_priority_init()` übergeben.	Auswahlkriterien für Ciphersuites für TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)

Um die Auswahl der Ciphersuite in den Dienstprogrammen zabbix_get und zabbix_sender zu überschreiben, verwenden Sie die Befehlszeilenparameter:

- `--tls-cipher13`

- `--tls-cipher`

Die neuen Parameter sind optional. Wenn ein Parameter nicht angegeben wird, wird der interne Standardwert verwendet. Wenn ein Parameter definiert ist, darf er nicht leer sein.

Wenn das Setzen eines TLSCipher*-Werts in der Kryptobibliothek fehlschlägt, dann werden der Server, Proxy oder Agent nicht gestartet und ein Fehler wird protokolliert.

Es ist wichtig zu verstehen, wann jeder Parameter anwendbar ist.

Ausgehende Verbindungen

Der einfachste Fall sind ausgehende Verbindungen:

- Für ausgehende Verbindungen mit Zertifikat - verwenden Sie TLSCipherCert13 oder TLSCipherCert
- Für ausgehende Verbindungen mit PSK - verwenden Sie TLSCipherPSK13 oder TLSCipherPSK
- Bei den Dienstprogrammen zabbix_get und zabbix_sender können die Befehlszeilen- parameter `--tls-cipher13` oder `--tls-cipher` verwendet werden (die Verschlüsselung wird eindeutig mit dem Parameter `--tls-connect` angegeben)

Eingehende Verbindungen

Bei eingehenden Verbindungen ist es etwas komplizierter, da die Regeln für Komponenten und Konfiguration spezifisch sind.

Für Zabbix **Agent**:

Agent connection setup	Cipher configuration
TLSConnect=cert	TLSCipherCert, TLSCipherCert13
TLSConnect=psk	TLSCipherPSK, TLSCipherPSK13
TLSAccept=cert	TLSCipherCert, TLSCipherCert13
TLSAccept=psk	TLSCipherPSK, TLSCipherPSK13
TLSAccept=cert,psk	TLSCipherAll, TLSCipherAll13

Für Zabbix **Server** und **Proxy**:

Connection setup	Cipher configuration
Outgoing connections using PSK	TLSCipherPSK, TLSCipherPSK13
Incoming connections using certificates	TLSCipherAll, TLSCipherAll13
Incoming connections using PSK if server has no certificate	TLSCipherPSK, TLSCipherPSK13
Incoming connections using PSK if server has certificate	TLSCipherAll, TLSCipherAll13

In den beiden obigen Tabellen ist ein Muster erkennbar:

- TLSCipherAll und TLSCipherAll13 können nur angegeben werden, wenn eine kombinierte Liste von Cipher Suites auf Basis von Zertifikaten **und** PSK verwendet wird. Dies ist in zwei Fällen der Fall: Server (Proxy) mit konfiguriertem Zertifikat (PSK-Cipher Suites sind auf dem Server immer konfiguriert, auf dem Proxy, wenn die Kryptobibliothek PSK unterstützt), Agent, der so konfiguriert ist, dass er sowohl eingehende Verbindungen auf Zertifikats- als auch auf PSK-Basis akzeptiert
- in anderen Fällen sind TLSCipherCert* und/oder TLSCipherPSK* ausreichend

Die folgenden Tabellen zeigen die integrierten Standardwerte für `TLSCipher*`. Sie können ein guter Ausgangspunkt für Ihre eigenen benutzerdefinierten Werte sein.

Parameter	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

Parameter	OpenSSL 1.1.1d [1]
TLSCipherCert13	

Parameter	OpenSSL 1.1.1d [1]
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] Die Standardwerte unterscheiden sich bei älteren OpenSSL-Versionen (1.0.1, 1.0.2, 1.1.0), bei LibreSSL und wenn OpenSSL ohne PSK-Unterstützung kompiliert wurde.

Beispiele für benutzerkonfigurierte Cipher Suites

Siehe unten die folgenden Beispiele für benutzerkonfigurierte Cipher Suites:

- Testen von Cipher-Strings und Zulassen nur von PFS Cipher Suites
- Wechsel von AES128 zu AES256

Testen von Cipher-Strings und Zulassen ausschließlich von PFS-Ciphersuites

Um zu sehen, welche Ciphersuites ausgewählt wurden, müssen Sie `DebugLevel=4` in der Konfigurationsdatei festlegen oder die Option `-vv` für zabbix_sender verwenden.

Möglicherweise müssen Sie mit den Parametern `TLSCipher*` experimentieren, bevor Sie die gewünschten Ciphersuites erhalten. Es ist umständlich, den Zabbix Server, Proxy oder Agent mehrmals neu zu starten, nur um die Parameter `TLSCipher*` anzupassen. Bequemer ist es, zabbix_sender oder den Befehl `openssl` zu verwenden. Sehen wir uns beide Möglichkeiten an.

1. Verwendung von zabbix_sender.

Erstellen wir eine Testkonfigurationsdatei, zum Beispiel `/home/zabbix/test.conf`, mit der Syntax einer `zabbix_agentd.conf`-Datei:

```
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
```

Für dieses Beispiel benötigen Sie gültige CA- und Agent-Zertifikate sowie einen PSK. Passen Sie die Pfade und Namen der Zertifikats- und PSK-Dateien an Ihre Umgebung an.

Wenn Sie keine Zertifikate, sondern ausschließlich PSK verwenden, können Sie eine einfachere Testdatei erstellen:

```
Hostname=nonexisting
ServerActive=nonexisting

TLSConnect=psk
TLSPSKIdentity=nonexisting
TLSPSKFile=/home/zabbix/agentd.psk
```

Die ausgewählten Ciphersuites können durch Ausführen von zabbix_sender angezeigt werden (Beispiel, kompiliert mit 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
```

Hier sehen Sie die standardmäßig ausgewählten Ciphersuites. Diese Standardwerte wurden gewählt, um die Interoperabilität mit Zabbix Agents sicherzustellen, die auf Systemen mit älteren OpenSSL-Versionen (ab 1.0.1) ausgeführt werden.

Auf neueren Systemen können Sie die Sicherheit erhöhen, indem Sie nur wenige Ciphersuites zulassen, zum Beispiel ausschließlich Ciphersuites mit PFS (Perfect Forward Secrecy). Konfigurieren wir die Parameter `TLSCipher*`, um nur Ciphersuites zuzulassen, die PFS unterstützen.

Attention:
Das Ergebnis ist bei Verwendung von PSK nicht mit Systemen interoperabel, die OpenSSL 1.0.1 und 1.0.2 verwenden. Die zertifikatsbasierte Verschlüsselung sollte funktionieren.

Fügen Sie der Konfigurationsdatei `test.conf` zwei Zeilen hinzu:

```
TLSCipherCert=EECDH+aRSA+AES128
TLSCipherPSK=kECDHEPSK+AES128
```

und testen Sie erneut:

```
$ 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
```

Die Listen „certificate ciphersuites“ und „PSK ciphersuites“ haben sich geändert - sie sind kürzer als zuvor und enthalten erwartungsgemäß nur TLS-1.3-Ciphersuites und TLS-1.2-ECDHE-*-Ciphersuites.

2. `TLSCipherAll` und `TLSCipherAll13` können nicht mit zabbix_sender getestet werden; sie wirken sich nicht auf den im obigen Beispiel angezeigten Wert „certificate and PSK ciphersuites“ aus. Um `TLSCipherAll` und `TLSCipherAll13` anzupassen, müssen Sie mit dem Agent, Proxy oder Server experimentieren.

Um ausschließlich PFS-Ciphersuites zuzulassen, müssen Sie daher möglicherweise bis zu drei Parameter hinzufügen:

```
TLSCipherCert=EECDH+aRSA+AES128
TLSCipherPSK=kECDHEPSK+AES128
TLSCipherAll=EECDH+aRSA+AES128:kECDHEPSK+AES128
```

zu `zabbix_agentd.conf`, `zabbix_proxy.conf` und `zabbix_server.conf`, wenn für jeden dieser Prozesse ein Zertifikat konfiguriert ist und der Agent zusätzlich PSK verwendet.

Wenn Ihre Zabbix-Umgebung ausschließlich PSK-basierte Verschlüsselung und keine Zertifikate verwendet, benötigen Sie nur einen Parameter:

```
TLSCipherPSK=kECDHEPSK+AES128
```

Nachdem Sie nun verstanden haben, wie dies funktioniert, können Sie die Auswahl der Ciphersuites auch außerhalb von Zabbix mit dem Befehl `openssl` testen. Testen wir alle drei Werte der Parameter `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
```

Möglicherweise bevorzugen Sie `openssl ciphers` mit der Option `-V`, um eine ausführlichere Ausgabe zu erhalten:

```
$ 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
```

Ebenso können Sie die Prioritätszeichenfolgen für GnuTLS testen:

```
$ 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
```

Wechsel von AES128 zu AES256

Zabbix verwendet AES128 standardmäßig als integrierte Voreinstellung für Daten. Nehmen wir an, Sie verwenden Zertifikate und möchten auf AES256 umstellen, unter OpenSSL 1.1.1.

Dies kann erreicht werden, indem die entsprechenden Parameter in `zabbix_server.conf` hinzugefügt werden:

```
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:
Obwohl nur zertifikatsbezogene Cipher Suites verwendet werden, werden auch die Parameter `TLSCipherPSK*` definiert, um ihre Standardwerte zu vermeiden, die weniger sichere Cipher für eine breitere Interoperabilität enthalten. PSK-Cipher Suites können auf Server/Proxy nicht vollständig deaktiviert werden.

Und in `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
```

Fehlerbehebung bei Verschlüsselungsproblemen Allgemeine Empfehlungen:

- Beginnen Sie damit, zu verstehen, welche Komponente im Problemfall als TLS-Client und welche als TLS-Server fungiert.
 Zabbix Server, Proxies und Agents können je nach Interaktion untereinander alle sowohl als TLS-Server als auch als TLS-Clients arbeiten.
 Beispielsweise fungiert der Zabbix Server beim Verbinden mit einem Agent für eine passive Prüfung als TLS-Client. Der Agent übernimmt dabei die Rolle des TLS-Servers.
 Der Zabbix Agent, der von einem Proxy eine Liste aktiver Prüfungen anfordert, fungiert als TLS-Client. Der Proxy übernimmt dabei die Rolle des TLS-Servers.
 Die Dienstprogramme `zabbix_get` und `zabbix_sender` fungieren immer als TLS-Clients.
- Zabbix verwendet gegenseitige Authentifizierung.
 Jede Seite überprüft ihren Gegenpart und kann die Verbindung ablehnen.
 Beispielsweise kann der Zabbix Server beim Verbinden mit einem Agent die Verbindung sofort schließen, wenn das Zertifikat des Agents ungültig ist. Umgekehrt kann der Zabbix Agent, der eine Verbindung vom Server annimmt, die Verbindung schließen, wenn der Server vom Agent nicht als vertrauenswürdig eingestuft wird.
- Prüfen Sie die Protokolldateien auf beiden Seiten - beim TLS-Client und beim TLS-Server.
 Die Seite, die die Verbindung ablehnt, kann den genauen Grund dafür protokollieren. Die andere Seite meldet oft nur einen allgemeinen Fehler (z. B. ”Connection closed by peer”, ”connection was non-properly terminated”).
- Manchmal führt eine falsch konfigurierte Verschlüsselung zu verwirrenden Fehlermeldungen, die in keiner Weise auf die eigentliche Ursache hinweisen.
 In den folgenden Unterabschnitten versuchen wir, eine (bei weitem nicht vollständige) Sammlung von Meldungen und möglichen Ursachen bereitzustellen, die bei der Fehlerbehebung helfen können.
 Bitte beachten Sie, dass unterschiedliche Kryptobibliotheken (OpenSSL, GnuTLS) in denselben Problemsituationen oft unterschiedliche Fehlermeldungen erzeugen.
 Manchmal hängen die Fehlermeldungen sogar von der jeweiligen Kombination der Kryptobibliotheken auf beiden Seiten ab.

Fehlerbehebung bei Verbindungsart- oder Berechtigungsproblemen Server ist so konfiguriert, dass es sich mit PSK mit dem Agent verbindet, der Agent akzeptiert jedoch nur unverschlüsselte Verbindungen

Im Server- oder Proxy-Log (mit *GnuTLS* 3.3.16)

```
Wert vom Agent konnte nicht abgerufen werden: zbx_tls_connect(): gnutls_handshake() failed: \
    -110 Die TLS-Verbindung wurde nicht ordnungsgemäß beendet.
```

Im Server- oder Proxy-Log (mit *OpenSSL* 1.0.2c)

```
Wert vom Agent konnte nicht abgerufen werden: TCP-Verbindung erfolgreich, TLS zu [[127.0.0.1]:10050] kann
    Verbindung vom Peer geschlossen. Prüfen Sie die zulässigen Verbindungstypen und Zugriffsrechte
```

Eine Seite verbindet sich mit Zertifikat, die andere Seite akzeptiert jedoch nur PSK oder umgekehrt

In einem beliebigen Log (mit *GnuTLS*):

```
eingehende Verbindung konnte nicht akzeptiert werden: von 127.0.0.1: zbx_tls_accept(): gnutls_handshake()
    -21 Eine unterstützte Cipher Suite konnte nicht ausgehandelt werden.
```

In einem beliebigen Log (mit *OpenSSL* 1.0.2c):

```
eingehende Verbindung konnte nicht akzeptiert werden: von 127.0.0.1: TLS-Handshake gab Fehlercode 1 zurück
    Datei .\ssl\s3_srvr.c Zeile 1411: error:1408A0C1:SSL routines:ssl3_get_client_hello:no shared cipher:\
    TLS write fatal alert "handshake failure"
```

Versuch, Zabbix sender mit TLS-Unterstützung zu verwenden, um Daten an einen ohne TLS kompilierten Zabbix Server/Proxy zu senden Im Log auf der Verbindungsseite:

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 pee
```

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 error: [0x00000000] The operation completed succ
...send value error: TCP successful, cannot establish TLS to [[192.168.1.2]:10051]: SSL_connect() I/O erro
```

Im Log auf der empfangenden Seite:

```
...failed to accept an incoming connection: from 127.0.0.1: support for TLS was not compiled in
```

Eine Seite verbindet sich mit PSK, die andere Seite verwendet LibreSSL oder wurde ohne Verschlüsselungsunterstützung kompiliert LibreSSL unterstützt PSK nicht.

Im Log der verbindenden Seite:

```
...TCP erfolgreich, TLS zu [[192.168.1.2]:10050] kann nicht aufgebaut werden: SSL_connect() I/O error: [0]
```

Im Log der annehmenden Seite:

```
...eingehende Verbindung konnte nicht angenommen werden: von 192.168.1.2: Unterstützung für PSK wurde nich
```

Im Zabbix Frontend:

```
Wert von Agent konnte nicht abgerufen werden: TCP erfolgreich, TLS zu [[192.168.1.2]:10050] kann nicht auf
```

Eine Seite verbindet sich mit PSK, die andere Seite verwendet OpenSSL mit deaktivierter PSK-Unterstützung Im Log der verbindenden Seite:

```
...TCP erfolgreich, TLS kann zu [[192.168.1.2]:10050] nicht aufgebaut werden: SSL_connect() setzte den Erg
```

Im Log der annehmenden Seite:

```
...eingehende Verbindung konnte nicht angenommen werden: von 192.168.1.2: TLS handshake setzte den Ergebni
```

Verschlüsselung mit Zertifikaten

Übersicht Zabbix kann RSA-Zertifikate im PEM-Format verwenden, die von einer öffentlichen oder einer internen Zertifizierungsstelle (CA) signiert wurden.

Die Zertifikatsprüfung erfolgt anhand eines vorkonfigurierten CA-Zertifikats. Optional können Certificate Revocation Lists (CRL) verwendet werden.

Jede Zabbix-Komponente kann nur ein Zertifikat konfiguriert haben.

Weitere Informationen zum Einrichten und Betreiben einer internen CA, zum Erzeugen und Signieren von Zertifikatsanforderungen sowie zum Widerrufen von Zertifikaten finden Sie in Tutorials wie dem OpenSSL PKI Tutorial v2.0.

Prüfen und testen Sie Ihre Zertifikatserweiterungen sorgfältig. Weitere Details finden Sie unter Einschränkungen bei der Verwendung von X.509 v3-Zertifikatserweiterungen.

Zertifikatskonfigurationsparameter Die folgenden Konfigurationsparameter werden für die Einrichtung von Zertifikaten auf Zabbix-Komponenten unterstützt.

Parameter	Mandatory	Description
TLSCAFile	yes	Vollständiger Pfadname einer Datei, die die Zertifikate der obersten CA(s) für die Überprüfung von Peer-Zertifikaten enthält. Wenn eine Zertifikatskette mit mehreren Mitgliedern verwendet wird, ordnen Sie die Zertifikate so an, dass die Zertifikate der unteren CA(s) zuerst und anschließend die Zertifikate der höheren CA(s) folgen. Zertifikate mehrerer CAs können in einer einzelnen Datei enthalten sein.
TLSCRLFile	no	Vollständiger Pfadname einer Datei, die Certificate Revocation Lists (CRL) enthält.
TLSCertFile	yes	Vollständiger Pfadname einer Datei, die das Zertifikat enthält. Wenn eine Zertifikatskette mit mehreren Mitgliedern verwendet wird, ordnen Sie die Zertifikate so an, dass zuerst das Zertifikat des Servers, Proxys oder Agents folgt, danach die Zertifikate der unteren CA(s) und abschließend die Zertifikate der höheren CA(s).
TLSKeyFile	yes	Vollständiger Pfadname einer Datei, die den privaten Schlüssel enthält. Stellen Sie sicher, dass diese Datei nur für den Zabbix-Benutzer lesbar ist, indem Sie geeignete Zugriffsrechte festlegen.
TLSServerCertIssuer	no	Zulässiger Aussteller des Serverzertifikats.
TLSServerCertSubject	no	Zulässiger Betreff des Serverzertifikats.

Konfigurationsbeispiele Nachdem die erforderlichen Zertifikate eingerichtet wurden, konfigurieren Sie die Zabbix-Komponenten so, dass sie eine zertifikatsbasierte Verschlüsselung verwenden.

Nachfolgend finden Sie detaillierte Schritte zur Konfiguration von:

- Zabbix Server
- Zabbix Proxy
- Zabbix Agent

Zabbix Server

1. Bereiten Sie die CA-Zertifikatsdatei vor.

Um Peer-Zertifikate zu überprüfen, muss der Zabbix Server Zugriff auf die Datei haben, die die selbstsignierten Root-CA-Zertifikate der obersten Ebene enthält. Wenn beispielsweise Zertifikate von zwei unabhängigen Root-CAs benötigt werden, legen Sie diese in einer Datei unter `/home/zabbix/zabbix_ca_file.crt` ab:

```
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. Legen Sie das Zabbix Server-Zertifikat bzw. die Zertifikatskette in einer Datei ab, beispielsweise unter `/home/zabbix/zabbix_server.cr`
Das erste Zertifikat ist das Zabbix Server-Zertifikat, gefolgt vom Zwischenzertifikat der CA:

```
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:
Verwenden Sie für Client- und Server-Zertifikate ausschließlich die oben genannten Attribute, um eine Beeinträchtigung des Zertifikatsüberprüfungsprozesses zu vermeiden. OpenSSL kann beispielsweise möglicherweise keine verschlüsselte Verbindung herstellen, wenn die Erweiterungen *X509v3 Subject Alternative Name* oder *Netscape Cert Type* verwendet werden. Weitere Informationen finden Sie unter Einschränkungen bei der Verwendung von X.509-v3-Zertifikatserweiterungen.

3. Legen Sie den privaten Schlüssel des Zabbix Servers in einer Datei ab, beispielsweise unter `/home/zabbix/zabbix_server.key`:

```
-----BEGIN PRIVATE KEY-----
MIIEwAIBADANBgkqhkiG9w0BAQEFAASCBKowggSmAgEAAoIBAQC9tIXIJoVnNXDl
...
IJLkhbybBYEf47MLhffWa7XvZTY=
-----END PRIVATE KEY-----
```

4. Bearbeiten Sie die TLS-Konfigurationsparameter in der Konfigurationsdatei des Zabbix Servers:

```
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSCertFile=/home/zabbix/zabbix_server.crt
TLSKeyFile=/home/zabbix/zabbix_server.key
```

Zabbix Proxy

1. Bereiten Sie Dateien mit den Zertifikaten der übergeordneten Zertifizierungsstellen, dem Zabbix-Proxy-Zertifikat bzw. der Zertifikatskette und dem privaten Schlüssel vor, wie im Abschnitt Zabbix-Server beschrieben. Bearbeiten Sie anschließend die Parameter `TLSCAFile`, `TLSCertFile` und `TLSKeyFile` in der Proxy-Konfigurationsdatei von Zabbix entsprechend.

2. Bearbeiten Sie zusätzliche TLS-Parameter in der Proxy-Konfigurationsdatei von Zabbix:

- Für einen aktiven Proxy: `TLSConnect=cert`
- Für einen passiven Proxy: `TLSAccept=cert`

Note:
Um die Sicherheit des Proxy zu erhöhen, können Sie auch die Parameter `TLSServerCertIssuer` und `TLSServerCertSubject` festlegen. Weitere Informationen finden Sie unter Zulässigen Zertifikatsaussteller und -betreff einschränken.

Die TLS-Parameter in der fertigen Proxy-Konfigurationsdatei können wie folgt aussehen:

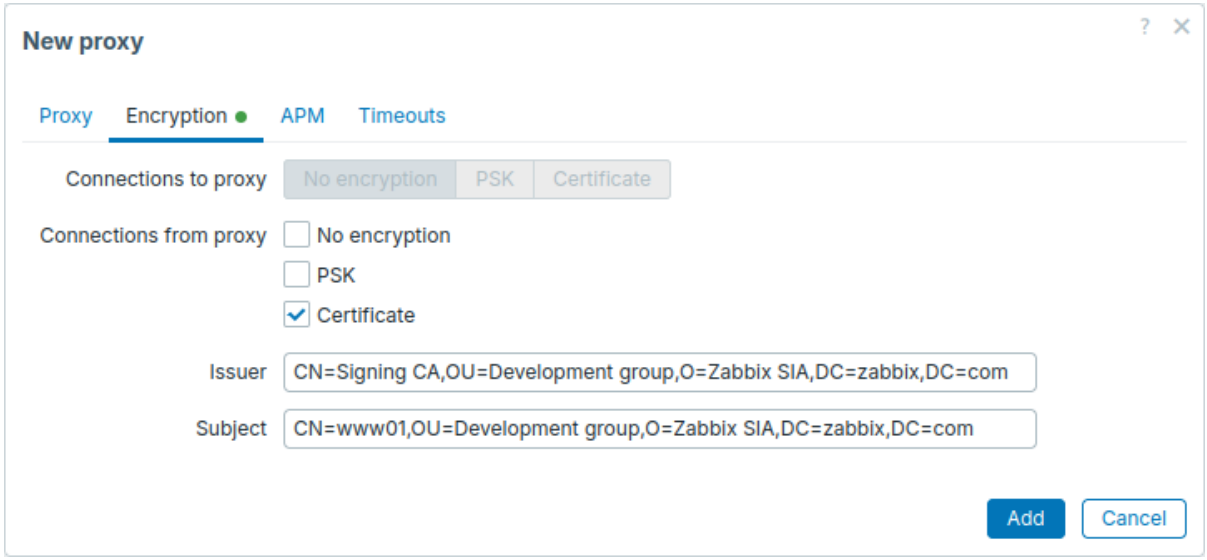
```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix server,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSCertFile=/home/zabbix/zabbix_proxy.crt
TLSKeyFile=/home/zabbix/zabbix_proxy.key
```

3. Konfigurieren Sie die Verschlüsselung für diesen Proxy im Zabbix Frontend:

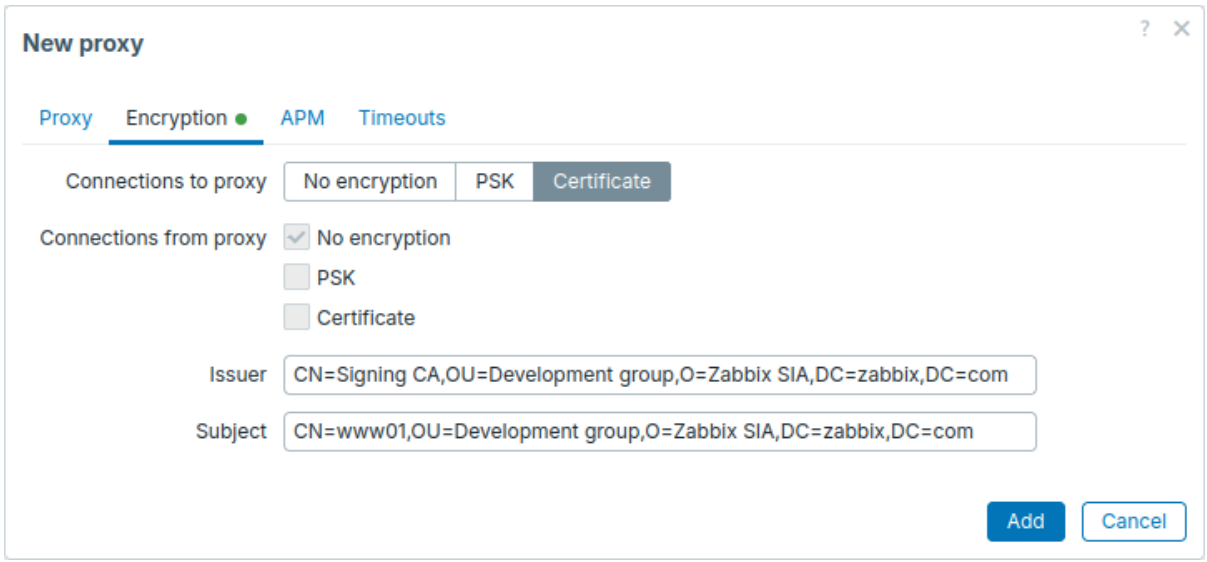
- Gehen Sie zu: *Administration* > *Proxies*.
- Wählen Sie den Proxy aus und klicken Sie auf die Registerkarte *Encryption*.

In den folgenden Beispielen sind die Felder *Issuer* und *Subject* ausgefüllt. Weitere Informationen dazu, warum und wie diese Felder verwendet werden, finden Sie unter Zulässigen Zertifikatsaussteller und -betreff einschränken.

Für einen aktiven Proxy:



Für einen passiven Proxy:



Zabbix Agent

1. Bereiten Sie Dateien mit den übergeordneten CA-Zertifikaten, dem Zabbix Agent-Zertifikat bzw. der Zertifikatskette und dem privaten Schlüssel vor, wie im Abschnitt Zabbix Server beschrieben. Bearbeiten Sie anschließend die Parameter `TLSCAFile`, `TLSCertFile` und `TLSKeyFile` in der Konfigurationsdatei des Zabbix Agents entsprechend.

2. Bearbeiten Sie zusätzliche TLS-Parameter in der Konfigurationsdatei des Zabbix Agents:

- Für einen aktiven Agent: `TLSConnect=cert`
- Für einen passiven Agent: `TLSAccept=cert`

Note:
Um die Sicherheit des Agents zu erhöhen, können Sie die Parameter `TLSServerCertIssuer` und `TLSServerCertSubject` festlegen. Weitere Informationen finden Sie unter Zulässigen Zertifikatsaussteller und -betreff einschränken.

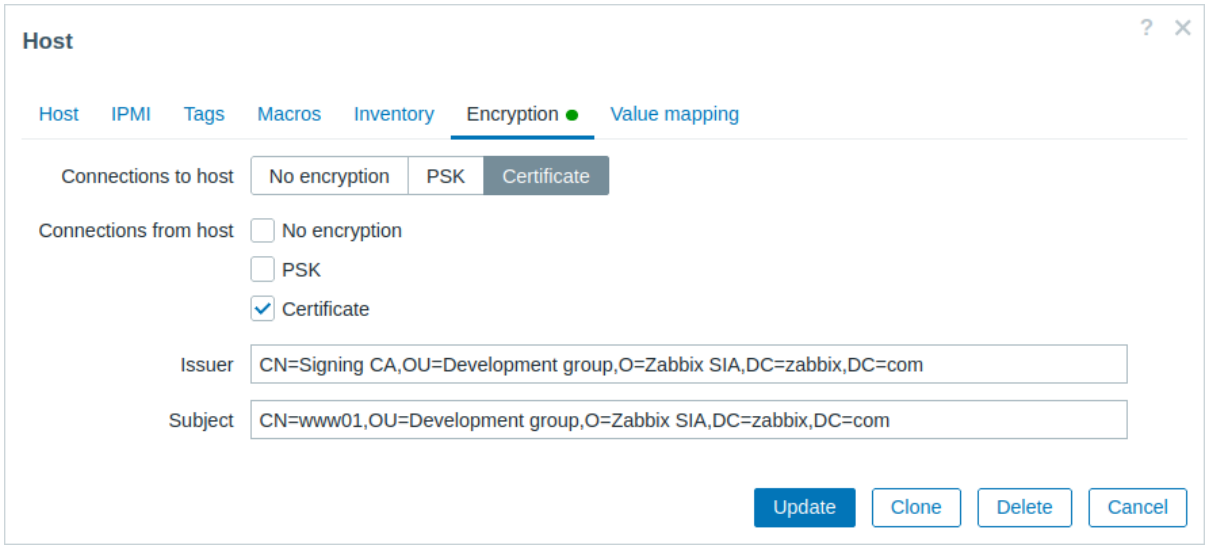
Die TLS-Parameter in der endgültigen Konfigurationsdatei des Agents können wie folgt aussehen. Beachten Sie, dass das Beispiel davon ausgeht, dass der Host von einem Proxy überwacht wird. Daher wird dieser als Zertifikats-Subject angegeben:

```
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. Konfigurieren Sie die Verschlüsselung im Zabbix Frontend für den von diesem Agent überwachten Host.

- Gehen Sie zu: *Datensammlung* > *Hosts*.
- Wählen Sie den Host aus und klicken Sie auf die Registerkarte *Verschlüsselung*.

Im folgenden Beispiel sind die Felder *Issuer* und *Subject* ausgefüllt. Weitere Informationen dazu, warum und wie diese Felder verwendet werden, finden Sie unter Zulässigen Zertifikatsaussteller und -betreff einschränken.



Zabbix-Webdienst

1. Bereiten Sie Dateien mit den übergeordneten CA-Zertifikaten, dem Zertifikat bzw. der Zertifikatskette des Zabbix-Webdienstes und dem privaten Schlüssel vor, wie im Abschnitt Zabbix-Server beschrieben. Bearbeiten Sie anschließend die Parameter `TLSCAFile`, `TLSCertFile` und `TLSKeyFile` in der Konfigurationsdatei des Webdienstes von Zabbix entsprechend.

2. Bearbeiten Sie einen zusätzlichen TLS-Parameter in der Konfigurationsdatei des Webdienstes von Zabbix: `TLSAccept=cert`

Die TLS-Parameter in der endgültigen Konfigurationsdatei des Webdienstes können wie folgt aussehen:

```
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSCertFile=/home/zabbix/zabbix_web_service.crt
TLSKeyFile=/home/zabbix/zabbix_web_service.key
```

3. Konfigurieren Sie den Zabbix-Server so, dass er eine Verbindung zum TLS-konfigurierten Zabbix-Webdienst herstellt, indem Sie den Parameter `WebServiceURL` in der Server-Konfigurationsdatei von Zabbix bearbeiten:

```
WebServiceURL=https://example.com:443/report
```

Einschränken des zulässigen Zertifikatsausstellers und -subjekts Wenn zwei Zabbix-Komponenten (z. B. Server und Agent) eine TLS-Verbindung herstellen, überprüfen sie gegenseitig ihre Zertifikate. Wenn ein Peer-Zertifikat von einer vertrauenswürdigen CA signiert wurde (mit einem in `TLSCAFile` vorkonfigurierten Zertifikat der obersten Ebene), gültig ist, nicht abgelaufen ist und weitere Prüfungen besteht, kann die Kommunikation zwischen den Komponenten fortgesetzt werden. In diesem einfachsten Fall werden der Zertifikatsaussteller und das Zertifikatssubjekt nicht überprüft.

Dies stellt jedoch ein Risiko dar: Jeder mit einem gültigen Zertifikat kann sich als jemand anderes ausgeben (beispielsweise könnte ein Host-Zertifikat verwendet werden, um sich als Server auszugeben). Dies kann in kleinen Umgebungen akzeptabel sein, in denen Zertifikate von einer dedizierten internen CA signiert werden und das Risiko einer Identitätsvortäuschung gering ist. In größeren oder sicherheitskritischeren Umgebungen ist dies jedoch möglicherweise nicht ausreichend.

Wenn Ihre CA der obersten Ebene Zertifikate ausstellt, die von Zabbix nicht akzeptiert werden sollen, oder wenn Sie das Risiko einer Identitätsvortäuschung reduzieren möchten, können Sie zulässige Zertifikate einschränken, indem Sie deren Aussteller und Subjekt angeben.

Beispielsweise könnten Sie in der Proxy-Konfigurationsdatei von Zabbix Folgendes angeben:

```
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
```

Mit diesen Einstellungen kommuniziert ein aktiver Proxy nicht mit einem Zabbix-Server, dessen Zertifikat einen anderen Aussteller oder ein anderes Subjekt aufweist. Ebenso akzeptiert ein passiver Proxy keine Anfragen von einem solchen Server.

Regeln für den Abgleich von `Issuer`- und `Subject`-Strings

Die Regeln für den Abgleich von `Issuer`- und `Subject`-Strings lauten wie folgt:

- `Issuer`- und `Subject`-Strings werden unabhängig voneinander geprüft. Beide sind optional.
- Ein nicht angegebener String bedeutet, dass jeder String akzeptiert wird.
- Strings werden *wie angegeben* verglichen und müssen exakt übereinstimmen.
- UTF-8-Zeichen werden unterstützt. Platzhalter (*) oder reguläre Ausdrücke werden jedoch nicht unterstützt.
- Die folgenden Anforderungen aus RFC 4514 werden implementiert - Zeichen, die ein Escaping erfordern (mit einem '\'-Backslash, U+005C):
  - überall im String: '"' (U+0022), '+' (U+002B), ',' (U+002C), ';' (U+003B), '<' (U+003C), '>' (U+003E), '\\' (U+005C);
  - am Anfang des Strings: Leerzeichen (' ', U+0020) oder Nummernzeichen ('#', U+0023);
  - am Ende des Strings: Leerzeichen (' ', U+0020).
- Nullzeichen (U+0000) werden nicht unterstützt. Wenn ein Nullzeichen gefunden wird, schlägt der Abgleich fehl.
- Die Standards RFC 4517 und RFC 4518 werden nicht unterstützt.

Wenn beispielsweise die Organisations- (O) Strings von `Issuer` und `Subject` nachgestellte Leerzeichen enthalten und der organisatorische Einheit- (OU) String von `Subject` doppelte Anführungszeichen enthält, müssen diese Zeichen maskiert werden:

```
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
```

Feldreihenfolge und Formatierung

Zabbix folgt den Empfehlungen von RFC 4514, der für diese Felder eine "umgekehrte" Reihenfolge festlegt: beginnend mit den Feldern der niedrigsten Ebene (CN), über die Felder der mittleren Ebene (OU, O) bis hin zu den Feldern der höchsten Ebene (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
```

Im Gegensatz dazu zeigt OpenSSL die Zeichenfolgen `Issuer` und `Subject` standardmäßig in der Reihenfolge von der höchsten zur niedrigsten Ebene an. Im folgenden Beispiel beginnen die Felder `Issuer` und `Subject` mit der höchsten Ebene (DC) und enden mit dem Feld der niedrigsten Ebene (CN). Auch die Formatierung mit Leerzeichen und Feldtrennzeichen variiert je nach verwendeten Optionen und entspricht daher nicht dem von Zabbix geforderten Format.

```
$ 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
```

Um die Zeichenfolgen *Issuer* und *Subject* korrekt für Zabbix zu formatieren, rufen Sie OpenSSL mit den folgenden Optionen auf:

```
$ 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
```

Die Ausgabe erfolgt dann in umgekehrter Reihenfolge, durch Kommas getrennt und kann in Zabbix-Konfigurationsdateien und im Frontend verwendet werden:

```
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
```

Einschränkungen bei der Verwendung von X.509 v3-Zertifikatserweiterungen Bei der Implementierung von X.509 v3-Zertifikaten in Zabbix werden bestimmte Erweiterungen möglicherweise nicht vollständig unterstützt oder können zu inkonsistentem Verhalten führen.

Subject Alternative Name-Erweiterung

Zabbix unterstützt die *Subject Alternative Name*-Erweiterung nicht, die verwendet wird, um alternative DNS-Namen wie IP-Adressen oder E-Mail-Adressen anzugeben. Zabbix kann nur den Wert im Feld *Subject* des Zertifikats validieren (siehe Einschränkung der zulässigen Zertifikatsaussteller und des Subjects). Wenn Zertifikate das Feld `subjectAltName` enthalten, kann das Ergebnis der Zertifikatsvalidierung je nach den spezifischen Kryptobibliotheken variieren, die zum Kompilieren der Zabbix-Komponenten verwendet wurden. Infolgedessen kann Zabbix Zertifikate je nach dieser Kombination entweder akzeptieren oder ablehnen.

Extended Key Usage-Erweiterung

Zabbix unterstützt die *Extended Key Usage*-Erweiterung. Wenn sie jedoch verwendet wird, ist es im Allgemeinen erforderlich, dass sowohl die Attribute *clientAuth* (für TLS WWW-Clientauthentifizierung) als auch *serverAuth* (für TLS WWW-Serverauthentifizierung) angegeben sind. Beispiel:

- Bei passiven Prüfungen, bei denen der Zabbix Agent als TLS-Server arbeitet, muss das Attribut *serverAuth* im Zertifikat des Agent enthalten sein.
- Bei aktiven Prüfungen, bei denen der Agent als TLS-Client arbeitet, muss das Attribut *clientAuth* im Zertifikat des Agent enthalten sein.

Während GnuTLS möglicherweise eine Warnung bei Verstößen gegen die Schlüsselverwendung ausgibt, erlaubt es in der Regel dennoch die Fortsetzung der Kommunikation trotz dieser Warnungen.

Name Constraints-Erweiterung

Die Unterstützung für die *Name Constraints*-Erweiterung variiert zwischen den Kryptobibliotheken. Stellen Sie sicher, dass die von Ihnen gewählte Bibliothek diese Erweiterung unterstützt. Diese Erweiterung kann Zabbix je nach verwendeter Bibliothek daran hindern, CA-Zertifikate zu laden, wenn dieser Abschnitt als kritisch markiert ist.

Certificate Revocation Lists (CRL) Wenn ein Zertifikat kompromittiert wurde, kann die Certificate Authority (CA) es widerrufen, indem sie das Zertifikat in eine Certificate Revocation List (CRL) aufnimmt. CRLs werden über Konfigurationsdateien verwaltet und können mithilfe des Parameters `TLSCRLFile` in den Konfigurationsdateien von Server, Proxy und Agent angegeben werden. Zum Beispiel:

```
TLSCRLFile=/home/zabbix/zabbix_crl_file.crt
```

In diesem Fall kann `zabbix_crl_file.crt` CRLs mehrerer CAs enthalten und könnte folgendermaßen aussehen:

```
-----BEGIN X509 CRL-----
MIIB/DCB5QIBATANBgkqhkiG9w0BAQUFADCBgTETMBEGCgmSJomT8ixkARkWA2Nv
...
treZeUPjb7LSmZ3K2hpbZN7SoOZcAoHQ3GWd9npuctg=
-----END X509 CRL-----
-----BEGIN X509 CRL-----
MIIB+TCB4gIBATANBgkqhkiG9w0BAQUFADB/MRMwEQYKCZImiZPyLGQBGRYDY29t
...
CAEebS2CND3ShBedZ8YSil59O6JvaDP61lR5lNs=
-----END X509 CRL-----
```

Die CRL-Datei wird nur beim Start von Zabbix geladen. Um die CRL zu aktualisieren, starten Sie Zabbix neu.

> **Attention:**
> Wenn Zabbix-Komponenten mit OpenSSL kompiliert wurden und CRLs verwendet werden, stellen Sie sicher, dass jede Root- und Zwischen-CA in den Zertifikatsketten über eine entsprechende CRL verfügt (auch wenn sie leer ist), die in `TLSCRLFile` enthalten ist.

Behebung von Zertifikatsproblemen OpenSSL wird mit CRLs verwendet, und für einige CA in der Zertifikatskette ist deren CRL nicht in `TLSCRLFile` enthalten

Im TLS-Serverprotokoll im Fall eines *OpenSSL*-Peers:

```
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"
```

Im TLS-Serverprotokoll im Fall eines *GnuTLS*-Peers:

```
failed to accept an incoming connection: from 127.0.0.1: TLS handshake with 127.0.0.1 returned error code
    file rsa_pk1.c line 103: error:0407006A: rsa routines:RSA_padding_check_PKCS1_type_1:\
    block type is not 01 file rsa_eay.c line 705: error:04067072: rsa routines:RSA_EAY_PUBLIC_DECRYPT:padd
```

CRL abgelaufen oder läuft während des Serverbetriebs ab

OpenSSL, im Server-Log:

- vor dem Ablauf:

```
kann keine Verbindung zu Proxy "proxy-openssl-1.0.1e" herstellen: TCP erfolgreich, TLS zu [[127.0.0.1]:200
    SSL_connect() gab SSL_ERROR_SSL zurück: Datei s3_clnt.c Zeile 1253: Fehler:14090086:\
    SSL routines:ssl3_get_server_certificate:certificate verify failed:\
    TLS write fatal alert "certificate revoked"
```

- nach dem Ablauf:

```
kann keine Verbindung zu Proxy "proxy-openssl-1.0.1e" herstellen: TCP erfolgreich, TLS zu [[127.0.0.1]:200
    SSL_connect() gab SSL_ERROR_SSL zurück: Datei s3_clnt.c Zeile 1253: Fehler:14090086:\
    SSL routines:ssl3_get_server_certificate:certificate verify failed:\
    TLS write fatal alert "certificate expired"
```

Der Punkt hier ist, dass bei gültiger CRL ein widerrufenes Zertifikat als ”certificate revoked” gemeldet wird. Wenn die CRL abläuft, ändert sich die Fehlermeldung zu ”certificate expired”, was ziemlich irreführend ist.

GnuTLS, im Server-Log:

- vor und nach dem Ablauf gleich:

```
kann keine Verbindung zu Proxy "proxy-openssl-1.0.1e" herstellen: TCP erfolgreich, TLS zu [[127.0.0.1]:200
      ungültiges Peer-Zertifikat: Das Zertifikat ist NICHT vertrauenswürdig. Die Zertifikatskette ist wide
```

Selbstsigniertes Zertifikat, unbekannte CA

OpenSSL, im Log:

```
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"'
```

Dies wurde beobachtet, als das Serverzertifikat versehentlich denselben Issuer- und Subject-String hatte, obwohl es von einer CA signiert war. Issuer und Subject sind im CA-Zertifikat der obersten Ebene gleich, dürfen jedoch im Serverzertifikat nicht gleich sein. (Dasselbe gilt für Proxy- und Agent-Zertifikate.)

Um zu prüfen, ob ein Zertifikat dieselben Issuer- und Subject-Einträge enthält, führen Sie Folgendes aus:

```
openssl x509 -in <yourcertificate.crt> -noout -text
```

Für das Root-Zertifikat (Zertifikat der obersten Ebene) ist es zulässig, dass Issuer und Subject identische Werte haben.

Verschlüsselung mit vorab gemeinsam genutzten Schlüsseln

Übersicht Jeder Pre-Shared Key (PSK) in Zabbix besteht tatsächlich aus einem Paar:

- nicht geheimer PSK-Identitätszeichenfolge,
- geheimer PSK-Zeichenfolge.

Die PSK-Identitätszeichenfolge ist eine nicht leere UTF-8-Zeichenfolge. Zum Beispiel „PSK ID 001 Zabbix agentd“. Sie ist ein eindeutiger Name, unter dem dieser spezifische PSK von den Zabbix-Komponenten referenziert wird. Fügen Sie keine vertraulichen Informationen in die PSK-Identitätszeichenfolge ein – sie wird unverschlüsselt über das Netzwerk übertragen.

Der PSK-Wert ist eine schwer zu erratende Zeichenfolge aus hexadezimalen Ziffern, zum Beispiel e560cb0d918d26d31b4f642181f5f570ad8

> **Warning:**
> Verwenden Sie niemals die in der Dokumentation angegebenen Beispiel-PSK-Werte in Produktionsumgebungen. Generieren Sie stattdessen einen eindeutigen, kryptografisch sicheren PSK.

Größenbeschränkungen Für die PSK-Identität und den PSK-Wert gibt es in Zabbix Größenbeschränkungen. In einigen Fällen kann eine Kryptobibliothek eine niedrigere Grenze haben:

Komponente	Maximale Größe der PSK-Identität	Minimale Größe des PSK-Werts	Maximale Größe des PSK-Werts
Zabbix	128 UTF-8-Zeichen	128 Bit (16-Byte-PSK, eingegeben als 32 Hexadezimalziffern)	2048 Bit (256-Byte-PSK, eingegeben als 512 Hexadezimalziffern)
GnuTLS	128 Byte (kann UTF-8-Zeichen enthalten)	-	2048 Bit (256-Byte-PSK, eingegeben als 512 Hexadezimalziffern)
OpenSSL 1.0.x, 1.1.0	127 Byte (kann UTF-8-Zeichen enthalten)	-	2048 Bit (256-Byte-PSK, eingegeben als 512 Hexadezimalziffern)
OpenSSL 1.1.1	127 Byte (kann UTF-8-Zeichen enthalten)	-	512 Bit (64-Byte-PSK, eingegeben als 128 Hexadezimalziffern)
OpenSSL 1.1.1a und höher	127 Byte (kann UTF-8-Zeichen enthalten)	-	2048 Bit (256-Byte-PSK, eingegeben als 512 Hexadezimalziffern)

Attention:
Das Zabbix Frontend ermöglicht unabhängig von den verwendeten Kryptobibliotheken die Konfiguration einer bis zu 128 Zeichen langen PSK-Identitätszeichenfolge und eines bis zu 2048 Bit langen PSK.
 Wenn einige Zabbix-Komponenten niedrigere Grenzen unterstützen, liegt es in der Verantwortung des Benutzers, die PSK-Identität und den PSK-Wert mit einer für diese Komponenten zulässigen Länge zu konfigurieren.
 Das Überschreiten der Größenbeschränkungen führt zu Kommunikationsfehlern zwischen Zabbix-Komponenten.

Bevor der Zabbix Server eine Verbindung zum Agent über PSK herstellt, sucht der Server die für diesen Agent konfigurierten PSK-Identität und den PSK-Wert in der Datenbank nach (tatsächlich im Konfigurations-Cache). Beim Empfang einer Verbindung verwendet der Agent die PSK-Identität und den PSK-Wert aus seiner Konfigurationsdatei. Wenn beide Parteien dieselbe PSK-Identitätszeichenfolge und denselben PSK-Wert haben, kann die Verbindung erfolgreich hergestellt werden.

Attention:
Jede PSK darf nur mit einem Wert verknüpft sein. Es liegt in der Verantwortung des Benutzers sicherzustellen, dass es keine zwei PSKs mit derselben Identitätszeichenfolge, aber unterschiedlichen Werten gibt. Andernfalls kann es zu unvorhersehbaren Fehlern oder Unterbrechungen der Kommunikation zwischen Zabbix-Komponenten kommen, die PSKs mit dieser PSK-Identitätszeichenfolge verwenden.

PSK generieren Beispielsweise kann eine 256-Bit-PSK (32 Byte) mit den folgenden Befehlen generiert werden:

- mit *OpenSSL*:

```
$ openssl rand -hex 32
af8ced32dfe8714e548694e2d29e1a14ba6fa13f216cb35c19d0feb1084b0429
```

- mit *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
```

Beachten Sie, dass „psktool“ oben eine Datenbankdatei mit einer PSK-Identität und der zugehörigen PSK generiert. Zabbix erwartet in der PSK-Datei nur eine PSK, daher müssen die Identitätszeichenfolge und der Doppelpunkt (':') aus der Datei entfernt werden.

Konfigurieren von PSK für die Kommunikation zwischen Server und Agent (Beispiel) Schreiben Sie auf dem Host des Agents den oben generierten PSK-Wert in eine Datei, zum Beispiel `/home/zabbix/zabbix_agentd.psk`. Die Datei muss den PSK-Wert in der ersten Zeile enthalten.

Legen Sie die Zugriffsrechte für die PSK-Datei fest – sie darf nur für den Zabbix-Benutzer lesbar sein.

Bearbeiten Sie die TLS-Parameter in der Agent-Konfigurationsdatei von Zabbix (`zabbix_agentd.conf`). Legen Sie zum Beispiel Folgendes fest:

```
TLSConnect=psk
TLSAccept=psk
TLSPSKFile=/home/zabbix/zabbix_agentd.psk
TLSPSKIdentity=PSK 001
```

Der Agent wird eine Verbindung zum Server herstellen (aktive Prüfungen) und vom Server sowie von `zabbix_get` nur Verbindungen akzeptieren, die PSK verwenden. Die PSK-Identität lautet „PSK 001“.

Starten Sie den Agent neu. Nun können Sie die Verbindung mit `zabbix_get` testen, zum Beispiel:

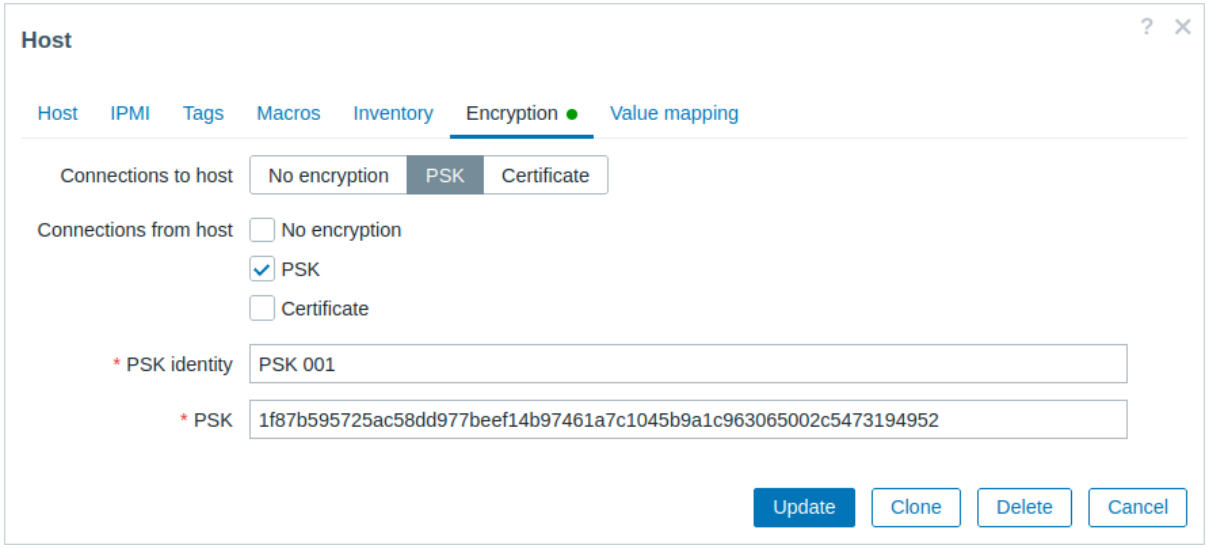
```
zabbix_get -s 127.0.0.1 -k "system.cpu.load[all,avg1]" --tls-connect=psk --tls-psk-identity="PSK 001" --tl
```

(Um Ausfallzeiten zu minimieren, lesen Sie nach, wie Sie den Verbindungstyp unter Verbindungsverschlüsselungsverwaltung ändern.)

Konfigurieren Sie die PSK-Verschlüsselung für diesen Agent im Zabbix Frontend:

1. Gehen Sie zu: *Datenerfassung* > *Hosts*.
2. Wählen Sie den Host aus und klicken Sie auf die Registerkarte *Verschlüsselung*.
3. Konfigurieren Sie die Details der PSK-Verschlüsselung.

Beispiel:



Alle Pflichtfelder sind mit einem roten Sternchen gekennzeichnet.

Wenn der Konfigurations-Cache mit der Datenbank synchronisiert wurde, verwenden neue Verbindungen PSK. Überprüfen Sie die Logdateien von Server und Agent auf Fehlermeldungen.

PSK für die Kommunikation zwischen Server und aktivem Proxy konfigurieren (Beispiel) Schreiben Sie auf dem Proxy den PSK-Wert in eine Datei, zum Beispiel `/home/zabbix/zabbix_proxy.psk`. Die Datei muss in der ersten Textzeile den PSK enthalten, zum Beispiel:

```
e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9
```

Legen Sie die Zugriffsrechte für die PSK-Datei fest - sie darf nur für den Zabbix-Benutzer lesbar sein.

Bearbeiten Sie die TLS-Parameter in der Proxy-Konfigurationsdatei von Zabbix (`zabbix_proxy.conf`). Legen Sie zum Beispiel Folgendes fest:

```
TLSConnect=psk
TLSPSKFile=/home/zabbix/zabbix_proxy.psk
TLSPSKIdentity=PSK 002
```

Der Proxy stellt über PSK eine Verbindung zum Server her. Die PSK-Identität lautet „PSK 002“.

(Informationen zum Minimieren von Ausfallzeiten finden Sie unter Verbindungsverschlüsselung verwalten.)

Konfigurieren Sie den PSK für diesen Proxy im Zabbix Frontend. Gehen Sie zu *Administration* > *Proxies*, wählen Sie den Proxy aus und öffnen Sie die Registerkarte *Encryption*. Aktivieren Sie unter „Connections from proxy“ die Option PSK. Fügen Sie in das Feld „PSK identity“ den Wert „PSK 002“ und in das Feld „PSK“ den Wert „e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba43
ein. Klicken Sie auf „Update“.

Starten Sie den Proxy neu. Er verwendet nun verschlüsselte Verbindungen auf PSK-Basis zum Server. Überprüfen Sie die Logdateien des Servers und des Proxys auf Fehlermeldungen.

Für einen passiven Proxy ist das Verfahren sehr ähnlich. Der einzige Unterschied besteht darin, dass Sie `TLSAccept=psk` in der Proxy-Konfigurationsdatei von Zabbix festlegen und im Zabbix Frontend unter *Connections to proxy* die Option PSK auswählen.

Fehlerbehebung bei PSK-Problemen PSK enthält eine ungerade Anzahl von Hex-Ziffern

Proxy oder Agent startet nicht, Meldung im Proxy- oder Agent-Log:

```
ungültige PSK in Datei "/home/zabbix/zabbix_proxy.psk"
```

PSK-Identitätszeichenfolge länger als 128 Bytes wird an GnuTLS übergeben

Im TLS-Client-Log:

```
gnutls_handshake() failed: -110 The TLS connection was non-properly terminated.
```

Im TLS-Server-Log:

```
gnutls_handshake() failed: -90 The SRP username supplied is illegal.
```

Zu langer PSK-Wert mit OpenSSL 1.1.1 verwendet

Im Log der verbindenden Seite:

```
...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
```

Im Log der akzeptierenden Seite:

```
...Message from 123.123.123.123 is missing header. Message ignored.
```

Dieses Problem tritt typischerweise auf, wenn OpenSSL von 1.0.x oder 1.1.0 auf 1.1.1 aktualisiert wird und der PSK-Wert länger als 512 Bit ist (64-Byte-PSK, eingegeben als 128 hexadezimale Ziffern).

Siehe auch: Größenlimits für Werte

Sichere Verbindung zur Datenbank

Übersicht

Dieser Abschnitt enthält Zabbix-Einrichtungsschritte und Konfigurationsbeispiele für sichere TLS-Verbindungen zwischen:

Datenbank	Zabbix-Komponenten
MySQL	Zabbix Frontend, Zabbix Server, Zabbix Proxy
PostgreSQL	Zabbix Frontend, Zabbix Server, Zabbix Proxy

Informationen zum Einrichten der Verbindungsverschlüsselung innerhalb des DBMS finden Sie in der offiziellen Dokumentation des jeweiligen Herstellers:

- MySQL: Replikationsdatenbankserver für Quelle und Replikat.
- MySQL: Datenbankserver für Gruppenreplikation usw.
- PostgreSQL Verschlüsselungsoptionen.

Alle Beispiele basieren auf den GA-Releases von MySQL CE (8.0) und PgSQL (13), die über die offiziellen Repositorys unter CentOS 8 verfügbar sind.

Anforderungen

Folgendes ist für die Einrichtung der Verschlüsselung erforderlich:

- Vom Entwickler unterstütztes Betriebssystem mit OpenSSL >=1.1.X oder einer Alternative.

> **Note:**
> Es wird empfohlen, Betriebssysteme mit End-of-Life-Status zu vermeiden, insbesondere bei Neuinstallationen

- Datenbank-Engine (RDBMS), installiert und gepflegt aus dem offiziellen Repository des Entwicklers. Betriebssysteme werden oft mit veralteten Versionen von Datenbanksoftware ausgeliefert, für die keine Unterstützung für Verschlüsselung implementiert ist, zum Beispiel auf RHEL-7-basierten Systemen und PostgreSQL 9.2, MariaDB 5.5 ohne Unterstützung für Verschlüsselung.

Terminologie

Durch das Setzen dieser Option wird die Verwendung einer TLS-Verbindung zur Datenbank vom Zabbix Server/Proxy und vom Frontend zur Datenbank erzwungen:

- `required` - Verbindung über TLS als Transportmodus ohne Identitätsprüfungen
- `verify_ca` - Verbindung über TLS und Überprüfung des Zertifikats
- `verify_full` - Verbindung über TLS, Überprüfung des Zertifikats und Überprüfung, dass die von DBHost angegebene Datenbankidentität (CN) mit ihrem Zertifikat übereinstimmt

Zabbix-Konfiguration

Frontend zur Datenbank

Eine sichere Verbindung zur Datenbank kann während der Installation des Frontends konfiguriert werden:

- Aktivieren Sie im Schritt DB-Verbindung konfigurieren das Kontrollkästchen *Datenbank-TLS-Verschlüsselung*, um die Transportverschlüsselung zu aktivieren.
- Aktivieren Sie das Kontrollkästchen *Datenbankzertifikat überprüfen*, das angezeigt wird, wenn das Feld *TLS-Verschlüsselung* aktiviert ist, um die Verschlüsselung mit Zertifikaten zu aktivieren.

Note:
Für MySQL ist das Kontrollkästchen *Datenbank-TLS-Verschlüsselung* deaktiviert, wenn *Datenbank-Host* auf localhost gesetzt ist, da eine Verbindung über eine Socket-Datei (unter Unix) oder den gemeinsamen Speicher (unter Windows) nicht verschlüsselt werden kann.
 Für PostgreSQL ist das Kontrollkästchen *TLS-Verschlüsselung* deaktiviert, wenn der Wert des Feldes *Datenbank-Host* mit einem Schrägstrich beginnt oder das Feld leer ist.

Die folgenden Parameter werden im Modus „TLS-Verschlüsselung mit Zertifikaten“ verfügbar, wenn beide Kontrollkästchen aktiviert sind:

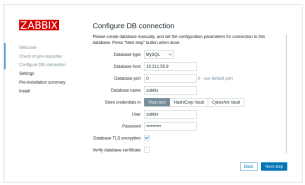
Parameter	Beschreibung
Datenbank-TLS-CA-Datei	Geben Sie den vollständigen Pfad zu einer gültigen TLS-Zertifizierungsstellendatei (CA) an.
Datenbank-TLS-Schlüsseldatei	Geben Sie den vollständigen Pfad zu einer gültigen TLS-Schlüsseldatei an.
Datenbank-TLS-Zertifikatsdatei	Geben Sie den vollständigen Pfad zu einer gültigen TLS-Zertifikatsdatei an.
Überprüfung des Datenbank-Hosts	Aktivieren Sie dieses Kontrollkästchen, um die Hostüberprüfung zu aktivieren. Für MySQL deaktiviert, da die PHP-MySQL-Bibliothek das Überspringen des Validierungsschritts für das Peer-Zertifikat nicht zulässt.
Datenbank-TLS-Chiffrenliste	Geben Sie eine benutzerdefinierte Liste gültiger Chiffren an. Das Format der Chiffrenliste muss dem OpenSSL-Standard entsprechen. Nur für MySQL verfügbar.

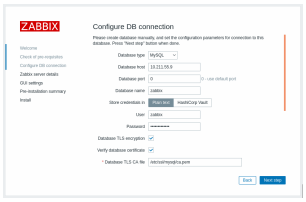
Attention:
TLS-Parameter müssen auf gültige Dateien verweisen. Wenn sie auf nicht vorhandene oder ungültige Dateien verweisen, führt dies zu einem Autorisierungsfehler.
 Wenn Zertifikatsdateien beschreibbar sind, generiert das Frontend im Bericht Systeminformationen eine Warnung: „TLS-Zertifikatsdateien müssen schreibgeschützt sein.“ (wird nur angezeigt, wenn der PHP-Benutzer Eigentümer des Zertifikats ist).
Durch Passwörter geschützte Zertifikate werden nicht unterstützt.

Anwendungsfälle

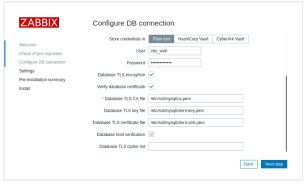
Das Zabbix Frontend verwendet eine GUI-Schnittstelle, um mögliche Optionen zu definieren: `required`, `verify_ca`, `verify_full`. Geben Sie die erforderlichen Optionen im Schritt *DB-Verbindungen konfigurieren* des Installationsassistenten an. Diese Optionen werden der Frontend-Konfigurationsdatei (`zabbix.conf.php`) folgendermaßen zugeordnet:

GUI-Einstellungen	Konfigurationsdatei	Beschreibung	Ergebnis

	...
	// Wird für die TLS-Verbindung verwendet.
	`$DB['ENCRYPTION'] = true;`
	`$DB['KEY_FILE'] = '';`
	`$DB['CERT_FILE'] = '';`
	`$DB['CA_FILE'] = '';`
	`$DB['VERIFY_HOST'] = false;`
	`$DB['CIPHER_LIST'] = '';`
	...
		1. *Datenbank-TLS-Verschlüsselung* aktivieren
		2. *Datenbankzertifikat überprüfen* nicht aktivieren
			`required`-Modus aktivieren.

	...
	`$DB['ENCRYPTION'] = true;`
	`$DB['KEY_FILE'] = '';`
	`$DB['CERT_FILE'] = '';`
	`$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';`
	`$DB['VERIFY_HOST'] = false;`
	`$DB['CIPHER_LIST'] = '';`
	...
		1. *Datenbank-TLS-Verschlüsselung* und *Datenbankzertifikat überprüfen* aktivieren
		2. Pfad zur *CA-Datei für Datenbank-TLS* angeben
			`verify_ca`-Modus aktivieren.

GUI-Einstellungen	Konfigurationsdatei	Beschreibung	Ergebnis

	...
	// Wird für die TLS-Verbindung mit einer strikt definierten Cipher-Liste verwendet.
	`$DB['ENCRYPTION'] = true;`
	`$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>';`
	...
	
	Oder:
	
	...
	// Wird für die TLS-Verbindung ohne definierte Cipher-Liste verwendet – Auswahl durch den 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. *Datenbank-TLS-Verschlüsselung* und *Datenbankzertifikat überprüfen* aktivieren
		2. Pfad zur *Schlüsseldatei für Datenbank-TLS* angeben
		3. Pfad zur *CA-Datei für Datenbank-TLS* angeben
		4. Pfad zur *Zertifikatsdatei für Datenbank-TLS* angeben
		5. *Cipher-Liste für Datenbank-TLS* angeben (optional)
			`verify_full`-Modus für MySQL aktivieren.

GUI-Einstellungen	Konfigurationsdatei	Beschreibung	Ergebnis

	...
	`$DB['ENCRYPTION'] = true;`
	`$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. *Datenbank-TLS-Verschlüsselung* und *Datenbankzertifikat überprüfen* aktivieren
		2. Pfad zur *Schlüsseldatei für Datenbank-TLS* angeben
		3. Pfad zur *CA-Datei für Datenbank-TLS* angeben
		4. Pfad zur *Zertifikatsdatei für Datenbank-TLS* angeben
		5. *Überprüfung des Datenbank-Hosts* aktivieren
			`verify_full`-Modus für PostgreSQL aktivieren.

Siehe auch: Beispiele für die Verschlüsselungskonfiguration für MySQL, Beispiele für die Verschlüsselungskonfiguration für PostgreSQL.

Konfiguration des Zabbix Servers/Proxys

Sichere Verbindungen zur Datenbank können mit den entsprechenden Parametern in der Server- und/oder Proxy-Konfigurationsdatei von Zabbix konfiguriert werden.

Konfiguration	Ergebnis
Keine `DBTLSConnect=required` festlegen	Verbindung zur Datenbank ohne Verschlüsselung. Server/Proxy stellen eine TLS-Verbindung zur Datenbank her. Eine unverschlüsselte Verbindung ist nicht zulässig.
1. `DBTLSConnect=verify_ca` festlegen 2. `DBTLSCAFile` festlegen - die Datei der TLS-Zertifizierungsstelle angeben	Server/Proxy stellen nach der Überprüfung des Datenbankzertifikats eine TLS-Verbindung zur Datenbank her.
1. `DBTLSConnect=verify_full` festlegen 2. `DBTLSCAFile` festlegen - die Datei der TLS-Zertifizierungsstelle angeben	Server/Proxy stellen nach der Überprüfung des Datenbankzertifikats und der Identität des Datenbank-Hosts eine TLS-Verbindung zur Datenbank her.
1. `DBTLSCAFile` festlegen - die Datei der TLS-Zertifizierungsstelle angeben 2. `DBTLSCertFile` festlegen - die Datei mit dem öffentlichen Client-Schlüsselzertifikat angeben 3. `DBTLSKeyFile` festlegen - die Datei mit dem privaten Client-Schlüssel angeben	Server/Proxy stellen beim Verbindungsaufbau zur Datenbank ein Clientzertifikat bereit.
`DBTLSCipher` festlegen - die Liste der Verschlüsselungschiffren, die der Client für Verbindungen mit TLS-Protokollen bis einschließlich TLS 1.2 zulässt oder `DBTLSCipher13` - die Liste der Verschlüsselungschiffren, die der Client für Verbindungen mit dem TLS-1.3-Protokoll zulässt	(MySQL) Die TLS-Verbindung wird mit einer Chiffre aus der bereitgestellten Liste hergestellt. (PostgreSQL) Das Festlegen dieser Option wird als Fehler betrachtet.

MySQL-Verschlüsselungskonfiguration

Übersicht

Dieser Abschnitt enthält mehrere Beispiele für die Verschlüsselungskonfiguration für CentOS 8.2 und MySQL 8.4.0 und kann als Schnellstartanleitung für die Verschlüsselung der Verbindung zur Datenbank verwendet werden.

Attention:
Wenn der MySQL-Host auf localhost gesetzt ist, sind Verschlüsselungsoptionen nicht verfügbar. In diesem Fall verwendet eine Verbindung zwischen Zabbix Frontend und der Datenbank eine Socket-Datei (unter Unix) oder Shared Memory (unter Windows) und kann nicht verschlüsselt werden.

Note:
Die Liste der Verschlüsselungskombinationen ist nicht auf die auf dieser Seite aufgeführten beschränkt. Es sind viele weitere Kombinationen verfügbar.

Voraussetzungen

Installieren Sie die MySQL-Datenbank aus dem offiziellen Repository.

Weitere Informationen zur Verwendung des MySQL-Repositories finden Sie in der MySQL-Dokumentation.

Der MySQL-Server ist bereit, sichere Verbindungen mit einem selbstsignierten Zertifikat zu akzeptieren.

Um zu sehen, welche Benutzer eine verschlüsselte Verbindung verwenden, führen Sie die folgende Abfrage aus (Performance Schema sollte auf ON gesetzt sein):

```
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;
```

Nur-Transport-Verschlüsselung

MySQL-Konfiguration

Moderne Versionen der Datenbank sind standardmäßig für den Verschlüsselungsmodus `required` vorbereitet. Nach der Ersteinrichtung und dem Start wird ein serverseitiges Zertifikat erstellt.

Erstellen Sie Benutzer und Rollen für die Hauptkomponenten (ersetzen Sie `<strong_password>` durch ein starkes, eindeutiges Passwort):

```
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'@'%';
```

Beachten Sie, dass das X.509-Protokoll nicht zur Überprüfung der Identität verwendet wird, sondern der Benutzer so konfiguriert ist, dass er nur verschlüsselte Verbindungen verwendet. Weitere Informationen zur Konfiguration von Benutzern finden Sie in der MySQL-Dokumentation.

Führen Sie den folgenden Befehl aus, um die Verbindung zu überprüfen (eine Socket-Verbindung kann nicht zum Testen sicherer Verbindungen verwendet werden):

```
mysql -u zbx_srv -p -h 10.211.55.9 --ssl-mode=REQUIRED
```

Überprüfen Sie den aktuellen Status und die verfügbaren Verschlüsselungssammlungen:

```
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

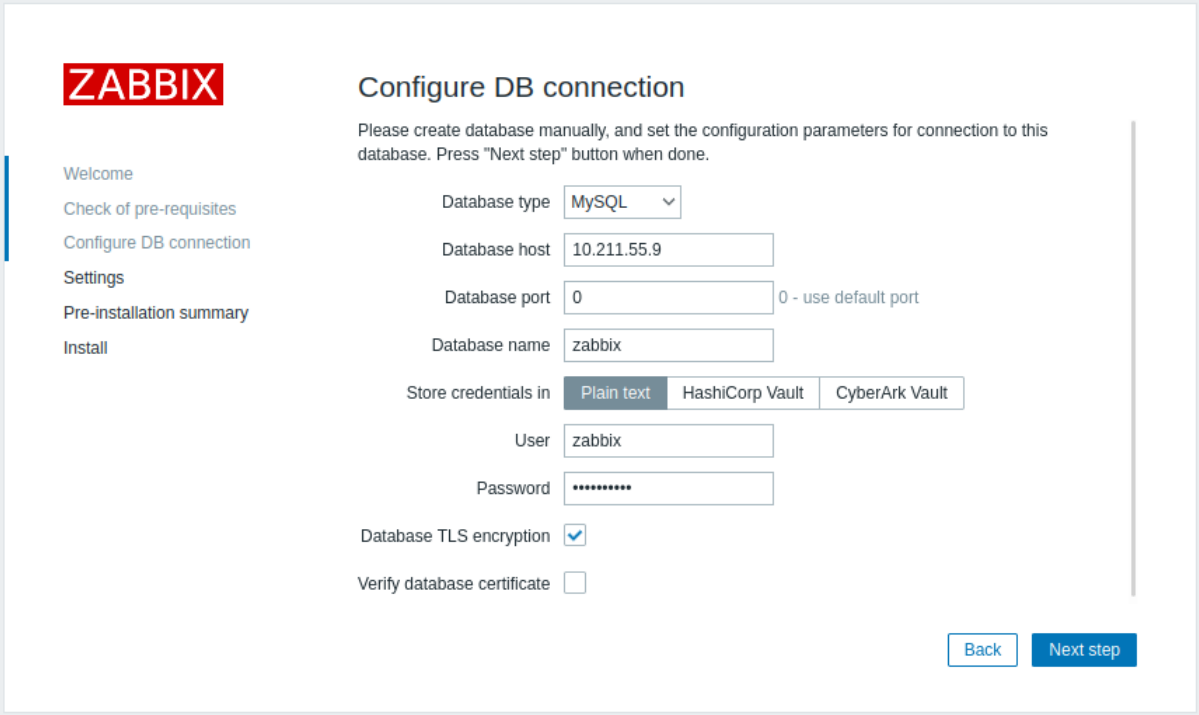

mysql> SHOW SESSION STATUS LIKE 'Ssl_cipher_list'\G;
*************************** 1. row ***************************
Variable_name: Ssl_cipher_list
Value: TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:TLS_AES_128_CCM_SHA256:E
1 row in set (0.00 sec)

ERROR:
No query specified
```

Frontend

Um nur die Transportverschlüsselung für Verbindungen zwischen dem Zabbix Frontend und der Datenbank zu aktivieren:

- Aktivieren Sie *Database TLS encryption*
- Lassen Sie *Verify database certificate* deaktiviert



Server

Um die reine Transportverschlüsselung für Verbindungen zwischen Server und Datenbank zu aktivieren, bearbeiten Sie die Server-Konfigurationsdatei:

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

Verschlüsselung mit Überprüfung der Zertifizierungsstelle

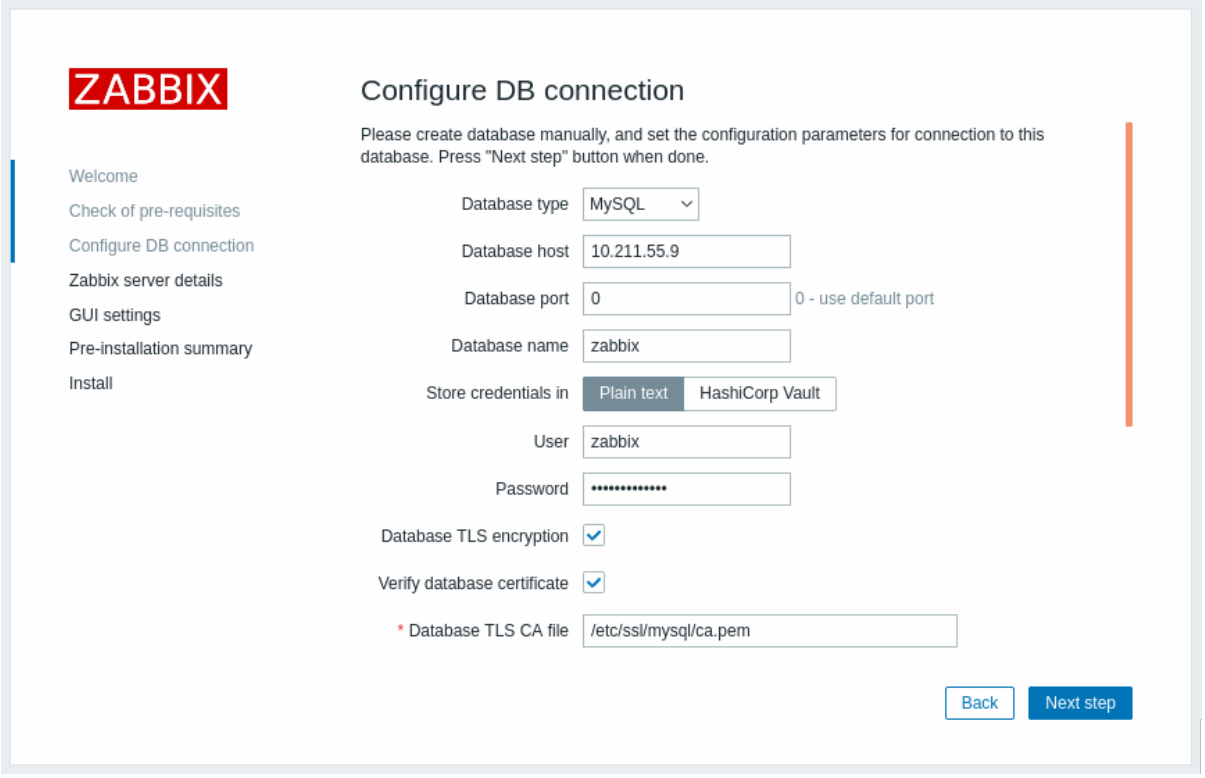
Kopieren Sie das erforderliche MySQL-CA-Zertifikat auf den Zabbix-Frontend-Server und weisen Sie die entsprechenden Berechtigungen zu, damit der Webserver diese Datei lesen kann.

Note:
Dieser Modus funktioniert unter RHEL 7 aufgrund älterer MySQL-Bibliotheken nicht.

Frontend

So aktivieren Sie die Verschlüsselung mit Zertifikatsüberprüfung für Verbindungen zwischen dem Zabbix-Frontend und der Datenbank:

- Aktivieren Sie *Datenbank-TLS-Verschlüsselung* und *Datenbankzertifikat überprüfen*
- Geben Sie den Pfad zur CA-Datei für Datenbank-TLS an



Alternativ kann dies in der Frontend-Konfigurationsdatei (`zabbix.conf.php`) festgelegt werden:

```
...
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '';
$DB['CERT_FILE'] = '';
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
$DB['VERIFY_HOST'] = false;
$DB['CIPHER_LIST'] = '';
...
```

Beheben Sie Probleme beim Benutzer, indem Sie mit dem Befehlszeilentool überprüfen, ob eine Verbindung für den erforderlichen Benutzer möglich ist:

```
    mysql -u zbx_web -p -h 10.211.55.9 --ssl-mode=REQUIRED --ssl-ca=/var/lib/mysql/ca.pem
```

Server

Um die Verschlüsselung mit Zertifikatsüberprüfung für Verbindungen zwischen dem Zabbix Server und der Datenbank zu aktivieren, bearbeiten Sie die Server-Konfigurationsdatei:

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_ca
DBTLSCAFile=/etc/ssl/mysql/ca.pem
...
```

Verschlüsselung mit vollständiger Verifizierung

MySQL-Konfiguration

Setzen Sie die Konfigurationsoption des MySQL-CE-Servers (*/etc/my.cnf.d/server-tls.cnf*) auf:

```
[mysqld]
...
#### in diesem Beispiel befinden sich die Schlüssel im MySQL-CE-Datadir-Verzeichnis
ssl_ca=ca.pem
ssl_cert=server-cert.pem
ssl_key=server-key.pem

require_secure_transport=ON
tls_version=TLSv1.3
...
```

Die Schlüssel für den MySQL-CE-Server und den Client (Zabbix Frontend) sollten gemäß der MySQL-CE-Dokumentation manuell erstellt werden: Creating SSL and RSA certificates and keys using MySQL oder Creating SSL certificates and keys using openssl

Attention:
Das Serverzertifikat von MySQL sollte das Feld Common Name enthalten, das auf den FQDN-Namen gesetzt ist, da das Zabbix Frontend den DNS-Namen zur Kommunikation mit der Datenbank oder die IP-Adresse des Datenbank-Hosts verwendet.

MySQL-Benutzer erstellen:

```
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;
```

Prüfen Sie, ob eine Anmeldung mit diesem Benutzer möglich ist:

```
mysql -u zbx_web -p -h 10.211.55.9 --ssl-mode=VERIFY_IDENTITY --ssl-ca=/var/lib/mysql/ca.pem --ssl-cert=/v
```

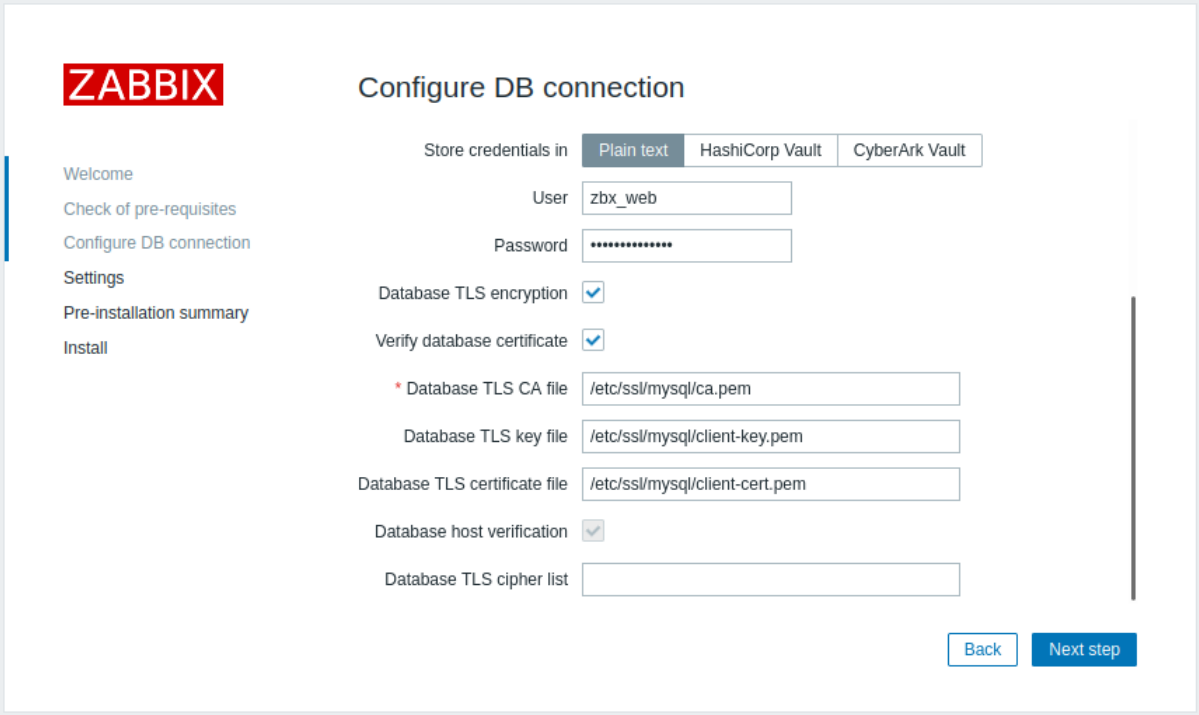
Frontend

So aktivieren Sie die Verschlüsselung mit vollständiger Überprüfung für Verbindungen zwischen dem Zabbix-Frontend und der Datenbank:

- Aktivieren Sie *Database TLS encryption* und *Verify database certificate*.
- Geben Sie den Pfad zur *Database TLS key file* an.
- Geben Sie den Pfad zur *Database TLS CA file* an.
- Geben Sie den Pfad zur *Database TLS certificate file* an.

Beachten Sie, dass *Database host verification* aktiviert und ausgegraut ist – dieser Schritt kann für MySQL nicht übersprungen werden.

Warning:
Wenn das Feld *Database TLS cipher list* leer bleibt, werden die gemeinsamen Chiffren aktiviert, die sowohl vom Frontend (Client) als auch vom Server zugelassen werden. Alternativ können die Chiffren entsprechend den Anforderungen an die Chiffrekonfiguration explizit festgelegt werden.



Alternativ kann dies in der Frontend-Konfigurationsdatei (`zabbix.conf.php`) festgelegt werden:

```
...
// 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'] = '';
...
```

Server

Um die Verschlüsselung mit vollständiger Überprüfung für Verbindungen zwischen dem Zabbix-Server und der Datenbank zu aktivieren, bearbeiten Sie die Server-Konfigurationsdatei:

```
...
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
...
```

PostgreSQL-Verschlüsselungskonfiguration

Übersicht

Dieser Abschnitt enthält mehrere Beispiele für die Verschlüsselungskonfiguration für CentOS 8.2 und PostgreSQL 13.

Note:
Die Verbindung zwischen dem Zabbix-Frontend und PostgreSQL kann nicht verschlüsselt werden (die Parameter in der GUI sind deaktiviert), wenn der Wert des Feldes *Database host* mit einem Schrägstrich beginnt oder das Feld leer ist.

Voraussetzungen

Installieren Sie die PostgreSQL-Datenbank über das offizielle Repository.

PostgreSQL ist standardmäßig nicht so konfiguriert, dass TLS-Verbindungen akzeptiert werden. Bitte folgen Sie den Anweisungen in der PostgreSQL-Dokumentation zur Zertifikatsvorbereitung mit postgresql.conf sowie zur Benutzerzugriffskontrolle über `ph_hba.conf`.

Standardmäßig ist der PostgreSQL-Socket an localhost gebunden, damit für Netzwerk-Remoteverbindungen das Lauschen auf der tatsächlichen Netzwerkschnittstelle erlaubt ist.

Die PostgreSQL-Einstellungen für alle Modi können wie folgt aussehen:

`/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'
...
```

Passen Sie für die Zugriffskontrolle `/var/lib/pgsql/13/data/pg_hba.conf` an:

```
...
### require
hostssl all all 0.0.0.0/0 md5

### verify CA
hostssl all all 0.0.0.0/0 md5 clientcert=verify-ca

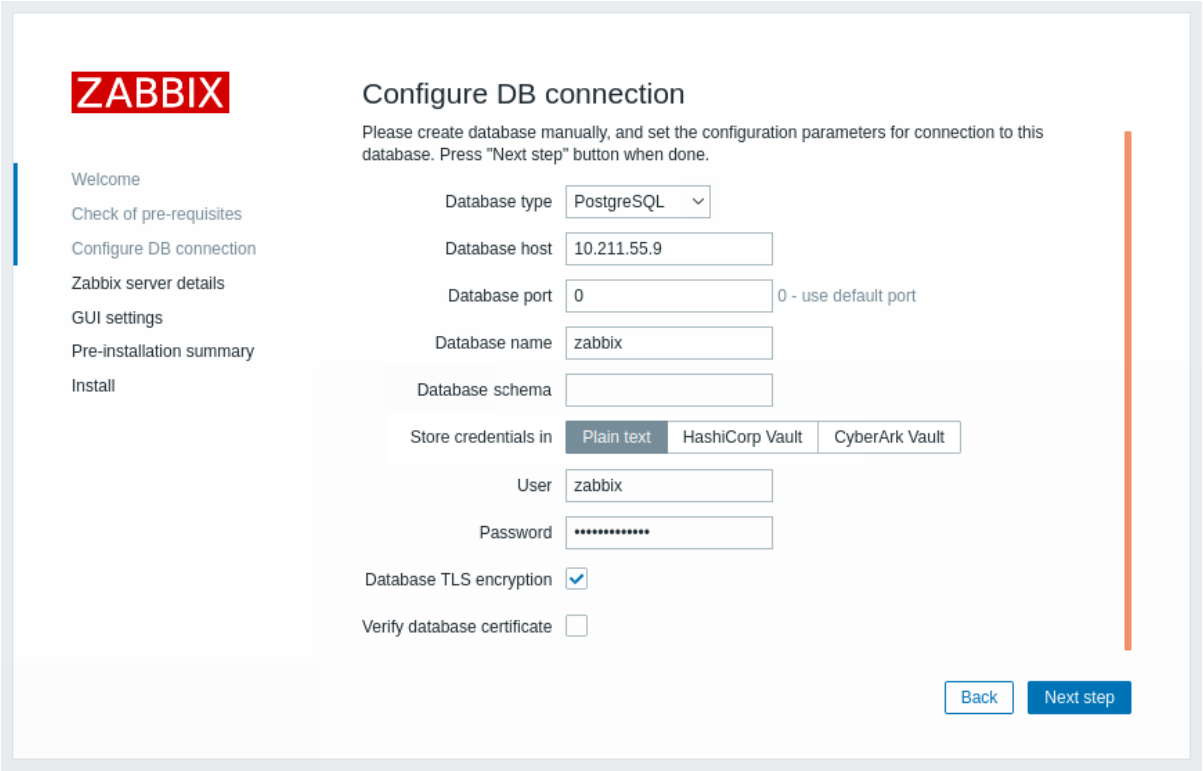
### verify full
hostssl all all 0.0.0.0/0 md5 clientcert=verify-full
...
```

Nur-Transport-Verschlüsselung

Frontend

Um nur die Transportverschlüsselung für Verbindungen zwischen dem Zabbix Frontend und der Datenbank zu aktivieren:

- *Database TLS encryption* aktivieren
- *Verify database certificate* deaktiviert lassen



Server

Um die reine Transportverschlüsselung für Verbindungen zwischen Server und Datenbank zu aktivieren, bearbeiten Sie die Server-Konfigurationsdatei:

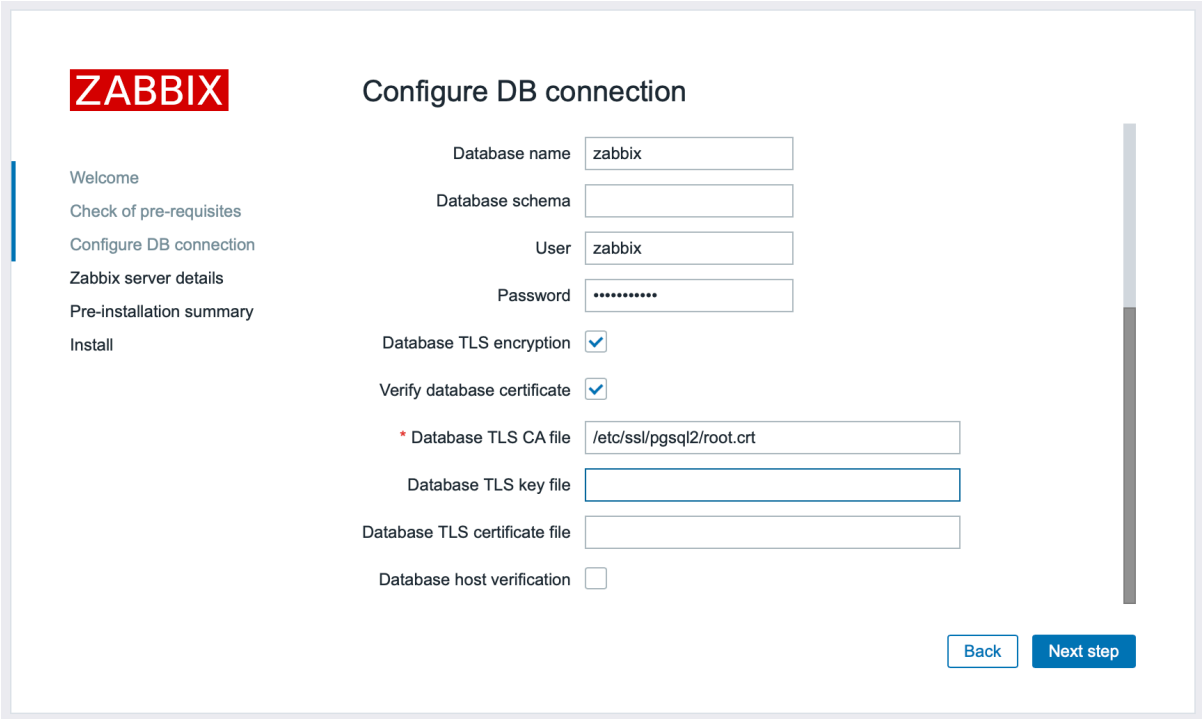
```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

Verschlüsselung mit Überprüfung der Zertifizierungsstelle

Frontend

Um die Verschlüsselung mit Zertifizierungsstellenprüfung für Verbindungen zwischen dem Zabbix-Frontend und der Datenbank zu aktivieren:

- Aktivieren Sie *Datenbank-TLS-Verschlüsselung* und *Datenbankzertifikat überprüfen*
- Geben Sie den Pfad zur *Datenbank-TLS-CA-Datei* an



Alternativ kann dies in der Frontend-Konfigurationsdatei (`zabbix.conf.php`) festgelegt werden:

```
...
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '';
$DB['CERT_FILE'] = '';
$DB['CA_FILE'] = '/etc/ssl/pgsql/root.crt';
$DB['VERIFY_HOST'] = false;
$DB['CIPHER_LIST'] = '';
...
```

Server

Um die Verschlüsselung mit Zertifikatsüberprüfung für Verbindungen zwischen dem Zabbix-Server und der Datenbank zu aktivieren, bearbeiten Sie die Server-Konfigurationsdatei (ersetzen Sie `<strong_password>` durch ein starkes, eindeutiges Passwort):

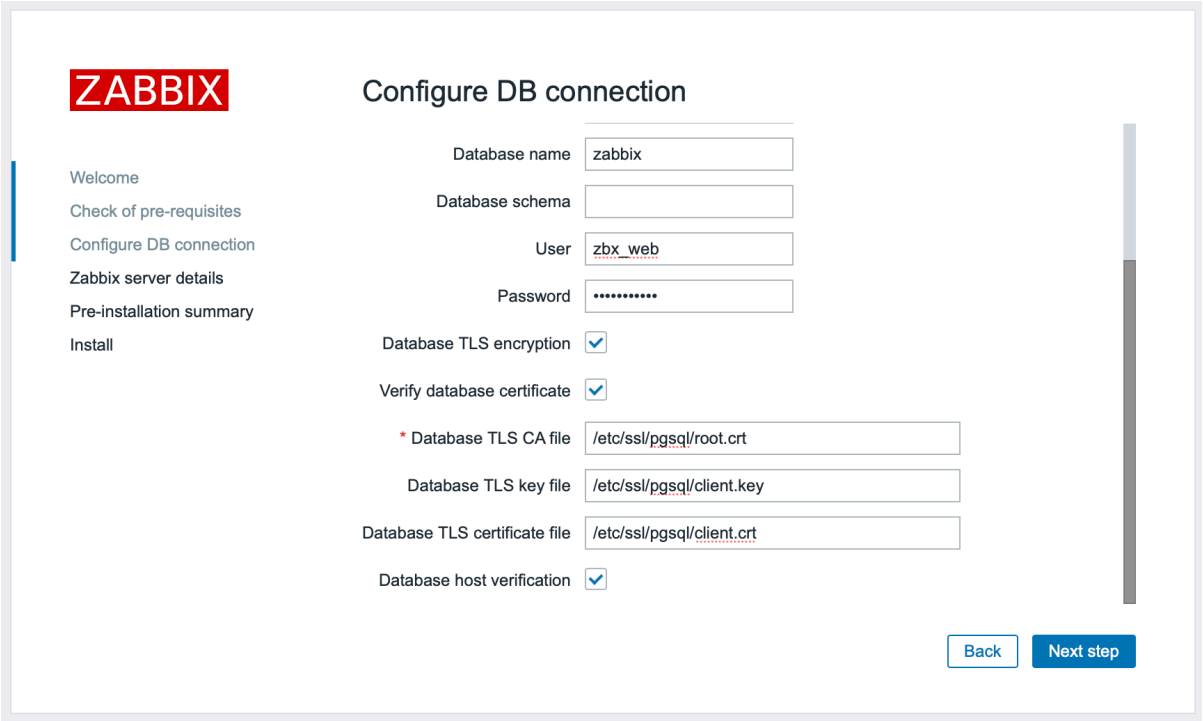
```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_ca
DBTLSCAFile=/etc/ssl/pgsql/root.crt
...
```

Verschlüsselung mit vollständiger Verifizierung

Frontend

Um die Verschlüsselung mit Zertifikat und die Identitätsprüfung des Datenbank-Hosts für Verbindungen zwischen dem Zabbix-Frontend und der Datenbank zu aktivieren:

- Aktivieren Sie *Datenbank-TLS-Verschlüsselung* und *Datenbankzertifikat überprüfen*
- Geben Sie den Pfad zur *Datei mit dem Datenbank-TLS-Schlüssel* an
- Geben Sie den Pfad zur *Datenbank-TLS-CA-Datei* an
- Geben Sie den Pfad zur *Datenbank-TLS-Zertifikatsdatei* an
- Aktivieren Sie *Datenbank-Host überprüfen*



Alternativ kann dies in der Frontend-Konfigurationsdatei (`zabbix.conf.php`) festgelegt werden:

```
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '';
$DB['CERT_FILE'] = '';
$DB['CA_FILE'] = '/etc/ssl/pgsql/root.crt';
$DB['VERIFY_HOST'] = true;
$DB['CIPHER_LIST'] = '';
...
```

Server

Um die Verschlüsselung mit Zertifikat und die Überprüfung der Identität des Datenbank-Hosts für Verbindungen zwischen dem Zabbix-Server und der Datenbank zu aktivieren, bearbeiten Sie die Server-Konfigurationsdatei:

```
...
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
...
```

Sichere Verbindung zwischen Frontend und Server

Übersicht

Dieser Abschnitt enthält Schritte zur Einrichtung von Zabbix und Konfigurationsbeispiele für sichere TLS-Verbindungen zwischen dem Zabbix Frontend und dem Zabbix Server.

Konfiguration

Standardmäßig ist die Kommunikation zwischen dem Zabbix Frontend und dem Zabbix Server nicht verschlüsselt.
Aktivieren Sie TLS auf beiden Seiten, um die Sicherheit zu erhöhen.
Nachfolgend finden Sie ein Beispiel für die einfachste Vorgehensweise.

1. Zertifikate und Schlüssel generieren.

Erstellen Sie ein Arbeitsverzeichnis:

```
sudo mkdir -p /etc/zabbix/ssl && cd /etc/zabbix/ssl
```

Erstellen Sie ein CA-Zertifikat (passen Sie den Wert `MyZabbixCA` an den tatsächlichen allgemeinen Namen an):

```
sudo openssl genrsa -out ca.key 4096
sudo openssl req -new -x509 -days 3650 -key ca.key -out ca.crt -subj "/CN=MyZabbixCA/"
```

Generieren Sie einen privaten Schlüssel und ein Zertifikat für den Zabbix Server (passen Sie den Wert `zabbix-server.example.com` an den tatsächlichen allgemeinen Namen an):

```
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
```

Generieren Sie einen privaten Schlüssel und ein Zertifikat für das Zabbix Frontend (passen Sie den Wert `zabbix-frontend-node` an den tatsächlichen allgemeinen Namen an):

```
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. Legen Sie die entsprechenden Berechtigungen fest.

Für den Zabbix Server (passen Sie Eigentümer und Gruppe entsprechend dem Benutzer des Zabbix-Server-Daemons Ihrer Distribution an):

```
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
```

Für das Frontend (passen Sie Eigentümer und Gruppe entsprechend dem Webserver-Benutzer Ihrer Distribution an):

```
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. Konfigurieren Sie den Zabbix Server.

Fügen Sie in der Server-Konfigurationsdatei (`zabbix_server.conf`) Folgendes hinzu:

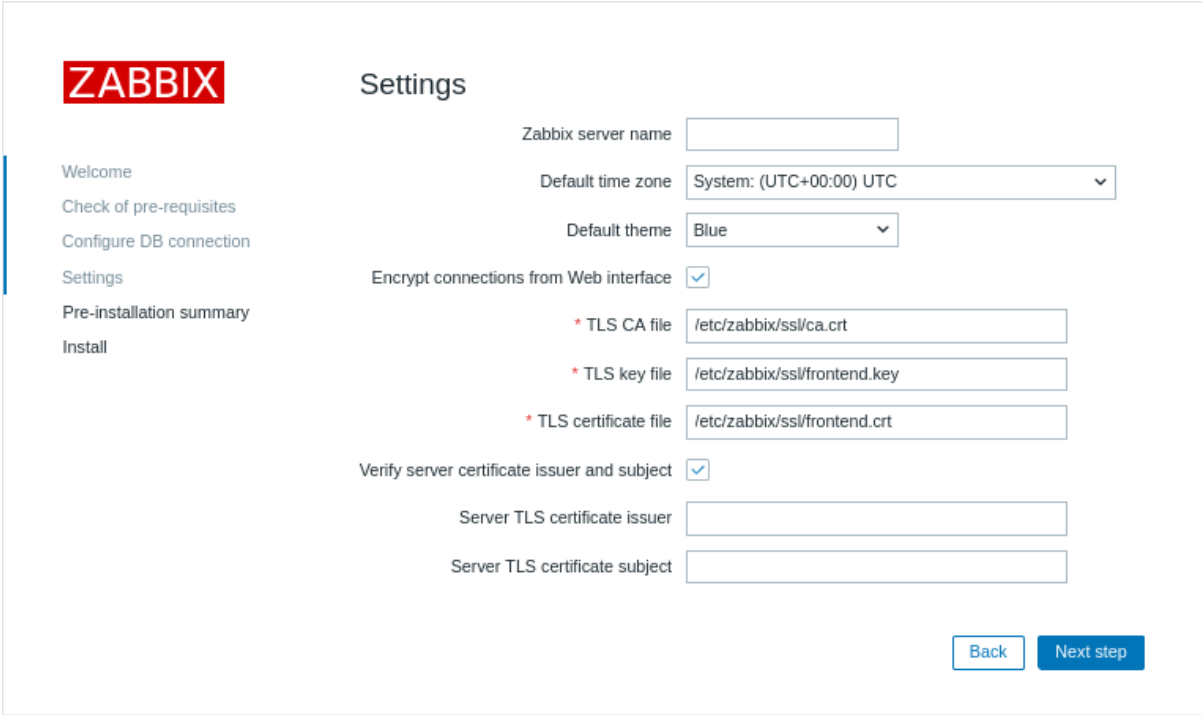
```
TLSFrontendAccept=cert
TLSCertFile=/etc/zabbix/ssl/server.crt
TLSKeyFile=/etc/zabbix/ssl/server.key
TLSCAFile=/etc/zabbix/ssl/ca.crt
### Optional:
### TLSFrontendCertIssuer=CN=MyZabbixCA
### TLSFrontendCertSubject=CN=zabbix-frontend-node
```

Starten Sie anschließend den Server neu:

```
sudo systemctl restart zabbix-server
```

4. Konfigurieren Sie das Zabbix Frontend.

Aktivieren Sie während der Installation der Weboberfläche die Option *Verbindungen von der Weboberfläche verschlüsseln* (und bei Bedarf die Option *Aussteller und Betreff des Serverzertifikats überprüfen*) und füllen Sie die Felder *TLS-CA-Datei*, *TLS-Schlüsseldatei* und *TLS-Zertifikatsdatei* aus (sowie bei Bedarf die Felder *Aussteller des Server-TLS-Zertifikats* und *Betreff des Server-TLS-Zertifikats*):



Parameter	Beschreibung
TLS-CA-Datei	Geben Sie den vollständigen Pfad zur Zertifikatsdatei der Zertifizierungsstelle (CA) an, die zur Überprüfung des Serverzertifikats verwendet wird.
TLS-Schlüsseldatei	Geben Sie den vollständigen Pfad zur privaten Clientschlüsseldatei an, die dem Clientzertifikat entspricht.
TLS-Zertifikatsdatei	Geben Sie den vollständigen Pfad zur Clientzertifikatsdatei an, wenn eine gegenseitige TLS-Authentifizierung erforderlich ist.
Aussteller des Server-TLS-Zertifikats	Geben Sie einen Distinguished Name (DN) des Ausstellers an, der mit dem Serverzertifikat abgeglichen werden soll.
Betreff des Server-TLS-Zertifikats	Geben Sie einen Distinguished Name (DN) des Betreffs an, der mit dem Serverzertifikat abgeglichen werden soll.

Bearbeiten Sie bei bestehenden Installationen die folgenden Parameter in der Frontend-Konfigurationsdatei:

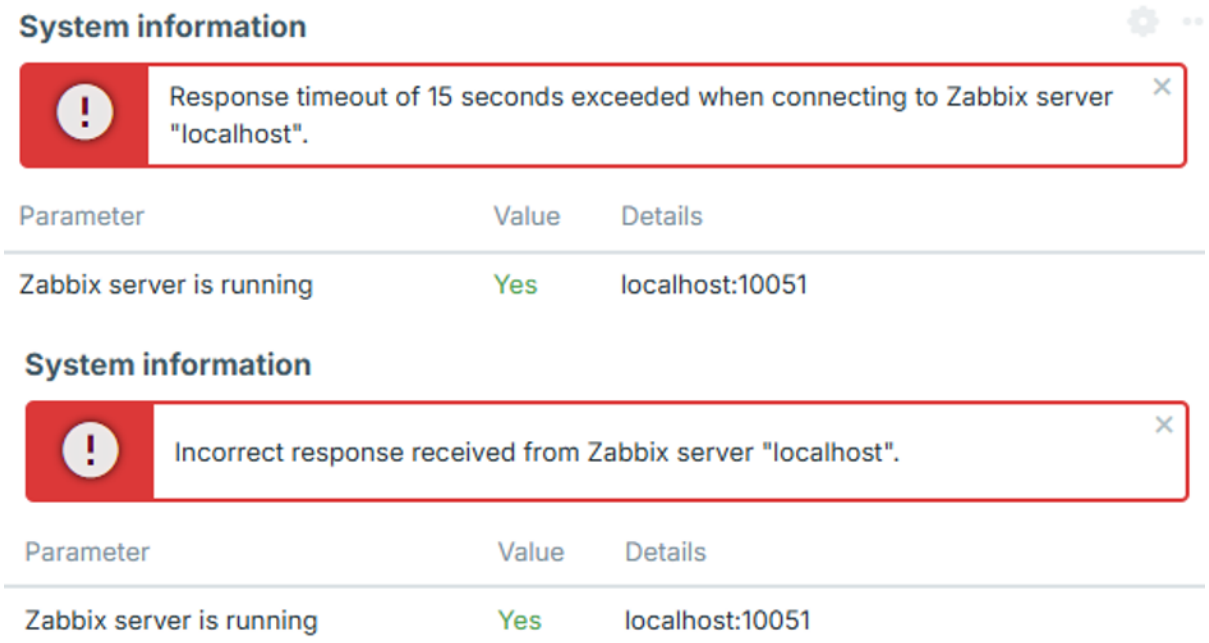
```
$ZBX_SERVER_TLS['ACTIVE'] = 'true';
$ZBX_SERVER_TLS['CA_FILE'] = '/etc/zabbix/ssl/ca.crt';
$ZBX_SERVER_TLS['KEY_FILE'] = '/etc/zabbix/ssl/frontend.key';
$ZBX_SERVER_TLS['CERT_FILE'] = '/etc/zabbix/ssl/frontend.crt';
// Optional:
// $ZBX_SERVER_TLS['CERTIFICATE_ISSUER']  = 'CN=MyZabbixCA';
// $ZBX_SERVER_TLS['CERTIFICATE_SUBJECT'] = 'CN=zabbix-server.example.com';
```

5. Überprüfen Sie die Verschlüsselung, indem Sie sicherstellen, dass im Zabbix Frontend oder in der Protokolldatei des Zabbix Servers keine Fehlermeldungen vorhanden sind:

```
tail -f /var/log/zabbix/zabbix_server.log
```

Fehlerbehebung

Eine falsche oder unvollständige TLS-Konfiguration kann Fehler im Zabbix Frontend verursachen oder darauf hinweisen, dass der Zabbix Server ausgeführt wird, die Verbindung jedoch nicht erfolgreich hergestellt werden kann:



Um das genaue Konfigurationsproblem zu identifizieren, setzen Sie die Protokollstufe des Zabbix Servers auf 4, reproduzieren Sie den Fehler im Frontend und überprüfen Sie das Protokoll des Zabbix Servers auf die entsprechende Fehlermeldung.

Die folgenden Fälle beschreiben häufige Fehler und deren mögliche Ursachen.

Die IP-Adresse des Frontends ist nicht zulässig

Das Protokoll des Zabbix-Servers enthält einen ähnlichen Fehler:

```
54283:20260819:105926.400 frontend connection from "127.0.0.1" is not allowed by FrontendAllowedIP
```

Die Verbindung wird nach dem TCP-Handshake getrennt, da die IP-Adresse des Frontends nicht autorisiert ist.

Fügen Sie die IP-Adresse des Frontends zum Parameter `FrontendAllowedIP` in `zabbix_server.conf` hinzu und starten Sie den Zabbix-Server neu.

Unverschlüsselte Verbindungen sind nicht zulässig

Das Protokoll des Zabbix-Servers enthält einen ähnlichen Fehler:

```
56884:20260819:111154.285 frontend connection of type "unencrypted" is not allowed by TLSFrontendAccept
```

Der Zabbix-Server ist so konfiguriert, dass er nur TLS-Verbindungen akzeptiert, beispielsweise mit `TLSFrontendAccept=cert`, aber das Frontend versucht, eine unverschlüsselte Verbindung herzustellen.

Aktivieren Sie die Verschlüsselung für das Frontend und konfigurieren Sie die erforderlichen TLS-Dateien in `zabbix.conf.php`, wie in Schritt 4 der Konfiguration beschrieben.

Frontend versucht TLS, während der Server unverschlüsselte Verbindungen erwartet

Das Protokoll des Zabbix-Servers enthält einen ähnlichen Fehler:

```
61091:20260819:115916.019 End of zbx_tls_accept():FAIL error:'TLS handshake set result code to 1: file ../
```

Die genaue Fehlermeldung hängt von der OpenSSL-Version ab.

Der Zabbix-Server ist so konfiguriert, dass er nur unverschlüsselte Verbindungen akzeptiert, beispielsweise mit `TLSFrontendAccept=unencr` während das Frontend versucht, eine TLS-Verbindung herzustellen.

Konfigurieren Sie den Zabbix-Server so, dass er TLS-Verbindungen akzeptiert, indem Sie `TLSFrontendAccept=cert` festlegen und `TLSCertFile` sowie `TLSKeyFile` in `zabbix_server.conf` definieren.
Alternativ können Sie die Verschlüsselung des Frontends in `zabbix.conf.php` deaktivieren.

Frontend-Zertifikat wird nicht vertraut

Das Protokoll des Zabbix-Servers enthält einen ähnlichen Fehler:

```
62248:20260819:124203.812 End of zbx_tls_accept():FAIL error:'self-signed certificate: TLS handshake set r
```

Das Frontend hat ein Zertifikat präsentiert, aber der Zabbix-Server vertraut der Zertifizierungsstelle (CA), die es signiert hat, nicht.

Dasselbe Problem kann auch in den Systeminformationen des Frontends angezeigt werden:

```
fread(): SSL operation failed with code 1. OpenSSL Error messages:

error:0A000418:SSL routines::tlsv1 alert unknown ca [zabbix.php:17 → require_once() → ZBase->run() → ZBase
```

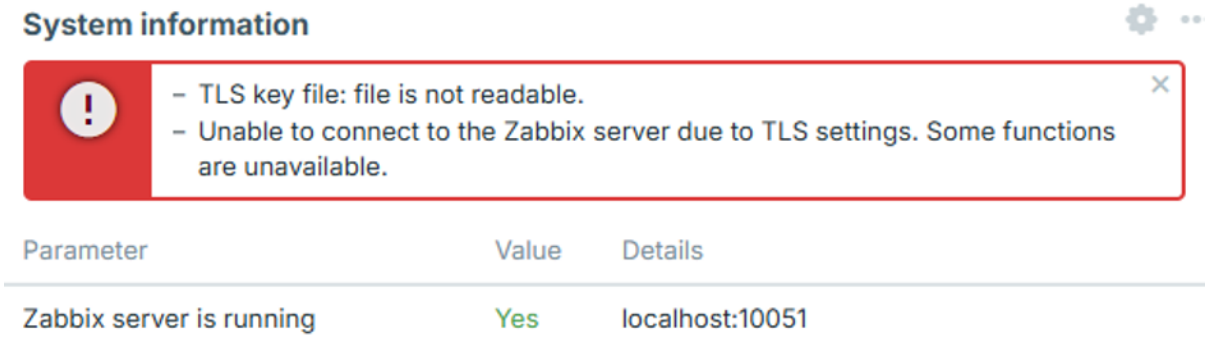
Das Frontend kann außerdem „Incorrect response received from Zabbix server "localhost"“ anzeigen.

Überprüfen Sie, ob das Frontend-Zertifikat von der CA signiert wurde, die in `zabbix_server.conf` durch `TLSCAFile` angegeben ist.

TLS-Schlüsseldatei ist nicht lesbar

Das Frontend zeigt möglicherweise „TLS key file: file is not readable.“ an.

Möglicherweise wird auch „Unable to connect to the Zabbix server due to TLS settings. Some functions are unavailable.“ angezeigt.

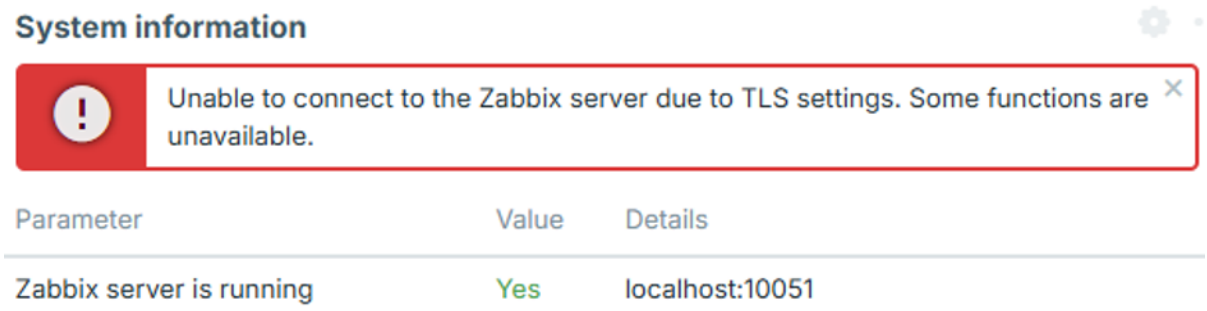


Der Benutzer des Webservers verfügt nicht über ausreichende Berechtigungen, um eine oder mehrere in `zabbix.conf.php` konfigurierte Zertifikat-, Schlüssel- oder CA-Dateien zu lesen. Dies kann auftreten, wenn Zertifikatdateien in einem eingeschränkten Verzeichnis wie /root/ oder im Home-Verzeichnis eines Benutzers gespeichert sind.

Verschieben Sie die Zertifikatdateien in ein für den Webserver zugängliches Verzeichnis und legen Sie die entsprechenden Berechtigungen fest, wie in Schritt 2 der Konfiguration beschrieben.

TLS-Dateien sind ungültig oder beschädigt

Das Frontend zeigt möglicherweise „Aufgrund der TLS-Einstellungen konnte keine Verbindung zum Zabbix-Server hergestellt werden. Einige Funktionen sind nicht verfügbar.“ an.



Dies kann auftreten, wenn die konfigurierten Zertifikats- oder Schlüsseldateien vorhanden und lesbar sind, ihr Inhalt jedoch leer, beschädigt oder anderweitig ungültig ist. In diesem Fall stellt das Frontend keine Verbindung zum Zabbix-Server her, und im Protokoll des Zabbix-Servers wird kein entsprechender Verbindungsversuch angezeigt.

Überprüfen Sie, ob die Zertifikats- und Schlüsseldateien gültige, korrekt formatierte Daten enthalten und ob der private Schlüssel dem Zertifikat entspricht.

Zugriffskontrolle

Übersicht

Dieser Abschnitt enthält bewährte Verfahren für die sichere Einrichtung der Zugriffskontrolle.

Prinzip der geringsten Privilegien

Benutzerkonten sollten jederzeit mit so wenigen Privilegien wie möglich ausgeführt werden. Das bedeutet, dass Benutzerkonten im Zabbix Frontend, Datenbankbenutzer oder der Benutzer für Zabbix Server-/Proxy-/Agent-Prozesse nur über die Privilegien verfügen sollten, die für die Ausführung der vorgesehenen Funktionen unbedingt erforderlich sind.

Attention:
Das Gewähren zusätzlicher Privilegien für den Benutzer 'zabbix' ermöglicht ihm den Zugriff auf Konfigurationsdateien und die Ausführung von Vorgängen, die die Sicherheit der Infrastruktur gefährden können.

Bei der Konfiguration von Benutzerkontenberechtigungen sollten die Frontend-Benutzertypen von Zabbix berücksichtigt werden. Beachten Sie, dass der Benutzertyp *Admin* zwar weniger Privilegien als der Benutzertyp *Super Admin* hat, er jedoch weiterhin Konfigurationen verwalten und benutzerdefinierte Skripte ausführen kann.

Note:
Einige Informationen sind auch für nicht privilegierte Benutzer verfügbar. Während beispielsweise *Alerts* → *Scripts* nur für *Super Admin*-Benutzer verfügbar ist, können Skripte auch über die Zabbix API abgerufen werden. In diesem Fall kann es helfen, die Skriptberechtigungen einzuschränken und sensible Informationen aus Skripten auszuschließen (zum Beispiel Zugangsdaten), um zu vermeiden, dass sensible Informationen in globalen Skripten offengelegt werden.

Sicheren Benutzer für Zabbix Agent

Standardmäßig teilen sich die Prozesse von Zabbix Server, Proxy und Agent (oder Agent 2) einen `zabbix`-Benutzer. Um zu verhindern, dass Zabbix Agent/Agent 2 (der auf demselben Rechner wie Server/Proxy ausgeführt wird) auf sensible Details in der Konfiguration von Server/Proxy zugreift (z. B. Datenbankanmeldedaten), sollte der Agent unter einem anderen Benutzer ausgeführt werden:

Für Zabbix Agent:

1. Erstellen Sie eine sichere Gruppe und einen Benutzer (z. B. `zabbix-agent`).
2. Setzen Sie diesen Benutzer im Konfigurationsparameter User der Agent-Konfigurationsdatei.
3. Starten Sie den Agent neu, um die Berechtigungen auf den neuen Benutzer zu reduzieren.

Für Zabbix Agent 2 muss die Konfiguration auf Service-Ebene angewendet werden, da die Konfigurationsdatei von Agent 2 den Parameter `User` nicht unterstützt. Ein Beispiel finden Sie unter ZBX-26442.

Schreibzugriff auf die SSL-Konfiguration entziehen (Windows)

Wenn Sie den Zabbix Agent unter Windows kompiliert haben und OpenSSL sich in einem ungeschützten Verzeichnis befindet (z. B. `C:\zabbix`, `c:\openssl-64bit`, `C:\OpenSSL-Win64-111-static` oder `C:\dev\openssl`), stellen Sie sicher, dass nicht-administrativen Benutzern der Schreibzugriff auf dieses Verzeichnis entzogen wird. Andernfalls lädt der Agent die SSL-Einstellungen aus einem Pfad, der von Benutzern ohne erhöhte Rechte geändert werden kann, was zu einer potenziellen Sicherheitslücke führt.

Absicherung der Sicherheit von Zabbix-Komponenten

Einige Funktionen können deaktiviert werden, um die Sicherheit der Zabbix-Komponenten zu erhöhen:

- Die globale Skriptausführung auf dem Zabbix-Server kann deaktiviert werden, indem `EnableGlobalScripts=0` in der Server-Konfiguration festgelegt wird.
- Die globale Skriptausführung auf dem Zabbix-Proxy ist standardmäßig deaktiviert und kann durch Festlegen von `EnableRemoteCommands=1` in der Proxy-Konfiguration aktiviert werden.
- Die globale Skriptausführung auf Zabbix-Agents ist standardmäßig deaktiviert und kann aktiviert werden, indem für jeden zulässigen Befehl ein Parameter `AllowKey=system.run[<command>,*]` oder `AllowKeyRegexp=^system\.run\[.*\]$` in der Agent-Konfiguration hinzugefügt wird.
- Der Zugriff auf die Konfiguration der HTTP-Authentifizierung von Benutzern ist standardmäßig aktiviert und kann deaktiviert werden, indem `$ZBX_FEATURE_FLAGS['http_auth_enabled'] = false` in der Frontend-Konfiguration festgelegt wird.
- Der Zugriff auf die Konfiguration der Module ist standardmäßig aktiviert und kann deaktiviert werden, indem `$ZBX_FEATURE_FLAGS['m = false` in der Frontend-Konfiguration festgelegt wird.
- Das Bearbeiten bestimmter Medientypen kann eingeschränkt werden, indem `$ZBX_FEATURE_FLAGS['media_type_denylist'] = [<value1>, <value2>, ...]` in der Frontend-Konfiguration festgelegt wird. Mögliche Werte: `'email'`, `'script'`, `'sms'`, `'webhook'` und `'push'`. Es können ein oder mehrere Werte angegeben werden.

Die Neuinstallation des Frontends durch Ausführen von `setup.php` ist eingeschränkt, wenn die Konfigurationsdatei `zabbix.conf.php` vorhanden ist.

UNC-Pfadzugriff unter Windows durch den Zabbix Agent

Zabbix Agents unter Windows folgen UNC-Pfaden (SMB-Freigaben wie `\\server\share\file.txt`) in Datenpunkten wie `vfs.file.*`, `vfs.dir.*`, `modbus.get` und `perf_counter*`. Dies kann in bestimmten Kontexten ein Sicherheitsrisiko darstellen.

Wenn Windows aufgefordert wird, auf einen UNC-Pfad zuzugreifen, versucht es, sich bei diesem Server zu authentifizieren. Das bedeutet, dass eine bösartige Anfrage an den Zabbix Agent den NTLM-Hash an den Server des Anfragenden weitergeben

kann. Benutzer können dieses Risiko mindern, indem sie die Konfigurationsparameter AllowKey, AllowKeyRegexp, DenyKey und DenyKeyRegexp nach Bedarf konfigurieren.

Absichern von MySQL/MariaDB

Übersicht

Dieser Abschnitt enthält Best Practices zum Absichern einer MySQL/MariaDB-Datenbank.

> **Note:**
> Für eine grundlegende Einrichtung siehe die Standardanweisungen zur Erstellung der MySQL/MariaDB-Datenbank, die das Anlegen des Benutzers 'zabbix' mit vollständigen Berechtigungen für die Zabbix-Datenbank umfassen. Dieser Benutzer ist der Datenbankbesitzer und verfügt außerdem über die erforderlichen Berechtigungen zum Ändern der Datenbankstruktur beim Upgrade von Zabbix.

Um die Sicherheit zu verbessern, wird empfohlen, zusätzliche Datenbankrollen und Benutzer mit minimalen Berechtigungen anzulegen. Diese Rollen und Benutzer sollten auf Grundlage des Prinzips der geringsten Privilegien konfiguriert werden, das heißt, sie sollten nur über Berechtigungen verfügen, die für die Ausführung der vorgesehenen Funktionen unbedingt erforderlich sind.

Erstellen von Benutzerrollen

Erstellen Sie die folgenden Rollen mit den entsprechenden Berechtigungen:

- **zbx_srv** - Rolle für den Betrieb von Zabbix Server und Proxy:

```
CREATE ROLE 'zbx_srv';
GRANT DELETE, INSERT, SELECT, UPDATE ON zabbix.* TO 'zbx_srv';
FLUSH PRIVILEGES;
```

- **zbx_web** - Rolle für den Betrieb von Zabbix Frontend und API:

```
CREATE ROLE 'zbx_web';
GRANT DELETE, INSERT, SELECT, UPDATE ON zabbix.* TO 'zbx_web';
FLUSH PRIVILEGES;
```

- **zbx_bckp** - Rolle für die Tabellensicherung:

```
CREATE ROLE 'zbx_bckp';
GRANT LOCK TABLES, TRIGGER, SELECT ON zabbix.* TO 'zbx_bckp';
GRANT process ON *.* TO 'zbx_bckp';
FLUSH PRIVILEGES;
```

> **Note:**
> Die Wiederherstellung und Aktualisierung von Tabellen sollte vom Datenbankbesitzer durchgeführt werden.

- **zbx_part** - Rolle mit einem reduzierten Satz von Berechtigungen für die Datenbankpartitionierung; beachten Sie, dass diese Rolle erst nach dem Erstellen der Datenbank angelegt werden kann, da sie Berechtigungen für bestimmte Datenbanktabellen vergibt:

```
CREATE ROLE 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history_uint TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history_str TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history_text TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.history_log TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.trends TO 'zbx_part';
GRANT SELECT, ALTER, DROP ON zabbix.trends_uint TO 'zbx_part';
-- For MariaDB: skip the next line (GRANT session_variables_admin ON *.* TO 'zbx_part';)
GRANT session_variables_admin ON *.* TO 'zbx_part';
GRANT SELECT ON zabbix.dbversion TO 'zbx_part';
GRANT SELECT, DELETE ON zabbix.housekeeper TO 'zbx_part';
FLUSH PRIVILEGES;
```

Nach dem Erstellen der Rollen können sie Benutzern zugewiesen werden.

Zuweisen von Benutzerrollen

Um die erstellten Benutzerrollen zuzuweisen, erstellen Sie Benutzer und weisen Sie ihnen die entsprechenden Rollen zu.
Ersetzen Sie <user>, <host>, <role> und <password> durch die entsprechenden Werte.
Stellen Sie sicher, dass <password> sicher und eindeutig ist.

```
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;
```

Um beispielsweise die Rolle für das Ausführen von Zabbix Server und Proxy zu erstellen und zuzuweisen:

```
CREATE USER 'usr_srv'@'localhost' IDENTIFIED BY 'strong-password';
GRANT 'zbx_srv' TO 'usr_srv'@'localhost';
SET DEFAULT ROLE ALL TO 'usr_srv'@'localhost';
FLUSH PRIVILEGES;
```

Absichern von PostgreSQL/TimescaleDB

Übersicht

Dieser Abschnitt enthält Best Practices zum Absichern einer PostgreSQL-Datenbank.

> **Note:**
> Für eine grundlegende Einrichtung siehe die Standardanweisungen zur Erstellung der PostgreSQL-Datenbank, die das Anlegen des Benutzers 'zabbix' mit vollständigen Berechtigungen für die Zabbix-Datenbank umfassen. Dieser Benutzer ist der Datenbankbesitzer und verfügt außerdem über die erforderlichen Berechtigungen zum Ändern der Datenbankstruktur beim Upgrade von Zabbix.

Zur Verbesserung der Sicherheit wird empfohlen, ein sicheres Schema-Nutzungsmuster zu konfigurieren sowie zusätzliche Datenbankrollen und Benutzer mit minimalen Berechtigungen anzulegen. Diese Rollen und Benutzer sollten auf Grundlage des Prinzips der geringsten Privilegien konfiguriert werden, das heißt, sie sollten nur über die Berechtigungen verfügen, die für die Ausführung der vorgesehenen Funktionen unbedingt erforderlich sind.

Datenbankeinrichtung

Erstellen Sie den Benutzer, der der Datenbankbesitzer sein wird, und erstellen Sie die Zabbix-Datenbank; der Datenbankbesitzer ist der Benutzer, der bei der Datenbankerstellung angegeben wird:

```
createuser -U postgres -h localhost --pwprompt usr_owner
createdb -U postgres -h localhost -O usr_owner -E Unicode -T template0 zabbix
```

> **Note:**
> Eine Neuinstallation oder ein Upgrade der Datenbank muss vom Datenbankbesitzer durchgeführt werden. Dies liegt daran, dass das Recht, ein Datenbankobjekt zu löschen oder seine Definition zu ändern, ein dem Datenbankbesitzer inhärentes Privileg ist und weder gewährt noch entzogen werden kann.

> **Attention:**
> Die folgenden Befehle auf dieser Seite müssen ausgeführt werden, während die Verbindung zu PostgreSQL speziell zur zabbix-Datenbank hergestellt ist.

Erstellen Sie das zabbix-Schema und legen Sie den Datenbankbesitzer (usr_owner) als Besitzer dieses Schemas fest:

```
CREATE SCHEMA zabbix AUTHORIZATION usr_owner;
```

Konfigurieren Sie ein sicheres Schema-Verwendungsmuster:

```
REVOKE CREATE ON SCHEMA public FROM PUBLIC;
REVOKE ALL ON DATABASE zabbix FROM PUBLIC;
-- Hinweis: search_path sollte auf das "zabbix"-Schema verweisen:
ALTER ROLE ALL IN DATABASE zabbix SET search_path = "zabbix";
```

Nachdem die Datenbank eingerichtet wurde, fahren Sie mit der Erstellung von Benutzerrollen fort.

Erstellen von Benutzerrollen

Erstellen Sie die folgenden Rollen mit den entsprechenden Berechtigungen:

- **zbx_srv** - Rolle für den Betrieb von Zabbix Server und Proxy:

```
CREATE ROLE zbx_srv;
GRANT CONNECT ON DATABASE zabbix TO zbx_srv;
GRANT USAGE ON SCHEMA zabbix TO zbx_srv;
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT DELETE, INSERT, SELECT, UPDATE ON TABLE
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT SELECT, UPDATE, USAGE ON sequences TO z
```

- **zbx_web** - Rolle für den Betrieb von Zabbix Frontend und API:

```
CREATE ROLE zbx_web;
GRANT CONNECT ON DATABASE zabbix TO zbx_web;
GRANT USAGE ON SCHEMA zabbix TO zbx_web;
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT DELETE, INSERT, SELECT, UPDATE ON TABLE
ALTER DEFAULT PRIVILEGES FOR ROLE usr_owner IN SCHEMA zabbix GRANT SELECT, UPDATE, USAGE ON sequences TO z
```

- **zbx_bckp** - Rolle für das Sichern von Tabellen:

```
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:**
> Die Wiederherstellung von Tabellen ist nur durch den Datenbankbesitzer möglich.

Nach dem Erstellen der Rollen können diese Benutzern zugewiesen werden.

Zuweisen von Benutzerrollen

Um die erstellten Benutzerrollen zuzuweisen, erstellen Sie Benutzer und weisen Sie ihnen die entsprechenden Rollen zu.
Ersetzen Sie <user>, <role> und <password> durch die entsprechenden Werte.
Stellen Sie sicher, dass <password> sicher und eindeutig ist.

```
CREATE USER <user> WITH ENCRYPTED password '<password>';
GRANT <role> TO <user>;
```

Um beispielsweise die Rolle für das Ausführen von Zabbix Server und Proxy zu erstellen und zuzuweisen:

```
CREATE USER usr_srv WITH ENCRYPTED password 'strong-password';
GRANT zbx_srv TO usr_srv;
```

Datenbankpartitionierung mit TimescaleDB

Die Datenbankpartitionierung wird durch TimescaleDB erleichtert. Um TimescaleDB zu nutzen, benötigt der Zabbix Server Datenbankbesitzerrechte.

Wenn das PostgreSQL-Schema zabbix in der Datenbank zabbix bereits erstellt wurde, können Sie TimescaleDB mit dem folgenden Befehl aktivieren:

```
echo "CREATE EXTENSION IF NOT EXISTS timescaledb WITH SCHEMA zabbix CASCADE;" | sudo -u postgres psql zabb
```

Kryptografie

Übersicht

Dieser Abschnitt enthält Best Practices für die sichere Einrichtung der Kryptografie.

Einrichten von SSL für das Zabbix-Frontend

Installieren Sie auf RHEL-basierten Systemen das Paket mod_ssl:

```
dnf install mod_ssl
```

Erstellen Sie ein Verzeichnis für SSL-Schlüssel:

```
mkdir -p /etc/httpd/ssl/private
chmod 700 /etc/httpd/ssl/private
```

Erstellen Sie das SSL-Zertifikat:

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/httpd/ssl/private/apache-selfsigned.key -
```

Füllen Sie die Eingabeaufforderungen entsprechend aus. Die wichtigste Zeile ist diejenige, die nach dem `Common Name` fragt. Sie müssen den Domainnamen eingeben, der mit Ihrem Server verknüpft werden soll. Sie können stattdessen die öffentliche IP-Adresse eingeben, wenn Sie keinen Domainnamen haben.

```
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 []:
```

Bearbeiten Sie die Apache-SSL-Konfigurationsdatei (`/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
```

Starten Sie den Apache-Dienst neu, um die Änderungen zu übernehmen:

```
systemctl restart httpd.service
```

Absicherung des Webservers

Übersicht

Dieser Abschnitt enthält bewährte Verfahren für die sichere Einrichtung des Webservers.

Erzwingen einer Weiterleitung der Root-URL auf die Zabbix-SSL-URL

Auf RHEL-basierten Systemen fügen Sie der Apache-Konfiguration (`/etc/httpd/conf/httpd.conf`) einen virtuellen Host hinzu und setzen Sie eine permanente Weiterleitung für das Dokumentenstammverzeichnis auf die Zabbix-SSL-URL. Beachten Sie, dass `example.com` durch den tatsächlichen Namen des Servers ersetzt werden sollte.

```
### Zeilen hinzufügen:

<VirtualHost *:*>
    ServerName example.com
    Redirect permanent / https://example.com
</VirtualHost>
```

Starten Sie den Apache-Dienst neu, um die Änderungen zu übernehmen:

```
systemctl restart httpd.service
```

Aktivieren von HTTP Strict Transport Security (HSTS) auf dem Webserver

Um das Zabbix Frontend vor Protocol-Downgrade-Angriffen zu schützen, empfehlen wir, die HSTS-Richtlinie auf dem Webserver zu aktivieren.

Um die HSTS-Richtlinie für Ihr Zabbix Frontend in der Apache-Konfiguration zu aktivieren, führen Sie die folgenden Schritte aus:

1. Suchen Sie die Konfigurationsdatei Ihres virtuellen Hosts:

- `/etc/httpd/conf/httpd.conf` auf RHEL-basierten Systemen
- `/etc/apache2/sites-available/000-default.conf` auf Debian/Ubuntu

2. Fügen Sie die folgende Direktive in die Konfigurationsdatei Ihres virtuellen Hosts ein:

```
<VirtualHost *:*>
    Header set Strict-Transport-Security "max-age=31536000"
</VirtualHost>
```

3. Starten Sie den Apache-Dienst neu, um die Änderungen zu übernehmen:

```
### Auf RHEL-basierten Systemen:
systemctl restart httpd.service

### Auf Debian/Ubuntu
systemctl restart apache2.service
```

Erzwingen sicherer und SameSite-Session-Cookies in Zabbix

Bei der Konfiguration von Zabbix ist es wichtig, die Attribute Secure und SameSite für Session-Cookies zu erzwingen, um die Sicherheit zu erhöhen und Cross-Site-Request-Forgery-(CSRF)-Angriffe zu verhindern. Das Erzwingen von `SameSite=Strict` kann jedoch in bestimmten Szenarien zu Problemen führen, zum Beispiel:

- Dashboard-URL-Widgets zeigen „user not logged in“ an, wenn Iframes aus derselben Domain eingebettet werden.
- Benutzer, die auf das Dashboard über HTTP statt HTTPS zugreifen, können Anmeldeprobleme haben.
- Es ist nicht möglich, URLs zu bestimmten Zabbix-Menübereichen oder Hosts zu teilen.

Um diese Probleme zu vermeiden, sollten Benutzer die Möglichkeit haben, die SameSite-Richtlinie anzupassen.

1. Sichere Cookies

Das Setzen des Flags `secure` stellt sicher, dass Cookies nur über HTTPS übertragen werden, wodurch eine Offenlegung über unverschlüsselte Verbindungen verhindert wird.

Um sichere Cookies in Zabbix zu aktivieren, fügen Sie die folgende Einstellung in der Webserver-Konfiguration hinzu oder ändern Sie sie:

Für Apache:

```
Header always edit Set-Cookie ^(.*)$ $1;Secure
```

Für Nginx:

```
proxy_cookie_path / "/; Secure";
```

Stellen Sie sicher, dass auf Ihr Zabbix-Frontend über HTTPS zugegriffen wird; andernfalls werden Cookies mit dem Flag `Secure` nicht gesendet.

2. Konfigurieren des SameSite-Attributs

Webserver-Einstellungen können auch das SameSite-Attribut erzwingen:

Für Apache:

```
<IfModule mod_headers.c>
    Header onsuccess edit Set-Cookie (.*) "$1; SameSite=Strict"
</IfModule>
```

Für Nginx (Version 1.19.3+):

```
proxy_cookie_flags ~ samesite=Strict; # Ersetzen Sie ~ durch 'zbx_session' für mehr Spezifität
```

Aktivieren von Content Security Policy (CSP) auf dem Webserver

Um das Zabbix-Frontend vor Cross-Site-Scripting-(XSS)-, Dateninjektions- und ähnlichen Angriffen zu schützen, empfehlen wir, Content Security Policy auf dem Webserver zu aktivieren. Konfigurieren Sie dazu den Webserver so, dass er den HTTP-Header zurückgibt.

> **Attention:**
> Die folgende CSP-Header-Konfiguration gilt nur für die Standardinstallation des Zabbix-Frontends und für Fälle, in denen der gesamte Inhalt aus der Domain der Site stammt (mit Ausnahme von Subdomains). Eine andere CSP-Header-Konfiguration kann erforderlich sein, wenn Sie beispielsweise das Widget *URL* so konfigurieren, dass Inhalte aus den Subdomains der Site oder aus externen Domains angezeigt werden, von *OpenStreetMap* auf eine andere Karten-Engine umstellen oder externe CSS-Dateien oder Widgets hinzufügen. Wenn Sie die Duo Universal Prompt-Methode für Multi-Faktor-Authentifizierung verwenden, stellen Sie sicher, dass Sie ”duo.com” zur CSP-Direktive in der Konfigurationsdatei Ihres virtuellen Hosts hinzufügen.

Um CSP für Ihr Zabbix-Frontend in der Apache-Konfiguration zu aktivieren, gehen Sie wie folgt vor:

1. Suchen Sie die Konfigurationsdatei Ihres virtuellen Hosts:

- `/etc/httpd/conf/httpd.conf` auf RHEL-basierten Systemen
- `/etc/apache2/sites-available/000-default.conf` auf Debian/Ubuntu

2. Fügen Sie die folgende Direktive zur Konfigurationsdatei Ihres virtuellen Hosts hinzu:

```
<VirtualHost *:*>
    Header set Content-Security-Policy: "default-src 'self' *.openstreetmap.org; script-src 'self' 'unsafe
</VirtualHost>
```

3. Starten Sie den Apache-Dienst neu, um die Änderungen zu übernehmen:

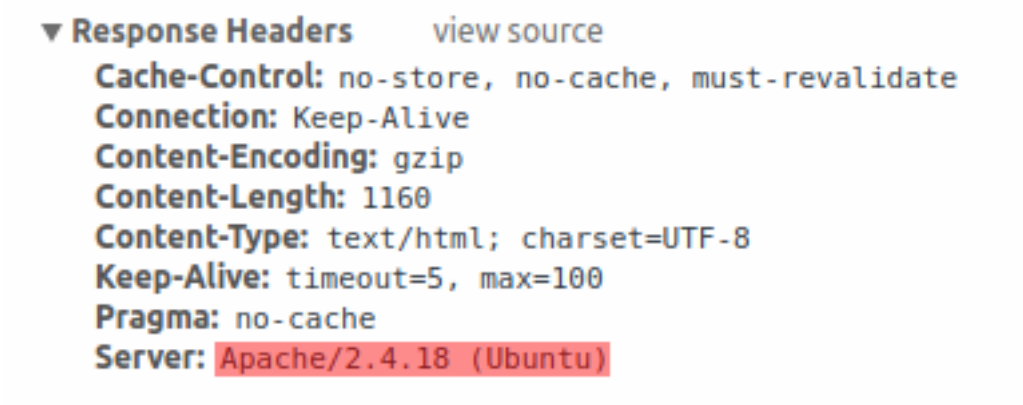
```
### Auf RHEL-basierten Systemen:
systemctl restart httpd.service

### Auf Debian/Ubuntu
systemctl restart apache2.service
```

Deaktivieren der Offenlegung von Webserver-Informationen

Zur Verbesserung der Sicherheit wird empfohlen, alle Webserversignaturen zu deaktivieren.

Standardmäßig legt der Webserver die Softwaresignatur offen:



Die Signatur kann deaktiviert werden, indem die folgenden Parameter zur Apache-Konfigurationsdatei hinzugefügt werden:

```
ServerSignature Off
ServerTokens Prod
```

Die PHP-Signatur (HTTP-Header `X-Powered-By`) kann durch Ändern der Konfigurationsdatei `php.ini` deaktiviert werden (standardmäßig ist die Signatur deaktiviert):

```
expose_php = Off
```

Damit Änderungen an der Konfigurationsdatei übernommen werden, ist ein Neustart des Webservers erforderlich.

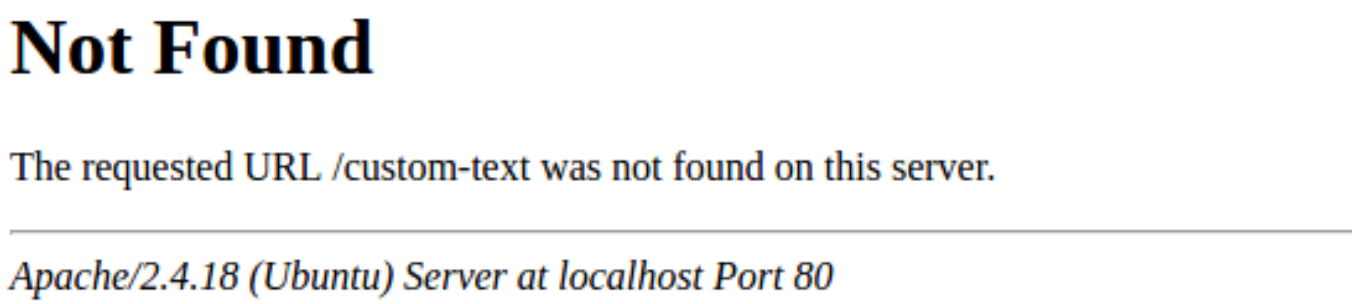
Für zusätzliche Sicherheit können Sie das Tool *mod_security* mit Apache verwenden (Paket *libapache2-mod-security2*). Mit diesem Tool kann die Serversignatur entfernt werden, anstatt nur die Version aus der Serversignatur zu entfernen. Die Serversignatur kann nach der Installation von *mod_security* durch Festlegen von „SecServerSignature“ auf einen beliebigen gewünschten Wert geändert werden.

Weitere Informationen zum Entfernen oder Ändern von Softwaresignaturen finden Sie in der Dokumentation Ihres Webservers.

Deaktivieren der Standardfehlerseiten des Webservers

Um die Offenlegung von Informationen zu vermeiden, wird empfohlen, die Standardfehlerseiten zu deaktivieren.

Standardmäßig verwendet ein Webserver integrierte Fehlerseiten:



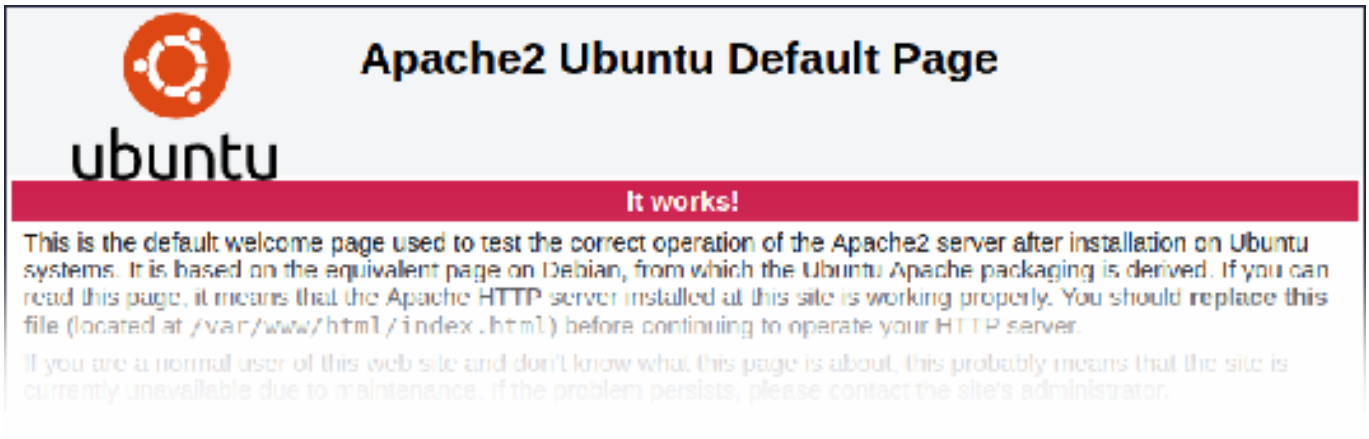
Diese Standardfehlerseiten sollten ersetzt/entfernt werden. Beispielsweise kann die Direktive „ErrorDocument“ verwendet werden, um eine benutzerdefinierte Fehlerseite bzw. einen benutzerdefinierten Fehlertext für den Apache-Webserver zu definieren.

Weitere Informationen zum Ersetzen/Entfernen von Standardfehlerseiten finden Sie in der Dokumentation Ihres Webservers.

Entfernen der Testseite des Webservers

Um die Offenlegung von Informationen zu vermeiden, wird empfohlen, die Testseite des Webservers zu entfernen.

Standardmäßig enthält das Webroot des Apache-Webservers die Testseite `index.html`:



Weitere Informationen zum Entfernen standardmäßiger Testseiten finden Sie in der Dokumentation Ihres Webservers.

Setzen des HTTP-Antwort-Headers X-Frame-Options

Standardmäßig ist Zabbix mit dem Parameter *Use X-Frame-Options HTTP header* auf `SAMEORIGIN` konfiguriert. Das bedeutet, dass Inhalte nur in einem Frame geladen werden können, der denselben Ursprung wie die Seite selbst hat.

Frontend-Elemente von Zabbix, die Inhalte von externen URLs abrufen (nämlich das URL-Dashboard-Widget), zeigen die abgerufenen Inhalte in einer Sandbox mit allen aktivierten Sandbox-Einschränkungen an.

Diese Einstellungen erhöhen die Sicherheit des Zabbix Frontend und bieten Schutz vor XSS- und Clickjacking-Angriffen. Benutzer mit der Rolle *Super admin* können die Parameter *Use iframe sandboxing* und *Use X-Frame-Options HTTP header* bei Bedarf ändern. Bitte wägen Sie die Risiken und Vorteile sorgfältig ab, bevor Sie die Standardeinstellungen ändern. Es wird nicht empfohlen, das iframe-Sandboxing oder den HTTP-Antwort-Header X-Frame-Options vollständig zu deaktivieren.

Ausblenden der Datei mit der Liste häufiger Passwörter

Um die Komplexität von Brute-Force-Angriffen auf Passwörter zu erhöhen, wird empfohlen, den Zugriff auf die Datei `ui/data/top_passwords` einzuschränken. Diese Datei enthält eine Liste der gebräuchlichsten und kontextbezogenen Passwörter und verhindert, dass Benutzer solche Passwörter festlegen (wenn der Parameter *Einfache, leicht zu erratende Passwörter vermeiden* in der Passwortrichtlinie aktiviert ist).

Um den Zugriff auf die Datei `top_passwords.txt` einzuschränken, passen Sie die Konfiguration Ihres Webservers an.

Bei Apache kann der Dateizugriff mithilfe der Datei `.htaccess` eingeschränkt werden:

```
<Files "top_passwords.txt">
    Order Allow,Deny
    Deny from all
</Files>
```

Bei NGINX kann der Dateizugriff mithilfe der Direktive `location` eingeschränkt werden:

```
location = /data/top_passwords.txt {
    deny all;
    return 404;
}
```

Speicherung von Geheimnissen

Übersicht Zabbix kann so konfiguriert werden, dass sensible Informationen aus einem sicheren Vault abgerufen werden. Die folgenden Secret-Management-Dienste werden unterstützt: HashiCorp Vault KV Secrets Engine - Version 2, CyberArk Vault CV12.

Secrets können zum Abrufen folgender Informationen verwendet werden:

- Benutzer-Makro-Werte
- Datenbankzugangsdaten

Zabbix bietet schreibgeschützten Zugriff auf die Secrets in einem Vault, wobei davon ausgegangen wird, dass die Secrets von einer anderen Person verwaltet werden.

Für Informationen zur Konfiguration eines bestimmten Vault-Anbieters siehe:

- HashiCorp-Konfiguration
- CyberArk-Konfiguration

Caching von geheimen Werten Standardmäßig werden die Werte von Vault-Geheimnismakros bei jeder Aktualisierung der Konfigurationsdaten vom Zabbix Server abgerufen und anschließend im Konfigurations-Cache gespeichert. Der Zabbix Proxy empfängt die Werte von Vault-Geheimnismakros bei jeder Konfigurationssynchronisierung vom Zabbix Server und speichert sie in seinem eigenen Konfigurations-Cache.

> **Attention:**
> Die Verschlüsselung zwischen Zabbix Server und Proxy muss aktiviert sein, andernfalls wird eine Warnmeldung auf dem Server protokolliert.

Es ist auch möglich, zu konfigurieren, dass die Makrowerte unabhängig voneinander vom Zabbix Server und Zabbix Proxy abgerufen werden.

Um die Aktualisierung zwischengespeicherter geheimer Werte aus einem Vault manuell auszulösen, verwenden Sie die Befehlszeilen-Option `secrets_reload`.

Für die Anmeldedaten der Zabbix-Frontend-Datenbank ist das Caching standardmäßig deaktiviert, kann jedoch aktiviert werden, indem die Option `$DB['VAULT_CACHE'] = true` in der Frontend-Konfigurationsdatei gesetzt wird. Die Anmeldedaten werden in einem lokalen Cache im temporären Dateiverzeichnis des Dateisystems gespeichert. Der Webserver muss das Schreiben in einen privaten temporären Ordner erlauben (bei Apache muss beispielsweise die Konfigurationsoption `PrivateTmp=True` gesetzt sein). Um festzulegen, wie oft der Daten-Cache aktualisiert bzw. ungültig gemacht wird, verwenden Sie die Konstante `ZBX_DATA_CACHE_TTL`.

TLS-Konfiguration Um TLS für die Kommunikation zwischen Zabbix-Komponenten und dem Vault zu konfigurieren, fügen Sie ein von einer Zertifizierungsstelle (CA) signiertes Zertifikat zum systemweiten standardmäßigen CA-Speicher hinzu.
Wenn Sie einen anderen Speicherort verwenden möchten, geben Sie das Verzeichnis im Zabbix-Konfigurationsparameter SSLCALocation Server/Proxy an,
legen Sie die Zertifikatdatei in diesem Verzeichnis ab und führen Sie dann den CLI-Befehl aus:

```
c_rehash .
```

CyberArk-Konfiguration

In diesem Abschnitt wird erläutert, wie Zabbix konfiguriert wird, um Geheimnisse aus CyberArk Vault CV12 abzurufen.

Der Vault sollte wie in der offiziellen CyberArk-Dokumentation beschrieben installiert und konfiguriert werden.

Informationen zur Konfiguration von TLS in Zabbix finden Sie unter *Speicherung von Geheimnissen*.

Datenbankanmeldedaten

Der Zugriff auf ein Secret mit Datenbankanmeldedaten wird für jede Zabbix-Komponente separat konfiguriert.

Server und Proxys

Um Datenbankanmeldedaten für den Zabbix-Server oder Proxy aus dem Vault abzurufen, geben Sie die folgenden Konfigurationsparameter in der Konfigurationsdatei an:

- `Vault` – welcher Vault-Anbieter verwendet werden soll;
- `VaultURL` – HTTP(S)-URL des Vault-Servers;
- `VaultDBPath` – Abfrage an das Vault-Secret mit den Datenbankanmeldedaten, die über die Schlüssel „Content“ und „UserName“ abgerufen werden (diese Option kann nur verwendet werden, wenn DBUser und DBPassword nicht angegeben sind);
- `VaultTLSCertFile`, `VaultTLSKeyFile` – Dateinamen des SSL-Zertifikats und des Schlüssels; die Konfiguration dieser Optionen ist nicht zwingend erforderlich, wird jedoch dringend empfohlen;

- `VaultPrefix` – benutzerdefiniertes Präfix für den Vault-Pfad oder die Abfrage, abhängig vom Vault; wenn nicht angegeben, wird der am besten geeignete Standardwert verwendet.

Attention:
Die Konfigurationsparameter `Vault`, `VaultURL`, `VaultTLSCertFile`, `VaultTLSKeyFile` und `VaultPrefix` werden auch für die Vault-Authentifizierung verwendet, wenn Secret-Vault-Makros durch den Zabbix-Server (und den Zabbix-Proxy, falls konfiguriert) verarbeitet werden. Der Zabbix-Server und die Proxys öffnen keine Secret-Makros, die DB-Anmeldedaten aus VaultDBPath enthalten.

Der Zabbix-Server und der Zabbix-Proxy lesen die Vault-bezogenen Konfigurationsparameter beim Start aus den Dateien `zabbix_server.conf` und `zabbix_proxy.conf`.

Beispiel

1. Geben Sie in `zabbix_server.conf` die folgenden Parameter an:

```
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 sendet die folgende API-Anfrage an den Vault:

```
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. Die Antwort des Vaults enthält die Schlüssel ”Content” und ”UserName”:

```
{
    "Content": <password>,
    "UserName": <username>,
    "Address": <address>,
    "Database": <Database>,
    "PasswordChangeInProcess":<PasswordChangeInProcess>
}
```

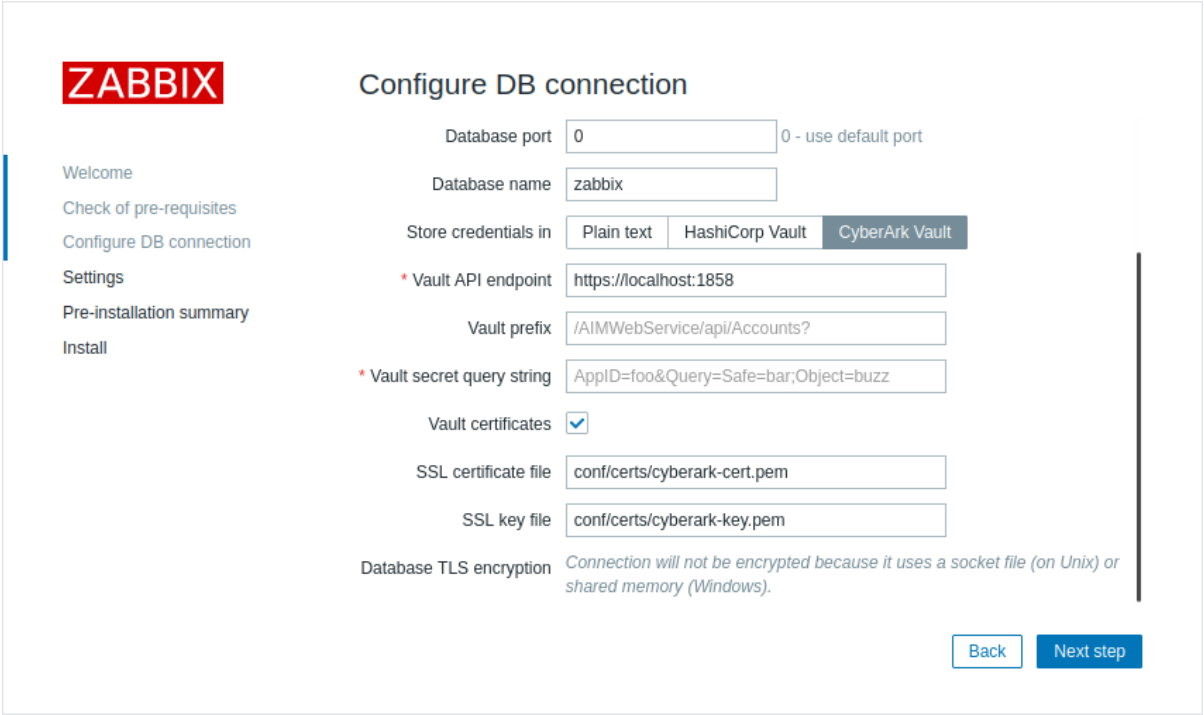
4. Infolgedessen verwendet Zabbix die folgenden Anmeldedaten für die Datenbankauthentifizierung:

- Benutzername: <username>
- Passwort: <password>

Frontend

Um Datenbankanmeldedaten aus dem Vault für das Zabbix-Frontend abzurufen, geben Sie während der Installation des Frontends die folgenden Parameter an.

1. Setzen Sie im Schritt *Configure DB Connection* den Parameter *Store credentials in* auf *CyberArk Vault*.



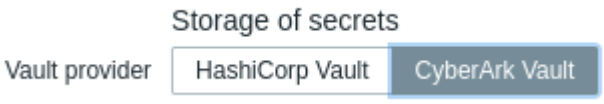
2. Füllen Sie anschließend die zusätzlichen Parameter aus:

Parameter	Mandatory	Default value	Description
Vault API endpoint	yes	https://localhost:1858	Geben Sie die URL für die Verbindung zum Vault im Format `scheme://host:port` an.
Vault prefix	no	/AIMWebService/api/Accounts?	Geben Sie ein benutzerdefiniertes Präfix für den Vault-Pfad oder die Abfrage an. Wenn kein Präfix angegeben wird, wird der Standardwert verwendet.
Vault secret query string	yes		Eine Abfrage, die angibt, aus welcher Quelle die Datenbankanmeldedaten abgerufen werden sollen. Beispiel: `AppID=foo&Query=Safe=bar;Object=buzz`
Vault certificates	no		Nach Aktivierung des Kontrollkästchens werden zusätzliche Parameter angezeigt, mit denen Sie die Clientauthentifizierung konfigurieren können. Obwohl dieser Parameter optional ist, wird dringend empfohlen, ihn für die Kommunikation mit dem CyberArk Vault zu aktivieren.
SSL certificate file	no	conf/certs/cyberark-cert.pem	Pfad zur SSL-Zertifikatsdatei. Die Datei muss im PEM-Format vorliegen. Wenn die Zertifikatsdatei auch den privaten Schlüssel enthält, lassen Sie den Parameter für die SSL-Schlüsseldatei leer.
SSL key file	no	conf/certs/cyberark-key.pem	Name der für die Clientauthentifizierung verwendeten privaten SSL-Schlüsseldatei. Die Datei muss im PEM-Format vorliegen.

Benutzer-Makro-Werte

Um CyberArk Vault zum Speichern von Benutzer-Makro-Werten vom Typ *Vault secret* zu verwenden, stellen Sie sicher, dass:

- der Zabbix Server konfiguriert ist, um mit CyberArk Vault zu arbeiten;
- der Parameter *Vault provider* unter *Administration > Allgemein > Sonstiges* auf "CyberArk Vault" gesetzt ist.



Note:
Der Zabbix Server (und der Zabbix Proxy, falls konfiguriert) benötigt Zugriff auf *Vault secret*-Makro-Werte aus dem Vault. Das Zabbix Frontend benötigt keinen solchen Zugriff.

Der Makro-Wert sollte eine Abfrage enthalten (als `query:key`).

Weitere Informationen zur Verarbeitung von Makro-Werten durch Zabbix finden Sie unter *Vault secret macros*.

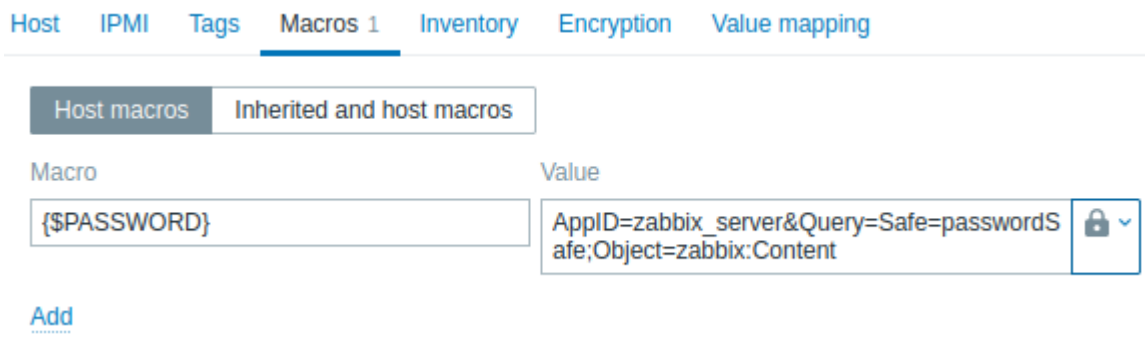
Abfragesyntax

Das Doppelpunktsymbol (`:`) ist für die Trennung der Abfrage vom Schlüssel reserviert.

Wenn eine Abfrage selbst einen Schrägstrich oder einen Doppelpunkt enthält, sollten diese Symbole URL-kodiert werden (`/` wird als `%2F` kodiert, `:` wird als `%3A` kodiert).

Beispiel

1. Fügen Sie in Zabbix ein Benutzermakro `{$PASSWORD}` vom Typ *Vault secret* mit dem Wert `AppID=zabbix_server&Query=Safe=pas` hinzu.



2. Zabbix sendet die folgende API-Anforderung an den Vault:

```
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. Die Antwort des Vaults enthält den Schlüssel `Content`:

```
{
    "Content": <password>,
    "UserName": <username>,
    "Address": <address>,
    "Database" :<Database>,
    "PasswordChangeInProcess":<PasswordChangeInProcess>
}
```

4. Als Ergebnis löst Zabbix das Makro `{$PASSWORD}` in den Wert `<password>` auf.

Vorhandene Konfiguration aktualisieren

So aktualisieren Sie eine vorhandene Konfiguration zum Abrufen von Geheimnissen aus einem CyberArk Vault:

1. Aktualisieren Sie die Parameter in der Konfigurationsdatei des Zabbix Servers oder Proxys wie im Abschnitt *Datenbankanmeldeinformationen* beschrieben.
2. Aktualisieren Sie die Parameter in der Frontend-Konfigurationsdatei von Zabbix:

```
$DB['VAULT']                    = 'CyberArk';
$DB['VAULT_URL']                = 'https://127.0.0.1:1858';
$DB['VAULT_DB_PATH']            = 'AppID=foo&Query=Safe=bar;Object=buzz';
$DB['VAULT_TOKEN']              = '';
$DB['VAULT_CERT_FILE']          = 'conf/certs/cyberark-cert.pem';
$DB['VAULT_KEY_FILE']           = 'conf/certs/cyberark-key.pem';
$DB['VAULT_PREFIX']             = '';
```

3. Konfigurieren Sie bei Bedarf Benutzermakros wie im Abschnitt *Werte von Benutzermakros* beschrieben.

Informationen zum Aktualisieren einer vorhandenen Konfiguration zum Abrufen von Geheimnissen aus einem HashiCorp Vault finden Sie unter *HashiCorp-Konfiguration*.

HashiCorp-Konfiguration

Übersicht

Dieser Abschnitt erklärt, wie Zabbix so konfiguriert wird, dass Geheimnisse aus der HashiCorp Vault KV Secrets Engine - Version 2 abgerufen werden.

Der Vault sollte wie in der offiziellen HashiCorp-Dokumentation beschrieben bereitgestellt und konfiguriert werden.

Informationen zur Konfiguration von TLS in Zabbix finden Sie unter *Speicherung von Geheimnissen*.

Abrufen von Datenbankanmeldedaten

Um ein Secret mit Datenbankanmeldedaten erfolgreich abzurufen, müssen beide Komponenten konfiguriert werden:

- Zabbix Server/Proxy
- Zabbix Frontend

Server/Proxy

Um den Zabbix Server oder Proxy zu konfigurieren, geben Sie die folgenden Konfigurationsparameter in der Konfigurationsdatei an:

- `Vault` – welcher Vault-Anbieter verwendet werden soll.
- `VaultToken` – Vault-Authentifizierungstoken (Details finden Sie in der Zabbix-Konfigurationsdatei für Server/Proxy); kann nicht zusammen mit `VaultAppRoleID` verwendet werden.
- `VaultAppRoleID` und `VaultAppSecretID` – Rollen-ID und Secret-ID für die Vault-Authentifizierung (Details finden Sie in der Zabbix-Konfigurationsdatei für Server/Proxy); können nicht zusammen mit `VaultToken` verwendet werden.
- `VaultURL` – HTTP(S)-URL des Vault-Servers.
- `VaultDBPath` – Pfad zum Vault-Secret mit den Datenbankanmeldedaten (diese Option kann nur verwendet werden, wenn DBUser und DBPassword nicht angegeben sind). Der Zabbix Server oder Proxy ruft die Anmeldedaten anhand der Schlüssel „password“ und „username“ ab.
- `VaultPrefix` – benutzerdefiniertes Präfix für den Vault-Pfad oder die Abfrage, abhängig vom Vault; wenn nicht angegeben, wird der am besten geeignete Standardwert verwendet. Beachten Sie, dass das Präfix mit / beginnen und enden muss.

Attention:
Die Konfigurationsparameter `Vault`, `VaultToken` (oder `VaultAppRoleID`/`VaultAppSecretID`), `VaultURL` und `VaultPrefix` werden auch für die Vault-Authentifizierung verwendet, wenn der Zabbix Server Secret-Vault-Makros verarbeitet (und der Zabbix Proxy, falls konfiguriert). Der Zabbix Server und Proxys öffnen keine Secret-Vault-Makros, die Datenbankanmeldedaten aus VaultDBPath enthalten.

Es wird dringend empfohlen, für verschiedene Proxys unterschiedliche Token zu verwenden.

Der Zabbix Server und der Zabbix Proxy lesen die Vault-bezogenen Konfigurationsparameter beim Start aus `zabbix_server.conf` und `zabbix_proxy.conf`. Zusätzlich lesen der Zabbix Server und der Zabbix Proxy die Umgebungsvariable `VAULT_TOKEN` einmalig beim Start aus und setzen sie zurück, damit sie über geforkte Skripte nicht verfügbar ist. Es ist ein Fehler, wenn sowohl `VaultToken` als auch `VAULT_TOKEN` einen Wert enthalten.

Beispiel

1. Geben Sie in `zabbix_server.conf` die folgenden Parameter an.

Mit VaultToken:

```
Vault=HashiCorp
VaultToken=hvs.CAESIIG_PILmULFYOsEyWHxkZ2mF2a8VPKNLE8eHqd4autYGGh4KHGh2cy5aeTY0NFNSaUp3ZnpWbDF1RUNjUkNTZEg
VaultURL=https://127.0.0.1:8200
VaultDBPath=database
VaultPrefix=/v1/secret/data/zabbix/
```

Oder mit VaultAppRoleID:

```
Vault=HashiCorp
VaultAppRoleID=6f1567f1-538d-7d70-e8e4-5d3e59471b2e
VaultAppSecretID=89dac8b1-f7d6-96a7-2ecf-f94d2de3d6f1
VaultURL=https://127.0.0.1:8200
VaultDBPath=database
VaultPrefix=/v1/secret/data/zabbix/
```

2. Führen Sie die folgenden CLI-Befehle aus, um das erforderliche Secret im Vault zu erstellen:

```
#### Aktivieren Sie den Mount-Punkt "secret/", falls er noch nicht aktiviert ist; beachten Sie, dass "kv-v
vault secrets enable -path=secret/ kv-v2

#### Legen Sie neue Secrets mit den Schlüsseln username und password unter dem Mount-Punkt "secret/" und d
vault kv put -mount=secret zabbix/database username=zabbix password=<password>

#### Prüfen Sie, ob das Secret erfolgreich hinzugefügt wurde.
vault kv get secret/zabbix/database

#### Testen Sie abschließend mit Curl; beachten Sie, dass "data" nach dem Mount-Punkt und "/v1" vor dem Mo
curl --header "X-Vault-Token: <VaultToken>" https://127.0.0.1:8200/v1/secret/data/zabbix/database
```

3. Dadurch ruft der Zabbix-Server die folgenden Anmeldedaten für die Datenbankauthentifizierung ab:

- Benutzername: zabbix
- Passwort: <password>

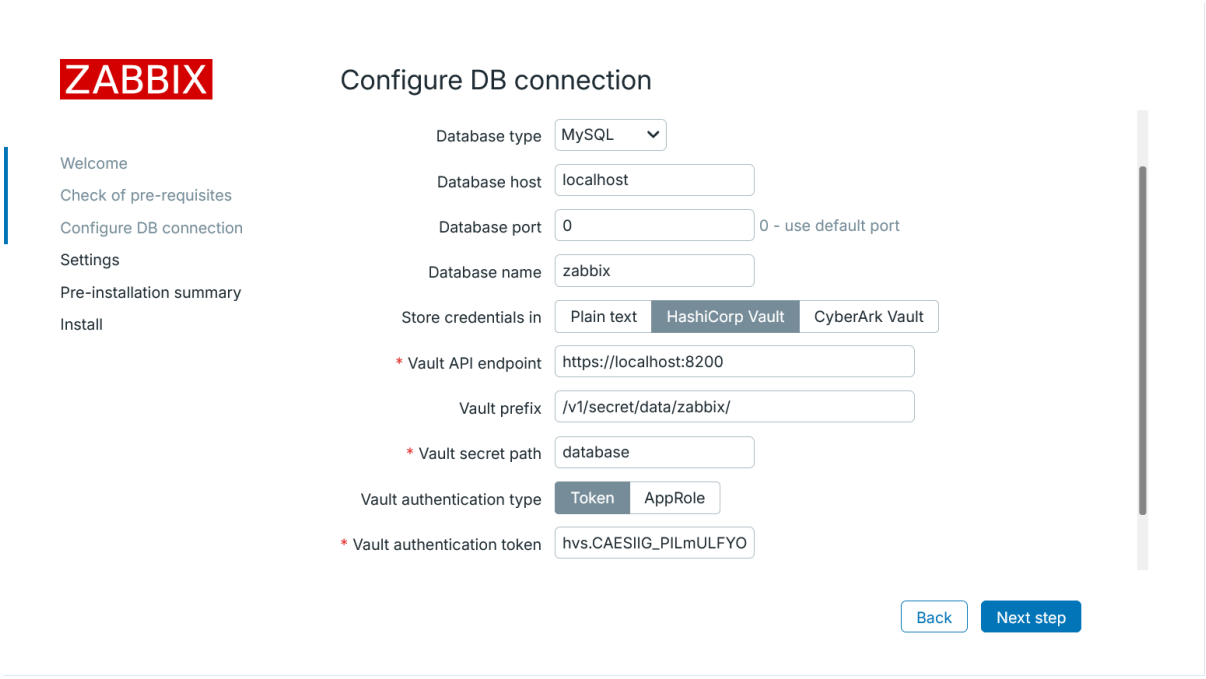
Frontend

Das Zabbix-Frontend kann so konfiguriert werden, dass es die Datenbankanmeldedaten entweder während der Installation des Frontends oder durch Aktualisieren der Frontend-Konfigurationsdatei (`zabbix.conf.php`) aus dem Vault abruft.

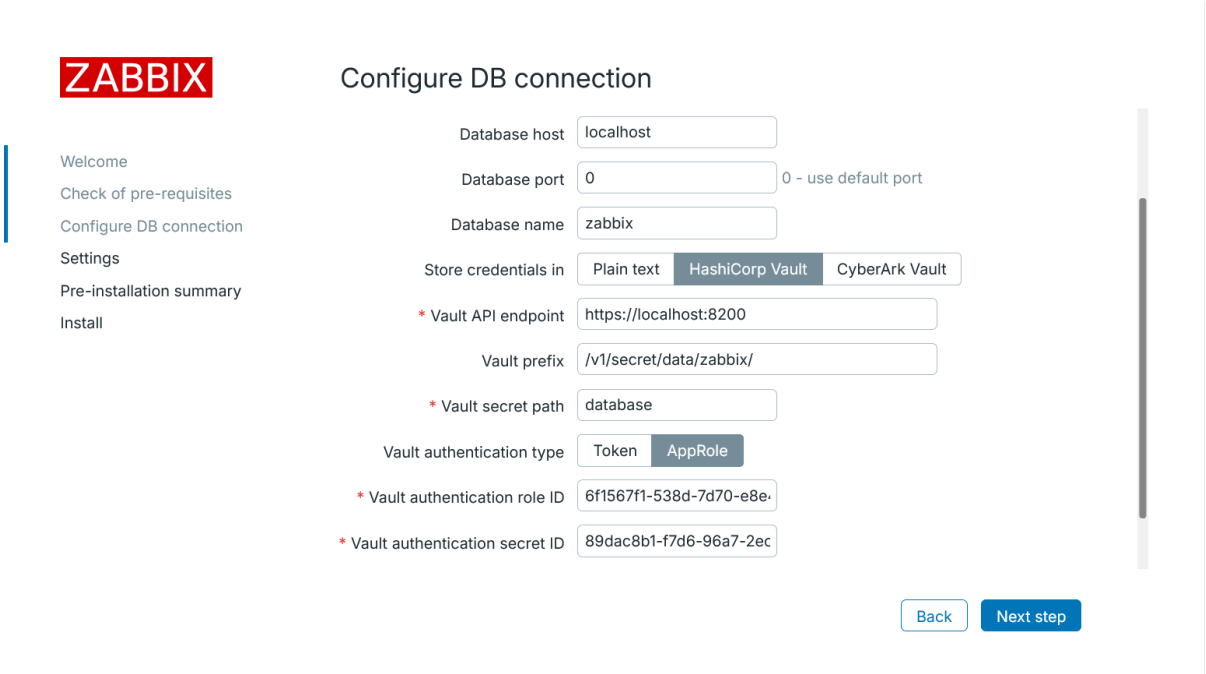
Attention:
Wenn sich die Vault-Anmeldedaten seit der vorherigen Frontend-Installation geändert haben, aktualisieren Sie die vorhandene Konfiguration.

Während der **Frontend-Installation** müssen die Verbindungsparameter im Schritt *Configure DB Connection* angegeben werden.

Konfigurationsbeispiel mit Token-Authentifizierung:



Konfigurationsbeispiel mit AppRole-Authentifizierung:



Parameter	Mandatory	Default value	Description
Store credentials in	no		Wählen Sie **HashiCorp Vault** aus.
Vault API endpoint	yes	https://localhost:8200	Geben Sie die URL für die Verbindung zum Vault im Format `scheme://host:port` an.
Vault prefix	no	/v1/secret/data/	Geben Sie ein benutzerdefiniertes Präfix für den Vault-Pfad oder die Abfrage an. Wenn kein Präfix angegeben wird, wird der Standardwert verwendet. Beispiel: `/v1/secret/data/zabbix/`
Vault secret path	yes		Ein Pfad zum Secret, aus dem die Datenbankanmeldedaten anhand der Schlüssel „password“ und „username“ abgerufen werden sollen. Beispiel: `database`
Vault authentication type	no		Wählen Sie den Vault-Authentifizierungstyp aus: **Token** – Token-Authentifizierung verwenden; **AppRole** – die AppRole-Authentifizierungsmethode verwenden.
Vault authentication token	yes		Geben Sie ein Authentifizierungstoken für den schreibgeschützten Zugriff auf den Secret-Pfad an. Informationen zum Erstellen von Token und Vault-Richtlinien finden Sie in der HashiCorp-Dokumentation. Dieses Feld ist nur verfügbar, wenn *Token* als Vault-Authentifizierungstyp ausgewählt ist.
Vault authentication role ID	yes		Geben Sie die Rollen-ID der AppRole an. Informationen zu AppRoles finden Sie in der HashiCorp-Dokumentation. Dieses Feld ist nur verfügbar, wenn *AppRole* als Vault-Authentifizierungstyp ausgewählt ist.
Vault authentication secret ID	yes		Geben Sie die für die AppRole ausgestellte Secret-ID an. Informationen zu AppRoles finden Sie in der HashiCorp-Dokumentation. Dieses Feld ist nur verfügbar, wenn *AppRole* als Vault-Authentifizierungstyp ausgewählt ist.

Abrufen von Benutzer-Makrowerten

Um HashiCorp Vault zum Speichern von *Vault secret*-Benutzermakrowerten zu verwenden, stellen Sie sicher, dass:

- Der Zabbix-Server/Proxy für die Arbeit mit HashiCorp Vault konfiguriert ist.
- Der Parameter *Vault provider* unter *Administration > Allgemein > Sonstiges* auf ”HashiCorp Vault” (Standard) gesetzt ist.



Note:
Der Zabbix-Server (und der Zabbix-Proxy, falls konfiguriert) benötigt Zugriff auf *Vault secret*-Makrowerte aus dem Vault. Das Zabbix-Frontend benötigt keinen solchen Zugriff.

Der Makrowert sollte einen Referenzpfad enthalten (als `path:key`, zum Beispiel `macros:password`). Das bei der Zabbix-Server/Proxy-Konfiguration angegebene Authentifizierungstoken (über den Parameter `VaultToken`) muss Lesezugriff auf diesen Pfad gewähren.

Weitere Informationen zur Verarbeitung von Makrowerten durch Zabbix finden Sie unter *Vault secret macros*.

Pfadsyntax

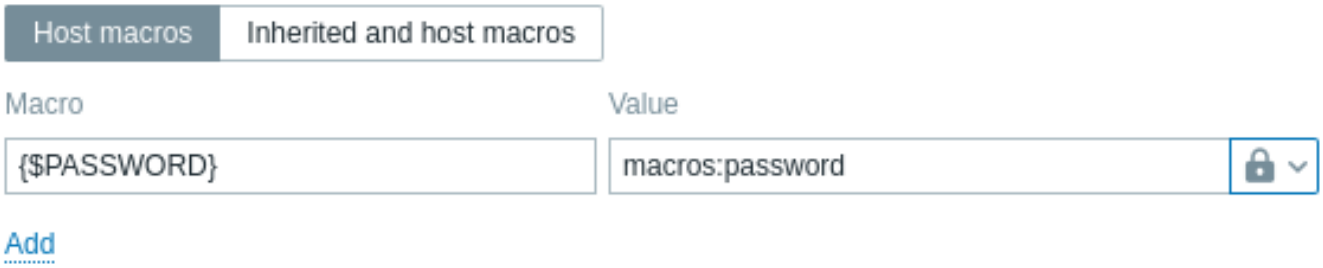
Die Symbole Schrägstrich (”/”) und Doppelpunkt (”:”) sind reserviert.

Ein Schrägstrich kann nur verwendet werden, um einen Mountpoint von einem Pfad zu trennen (z. B. *secret/zabbix*, wobei der Mountpoint ”secret” und der Pfad ”zabbix” ist). Bei Vault-Makros kann ein Doppelpunkt nur verwendet werden, um einen Pfad/Query von einem Schlüssel zu trennen.

Es ist möglich, die Symbole Schrägstrich und Doppelpunkt per URL-Encoding zu kodieren, wenn ein Mountpoint mit einem Namen erstellt werden soll, der durch einen Schrägstrich getrennt ist (z. B. kann *foo/bar/zabbix*, wobei der Mountpoint ”foo/bar” und der Pfad ”zabbix” ist, als ”foo%2Fbar/zabbix” kodiert werden) und wenn ein Mountpoint-Name oder Pfad einen Doppelpunkt enthalten muss.

Beispiel

1. Fügen Sie in Zabbix ein Benutzermakro {$PASSWORD} vom Typ ”Vault secret” mit dem Wert `macros:password` hinzu



2. Führen Sie die folgenden CLI-Befehle aus, um das erforderliche Secret im Vault zu erstellen:

```
#### Aktivieren Sie den Mount-Punkt "secret/", falls er noch nicht aktiviert ist; beachten Sie, dass "kv-v
vault secrets enable -path=secret/ kv-v2

#### Legen Sie ein neues Secret mit dem Schlüssel "password" unter dem Mount-Punkt "secret/" und dem Pfad
vault kv put -mount=secret zabbix/macros password=<password>

#### Prüfen Sie, ob das Secret erfolgreich hinzugefügt wurde.
vault kv get secret/zabbix/macros

#### Testen Sie abschließend mit Curl; beachten Sie, dass "data" nach dem Mount-Punkt und "/v1" vor dem Ma
curl --header "X-Vault-Token: <VaultToken>" https://127.0.0.1:8200/v1/secret/data/zabbix/macros
```

3. Dadurch löst Zabbix das Makro {$PASSWORD} in den Wert <password> auf

Vorhandene Konfiguration aktualisieren

So aktualisieren Sie eine vorhandene Konfiguration zum Abrufen von Geheimnissen aus einem HashiCorp Vault:

1. Aktualisieren Sie die Parameter der Konfigurationsdatei des Zabbix Servers oder Proxys wie im Abschnitt *Datenbankanmelde-informationen* beschrieben.

2. Aktualisieren Sie die Parameter der Frontend-Konfigurationsdatei von Zabbix:

Verwendung eines Vault-Tokens:

```
$DB['VAULT']                    = 'HashiCorp';
$DB['VAULT_URL']                = 'https://localhost:8200';
$DB['VAULT_DB_PATH']            = 'database';
$DB['VAULT_TOKEN']              = 'hvs.CAESIIG_PILmULFYOsEyWHxkZ2mF2a8VPKNLE8eHqd4autYGGh4KHGh2cy5aeTY0NFN
$DB['VAULT_PREFIX']             = '/v1/secret/data/zabbix/';
```

Oder mithilfe der Vault-AppRole-Methode:

```
$DB['VAULT']                    = 'HashiCorp';
$DB['VAULT_URL']                = 'https://localhost:8200';
$DB['VAULT_DB_PATH']            = 'database';
$DB['VAULT_APP_ROLE_ID']        = '6f1567f1-538d-7d70-e8e4-5d3e59471b2e';
$DB['VAULT_APP_SECRET_ID']      = '89dac8b1-f7d6-96a7-2ecf-f94d2de3d6f1';
$DB['VAULT_PREFIX']             = '/v1/secret/data/zabbix/';
```

3. Konfigurieren Sie bei Bedarf Benutzermakros wie im Abschnitt *Werte von Benutzermakros* beschrieben.

Informationen zum Aktualisieren einer vorhandenen Konfiguration zum Abrufen von Geheimnissen aus einem CyberArk Vault finden Sie unter *CyberArk-Konfiguration*.

Fehlerbehebung

1. In der Zabbix Server/Proxy-Protokolldatei:

```
237839:20260729:130916.425 Vault token renew started to fail: cannot login into HashiCorp vault with AppRo
```

Wahrscheinliche Ursache: HashiCorp Vault wurde (neu) gestartet, aber noch nicht entsiegelt (in der Standardkonfiguration erforderlich, bei konfiguriertem Auto Unseal nicht erforderlich).

2. In der Zabbix Server/Proxy-Protokolldatei:

```
235545:20260729:125617.944 Vault token renew started to fail: cannot login into HashiCorp vault with AppRo
```

oder

```
235545:20260729:125718.537 Vault token renew failed: cannot login into HashiCorp vault with AppRole method
```

Wahrscheinliche Ursache: VaultAppSecretID ist in HashiCorp Vault abgelaufen oder hat sein Nutzungslimit erreicht.

Google Chrome TLS-Zertifikatsvertrauen

Übersicht

Diese Seite enthält Zabbix-Einrichtungsschritte und Konfigurationsbeispiele für die Verwendung von Google Chrome mit Zabbix Frontend oder zabbix-web-service-Endpunkten, die durch ein selbstsigniertes Zertifikat oder eine private Zertifizierungsstelle abgesichert sind.

Diese Anweisungen setzen voraus, dass der Ziel-Webserver bereits für HTTPS konfiguriert ist. Zur Konfiguration von TLS für das Zabbix Frontend siehe Sichere Verbindung zum Frontend.

Konfiguration

Google Chrome unter Linux verwendet eine NSS-Zertifikatsdatenbank pro Benutzer für vertrauenswürdige Zertifikate. Damit Chrome einem selbstsignierten Zertifikat oder einer privaten Zertifizierungsstelle vertraut, fügen Sie die erforderlichen Zertifikate zur NSS-Datenbank des Benutzerkontos hinzu, unter dem Chrome ausgeführt wird.

1. Installieren Sie die erforderlichen Pakete.

Für Debian/Ubuntu:

```
sudo apt install ca-certificates libnss3-tools
```

Für RHEL-basierte Systeme:

```
sudo dnf install ca-certificates nss-tools
```

Oder:

```
sudo yum install ca-certificates nss-tools
```

2. Bereiten Sie die Zertifikatsdateien vor.

Für jedes benötigte Zertifikat: Erstellen Sie eine PEM-basierte `.crt`-Datei für das selbstsignierte Zertifikat oder separate PEM-basierte `.crt`-Dateien für das Root-Zertifizierungsstellenzertifikat und, falls erforderlich, für jedes Zwischenzertifikat.

3. Erstellen Sie das Verzeichnis der NSS-Datenbank.

Verwenden Sie das Home-Verzeichnis des Benutzers, unter dem Chrome ausgeführt wird.

Zum Beispiel:

```
sudo mkdir -p /var/lib/zabbix/.pki/nssdb
sudo -u zabbix certutil -N -d sql:/var/lib/zabbix/.pki/nssdb
sudo chown -R zabbix:zabbix /var/lib/zabbix/.pki/nssdb
```

Wenn Chrome auf dem Zielsystem einen anderen Speicherort für die NSS-Datenbank verwendet, nutzen Sie stattdessen dieses Verzeichnis.

Stellen Sie sicher, dass die Zertifikatsdateien für den Benutzer lesbar sind, unter dem Google Chrome ausgeführt wird. Kopieren Sie sie bei Bedarf an einen für diesen Benutzer zugänglichen Speicherort.

4. Importieren Sie die Zertifikate.

Bei Verwendung eines selbstsignierten Zertifikats:

```
sudo -u zabbix certutil -d sql:/var/lib/zabbix/.pki/nssdb \
  -A -t "P,," \
  -n "Zabbix selbstsigniertes Zertifikat" \
  -i /path/to/self-signed.crt
```

Bei Verwendung einer privaten Zertifizierungsstelle importieren Sie das Root-Zertifizierungsstellenzertifikat in die Datenbank:

```
sudo -u zabbix certutil -d sql:/var/lib/zabbix/.pki/nssdb \
  -A -t "C,," \
  -n "Zabbix Root-Zertifizierungsstellenzertifikat" \
  -i /path/to/root-ca.crt
```

Wenn Zwischenzertifikate verwendet werden, importieren Sie jedes einzelne separat:

```
sudo -u zabbix certutil -d sql:/var/lib/zabbix/.pki/nssdb \
  -A -t ",," \
  -n "Zabbix Zwischenzertifikat" \
  -i /path/to/intermediate-ca.crt
```

5. Überprüfen Sie die Zertifikatsdatenbank.

```
sudo -u zabbix certutil -d sql:/var/lib/zabbix/.pki/nssdb -L
```

6. Testen Sie Chrome.

Google Chrome verwendet die Zertifikatsdatenbank des Benutzerkontos, das es startet. Wenn das Benutzerkonto kein beschreibbares Home-Verzeichnis hat, konfigurieren Sie vor dem Test ein beschreibbares Home-Verzeichnis und XDG-Verzeichnisse.

Beispiel (ersetzen Sie `google-chrome` durch `chromium`, wenn Chromium verwendet wird.):

```
sudo -u zabbix env \
  HOME=/var/lib/zabbix-home \
  XDG_CONFIG_HOME=/var/lib/zabbix-home/.config \
  XDG_DATA_HOME=/var/lib/zabbix-home/.local/share \
  XDG_CACHE_HOME=/var/lib/zabbix-home/.cache \
  google-chrome --headless --disable-gpu \
  --user-data-dir=/var/lib/zabbix-home/chrome-profile \
  --dump-dom https://FQDN_OF_ZABBIX_SERVER
```

Wenn die Konfiguration korrekt ist, gibt der Befehl das HTML der angeforderten Seite zurück.

Wenn Chrome dem Zertifikat nicht vertraut, gibt der Befehl eine Zertifikatsfehlerseite zurück, die Meldungen wie `NET::ERR_CERT_AUTHORITY` enthält.

Fehlerbehebung

Fehler	Mögliche Ursache
NET::ERR_CERT_AUTHORITY_INVALID	Das Zertifikat wurde in das falsche Benutzerprofil importiert, das Zertifikat der Stammzertifizierungsstelle fehlt oder ein Zwischenzertifikat fehlt.
NET::ERR_CERT_COMMON_NAME_INVALID	Das Zertifikat stimmt nicht mit dem Hostnamen überein, der in der Browser-URL verwendet wird.
NET::ERR_CERT_DATE_INVALID	Das Zertifikat ist abgelaufen, noch nicht gültig oder die Systemuhr ist falsch.
Chrome zeigt nach dem Import weiterhin eine Zertifikatswarnung an	Chrome verwendet für den aktuellen Benutzer eine andere NSS-Datenbank, oder die Vertrauenskennzeichen des Zertifikats sind falsch.

Sicherheitsbewährte Praktiken

Übersicht Dieser Abschnitt enthält Best Practices für die sichere Einrichtung von Zabbix.

Die in diesem Abschnitt beschriebenen Vorgehensweisen sind für die Funktionsweise von Zabbix nicht erforderlich, werden jedoch für eine bessere Systemsicherheit empfohlen.

Best Practices zur Absicherung der mobilen Zabbix-App und des Zabbix-Bridge-Adapters finden Sie unter Sicherer mobiler Zugriff.

UTF-8-Kodierung UTF-8 ist die einzige von Zabbix unterstützte Kodierung. Es ist bekannt, dass sie ohne Sicherheitslücken funktioniert. Benutzer sollten sich bewusst sein, dass bei der Verwendung einiger anderer Kodierungen bekannte Sicherheitsprobleme auftreten.

Windows-Installer-Pfade Bei der Verwendung von Windows-Installern wird empfohlen, die vom Installer bereitgestellten Standardpfade zu verwenden. Die Verwendung benutzerdefinierter Pfade ohne die entsprechenden Berechtigungen könnte die Sicherheit der Installation beeinträchtigen.

Makros in benutzerdefinierten globalen Skripten Zur Erhöhung der Sicherheit wird empfohlen, Makrofunktionen anstelle von einfachen Makros in benutzerdefinierten globalen Skripten zu verwenden, da Makros nicht automatisch maskiert werden.

Zabbix-Sicherheitsmeldungen und CVE-Datenbank Siehe *Zabbix-Sicherheitsmeldungen und CVE-Datenbank*.

HTML-E-Mail-Vorlagen Beim Erstellen oder Bearbeiten von Nachrichtenvorlagen, die für HTML-E-Mails verwendet werden, müssen Sie jedes Makro immer mit der Makrofunktion `htmlencode()` umschließen. Zum Beispiel:

```
<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
```

Die Verwendung von `htmlencode()` stellt sicher, dass alle HTML-Zeichen in Makrowerten maskiert werden, und verhindert das Einschleusen von HTML in Benachrichtigungen (beispielsweise, dass ein Angreifer einen schädlichen Phishing-Link in eine Benachrichtigung einfügt).

Hinweis: Die von Zabbix bereitgestellten Standard-HTML-E-Mail-Nachrichten wenden `htmlencode()` bereits auf Makros an. Diese Anleitung gilt beim Bearbeiten vorhandener Vorlagen oder beim Erstellen neuer Vorlagen. Überprüfen Sie, dass Makros codiert sind, bevor Sie eine Vorlage zum Senden von HTML-E-Mails verwenden.

Weboberfläche

Übersicht Für einen einfachen Zugriff auf Zabbix von überall und von jeder Plattform aus steht eine webbasierte Oberfläche zur Verfügung.

> **Note:**
> Wenn Sie mehr als eine Frontend-Instanz verwenden, stellen Sie sicher, dass die Gebietsschemata und Bibliotheken (LDAP, SAML usw.) für alle Frontends identisch installiert und konfiguriert sind.

Frontend-Hilfe In den Bereichen und Formularen des Zabbix-Frontends befindet sich oben rechts ein Hilfe-Link , der direkt zu den entsprechenden Abschnitten der Dokumentation führt.

Bei Formularfeldern mit einem Hilfe-Symbol wird der entsprechende Hinweis angezeigt, wenn Sie den Mauszeiger über das Symbol bewegen.

Kiosk-Modus Klicken Sie in Frontend-Bereichen mit der Schaltfläche *Kiosk-Modus* (in der oberen rechten Ecke) auf die Schaltfläche, um den Bereich in den Kiosk-Modus zu wechseln. Im Kiosk-Modus wird nur der Inhalt des Bereichs angezeigt.

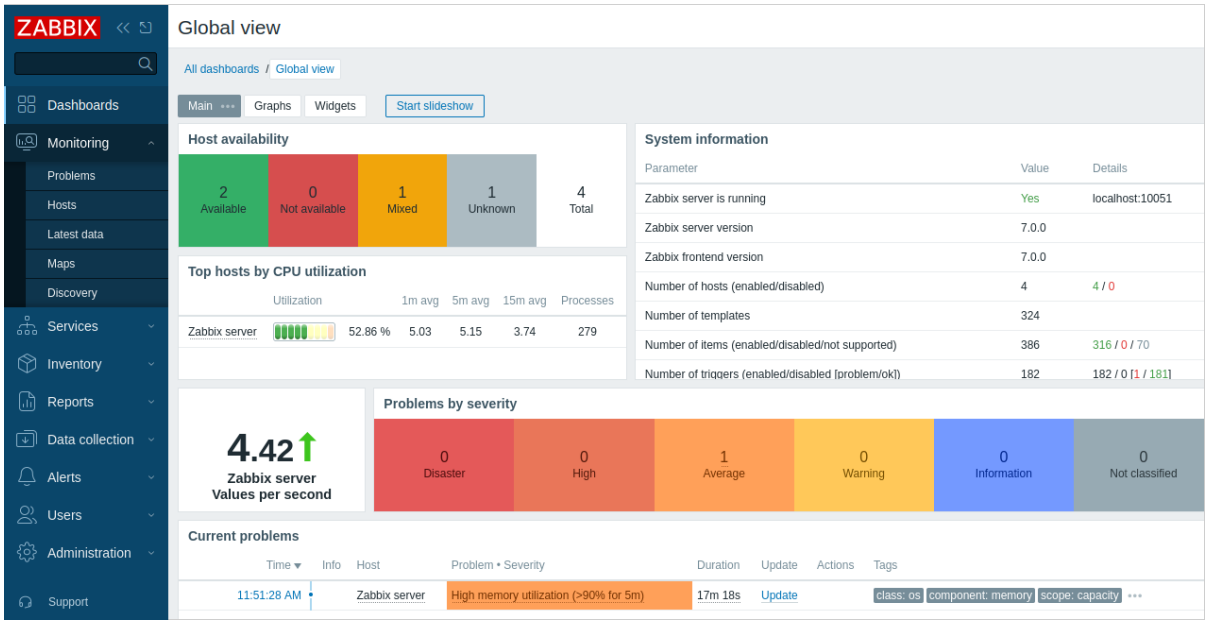
Um den Kiosk-Modus zu verlassen, bewegen Sie den Mauszeiger über den Inhalt und klicken Sie auf die Schaltfläche *Normale Ansicht*, die in der oberen rechten Ecke angezeigt wird.

Menü

Übersicht

Ein vertikales Menü in einer Seitenleiste bietet Zugriff auf verschiedene Bereiche des Zabbix Frontend.

Im Standard-Theme ist das Menü dunkelblau.

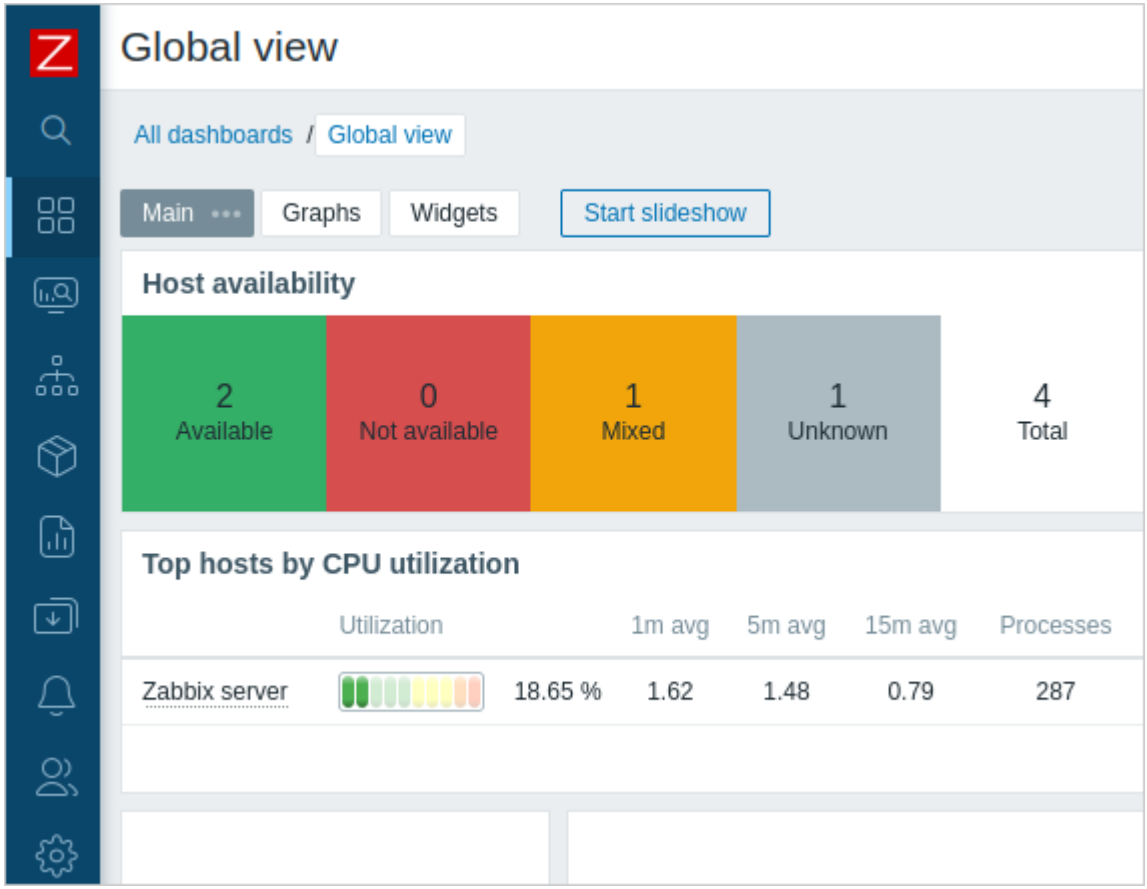


Arbeiten mit dem Menü

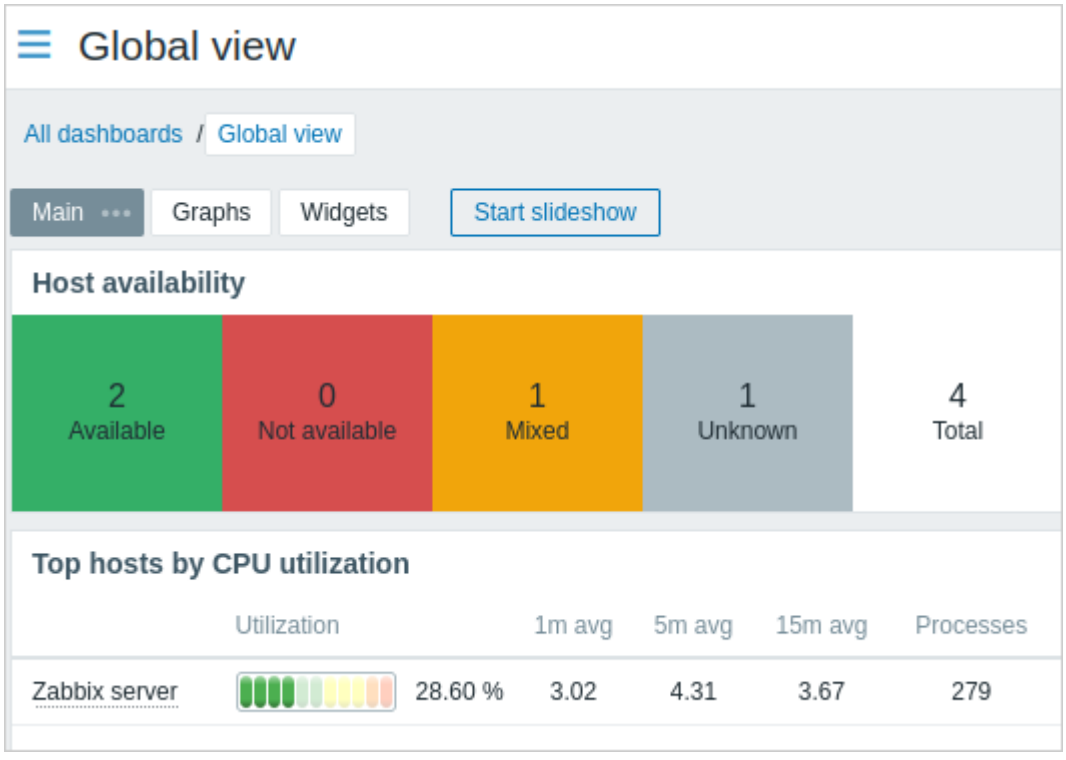
Ein Feld für die globale Suche befindet sich unter dem Zabbix-Logo.

Das Menü kann eingeklappt oder vollständig ausgeblendet werden:

- Zum Einklappen klicken Sie auf neben dem Zabbix-Logo. Im eingeklappten Menü sind nur die Symbole sichtbar.



- Zum Ausblenden klicken Sie auf  neben dem Zabbix-Logo. Im ausgeblendeten Menü ist alles verborgen.



Eingeklapptes Menü

Wenn das Menü auf die Anzeige von Symbolen beschränkt ist, wird das vollständige Menü wieder eingeblendet, sobald der Mauszeiger darauf platziert wird. Beachten Sie, dass es über dem Seiteninhalt eingeblendet wird. Um den Seiteninhalt nach rechts zu verschieben, müssen Sie auf die Schaltfläche zum Ausklappen klicken. Wird der Mauszeiger wieder außerhalb des vollständigen Menüs platziert, wird das Menü nach zwei Sekunden erneut eingeklappt.

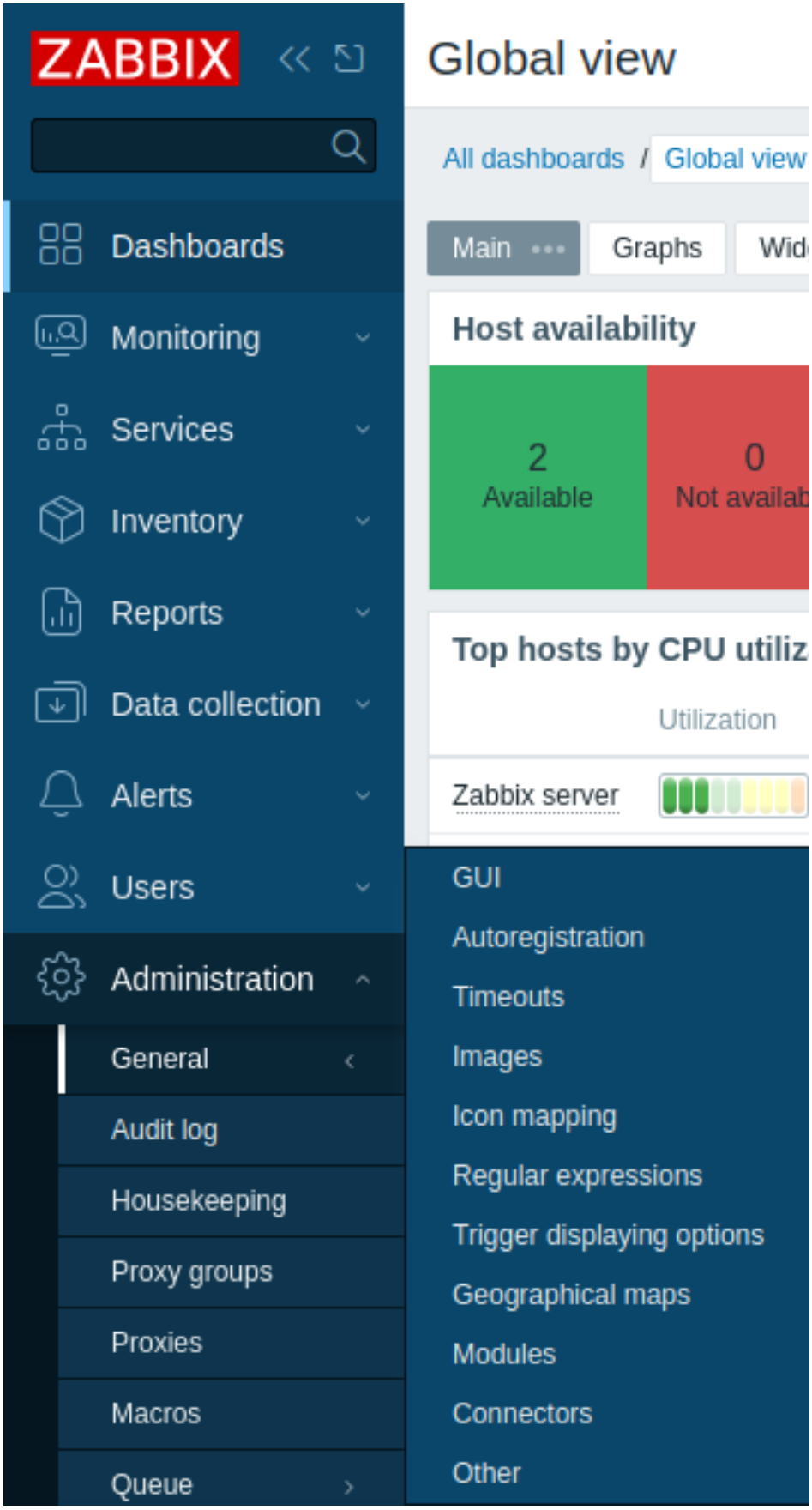
Sie können ein eingeklapptes Menü auch vollständig einblenden, indem Sie die Taste Tab drücken. Durch wiederholtes Drücken der Taste Tab können Sie das nächste Menüelement fokussieren.

Versteckte Menü

Auch wenn das Menü vollständig ausgeblendet ist, ist ein vollständiges Menü nur einen Mausklick entfernt, indem Sie auf das Burger-Symbol  klicken. Beachten Sie, dass es über dem Seiteninhalt wieder eingeblendet wird; um den Seiteninhalt nach rechts zu verschieben, müssen Sie das Menü einblenden, indem Sie auf die Schaltfläche zum Anzeigen der Seitenleiste klicken.

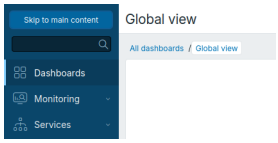
Menüebenen

Es gibt bis zu drei Ebenen im Menü.



Barrierefreiheit: Navigation überspringen

Der Link *Zum Hauptinhalt springen* ermöglicht es Benutzern von Tastatur und Bildschirmleseprogrammen, die wiederholte Navigation in der Seitenleiste zu überspringen und direkt zum Hauptinhalt zu wechseln.



Er bleibt bei der normalen Nutzung ausgeblendet und erscheint nur bei Fokus, beispielsweise wenn nach dem Laden der Seite die Taste Tab gedrückt wird oder wenn er mit einem Bildschirmleseprogramm aufgerufen wird.

Kontextmenüs

Zusätzlich zum Hauptmenü bietet Zabbix Kontextmenüs für Host, Datenpunkt und Ereignis für den schnellen Zugriff auf häufig verwendete Objekte wie die neuesten Werte, das einfache Diagramm, das Konfigurationsformular, zugehörige Skripte oder externe Links.

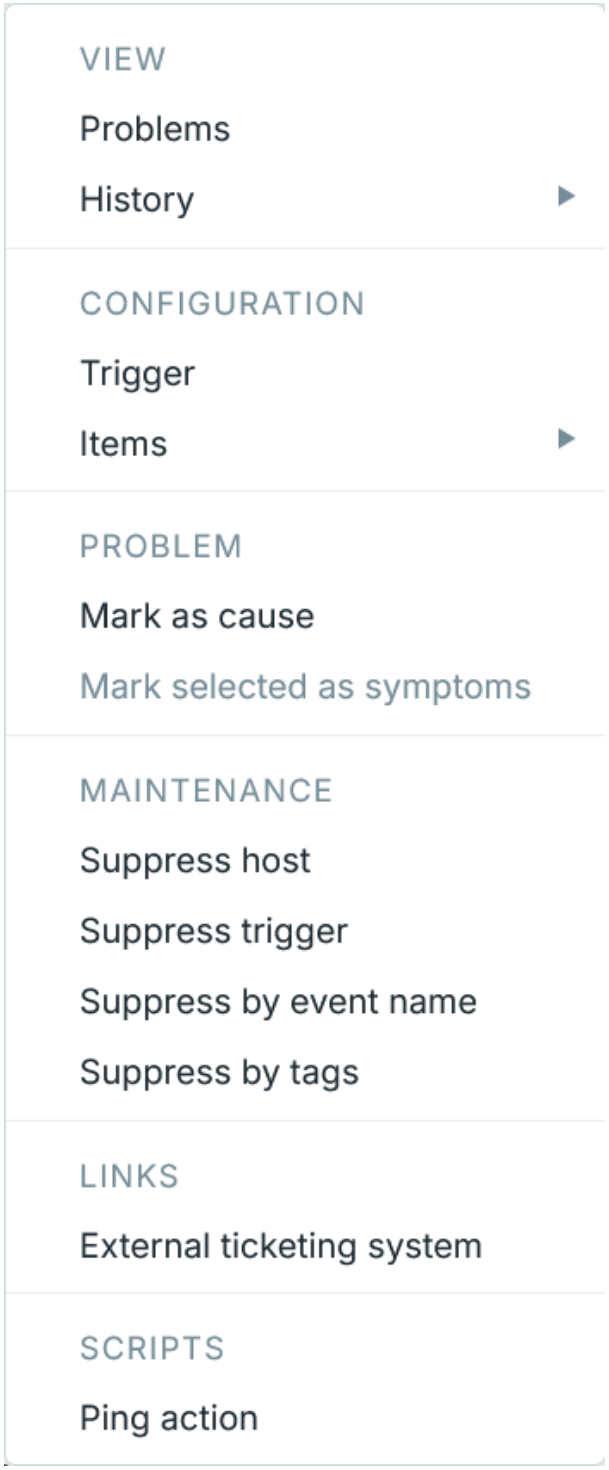
Die Kontextmenüs sind zugänglich, indem Sie an unterstützten Stellen auf den Host-, Datenpunkt- oder Problem-/Auslöser-Namen klicken.

Ereignismenü

Übersicht

Das Ereignismenü enthält Verknüpfungen zu Aktionen oder Frontend-Bereichen, die für ein Ereignis häufig benötigt werden.

Das Ereignismenü kann durch Klicken auf den Ereignisnamen geöffnet werden.



Inhalt

Das Ereigniskontextmenü enthält mehrere Abschnitte: *Ansicht*, *Aktionen*, *Konfiguration*, *Problem*, *Wartung*, *Links* und *Skripte*. Für nicht konfigurierte Entitäten sind die Links deaktiviert und werden grau angezeigt. Die Abschnitte *Skripte* und *Links* werden angezeigt, wenn die entsprechenden Entitäten konfiguriert sind.

Der Abschnitt *Ansicht* enthält Links zu:

- **Probleme** – öffnet die Liste der ungelösten Probleme des zugrunde liegenden Auslösers;
- **Verlauf** – führt zum Diagramm bzw. zum Datenpunktverlauf unter *Neueste Daten* für den bzw. die zugrunde liegenden Datenpunkte. Wenn ein Auslöser mehrere Datenpunkte verwendet, sind für jeden Datenpunkt Links verfügbar.

Der Abschnitt *Aktionen* ist nur in Widgets vom Typ *Auslöserübersicht* verfügbar. Er enthält einen Link zu:

- **Problem aktualisieren** – öffnet die Ansicht Problem aktualisieren.

Der Abschnitt *Konfiguration* enthält Links zur Konfiguration von:

- **Auslöser**, der das Problem ausgelöst hat;
- **Datenpunkten**, die im Ausdruck des Auslösers verwendet werden.

Note:
Beachten Sie, dass der Abschnitt *Konfiguration* nur für Benutzer mit den Rollen „Admin“ und „Super Admin“ verfügbar ist.

Der Abschnitt *Problem* enthält folgende Optionen:

- **Als Ursache markieren** – markiert das Problem als Ursache;
- **Ausgewählte als Symptome markieren** – markiert die ausgewählten Probleme als Symptome dieses Problems.

Der Abschnitt *Wartung* enthält Optionen zum Starten einer Ad-hoc-Wartung:

- **Host unterdrücken** – startet einen Wartungszeitraum für den Host;
- **Auslöser unterdrücken** – startet einen Wartungszeitraum für den Auslöser;
- **Nach Ereignisnamen unterdrücken** – startet einen Wartungszeitraum für den Ereignisnamen;
- **Nach Tags unterdrücken** – startet einen Wartungszeitraum für Ereignisse mit den entsprechenden Tags.

Wenn Sie die entsprechende Option auswählen, wird ein Konfigurationsformular für den Wartungszeitraum geöffnet.

Ein einmaliger Wartungszeitraum von einer Stunde (konfigurierbar in den Benutzereinstellungen und im Profil), der ab jetzt beginnt, ist vorausgewählt. Der entsprechende Host, Auslöser, Ereignisname bzw. die entsprechenden Tags sind ebenfalls vorausgewählt. Der Name der Wartung wird im Format `Ad-hoc: <event name>` vorausgefüllt.

Der Abschnitt *Links* enthält Links zu:

- einer konfigurierten Auslöser-URL;
- benutzerdefinierten Links, die in Globalen Skripten konfiguriert sind (mit dem Geltungsbereich „Manuelle Ereignisaktion“ und dem Typ „URL“);
- einem konfigurierten externen Ticket für das Problem (siehe die Option *Ereignismenüeintrag einschließen* beim Konfigurieren von webhooks).

Der Abschnitt *Skripte* enthält Links zum Ausführen eines globalen Skripts (mit dem Geltungsbereich *Manuelle Ereignisaktion*). Diese Funktion kann nützlich sein, um Skripte auszuführen, die zur Verwaltung von Problemtickets in externen Systemen verwendet werden.

Unterstützte Orte

Das Kontextmenü für Ereignisse ist verfügbar, wenn Sie in verschiedenen Bereichen des Frontends auf den Namen eines Problems oder Ereignisses klicken, zum Beispiel:

- Dashboards Widgets, wie das *Problems*-Widget, das *Trigger overview*-Widget usw.
- *Monitoring* > *Problems*
- *Monitoring* > *Problems* > Ereignisdetails
- *Reports* > *Top 100 triggers* (globale Skripte und der Zugriff auf externe Tickets werden an diesem Ort nicht unterstützt)

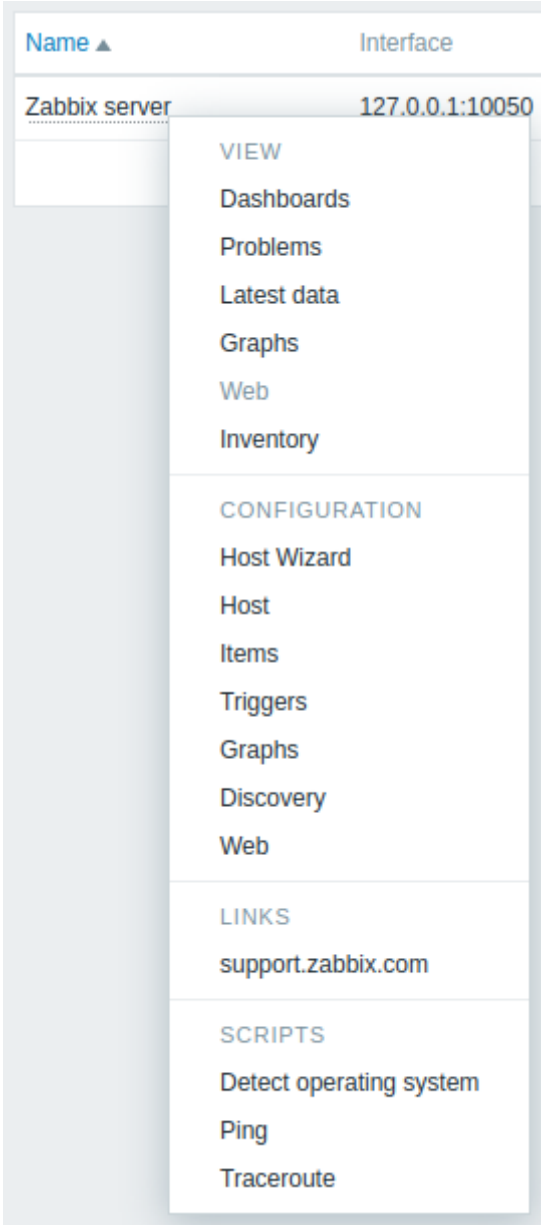
Host-Menü

Übersicht

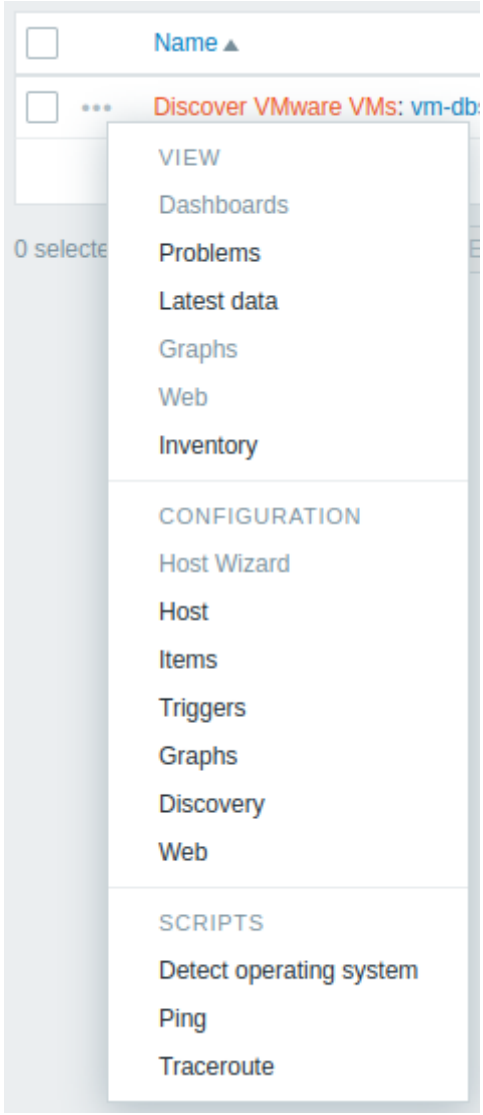
Das Host-Menü enthält Verknüpfungen zu Aktionen oder Frontend-Bereichen, die für einen Host häufig benötigt werden.

Das Host-Menü kann geöffnet werden, indem je nach Frontend-Bereich auf den Hostnamen oder das Drei-Punkte-Symbol geklickt wird, zum Beispiel:

Monitoring > *Latest data*



Data collection > *Hosts*



Inhalt

Das Kontextmenü des Hosts enthält vier Bereiche: *Ansicht*, *Konfiguration*, *Links* und *Skripte*. Für Entitäten, die nicht konfiguriert sind, sind die Links deaktiviert und werden grau angezeigt. Die Bereiche *Skripte* und *Links* werden angezeigt, wenn die entsprechenden Entitäten konfiguriert sind.

Der Bereich *Ansicht* enthält Links zu:

- **Dashboards** – öffnet Widgets und Graphen.
- **Probleme** – öffnet den Bereich *Probleme* mit der Liste der ungelösten Probleme des zugrunde liegenden Auslösers.
- **Neueste Daten** – öffnet den Bereich *Neueste Daten* mit der Liste aller neuesten Daten des aktuellen Hosts.
- **Graphen** – öffnet einfache Graphen des aktuellen Hosts.
- **Web** – öffnet den Link zu den konfigurierten Web-Szenarien.
- **Inventar** – öffnet den Link zum Inventar des aktuellen Hosts.

Der Bereich *Konfiguration* enthält Links zu:

- **Host-Assistent** – öffnet den Host-Assistenten für den aktuellen Host (für erkannte Hosts deaktiviert).
- **Host** – Konfigurationsformular des aktuellen Hosts.
- **Datenpunkte** – Liste der Datenpunkte des aktuellen Hosts.
- **Auslöser** – Liste der Auslöser des aktuellen Hosts.
- **Graphen** – einfache Graphen des aktuellen Hosts.
- **Erkennung** – Liste der Low-Level-Erkennungsregeln des aktuellen Hosts.

- **Web** – Liste der Web-Szenarien des aktuellen Hosts.

> **Note:**
> Beachten Sie, dass der Konfigurationsbereich nur für Benutzer mit den Rollen „Admin“ und „Super-Admin“ verfügbar ist.

Der Bereich *Links* enthält Links zu:

- einer konfigurierten Auslöser-URL.
- benutzerdefinierten Links, die in Globalen Skripten konfiguriert sind (mit dem Geltungsbereich *Manuelle Host-Aktion* und dem Typ „URL“).

Der Bereich *Skripte* ermöglicht die Ausführung von globalen Skripten, die für den aktuellen Host konfiguriert sind. Damit diese Skripte im Host-Menü verfügbar sind, muss ihr Geltungsbereich als *Manuelle Host-Aktion* definiert sein.

Unterstützte Orte

Das Host-Menü ist zugänglich, indem Sie in verschiedenen Frontend-Bereichen auf einen Hostnamen klicken, zum Beispiel:

- Dashboards Widgets, wie *Problems*, *Top items*, *Trigger overview* usw.
- *Monitoring* > *Problems*
- *Monitoring* > *Problems* > Ereignisdetails
- *Monitoring* > *Hosts*
- *Monitoring* > *Hosts* > *Web Monitoring*
- *Monitoring* > *Latest data*
- *Monitoring* > *Maps*
- *Inventory* > *Hosts*
- *Reports* > *Top 100 triggers*

Das Host-Menü ist zugänglich, indem Sie in Data collection > Hosts auf das Drei-Punkte-Symbol klicken.

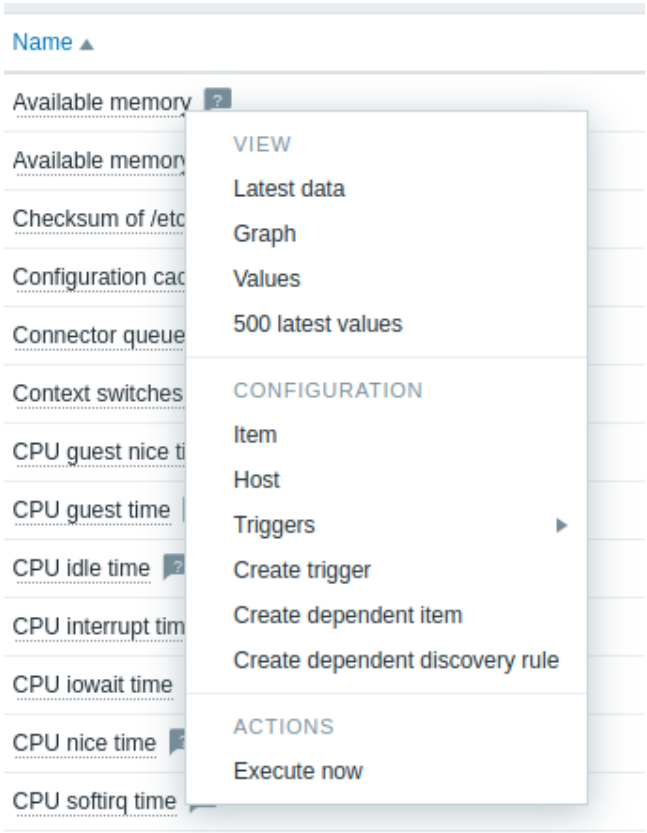
Datenpunkt-Menü

Übersicht

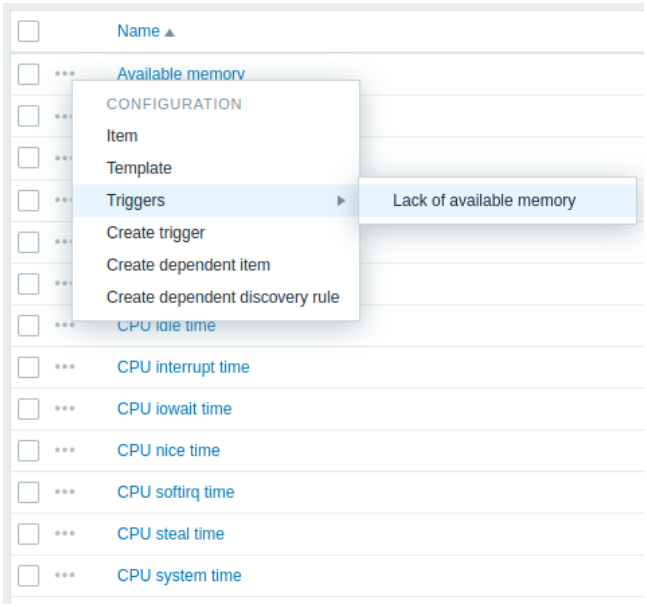
Das Datenpunkt-Menü enthält Verknüpfungen zu Aktionen oder Frontend-Bereichen, die für einen Datenpunkt häufig benötigt werden.

Das Datenpunkt-Menü kann je nach Frontend-Bereich durch Klicken auf den Datenpunktnamen oder das Drei-Punkte-Symbol geöffnet werden, zum Beispiel:

Monitoring > *Neueste Daten*



Datenerfassung > *Vorlagen* > *Datenpunkte*



Inhalt

Das Datenpunktmenü enthält drei Abschnitte: *Ansicht*, *Konfiguration* und *Aktionen*.

Der Abschnitt *Ansicht* enthält die folgenden Optionen:

- **Neueste Daten** – öffnet den Abschnitt *Neueste Daten*, gefiltert nach dem aktuellen Host und Datenpunkt;
- **Graph** – öffnet einen einfachen Graphen des aktuellen Datenpunkts;
- **Werte** – öffnet die Liste aller Werte, die in den letzten 60 Minuten für den aktuellen Datenpunkt empfangen wurden;
- **500 neueste Werte** – öffnet die Liste der 500 neuesten Werte für den aktuellen Datenpunkt.

Der Abschnitt *Konfiguration* (nur für Benutzer des Typs *Admin* und *Super-Admin* verfügbar) enthält die folgenden Optionen:

- **Datenpunkt** – öffnet das Datenpunkt-Konfigurationsformular des aktuellen Datenpunkts;
- **Vorlage** – öffnet das Vorlagen-Konfigurationsformular der Vorlage, zu der der Datenpunkt gehört (nur verfügbar, wenn das Datenpunktmenü über *Datensammlung* > *Vorlagen* > *Datenpunkte* aufgerufen wird);
- **Host** – öffnet das Host-Konfigurationsformular des Hosts, zu dem der Datenpunkt gehört;
- **Auslöser** – öffnet beim Überfahren mit dem Mauszeiger eine Liste der Auslöser des Datenpunkts, sofern vorhanden; durch Klicken auf einen Auslöser wird dessen Auslöser-Konfigurationsformular geöffnet;
- **Auslöser erstellen** – öffnet das Auslöser-Konfigurationsformular, um einen Auslöser für diesen Datenpunkt zu erstellen;
- **Abhängigen Datenpunkt erstellen** – öffnet das Datenpunkt-Konfigurationsformular, um einen abhängigen Datenpunkt zu erstellen, wobei der aktuelle Datenpunkt als Master-Datenpunkt verwendet wird;
- **Abhängige Discovery-Regel erstellen** – öffnet das Konfigurationsformular für Discovery-Regeln, um eine abhängige Discovery-Regel zu erstellen, wobei der aktuelle Datenpunkt als Master-Datenpunkt verwendet wird.

Der Abschnitt *Aktionen* enthält die folgende Option:

- **Jetzt ausführen** – führt sofort eine Prüfung auf einen neuen Datenpunktwert aus.

Unterstützte Orte

Das Datenpunkt-Menü ist zugänglich, indem Sie in verschiedenen Frontend-Bereichen auf einen Datenpunktnamen klicken, zum Beispiel:

- *Monitoring* > *Neueste Daten*
- *Datenerfassung* > *Hosts* > *Datenpunkte*
- *Datenerfassung* > *Hosts* > *Erkennungsregeln* > *Datenpunktvorlagen*

Das Datenpunkt-Menü ist zugänglich, indem Sie in dem Dashboard-Widget *Top-Datenpunkte* auf einen Datenpunktwert klicken.

Frontend-Bereiche

Menüstruktur Das Zabbix-Frontend weist folgende Struktur auf:

- Dashboards
- Überwachung
  - Probleme
  - Hosts
  - Neueste Daten
  - Karten
  - Discovery
- APM
  - Traces
  - Metriken
  - Protokolle
- Services
  - Services
  - SLA
  - SLA-Bericht
- Inventar
  - Übersicht
  - Hosts
- Berichte
  - Systeminformationen
  - Geplante Berichte
  - Verfügbarkeitsbericht
  - Top 100 Auslöser
  - Auditprotokoll
  - Aktionsprotokoll
  - Benachrichtigungen
- Datensammlung
  - Vorlagengruppen
  - Hostgruppen
  - Vorlagen
  - Hosts
  - Wartung
  - Ereigniskorrelation
  - Discovery
- Alarme
  - Aktionen
    - Auslöseraktionen
    - Serviceaktionen
    - Discovery-Aktionen
    - Aktionen zur automatischen Registrierung
    - Interne Aktionen
  - Medientypen
  - Skripte
- Benutzer
  - Benutzergruppen
  - Benutzerrollen
  - Benutzer
  - API-Tokens
  - Authentifizierung
  - Geräte
- Administration
  - Allgemein
    - GUI
    - Automatische Registrierung
    - Zeitüberschreitungen

    - Bilder
    - Symbolzuordnung
    - Reguläre Ausdrücke
    - Optionen zur Anzeige von Auslösern
    - Geografische Karten
    - Module
    - Konnektoren
    - Sonstiges
- Auditprotokoll
- Housekeeping
- Proxy-Gruppen
- Proxys
- Makros
- Warteschlange
    - Warteschlangenübersicht
    - Warteschlangenübersicht nach Proxy
    - Warteschlangendetails

Dashboards

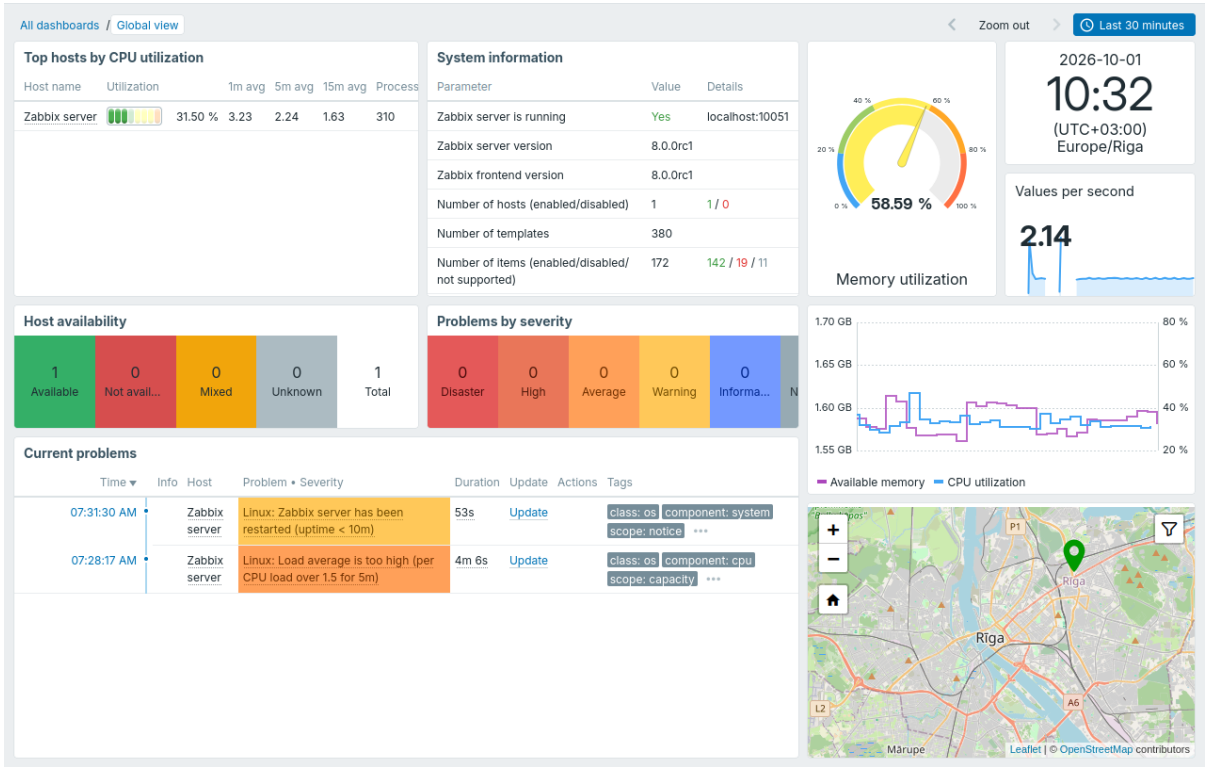
Dashboards bieten Ihnen einen Überblick über Ihre überwachte Umgebung. Sie können sie erstellen und an Ihre Anforderungen anpassen.

Die Bausteine sind Widgets: ein Diagramm, eine Karte, eine Liste aktueller Probleme usw., die jeweils dazu dienen, einen bestimmten Informationstyp anzuzeigen.

Widgets können auf mehreren Dashboard-Seiten organisiert werden. Solche Seiten können als Diashow angezeigt werden. Sie können auch zu jeder Seite navigieren, indem Sie auf deren Registerkarte klicken.

Sie können mehrere Dashboards erstellen, allerdings wird immer nur eines gleichzeitig angezeigt.

Zabbix wird mit einigen Standard-Dashboards ausgeliefert. Eines davon ist das Dashboard *Globale Ansicht*, das den Zustand und Status Ihrer überwachten Hosts einschließlich des Zabbix-Servers anzeigt:

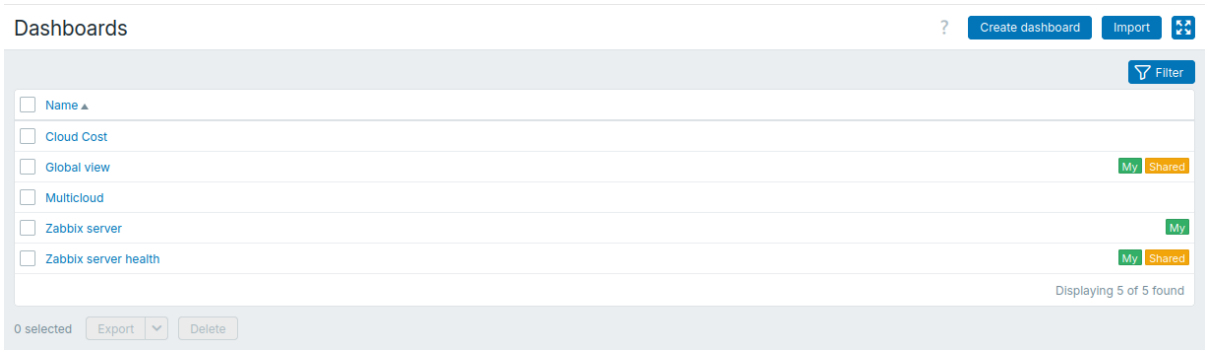


Weitere Dashboards finden Sie im Dashboards-Repository. Diese Dashboards sind für die Verwendung mit vorhandenen Zabbix-Vorlagen konzipiert und können heruntergeladen und in Ihre Zabbix-Umgebung importiert werden. Eine detaillierte Beschreibung jedes Dashboards einschließlich der Voraussetzungen finden Sie in der `README`-Datei des jeweiligen Dashboards.

Dashboard-Liste

Um die Liste der Dashboards zu öffnen, haben Sie folgende Möglichkeiten:

- Klicken Sie im Menü auf *Dashboards*.
- Klicken Sie beim Anzeigen eines Dashboards auf *Alle Dashboards*, um die Dashboard-Liste zu öffnen.



Der Zugriff auf das Menü *Dashboards* hängt von der Benutzerrolle und der Berechtigung *Dashboard* unter *Zugriff auf UI-Elemente* ab.

Hier können Sie: - Dashboards erstellen. - Dashboards löschen. Sie können ein Dashboard löschen, indem Sie das entsprechende Kontrollkästchen auswählen und auf die Schaltfläche *Löschen* klicken. - Dashboards importieren. - Dashboards exportieren.

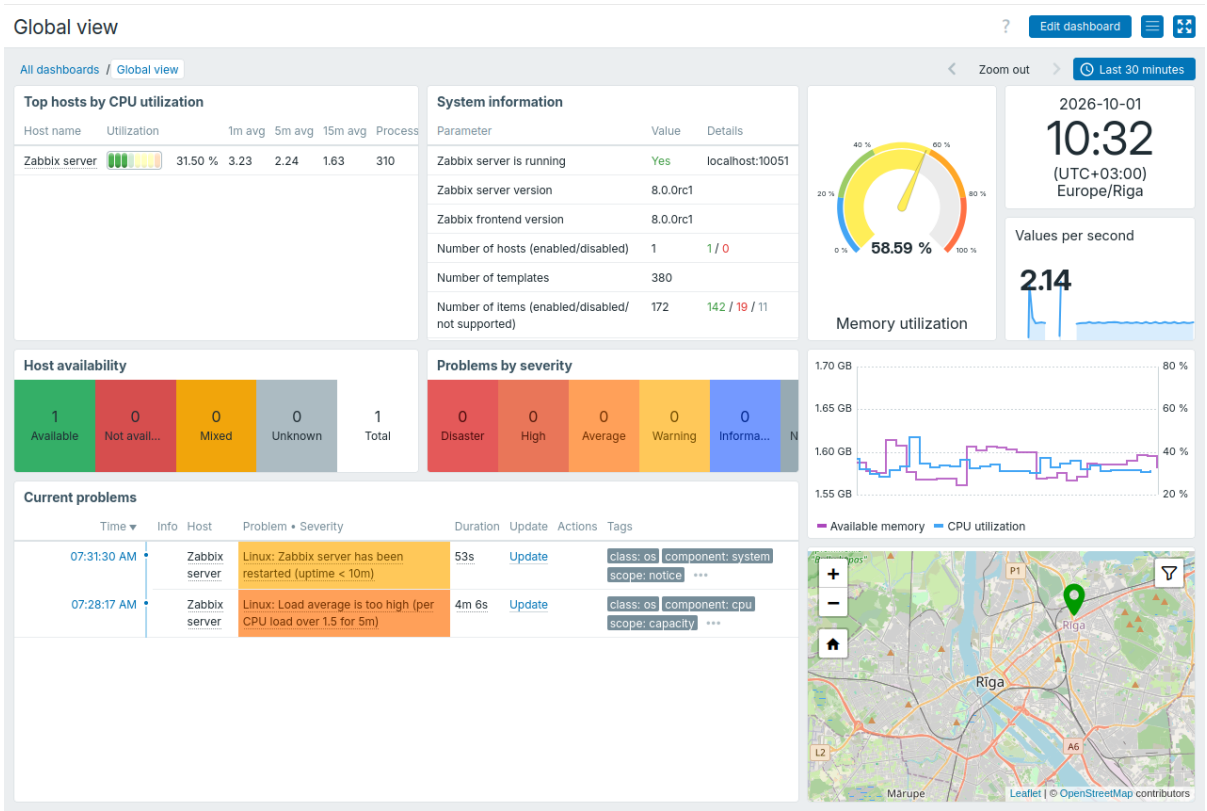
Dashboards verfügen über Tags, die ihre Eigentümerschaft und Freigabeeinstellungen angeben:

- *Meine* – Das Dashboard gehört Ihnen und ist nur für Sie sichtbar.
- *Meine + Freigegeben* – Das Dashboard gehört Ihnen, ist aber öffentlich oder für bestimmte Benutzer oder Benutzergruppen freigegeben.
- Dashboards ohne Tags gehören anderen Benutzern, sind aber öffentlich oder für Sie freigegeben.

Sie können die Liste nach Dashboard-Name und Eigentümer filtern.

Dashboard anzeigen

Klicken Sie in der Dashboard-Liste auf den Namen eines Dashboards, um das Dashboard anzuzeigen:



Die Zabbix-Benutzeroberfläche merkt sich das zuletzt geöffnete Dashboard für die nächste Sitzung.

Note:
Die Daten, die Sie auf einem Dashboard sehen können, hängen von Ihrem Zugriff auf Hosts ab.

Für Widgets, die Diagramme anzeigen, legt der Zeitraumauswahlschalter oberhalb der Widgets den Zeitbereich fest. Er zeigt den aktuellen Zeitbereich als Bezeichnung an. Klicken Sie auf die Bezeichnung, um den Auswahlschalter zu öffnen oder zu schließen.

Von hier aus können Sie Dashboards wie folgt verwalten:

- Das Dashboard bearbeiten.
- Die folgenden Aktionen über das Menü *Aktionen* des Dashboards ausführen:
  - *Freigabe* - die Freigabeeinstellungen für das Dashboard bearbeiten.
  - *Neu erstellen* - ein neues Dashboard erstellen.
  - *Klonen* - ein neues Dashboard durch Klonen des aktuellen Dashboards erstellen. Sie können die Dashboard-Eigenschaften vor dem Speichern bearbeiten.
  - *Löschen* - das Dashboard löschen.
  - *Exportieren* - das Dashboard exportieren.
  - *Neuen Bericht erstellen* - einen geplanten Bericht erstellen. Diese Option ist deaktiviert, wenn Sie keine Berechtigung zum Verwalten geplanter Berichte haben.
  - *Zugehörige Berichte anzeigen* - eine Liste vorhandener Berichte für das aktuelle Dashboard anzeigen. Diese Option ist deaktiviert, wenn keine zugehörigen Berichte vorhanden sind oder Sie keine Berechtigung zum Verwalten geplanter Berichte haben.
- In den Kioskmodus wechseln.
- Die Diashow des Dashboards starten oder stoppen.

Im Dashboard-Anzeigemodus stehen für Widgets die folgenden Optionen zur Verfügung:

- Das Widget bearbeiten.
- Die folgenden Aktionen über das Menü *Aktionen* des Widgets ... ausführen:
  - *Kopieren* - das Widget kopieren.
  - *Bild herunterladen* - das Widget als PNG-Bild herunterladen. Diese Option ist für Widgets verfügbar, die diese Funktion unterstützen (siehe die Seiten der einzelnen Widgets).
  - *Aktualisierungsintervall* - die Häufigkeit der Aktualisierung des Widget-Inhalts auswählen.

Für Dashboards mit mehreren Seiten stehen im Dashboard-Anzeigemodus die folgenden Aktionen im Menü *Aktionen* der Seite  zur Verfügung:

- *Kopieren* - die Seite kopieren.
- *Eigenschaften* - die Seiteneigenschaften anpassen, einschließlich des Namens und des Diashow-Intervalls.

Diashow

Wenn das Kontrollkästchen *Diashow starten* in den Dashboard-Eigenschaften ausgewählt ist, wird die Diashow automatisch gestartet, sobald das Dashboard mehr als eine Seite enthält. Andernfalls können Sie die Diashow durch Klicken auf die Schaltfläche *Diashow starten* starten.



Die Seiten werden entsprechend den in den Eigenschaften des Dashboards und der einzelnen Seiten angegebenen Intervallen gewechselt. Wenn die Eigenschaft *Diashow-Intervall* für eine einzelne Seite definiert ist, wird das für diese Seite angegebene Intervall angewendet. Andernfalls wird das *Standardintervall für Diashows* des Dashboards angewendet.

Um die Diashow zu beenden, klicken Sie auf *Diashow beenden*.

Verwenden Sie im Kioskmodus die folgenden Schaltflächen:

-  , um die Diashow zu starten;
-  , um die Diashow zu beenden;
-  , um zur vorherigen Seite zu navigieren;
-  , um zur nächsten Seite zu navigieren.

Dashboard teilen

Dashboards können öffentlich oder privat sein.

Öffentliche Dashboards sind für alle Benutzer sichtbar. Private Dashboards sind nur für ihre Besitzer sowie für die Benutzer oder Benutzergruppen sichtbar, mit denen sie geteilt wurden.

Um die Freigabeeinstellungen eines Dashboards zu ändern, klicken Sie im Menü *Aktionen* des Dashboards auf *Freigeben*, während Sie das Dashboard anzeigen:

Parameter	Beschreibung
Typ	Wählen Sie den Dashboard-Typ aus: **Privat** – Das Dashboard ist nur für ausgewählte Benutzergruppen und Benutzer sichtbar. **Öffentlich** – Das Dashboard ist für alle Benutzer sichtbar.
Liste der Freigaben für Benutzergruppen	Wählen Sie die Benutzergruppen aus, die auf das Dashboard zugreifen können. Sie können Lesezugriff oder Lese- und Schreibzugriff gewähren.
Liste der Freigaben für Benutzer	Wählen Sie die Benutzer aus, die auf das Dashboard zugreifen können. Sie können Lesezugriff oder Lese- und Schreibzugriff gewähren.

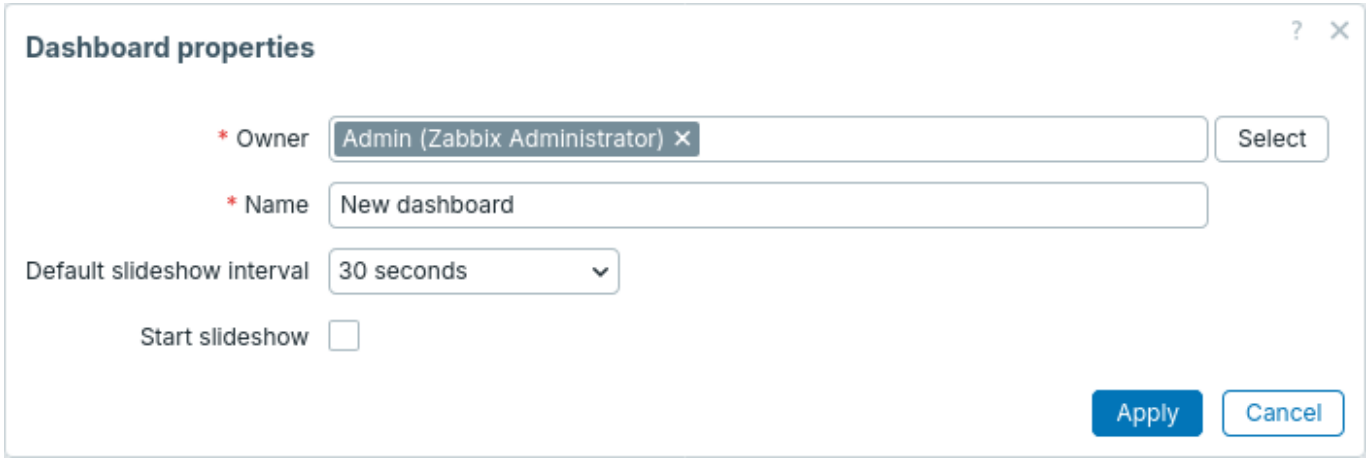
Unter Berechtigungen für Dashboards erfahren Sie, wie Freigabeoptionen die für Benutzer verfügbaren Aktionen einschränken.

Dashboard erstellen

Sie können ein Dashboard auf zwei Arten erstellen:

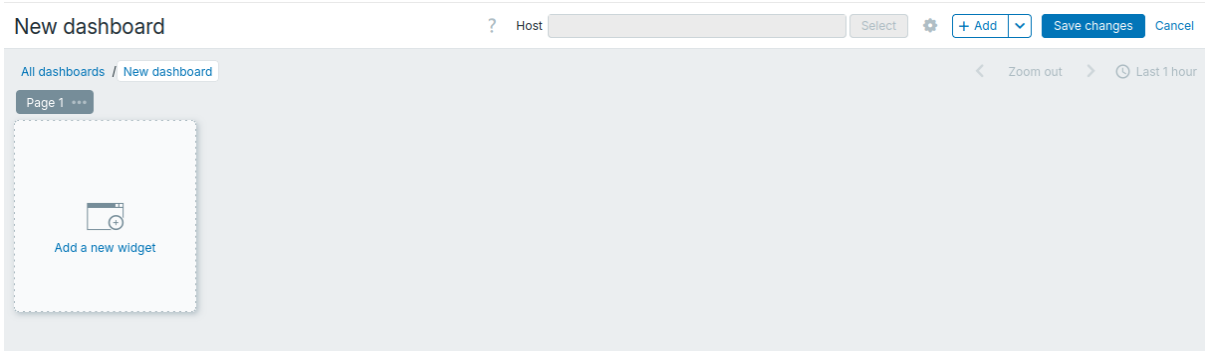
- Klicken Sie in der Dashboard-Liste auf die Schaltfläche *Dashboard erstellen*.
- Klicken Sie im Anzeigemodus des Dashboards im Menü *Aktionen* des Dashboards auf *Neu erstellen*.

Geben Sie anschließend die Dashboard-Eigenschaften an:



Parameter	Beschreibung
Besitzer	Wählen Sie den Benutzer aus, dem das Dashboard gehören soll.
Name	Geben Sie den Namen des Dashboards ein.
Standardintervall der Diashow	Wählen Sie aus, wie lange jede Dashboard-Seite angezeigt wird, bevor die Diashow zur nächsten Seite wechselt.
Diashow starten	Aktivieren Sie dieses Kontrollkästchen, um die Diashow automatisch zu starten, sobald das Dashboard mehr als eine Seite enthält.

Klicken Sie anschließend auf *Anwenden*. Das neue Dashboard wird im Bearbeitungsmodus geöffnet:



Von hier aus können Sie Dashboard-Widgets und -Seiten hinzufügen.

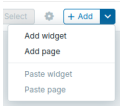
Beachten Sie, dass die Mindestbreite eines Dashboards 1200 Pixel beträgt. In schmaleren Fenstern wird eine horizontale Bildlaufleiste angezeigt. Ein breiteres Dashboard füllt immer das Browserfenster aus; seine Widgets können nicht über die Fensterränder hinausgehen.

Klicken Sie auf die Schaltfläche *Änderungen speichern*, um das Dashboard zu speichern, oder auf *Abbrechen*, um zum vorherigen Bildschirm zurückzukehren.

Widgets hinzufügen

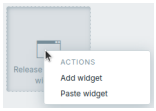
Im Bearbeitungsmodus des Dashboards können Sie Widgets auf zwei Arten hinzufügen:

- Klicken Sie auf die Schaltfläche *Hinzufügen*. Alternativ können Sie auf den Dropdown-Pfeil der Schaltfläche *Hinzufügen* klicken und *Widget hinzufügen* auswählen.



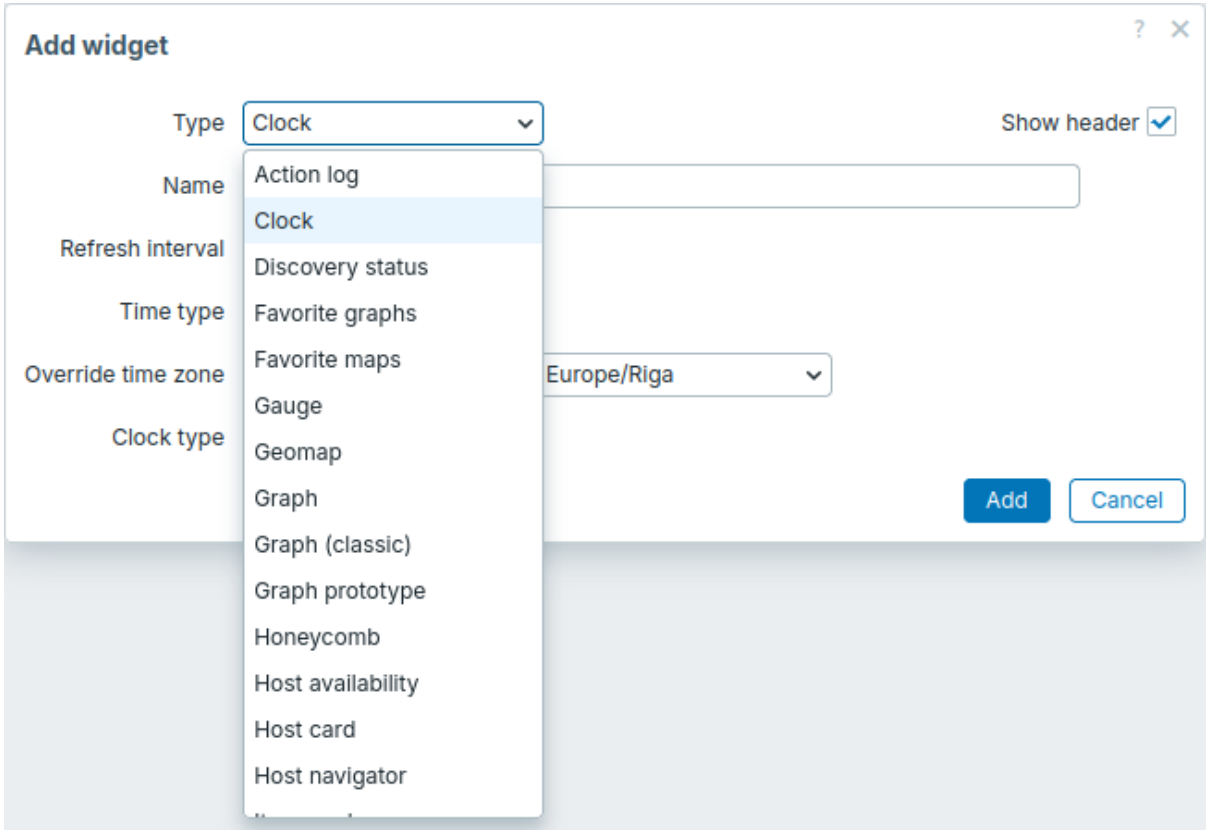
Der Platzhalter für das Widget wird in seiner Standardgröße an der ersten verfügbaren freien Stelle im Dashboard hinzugefügt, und das Widget-Konfigurationsformular wird geöffnet.

- Bewegen Sie den Mauszeiger über einen freien Bereich des Dashboards. Klicken Sie anschließend auf den Widget-Platzhalter, um das Widget-Konfigurationsformular zu öffnen. Alternativ können Sie auf den Platzhalter klicken und ihn ziehen, um seine Größe zu ändern. Lassen Sie ihn anschließend los, um das Widget-Konfigurationsformular zu öffnen. Wenn ein Widget in die Zwischenablage kopiert wurde, werden Sie zunächst aufgefordert, *Widget hinzufügen* oder *Widget einfügen* auszuwählen.



Führen Sie anschließend die folgenden Schritte aus:

1. Wählen Sie im Widget-Konfigurationsformular den Widget-Typ aus.
2. Konfigurieren Sie die Widget-Parameter.
3. Klicken Sie auf *Hinzufügen*, um das Widget zum Dashboard hinzuzufügen.



Das Widget wird in seiner Standardgröße hinzugefügt oder an den verfügbaren Platz angepasst. Sie können die Größe und Position von Widgets wie im Abschnitt *Widgets bearbeiten* beschrieben ändern.

Sie können Widgets auch kopieren und einfügen, um neue Widgets mit derselben Konfiguration zu erstellen – entweder innerhalb desselben Dashboards oder zwischen Dashboards, die in verschiedenen Browser-Tabs geöffnet sind.

Widgets kopieren/einfügen

So kopieren Sie ein Widget und fügen es ein:

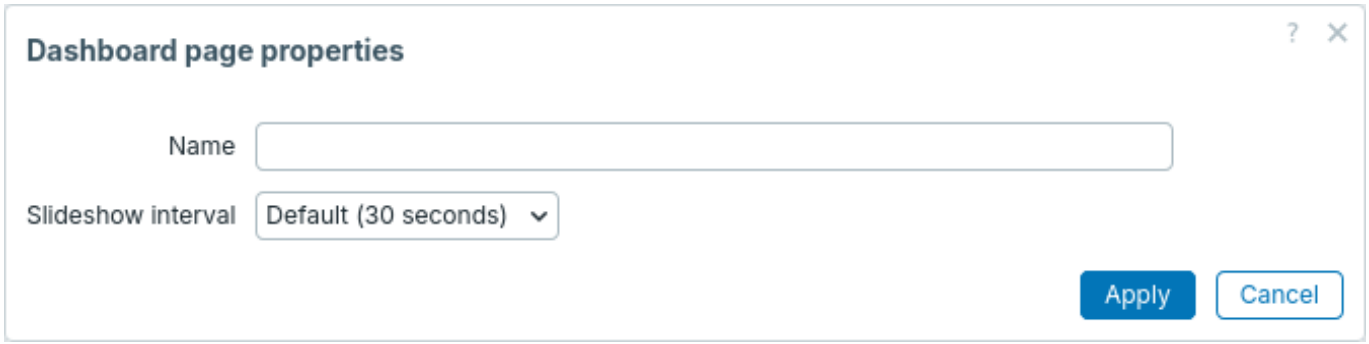
1. Wählen Sie im Menü *Aktionen* des Widgets ... die Option *Kopieren* aus.
2. Klicken Sie im Bearbeitungsmodus des Dashboards auf den Dropdown-Pfeil der Schaltfläche *Hinzufügen* und wählen Sie *Widget einfügen* aus. Alternativ können Sie auf einen leeren Bereich im Dashboard klicken und *Widget einfügen* auswählen.

Sie können ein kopiertes Widget auch über einem vorhandenen Widget einfügen, indem Sie die Option *Einfügen* in dessen Menü *Aktionen* verwenden.

Seiten hinzufügen

So fügen Sie einem Dashboard eine neue Seite hinzu:

1. Klicken Sie im Bearbeitungsmodus des Dashboards auf den Dropdown-Pfeil der Schaltfläche *Hinzufügen* und wählen Sie *Seite hinzufügen* aus.
2. Füllen Sie die Seiteneigenschaften aus:
   - *Name* – geben Sie den Namen der Seite ein. Wenn Sie den Namen leer lassen, wird die Seite mit dem Namen *Seite N* hinzugefügt, wobei „N“ die fortlaufende Seitennummer ist.
   - *Diashow-Intervall* – wählen Sie aus, wie lange die Seite angezeigt wird, bevor die Diashow zur nächsten Seite wechselt.



3. Klicken Sie auf *Anwenden*.

Für die neue Seite wird eine neue Registerkarte (z. B. *Seite 2*) hinzugefügt.



Sie können auch Seiten kopieren und einfügen, um neue Seiten mit denselben Eigenschaften zu erstellen – entweder innerhalb desselben Dashboards oder zwischen Dashboards, die in verschiedenen Registerkarten geöffnet sind.

Seiten kopieren/einfügen

So kopieren Sie eine Seite und fügen sie ein:

1. Wählen Sie im Menü *Aktionen* der Seite die Option *Kopieren* aus.
2. Klicken Sie im Bearbeitungsmodus des Dashboards auf den Dropdown-Pfeil der Schaltfläche *Hinzufügen* und wählen Sie *Seite einfügen* aus.

Dashboard bearbeiten

Um ein Dashboard zu bearbeiten, klicken Sie oben rechts auf die Schaltfläche *Dashboard bearbeiten*. Alternativ können Sie bei einem Widget auf die Schaltfläche *Bearbeiten* klicken. Das Dashboard wird im Bearbeitungsmodus geöffnet.

Von hier aus können Sie Dashboards wie folgt verwalten:

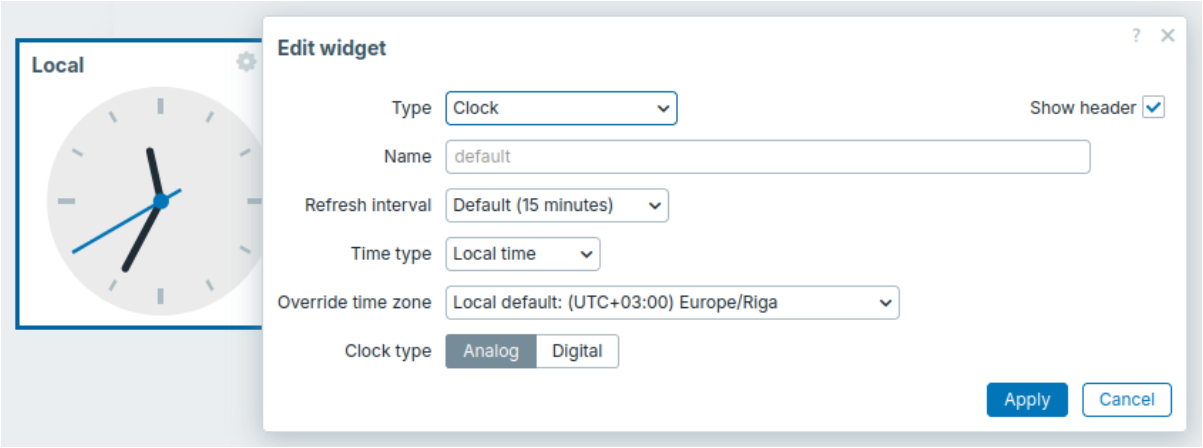
- Bearbeiten Sie die Dashboard-Eigenschaften, indem Sie auf die Schaltfläche mit dem Zahnradsymbol klicken.
- Bearbeiten Sie Dashboard-Widgets.
- Greifen Sie auf die folgenden Dashboard-Aktionen zu, indem Sie auf den Dropdown-Pfeil der Schaltfläche *Hinzufügen* klicken:
  - *Widget hinzufügen* – Fügen Sie ein Widget zum Dashboard hinzu.
  - *Seite hinzufügen* – Fügen Sie eine Seite zum Dashboard hinzu.
  - *Widget einfügen* – Fügen Sie ein kopiertes Widget ein. Diese Option ist deaktiviert, wenn kein Widget kopiert wurde. Es kann jeweils nur ein Widget kopiert werden.
  - *Seite einfügen* – Fügen Sie eine kopierte Seite ein. Diese Option ist deaktiviert, wenn keine Seite kopiert wurde. Es kann jeweils nur eine Seite kopiert werden.

Klicken Sie nach der Bearbeitung des Dashboards auf die Schaltfläche *Änderungen speichern*, um das Dashboard zu speichern, oder auf *Abbrechen*, um zum vorherigen Bildschirm zurückzukehren.

Widgets bearbeiten

Ein Widget kann im Dashboard-Bearbeitungsmodus bearbeitet werden, indem Sie auf die Schaltfläche *Bearbeiten* klicken. Alternativ können Sie beim Anzeigen eines Dashboards den Mauszeiger über ein Widget bewegen und auf die Schaltfläche *Bearbeiten* klicken.

Während ein Widget bearbeitet wird, ist es hervorgehoben, und Änderungen werden in Echtzeit als Vorschau angezeigt.



Sie können das Widget und sein Konfigurationsformular neu positionieren, indem Sie deren Titelleisten ziehen, und die Größe des Widgets ändern, indem Sie an seinen Ziehpunkten ziehen.



Beim Bearbeiten eines Widgets sind über das Menü *Aktionen* ... des Widgets folgende Aktionen verfügbar:

- *Kopieren* – kopiert das Widget.
- *Einfügen* – fügt ein kopiertes Widget über diesem Widget ein.
  Diese Option ist deaktiviert, wenn kein Widget kopiert wurde.
- *Löschen* – löscht das Widget.

Wenn obligatorische Widget-Parameter nicht konfiguriert sind, wechselt das Widget in den Status *Widget ist nicht vollständig konfiguriert*.

Klicken Sie nach der Bearbeitung im Widget-Konfigurationsformular auf *Anwenden* und anschließend oben rechts im Dashboard auf *Änderungen speichern*, um Ihre Änderungen anzuwenden.

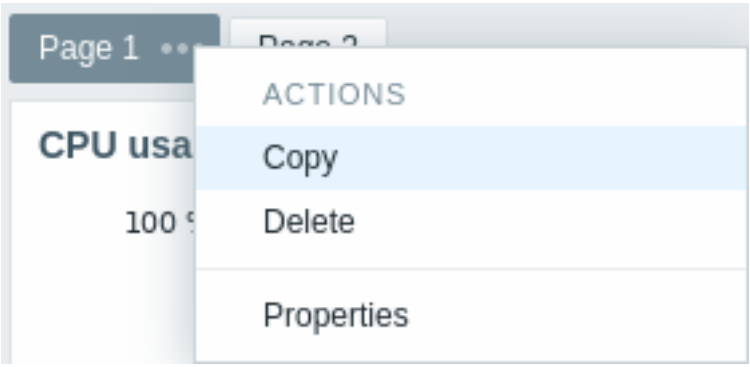
Seiten bearbeiten

Seiten können im Bearbeitungsmodus des Dashboards bearbeitet werden.

Die Seiten können durch Ziehen und Ablegen der Seitenregisterkarten neu angeordnet werden.
Durch die Neuanordnung werden Seiten nicht umbenannt, außer bei Seiten mit automatisch zugewiesenen Namen (*Page N*), deren Nummerierung angepasst wird.

Beim Bearbeiten einer Seite sind über das Seitenmenü *Actions* die folgenden Aktionen verfügbar:

- *Copy* – Kopieren der Seite.
- *Delete* – Löschen der Seite.
- *Properties* – Anpassen der Seiteneigenschaften.



Berechtigungen für Dashboards

Die Berechtigungen für Dashboards sind für reguläre Benutzer und Benutzer des Typs *Admin* wie unten beschrieben eingeschränkt, sofern der Zugriff auf Dashboards und die Option *Dashboards erstellen und bearbeiten* für ihre zugewiesene Benutzerrolle aktiviert sind:

- Sie können ein öffentliches Dashboard anzeigen und klonen, auch wenn den Freigabeeinstellungen des Dashboards keine Benutzer oder Benutzergruppen hinzugefügt wurden.
- Sie können ein privates Dashboard anzeigen und klonen, wenn sie mindestens über *Leserechte* dafür verfügen, wie in den Freigabeeinstellungen angegeben.
- Sie können ein Dashboard nur bearbeiten und löschen, wenn sie über *Lese-/Schreibrechte* dafür verfügen, wie in den Freigabeeinstellungen angegeben.
- Sie können den Besitzer des Dashboards nicht ändern.

Dashboard-Widgets

Übersicht

Zabbix bietet eine Vielzahl von Dashboard-Widgets, die Monitoring-Daten visualisieren und zusammenfassen und Ihnen so einen klaren Überblick über Systemleistung und Verfügbarkeit geben.

Nachfolgend finden Sie eine Übersicht aller Widgets, gruppiert nach ihrem primären Zweck.

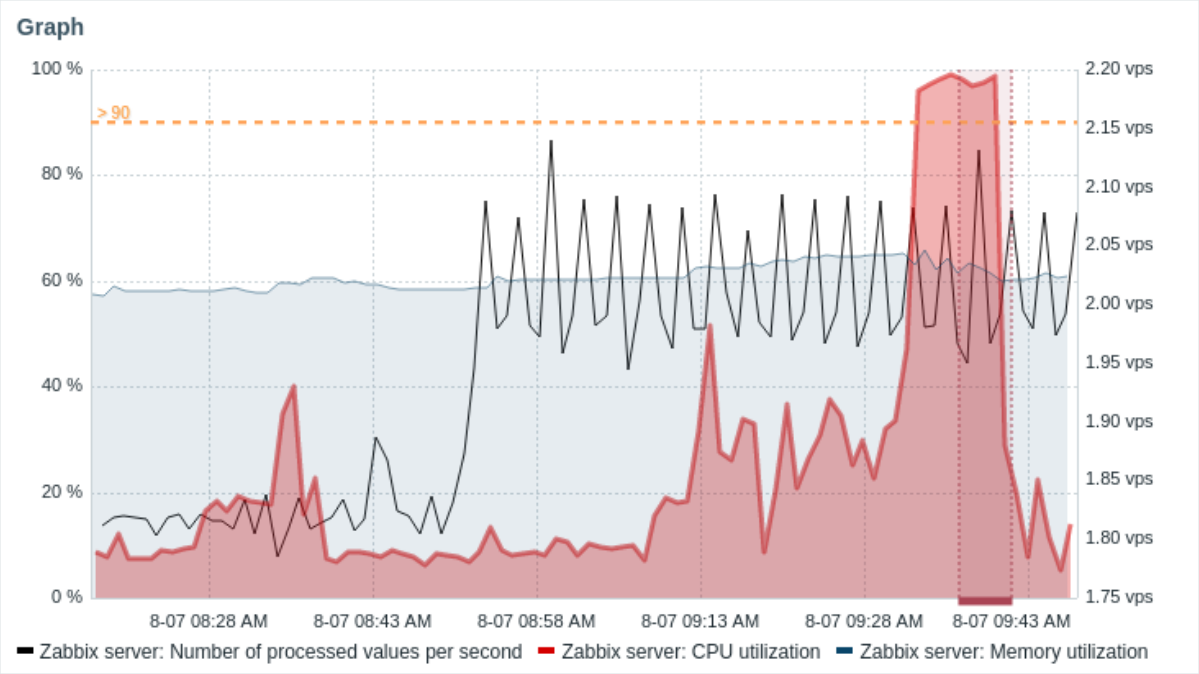
Ausführliche Informationen zur Widget-Konfiguration finden Sie unter Dashboard-Widget-Parameter.

Datenvisualisierung

Diese Widgets stellen Daten in verschiedenen visuellen Formaten dar und helfen Ihnen, Daten und Probleme auf unterschiedliche Weise zu überwachen.

Graph

Das Widget Graph zeigt numerische Datenpunkte als vektorbasiertes Diagramm an. Es kann Ihnen helfen, Metriken zu verfolgen, Probleme zu erkennen und Werte im Zeitverlauf sowie Host-übergreifend zu vergleichen.

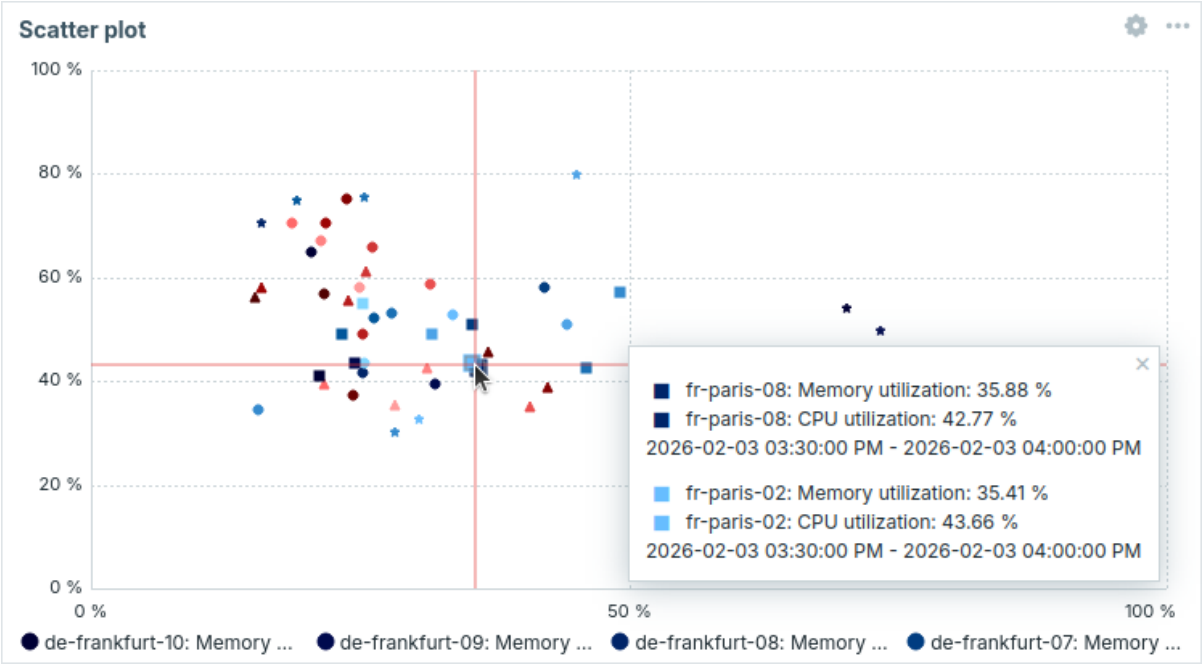


Siehe auch:

- Graph (classic) - das ältere Graph-Widget, das numerische Datenpunkte als bildbasiertes Diagramm anzeigt.
- Graph prototype - zeigt ein Raster automatisch erstellter benutzerdefinierter Diagramme basierend auf Diagrammprototypen oder Datenpunktprototypen von Low-Level-Discovery-Regeln an.
- Favorite graphs - zeigt eine alphabetische Liste von Verknüpfungen zu Diagrammen an, die vom aktuellen Benutzer als Favoriten markiert wurden.

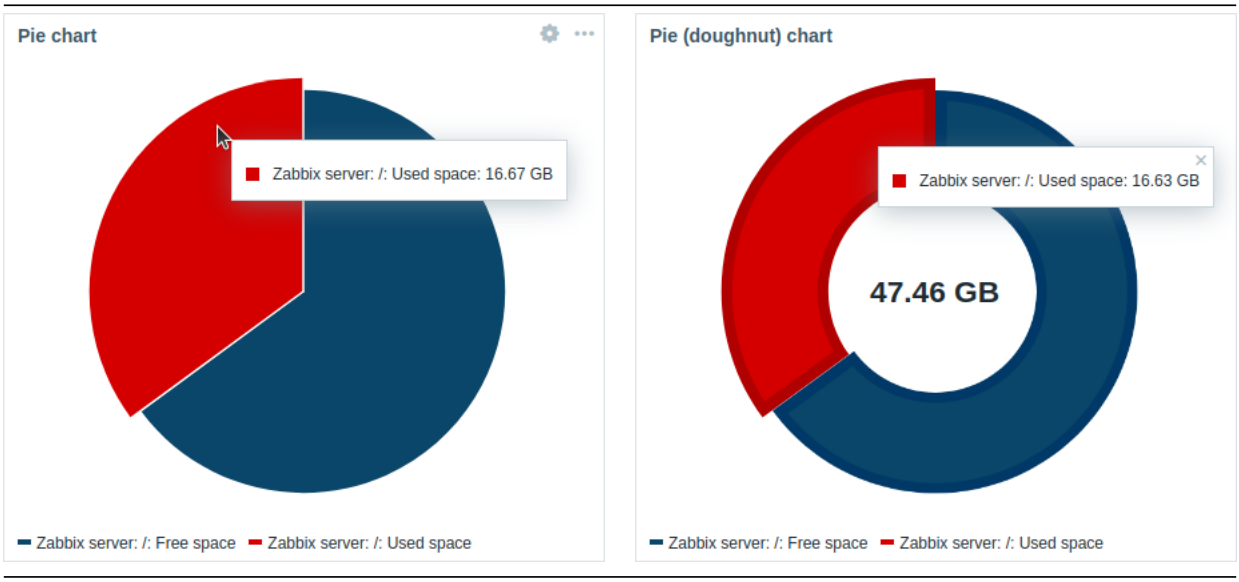
Streudiagramm

Das Widget Scatter plot zeigt die Beziehung zwischen zwei Metriken an, indem einzelne Datenpunkte entlang einer X- und Y-Achse dargestellt werden. Dies hilft dabei, Muster, Cluster, Korrelationen und Ausreißer im Datensatz sichtbar zu machen.



Kreisdiagramm

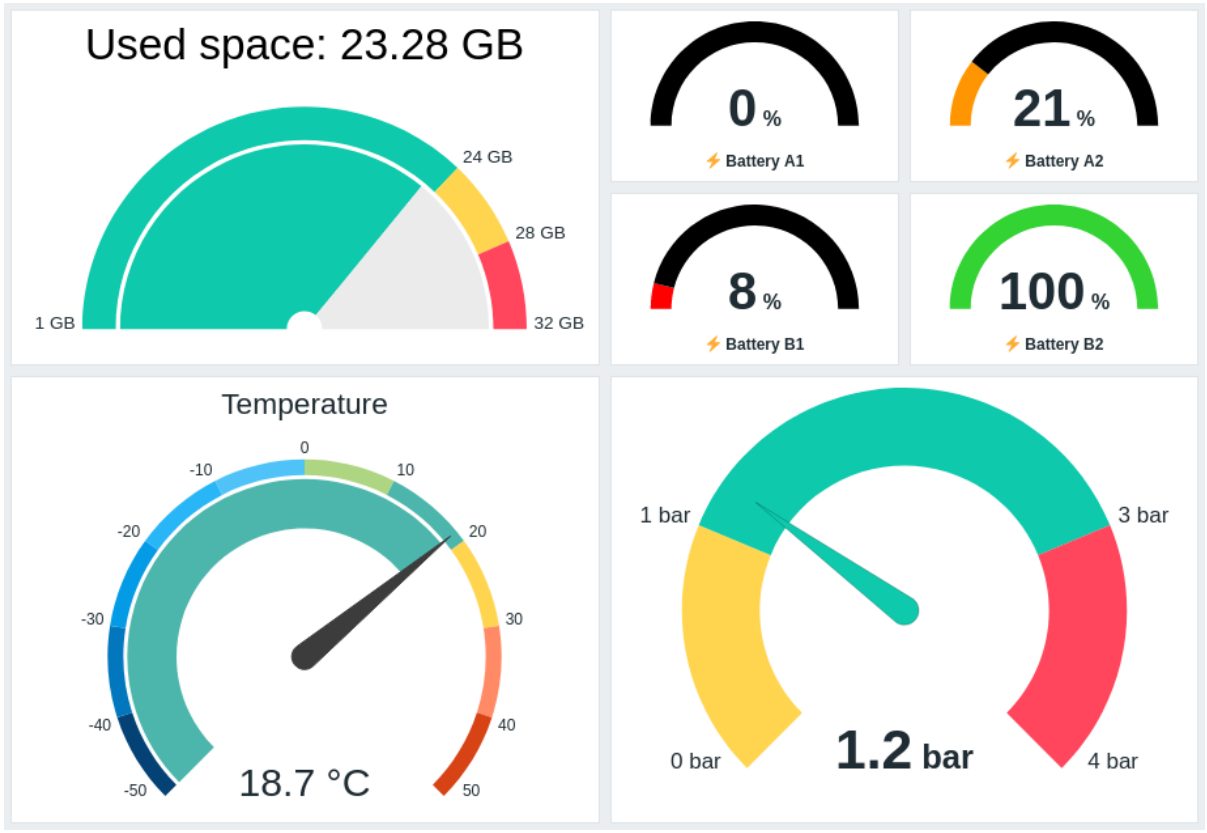
Das Widget Kreisdiagramm zeigt numerische Datenpunkt-Daten als vektorbasiertes Kreis- oder Ringdiagramm an. Dieses Widget kann Ihnen dabei helfen zu visualisieren, wie Datenpunkte oder Hosts zum gesamten Datensatz beitragen.



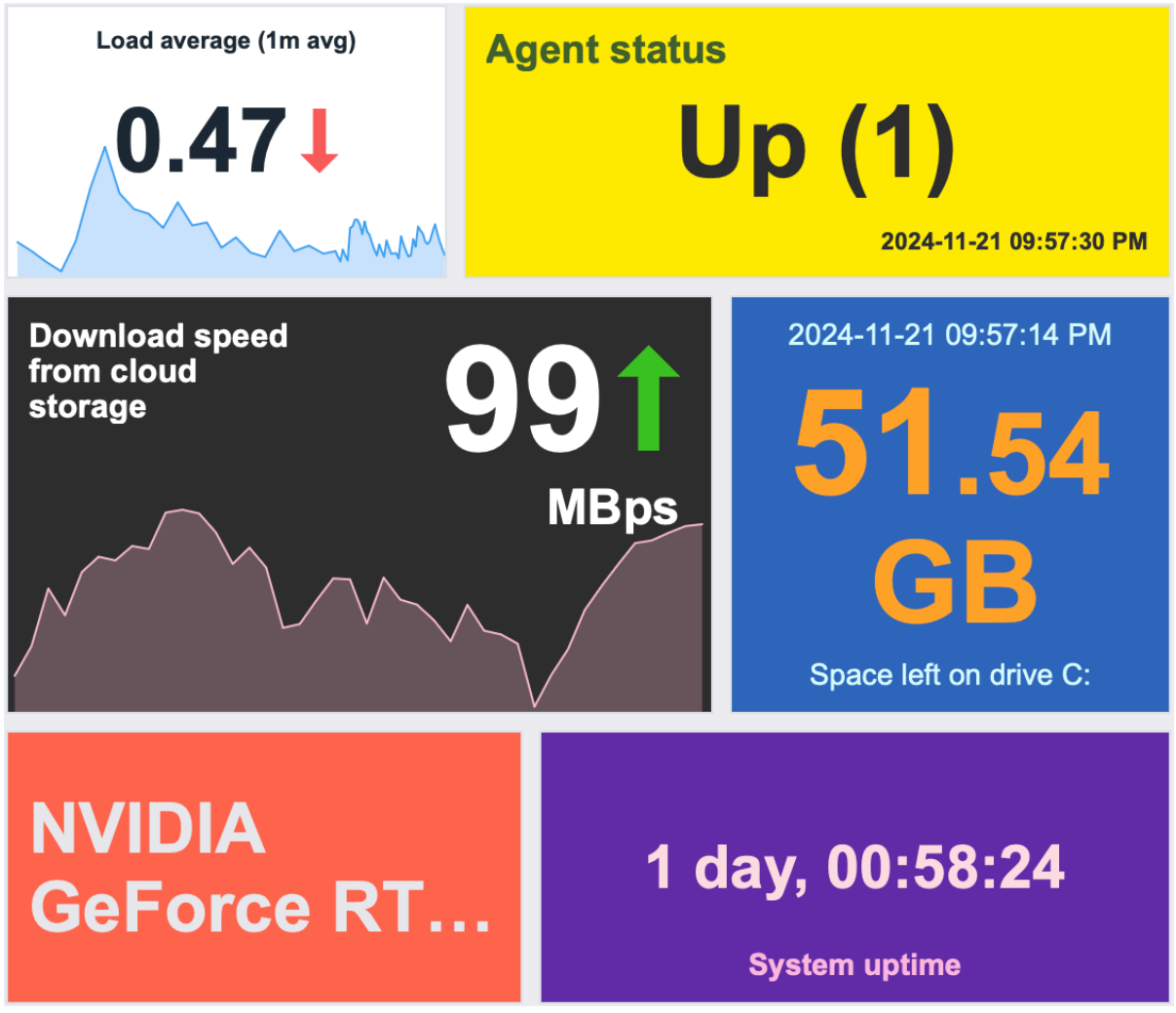
Messanzeige/Datenpunktwert

Die Widgets Messanzeige/Datenpunktwert zeigen den Wert eines einzelnen Datenpunkts an. Sie sind nützlich, um wichtige Metriken im Blick zu behalten, Schwellenwerte zu visualisieren und plötzliche Änderungen in den Daten zu erkennen.

Widget „Messanzeige“:

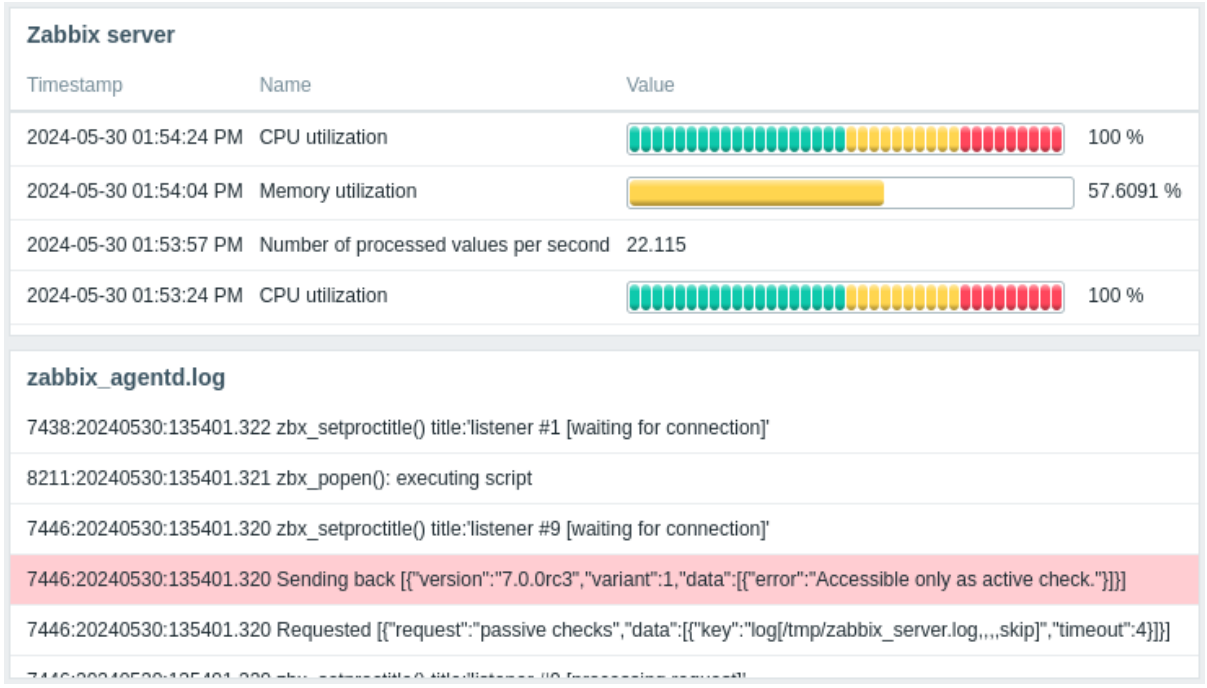


Widget „Datenpunktwert“:



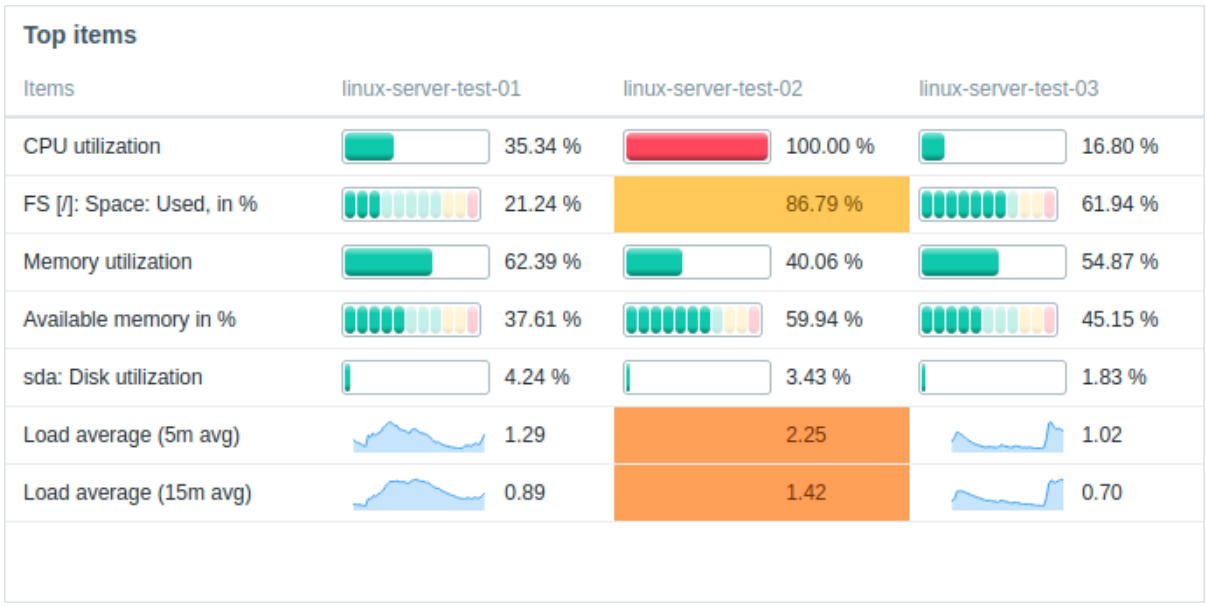
Datenpunkt-Verlauf

Das Widget Datenpunkt-Verlauf zeigt die neuesten Daten für alle Datenpunkt-Typen (numerisch, Text usw.) in Tabellenform an. Es kann Fortschrittsbalken und Bilder anzeigen (nützlich für Browser-Datenpunkte) und Werte hervorheben (nützlich für die Überwachung von Protokolldateien).



Top-Datenpunkte

Das Widget Top-Datenpunkte zeigt die höchsten/niedrigsten Werte für ausgewählte Datenpunkte an und bietet einen schnellen Überblick über deren Leistung.

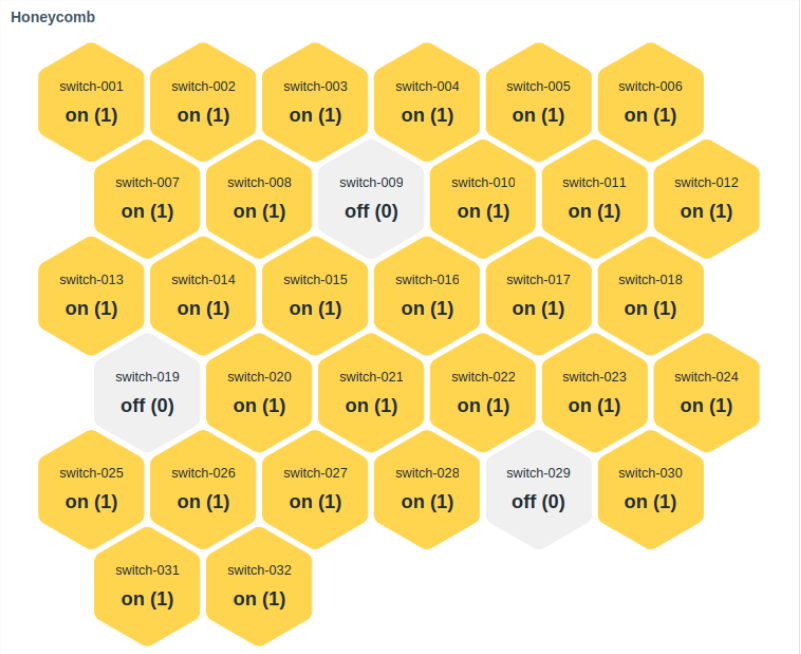


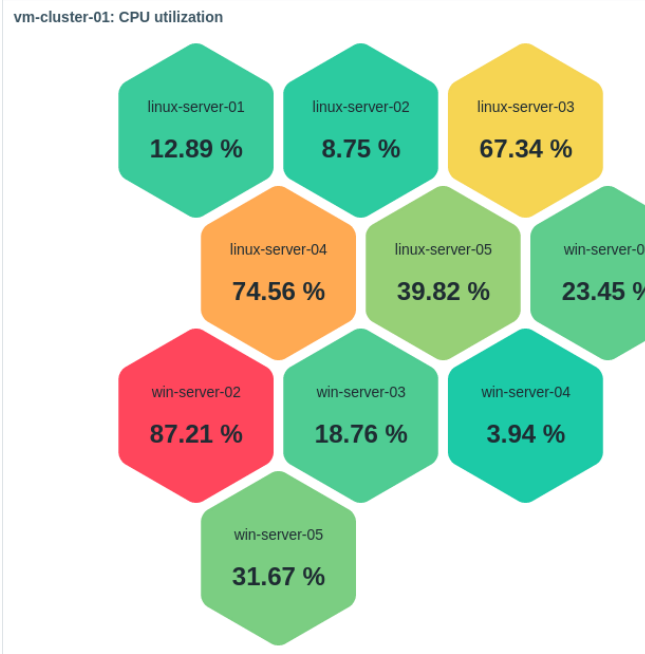
Siehe auch:

- Top-Hosts – zeigt die höchsten/niedrigsten Werte für ausgewählte Datenpunkte von mehreren Hosts an.
- Top-Auslöser – zeigt Auslöser mit der höchsten Anzahl an Problemen an.

Honeycomb

Das Widget Honeycomb zeigt Hosts oder Datenpunkte als sechseckige Zellen in einem Raster an. Dieses Layout erleichtert es, problematische Hosts/Datenpunkte zu erkennen, Cluster von Problemen zu identifizieren, Gruppen auf einen Blick zu vergleichen und mehr.



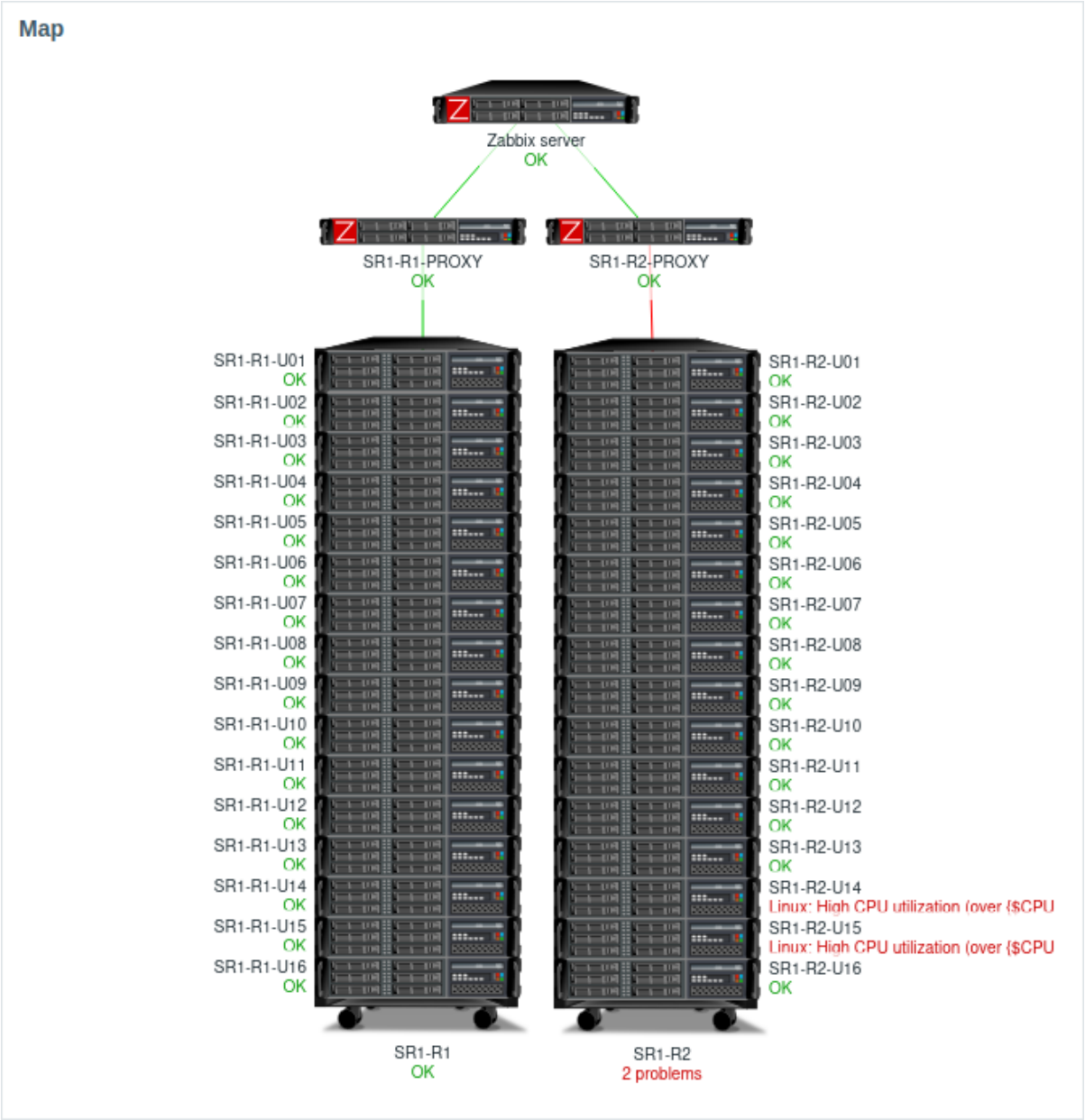


Karten

Karten-Widgets visualisieren die Netzwerktopologie oder Host-Standorte und helfen dabei, den Gesamtzustand sowie den zustand bestimmter Standorte zu beurteilen.

Karte

Das Widget Map kann eine Netzwerkkarte oder eine ähnliche Visualisierung anzeigen und bietet Ihnen einen dynamischen Überblick über Ihr Netzwerk.

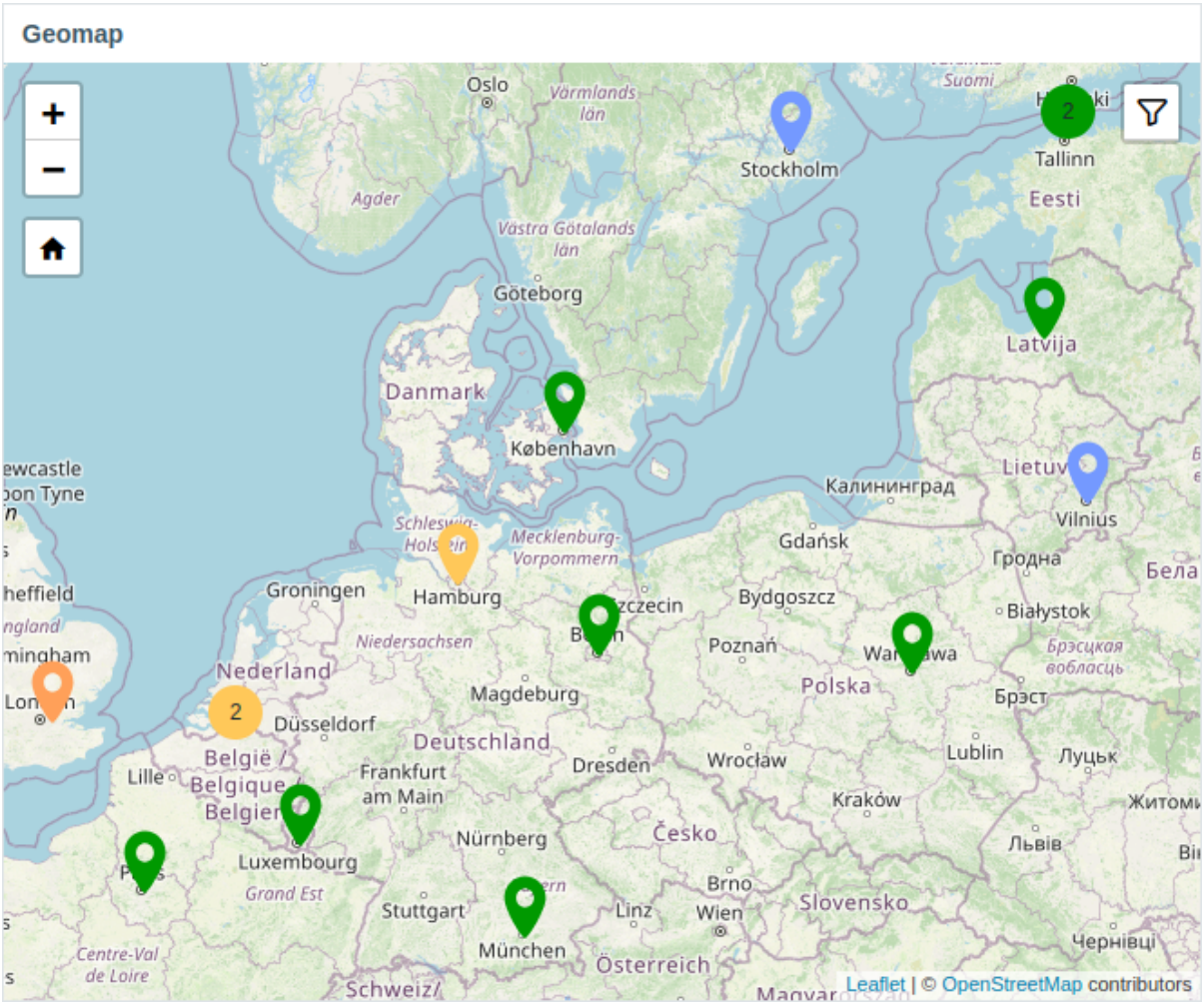


Siehe auch:

- Navigationsbaum für Karten – zeigt eine Hierarchie vorhandener Karten sowie die Problemanzahl für jede Karte und Kartengruppe an.
- Bevorzugte Karten – zeigt eine alphabetisch sortierte Liste von Verknüpfungen zu Karten an, die vom aktuellen Benutzer als Favoriten markiert wurden.

Geomap

Das Widget Geomap zeigt Hosts als Markierungen auf einer interaktiven, Leaflet-basierten Karte an, wobei die Farben der Markierungen das jeweils schwerwiegendste Problem des Hosts anzeigen.

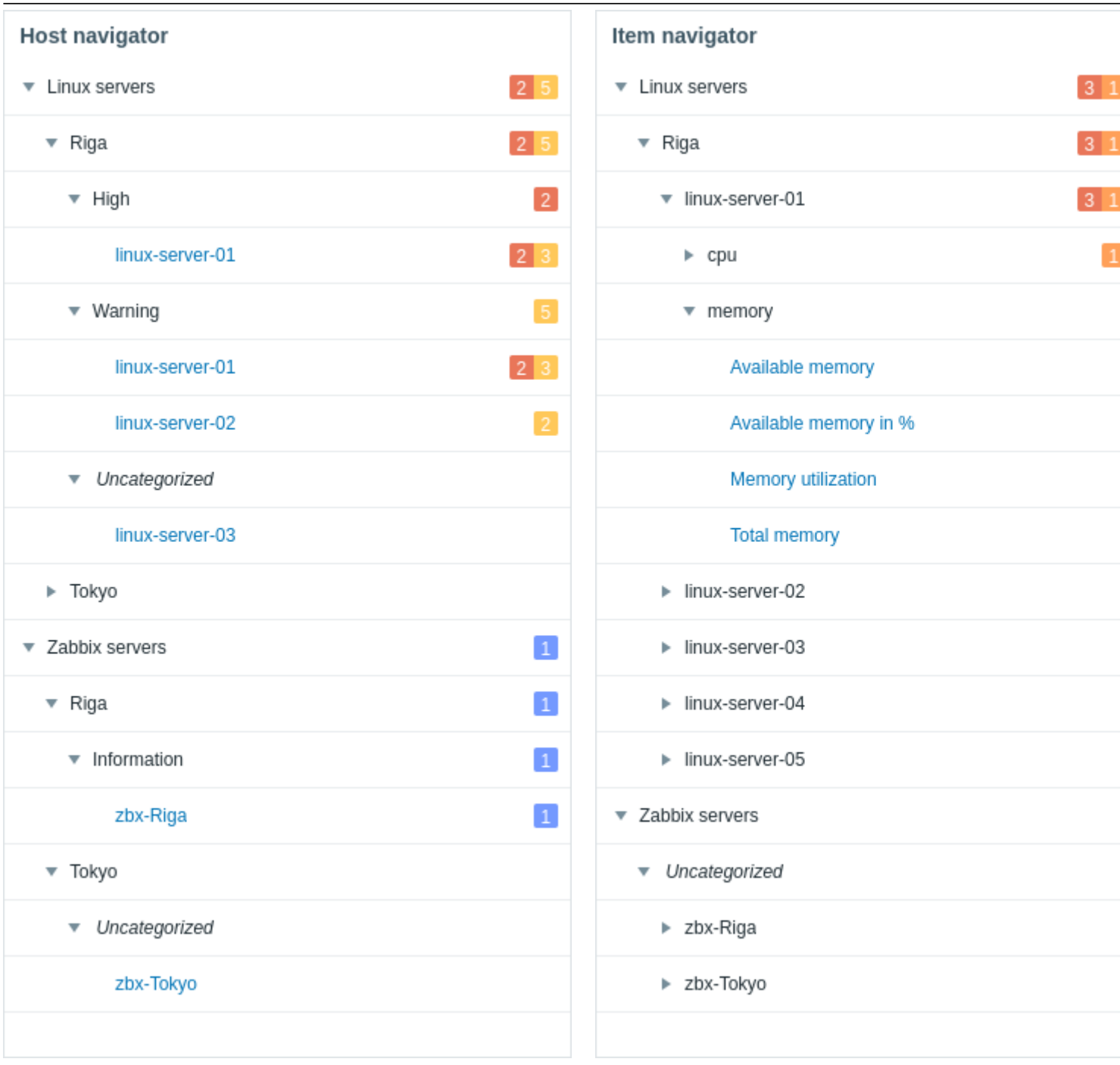


Host-/Datenpunkt-Details

Diese Widgets stellen Listen und Detailansichten von Hosts oder Datenpunkten bereit.

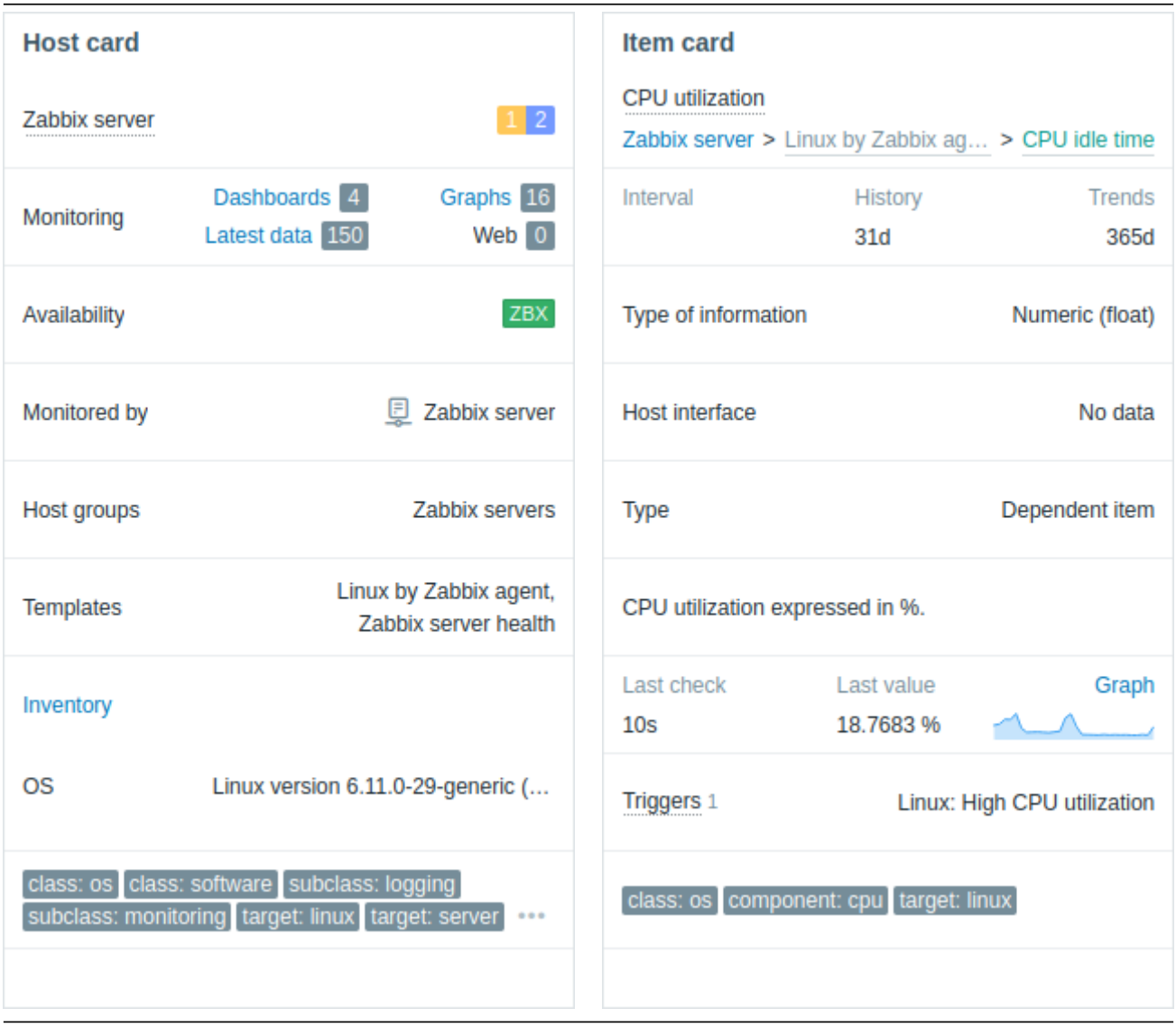
Host-/Datenpunkt-Navigator

Die Widgets Host-Navigator und Datenpunkt-Navigator zeigen eine Liste von Hosts/Datenpunkten basierend auf verschiedenen Filter- und Gruppierungsoptionen an. Diese Widgets sind besonders nützlich, um zu steuern, was andere Widgets basierend auf dem ausgewählten Host/Datenpunkt anzeigen.



Host-/Datenpunkt-Karte

Die Widgets Host card/Item card zeigen Details zu einem einzelnen Host/Datenpunkt an, sodass Sie schnell den Status und die Konfiguration eines Hosts oder Datenpunkts beurteilen können (Host-Verfügbarkeit, Probleme, Inventar, neueste Daten des Datenpunkts, Fehler, Beschreibung usw.).



Status und Berichte

Diese Widgets zeigen Statusdaten (z. B. Verfügbarkeit, Auslöser, Probleme) sowie Berichte an.

Host-Verfügbarkeit

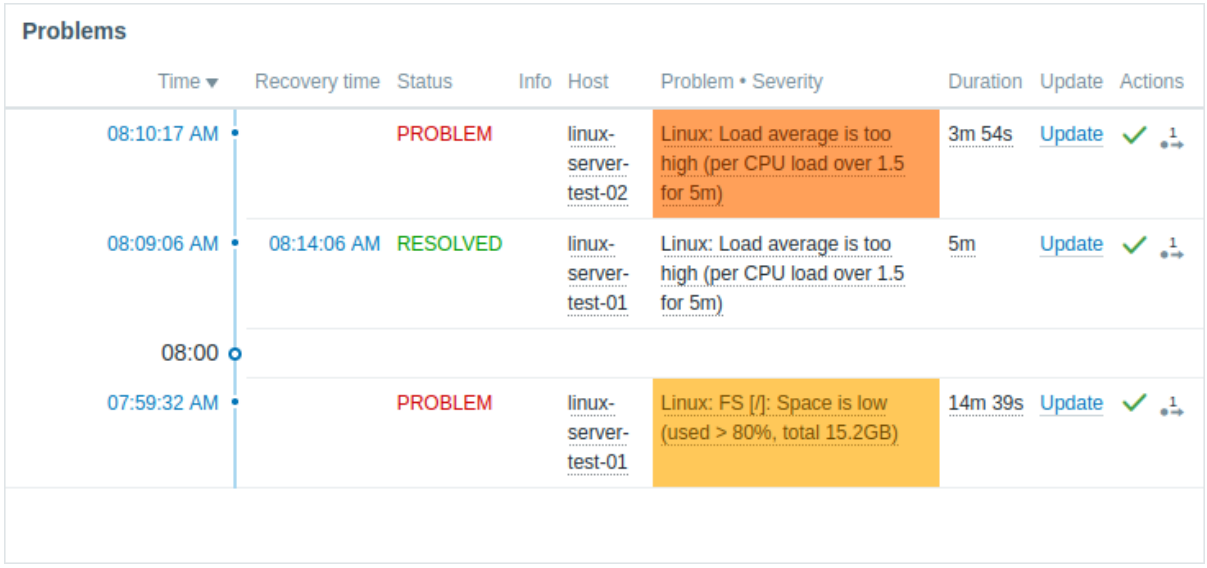
Das Widget Host-Verfügbarkeit zeigt die Verfügbarkeit von Hosts in den ausgewählten Host-Gruppen an und ermöglicht es Ihnen, die Anzahl der Hosts zu überwachen, die verfügbar oder nicht verfügbar sind.

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

Probleme

Das Widget Probleme zeigt aktuelle Probleme an, die nach verschiedenen Parametern gefiltert sind (Host-Gruppen, Hosts, Problemnamen usw.), und bietet Ihnen so eine klare Übersicht darüber, was Aufmerksamkeit erfordert.



Siehe auch:

- Probleme nach Schweregrad – zeigt die Anzahl der Ursprungsprobleme gruppiert nach Schweregrad an.
- Problem-Hosts – zeigt die Anzahl der Ursprungsprobleme pro Host-Gruppe an und zeigt den höchsten Problemschweregrad in jeder Gruppe.

Auslöser-Übersicht

Das Widget Trigger overview zeigt die aktuellen Zustände von Host-Auslösern als Tabelle mit farbigen Blöcken an. Es bietet eine schnelle visuelle Übersicht über den Zustand und die Aktivität von Auslösern auf verschiedenen Hosts.



Aktionsprotokoll

Das Widget Aktionsprotokoll zeigt Details zu Operationen (Benachrichtigungen, Remote-Befehle) an, die innerhalb einer Aktion ausgeführt werden. Das detaillierte Protokoll unterstützt bei der Auditierung, Fehlerbehebung und Überwachung der Ausführung jeder Aktion.

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

SLA-Bericht

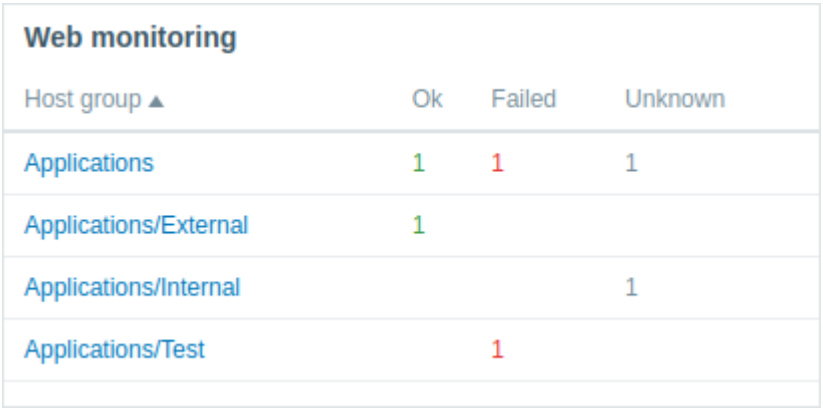
Das Widget SLA report zeigt SLA-Berichte an und hilft Ihnen, die Service-Performance anhand definierter Ziele zu überwachen.

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

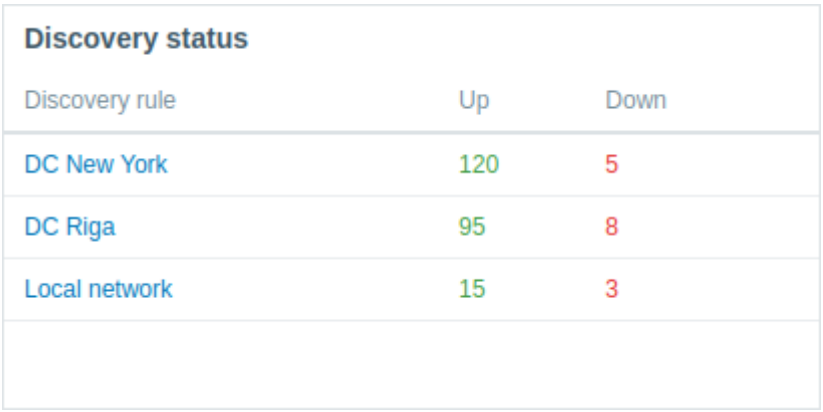
Webüberwachung

Das Widget Web monitoring zeigt den Status aktiver Webszenarien an.



Discovery-Status

Das Widget Discovery status zeigt den Status von Geräten an, die durch aktivierte Regeln zur Netzwerkdiscovery erkannt wurden. Es hilft Ihnen dabei, zu überwachen, ob die Netzwerkdiscovery korrekt funktioniert und wie viele Geräte erreichbar oder nicht erreichbar sind.



Dashboard-Dienstprogramme und Systeminformationen

Diese Widgets bieten ergänzende Informationen und Werkzeuge, um Ihre Dashboards zu erweitern.

Systeminformationen

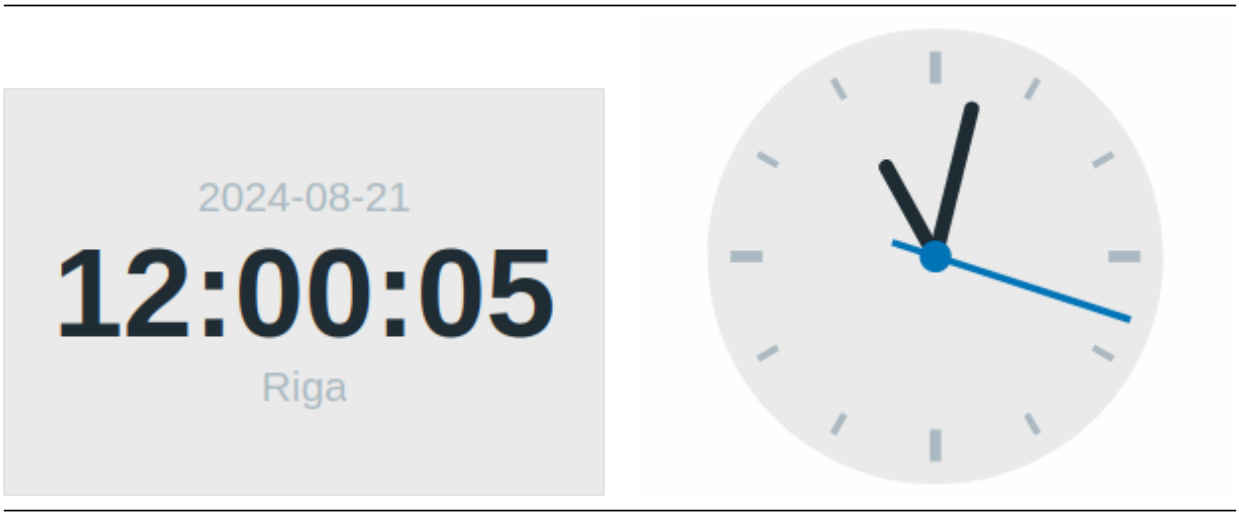
Das Widget Systeminformationen zeigt eine Zusammenfassung wichtiger Zabbix-Server- und Systemdaten oder Details zu Hochverfügbarkeitsknoten an.

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	

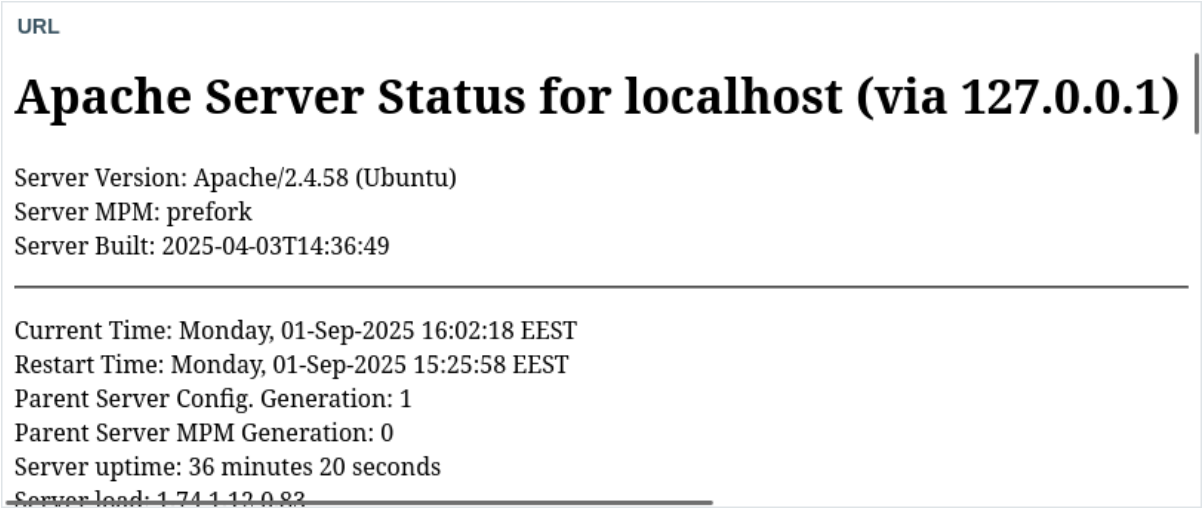
Uhr

Das Widget Clock zeigt die lokale Zeit, die Serverzeit oder die Zeit des Hosts im analogen oder digitalen Format an.



URL

Das URL Widget zeigt den Inhalt an, der von einer URL abgerufen wurde.



Dashboard-Widget-Parameter

Übersicht

Diese Seite beschreibt Parameter, die für alle Dashboard-Widgets gemeinsam sind, sowie dynamische Parameter, die es mehreren Widgets ermöglichen, Konfigurationsdaten untereinander oder mit dem Dashboard auszutauschen.

Um die spezifischen Parameter für jedes Widget anzuzeigen, verwenden Sie bitte die Seitenleiste, um zur entsprechenden Widget-Seite zu navigieren.

Allgemeine Parameter

Die folgenden Parameter gelten für jedes einzelne Widget:

Name	Geben Sie einen Widget-Namen ein.
Aktualisierungsintervall	Konfigurieren Sie das standardmäßige Aktualisierungsintervall. Die standardmäßigen Aktualisierungsintervalle für Widgets reichen je nach Widget-Typ von *Keine Aktualisierung* bis *15 Minuten*. Beispiele: - *Keine Aktualisierung* für das Widget *URL*; - *1 Minute* für das Widget *Aktionsprotokoll*; - *15 Minuten* für das Widget *Uhr*. Aktualisierungsintervalle können für alle Benutzer auf einen Standardwert gesetzt werden. Wechseln Sie das Dashboard in den Bearbeitungsmodus, klicken Sie auf die Schaltfläche Widget bearbeiten und wählen Sie das gewünschte Aktualisierungsintervall aus der Dropdown-Liste aus. Jeder Benutzer kann außerdem ein eigenes Aktualisierungsintervall für Widgets festlegen. Klicken Sie in der Ansicht des Dashboards auf die Schaltfläche mit den drei Punkten eines Widgets und wählen Sie das gewünschte Aktualisierungsintervall aus der Dropdown-Liste aus. Beachten Sie, dass das individuelle Aktualisierungsintervall eines Benutzers Vorrang vor der Widget-Einstellung hat und auch dann beibehalten wird, wenn die Einstellung des Widgets geändert wird.
Kopfzeile anzeigen	Aktivieren Sie das Kontrollkästchen, um die Kopfzeile des Widgets dauerhaft anzuzeigen. Wenn das Kontrollkästchen nicht aktiviert ist, wird die Kopfzeile ausgeblendet, um Platz zu sparen, und nur angezeigt, wenn sich der Mauszeiger über dem Widget befindet (sowohl im Ansichts- als auch im Bearbeitungsmodus). Beim Ziehen eines Widgets an eine neue Position wird die Kopfzeile ebenfalls halbtransparent angezeigt.

Dynamische Parameter

Mehrere Widget-Parameter (z. B. *Hosts*, *Host überschreiben*, *Zeitperiode*) unterstützen dynamische Datenquellen. Anstatt statische Werte zu konfigurieren, können Sie diese Parameter mit anderen Widgets oder dem Dashboard selbst verknüpfen. Widgets werden dann automatisch auf Grundlage von Auswahlen oder Übertragungen aus anderen Teilen des Dashboards aktualisiert.

Widget-Kompatibilität

Einige Widgets können Konfigurationsdaten an andere Widgets übertragen, einige können Daten empfangen, und einige können beides. Zum Beispiel:

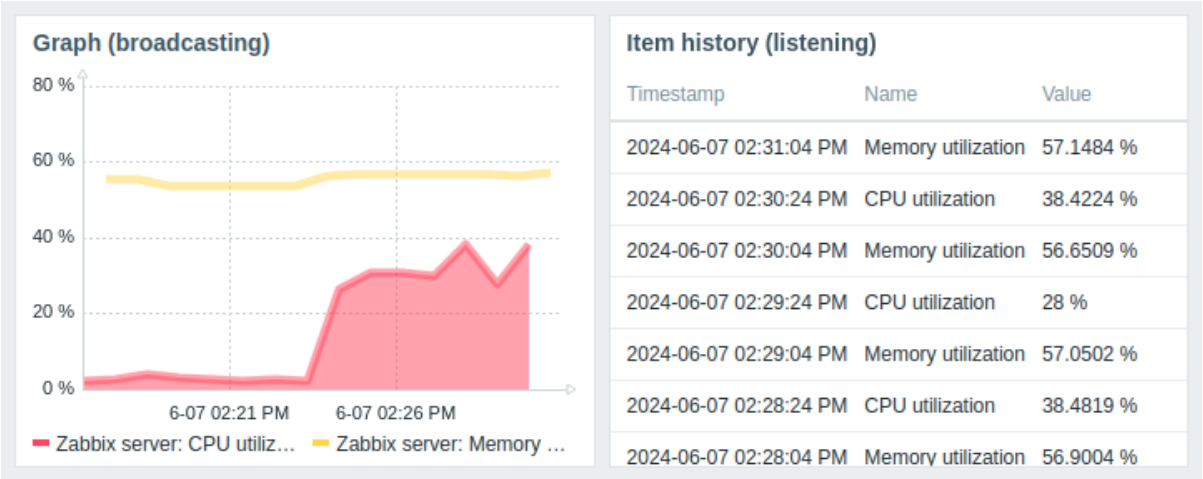
- Das Widget *Aktionsprotokoll* kann Zeitraumbereichsdaten nur von den Widgets *Graph*, *Graph (classic)* und *Graph prototype* abrufen.
- Das Widget *Geomap* kann Host-Daten an Widgets übertragen, die diese empfangen (*Honeycomb*, *Top items* usw.), und kann außerdem Hostgruppen- und Host-Daten von Widgets empfangen, die diese übertragen (*Honeycomb*, *Problem hosts* usw.).
- Das Widget *Uhr* kann weder Daten übertragen noch empfangen.

Die folgende Tabelle zeigt die Sende- und Empfangsfähigkeiten jedes Widgets.

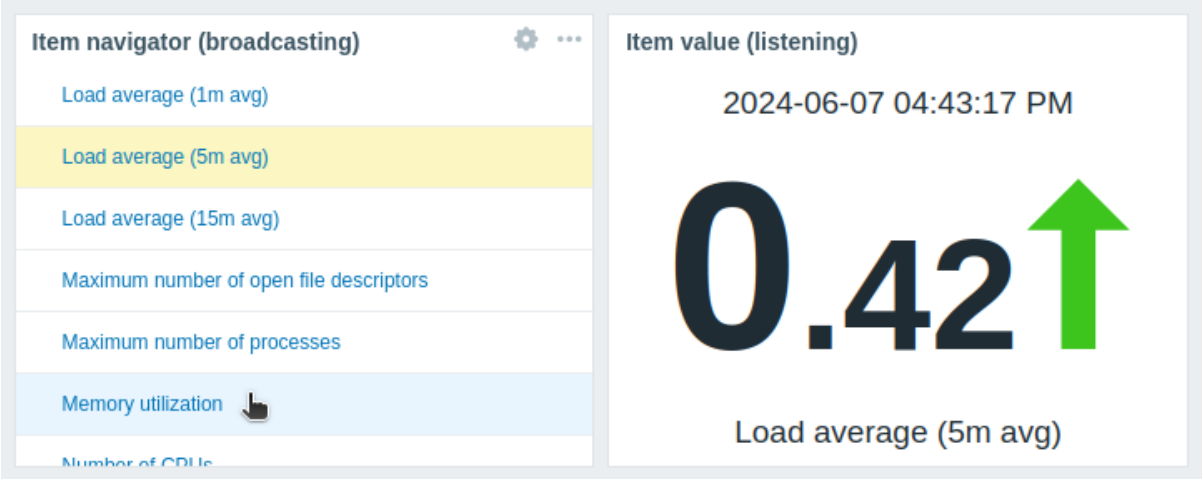
Widget	Überträgt	Empfängt
Aktionsprotokoll	-	Zeitraum
Uhr	-	-
Discovery-Status	-	-
Favorisierte Graphen	-	-
Favorisierte Karten	-	-
Messanzeige	-	Hosts, Datenpunkte
Geomap	Hosts	Hostgruppen, Hosts
Graph	Zeitraum, Datenpunkte	Hosts, Datenpunkte, Zeitraum
Graph (classic)	Zeitraum	Hosts, Datenpunkte, Graphen, Zeitraum
Graph prototype	Zeitraum	Hosts, Zeitraum
Honeycomb	Hosts, Datenpunkte	Hostgruppen, Hosts
Host-Verfügbarkeit	-	Hostgruppen
Host-Karte	-	Hosts
Host-Navigator	Hosts	Hostgruppen
Datenpunkt-Karte	-	Hosts, Datenpunkte, Zeitraum
Datenpunkt-Verlauf	Datenpunkte	Hosts, Zeitraum
Datenpunkt-Navigator	Datenpunkte	Hostgruppen, Hosts
Datenpunkt-Wert	-	Hosts, Datenpunkte, Zeitraum
Karte	Hostgruppen, Hosts	Karten
Karten-Navigationsbaum	Karten	-
Kreisdiagramm	-	Datenpunkte, Zeitraum
Problem-Hosts	Hostgruppen	Hostgruppen, Hosts
Probleme	Ereignisse	Hostgruppen, Hosts
Probleme nach Schweregrad	Hostgruppen	Hostgruppen, Hosts
Streudiagramm	Zeitraum, Datenpunkte	Hosts, Datenpunkte, Zeitraum
SLA-Bericht	-	-
Systeminformationen	-	-
Top-Hosts	Hosts	Hostgruppen, Hosts
Top-Datenpunkte	-	Hostgruppen, Hosts
Top-Auslöser	-	Zeitraum
Auslöser-Übersicht	-	Hostgruppen, Hosts
URL	-	Hosts
Web-Überwachung	Hostgruppen	Hostgruppen, Hosts

Widgets unterscheiden sich darin, wie sie Daten an andere Widgets **übertragen**.

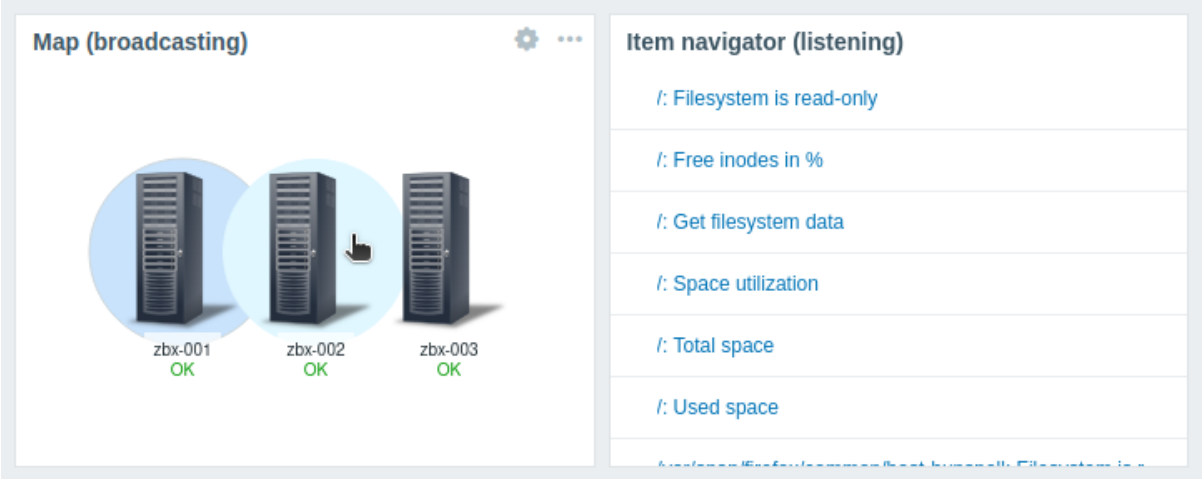
Widgets, die das Übertragen unterstützen, beginnen nach ihrer Erstellung automatisch damit. Zum Beispiel überträgt das Widget *Graph* sofort Daten zum Zeitraum an zuhörende Widgets.



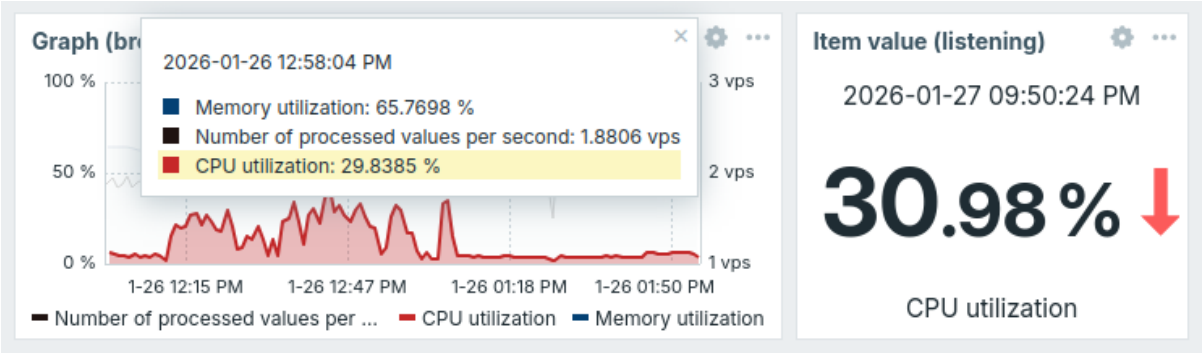
Widgets, die die Auswahl von Entitäten unterstützen, übertragen automatisch Daten für die erste verfügbare Entität. Zum Beispiel überträgt das Widget *Item navigator* Daten für den ersten Datenpunkt in seiner Datenpunktliste. Wenn ein anderer Datenpunkt ausgewählt wird, überträgt es Daten für diesen Datenpunkt. Beim Überfahren mit der Maus wird der Datenpunkt hellblau hervorgehoben; bei Auswahl wird er gelb hervorgehoben.



Das Widget *Map* verhält sich ähnlich und überträgt Daten für das erste Element, das dem oberen linken Rand des sichtbaren Bereichs des Widgets am nächsten liegt (beim Widget *Geomap* am nächsten zur Mitte). Wenn ein anderes Element ausgewählt wird, überträgt es Daten für dieses Element. Beim Überfahren mit der Maus wird das Element hellblau hervorgehoben; bei Auswahl wird es dunkelblau hervorgehoben.

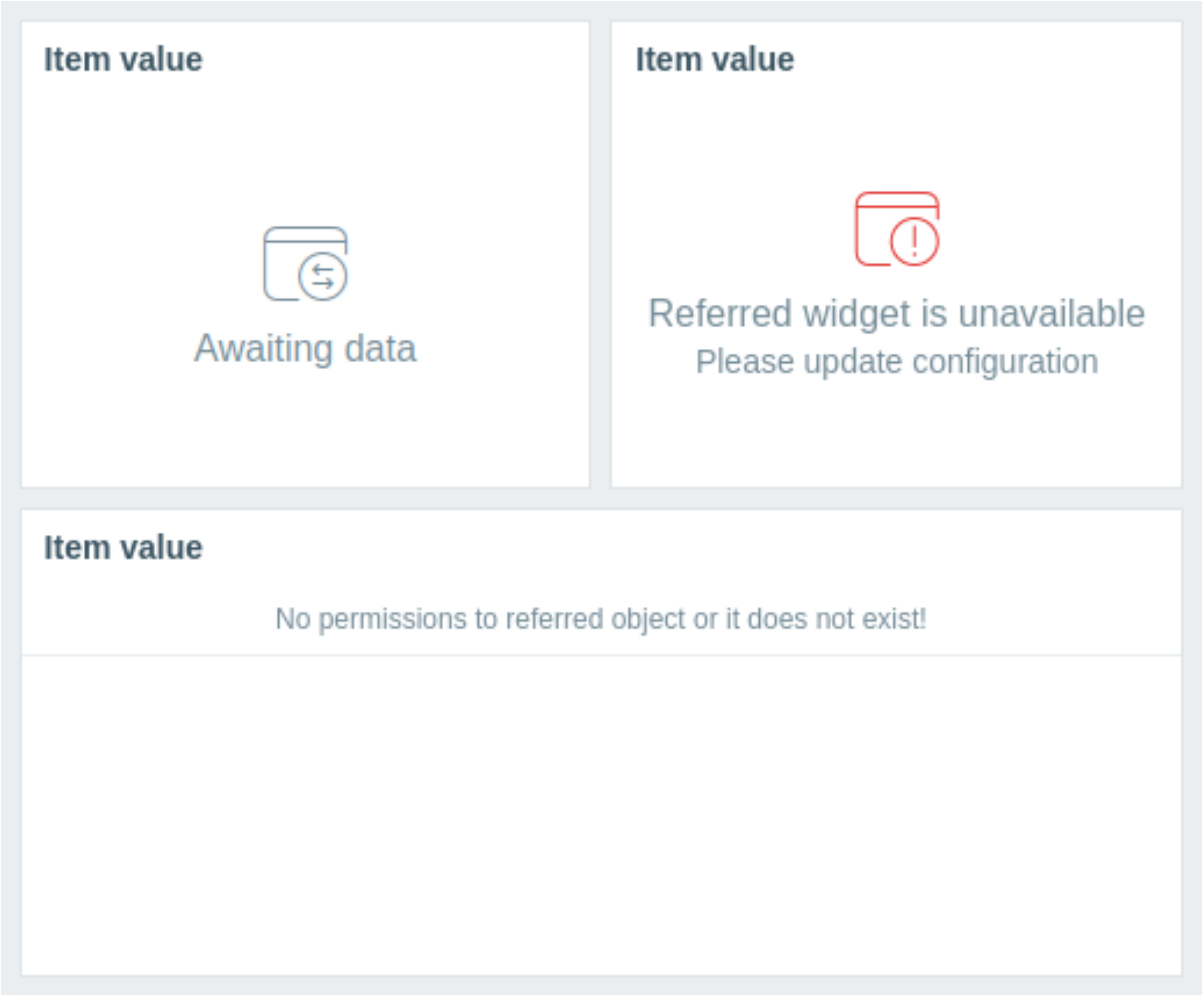


Widgets wie *Graph* und *Scatter plot* können Datenpunkte übertragen, wenn in ihren Tooltips ein Datenpunkt ausgewählt wird:



Widgets haben außerdem ein spezifisches Verhalten, wenn sie auf Daten anderer Widgets **reagieren**:

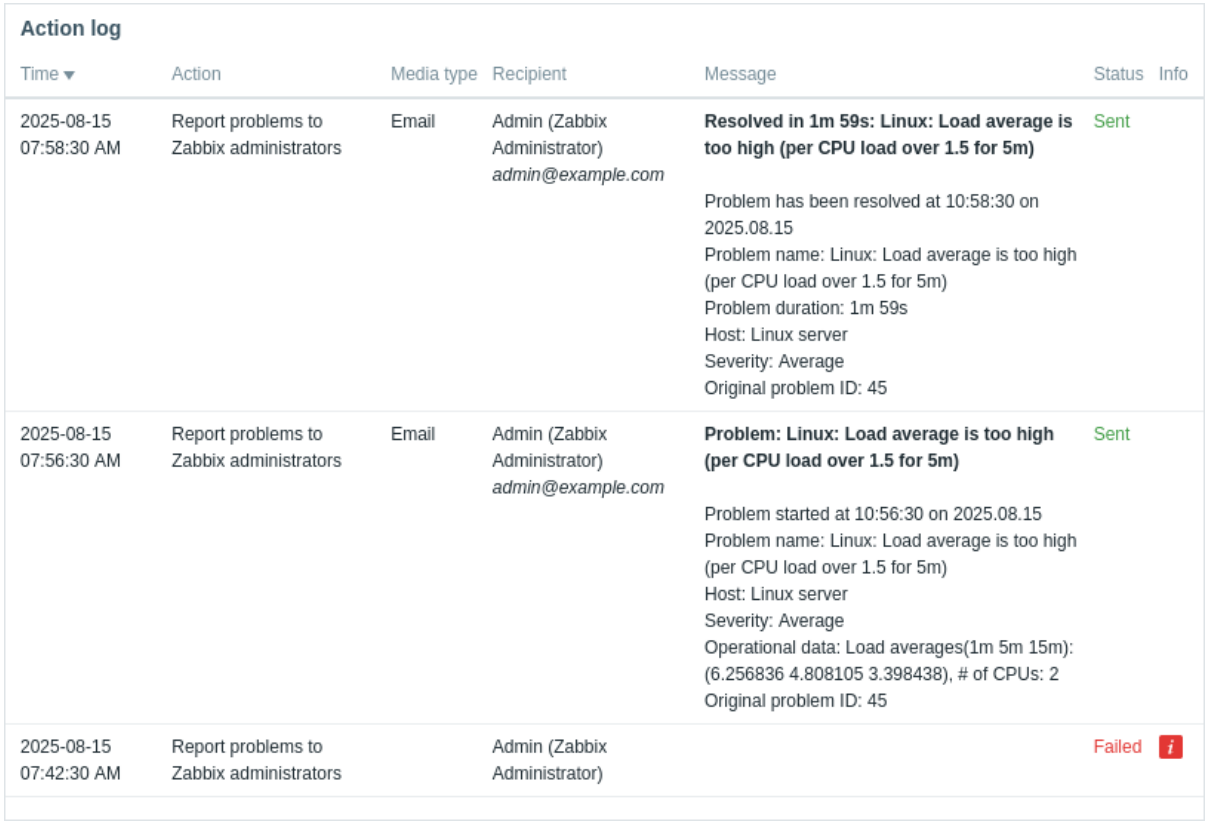
- Wenn das Widget der Datenquelle keine Daten überträgt, wechselt das zuhörende Widget in den Status *Awaiting data*.
- Wenn das Widget der Datenquelle gelöscht, durch ein inkompatibles Widget ersetzt oder auf eine andere Dashboard-Seite verschoben wurde, wechselt das zuhörende Widget in den Status *Referred widget is unavailable*.
- Wenn dem angegebenen Host in der Datenquelle (Widget oder Dashboard) die im zuhörenden Widget konfigurierte Entität (Datenpunkt, Graph, Map usw.) fehlt oder wenn dem Benutzer die Berechtigungen für den Zugriff auf den Host fehlen, zeigt das zuhörende Widget die folgende Meldung an: *"No permissions to referred object or it does not exist!"*



Aktionsprotokoll

Übersicht

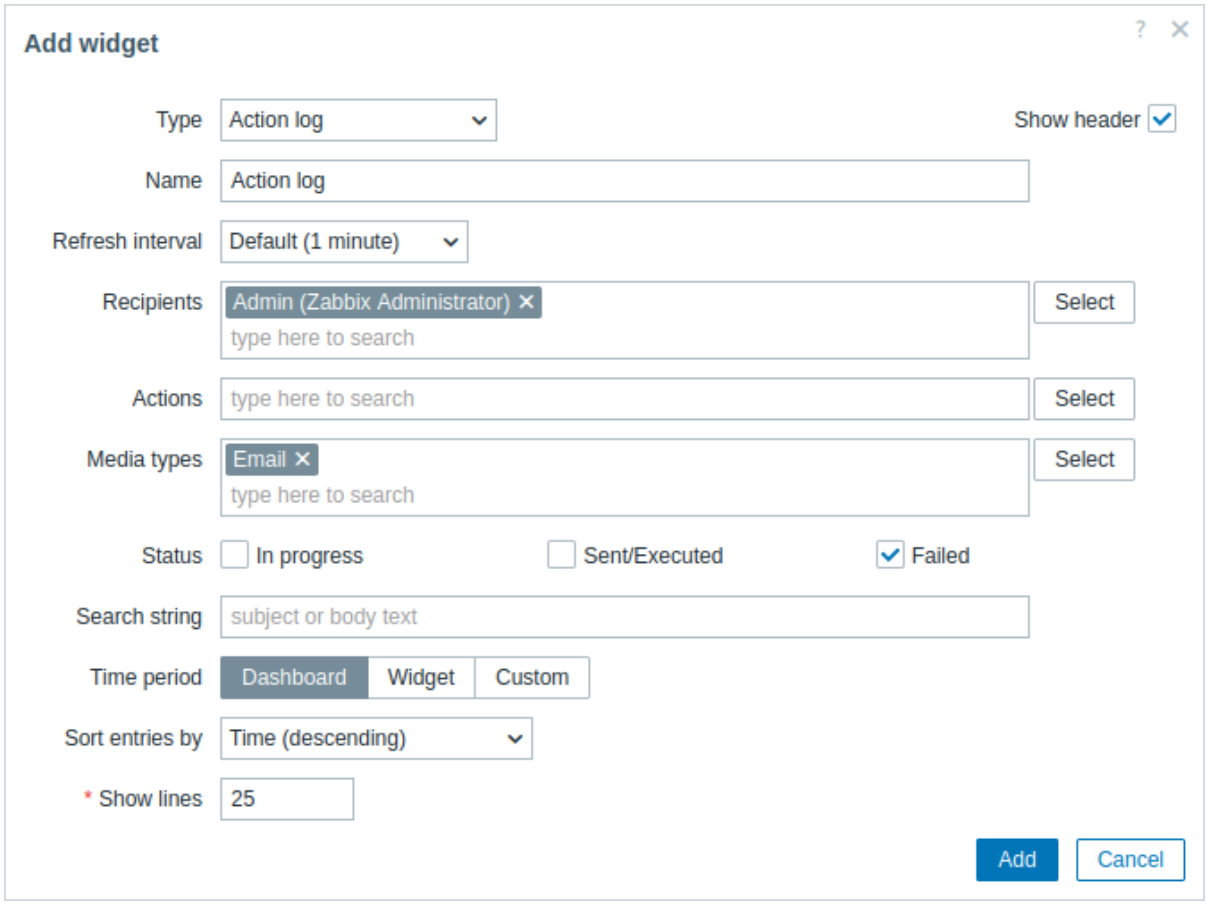
Das Widget *Aktionsprotokoll* zeigt Details zu Operationen (Benachrichtigungen, Remote-Befehle) an, die innerhalb einer Aktion ausgeführt wurden. Das detaillierte Protokoll unterstützt bei Audits, der Fehlerbehebung und der Überwachung der Ausführung jeder Aktion.



Es zeigt dieselben Daten wie *Berichte* > *Aktionsprotokoll* an und kann bis zu 1000 Einträge anzeigen.

Konfiguration

Wählen Sie zur Konfiguration *Aktionsprotokoll* als Typ aus:



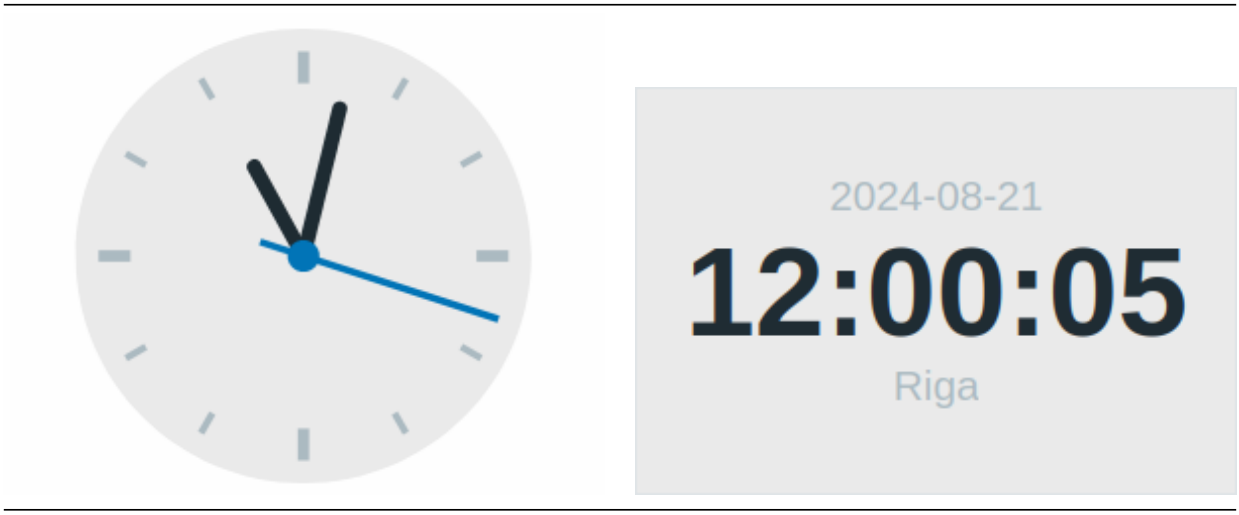
Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

Empfänger	Filtern Sie Einträge nach Empfängern. Dieses Feld verfügt über eine Autovervollständigung. Wenn Sie also beginnen, den Namen eines Empfängers einzugeben, wird eine Dropdown-Liste mit passenden Empfängern angeboten. Wenn keine Empfänger ausgewählt sind, werden Details zu Aktionsoperationen für alle Empfänger angezeigt.
Aktionen	Filtern Sie Einträge nach Aktionen. Dieses Feld verfügt über eine Autovervollständigung. Wenn Sie also beginnen, den Namen einer Aktion einzugeben, wird eine Dropdown-Liste mit passenden Aktionen angeboten. Wenn keine Aktionen ausgewählt sind, werden Details zu Aktionsoperationen für alle Aktionen angezeigt.
Medientypen	Filtern Sie Einträge nach Medientypen. Dieses Feld verfügt über eine Autovervollständigung. Wenn Sie also beginnen, den Namen eines Medientyps einzugeben, wird eine Dropdown-Liste mit passenden Medientypen angeboten. Wenn keine Medientypen ausgewählt sind, werden Details zu Aktionsoperationen für alle Medientypen angezeigt.
Status	Aktivieren Sie das Kontrollkästchen, um Einträge nach dem jeweiligen Status zu filtern: **In Bearbeitung** - Aktionsoperationen, die in Bearbeitung sind, werden angezeigt; **Gesendet/Ausgeführt** - Aktionsoperationen, die eine Benachrichtigung gesendet haben oder ausgeführt wurden, werden angezeigt; **Fehlgeschlagen** - fehlgeschlagene Aktionsoperationen werden angezeigt.
Suchzeichenfolge	Filtern Sie Einträge nach dem Inhalt der Nachricht/des Remote-Befehls. Wenn Sie hier eine Zeichenfolge eingeben, werden nur die Aktionsoperationen angezeigt, deren Nachricht/Remote-Befehl die eingegebene Zeichenfolge enthält. Makros werden nicht aufgelöst.
Zeitraum	Filtern Sie Einträge nach Zeitraum. Wählen Sie die Datenquelle für den Zeitraum aus: **Dashboard** - den Zeitraumauswahl des Dashboards verwenden; **Widget** - ein kompatibles Widget verwenden (im Parameter *Widget* festgelegt); **Benutzerdefiniert** - einen benutzerdefinierten Zeitraum verwenden, der in den Parametern *Von* und *Bis* festgelegt ist; falls gesetzt, wird in der oberen rechten Ecke des Widgets ein Uhrsymbol angezeigt, das beim Überfahren mit der Maus auf die eingestellte Zeit hinweist. Beachten Sie, dass kompatible Widgets unabhängig von der Konfiguration des *Zeitraums* des Widgets weiterhin als Datenquelle für den Zeitraum verwendet werden können.
Widget	Geben Sie ein kompatibles Widget als Datenquelle für den Zeitraum ein oder wählen Sie es aus. Dieser Parameter ist verfügbar, wenn *Zeitraum* auf "Widget" gesetzt ist.
Von	Geben Sie den Beginn des Zeitraums ein oder wählen Sie ihn aus. Die Syntax für relative Zeitangaben (`now`, `now/d`, `now/w-1w` usw.) wird unterstützt. Dieser Parameter ist verfügbar, wenn *Zeitraum* auf "Benutzerdefiniert" gesetzt ist.
Bis	Geben Sie das Ende des Zeitraums ein oder wählen Sie es aus. Die Syntax für relative Zeitangaben (`now`, `now/d`, `now/w-1w` usw.) wird unterstützt. Dieser Parameter ist verfügbar, wenn *Zeitraum* auf "Benutzerdefiniert" gesetzt ist.
Einträge sortieren nach	Sortieren Sie Einträge nach: **Zeit** (absteigend oder aufsteigend); **Typ** (absteigend oder aufsteigend); **Status** (absteigend oder aufsteigend); **Empfänger** (absteigend oder aufsteigend).
Zeilen anzeigen	Legen Sie fest, wie viele Zeilen des Aktionsprotokolls im Widget angezeigt werden.

Uhr

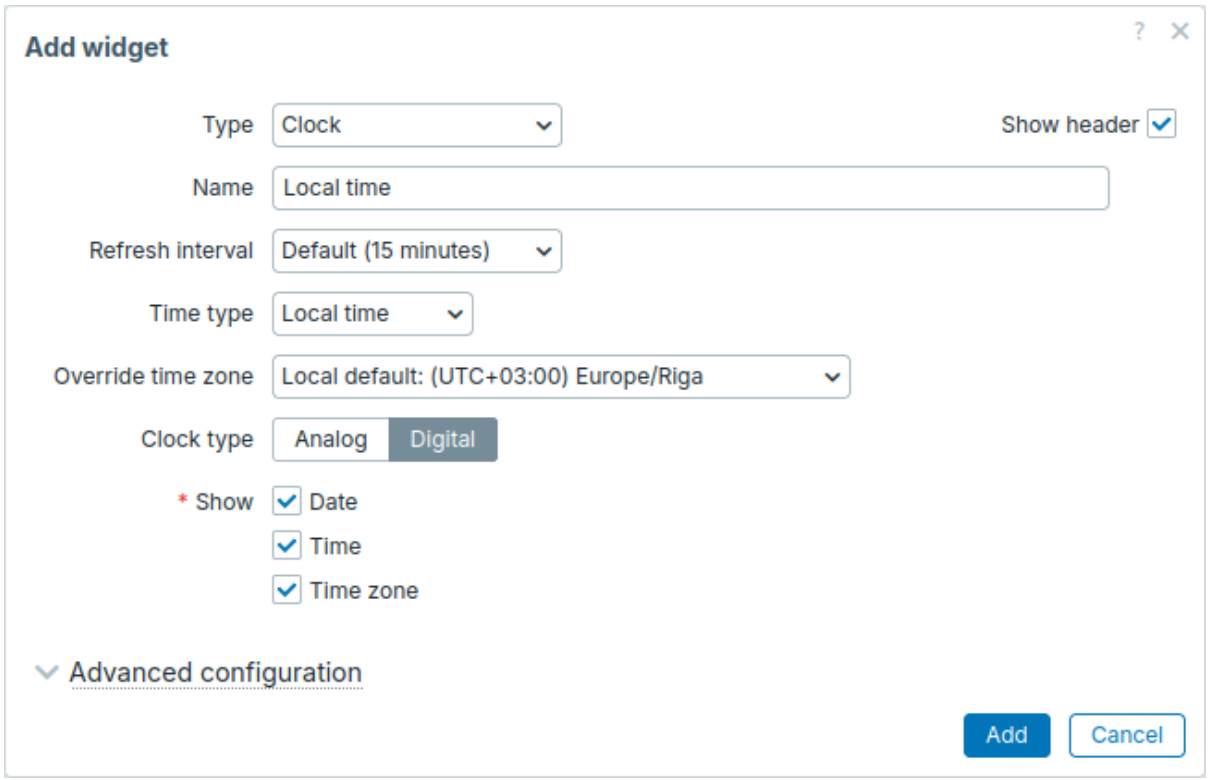
Übersicht

Das Widget *Uhr* zeigt die lokale Zeit, die Server-Zeit oder die Host-Zeit im analogen oder digitalen Format an.



Konfiguration

Wählen Sie zur Konfiguration *Uhr* als Typ aus:



Zusätzlich zu den Parametern, die für alle Widgets allgemein gelten, können Sie die folgenden spezifischen Optionen festlegen:

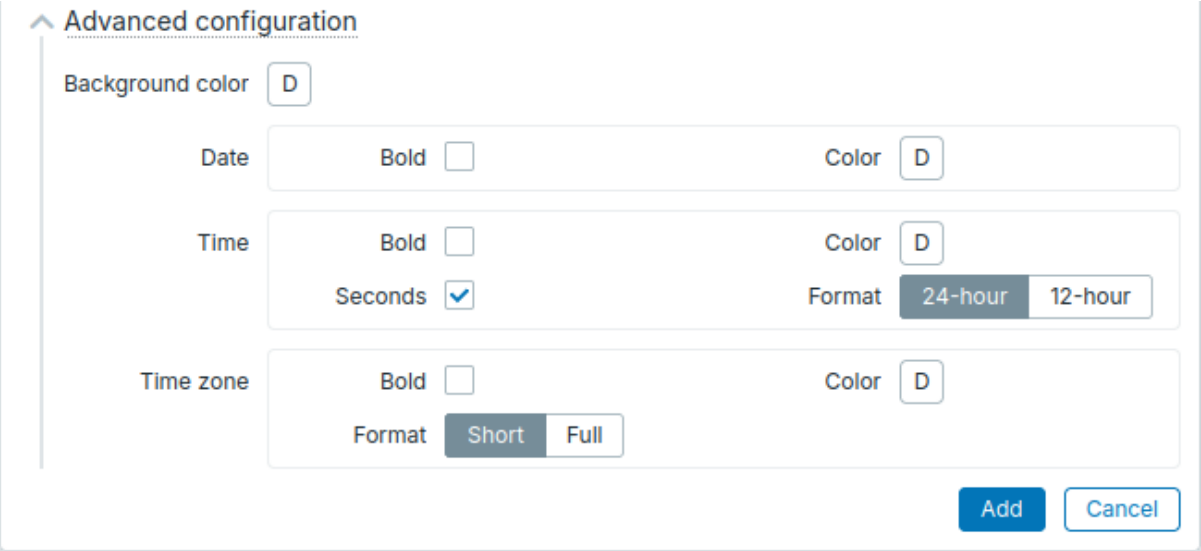
Zeittyp	Wählen Sie die Zeitquelle aus: **Lokale Zeit** – Systemzeit des Geräts, auf dem der Browser ausgeführt wird; **Serverzeit** – global oder für den Zabbix-Benutzer festgelegte Zeitzone; **Hostzeit** – aus einem Datenpunkt eines Hosts abgerufene Zeit.
Datenpunkt	Wählen Sie den Datenpunkt mit dem Schlüssel `system.localtime[local]` aus, um die Hostzeit anzuzeigen. Dieser Parameter ist verfügbar, wenn *Zeittyp* auf „Hostzeit“ gesetzt ist.
Uhrtyp	Wählen Sie den Uhrtyp aus (analog oder digital).
Zeitzone überschreiben	Wählen Sie die Zeitzone aus. Dieser Parameter ist für die Zeittypen *Lokale Zeit* und *Serverzeit* verfügbar.
Anzeigen	Wählen Sie eine oder mehrere Informationseinheiten (Datum, Uhrzeit, Zeitzone) aus, die in der Digitaluhr angezeigt werden sollen. Die Größe der Einheiten wird automatisch an die angezeigten Einheiten, die Widgetgröße und die Anzeigeskalierung angepasst. Dieser Parameter ist verfügbar, wenn *Uhrtyp* auf „Digital“ gesetzt ist.

Erweiterte Konfiguration	Klicken Sie auf die Überschrift *Erweiterte Konfiguration*, um Optionen für die erweiterte Konfiguration der Digitaluhr anzuzeigen. Dieser Abschnitt ist verfügbar, wenn *Uhrtyp* auf „Digital“ gesetzt ist.

Erweiterte Konfiguration

Erweiterte Konfigurationsoptionen sind im aufklappbaren Abschnitt *Erweiterte Konfiguration* verfügbar, und zwar nur für die Elemente, die im Feld *Anzeigen* ausgewählt sind (siehe oben).

Zusätzlich können Sie mit der erweiterten Konfiguration die Hintergrundfarbe für das gesamte Widget ändern.

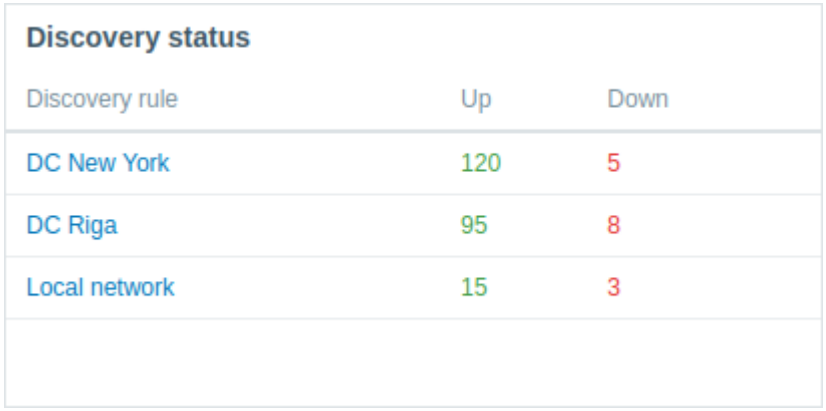


Hintergrundfarbe		Wählen Sie die Hintergrundfarbe aus der Farbauswahl aus. D steht für die Standardfarbe (abhängig vom Frontend-Theme). Um den Standardwert wiederherzustellen, klicken Sie in der Farbauswahl auf die Schaltfläche *Standard verwenden*.
Datum		Dieser Abschnitt ist verfügbar, wenn das Kontrollkästchen *Datum* für den Parameter *Anzeigen* ausgewählt ist.
	Fett	Aktivieren Sie das Kontrollkästchen, um das Datum fett anzuzeigen.
	Farbe	Wählen Sie die Datumsfarbe aus der Farbauswahl aus. D steht für die Standardfarbe (abhängig vom Frontend-Theme). Um den Standardwert wiederherzustellen, klicken Sie in der Farbauswahl auf die Schaltfläche *Standard verwenden*.
Uhrzeit		Dieser Abschnitt ist verfügbar, wenn das Kontrollkästchen *Uhrzeit* für den Parameter *Anzeigen* ausgewählt ist.
	Fett	Aktivieren Sie das Kontrollkästchen, um die Uhrzeit fett anzuzeigen.
	Farbe	Wählen Sie die Farbe der Uhrzeit aus der Farbauswahl aus. D steht für die Standardfarbe (abhängig vom Frontend-Theme). Um den Standardwert wiederherzustellen, klicken Sie in der Farbauswahl auf die Schaltfläche *Standard verwenden*.
	Sekunden	Aktivieren Sie das Kontrollkästchen, um Sekunden anzuzeigen. Andernfalls werden nur Stunden und Minuten angezeigt.
	Format	Wählen Sie aus, ob die Uhrzeit im 24-Stunden- oder 12-Stunden-Format angezeigt werden soll.
Zeitzone		Dieser Abschnitt ist verfügbar, wenn das Kontrollkästchen *Zeitzone* für den Parameter *Anzeigen* ausgewählt ist.
	Fett	Aktivieren Sie das Kontrollkästchen, um die Zeitzone fett anzuzeigen.
	Farbe	Wählen Sie die Farbe der Zeitzone aus der Farbauswahl aus. D steht für die Standardfarbe (abhängig vom Frontend-Theme). Um den Standardwert wiederherzustellen, klicken Sie in der Farbauswahl auf die Schaltfläche *Standard verwenden*.
	Format	Wählen Sie aus, ob die Zeitzone im Kurzformat (z. B. `New York`) oder im vollständigen Format (z. B. `(UTC-04:00) America/New York`) angezeigt werden soll.

Erkennungsstatus

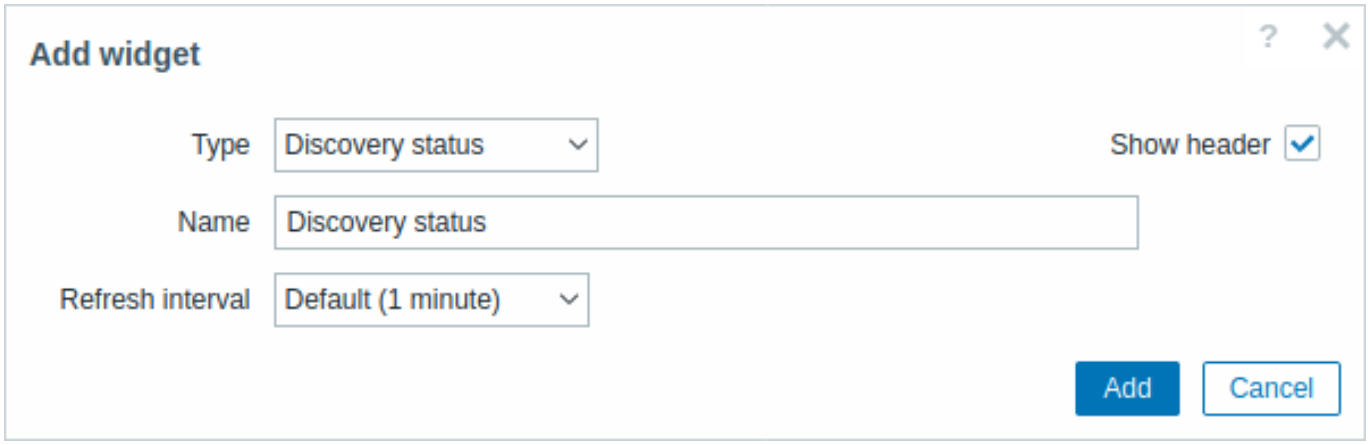
Übersicht

Das Widget *Discovery status* zeigt den Status von Geräten an, die durch aktivierte Regeln für die Netzwerkdiscovery erkannt wurden. Es hilft dabei zu überwachen, ob die Netzwerkdiscovery korrekt funktioniert und wie viele Geräte erreichbar oder nicht erreichbar sind.



Konfiguration

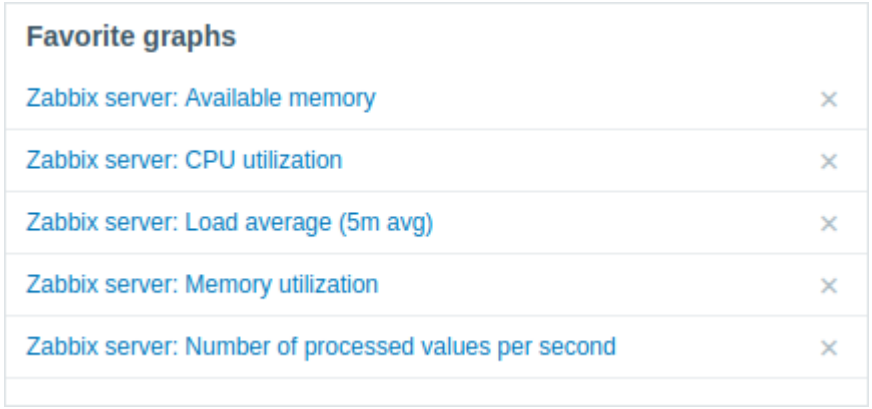
Alle Konfigurationsparameter für dieses Widget sind für alle Widgets gemeinsam.



Favoriten-Diagramme

Übersicht

Das Widget *Favorite graphs* zeigt eine alphabetische Liste von Verknüpfungen zu Diagrammen an, die vom aktuellen Benutzer als Favoriten markiert wurden.



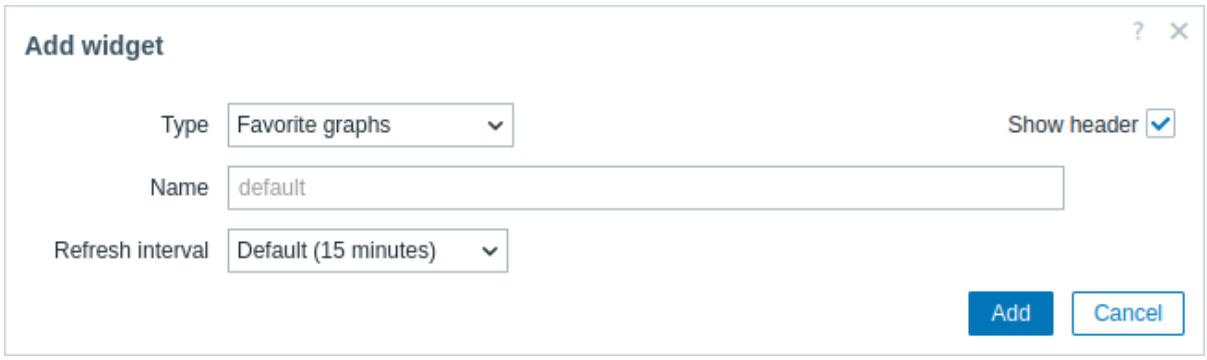
Durch Klicken auf x wird das Diagramm aus der Liste und aus den Favoriten entfernt.

Diagramme werden als Favoriten markiert, indem beim Anzeigen von Datenpunkt-Diagrammen unter *Monitoring* > *Latest data* auf die Schaltfläche *Add to favorites* geklickt wird.



Konfiguration

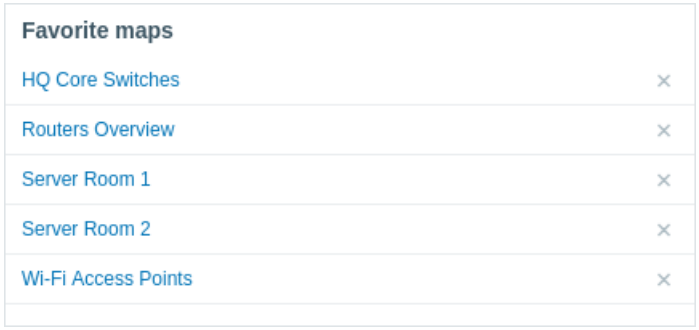
Alle Konfigurationsparameter für dieses Widget sind für alle Widgets gemeinsam.



Favoritenkarten

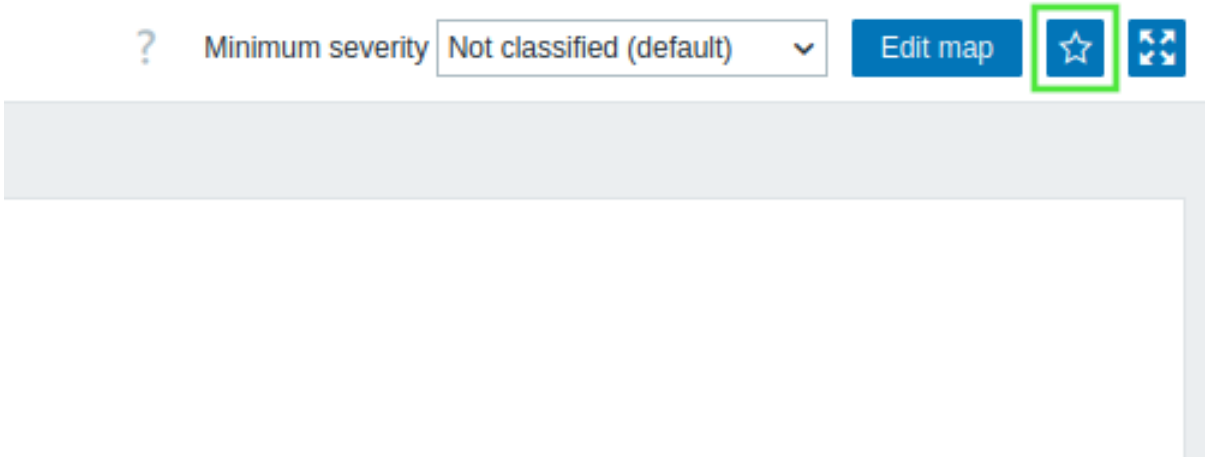
Übersicht

Das Widget *Bevorzugte Karten* zeigt eine alphabetisch sortierte Liste von Verknüpfungen zu Karten an, die vom aktuellen Benutzer als Favoriten markiert wurden.



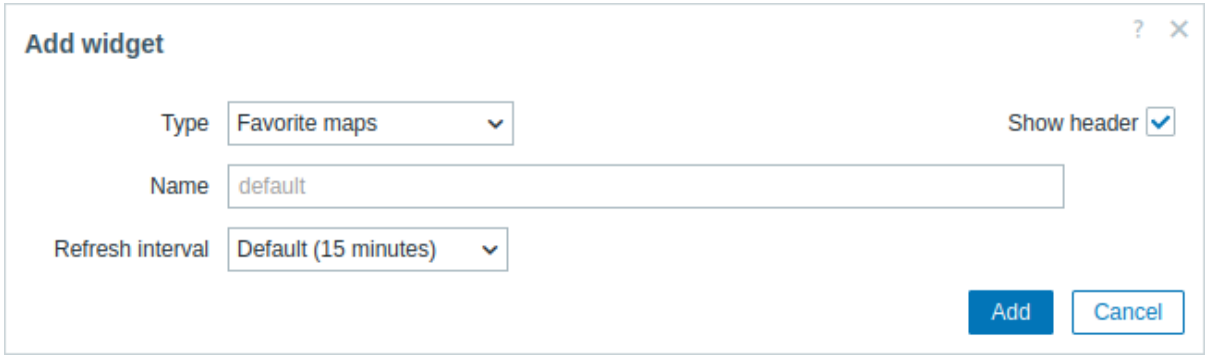
Durch Klicken auf x wird die Karte aus den Favoriten entfernt.

Karten werden als Favoriten markiert, indem beim Anzeigen von Karten in *Monitoring* > *Karten* auf die Schaltfläche  *Zu Favoriten hinzufügen* geklickt wird.



Konfiguration

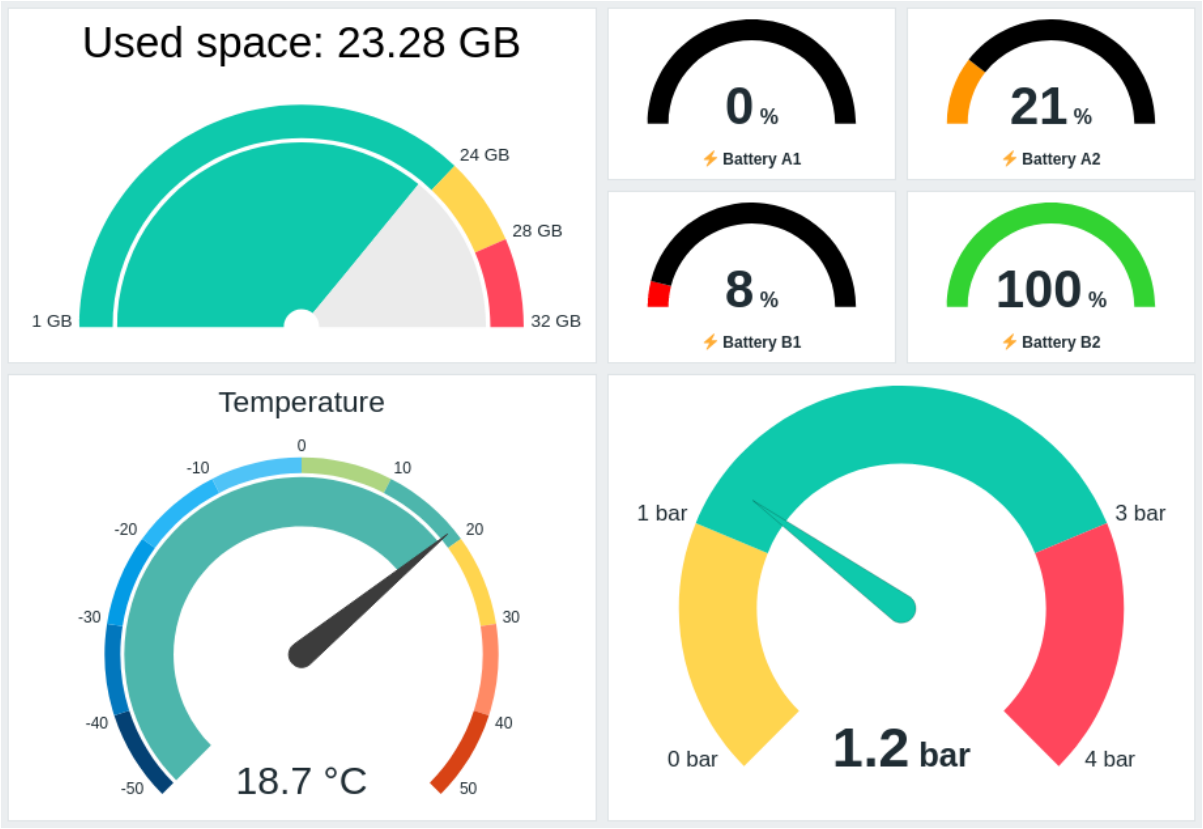
Alle Konfigurationsparameter für dieses Widget sind für alle Widgets gemeinsam.



Gauge

Übersicht

Das Widget *Messgerät* zeigt den Wert eines einzelnen numerischen Datenpunkts als Messgerät an. Es eignet sich, um wichtige Metriken im Blick zu behalten, Schwellenwerte zu visualisieren und plötzliche Änderungen zu erkennen.



Sie können das Widget so konfigurieren, dass es Folgendes anzeigt:

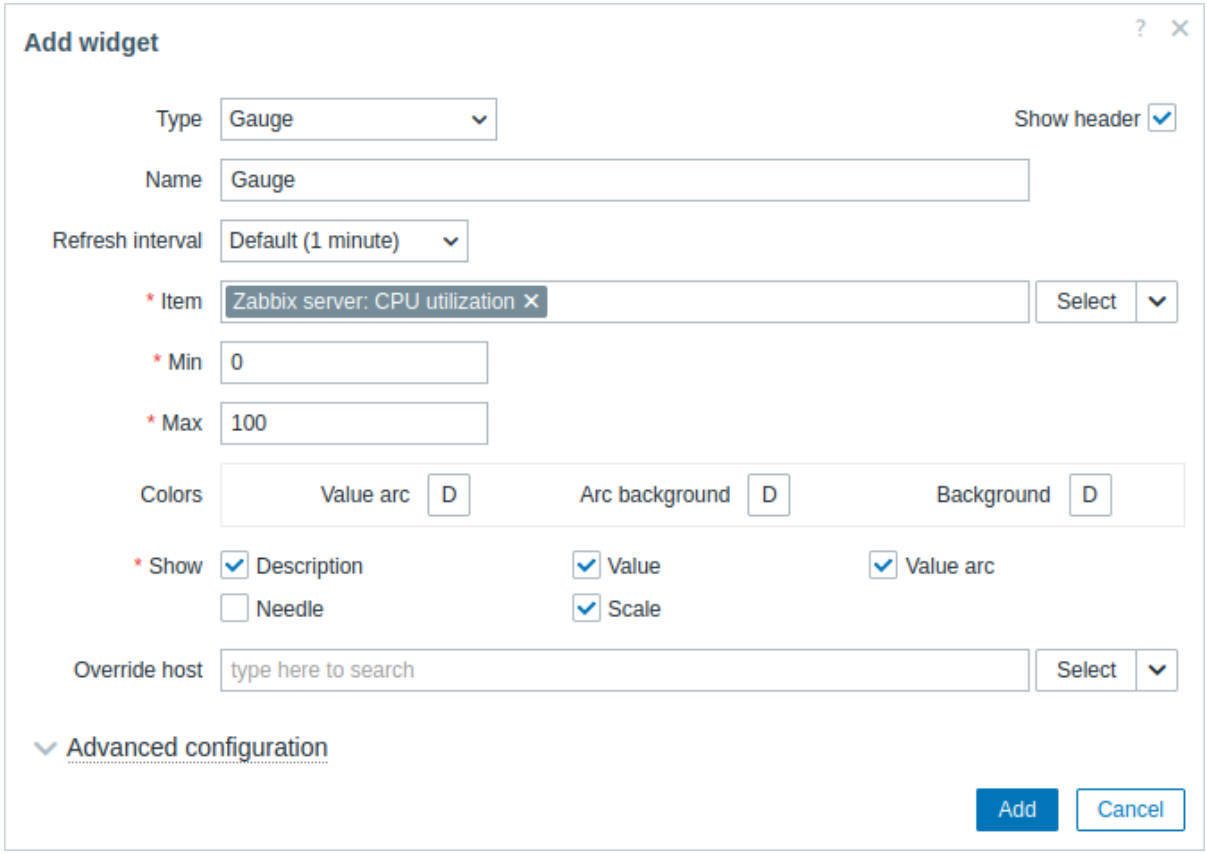
- Beschreibung des Datenpunkts (Verwendeter Speicherplatz: {ITEM.LASTVALUE}, Temperatur)
- Wert und Einheiten des Datenpunkts (21 %, 18.7 °C, 1.2 bar)
- Messgerätskala (1GB/24GB/28GB/32GB, -50/-40/-30/usw.)
- Messgerätbogen (Wertbogen und Schwellenwertbogen)
- Messgerätnadel

Wenn Sie auf das Widget klicken, wird ein einfaches Diagramm für den Datenpunkt geöffnet.

Die im Widget *Messgerät* angezeigten Informationen können als PNG-Bild heruntergeladen werden. Wählen Sie dazu die Option *Bild herunterladen* im Menü *Aktionen* des Widgets ⋯, das angezeigt wird, wenn Sie den Mauszeiger über das Widget bewegen. Abhängig von Ihren Browsereinstellungen wird das Bild entweder automatisch im Ordner `Downloads` gespeichert oder ein Dialogfeld *Speichern* geöffnet, in dem Sie den Speicherort auswählen können.

Konfiguration

Wählen Sie zur Konfiguration *Gauge* als Typ aus:



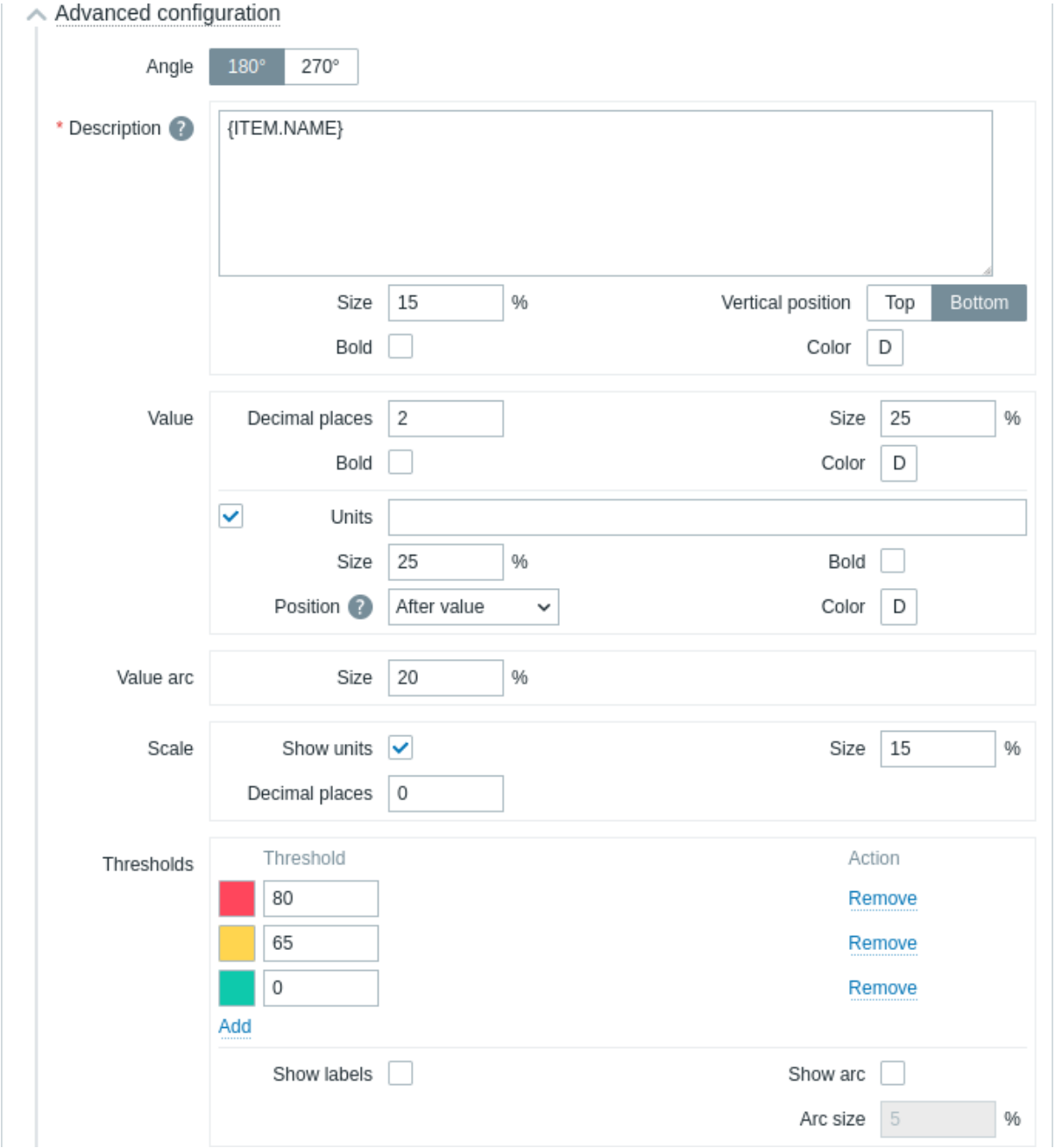
Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

Datenpunkt	Wählen Sie den Datenpunkt aus. Alternativ können Sie ein kompatibles Widget als Datenquelle für Datenpunkte auswählen. Dieses Feld verfügt über eine Autovervollständigung; wenn Sie also beginnen, den Namen eines Datenpunkts einzugeben, wird eine Dropdown-Liste mit passenden Datenpunkten angeboten. Beachten Sie, dass Sie nur Datenpunkte auswählen können, die numerische Daten zurückgeben (mit Ausnahme von Binärdaten).
Min	Geben Sie den Mindestwert der Anzeige ein. Suffixe (zum Beispiel „1d“, „2w“, „4K“, „8G“) werden unterstützt. Wertzuordnungen werden unterstützt.
Max	Geben Sie den Höchstwert der Anzeige ein. Suffixe (zum Beispiel „1d“, „2w“, „4K“, „8G“) werden unterstützt. Wertzuordnungen werden unterstützt.
Farben	Wählen Sie die Farbe in der Farbauswahl aus: **Wertbogen** - wählen Sie die Farbe des Wertbogens der Anzeige aus; **Bogenhintergrund** - wählen Sie die Hintergrundfarbe des Wertbogens der Anzeige und des Schwellenwertbogens der Anzeige aus; **Hintergrund** - wählen Sie die Hintergrundfarbe des Widgets aus. „D“ steht für die Standardfarbe, die vom Frontend-Theme abhängt. Wenn *Schwellenwerte* festgelegt sind, hängt die Standardfarbe für **Wertbogen** von der Schwellenwertfarbe ab. Um zur Standardfarbe zurückzukehren, klicken Sie in der Farbauswahl auf die Schaltfläche *Use default*.
Anzeigen	Aktivieren Sie das Kontrollkästchen, um das jeweilige Element der Anzeige anzuzeigen – Beschreibung, Wert, Wertbogen, Zeiger, Skala (der Mindest- und Höchstwert der Anzeige am Anfang und Ende des Anzeigebogens). Deaktivieren Sie es, um das Element auszublenden. Es muss mindestens ein Element ausgewählt sein. Beachten Sie, dass der Zeiger und die Skala der Anzeige angezeigt werden können, wenn der Wertbogen der Anzeige oder der Schwellenwertbogen der Anzeige (siehe Optionen der erweiterten Konfiguration) angezeigt wird. Beachten Sie außerdem, dass der Wert unter dem Zeiger platziert wird, wenn der Zeiger angezeigt wird; wenn der Zeiger ausgeblendet ist, wird der Wert am unteren Rand des Anzeigebogens ausgerichtet.

Host überschreiben	Wählen Sie ein kompatibles Widget oder den Dashboard-Host-Auswahlfilter als Datenquelle für Hosts aus. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Erweiterte Konfiguration	Klicken Sie auf die Überschrift *Erweiterte Konfiguration*, um die Optionen der erweiterten Konfiguration anzuzeigen. Hier können Sie auch die im Feld *Anzeigen* ausgewählten Elemente der Anzeige anpassen.

Erweiterte Konfiguration

Erweiterte Konfigurationsoptionen sind im einklappbaren Abschnitt *Erweiterte Konfiguration* verfügbar:

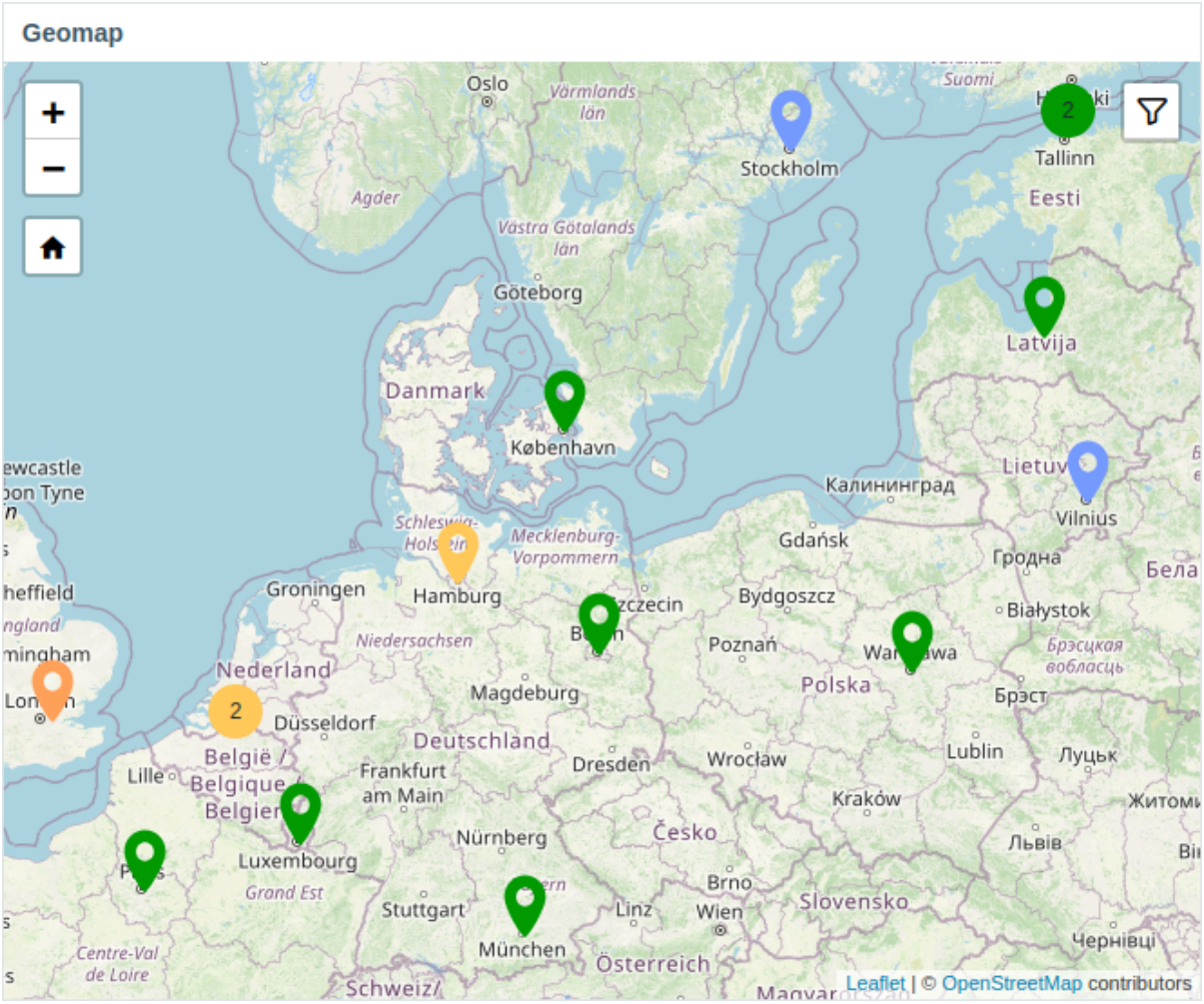


Winkel	Wählen Sie den Winkel der Anzeige (180° oder 270°).
Beschreibung	
Beschreibung	Geben Sie die Beschreibung des Datenpunkts ein. Diese Beschreibung kann den Standardnamen des Datenpunkts überschreiben. Mehrzeilige Beschreibungen werden unterstützt. Eine Kombination aus Text und unterstützten Makros ist möglich. {HOST.*}, {ITEM.*}, {INVENTORY.*} und Benutzermakros werden unterstützt.

Größe	Geben Sie die Höhe der Schriftgröße für die Beschreibung des Datenpunkts ein (in Prozent, relativ zur Gesamthöhe des Widgets).
Vertikale Position	Wählen Sie die vertikale Position der Beschreibung des Datenpunkts (oben oder unten, relativ zum Bogen der Anzeige).
Fett	Aktivieren Sie das Kontrollkästchen, um die Beschreibung des Datenpunkts fett darzustellen.
Farbe	Wählen Sie die Farbe der Beschreibung des Datenpunkts in der Farbauswahl aus. ”D” steht für die Standardfarbe, die vom Frontend-Theme abhängt. Um zur Standardfarbe zurückzukehren, klicken Sie in der Farbauswahl auf die Schaltfläche *Standard verwenden*.
Wert	
Dezimalstellen	Geben Sie die Anzahl der Dezimalstellen ein, die zusammen mit dem Wert angezeigt werden sollen. Diese Option betrifft nur Datenpunkte, die Daten vom Typ numeric (float) zurückgeben.
Größe	Geben Sie die Höhe der Schriftgröße für den Wert ein (in Prozent, relativ zur Höhe des Bogens der Anzeige).
Fett	Aktivieren Sie das Kontrollkästchen, um den Wert fett darzustellen.
Farbe	Wählen Sie die Farbe des Werts in der Farbauswahl aus. ”D” steht für die Standardfarbe, die vom Frontend-Theme abhängt. Um zur Standardfarbe zurückzukehren, klicken Sie in der Farbauswahl auf die Schaltfläche *Standard verwenden*.
Einheiten	
Einheiten	Aktivieren Sie das Kontrollkästchen, um Einheiten zusammen mit dem Wert des Datenpunkts anzuzeigen. Wenn Sie einen Einheitsnamen eingeben, überschreibt dieser die in der Datenpunkt-Konfiguration festgelegten Einheiten.
Größe	Geben Sie die Höhe der Schriftgröße für die Einheiten des Datenpunkts ein (in Prozent, relativ zur Höhe des Bogens der Anzeige).
Fett	Aktivieren Sie das Kontrollkästchen, um die Einheiten des Datenpunkts fett darzustellen.
Position	Wählen Sie die Position der Einheiten des Datenpunkts (oberhalb, unterhalb, vor oder nach dem Wert des Datenpunkts). Diese Option wird für die folgenden zeitbezogenen Einheiten ignoriert: unixtime, uptime, s.
Farbe	Wählen Sie die Farbe der Einheiten des Datenpunkts in der Farbauswahl aus. ”D” steht für die Standardfarbe, die vom Frontend-Theme abhängt. Um zur Standardfarbe zurückzukehren, klicken Sie in der Farbauswahl auf die Schaltfläche *Standard verwenden*.
Wertbogen	
Bogengröße	Geben Sie die Höhe der Größe des Wertbogens der Anzeige ein (in Prozent, relativ zum Radius des Bogens der Anzeige).
Zeiger	
Farbe	Wählen Sie die Farbe des Zeigers der Anzeige in der Farbauswahl aus. ”D” steht für die Standardfarbe, die vom Frontend-Theme abhängt. Wenn *Schwellenwerte* festgelegt sind, hängt die Standardfarbe des Zeigers von der Farbe des Schwellenwerts ab. Um zur Standardfarbe zurückzukehren, klicken Sie in der Farbauswahl auf die Schaltfläche *Standard verwenden*.
Skala	
Einheiten anzeigen	Aktivieren Sie das Kontrollkästchen, um Einheiten zusammen mit dem Minimal- und Maximalwert der Anzeige anzuzeigen.
Größe	Geben Sie die Höhe der Schriftgröße für den Minimal- und Maximalwert der Anzeige ein (in Prozent, relativ zur Höhe des Bogens der Anzeige).
Dezimalstellen	Geben Sie die Anzahl der Dezimalstellen ein, die zusammen mit dem Minimal- und Maximalwert der Anzeige angezeigt werden sollen. Diese Option betrifft nur Datenpunkte, die Daten vom Typ numeric (float) zurückgeben.
Schwellenwerte	
Schwellenwerte	Klicken Sie auf *Hinzufügen*, um einen Schwellenwert hinzuzufügen, wählen Sie eine Schwellenwertfarbe in der Farbauswahl aus und geben Sie einen numerischen Wert an. Die Liste der Schwellenwerte wird beim Speichern in aufsteigender Reihenfolge sortiert. Beachten Sie, dass die als Schwellenwerte konfigurierten Farben nur für numerische Datenpunkte korrekt angezeigt werden. Suffixe (zum Beispiel ”1d”, ”2w”, ”4K”, ”8G”) werden unterstützt. Wertzuordnungen werden unterstützt.
Beschriftungen anzeigen	Aktivieren Sie das Kontrollkästchen, um Schwellenwerte als Beschriftungen auf der Skala der Anzeige anzuzeigen.
Bogen anzeigen	Aktivieren Sie das Kontrollkästchen, um den Schwellenwertbogen der Anzeige anzuzeigen.
Bogengröße	Geben Sie die Höhe der Größe des Schwellenwertbogens der Anzeige ein (in Prozent, relativ zum Radius des Bogens der Anzeige).

Übersicht

Das *Geomap*-Widget zeigt Hosts als Marker auf einer interaktiven, Leaflet-basierten Karte an, wobei die Markerfarben das jeweils schwerwiegendste Problem des Hosts anzeigen.

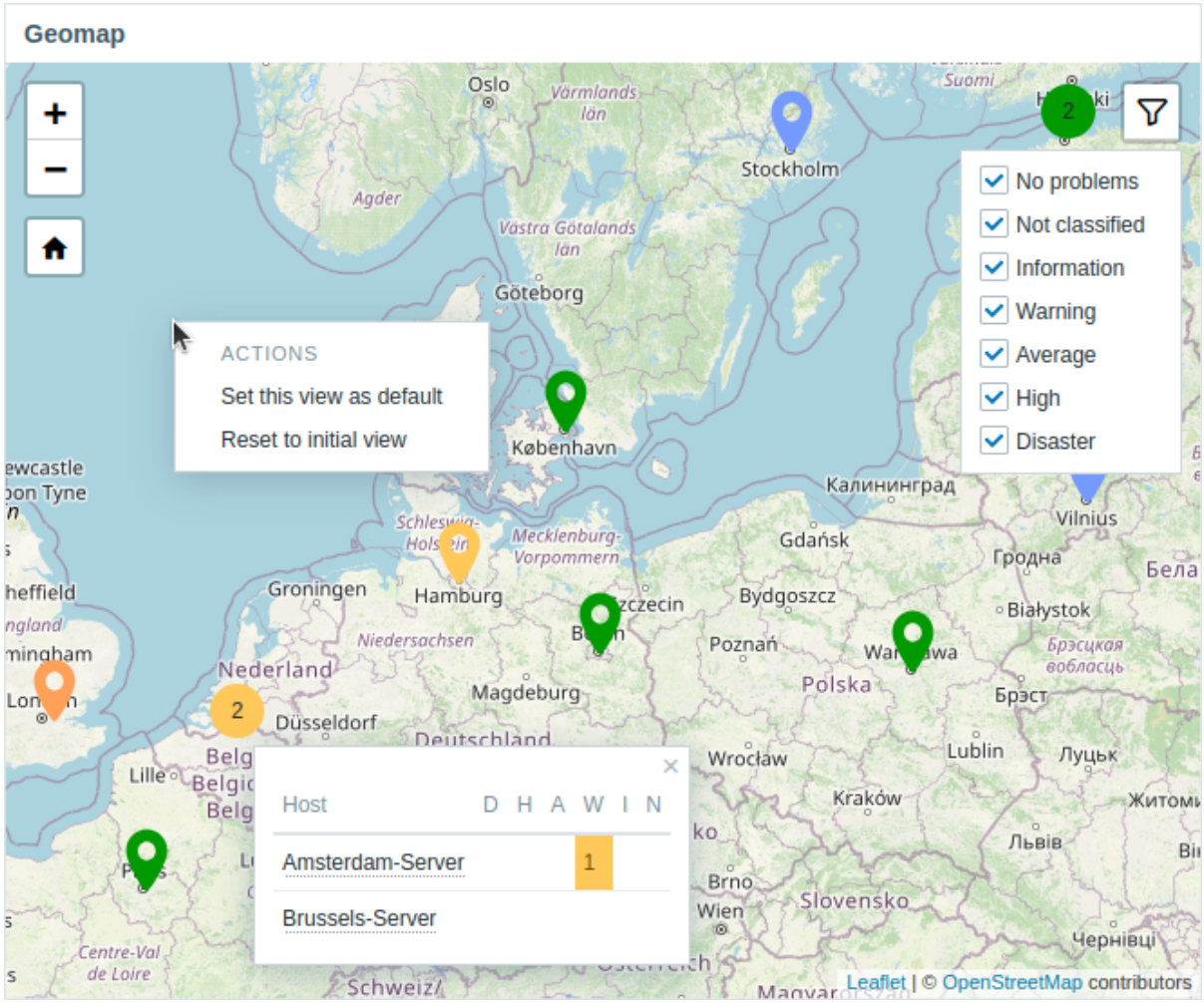


Note:
Zabbix bietet mehrere vordefinierte Anbieter für Kartenkacheln und ermöglicht es Ihnen, benutzerdefinierte Anbieter unter *Administration* > *General* > *Geographical maps* hinzuzufügen.

Standardmäßig zeigt das Widget alle aktivierten Hosts an, die in ihrem Inventar gültige Koordinaten haben (Breitengrad -90 bis 90, Längengrad -180 bis 180).

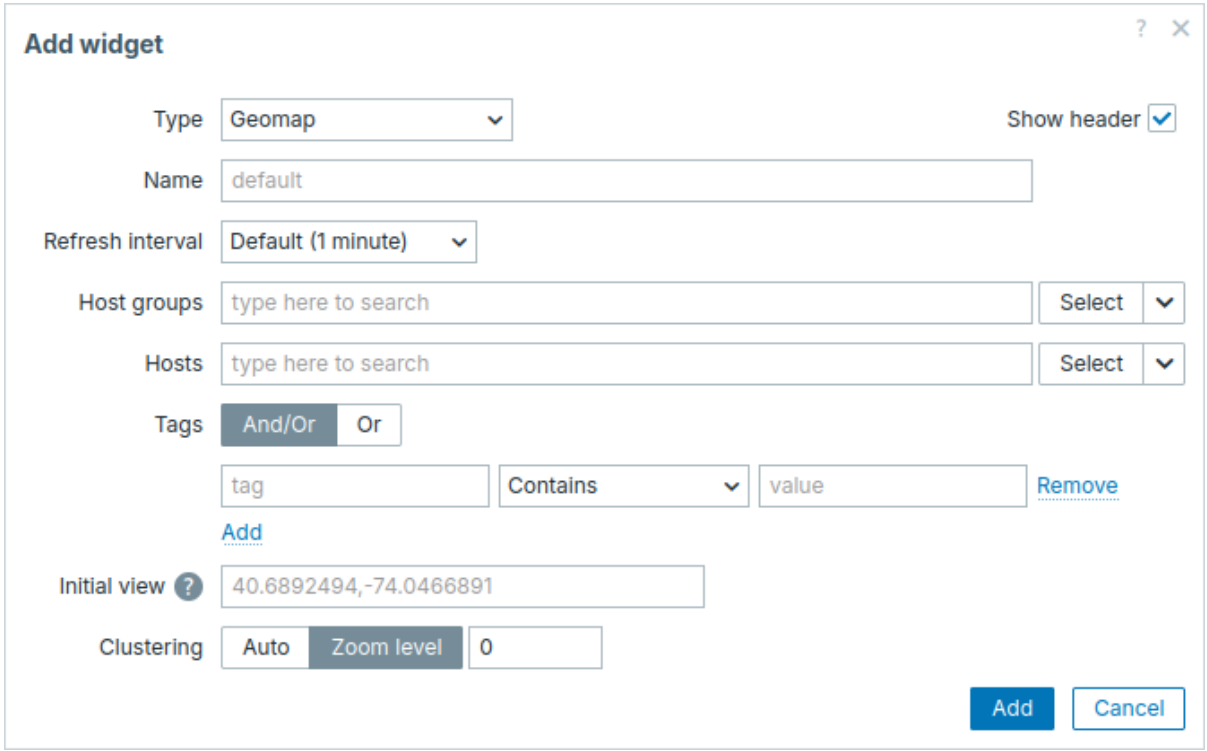
Verwenden Sie die folgenden Optionen, um mit dem Widget zu interagieren:

- Durch Klicken auf einen Marker werden der Hostname und ungelöste Probleme angezeigt; durch Klicken auf den Hostnamen wird das Host-Menü geöffnet.
- Filtern Sie Hosts nach Problemschwere über das Filtersymbol in der oberen rechten Ecke.
- Klicken Sie mit der rechten Maustaste auf die Karte, um die Standardansicht festzulegen oder zurückzusetzen; verwenden Sie das Home-Symbol, um zur Standardansicht zurückzukehren.



Konfiguration

Um das Widget hinzuzufügen, wählen Sie *Geomap* als Typ aus.



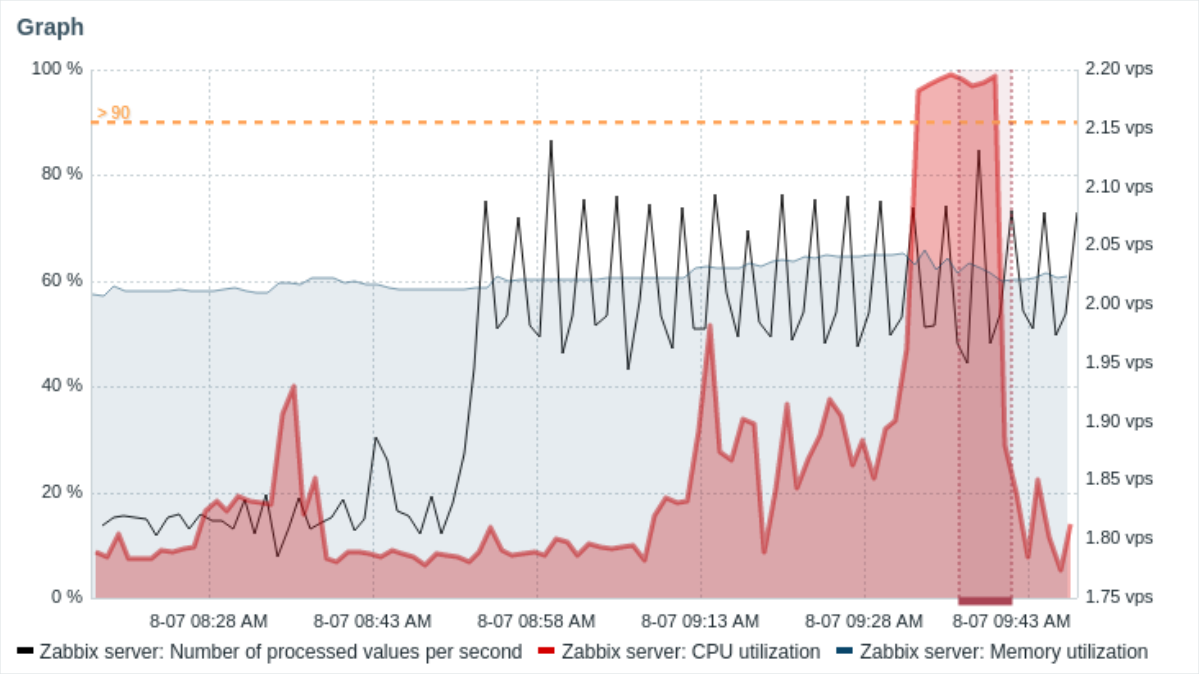
Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

Host-Gruppen	Wählen Sie die Host-Gruppen aus, die auf der Karte angezeigt werden sollen. Alternativ können Sie ein kompatibles Widget als Datenquelle für Host-Gruppen auswählen. Dieses Feld unterstützt die automatische Vervollständigung. Wenn Sie also beginnen, den Namen einer Gruppe einzugeben, wird eine Dropdown-Liste mit passenden Gruppen angeboten. Wenn in den Feldern *Host-Gruppen* und *Hosts* nichts ausgewählt ist, werden alle Hosts mit gültigen Koordinaten angezeigt. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Hosts	Wählen Sie die Hosts aus, die auf der Karte angezeigt werden sollen. Alternativ können Sie ein kompatibles Widget oder das Dashboard als Datenquelle für Hosts auswählen. Dieses Feld unterstützt die automatische Vervollständigung. Wenn Sie also beginnen, den Namen eines Hosts einzugeben, wird eine Dropdown-Liste mit passenden Hosts angeboten. Wenn in den Feldern *Host-Gruppen* und *Hosts* nichts ausgewählt ist, werden alle Hosts mit gültigen Koordinaten angezeigt. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Tags	Geben Sie Tags an, um die Anzahl der im Widget angezeigten Hosts zu begrenzen. Es ist möglich, bestimmte Tags und Tag-Werte sowohl einzuschließen als auch auszuschließen. Es können mehrere Bedingungen festgelegt werden. Der Abgleich von Tag-Namen ist immer groß-/kleinschreibungssensitiv. Für jede Bedingung stehen mehrere Operatoren zur Verfügung: **Existiert** - die angegebenen Tag-Namen einschließen; **Gleich** - die angegebenen Tag-Namen und -Werte einschließen (groß-/kleinschreibungssensitiv); **Enthält** - die angegebenen Tag-Namen einschließen, bei denen die Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv); **Existiert nicht** - die angegebenen Tag-Namen ausschließen; **Ungleich** - die angegebenen Tag-Namen und -Werte ausschließen (groß-/kleinschreibungssensitiv); **Enthält nicht** - die angegebenen Tag-Namen ausschließen, bei denen die Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv). Für Bedingungen gibt es zwei Berechnungstypen: **Und/Oder** - alle Bedingungen müssen erfüllt sein; Bedingungen mit demselben Tag-Namen werden durch die Bedingung *Oder* gruppiert; **Oder** - es genügt, wenn eine Bedingung erfüllt ist. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Anfangsansicht	Durch Kommas getrennte Mittelpunktkoordinaten und eine optionale Zoomstufe, die beim ersten Laden des Widgets im Format `<latitude>,<longitude>,<zoom>` angezeigt werden, z. B. Wenn ein anfänglicher Zoom angegeben ist, wird das Geomap-Widget mit der angegebenen Zoomstufe geladen. Andernfalls wird der anfängliche Zoom als die Hälfte des maximalen Zooms für den jeweiligen Kachelanbieter berechnet. Die Anfangsansicht wird ignoriert, wenn die Standardansicht festgelegt ist (siehe unten). Beispiele: 40.6892494,-74.0466891 40.6892494,-122.0466891
Clustering	Geben Sie an, wie nahe beieinander liegende Host-Markierungen zu einer einzelnen Markierung mit Anzahl zusammengefasst werden: **Auto** - Markierungen werden automatisch zusammengefasst; **Zoomstufe** - Markierungen werden basierend auf der angegebenen Zoomstufe zusammengefasst (0 - maximal herausgezoomt, 30 - maximal hineingezoomt). Wenn der Parameter beispielsweise auf 0 gesetzt ist, werden Markierungen bei keiner Zoomstufe zusammengefasst; wenn er auf 15 gesetzt ist, werden Markierungen von der maximal herausgezoomten Ansicht bis zur Zoomstufe 15 zusammengefasst; wenn er auf 30 gesetzt ist, werden Markierungen bei allen Zoomstufen zusammengefasst. Beachten Sie, dass die maximale Hineinzoomstufe vom Kartenkachelanbieter bestimmt wird.

Graph

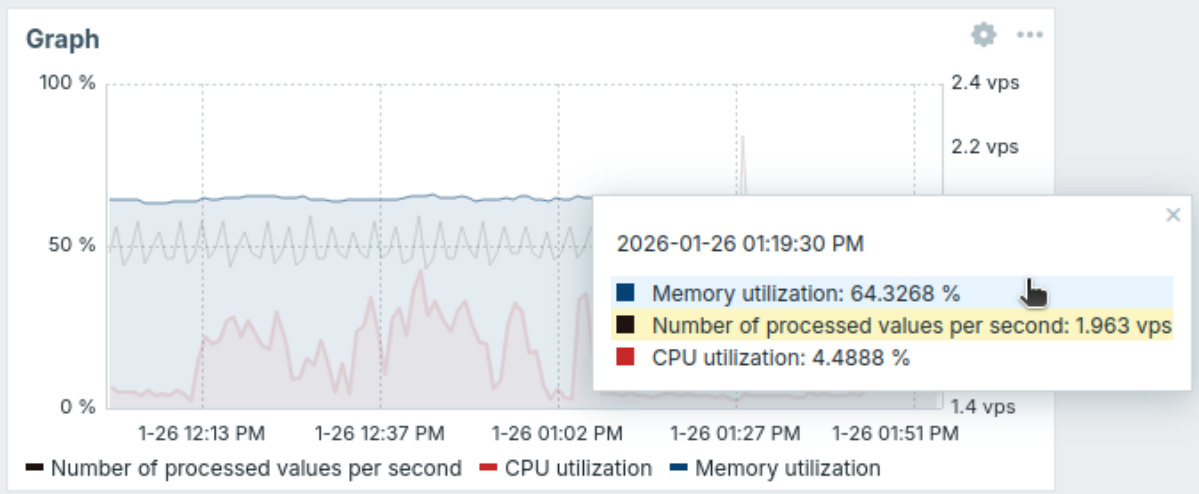
Übersicht

Das Widget *Graph* zeigt numerische Daten von Datenpunkten als vektorbasiertes Diagramm an. Es kann Ihnen dabei helfen, Trends zu verfolgen, Probleme zu erkennen und Werte im Zeitverlauf sowie hostübergreifend zu vergleichen.

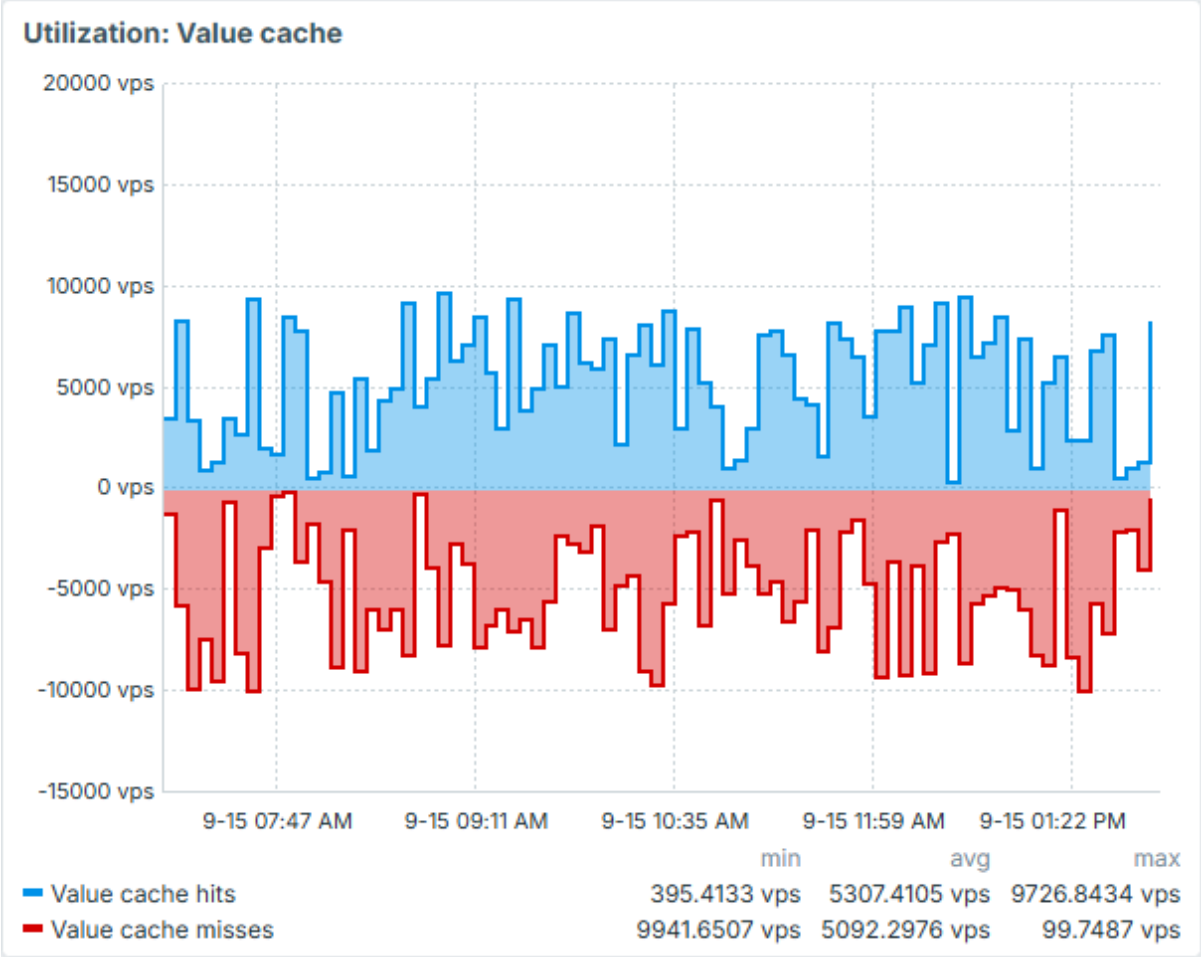


Wenn Sie den Mauszeiger über das Diagramm bewegen, wird eine QuickInfo mit einer nach Wert absteigend sortierten Liste von Datenpunkten angezeigt.

Wenn Sie den Mauszeiger über einen Datenpunkt in der QuickInfo bewegen, wird dessen Graph hervorgehoben (die anderen werden abgedunkelt). Durch die Auswahl des Datenpunkts werden dessen Daten an andere Widgets übertragen.



Das Diagramm kann mit invertierten Werten der Y-Achse angezeigt werden. Dies wirkt sich nur auf die visuelle Darstellung aus und ändert die ursprünglichen Werte nicht.

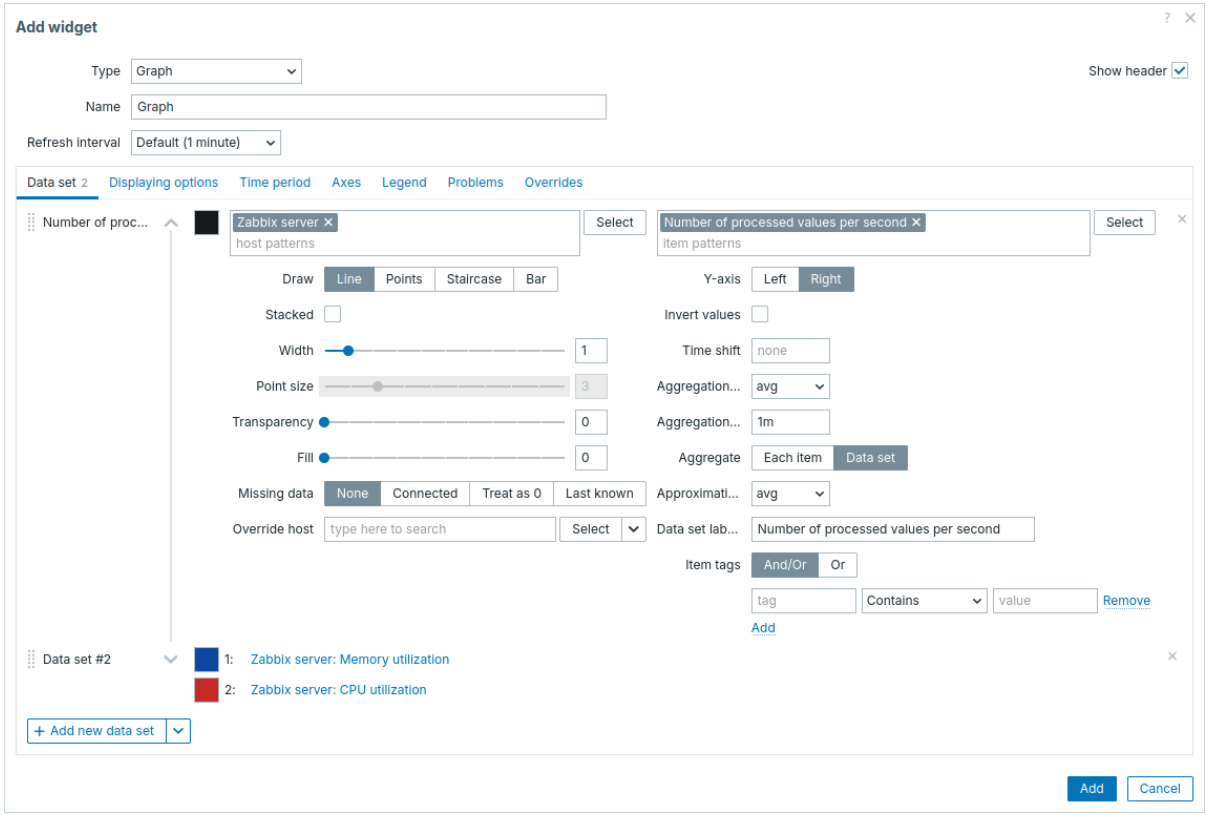


Das ältere Widget Graph (classic) ist ebenfalls verfügbar.

Die im Widget *Graph* angezeigten Informationen können als PNG-Bild heruntergeladen werden. Wählen Sie dazu die Option *Download image* im Menü *Aktionen* des Widgets ⋮, das angezeigt wird, wenn Sie den Mauszeiger über das Widget bewegen. Abhängig von Ihren Browsereinstellungen wird das Bild entweder automatisch im Ordner `Downloads` gespeichert oder ein Dialogfeld *Speichern* geöffnet, in dem Sie den Speicherort auswählen können.

Konfiguration

Wählen Sie zur Konfiguration *Graph* als Typ aus:



Datensatz

Auf der Registerkarte **Datensatz** können Sie Daten für das Diagramm auswählen, indem Sie Datensätze hinzufügen. Es können zwei Arten von Datensätzen hinzugefügt werden:

- *Datenpunktmuster* – Daten von übereinstimmenden Datenpunkten werden angezeigt. Sie können eine einzelne Grundfarbe auswählen oder eine Palettenzeile auswählen, um jedem übereinstimmenden Datenpunkt eine eigene Farbe zuzuweisen.
- *Datenpunktliste* – Daten der ausgewählten Datenpunkte werden angezeigt. Sie können die Farbe jedes Datenpunkts einzeln über die Farbauswahl festlegen.

Standardmäßig wird ein Datensatz vom Typ *Datenpunktmuster* hinzugefügt.

Datensatz	Für den Datensatz **Datenpunktmuster**: Wählen Sie Host- und Datenpunktmuster aus oder geben Sie diese ein. Die Daten der Datenpunkte, die diesen Mustern entsprechen, werden im Diagramm angezeigt. Es können bis zu 50 Datenpunkte angezeigt werden. Für die Auswahl können Platzhaltermuster verwendet werden (z. B. liefert ∗ Ergebnisse, die null oder mehr Zeichen enthalten). Um ein Platzhaltermuster anzugeben, geben Sie die Zeichenfolge manuell ein und drücken Sie die Eingabetaste. Das Platzhaltersymbol wird immer interpretiert. Daher ist es beispielsweise nicht möglich, einen Datenpunkt namens *item** einzeln hinzuzufügen, wenn andere übereinstimmende Datenpunkte vorhanden sind (z. B. *item2*, *item3*). Die Angabe von Host- und Datenpunktmustern ist für Datensätze vom Typ „Datenpunktmuster“ obligatorisch. Siehe auch: Details zur Datensatzkonfiguration. Für den Datensatz **Datenpunktliste**: Wählen Sie Datenpunkte für das Diagramm aus, indem Sie auf die Schaltfläche *Datenpunkt hinzufügen* klicken. Sie können auch kompatible Widgets als Datenquelle für Datenpunkte auswählen, indem Sie auf die Schaltfläche *Widget hinzufügen* klicken. Die Angabe von Datenpunkten oder Widgets ist für Datensätze vom Typ „Datenpunktliste“ obligatorisch. Siehe auch: Details zur Datensatzkonfiguration. Beachten Sie, dass nur numerische Datenpunkttypen zulässig sind. Bei der Konfiguration des Widgets auf einem Vorlagen-Dashboard ist der Parameter zur Angabe von Hostmustern nicht verfügbar. Der Parameter zur Angabe einer Datenpunktliste erlaubt nur die Auswahl von auf der Vorlage konfigurierten Datenpunkten.
Darstellung	Wählen Sie den Darstellungstyp der Metrik aus. Mögliche Darstellungstypen: *Linie* (standardmäßig festgelegt), *Punkte*, *Treppen* und *Balken*. Beachten Sie, dass bei nur einem Datenpunkt im Linien- oder Treppendiagramm dieser unabhängig vom Darstellungstyp als Punkt gezeichnet wird. Die Punktgröße wird anhand der Linienbreite berechnet, kann jedoch nicht kleiner als 3 Pixel sein, selbst wenn die Linienbreite geringer ist.
Gestapelt	Aktivieren Sie das Kontrollkästchen, um Daten gestapelt anzuzeigen (gefüllte Bereiche werden angezeigt). Diese Option ist deaktiviert, wenn der Darstellungstyp *Punkte* ausgewählt ist.
Breite	Legen Sie die Linienbreite fest. Diese Option ist verfügbar, wenn der Darstellungstyp *Linie* oder *Treppen* ausgewählt ist.
Punktgröße	Legen Sie die Punktgröße fest. Diese Option ist verfügbar, wenn der Darstellungstyp *Punkte* ausgewählt ist.
Transparenz	Legen Sie den Transparenzgrad fest.
Füllung	Legen Sie den Füllgrad fest. Diese Option ist verfügbar, wenn der Darstellungstyp *Linie* oder *Treppen* ausgewählt ist.
Fehlende Daten	Wählen Sie die Option für die Anzeige fehlender Daten aus: **Keine** – die Lücke bleibt leer; **Verbunden** – zwei Randwerte werden verbunden; **Als 0 behandeln** – fehlende Daten werden als Werte 0 angezeigt; **Letzter bekannter Wert** – fehlende Daten werden mit demselben Wert wie der letzte bekannte Wert angezeigt; gilt nicht für die Darstellungstypen *Punkte* und *Balken*.
Host überschreiben	Wählen Sie ein kompatibles Widget oder die Hostauswahl des Dashboards als Datenquelle für Hosts aus. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Y-Achse	Wählen Sie die Seite des Diagramms aus, auf der die Y-Achse angezeigt wird.
Werte invertieren	Aktivieren Sie das Kontrollkästchen, um alle Werte der Y-Achse vor der Anzeige im Diagramm mit -1 zu multiplizieren. Dies betrifft nur die visuelle Darstellung. Die ursprünglichen Daten bleiben unverändert. Diese Option wirkt sich nicht auf den Tooltip, die Legende oder die Perzentillinie aus. Wenn für die Achse die Option *Einfache Auslöser* ausgewählt ist, werden die Schwellenwerte der Auslöser invertiert angezeigt.

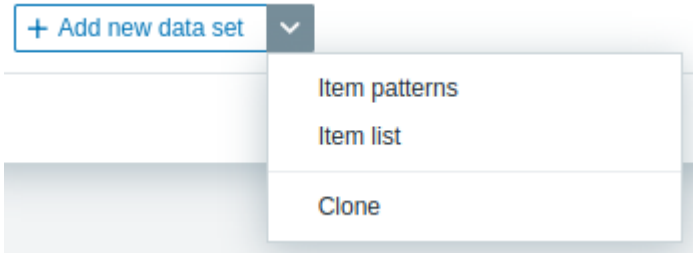
Zeitverschiebung	Geben Sie bei Bedarf eine Zeitverschiebung an. In diesem Feld können Sie Zeitsuffixe verwenden. Negative Werte sind zulässig.
Aggregationsfunktion	Geben Sie an, welche Aggregationsfunktion für jeden Datenpunkt oder für einen gesamten Datensatz innerhalb des ausgewählten *Aggregationsintervalls* verwendet werden soll: **min** – den kleinsten Wert anzeigen; **max** – den größten Wert anzeigen; **avg** – den Durchschnittswert anzeigen; **sum** – die Summe der Werte anzeigen; **count** – die Anzahl der Werte anzeigen; **first** – den ersten Wert anzeigen; **last** – den letzten Wert anzeigen. Wenn **nicht verwendet** ausgewählt ist, werden alle Werte angezeigt (keine Aggregation). Für *min*, *max*, *avg* und *sum* können nur numerische Daten angezeigt werden. Für *count* werden nicht numerische Daten in numerische Daten umgewandelt. Siehe auch: Aggregation in Diagrammen.
Aggregationsintervall	Geben Sie das Intervall für die Aggregation von Werten an. In diesem Feld können Sie Zeitsuffixe verwenden. Ein numerischer Wert ohne Suffix wird als Sekunden interpretiert. Beachten Sie, dass bei der Konfiguration des Widgets zur Anzeige historischer Daten auf Grundlage von Trends (*Auswahl der Verlaufsdaten* ist auf *Trends* oder *Automatisch* gesetzt) empfohlen wird, ein Aggregationsintervall zu verwenden, das ein Vielfaches von 1 Stunde ist (z. B. 3600, 60m, 1h, 3h usw.). Trends speichern stündlich aggregierte Werte. Daher können Aggregationsintervalle, die kein Vielfaches von 1 Stunde sind (z. B. 100s, 7min, 15min, 90min usw.), zu schwer interpretierbaren Ergebnissen führen.
Aggregieren	Geben Sie an, ob aggregiert werden soll: **Jeder Datenpunkt** – jeder Datenpunkt im Datensatz wird separat aggregiert und angezeigt; **Datensatz** – alle Datenpunkte des Datensatzes werden aggregiert und als ein Wert angezeigt.
Näherung	Geben Sie an, welcher Wert angezeigt werden soll, wenn mehr als ein Wert pro vertikalem Diagrammpixel vorhanden ist: **alle** – den kleinsten, den größten und den Durchschnittswert anzeigen; **min** – den kleinsten Wert anzeigen; **max** – den größten Wert anzeigen; **avg** – den Durchschnittswert anzeigen. Diese Einstellung ist nützlich, wenn ein Diagramm für einen großen Zeitraum mit einem kurzen Aktualisierungsintervall angezeigt wird (z. B. ein Jahr mit Werten, die alle 10 Minuten erfasst wurden).
Datensatzbezeichnung	Geben Sie die Datensatzbezeichnung an, die in der Diagrammkonfiguration *Datensatz* und in der Diagramm-*Legende* (für aggregierte Datensätze) angezeigt wird. Alle Datensätze werden nummeriert, einschließlich der Datensätze mit einer angegebenen *Datensatzbezeichnung*. Wenn keine Bezeichnung angegeben ist, wird der Datensatz automatisch entsprechend seiner Nummer bezeichnet (z. B. „Datensatz Nr. 2“, „Datensatz Nr. 3“ usw.). Die Nummerierung der Datensätze wird nach dem Neuanordnen oder Ziehen von Datensätzen neu berechnet. Zu lange Datensatzbezeichnungen werden für die Anzeige gekürzt (z. B. „Anzahl der Proz...“).

Datenpunkt-Tags	Geben Sie Tags an, um die im Widget angezeigten Datenpunkte zu filtern. Es ist möglich, bestimmte Tags und Tagwerte einzuschließen oder auszuschließen. Es können mehrere Bedingungen festgelegt werden. Beim Abgleich von Tagnamen wird immer zwischen Groß- und Kleinschreibung unterschieden. Für jede Bedingung stehen mehrere Operatoren zur Verfügung: **Existiert** – die angegebenen Tagnamen einschließen; **Ist gleich** – die angegebenen Tagnamen und Tagwerte einschließen (Groß-/Kleinschreibung wird berücksichtigt); **Enthält** – die angegebenen Tagnamen einschließen, deren Tagwerte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich ohne Berücksichtigung der Groß-/Kleinschreibung); **Existiert nicht** – die angegebenen Tagnamen ausschließen; **Ist nicht gleich** – die angegebenen Tagnamen und Tagwerte ausschließen (Groß-/Kleinschreibung wird berücksichtigt); **Enthält nicht** – die angegebenen Tagnamen ausschließen, deren Tagwerte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich ohne Berücksichtigung der Groß-/Kleinschreibung). Für Bedingungen gibt es zwei Berechnungstypen: **Und/Oder** – alle Bedingungen müssen erfüllt sein; Bedingungen mit demselben Tagnamen werden durch die Oder-Bedingung gruppiert; **Oder** – es reicht aus, wenn eine Bedingung erfüllt ist.

Details zur Datenkonfiguration

Vorhandene Datensätze werden in einer Liste angezeigt. Sie können:

- Auf das Verschiebesymbol klicken und einen Datensatz an eine neue Position in der Liste ziehen.
- Auf das Erweiterungssymbol klicken, um die Details des Datensatzes einzublenden. Wenn der Datensatz erweitert ist, wird dieses Symbol zu einem Reduzierungssymbol.
- Auf das Farbsymbol klicken, um die Farbauswahl zu öffnen. Sie können einen Hexadezimalcode eingeben, einfarbiges Farbfeld auswählen oder zur Registerkarte *Palette* wechseln und eine Zeile vordefinierter Farben auswählen. Die ausgewählte Farbe wird direkt auf Datensätze vom Typ *Item list* angewendet oder dient als Grundlage für generierte Farbabstufungen in *Item patterns*. Verwenden Sie die Tabulatortaste, um zwischen den Steuerelementen des Dialogfelds zu wechseln, die Pfeiltasten, um durch Farbfelder oder Palettenzeilen zu navigieren, die Eingabetaste zum Auswählen und Esc zum Abbrechen.
- Auf die Schaltfläche *Add new data set* klicken, um einen leeren Datensatz hinzuzufügen, in dem Sie Host- und Datenpunktmuster auswählen können. Wenn Sie auf das nach unten zeigende Symbol neben der Schaltfläche *Add new data set* klicken, wird ein Dropdown-Menü angezeigt. Dort können Sie einen neuen Datensatz vom Typ *Item patterns* oder *Item list* hinzufügen oder den aktuell geöffneten Datensatz *Clone*. Wenn alle Datensätze reduziert sind, ist die Option *Clone* nicht verfügbar.

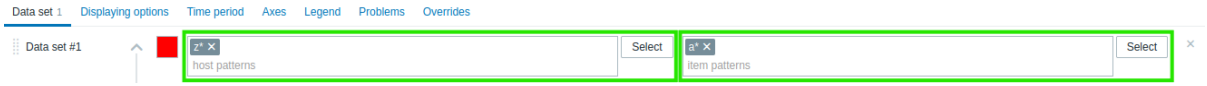


Der Datensatz **Datenpunkt-Muster** enthält die Felder *Host-Muster* und *Datenpunkt-Muster*, die beide vollständige Namen oder Muster mit einem Platzhaltersymbol (*) erkennen. Diese Funktion ermöglicht es Ihnen, alle Host-Namen und Datenpunkt-Namen auszuwählen, die das gewählte Muster enthalten. Während Sie den Datenpunkt-Namen oder das Datenpunkt-Muster im Feld *Datenpunkt-Muster* eingeben, werden in der Dropdown-Liste nur Datenpunkte angezeigt, die zu den ausgewählten Host-Namen gehören.

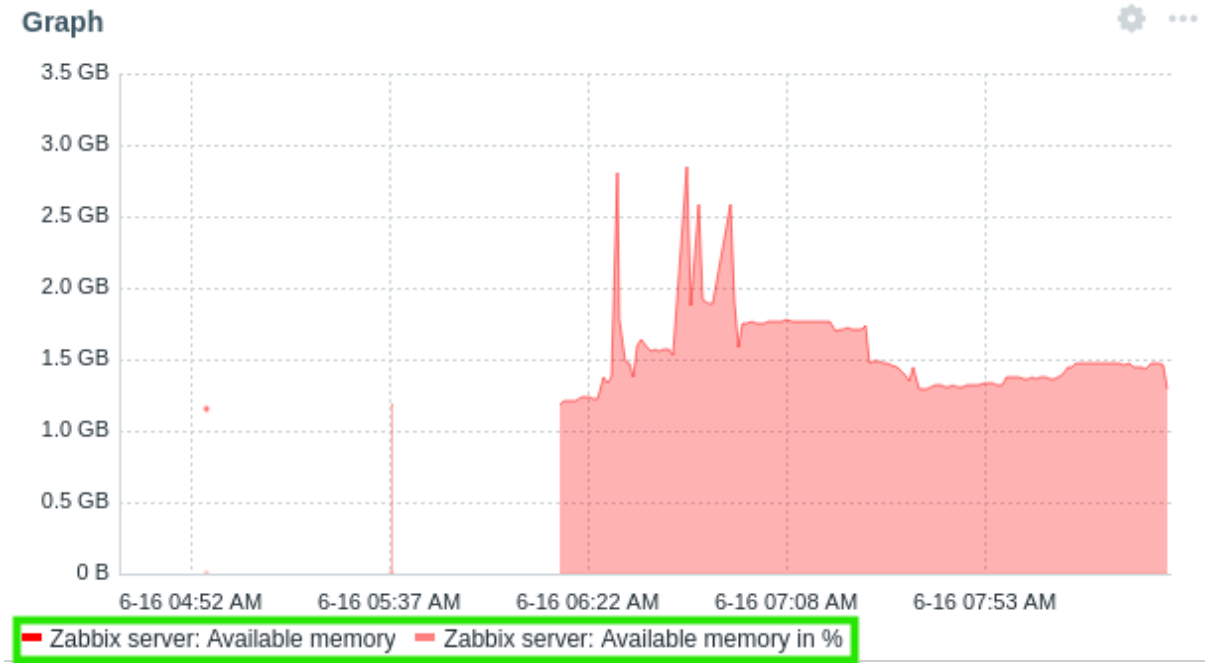
Wenn Sie zum Beispiel das Muster **z*** in das Feld *Host-Muster* eingegeben haben, zeigt die Dropdown-Liste alle Host-Namen an, die dieses Muster enthalten: **z*, Zabbix server, Zabbix proxy**. Nach dem Drücken von *Enter* wird dieses Muster übernommen und als **z*** angezeigt. Wenn Sie entsprechend das Muster **a*** in das Feld *Datenpunkt-Muster* eingegeben haben, zeigt die Dropdown-Liste alle Datenpunkt-Namen an, die dieses Muster enthalten: **a*, Available memory, Available memory in %**.



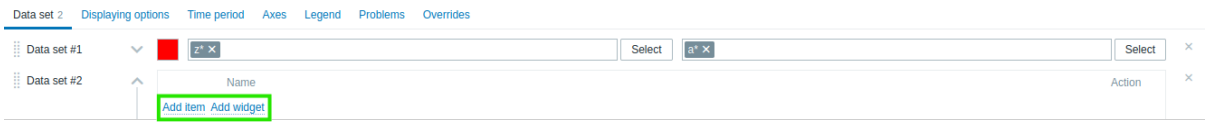
Nach dem Drücken von *Enter* wird das Muster übernommen und als **a*** angezeigt.



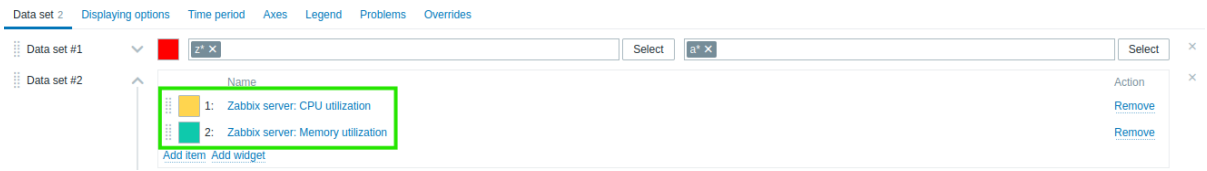
Das Diagramm zeigt dann alle Datenpunkte an, die zu den ausgewählten Host-Namen gehören.



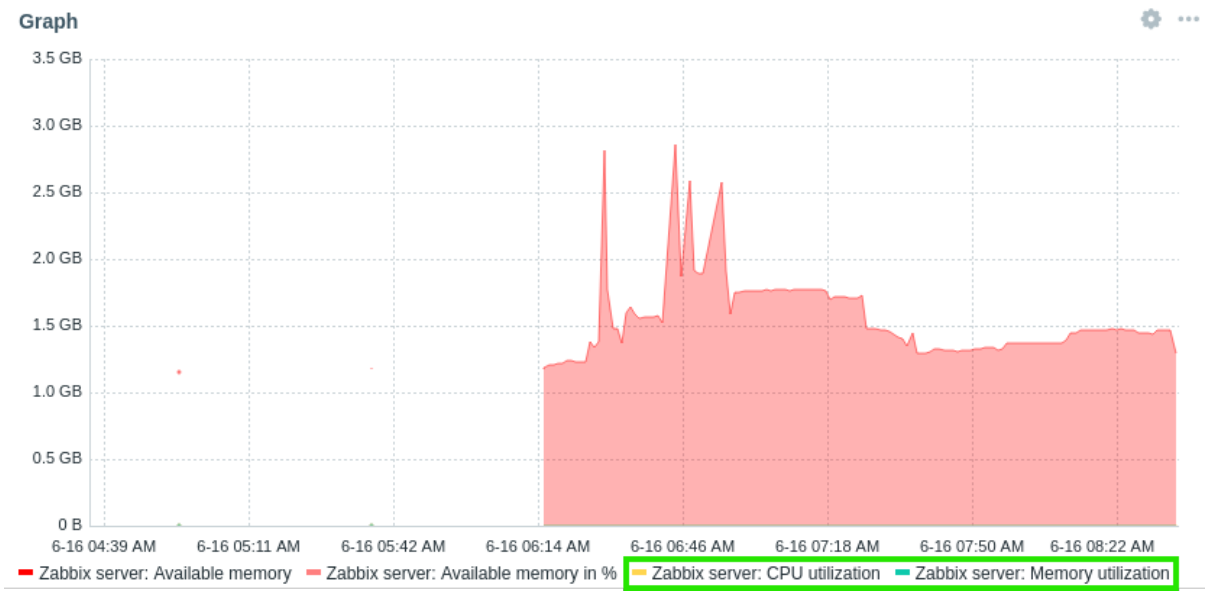
Der Datensatz **Datenpunktliste** enthält die Schaltfläche *Datenpunkt hinzufügen*, mit der Sie Datenpunkte hinzufügen können, die im Diagramm angezeigt werden sollen. Sie können auch kompatible Widgets als Datenquelle für Datenpunkte hinzufügen, indem Sie auf die Schaltfläche *Widget hinzufügen* klicken.



Wenn Sie beispielsweise auf die Schaltfläche *Datenpunkt hinzufügen* klicken, wird ein Pop-up-Fenster mit dem Parameter *Host* geöffnet. Nachdem Sie einen Host ausgewählt haben, werden alle seine zur Auswahl verfügbaren Datenpunkte in einer Liste angezeigt.



Nachdem Sie einen oder mehrere Datenpunkte ausgewählt haben, werden diese in der Datenpunktliste des Datensatzes und im Diagramm angezeigt.

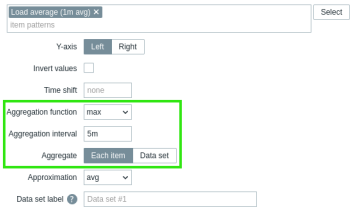


Aggregation in Diagrammen

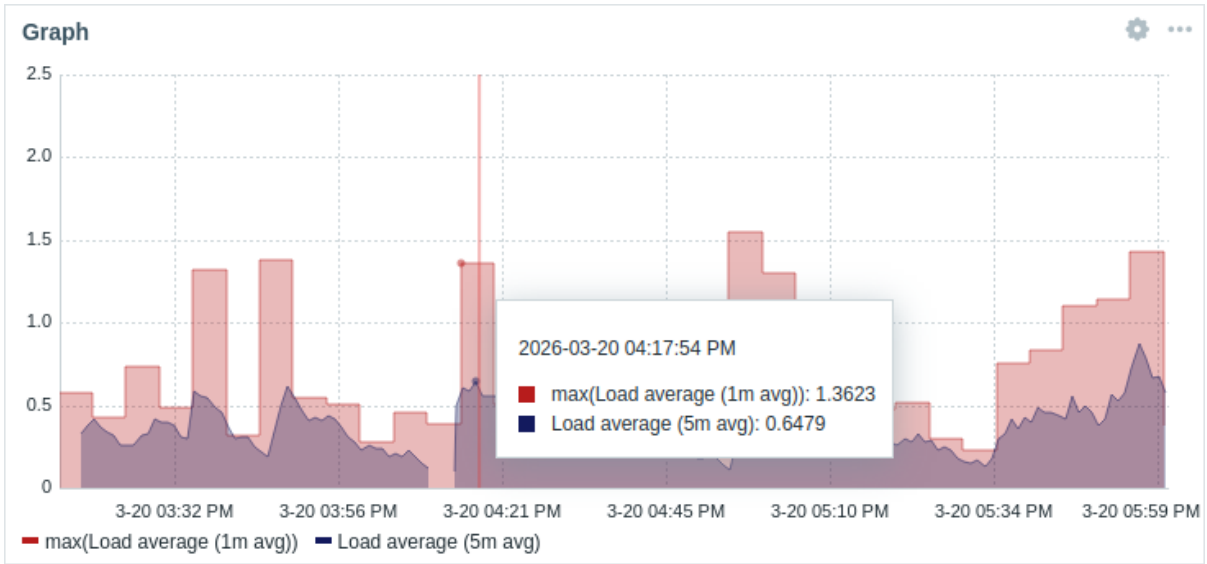
Durch die Aggregation kann für das angegebene Intervall (5 Minuten, eine Stunde, ein Tag usw.) anstelle aller einzelnen Werte ein aggregierter Wert (Minimum, Maximum, Durchschnitt usw.) angezeigt werden. Innerhalb eines Datensatzes kann die Aggregation entweder auf jeden Datenpunkt einzeln oder auf den gesamten Datensatz angewendet werden.

Einige Anwendungsfälle für die Aggregationsfunktion sind: - Die durchschnittliche Anzahl der Anfragen an einen NGINX-Server pro Tag. - Der geringste verfügbare Speicherplatz unter den Clustern pro Woche.

Im folgenden Beispiel wird die Aggregationsfunktion „max“ auf den Datenpunkt mit einem Aggregationsintervall von fünf Minuten angewendet:



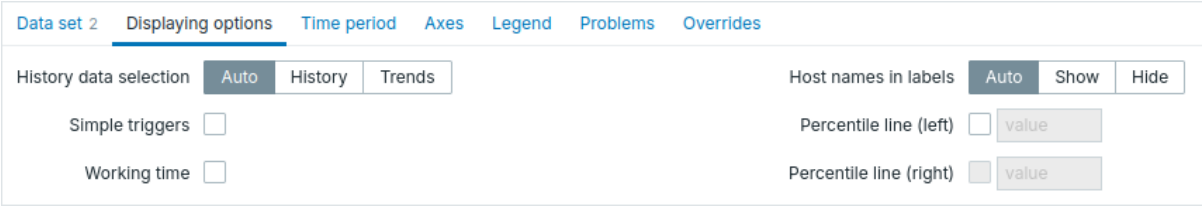
Das Beispiel zeigt die maximale CPU-Auslastung über fünf Minuten (aggregiert aus durchschnittlichen Werten im Abstand von einer Minute) im Vergleich zur durchschnittlichen CPU-Auslastung über fünf Minuten.



Wenn *Aggregationsfunktion anzeigen* für die Legende ausgewählt ist, werden Datenpunkte in Klammern angezeigt, wobei ihnen in der Legende und im Tooltip die Aggregationsfunktion vorangestellt wird. Wenn für ein Diagramm eine *Bezeichnung des Datensatzes* angegeben ist, wird die Bezeichnung in Klammern angezeigt, wobei ihr die Aggregationsfunktion vorangestellt wird.

Anzeigeoptionen

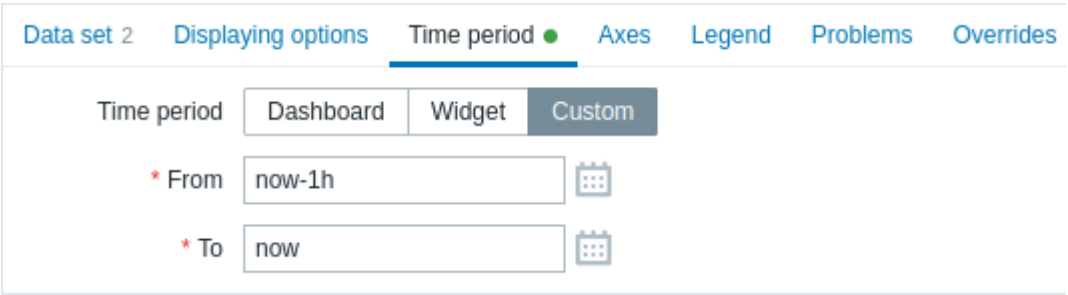
Auf der Registerkarte **Anzeigeoptionen** können Sie die Auswahl der Verlaufsdaten festlegen:



Auswahl der Verlaufsdaten	Legen Sie die Quelle der Diagrammdaten fest: **Auto** – Daten werden gemäß dem klassischen Diagramm-Algorithmus bezogen (Standard); **History** – Daten aus dem Verlauf; **Trends** – Daten aus Trends.
Einfache Auslöser	Aktivieren Sie das Kontrollkästchen, um die Schwellenwerte der Auslöser für einfache Auslöser anzuzeigen. Die Schwellenwerte werden als gestrichelte Linien in der Farbe des Auslöser-Schweregrads dargestellt. Ein einfacher Auslöser ist ein Auslöser mit einer Funktion (nur `last`, `max`, `min`, `avg`) für einen Datenpunkt im Ausdruck. Es können maximal drei Auslöser dargestellt werden. Beachten Sie, dass sich der Auslöser innerhalb des dargestellten Bereichs befinden muss, damit er sichtbar ist.
Arbeitszeit	Aktivieren Sie das Kontrollkästchen, um die Arbeitszeit im Diagramm anzuzeigen. Die Arbeitszeit (Arbeitstage) wird in Diagrammen als weißer Hintergrund dargestellt, während die arbeitsfreie Zeit grau angezeigt wird (mit dem standardmäßigen Frontend-Theme *Original blue*).
Host-Namen in Beschriftungen	Wählen Sie aus, ob Host-Namen in der Diagrammlegende und im Tooltip angezeigt werden sollen, der beim Bewegen des Mauszeigers über die Diagrammdaten erscheint: **Auto** – Host-Namen werden nur angezeigt, wenn in den Datensätzen mehr als ein Host vorhanden ist (Standard). **Show** – Host-Namen werden angezeigt. **Hide** – Host-Namen werden ausgeblendet. Dieser Parameter ist beim Konfigurieren des Widgets auf einem Vorlagen-Dashboard nicht verfügbar.
Perzentillinie (links)	Aktivieren Sie das Kontrollkästchen und geben Sie den Perzentilwert ein, um das angegebene Perzentil als Linie auf der linken Y-Achse des Diagramms anzuzeigen. Wenn beispielsweise das 95-%-Perzentil festgelegt ist, befindet sich die Perzentillinie auf der Höhe, unter der 95 Prozent der Werte liegen.
Perzentillinie (rechts)	Aktivieren Sie das Kontrollkästchen und geben Sie den Perzentilwert ein, um das angegebene Perzentil als Linie auf der rechten Y-Achse des Diagramms anzuzeigen. Wenn beispielsweise das 95-%-Perzentil festgelegt ist, befindet sich die Perzentillinie auf der Höhe, unter der 95 Prozent der Werte liegen.

Zeitraum

Auf der Registerkarte **Zeitraum** können Sie einen Zeitraum festlegen, für den Daten im Diagramm angezeigt werden:

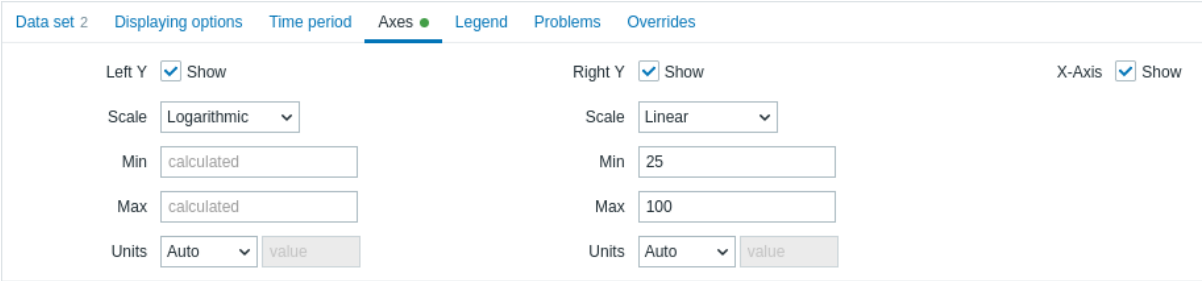


Zeitraum	Wählen Sie die Datenquelle für den Zeitraum aus: **Dashboard** – verwendet den Zeitraumauswahlschalter des Dashboards; **Widget** – verwendet ein kompatibles Widget (im Parameter *Widget* festgelegt); **Benutzerdefiniert** – verwendet einen benutzerdefinierten Zeitraum, der in den Parametern *Von* und *Bis* festgelegt wird; wenn dieser festgelegt ist, wird in der oberen rechten Ecke des Widgets ein Uhrsymbol angezeigt, das bei Mouseover die festgelegte Zeit angibt. Beachten Sie, dass kompatible Widgets unabhängig von der Konfiguration *Zeitraum* des Widgets weiterhin als Datenquelle für den Zeitraum verwendet werden können.
Widget	Geben Sie ein kompatibles Widget als Datenquelle für den Zeitraum ein oder wählen Sie es aus. Dieser Parameter ist verfügbar, wenn *Zeitraum* auf „Widget“ gesetzt ist.

Von	Geben Sie den Beginn des Zeitraums ein oder wählen Sie ihn aus. Die Syntax für relative Zeitangaben (`now`, `now/d`, `now/w-1w` usw.) wird unterstützt. Dieser Parameter ist verfügbar, wenn *Zeitraum* auf „Benutzerdefiniert“ gesetzt ist.
Bis	Geben Sie das Ende des Zeitraums ein oder wählen Sie es aus. Die Syntax für relative Zeitangaben (`now`, `now/d`, `now/w-1w` usw.) wird unterstützt. Dieser Parameter ist verfügbar, wenn *Zeitraum* auf „Benutzerdefiniert“ gesetzt ist.

Achsen

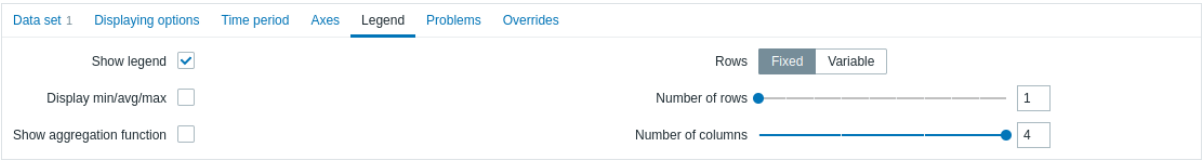
Auf der Registerkarte **Achsen** können Sie festlegen, wie Achsen angezeigt werden:



Linke Y-Achse	Aktivieren Sie dieses Kontrollkästchen, um die linke Y-Achse sichtbar zu machen. Das Kontrollkästchen kann deaktiviert sein, wenn es entweder auf der Registerkarte *Datensatz* oder *Überschreibungen* nicht ausgewählt wurde.
Rechte Y-Achse	Aktivieren Sie dieses Kontrollkästchen, um die rechte Y-Achse sichtbar zu machen. Das Kontrollkästchen kann deaktiviert sein, wenn es entweder auf der Registerkarte *Datensatz* oder *Überschreibungen* nicht ausgewählt wurde.
X-Achse	Deaktivieren Sie dieses Kontrollkästchen, um die X-Achse auszublenden (standardmäßig aktiviert).
Skalierung	Wählen Sie die Skalierung für die Werte der Diagrammachse aus der Dropdown-Liste aus: **Linear** – Die Achsenwerte erhöhen sich um einen festen Betrag (z. B. 10, 20, 30). Dies eignet sich für Daten, die sich gleichmäßig ändern oder einen kleinen bis mittleren Wertebereich abdecken. **Logarithmisch** – Die Achsenwerte erhöhen sich exponentiell (z. B. 10, 100, 1000). Dies eignet sich für Daten, die sich schnell ändern oder einen großen Wertebereich abdecken.
Min	Legen Sie den Mindestwert der entsprechenden Achse fest. Der Mindestwert des sichtbaren Bereichs der Y-Achse wird angegeben.
Max	Legen Sie den Höchstwert der entsprechenden Achse fest. Der Höchstwert des sichtbaren Bereichs der Y-Achse wird angegeben.
Einheiten	Wählen Sie die Einheit für die Werte der Diagrammachse aus der Dropdown-Liste aus: **Automatisch** – Die Achsenwerte werden mit der Einheit des ersten Datenpunkts im Datensatz angezeigt. **Statisch** – Die Achsenwerte werden mit der im Eingabefeld *Wert* angegebenen Einheit angezeigt. Wenn das Feld leer bleibt, werden nur numerische Werte angezeigt.

Legende

Auf der Registerkarte **Legende** können Sie die Legende des Diagramms anpassen:

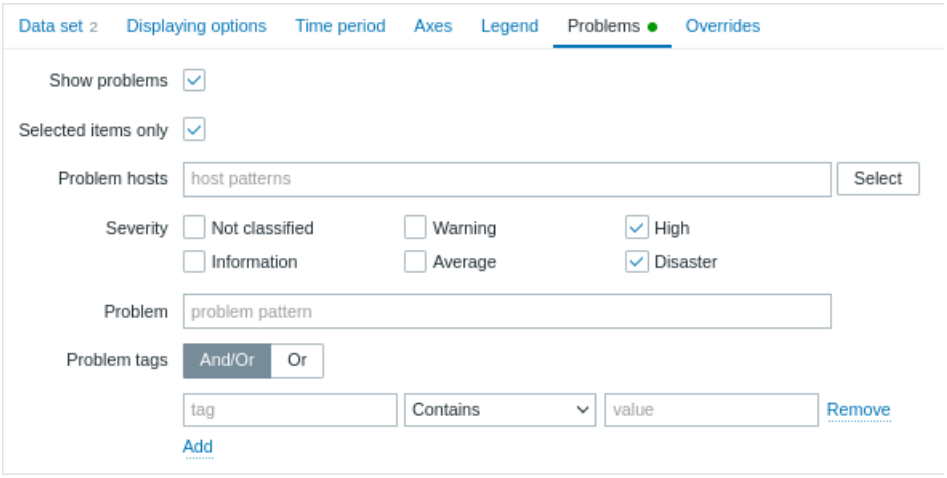


Legende anzeigen	Deaktivieren Sie dieses Kontrollkästchen, um die Legende im Diagramm auszublenden (standardmäßig aktiviert).
Min./Durchschn./Max. anzeigen	Aktivieren Sie dieses Kontrollkästchen, um die Minimal-, Durchschnitts- und Maximalwerte des Datenpunkts in der Legende anzuzeigen.
Aggregationsfunktion anzeigen	Aktivieren Sie dieses Kontrollkästchen, um die Aggregationsfunktion in der Legende anzuzeigen.

Zeilen	Wählen Sie den Anzeigemodus für die Zeilen der Legende aus: **Fest** – Die Anzahl der angezeigten Zeilen wird durch den Wert des Parameters *Anzahl der Zeilen* bestimmt; **Variabel** – Die Anzahl der angezeigten Zeilen wird durch die Anzahl der konfigurierten Datenpunkte bestimmt, darf jedoch den Wert des Parameters *Maximale Anzahl der Zeilen* nicht überschreiten.
Anzahl der Zeilen/ Maximale Anzahl der Zeilen	Wenn *Zeilen* auf „Fest“ gesetzt ist, legen Sie die Anzahl der anzuzeigenden Legendenzeilen fest (1–10). Wenn *Zeilen* auf „Variabel“ gesetzt ist, legen Sie die maximale Anzahl der anzuzeigenden Legendenzeilen fest (1–10).
Anzahl der Spalten	Legen Sie die Anzahl der anzuzeigenden Legendenspalten fest (1–4). Dieser Parameter ist verfügbar, wenn *Min./Durchschn./Max. anzeigen* deaktiviert ist.

Probleme

Auf der Registerkarte **Probleme** können Sie die Problemanzeige anpassen:

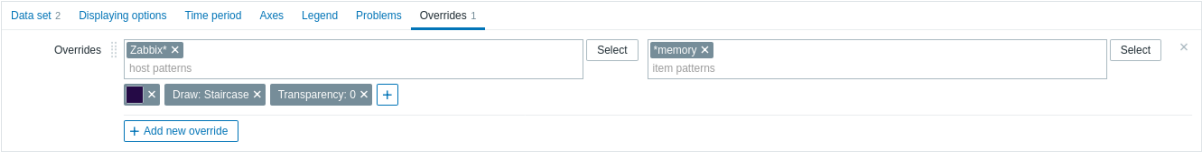


Probleme anzeigen	Aktivieren Sie dieses Kontrollkästchen, um die Anzeige von Problemen im Diagramm zu aktivieren (standardmäßig nicht aktiviert, d. h. deaktiviert).
Nur ausgewählte Datenpunkte	Aktivieren Sie dieses Kontrollkästchen, um nur Probleme für die ausgewählten Datenpunkte im Diagramm anzuzeigen.
Problem-Hosts	Wählen Sie die Problem-Hosts aus, die im Diagramm angezeigt werden sollen. Platzhaltermuster können verwendet werden (z. B. gibt ∗ Ergebnisse zurück, die auf null oder mehr Zeichen zutreffen). Um ein Platzhaltermuster anzugeben, geben Sie die Zeichenfolge manuell ein und drücken Sie die *Eingabetaste*. Beachten Sie während der Eingabe, dass alle übereinstimmenden Hosts in der Dropdown-Liste angezeigt werden. Dieser Parameter ist beim Konfigurieren des Widgets auf einem Vorlagen-Dashboard nicht verfügbar.
Schweregrad	Markieren Sie die Problem-Schweregrade, nach denen die im Diagramm anzuzeigenden Probleme gefiltert werden sollen. Wenn kein Schweregrad markiert ist, werden alle Probleme angezeigt.
Problem	Geben Sie den Namen des Problems an, das im Diagramm angezeigt werden soll.

Problem-Tags	Geben Sie Problem-Tags an, um die Anzahl der im Widget angezeigten Probleme zu begrenzen. Es ist möglich, bestimmte Tags und Tag-Werte einzuschließen oder auszuschließen. Es können mehrere Bedingungen festgelegt werden. Beim Abgleich von Tag-Namen wird immer zwischen Groß- und Kleinschreibung unterschieden. Für jede Bedingung stehen mehrere Operatoren zur Verfügung: **Existiert** – die angegebenen Tag-Namen einschließen; **Ist gleich** – die angegebenen Tag-Namen und -Werte einschließen (Groß-/Kleinschreibung wird berücksichtigt); **Enthält** – die angegebenen Tag-Namen einschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich ohne Berücksichtigung der Groß-/Kleinschreibung); **Existiert nicht** – die angegebenen Tag-Namen ausschließen; **Ist nicht gleich** – die angegebenen Tag-Namen und -Werte ausschließen (Groß-/Kleinschreibung wird berücksichtigt); **Enthält nicht** – die angegebenen Tag-Namen ausschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich ohne Berücksichtigung der Groß-/Kleinschreibung). Für Bedingungen gibt es zwei Berechnungstypen: **Und/Oder** – alle Bedingungen müssen erfüllt sein; Bedingungen mit demselben Tag-Namen werden durch die *Oder*-Bedingung gruppiert; **Oder** – es reicht aus, wenn eine Bedingung erfüllt ist.

Überschreibungen

Auf der Registerkarte **Überschreibungen** können Sie benutzerdefinierte Überschreibungen für Datensätze hinzufügen:



Überschreibungen sind nützlich, wenn mehrere Datenpunkte mithilfe des Platzhalters ∗ für einen Datensatz ausgewählt werden und Sie ändern möchten, wie die Datenpunkte standardmäßig angezeigt werden (z. B. die standardmäßige Grundfarbe oder eine andere Eigenschaft).

Vorhandene Überschreibungen (falls vorhanden) werden in einer Liste angezeigt. So fügen Sie eine neue Überschreibung hinzu:

- Klicken Sie auf die Schaltfläche 

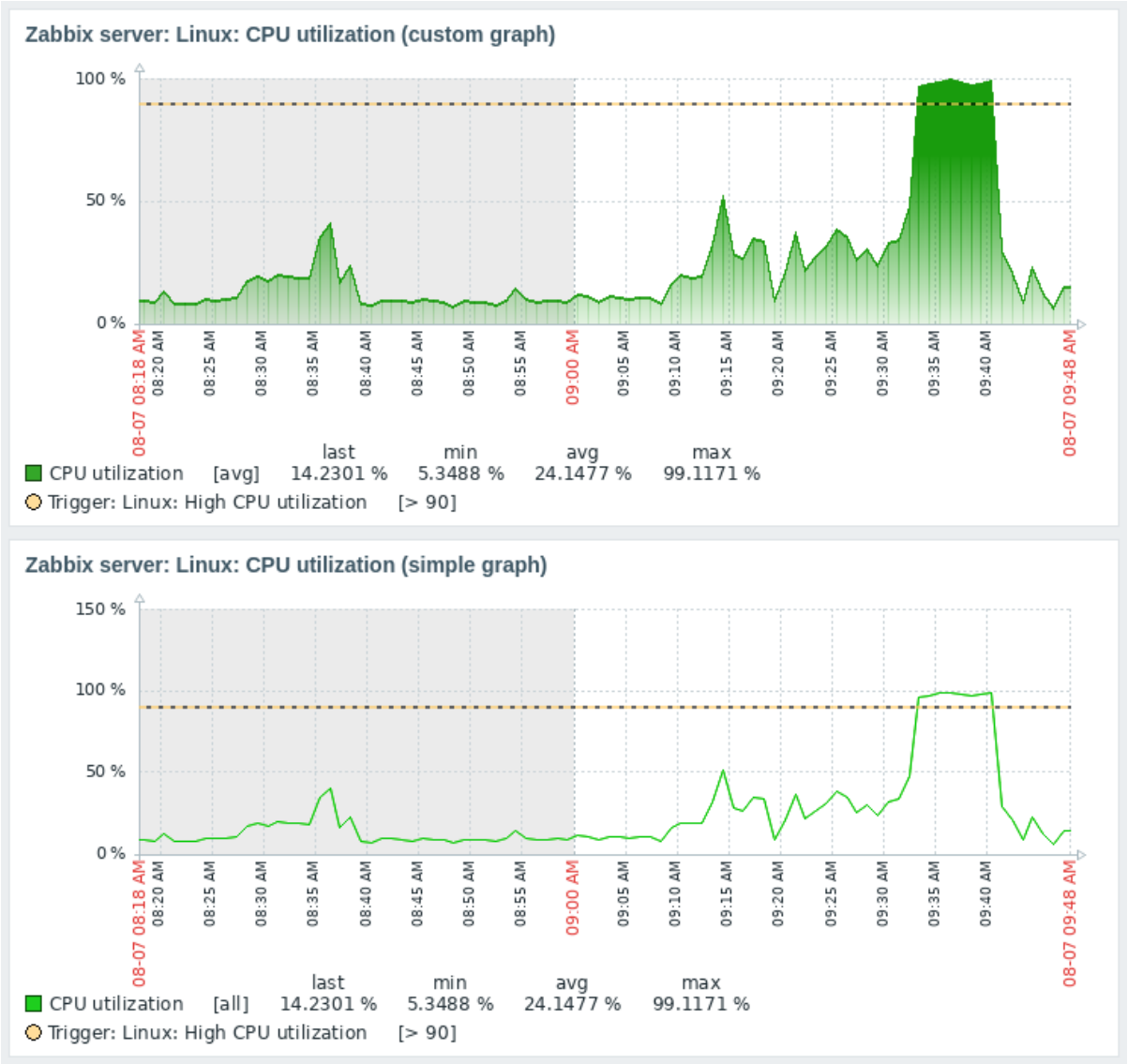
.
- Wählen Sie Hosts und Datenpunkte für die Überschreibung aus. Alternativ können Sie Host- und Datenpunktmuster eingeben. Platzhaltermuster können verwendet werden (beispielsweise liefert ∗ Ergebnisse, die null oder mehr Zeichen entsprechen). Um ein Platzhaltermuster anzugeben, geben Sie die Zeichenfolge manuell ein und drücken Sie die *Eingabetaste*. Beachten Sie während der Eingabe, dass alle übereinstimmenden Hosts in der Dropdown-Liste angezeigt werden. Das Platzhaltersymbol wird immer interpretiert. Daher ist es beispielsweise nicht möglich, einen Datenpunkt namens „item*“ einzeln hinzuzufügen, wenn andere übereinstimmende Datenpunkte vorhanden sind (z. B. item2, item3). Die Parameter für das Hostmuster und das Datenpunktmuster sind obligatorisch. Der Parameter zum Angeben von Hostmustern ist beim Konfigurieren des Widgets auf einem Vorlagen-Dashboard nicht verfügbar. Der Parameter zum Angeben einer Datenpunktliste ermöglicht beim Konfigurieren des Widgets auf einem Vorlagen-Dashboard nur die Auswahl von Datenpunkten, die auf der Vorlage konfiguriert sind.
- Klicken Sie auf 

, um Parameter für die Überschreibung auszuwählen. Mindestens ein Parameter für die Überschreibung muss ausgewählt werden. Beschreibungen der Parameter finden Sie oben auf der Registerkarte *Datensatz*.

Diagramm (klassisch)

Übersicht

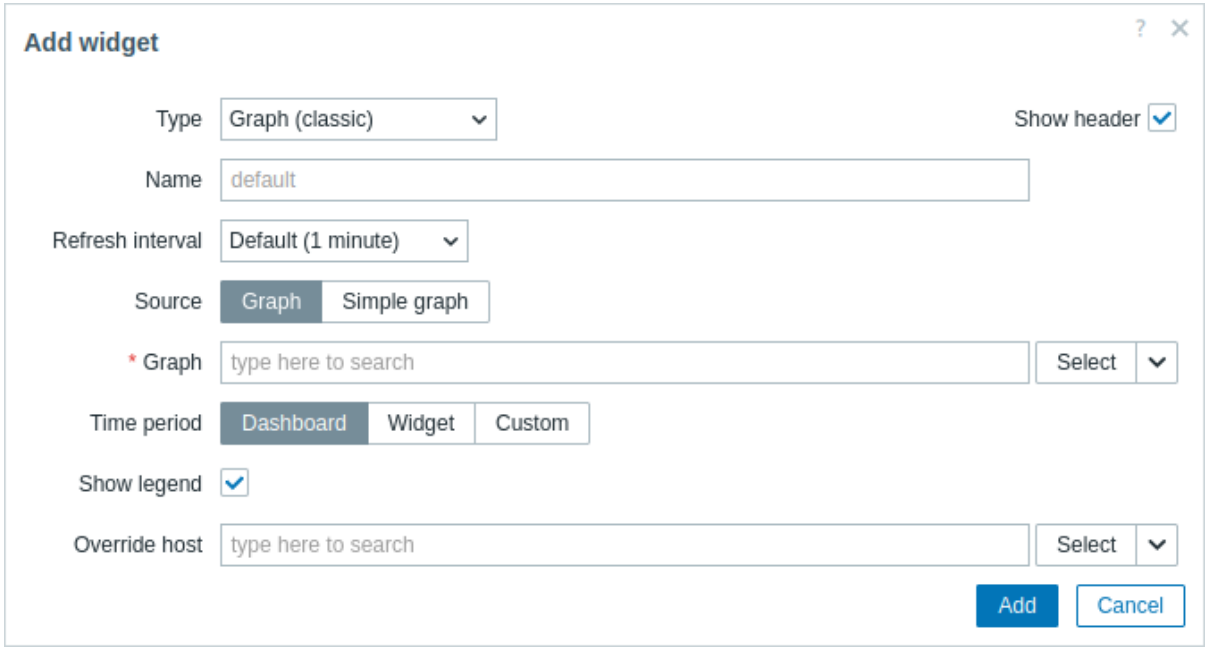
Das Widget *Graph (klassisch)* zeigt numerische Daten von Datenpunkten als bildbasierten benutzerdefinierten Graphen oder einfachen Graphen an.
Es unterstützt die Auswahl eines Zeitraums, Auslöserlinien und die optionale Anzeige einer Legende.



Die im Widget *Graph (klassisch)* angezeigten Informationen können als PNG-Bild heruntergeladen werden. Wählen Sie dazu die Option *Bild herunterladen* im Menü *Aktionen* des Widgets ⋮, das angezeigt wird, wenn Sie den Mauszeiger über das Widget bewegen.
Abhängig von Ihren Browsereinstellungen wird das Bild entweder automatisch im Ordner `Downloads` gespeichert oder ein Dialogfeld *Speichern* geöffnet, in dem Sie den Speicherort auswählen können.

Konfiguration

Um zu konfigurieren, wählen Sie *Graph (classic)* als Typ aus:



Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

Quelle	Wählen Sie den Graph-Typ aus: **Graph** - benutzerdefinierter Graph; **Simple graph** - einfacher Graph.
Graph	Wählen Sie den benutzerdefinierten Graph aus, der angezeigt werden soll. Alternativ können Sie ein kompatibles Widget als Datenquelle für Graphen auswählen. Dieser Parameter ist verfügbar, wenn *Quelle* auf ”Graph” gesetzt ist.
Datenpunkt	Wählen Sie den Datenpunkt aus, der in einem einfachen Graph angezeigt werden soll. Alternativ können Sie ein kompatibles Widget als Datenquelle für Datenpunkte auswählen. Dieser Parameter ist verfügbar, wenn *Quelle* auf ”Simple graph” gesetzt ist.
Zeitperiode	Legen Sie eine Zeitperiode fest, für die Daten im Graph angezeigt werden sollen. Wählen Sie die Datenquelle für die Zeitperiode aus: **Dashboard** - den Zeitperiodenwähler des Dashboards verwenden; **Widget** - ein kompatibles Widget verwenden (im Parameter *Widget* festgelegt); **Benutzerdefiniert** - eine benutzerdefinierte Zeitperiode verwenden, die in den Parametern *Von* und *Bis* festgelegt ist; wenn gesetzt, wird in der oberen rechten Ecke des Widgets ein Uhrsymbol angezeigt, das beim Überfahren mit der Maus die eingestellte Zeit anzeigt. Beachten Sie, dass kompatible Widgets unabhängig von der Konfiguration der *Zeitperiode* des Widgets diese weiterhin als Datenquelle für die Zeitperiode verwenden können.
Widget	Geben Sie ein kompatibles Widget als Datenquelle für die Zeitperiode ein oder wählen Sie es aus. Dieser Parameter ist verfügbar, wenn *Zeitperiode* auf ”Widget” gesetzt ist.
Von	Geben Sie den Beginn der Zeitperiode ein oder wählen Sie ihn aus. Die Syntax für relative Zeitangaben (`now`, `now/d`, `now/w-1w` usw.) wird unterstützt. Dieser Parameter ist verfügbar, wenn *Zeitperiode* auf ”Benutzerdefiniert” gesetzt ist.
Bis	Geben Sie das Ende der Zeitperiode ein oder wählen Sie es aus. Die Syntax für relative Zeitangaben (`now`, `now/d`, `now/w-1w` usw.) wird unterstützt. Dieser Parameter ist verfügbar, wenn *Zeitperiode* auf ”Benutzerdefiniert” gesetzt ist.
Legende anzeigen	Deaktivieren Sie dieses Kontrollkästchen, um die Legende im Graph auszublenden (standardmäßig aktiviert).
Host überschreiben	Wählen Sie ein kompatibles Widget oder den Host-Auswahlfilter des Dashboards als Datenquelle für Hosts aus. Dieser Parameter ist nicht verfügbar, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.

Die Legende des Graph besteht aus drei Abschnitten:

- Datenpunkte und ihre aggregierten Werte
- Perzentile (falls konfiguriert)
- Auslöser (falls welche mit den angezeigten Datenpunkten verknüpft sind)

Wenn die Höhe des Graph innerhalb des Widgets nicht ausreicht, wird die Legende möglicherweise nicht oder nur teilweise

angezeigt. Auslöser und Perzentile werden zuerst ausgeblendet, gefolgt von der Datenpunkt-Legende. Um die vollständige Legende anzuzeigen, erhöhen Sie die vertikale Größe des Widgets.

Note:
Es können nicht mehr als 3 Auslöser-Linien angezeigt werden. Wenn es mehr Auslöser gibt, werden die Auslöser mit niedrigerem Schweregrad bevorzugt angezeigt.

Diagramm-Prototyp

Übersicht

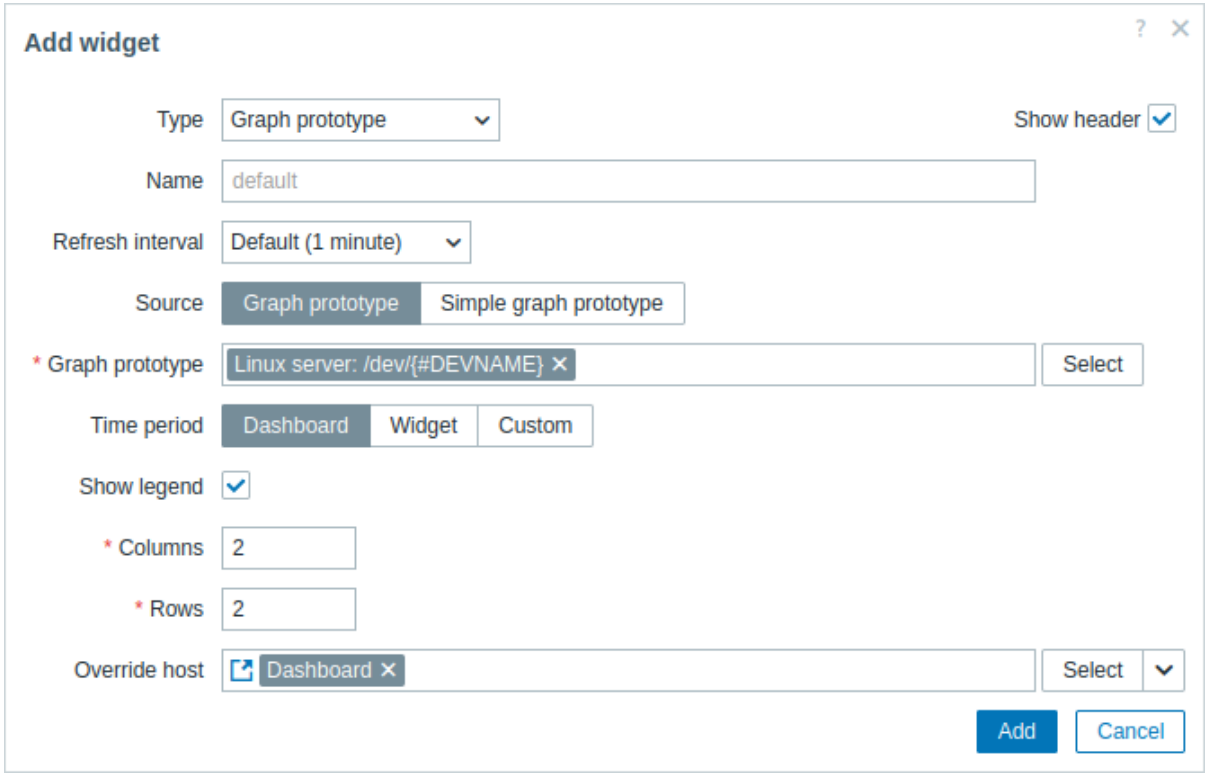
Das Widget *Graph prototype* zeigt ein Raster automatisch erstellter benutzerdefinierter Graphen basierend auf Graph-Prototypen oder Datenpunkt-Prototypen von Low-Level-Discovery-Regeln an.



Die Graphen werden in einem konfigurierbaren Rasterlayout angeordnet; wenn die Anzahl der Graphen das Layout überschreitet, wird im Widget eine Seitennavigation aktiviert.

Konfiguration

Wählen Sie zur Konfiguration *Graph-Prototyp* als Widget-Typ aus:



Zusätzlich zu den für alle Widgets gemeinsamen Parametern können Sie die folgenden spezifischen Optionen festlegen:

Quelle	Wählen Sie die Quelle der Graphen aus: **Graph-Prototyp** oder **Einfacher Graph-Prototyp**.
Graph-Prototyp	Wählen Sie einen Graph-Prototyp aus, um die durch den Graph-Prototyp ermittelten Graphen anzuzeigen. Dieser Parameter ist verfügbar, wenn *Quelle* auf „Graph-Prototyp“ gesetzt ist.
Datenpunkt-Prototyp	Wählen Sie einen Datenpunkt-Prototyp aus, um einfache Graphen für die durch den Datenpunkt-Prototyp ermittelten Datenpunkte anzuzeigen. Dieser Parameter ist verfügbar, wenn *Quelle* auf „Einfacher Graph-Prototyp“ gesetzt ist.
Zeitraum	Legen Sie einen Zeitraum fest, für den Daten in den Graphen angezeigt werden sollen. Wählen Sie die Datenquelle für den Zeitraum aus: **Dashboard** – verwendet die Zeitraumauswahl des Dashboards; **Widget** – verwendet ein kompatibles Widget (im Parameter *Widget* festgelegt); **Benutzerdefiniert** – verwendet einen benutzerdefinierten Zeitraum, der in den Parametern *Von* und *Bis* festgelegt ist; wenn dieser festgelegt ist, wird in der oberen rechten Ecke des Widgets ein Uhrsymbol angezeigt, das bei einem Mouseover die festgelegte Zeit angibt. Beachten Sie, dass kompatible Widgets unabhängig von der *Zeitraum*-Konfiguration des Widgets weiterhin als Datenquelle für den Zeitraum verwendet werden können.
Widget	Geben Sie ein kompatibles Widget als Datenquelle für den Zeitraum ein oder wählen Sie es aus. Dieser Parameter ist verfügbar, wenn *Zeitraum* auf „Widget“ gesetzt ist.
Von	Geben Sie den Beginn des Zeitraums ein oder wählen Sie ihn aus. Die Syntax für relative Zeitangaben (`now`, `now/d`, `now/w-1w` usw.) wird unterstützt. Dieser Parameter ist verfügbar, wenn *Zeitraum* auf „Benutzerdefiniert“ gesetzt ist.
Bis	Geben Sie das Ende des Zeitraums ein oder wählen Sie es aus. Die Syntax für relative Zeitangaben (`now`, `now/d`, `now/w-1w` usw.) wird unterstützt. Dieser Parameter ist verfügbar, wenn *Zeitraum* auf „Benutzerdefiniert“ gesetzt ist.
Legende anzeigen	Deaktivieren Sie dieses Kontrollkästchen, um die Legende in den Graphen auszublenden (standardmäßig aktiviert).
Host überschreiben	Wählen Sie ein kompatibles Widget oder die Host-Auswahl des Dashboards als Datenquelle für Hosts aus. Dieser Parameter ist nicht verfügbar, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Spalten	Geben Sie die Anzahl der Graphspalten ein, die innerhalb eines Graph-Prototyp-Widgets angezeigt werden sollen.
Zeilen	Geben Sie die Anzahl der Graphzeilen ein, die innerhalb eines Graph-Prototyp-Widgets angezeigt werden sollen.

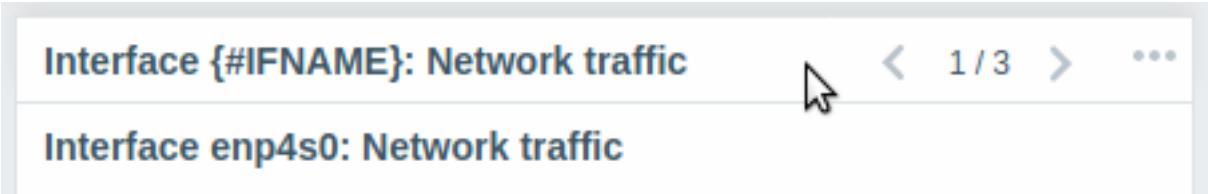
Die Legende des Graphen besteht aus drei Abschnitten:

- Datenpunkte und ihre aggregierten Werte
- Perzentile (falls konfiguriert)
- Auslöser (falls den angezeigten Datenpunkten welche zugeordnet sind)

Wenn die Höhe des Graphen innerhalb des Widgets nicht ausreicht, wird die Legende möglicherweise nicht oder nur teilweise angezeigt. Auslöser und Perzentile werden zuerst ausgeblendet, gefolgt von der Datenpunktlegende. Um die vollständige Legende anzuzeigen, vergrößern Sie die vertikale Größe des Widgets.

Note:
Es können höchstens 3 Auslöserzeilen angezeigt werden. Wenn mehr Auslöser vorhanden sind, werden Auslöser mit niedrigerem Schweregrad bevorzugt angezeigt.

Mit den Parametern *Spalten* und *Zeilen* können zwar mehrere Graphen im Widget untergebracht werden, dennoch kann es mehr ermittelte Graphen geben, als Spalten/Zeilen im Widget vorhanden sind. In diesem Fall wird die Seitennavigation im Widget verfügbar, und eine ausklappbare Kopfzeile ermöglicht den Wechsel zwischen den Seiten mithilfe der Pfeile nach links und rechts:

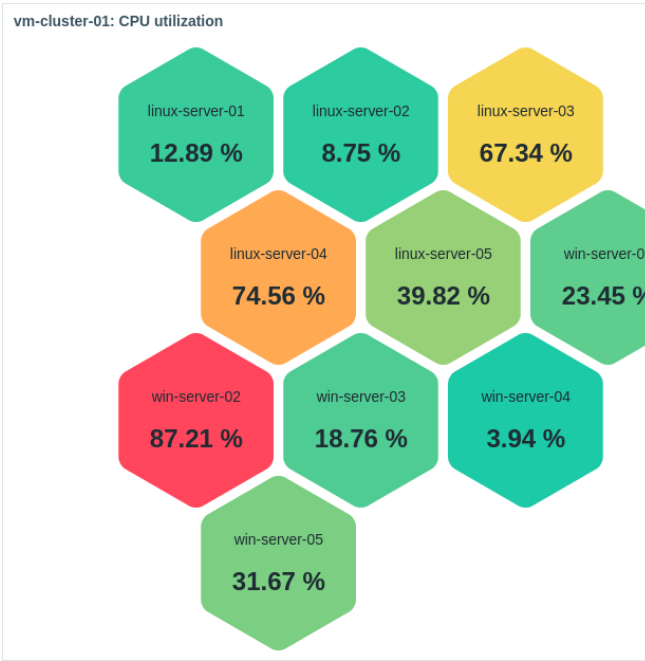


Honeycomb

Übersicht

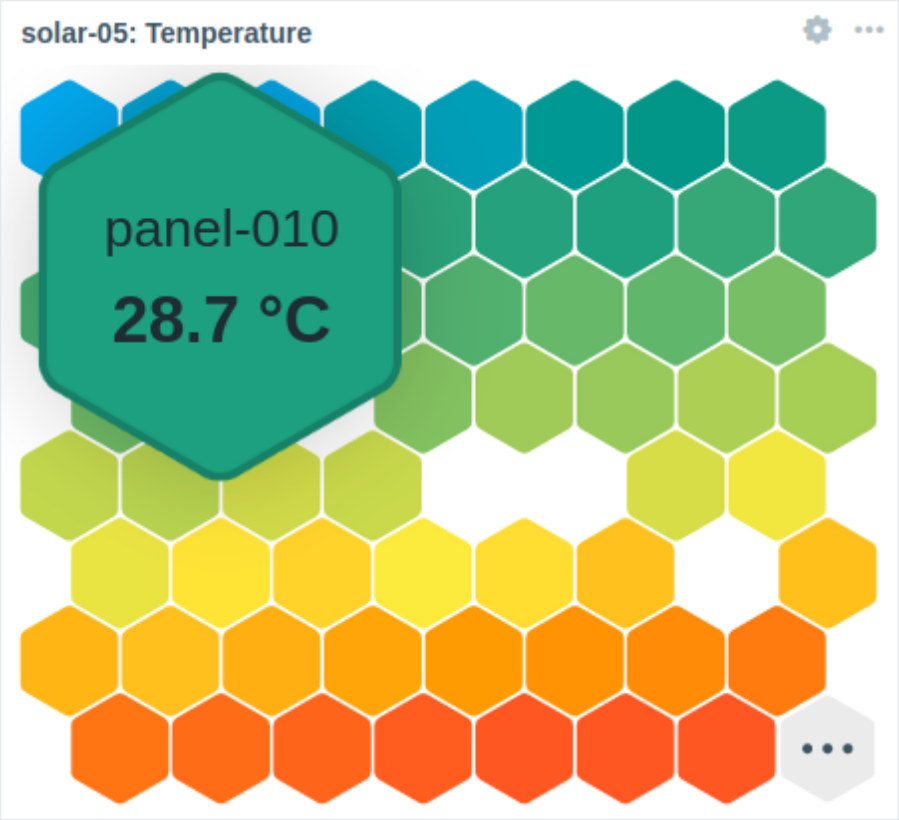
Das Widget *Honeycomb* zeigt Hosts oder Datenpunkte als sechseckige Zellen in einem Raster an. Dieses Layout erleichtert das Erkennen problematischer Hosts/Datenpunkte, das Auffinden von Problemclustern, den Vergleich von Gruppen auf einen Blick und vieles mehr.





Die Anzahl der angezeigten Zellen wird durch die Widget-Größe und die minimale Zellengröße (32 px) begrenzt. Wenn nicht alle Zellen hineinpassen, zeigt die letzte Zelle ein Auslassungszeichen an. Das Widget kann in der Größe angepasst werden, wobei sich die Anzahl, Größe und Positionierung der Zellen dynamisch ändern.

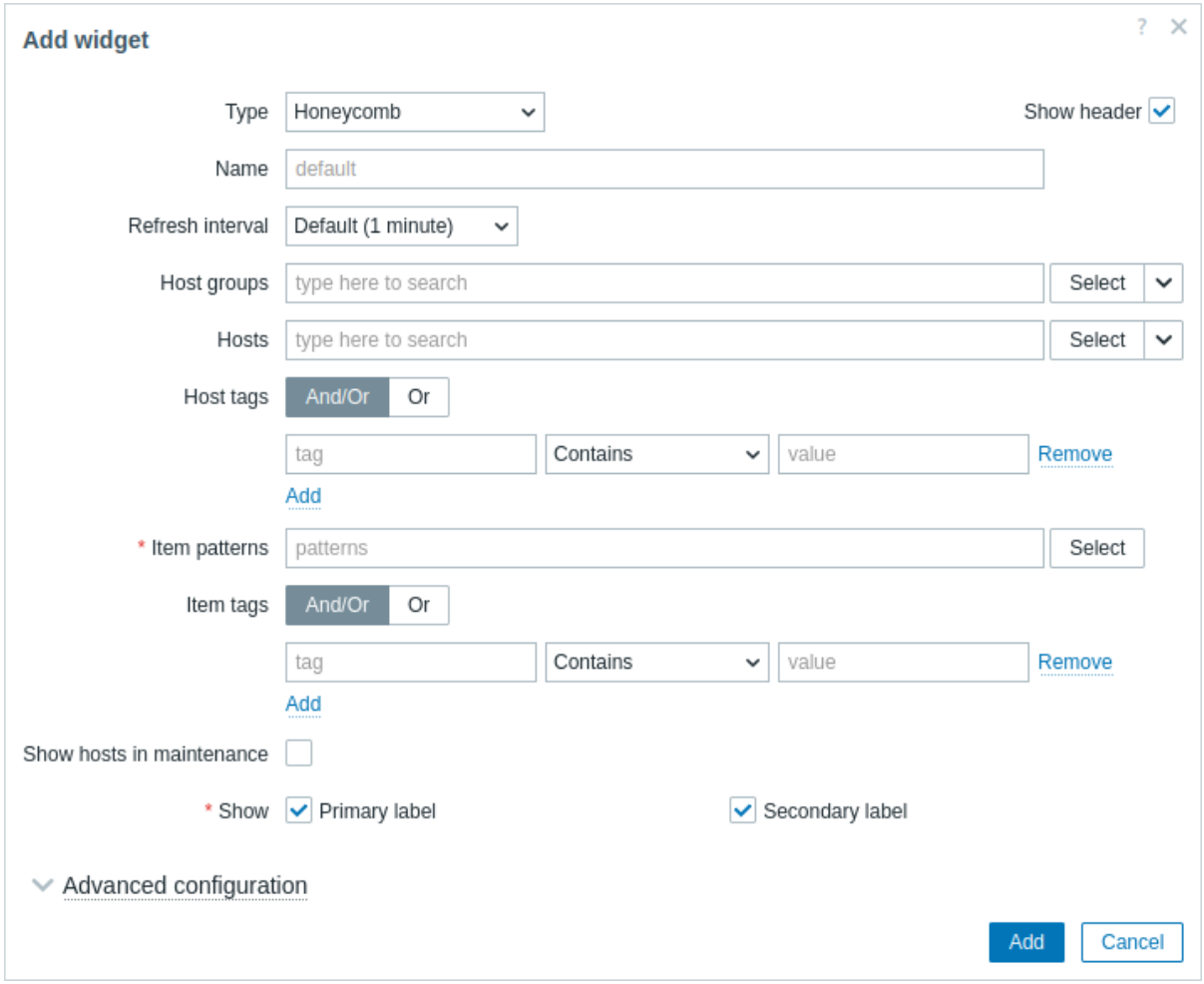
Beim Überfahren mit der Maus wird die fokussierte Zelle zur besseren Sichtbarkeit vergrößert. Durch Klicken auf eine Zelle wird deren Rand hervorgehoben, bis eine andere Zelle ausgewählt wird.



Die im Widget *Honeycomb* angezeigten Informationen können als PNG-Bild heruntergeladen werden, indem Sie im Widget-Menü *Aktionen* …, das beim Überfahren des Widgets mit der Maus angezeigt wird, die Option *Bild herunterladen* auswählen. Abhängig von Ihren Browsereinstellungen wird das Bild entweder automatisch im Ordner `Downloads` gespeichert oder ein Dialogfeld *Speichern* geöffnet, in dem Sie den Speicherort auswählen können.

Konfiguration

Um die Konfiguration vorzunehmen, wählen Sie *Honeycomb* als Typ aus:



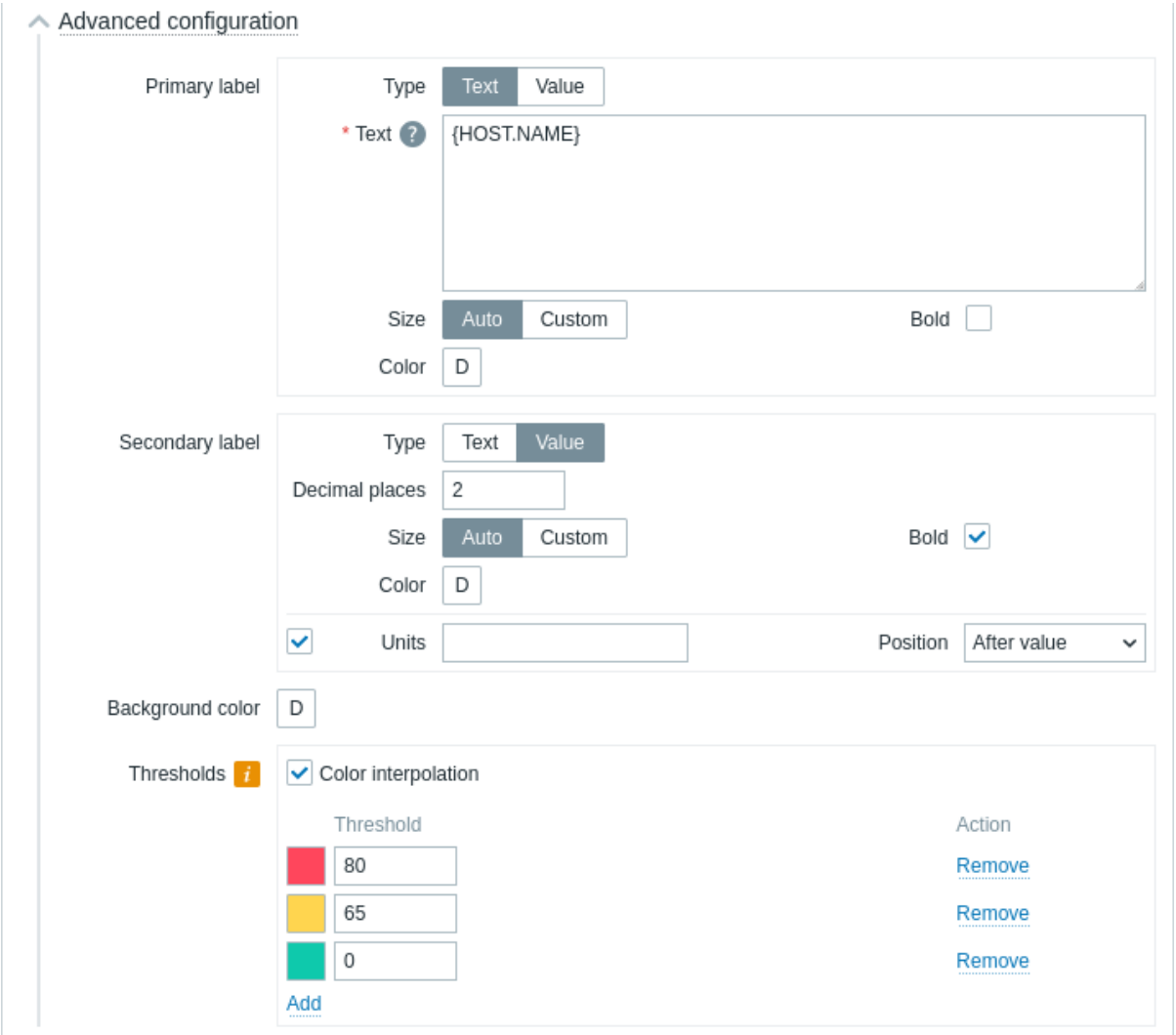
Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

Host-Gruppen	Wählen Sie Host-Gruppen aus. Alternativ können Sie ein kompatibles Widget als Datenquelle für Host-Gruppen auswählen. Dieses Feld verfügt über eine Autovervollständigung; wenn Sie beginnen, den Namen einer Gruppe einzugeben, wird eine Dropdown-Liste mit passenden Gruppen angeboten. Durch die Auswahl einer übergeordneten Host-Gruppe werden implizit alle darin verschachtelten Host-Gruppen ausgewählt; wenn keine Host-Gruppen ausgewählt sind, zeigt das Widget alle Host-Gruppen an, die Hosts mit Datenpunkten enthalten, die dem ausgewählten *Datenpunktmuster* entsprechen (siehe unten). Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Hosts	Wählen Sie Hosts aus. Alternativ können Sie ein kompatibles Widget oder das Dashboard als Datenquelle für Hosts auswählen. Dieses Feld verfügt über eine Autovervollständigung; wenn Sie beginnen, den Namen eines Hosts einzugeben, wird eine Dropdown-Liste mit passenden Hosts angeboten. Wenn keine Hosts ausgewählt sind, zeigt das Widget alle Hosts mit Datenpunkten an, die dem ausgewählten *Datenpunktmuster* entsprechen (siehe unten). Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.

Host-Tags	Geben Sie Tags an, um die Anzahl der im Widget angezeigten Hosts zu begrenzen. Es ist möglich, bestimmte Tags und Tag-Werte sowohl einzuschließen als auch auszuschließen. Es können mehrere Bedingungen festgelegt werden. Der Abgleich von Tag-Namen ist immer groß-/kleinschreibungssensitiv. Für jede Bedingung stehen mehrere Operatoren zur Verfügung: **Existiert** - die angegebenen Tag-Namen einschließen; **Gleich** - die angegebenen Tag-Namen und -Werte einschließen (groß-/kleinschreibungssensitiv); **Enthält** - die angegebenen Tag-Namen einschließen, bei denen die Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv); **Existiert nicht** - die angegebenen Tag-Namen ausschließen; **Ungleich** - die angegebenen Tag-Namen und -Werte ausschließen (groß-/kleinschreibungssensitiv); **Enthält nicht** - die angegebenen Tag-Namen ausschließen, bei denen die Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv). Für Bedingungen gibt es zwei Berechnungstypen: **Und/Oder** - alle Bedingungen müssen erfüllt sein; Bedingungen mit demselben Tag-Namen werden durch die Oder-Bedingung gruppiert; **Oder** - es genügt, wenn eine Bedingung erfüllt ist. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Datenpunktmuster	Geben Sie Datenpunktmuster ein oder wählen Sie vorhandene Datenpunkte als Datenpunktmuster aus. Daten von Datenpunkten, die den eingegebenen oder ausgewählten Mustern entsprechen, werden in der Honeycomb angezeigt. Der Parameter *Datenpunktmuster* ist obligatorisch. Für die Auswahl können Platzhaltermuster verwendet werden (zum Beispiel gibt * Datenpunkte zurück, die null oder mehr Zeichen entsprechen; `Zabbix*` gibt Datenpunkte zurück, die mit „Zabbix“ beginnen). Um ein Platzhaltermuster anzugeben, geben Sie die Zeichenfolge manuell ein und drücken Sie *Enter*. Wenn Sie mit der Eingabe beginnen, zeigt eine Dropdown-Liste passende Datenpunkte an, beschränkt auf diejenigen, die zu ausgewählten *Hosts* oder zu Hosts innerhalb ausgewählter *Host-Gruppen* gehören, falls vorhanden. Das Platzhaltersymbol wird immer interpretiert; daher ist es nicht möglich, zum Beispiel einen Datenpunkt mit dem Namen *item** einzeln hinzuzufügen, wenn es andere passende Datenpunkte gibt (z. B. item2, item3). Wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird, erlaubt dieser Parameter nur die Auswahl von auf der Vorlage konfigurierten Datenpunkten.
Datenpunkt-Tags	Geben Sie Tags an, um die Anzahl der im Widget angezeigten Datenpunkte zu begrenzen. Weitere Informationen finden Sie oben unter *Host-Tags*.
Hosts in Wartung anzeigen	Aktivieren Sie dieses Kontrollkästchen, um Hosts in Wartung anzuzeigen (in diesem Fall wird neben dem Host-Namen ein Wartungssymbol angezeigt). Dieser Parameter ist mit *Daten in Wartung anzeigen* beschriftet, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Anzeigen	Aktivieren Sie dieses Kontrollkästchen, um das jeweilige Honeycomb-Zellenelement anzuzeigen – primäre Beschriftung, sekundäre Beschriftung. Mindestens ein Element muss ausgewählt sein.
Erweiterte Konfiguration	Klicken Sie auf die Überschrift *Erweiterte Konfiguration*, um die Optionen der erweiterten Konfiguration anzuzeigen.

Erweiterte Konfiguration

Erweiterte Konfigurationsoptionen sind im einklappbaren Abschnitt *Erweiterte Konfiguration* verfügbar und nur für Elemente ausgewählt, die im Feld *Anzeigen* ausgewählt wurden (siehe oben), sowie für die Hintergrundfarbe oder Schwellenwerte von Wabenzellen.



Primäre/Sekundäre Beschriftung

Typ — Wählen Sie den Beschriftungstyp aus:
Text - die Beschriftung zeigt den für den Parameter *Text* angegebenen Text an;
Wert - die Beschriftung zeigt den Datenpunktwert mit der für den Parameter *Dezimalstellen* angegebenen Anzahl an Dezimalstellen an.

Text — Geben Sie den Beschriftungstext ein. Dieser Text kann den standardmäßigen Datenpunktnamen überschreiben.
Mehrzeiliger Text wird unterstützt. Eine Kombination aus Text und unterstützten Makros ist möglich.
`{HOST.\*}`, `{ITEM.\*}`, `{INVENTORY.\*}` und Benutzermakros werden unterstützt.
Wabenzellen werden alphabetisch nach Hostname und innerhalb jedes Hosts nach Datenpunktname sortiert.
Dieser Parameter ist verfügbar, wenn *Typ* auf *Text* gesetzt ist.

Dezimalstellen — Geben Sie die Anzahl der mit dem Wert anzuzeigenden Dezimalstellen ein.
Dieser Parameter ist verfügbar, wenn *Typ* auf *Wert* gesetzt ist, und betrifft nur Datenpunkte, die Daten vom Typ numerisch (Gleitkommazahl) zurückgeben.

Größe — Wählen Sie die Beschriftungsgröße aus:
Automatisch - automatisch angepasste Beschriftungsgröße verwenden;
Benutzerdefiniert - eine benutzerdefinierte Beschriftungsgröße eingeben (in Prozent, relativ zur Größe der Wabenzelle).
Beachten Sie, dass Beschriftungen, die nicht in die Größe der Wabenzelle passen, abgeschnitten werden.

Fett — Aktivieren Sie das Kontrollkästchen, um Beschriftungen fett darzustellen.

Farbe — Wählen Sie die Beschriftungsfarbe in der Farbauswahl aus.
”D” steht für die Standardfarbe, die vom Frontend-Theme abhängt. Um zur Standardfarbe zurückzukehren, klicken Sie in der Farbauswahl auf die Schaltfläche *Standard verwenden*.

Einheiten

Einheiten	Aktivieren Sie das Kontrollkästchen, um Einheiten zusammen mit dem Datenpunktwert anzuzeigen. Wenn Sie einen Einheitsnamen eingeben, überschreibt dieser die in der Datenpunktkonfiguration festgelegten Einheiten. Dieser Parameter ist verfügbar, wenn *Typ* auf *Wert* gesetzt ist.
Position	Wählen Sie die Position der Datenpunkteinheiten aus (vor oder nach dem Datenpunktwert). Dieser Parameter wird für die folgenden zeitbezogenen Einheiten ignoriert: unixtime, uptime, s. Dieser Parameter ist verfügbar, wenn *Typ* auf *Wert* gesetzt ist und die Option *Einheiten* ausgewählt ist.
Hintergrundfarbe	
Hintergrundfarbe	Wählen Sie die Hintergrundfarbe der Wabenzellen in der Farbauswahl aus. ”D” steht für die Standardfarbe, die vom Frontend-Theme abhängt. Um zur Standardfarbe zurückzukehren, klicken Sie in der Farbauswahl auf die Schaltfläche *Standard verwenden*.
Schwellenwerte	
Farbinterpolation	Aktivieren Sie das Kontrollkästchen, um weiche Übergänge zwischen Schwellenwertfarben für Wabenzellen zu aktivieren. Dieser Parameter ist verfügbar, wenn zwei oder mehr Schwellenwerte festgelegt sind.
Schwellenwert	Klicken Sie auf *Hinzufügen*, um einen Schwellenwert hinzuzufügen, wählen Sie eine Schwellenwertfarbe in der Farbauswahl aus und geben Sie einen numerischen Wert an. Die Liste der Schwellenwerte wird beim Speichern in aufsteigender Reihenfolge sortiert. Beachten Sie, dass die als Schwellenwerte konfigurierten Farben nur für numerische Datenpunkte korrekt angezeigt werden. Suffixe (zum Beispiel „1d“, „2w“, „4K“, „8G“) werden unterstützt. Wertzuordnungen werden unterstützt.

Host-Verfügbarkeit

Übersicht

Das Widget *Host-Verfügbarkeit* zeigt die Verfügbarkeit von Hosts in ausgewählten Host-Gruppen an und ermöglicht es Ihnen, die Anzahl der Hosts zu überwachen, die erreichbar oder nicht erreichbar sind.

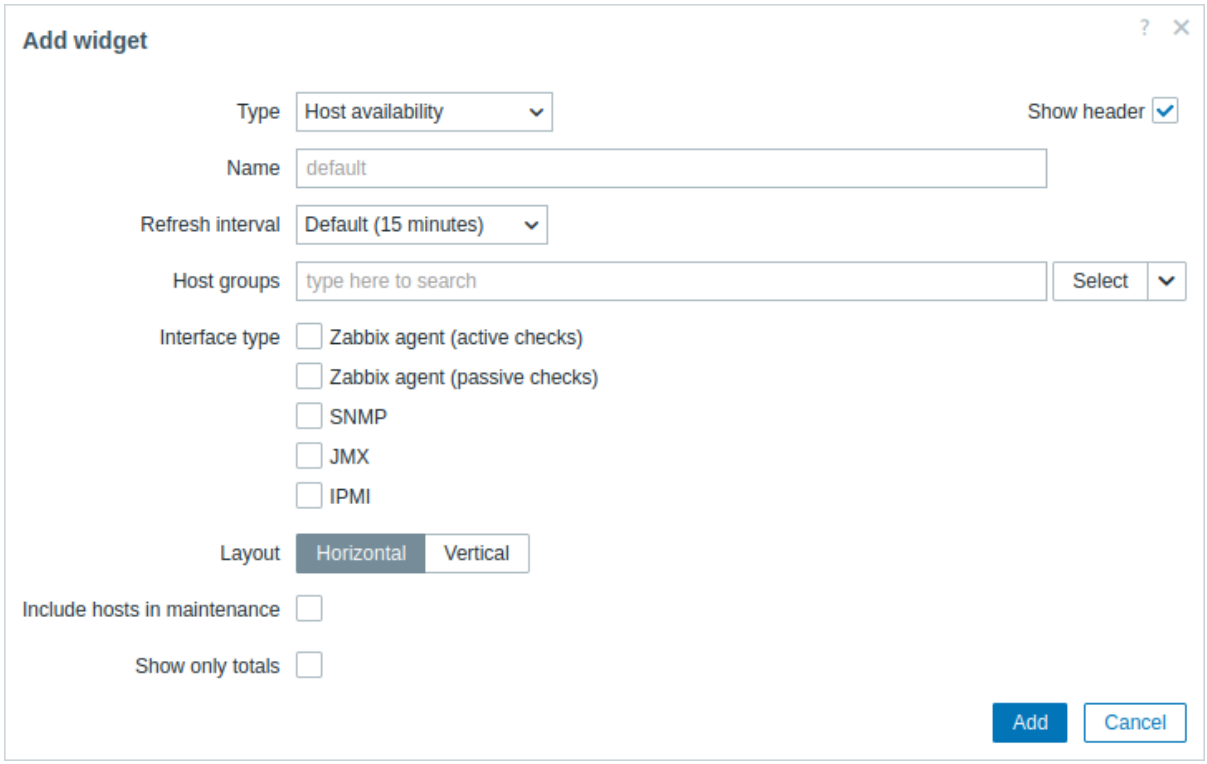


Die Host-Verfügbarkeit wird wie folgt gezählt:

- *Verfügbar* - Hosts, bei denen alle Schnittstellen verfügbar sind
- *Nicht verfügbar* - Hosts, bei denen alle Schnittstellen nicht verfügbar sind
- *Gemischt* - Hosts mit mindestens einer nicht verfügbaren Schnittstelle und mindestens einer verfügbaren oder unbekannten Schnittstelle
- *Unbekannt* - Hosts mit mindestens einer unbekannten Schnittstelle, aber keiner nicht verfügbaren
- *Gesamt* - Gesamtzahl aller Hosts

Konfiguration

Um die Konfiguration vorzunehmen, wählen Sie *Host-Verfügbarkeit* als Typ aus:



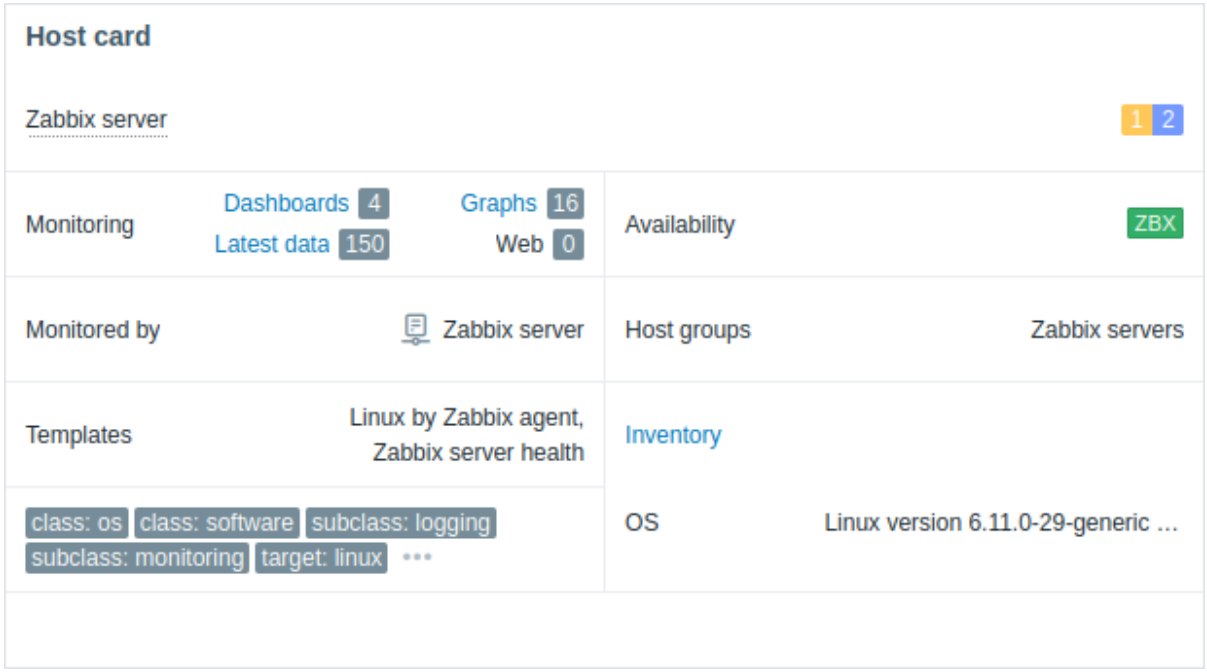
Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

Host-Gruppen	Wählen Sie Host-Gruppen aus. Alternativ können Sie ein kompatibles Widget als Datenquelle für Host-Gruppen auswählen. Dieses Feld unterstützt Autovervollständigung; wenn Sie also beginnen, den Namen einer Gruppe einzugeben, wird eine Dropdown-Liste mit passenden Gruppen angeboten. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Schnittstellentyp	Wählen Sie aus, für welche Host-Schnittstellen Sie Verfügbarkeits-daten sehen möchten – *Zabbix-Agent (aktive Prüfungen)*, *Zabbix-Agent (passive Prüfungen)*, *SNMP*, *JMX*, *IPMI*. Bei *Zabbix-Agent (aktive Prüfungen)* ist die Zelle *Gemischt* immer leer, da Datenpunkte dieses Typs nicht mehrere Schnittstellen haben können. Wenn nichts ausgewählt ist, wird standardmäßig die Verfügbarkeit aller Schnittstellen angezeigt.
Layout	Wählen Sie die horizontale Anzeige (Spalten) oder die vertikale Anzeige (Zeilen).
Hosts in Wartung einschließen	Beziehen Sie Hosts, die sich in Wartung befinden, in die Statistik ein. Dieser Parameter ist mit *Daten in Wartung anzeigen* beschriftet, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Nur Summen anzeigen	Wenn diese Option aktiviert ist, wird die Gesamtzahl der Hosts ohne Aufschlüsselung nach Schnittstellen angezeigt. Diese Option ist deaktiviert, wenn nur eine Schnittstelle ausgewählt ist.

Host-Karte

Übersicht

Das Widget *Host card* zeigt Details zu einem einzelnen Host an und ermöglicht es Ihnen, den Status und die Konfiguration eines Hosts schnell zu beurteilen (Verfügbarkeit, Probleme, Inventar usw.).

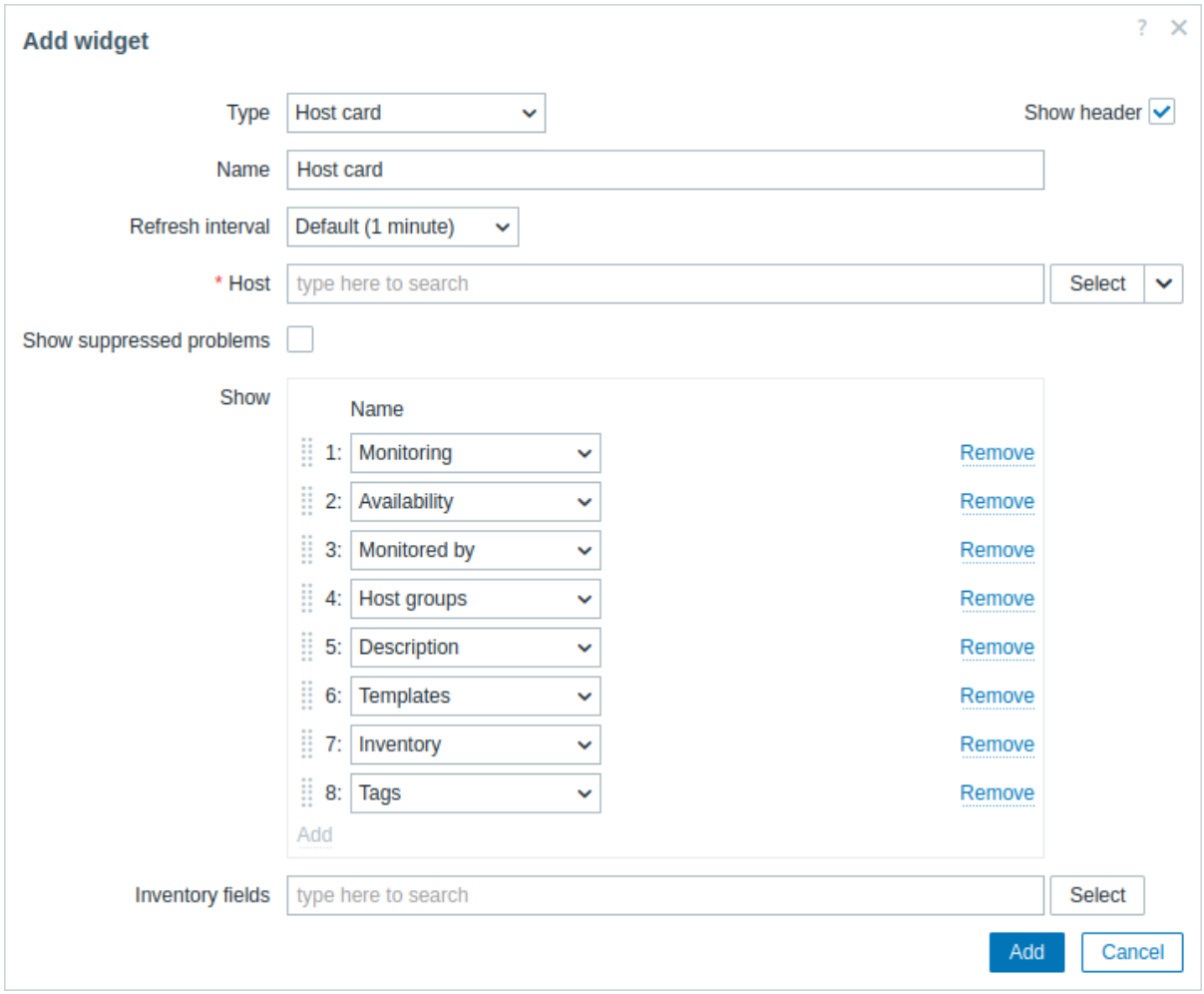


Der Host kann direkt in der Widget-Konfiguration festgelegt oder entweder aus einem kompatiblen Widget oder über den Host-Selektor im Dashboard ausgewählt werden.

Das Widget kann mehrere Abschnitte anzeigen, die jeweils unterschiedliche Informationen darstellen. Das Layout der Abschnitte wird automatisch an die Breite des Widgets angepasst; bei horizontaler Erweiterung werden die Abschnitte automatisch in mehrere Spalten umgeordnet.

Konfiguration

Wählen Sie zur Konfiguration als Typ *Host card* aus:



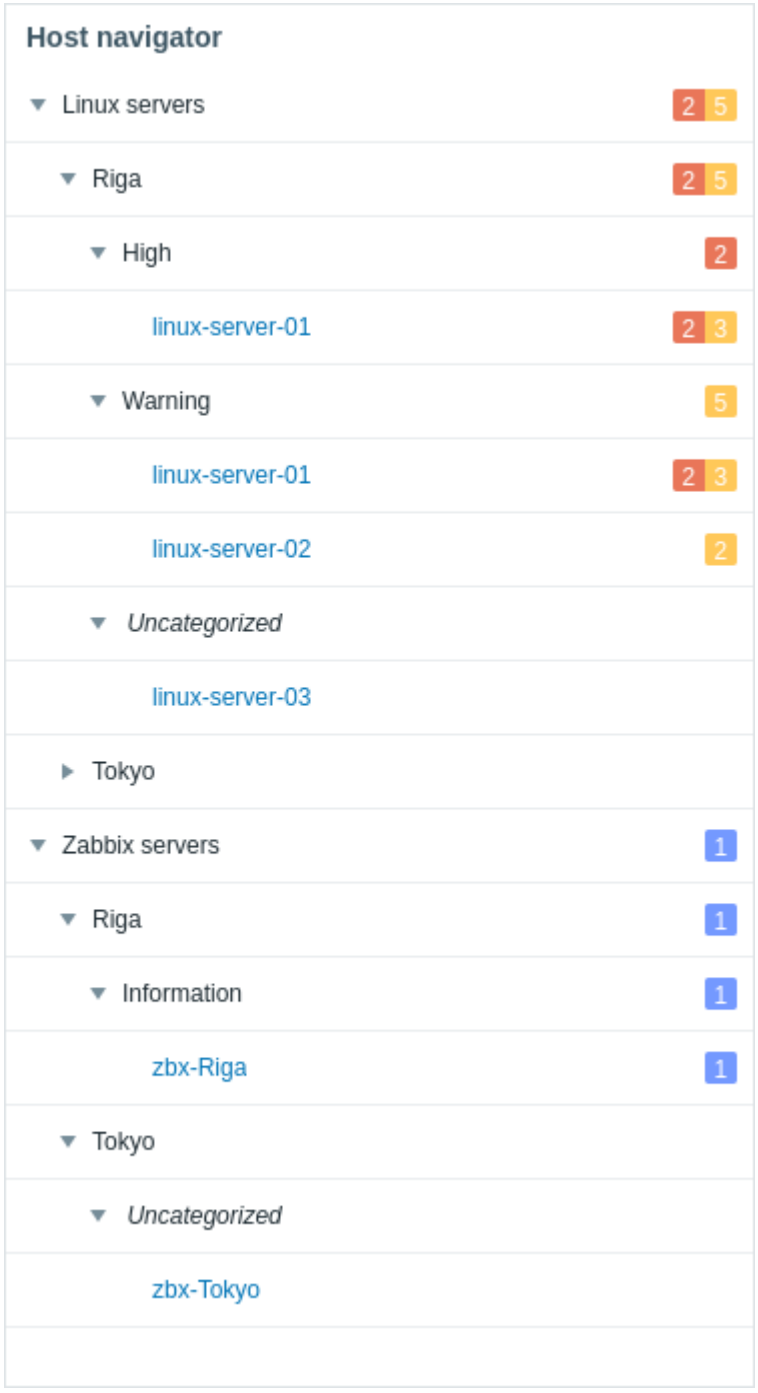
Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Parameter festlegen:

Host	Wählen Sie den Host aus. Alternativ können Sie ein kompatibles Widget oder das Dashboard als Datenquelle für Hosts auswählen. Während der Eingabe werden passende Vorschläge angezeigt. Im oberen Abschnitt des Widgets wird immer der **Host-Name** angezeigt (ein Klick darauf öffnet das Host-Menü): - Wenn der Host deaktiviert ist, wird neben dem Host-Namen eine rote Kennzeichnung *Deaktiviert* angezeigt. - Wenn sich der Host in Wartung befindet, wird neben dem Host-Namen das Wartungssymbol angezeigt. Im oberen Abschnitt des Widgets wird außerdem die Anzahl der **Host-Probleme** angezeigt, gruppiert nach Schweregrad (ein Klick darauf öffnet die Probleme des Hosts). Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Show suppressed problems	Aktivieren Sie dieses Kontrollkästchen, um unterdrückte Probleme in die Problemberechnung einzubeziehen.
Show	Fügen Sie Abschnitte hinzu, die im Widget angezeigt werden sollen. Abschnitte können per Drag-and-drop neu angeordnet werden.
Host groups	Zeigt die Host-Gruppen an, denen der Host angehört.
Description	Zeigt die Beschreibung des Hosts an.
Monitoring	Zeigt Schnelllinks zur Navigation zu den Dashboards, Letzte Daten, Graphen und Webszenarien des Hosts an. Die Zahl neben jedem Link gibt die Anzahl der jeweiligen Entitäten an.
Availability	Zeigt die Verfügbarkeit des Hosts nach Schnittstelle an.
Monitored by	Zeigt an, ob der Host vom Zabbix-Server oder von einem bestimmten Proxy oder einer Proxy-Gruppe überwacht wird.
Templates	Zeigt die mit dem Host verknüpften Vorlagen an.
Inventory	Zeigt die Inventarfelder des Hosts an.
Tags	Zeigt die Tags des Hosts an.
Inventory fields	Wählen Sie die Inventarfelder aus, die angezeigt werden sollen. Während der Eingabe werden passende Vorschläge angezeigt. Wenn keine Inventarfelder angegeben sind, werden alle ausgefüllten Inventarfelder angezeigt. Dieser Parameter ist verfügbar, wenn *Show Inventory* enthält.

Host-Navigator

Übersicht

Das Widget *Host navigator* zeigt eine Liste von Hosts basierend auf verschiedenen Filter- und Gruppierungsoptionen an.



Gruppen, nach denen Hosts organisiert sind, können ein- oder ausgeklappt werden.

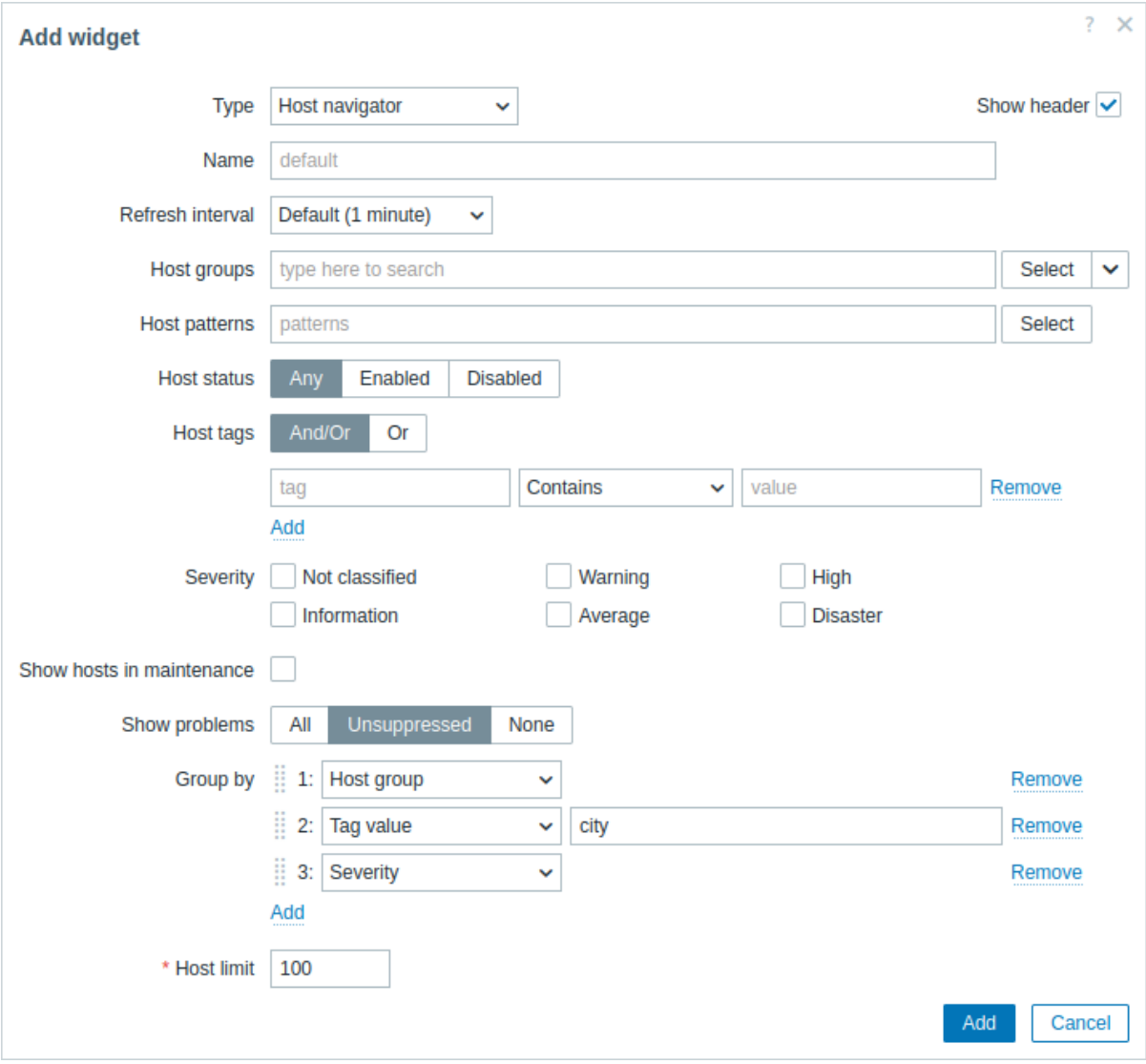
Für Gruppen, Probleme und Hosts in Wartung sind zusätzliche Details über Mouseover-Hinweise verfügbar.

Das Widget ist besonders nützlich, um zu steuern, was andere Widgets basierend auf dem ausgewählten Host anzeigen.



Konfiguration

Wählen Sie zur Konfiguration *Host navigator* als Typ aus:



Zusätzlich zu den für alle Widgets gemeinsamen Parametern können Sie die folgenden spezifischen Optionen festlegen:

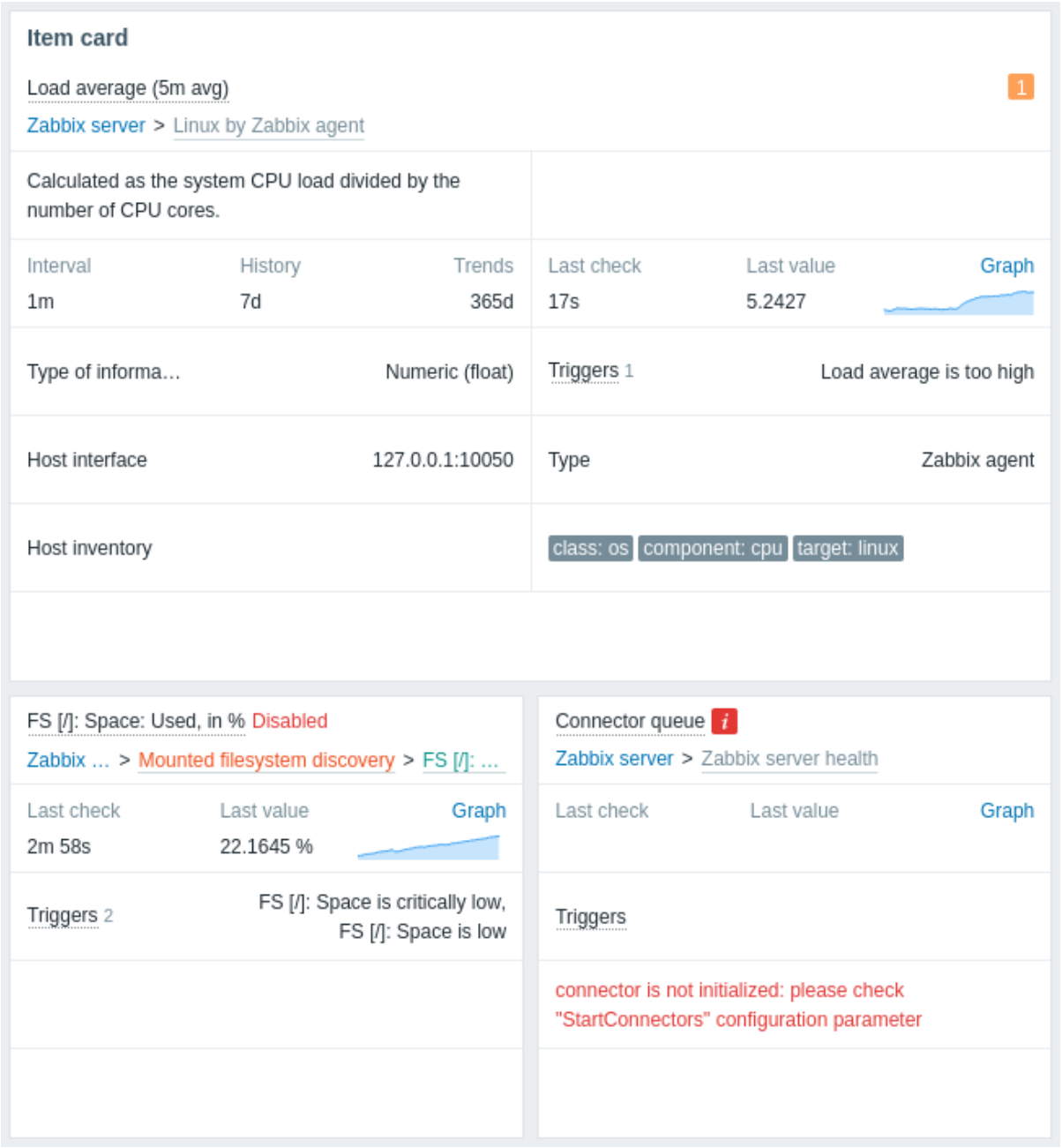
Host groups	Wählen Sie Hostgruppen aus. Alternativ können Sie ein kompatibles Widget als Datenquelle für Hostgruppen auswählen. Dieses Feld unterstützt die automatische Vervollständigung. Wenn Sie beginnen, den Namen einer Gruppe einzugeben, wird eine Dropdown-Liste mit passenden Gruppen angezeigt. Durch die Auswahl einer übergeordneten Hostgruppe werden implizit alle verschachtelten Hostgruppen ausgewählt. Wenn keine Hostgruppen ausgewählt sind, zeigt das Widget alle Hosts aus allen Hostgruppen an. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Host patterns	Geben Sie Hostmuster ein oder wählen Sie vorhandene Hosts als Hostmuster aus. Hosts, die den angegebenen Mustern entsprechen, werden im Host navigator angezeigt. Dieses Feld unterstützt die automatische Vervollständigung. Wenn Sie beginnen, den Namen eines Hosts einzugeben, wird eine Dropdown-Liste mit passenden Hosts angezeigt. Wenn keine Hosts ausgewählt sind, zeigt das Widget alle Hosts an. Für die Auswahl können Platzhaltermuster verwendet werden (beispielsweise gibt ∗ Hosts zurück, die null oder mehr Zeichen entsprechen; `Zabbix*` gibt Hosts zurück, die mit „Zabbix“ beginnen). Um ein Platzhaltermuster anzugeben, geben Sie die Zeichenfolge manuell ein und drücken Sie die Eingabetaste. Sobald Sie mit der Eingabe beginnen, zeigt eine Dropdown-Liste passende Hosts an, beschränkt auf Hosts, die zu den Hosts innerhalb der ausgewählten *Host groups* gehören, sofern vorhanden. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Host status	Filtern Sie, welche Hosts anhand ihres Status angezeigt werden sollen (beliebig, aktiviert, deaktiviert). Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Host tags	Geben Sie Tags an, um die im Widget angezeigten Hosts zu filtern. Sie können bestimmte Tags und Tagwerte sowohl einschließen als auch ausschließen. Es können mehrere Bedingungen festgelegt werden. Beim Abgleich von Tagnamen wird immer zwischen Groß- und Kleinschreibung unterschieden. Für jede Bedingung stehen mehrere Operatoren zur Verfügung: **Exists** - die angegebenen Tagnamen einschließen; **Equals** - die angegebenen Tagnamen und Tagwerte einschließen (Groß-/Kleinschreibung wird berücksichtigt); **Contains** - die angegebenen Tagnamen einschließen, deren Tagwerte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich ohne Berücksichtigung der Groß-/Kleinschreibung); **Does not exist** - die angegebenen Tagnamen ausschließen; **Does not equal** - die angegebenen Tagnamen und Tagwerte ausschließen (Groß-/Kleinschreibung wird berücksichtigt); **Does not contain** - die angegebenen Tagnamen ausschließen, deren Tagwerte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich ohne Berücksichtigung der Groß-/Kleinschreibung). Für Bedingungen gibt es zwei Berechnungstypen: **And/Or** - alle Bedingungen müssen erfüllt sein; Bedingungen mit demselben Tagnamen werden mit der Oder-Bedingung gruppiert; **Or** - es genügt, wenn eine Bedingung erfüllt ist. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Severity	Markieren Sie Problemprioritäten, um Hosts mit Problemen zu filtern, die im Widget angezeigt werden sollen. Wenn keine Prioritäten markiert sind, werden alle Hosts mit allen Problemen angezeigt.

Show hosts in maintenance	Aktivieren Sie dieses Kontrollkästchen, um Hosts im Wartungszeitraum anzuzeigen (in diesem Fall wird neben dem Hostnamen ein Wartungssymbol angezeigt). Dieser Parameter trägt den Namen *Show data in maintenance*, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Show problems	Filtern Sie, welche Probleme anhand ihres Status angezeigt werden sollen (alle, nicht unterdrückte, keine). Probleme werden durch farbige Blöcke am Ende der Hostzeile angezeigt. Die Farben basieren auf der Problempriorität und können unter *Administration* > *General* > *Trigger displaying options* angepasst werden. Problemprioritäten können beim Aktualisieren von Problemen geändert werden.
Group by	Fügen Sie ein Gruppierungsattribut hinzu, nach dem die ausgewählten Hosts gruppiert werden sollen: **Host group** - Hosts nach ihrer Hostgruppe gruppieren; **Tag value** - geben Sie einen Tagnamen ein, um Hosts nach den Werten dieses Tags zu gruppieren (geben Sie beispielsweise „city“ ein, um Hosts nach den Werten „Riga“, „Tokyo“ usw. zu gruppieren); **Severity** - Hosts nach ihren Problemprioritäten gruppieren. Wenn *Show problems* so konfiguriert ist, dass Probleme angezeigt werden, werden sie wie folgt dargestellt: - für jede Prioritätsgruppe wird nur die entsprechende Problemanzahl angezeigt; - für jeden Host werden alle seine Problemzahlen angezeigt. Beachten Sie, dass Hosts nur nach den im Parameter *Severity* markierten Prioritäten gruppiert werden. Wenn keine Prioritäten markiert sind, werden alle Hosts nach allen Prioritäten gruppiert. Gruppierungsattribute können mithilfe des Griffs vor dem Gruppennamen durch Ziehen nach oben oder unten neu angeordnet werden. Beachten Sie, dass die Reihenfolge der Gruppierungsattribute die Reihenfolge der Gruppenschachtelung bestimmt. Wenn Sie beispielsweise mehrere Tagnamen angeben (1: color, 2: city), werden Hosts zuerst nach Farbe (rot, blau usw.) und anschließend nach Stadt (Riga, Tokyo usw.) gruppiert. Ein Host kann abhängig von den konfigurierten Gruppierungsattributen in mehreren Gruppen angezeigt werden (beispielsweise bei einer Gruppierung nach Hostgruppe, wenn der Host mehreren Hostgruppen angehört). Durch Klicken auf solche Hosts werden sie in allen Gruppen ausgewählt und hervorgehoben. Hosts, die den konfigurierten Gruppierungsattributen nicht entsprechen, werden in der Gruppe *Uncategorized* angezeigt. Es können bis zu 10 Gruppierungsattribute angegeben werden, und alle müssen eindeutig sein. Wenn keine Gruppierungsattribute angegeben sind, werden Hosts nicht gruppiert.
Host limit	Geben Sie die maximale Anzahl der anzuzeigenden Hosts ein. Zulässige Werte liegen zwischen 1 und 9999. Wenn mehr Hosts zur Anzeige verfügbar sind als das festgelegte Limit, wird unter den angezeigten Hosts eine entsprechende Meldung angezeigt (beispielsweise „100 of 100+ hosts are shown“). Beachten Sie, dass das konfigurierte Hostlimit auch die Anzeige konfigurierter Gruppen beeinflusst. Wenn das Hostlimit beispielsweise auf 100 gesetzt ist und Hosts nach Tagwerten gruppiert werden (mehr als 200), werden im Widget nur die ersten 100 Tagwerte mit den entsprechenden Hosts angezeigt. Dieser Parameter wird durch den Parameter *Limit for search and filter results* unter *Administration* > *General* > *GUI* nicht beeinflusst. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.

Datenpunkt-Karte

Übersicht

Das Widget *Datenpunkt-Karte* zeigt Details zu einem einzelnen Datenpunkt an und ermöglicht es Ihnen, den Status und die Konfiguration eines Datenpunkts schnell zu beurteilen (letzte Daten, Fehler, Beschreibung usw.).

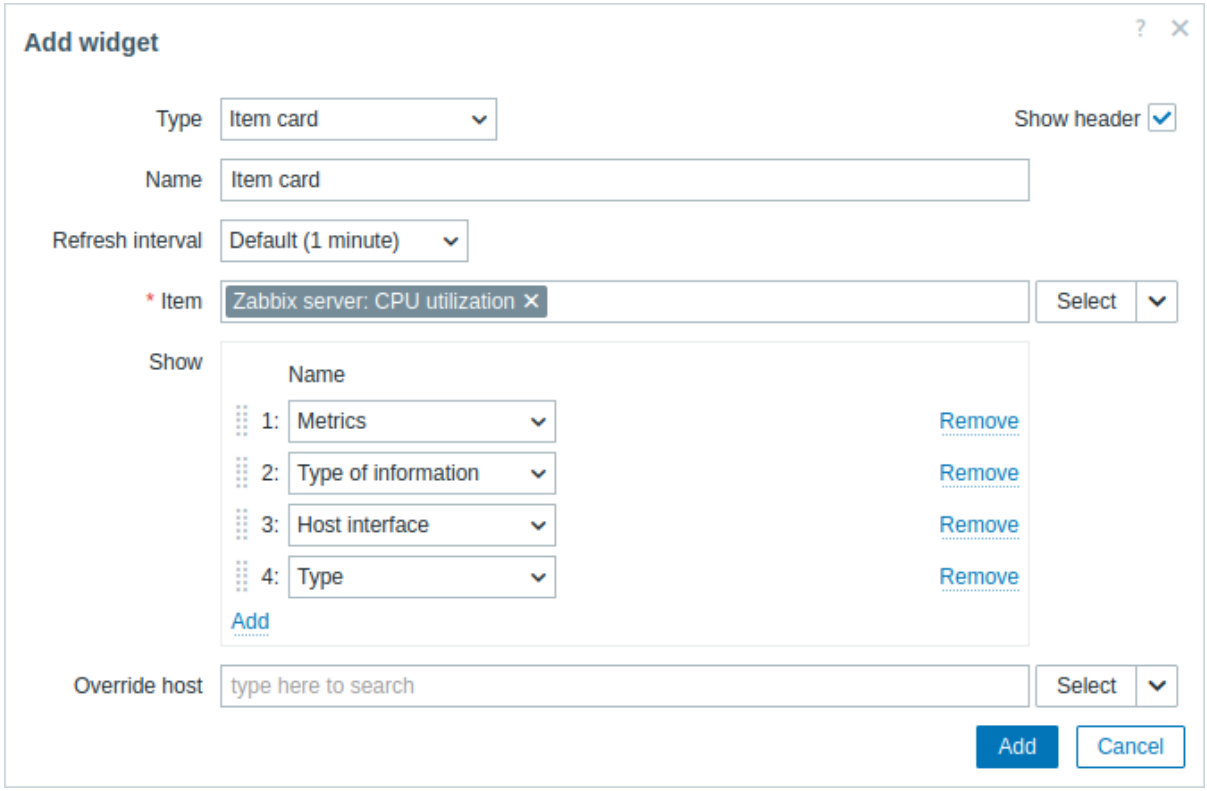


Der Datenpunkt kann direkt in der Widget-Konfiguration festgelegt oder aus einem kompatiblen Widget im Dashboard ausgewählt werden.

Das Widget kann mehrere Abschnitte anzeigen, die jeweils unterschiedliche Informationen darstellen. Das Layout der Abschnitte wird automatisch an die Breite des Widgets angepasst; bei horizontaler Erweiterung werden die Abschnitte automatisch in mehrere Spalten umgeordnet.

Konfiguration

Wählen Sie zur Konfiguration *Datenpunkt-Karte* als Typ aus:



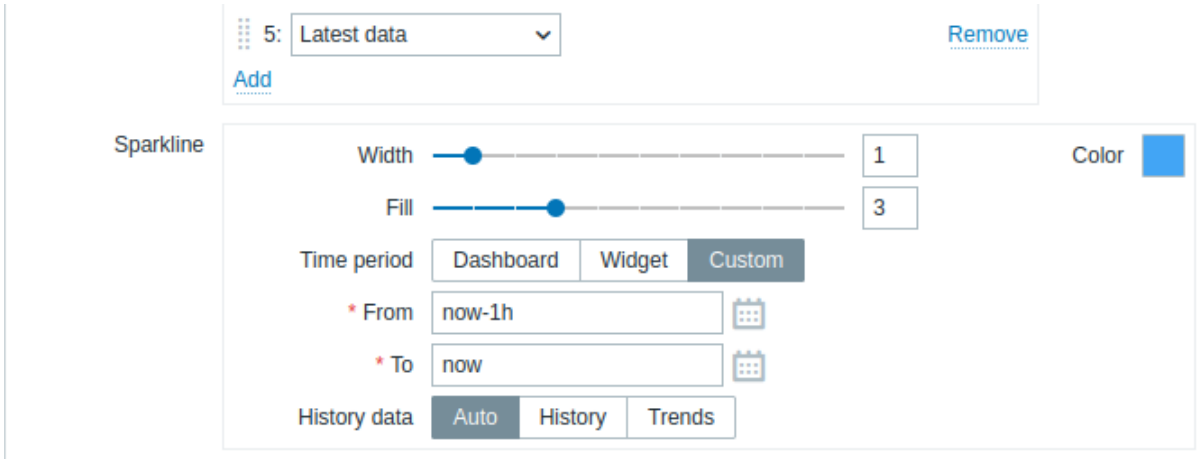
Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

Datenpunkt	Wählen Sie den Datenpunkt aus. Alternativ können Sie ein kompatibles Widget als Datenquelle für Datenpunkte auswählen. Während der Eingabe werden passende Vorschläge angezeigt. Im oberen Abschnitt des Widgets wird immer der **Datenpunktname** angezeigt (ein Klick darauf öffnet das Datenpunkt-Menü): - Wenn der Datenpunkt deaktiviert ist, wird neben dem Datenpunktnamen eine rote Beschriftung *Deaktiviert* angezeigt. - Im Fehlerfall wird ein quadratisches Symbol mit dem Buchstaben „i“ angezeigt. Bei Problemen wird für jeden Problemschweregrad ein quadratisches Symbol mit der Anzahl der Probleme angezeigt. Wenn Sie den Mauszeiger über ein Symbol bewegen, wird ein Pop-up mit Details geöffnet. - Auf einem Vorlagen-Dashboard wird der technische Name des Datenpunkts (anstelle des sichtbaren Namens) angezeigt. Im oberen Abschnitt des Widgets wird außerdem der **Datenpunktpfad** angezeigt – sein Host und, falls zutreffend, Vorlage, Low-Level-Discovery-Regel (LLD) oder Master-Datenpunkt (ein Klick auf eine Entität öffnet deren Konfiguration). Der Datenpunktpfad wird nur Benutzern mit Berechtigungen für die jeweiligen Entitäten angezeigt.
Anzeigen	Fügen Sie Abschnitte hinzu, die im Widget angezeigt werden sollen. Abschnitte können per Drag-and-drop neu angeordnet werden.
Beschreibung	Zeigt die Beschreibung des Datenpunkts an. Makros werden aufgelöst.
Fehlertext	Zeigt den Fehler des Datenpunkts an (falls vorhanden).
Metriken	Zeigt die folgenden Informationen an: - Aktualisierungsintervall des Datenpunkts (außer bei Trapper-Datenpunkten); - Verlaufskonfiguration des Datenpunkts (falls vorhanden); - Trendkonfiguration des Datenpunkts (falls vorhanden). Makros werden aufgelöst. Wenn der Datenpunkt benutzerdefinierte Intervalle verwendet, wird neben dem Aktualisierungsintervall des Datenpunkts ein Hilfesymbol angezeigt. Wenn Sie den Mauszeiger über das Symbol bewegen, wird ein Pop-up mit Details zu den benutzerdefinierten Intervallen geöffnet.

Letzte Daten	Zeigt die folgenden Informationen an: - Zeit seit der letzten Prüfung des Datenpunkts; - Letzter Wert des Datenpunkts; - Link zum einfachen Graphen/Verlauf der Datenpunktwerte (wenn Verlauf oder Trends gespeichert werden); - Konfiguriertes Sparkline-Diagramm (wenn der Datenpunkt numerisch ist). Bei Datenpunkten vom Typ Binär wird anstelle des letzten Werts ein Vorschaubild oder die Option *Anzeigen* dargestellt. Wenn Sie den Mauszeiger über das Vorschaubild bewegen, wird ein Pop-up mit dem Bild geöffnet. Wenn Sie den Mauszeiger über *Anzeigen* bewegen, wird ein Pop-up mit dem Datenpunktwert (Base64-Zeichenfolge) geöffnet. Auf einem Vorlagen-Dashboard wird anstelle des letzten Werts des Datenpunkts *Keine Daten* angezeigt.
Informationstyp	Zeigt den Informationstyp des Datenpunkts an.
Auslöser	Zeigt die folgenden Informationen an: - Beschriftung *Auslöser* mit der Anzahl der Auslöser (wenn Sie den Mauszeiger über *Auslöser* bewegen, wird ein Pop-up mit Details zu den Auslösern geöffnet); - Liste der Auslösernamen (Makros werden aufgelöst).
Host-Schnittstelle	Zeigt die vom Datenpunkt verwendete Host-Schnittstelle an oder *Keine Daten*, wenn der Datenpunkt keine Host-Schnittstelle hat (z. B. abhängige Datenpunkte, einfache Prüfungen). Auf einem Vorlagen-Dashboard wird *Keine Daten* angezeigt.
Typ	Zeigt den Typ des Datenpunkts an.
Host-Inventar	Zeigt das Host-Inventarfeld an, das durch den Datenpunktwert befüllt wird.
Tags	Zeigt die Tags des Datenpunkts an.
Host überschreiben	Wählen Sie ein kompatibles Widget oder den Host-Selektor des Dashboards als Datenquelle für Hosts aus. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.

Sparkline

Konfigurationsoptionen für das Sparkline-Diagramm sind nur verfügbar, wenn der Abschnitt *Neueste Daten* zum Widget hinzugefügt wurde.



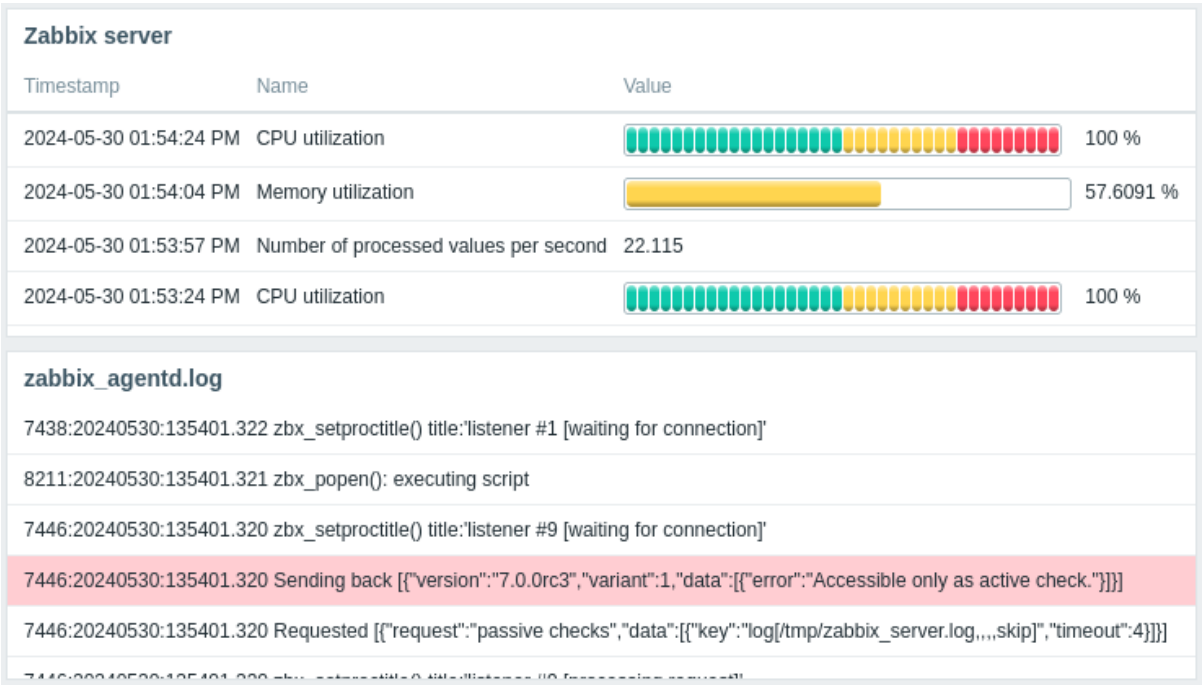
Breite	Legen Sie die Dicke der Diagrammlinie fest, indem Sie den Schieberegler verwenden oder manuell einen Wert im Bereich von 0 bis 10 eingeben.
Farbe	Wählen Sie die Linien- und Füllfarbe aus.
Füllung	Legen Sie die Transparenz der Füllfarbe fest, indem Sie den Schieberegler verwenden oder manuell einen Wert im Bereich von 0 bis 10 eingeben.

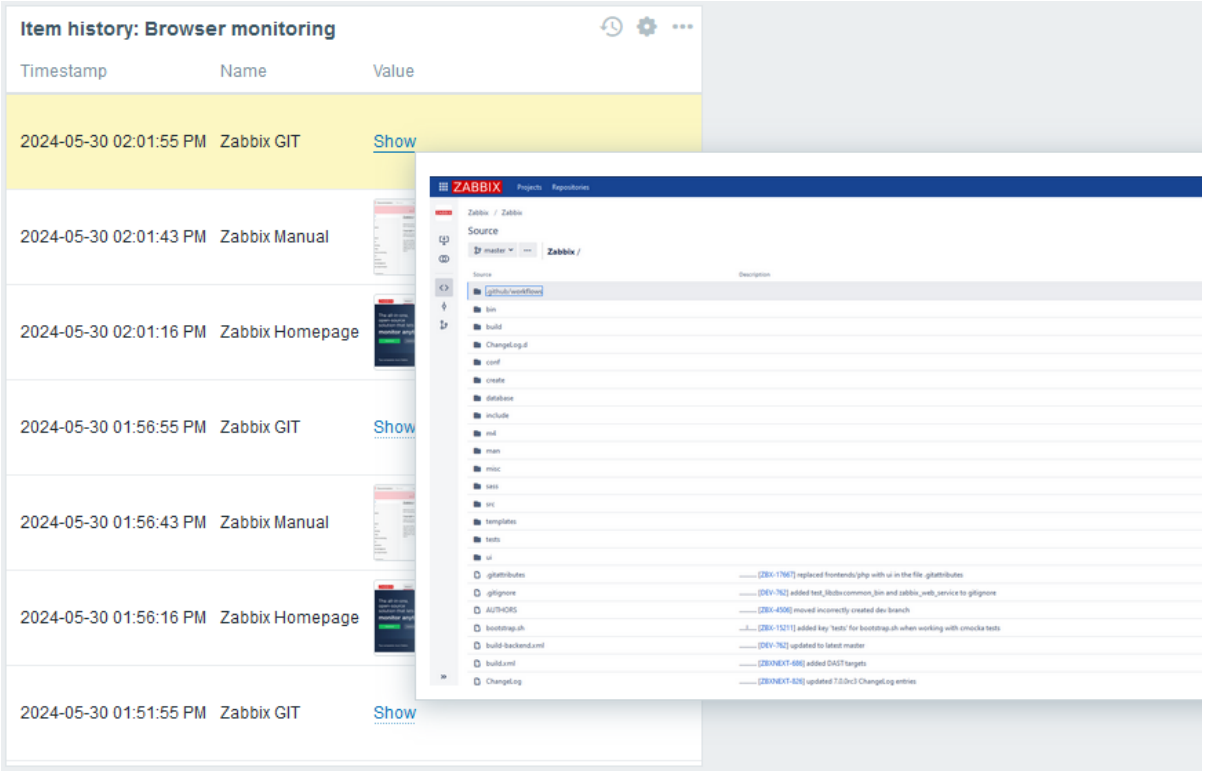
Zeitperiode	Wählen Sie die Datenquelle für die Zeitperiode aus: **Dashboard** - den Zeitperiodenwähler des Dashboards verwenden; **Widget** - ein kompatibles Widget verwenden (im Parameter *Widget* festgelegt); **Benutzerdefiniert** - eine benutzerdefinierte Zeitperiode verwenden, die in den Parametern *Von* und *Bis* festgelegt ist; wenn gesetzt, wird in der oberen rechten Ecke des Widgets ein Uhrsymbol angezeigt, das beim Überfahren mit der Maus auf die eingestellte Zeit hinweist. Beachten Sie, dass kompatible Widgets unabhängig von der Konfiguration der *Zeitperiode* des Widgets diese weiterhin als Datenquelle für die Zeitperiode verwenden können.
Widget	Geben Sie ein kompatibles Widget als Datenquelle für die Zeitperiode ein oder wählen Sie es aus. Dieser Parameter ist verfügbar, wenn *Zeitperiode* auf „Widget“ gesetzt ist.
Von	Geben Sie den Beginn der Zeitperiode ein oder wählen Sie ihn aus. Die Syntax für relative Zeitangaben (`now`, `now/d`, `now/w-1w` usw.) wird unterstützt. Dieser Parameter ist verfügbar, wenn *Zeitperiode* auf „Benutzerdefiniert“ gesetzt ist.
Bis	Geben Sie das Ende der Zeitperiode ein oder wählen Sie es aus. Die Syntax für relative Zeitangaben (`now`, `now/d`, `now/w-1w` usw.) wird unterstützt. Dieser Parameter ist verfügbar, wenn *Zeitperiode* auf „Benutzerdefiniert“ gesetzt ist.
Verlaufsdaten	Daten aus Verlauf oder Trends verwenden: **Auto** - automatische Auswahl; **Verlauf** - Verlaufsdaten verwenden; **Trends** - Trenddaten verwenden. Dieser Parameter gilt nur für numerische Daten. Nicht numerische Daten werden immer aus dem Verlauf entnommen.

Datenpunkt-Historie

Übersicht

Das Widget *Item history* zeigt die neuesten Daten für alle Datentypen (numerisch, Text usw.) in Tabellenform an. Es kann Fortschrittsbalken und Bilder anzeigen (nützlich für Browser-Items) und Werte hervorheben (nützlich für die Überwachung von Protokolldateien).

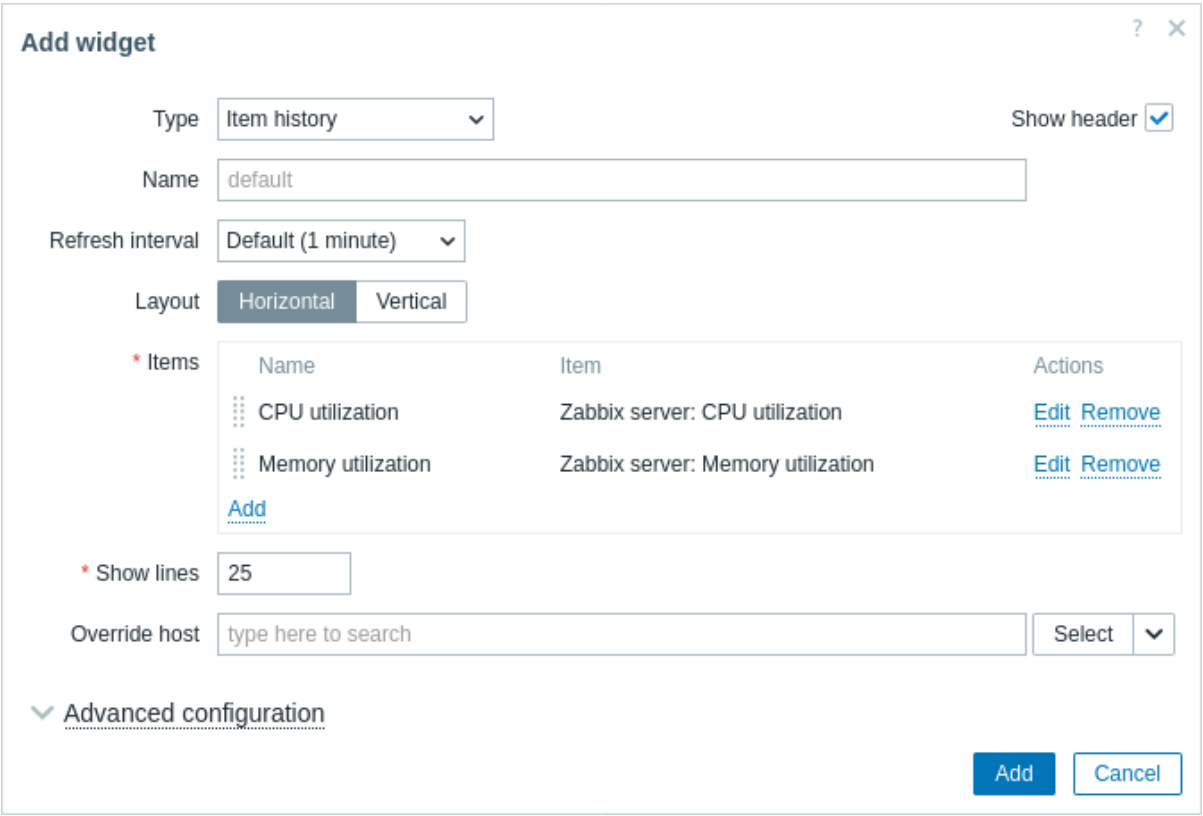




Es können bis zu 1000 Datensätze angezeigt werden.

Konfiguration

Wählen Sie zur Konfiguration *Datenpunkthistorie* als Typ aus:



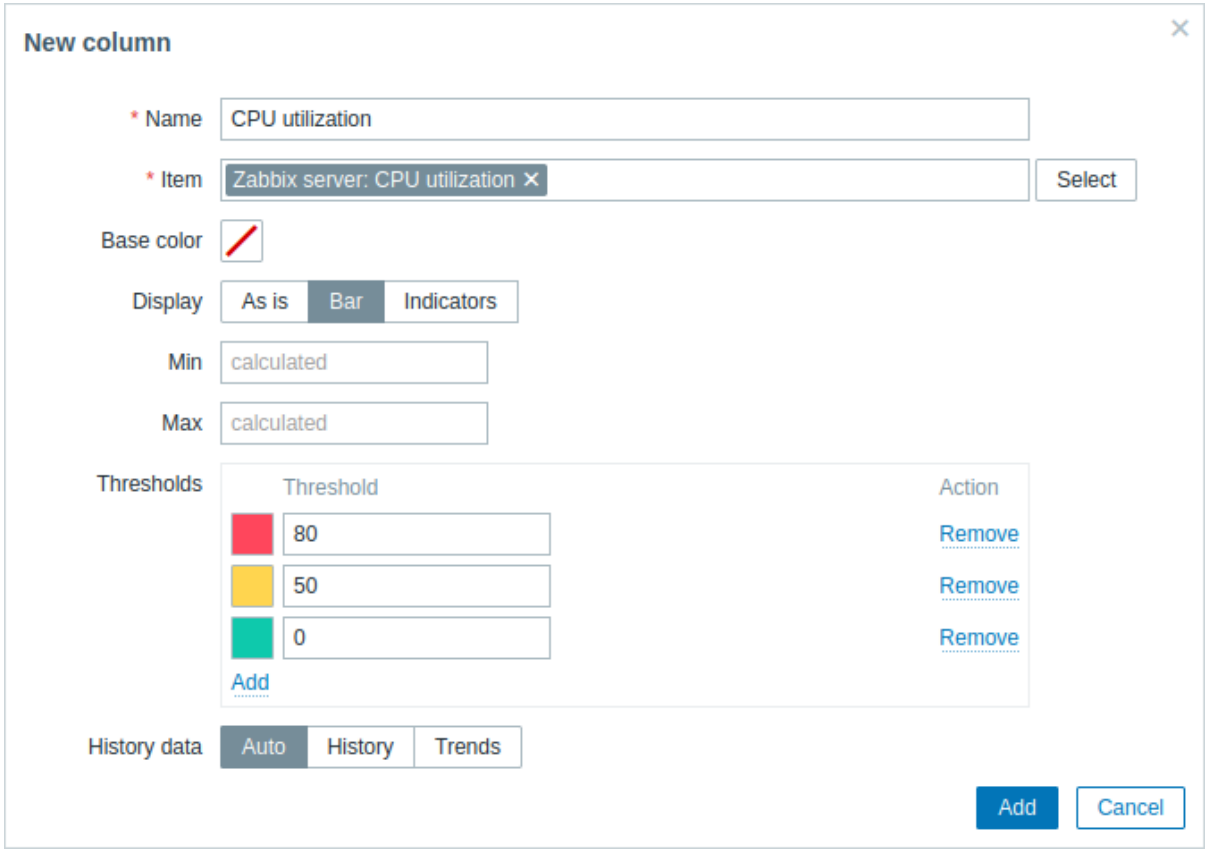
Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

Layout	Wählen Sie die Layout-Option für Datenpunktspalten aus: **Horizontal** - Datenpunkte werden horizontal angezeigt, Werte vertikal; **Vertikal** - Datenpunkte werden vertikal angezeigt, Werte horizontal.

Items	Fügen Sie Datenpunkt-Spalten zur Anzeige hinzu. Die Reihenfolge der Datenpunkte bestimmt ihre Anzeigereihenfolge. Datenpunkte können neu angeordnet werden, indem Sie sie mithilfe des Griffs vor dem Datenpunktnamen nach oben oder unten ziehen.
Show lines	Geben Sie die Anzahl der anzuzeigenden Datenpunktwert-Zeilen an.
Override host	Wählen Sie ein kompatibles Widget oder die Dashboard-Host-Auswahl als Datenquelle für Hosts aus. Dieser Parameter ist nicht verfügbar, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Advanced configuration	Klicken Sie auf die Beschriftung *Erweiterte Konfiguration*, um die Optionen der erweiterten Konfiguration anzuzeigen.

Spaltenkonfiguration

Um Datenpunkt-Spalten zu konfigurieren, klicken Sie im Parameter *Datenpunkte* auf *Hinzufügen*:



Allgemeine Spaltenparameter:

Name	Geben Sie den Namen der Spalte ein. Wenn dieses Feld leer bleibt, wird der Datenpunktname aus dem Parameter *Datenpunkt* verwendet.
Item	Wählen Sie den Datenpunkt aus. Beachten Sie, dass die Konfigurationsparameter der Spalte je nach Informationstyp des ausgewählten Datenpunkts variieren; weitere Informationen finden Sie bei den einzelnen Parametern unten. Beim Konfigurieren des Widgets auf einem Vorlagen-Dashboard können nur auf der Vorlage konfigurierte Datenpunkte ausgewählt werden.
Base color	Wählen Sie die Hintergrundfarbe der Spalte oder die Füllfarbe aus, wenn *Display* auf „Bar“ oder „Indicators“ gesetzt ist. Beachten Sie, dass die Grundfarbe durch Schwellenwert- oder Hervorhebungsfarben überschrieben werden kann.

Spaltenparameter, die spezifisch für Datenpunkte vom numerischen Typ sind:

Anzeige	Wählen Sie aus, wie der Datenpunktwert angezeigt werden soll: **Wie er ist** - als normaler Text; **Balken** - als durchgehender, farbig gefüllter Balken; **Indikatoren** - als segmentierter, farbig gefüllter Balken.
Min	Geben Sie den Mindestwert für Balken/Indikatoren ein. Wenn das Feld leer bleibt, verwendet das Widget den Mindestwert des Datenpunkts. Dieser Parameter ist nur verfügbar, wenn *Anzeige* auf „Balken“ oder „Indikatoren“ gesetzt ist.
Max	Geben Sie den Höchstwert für Balken/Indikatoren ein. Wenn das Feld leer bleibt, verwendet das Widget den Höchstwert des Datenpunkts. Dieser Parameter ist nur verfügbar, wenn *Anzeige* auf „Balken“ oder „Indikatoren“ gesetzt ist.
Schwellenwerte	Klicken Sie auf *Hinzufügen*, um einen Schwellenwert hinzuzufügen, wählen Sie eine Schwellenwertfarbe in der Farbauswahl aus und geben Sie einen numerischen Wert an. Die Liste der Schwellenwerte wird beim Speichern in aufsteigender Reihenfolge sortiert. Suffixe (zum Beispiel „1d“, „2w“, „4K“, „8G“) werden unterstützt. Wertzuordnungen werden unterstützt.
Verlaufsdaten	Wählen Sie aus, ob Daten aus der Historie oder aus Trends verwendet werden sollen: **Auto** - automatische Auswahl; **Historie** - Historiendaten verwenden; **Trends** - Trenddaten verwenden.

Spaltenparameter, die spezifisch für Datenpunkte vom Typ Zeichen, Text, Log und JSON sind:

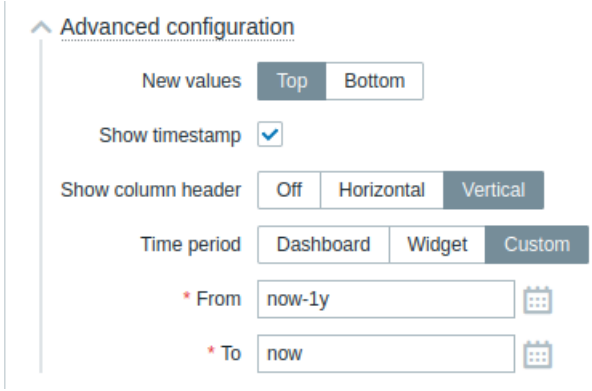
Hervorhebungen	Klicken Sie auf *Hinzufügen*, um eine Hervorhebung hinzuzufügen, wählen Sie eine Hervorhebungsfarbe aus der Farbauswahl und geben Sie einen regulären Ausdruck an. Die ausgewählte Farbe wird als Hintergrundfarbe für Datenpunktwerte verwendet, bei denen der angegebene reguläre Ausdruck mit dem Text übereinstimmt.
Anzeige	Wählen Sie aus, wie der Datenpunktwert angezeigt werden soll: **Unverändert** - genau wie empfangen. Zeilenumbrüche bleiben erhalten, wenn der Wert Zeilenumbrüche enthält. Der Zeilenumbruch wird angewendet, wenn *Layout* auf ”Vertikal” gesetzt ist und mehr als eine Spalte konfiguriert ist; **HTML** - als HTML-formatierter Text; **Einzeilig** - als einzelne Zeile, gekürzt auf eine angegebene Länge (1-500 Zeichen). Wenn Sie mit der Maus über den gekürzten Wert fahren oder darauf klicken, wird ein Popup mit dem vollständigen Wert geöffnet.
Monospace-Schriftart verwenden	Aktivieren Sie dieses Kontrollkästchen, um den Datenpunktwert in einer Monospace-Schriftart anzuzeigen (standardmäßig nicht aktiviert).
Lokale Zeit anzeigen	Aktivieren Sie dieses Kontrollkästchen, um in der Zeitstempelspalte die lokale Zeit anstelle des Zeitstempels anzuzeigen. Beachten Sie, dass auch das Kontrollkästchen *Zeitstempel anzeigen* in der erweiterten Konfiguration aktiviert sein muss. Dieser Parameter ist nur für Datenpunkte vom Typ Log verfügbar.

Spaltenparameter speziell für Datenpunkte vom Binärtyp:

Miniaturansicht anzeigen	Aktivieren Sie dieses Kontrollkästchen, um für Bild-Binärdateien eine Miniaturansicht oder für Nicht-Bild-Binärdateien die Option „Anzeigen“ darzustellen. Deaktivieren Sie dieses Kontrollkästchen, um für alle Binär-Datenpunktwerte die Option „Anzeigen“ darzustellen. Wenn Sie mit der Maus über die Option „Anzeigen“ fahren oder darauf klicken, wird ein Pop-up-Fenster mit dem Datenpunktwert (Bild oder Base64-Zeichenfolge) geöffnet. Wenn der Datenpunktwert eine leere Zeichenfolge ist, wird die Option „Anzeigen“ dargestellt; wenn Sie mit der Maus darüber fahren oder darauf klicken, wird ein Pop-up mit „Leere Zeichenfolge“ geöffnet.

Erweiterte Konfiguration

Erweiterte Konfigurationsoptionen sind im einklappbaren Abschnitt *Erweiterte Konfiguration* verfügbar:

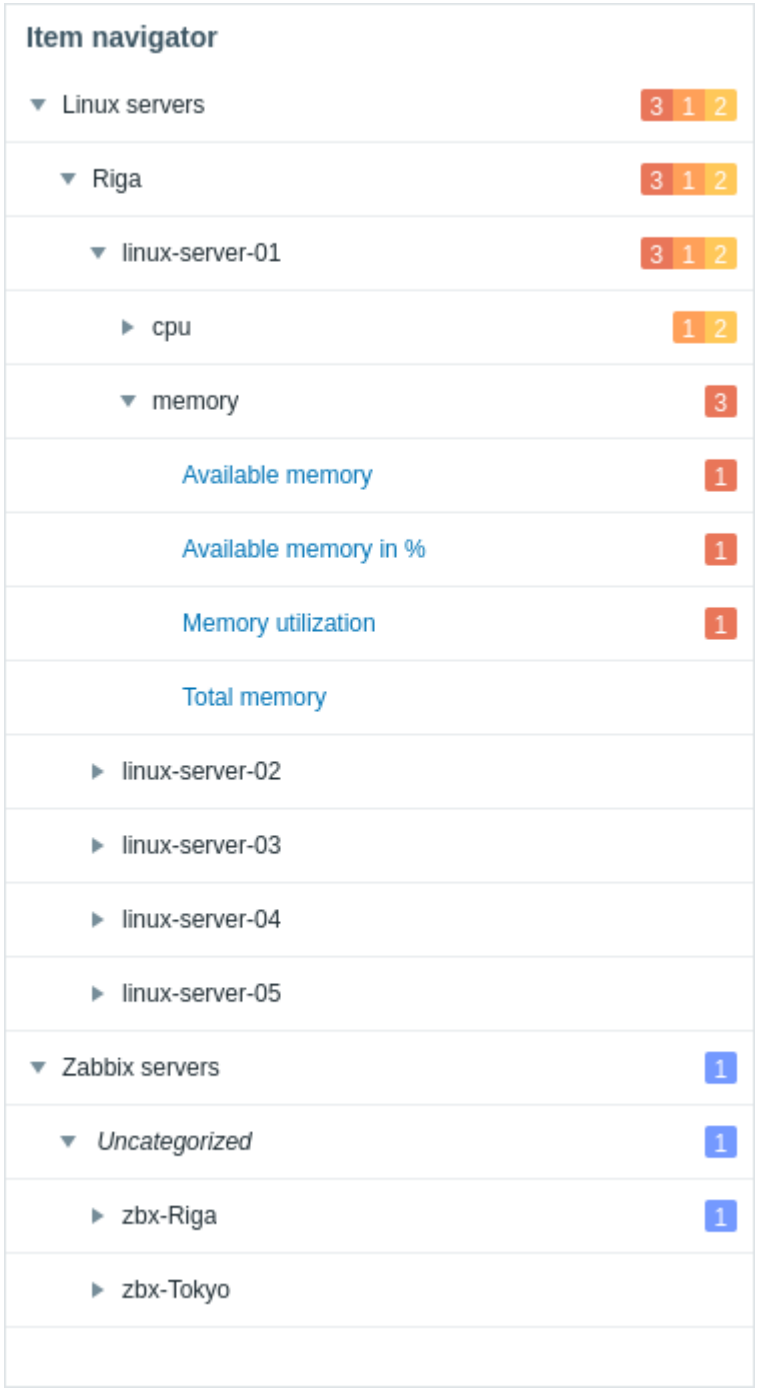


Neue Werte	Wählen Sie aus, wo neue Datenpunktwerte hinzugefügt werden sollen: **Oben** - am Anfang der Spalten; **Unten** - am Ende der Spalten.
Zeitstempel anzeigen	Aktivieren Sie dieses Kontrollkästchen, um die Zeitstempelspalte anzuzeigen (standardmäßig nicht aktiviert).
Spaltenüberschrift anzeigen	Wählen Sie die Ausrichtung der Spaltenüberschrift: **Aus** - die Überschrift ausblenden; **Horizontal** - die Überschrift horizontal anzeigen; **Vertikal** - die Überschrift vertikal anzeigen.
Zeitraum	Wählen Sie die Datenquelle für den Zeitraum aus: **Dashboard** - den Zeitraumauswahl des Dashboards verwenden; **Widget** - ein kompatibles Widget verwenden (im Parameter *Widget* festgelegt); **Benutzerdefiniert** - einen benutzerdefinierten Zeitraum verwenden, der in den Parametern *Von* und *Bis* festgelegt ist; wenn gesetzt, wird in der oberen rechten Ecke des Widgets ein Uhrsymbol angezeigt, das beim Überfahren mit der Maus auf die eingestellte Zeit hinweist. Beachten Sie, dass kompatible Widgets unabhängig von der *Zeitraum*-Konfiguration des Widgets weiterhin als Datenquelle für den Zeitraum verwendet werden können.
Widget	Geben Sie ein kompatibles Widget (*Graph*, *Graph (classic)*, *Graph prototype*) als Datenquelle für den Zeitraum ein oder wählen Sie es aus. Dieser Parameter ist verfügbar, wenn *Zeitraum* auf "Widget" gesetzt ist.
Von	Geben Sie den Beginn des Zeitraums ein oder wählen Sie ihn aus. Die Syntax für relative Zeitangaben (`now`, `now/d`, `now/w-1w` usw.) wird unterstützt. Dieser Parameter ist verfügbar, wenn *Zeitraum* auf "Benutzerdefiniert" gesetzt ist.
Bis	Geben Sie das Ende des Zeitraums ein oder wählen Sie es aus. Die Syntax für relative Zeitangaben (`now`, `now/d`, `now/w-1w` usw.) wird unterstützt. Dieser Parameter ist verfügbar, wenn *Zeitraum* auf "Benutzerdefiniert" gesetzt ist.

Datenpunkt-Navigator

Übersicht

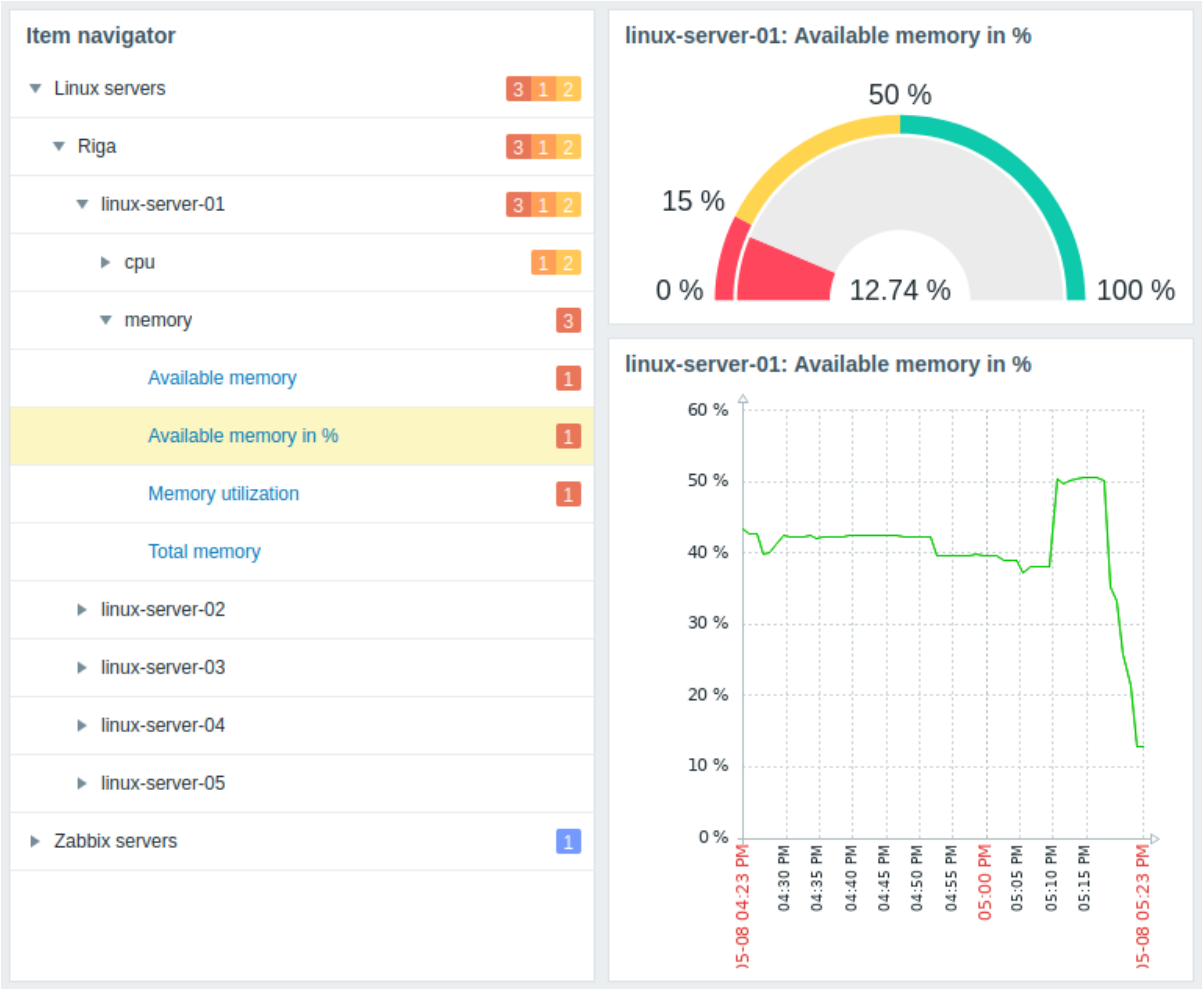
Das Widget *Datenpunkt-Navigator* zeigt eine Liste von Datenpunkten basierend auf verschiedenen Filter- und Gruppierungsoptionen an.



Gruppen, nach denen Datenpunkte organisiert sind, können ein- oder ausgeklappt werden.

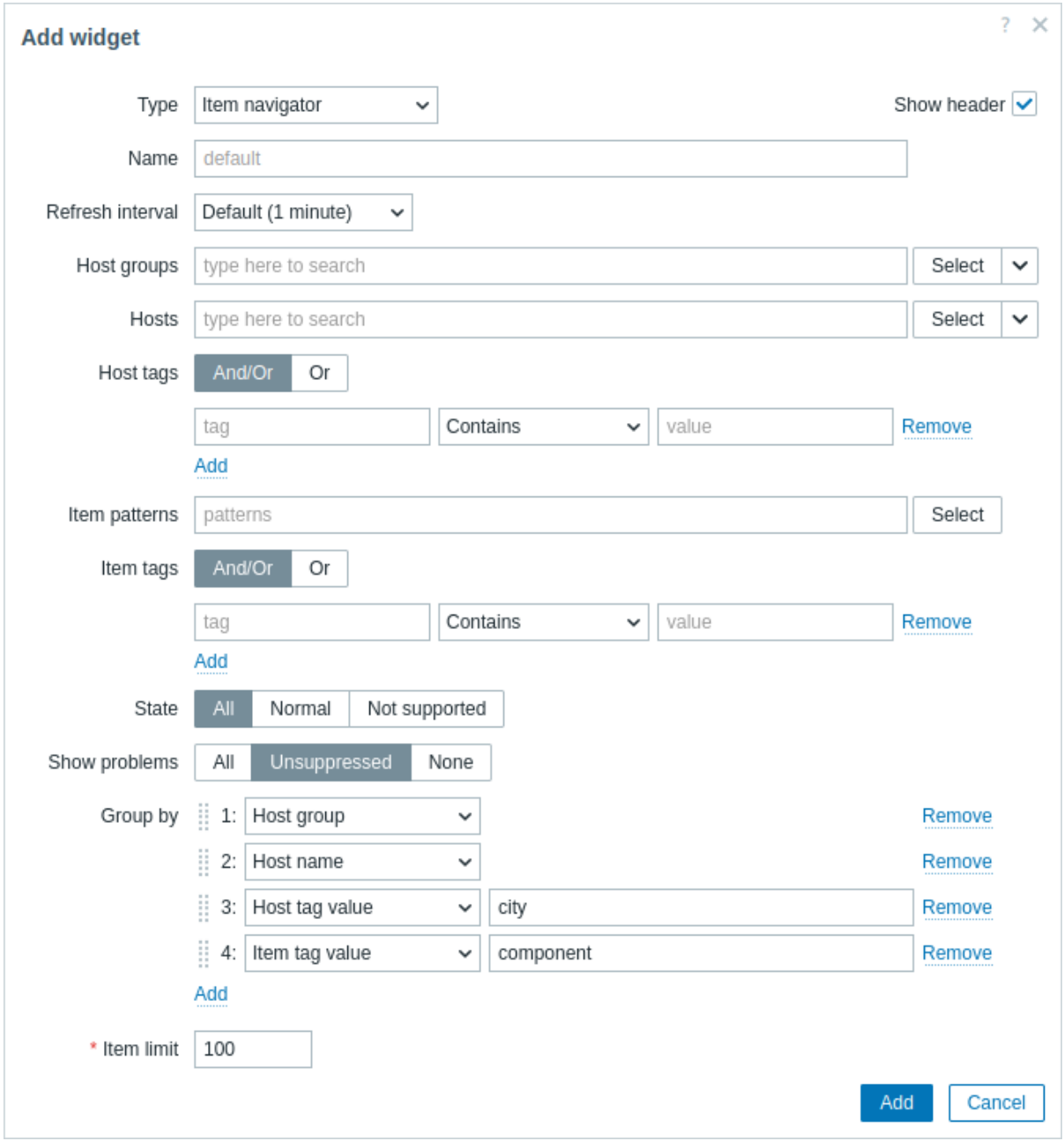
Für Gruppen und Probleme sind zusätzliche Details über Mouseover-Hinweise zugänglich.

Das Widget ist besonders nützlich, um zu steuern, was andere Widgets basierend auf dem ausgewählten Datenpunkt anzeigen.



Konfiguration

Wählen Sie zur Konfiguration *Item navigator* als Typ aus:



Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

Host groups	Wählen Sie Hostgruppen aus. Alternativ können Sie ein kompatibles Widget als Datenquelle für Hostgruppen auswählen. Dieses Feld unterstützt die automatische Vervollständigung. Wenn Sie beginnen, den Namen einer Gruppe einzugeben, wird eine Dropdown-Liste mit passenden Gruppen angezeigt. Durch die Auswahl einer übergeordneten Hostgruppe werden implizit alle verschachtelten Hostgruppen ausgewählt. Wenn keine Hostgruppen ausgewählt sind, zeigt das Widget Datenpunkte an, die zu allen Hosts aus allen Hostgruppen gehören. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Hosts	Wählen Sie Hosts aus. Alternativ können Sie ein kompatibles Widget oder das Dashboard als Datenquelle für Hosts auswählen. Dieses Feld unterstützt die automatische Vervollständigung. Wenn Sie beginnen, den Namen eines Hosts einzugeben, wird eine Dropdown-Liste mit passenden Hosts angezeigt. Wenn keine Hosts ausgewählt sind, zeigt das Widget Datenpunkte an, die zu allen Hosts gehören. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.

Host tags	Geben Sie Host-Tags an, nach denen die im Widget angezeigten Datenpunkte gefiltert werden sollen. Sie können bestimmte Tags und Tag-Werte sowohl einschließen als auch ausschließen. Es können mehrere Bedingungen festgelegt werden. Beim Abgleich von Tag-Namen wird immer zwischen Groß- und Kleinschreibung unterschieden. Für jede Bedingung stehen mehrere Operatoren zur Verfügung: **Exists** - die angegebenen Tag-Namen einschließen; **Equals** - die angegebenen Tag-Namen und -Werte einschließen (Groß-/Kleinschreibung wird berücksichtigt); **Contains** - die angegebenen Tag-Namen einschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, ohne Berücksichtigung der Groß-/Kleinschreibung); **Does not exist** - die angegebenen Tag-Namen ausschließen; **Does not equal** - die angegebenen Tag-Namen und -Werte ausschließen (Groß-/Kleinschreibung wird berücksichtigt); **Does not contain** - die angegebenen Tag-Namen ausschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, ohne Berücksichtigung der Groß-/Kleinschreibung). Für Bedingungen gibt es zwei Berechnungstypen: **And/Or** - alle Bedingungen müssen erfüllt sein; Bedingungen mit demselben Tag-Namen werden mit der Or-Bedingung gruppiert; **Or** - es reicht aus, wenn eine Bedingung erfüllt ist. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Item patterns	Geben Sie Datenpunktmuster ein oder wählen Sie vorhandene Datenpunkte als Datenpunktmuster aus. Datenpunkte, die den angegebenen Mustern entsprechen, werden im Item navigator angezeigt. Für die Auswahl können Platzhaltermuster verwendet werden (z. B. gibt ∗ Datenpunkte zurück, die null oder mehr Zeichen entsprechen; `Zabbix*` gibt Datenpunkte zurück, die mit „Zabbix“ beginnen). Um ein Platzhaltermuster anzugeben, geben Sie die Zeichenfolge manuell ein und drücken Sie *Enter*. Während der Eingabe wird eine Dropdown-Liste mit passenden Datenpunkten angezeigt, die auf Datenpunkte beschränkt ist, die zu den ausgewählten *Hosts* oder zu Hosts innerhalb der ausgewählten *Host groups* gehören, sofern vorhanden. Das Platzhaltersymbol wird immer interpretiert. Daher ist es beispielsweise nicht möglich, einen Datenpunkt namens *item** einzeln hinzuzufügen, wenn andere passende Datenpunkte vorhanden sind (z. B. item2, item3). Bei der Konfiguration des Widgets auf einem Vorlagen-Dashboard können mit diesem Parameter nur auf der Vorlage konfigurierte Datenpunkte ausgewählt werden.

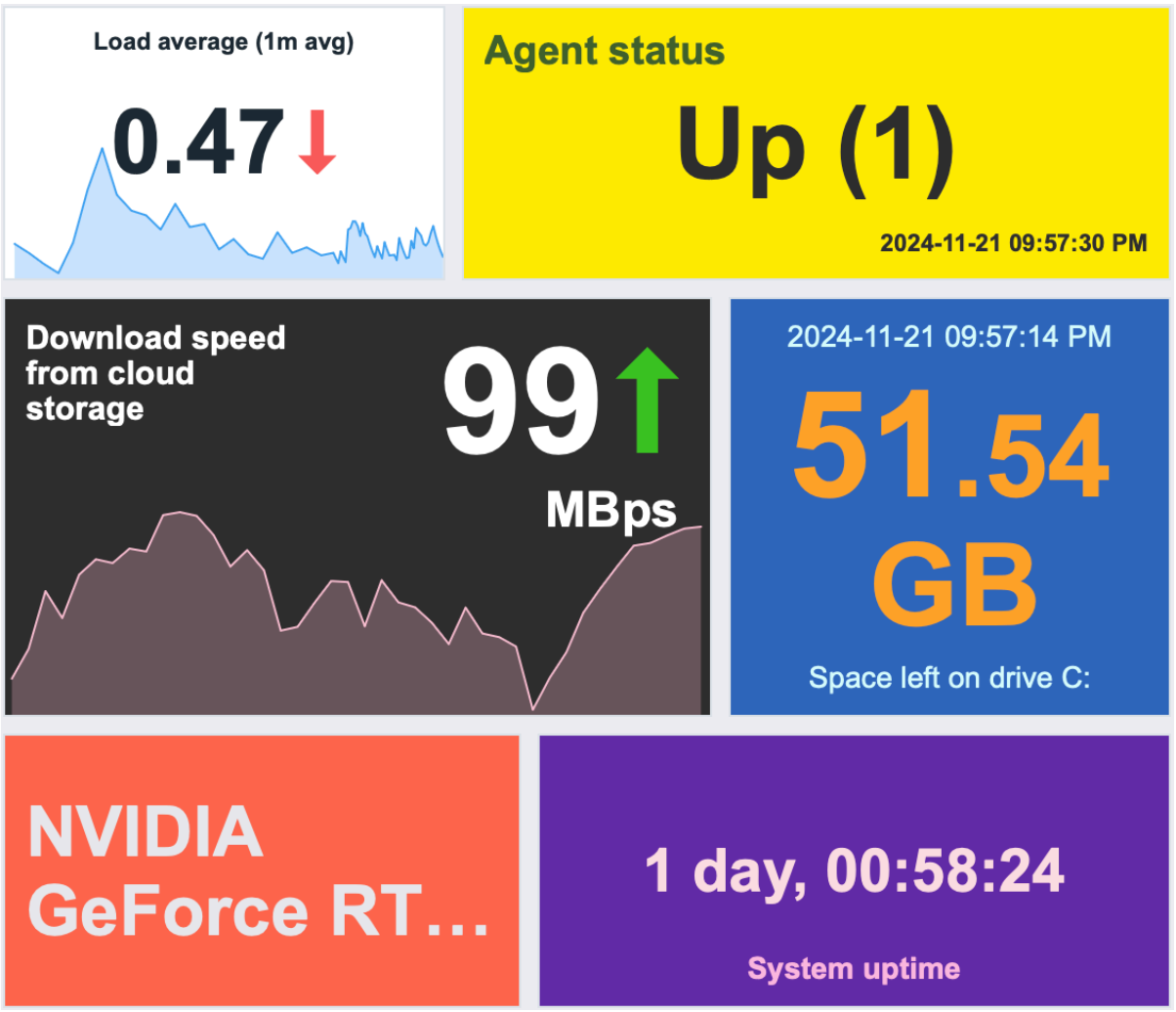
Item tags	Geben Sie Datenpunkt-Tags an, nach denen die im Widget angezeigten Datenpunkte gefiltert werden sollen. Sie können bestimmte Tags und Tag-Werte sowohl einschließen als auch ausschließen. Es können mehrere Bedingungen festgelegt werden. Beim Abgleich von Tag-Namen wird immer zwischen Groß- und Kleinschreibung unterschieden. Für jede Bedingung stehen mehrere Operatoren zur Verfügung: **Exists** - die angegebenen Tag-Namen einschließen; **Equals** - die angegebenen Tag-Namen und -Werte einschließen (Groß-/Kleinschreibung wird berücksichtigt); **Contains** - die angegebenen Tag-Namen einschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, ohne Berücksichtigung der Groß-/Kleinschreibung); **Does not exist** - die angegebenen Tag-Namen ausschließen; **Does not equal** - die angegebenen Tag-Namen und -Werte ausschließen (Groß-/Kleinschreibung wird berücksichtigt); **Does not contain** - die angegebenen Tag-Namen ausschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, ohne Berücksichtigung der Groß-/Kleinschreibung). Für Bedingungen gibt es zwei Berechnungstypen: **And/Or** - alle Bedingungen müssen erfüllt sein; Bedingungen mit demselben Tag-Namen werden mit der Or-Bedingung gruppiert; **Or** - es reicht aus, wenn eine Bedingung erfüllt ist.
State	Filtern Sie, welche Datenpunkte anhand ihres Status angezeigt werden sollen (alle, normal, nicht unterstützt).
Show problems	Filtern Sie, welche Probleme anhand ihres Status angezeigt werden sollen (alle, nicht unterdrückte, keine). Probleme werden am Ende der Datenpunktzeile durch farbige Blöcke angezeigt. Die Farben basieren auf dem Schweregrad des Problems, der unter *Administration > General > Trigger displaying options* angepasst werden kann. Problem-Schweregrade können beim Aktualisieren von Problemen geändert werden.
Group by	Fügen Sie ein Gruppierungsattribut hinzu, nach dem Datenpunkte gruppiert werden sollen: **Host group** - Datenpunkte nach den Hostgruppen ihrer Hosts gruppieren; **Host name** - Datenpunkte nach ihren Hosts gruppieren; **Host tag value** - geben Sie einen Tag-Namen ein, um Datenpunkte nach den Werten dieses Host-Tags zu gruppieren (geben Sie beispielsweise „city“ ein, um Datenpunkte nach den Werten „Riga“, „Tokyo“ usw. zu gruppieren); **Item tag value** - geben Sie einen Tag-Namen ein, um Datenpunkte nach den Werten dieses Datenpunkt-Tags zu gruppieren (geben Sie beispielsweise „component“ ein, um Datenpunkte nach den Werten „cpu“, „memory“ usw. zu gruppieren). Die Reihenfolge der Gruppierungsattribute kann durch Ziehen am Griff vor dem Gruppennamen nach oben oder unten geändert werden. Beachten Sie, dass die Reihenfolge der Gruppierungsattribute die Verschachtelungsreihenfolge der Gruppen bestimmt. Wenn Sie beispielsweise mehrere Host-Tag-Namen angeben (1: color, 2: city), werden Datenpunkte zuerst nach color (red, blue usw.) und anschließend nach city (Riga, Tokyo usw.) gruppiert. Ein Datenpunkt kann abhängig von den konfigurierten Gruppierungsattributen in mehreren Gruppen angezeigt werden (beispielsweise bei einer Gruppierung nach Hostgruppe, wenn der Host des Datenpunkts zu mehreren Hostgruppen gehört). Durch Klicken auf solche Datenpunkte werden sie in allen Gruppen ausgewählt und hervorgehoben. Datenpunkte, die den konfigurierten Gruppierungsattributen nicht entsprechen, werden in der Gruppe *Uncategorized* angezeigt. Wenn *Show problems* für die Anzeige von Problemen konfiguriert ist, werden diese wie folgt angezeigt: - für jede Gruppe wird die Anzahl aller Probleme der untergeordneten Datenpunkte angezeigt; - für jeden Datenpunkt wird nur dessen Problemanzahl angezeigt. Es können bis zu 10 Gruppierungsattribute angegeben werden, und alle müssen eindeutig sein. Wenn keine Gruppierungsattribute angegeben sind, werden die Datenpunkte nicht gruppiert.

Item limit	Geben Sie die maximale Anzahl der anzuzeigenden Datenpunkte ein. Zulässige Werte liegen zwischen 1 und 9999. Wenn mehr Datenpunkte zur Anzeige verfügbar sind als durch das Limit festgelegt, wird unter den angezeigten Datenpunkten eine entsprechende Meldung angezeigt (z. B. „100 of 100+ items are shown“). Beachten Sie, dass das konfigurierte Datenpunktlimit auch die Anzeige der konfigurierten Gruppen beeinflusst. Wenn das Datenpunktlimit beispielsweise auf 100 gesetzt ist und Datenpunkte nach ihren Hosts gruppiert werden (wobei jeder Host 200 Datenpunkte enthält), wird im Widget nur der erste Host mit seinen 100 Datenpunkten angezeigt. Dieser Parameter wird durch den Parameter *Limit for search and filter results* unter *Administration* > *General* > *GUI* nicht beeinflusst.

Datenpunktwert

Übersicht

Das Widget *Item value* zeigt den Wert eines einzelnen numerischen oder Zeichenfolgen-Datenpunkts an. Es ist nützlich, um wichtige Metriken im Blick zu behalten, Schwellenwerte zu visualisieren und plötzliche Änderungen in den Daten zu erkennen.



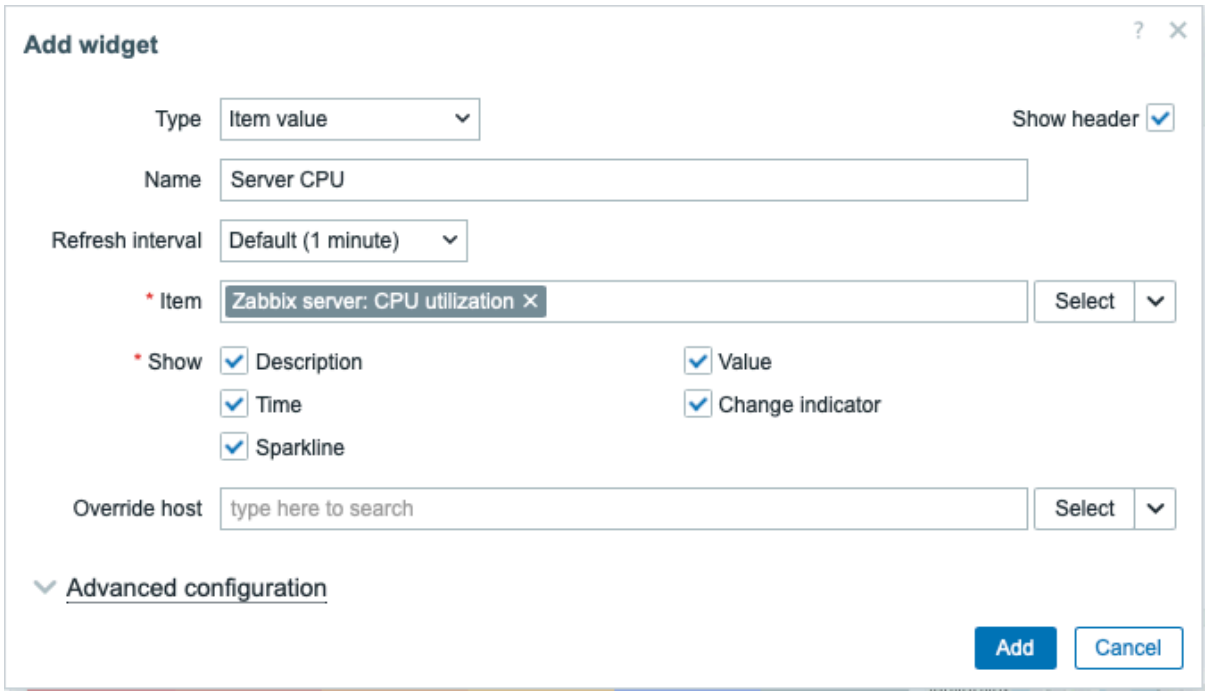
Sie können das Widget so konfigurieren, dass es Folgendes anzeigt:

- Zeitstempel der Metrik (2024-11-21 09:57:30)
- Beschreibung des Datenpunkts (Load average (1m), Agent status)
- Wert des Datenpunkts, Änderungsindikator und Einheiten (0.47 ↓, 99 ↑ Mbps)
- Hintergrundfarbe
- Sparkline-Diagramm für Werte aus dem angegebenen Zeitraum

Durch Klicken auf das Widget wird ein Ad-hoc-Diagramm für numerische Datenpunkte oder die neuesten Daten für Zeichenfolgen-Datenpunkte geöffnet.

Konfiguration

Um zu konfigurieren, wählen Sie *Datenpunkt-Wert* als Widget-Typ aus:



Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

Datenpunkt	Wählen Sie den Datenpunkt aus. Alternativ können Sie ein kompatibles Widget als Datenquelle für den Datenpunkt auswählen. Datenpunkte, die Binärdaten zurückgeben, werden nicht unterstützt.
Anzeigen	Aktivieren Sie das Kontrollkästchen, um das jeweilige Element anzuzeigen: Beschreibung, Wert, Zeit, Änderungsindikator, Sparkline. Deaktivieren Sie es, um das Element auszublenden. Mindestens ein Element muss ausgewählt sein.
Host überschreiben	Wählen Sie ein kompatibles Widget oder den Dashboard-Host-Selektor als Datenquelle für Hosts aus. Dieser Parameter ist nicht verfügbar, wenn Sie das Widget in einem Vorlagen-Dashboard konfigurieren.
Erweiterte Konfiguration	Klicken Sie auf die Überschrift *Erweiterte Konfiguration*, um die Optionen der erweiterten Konfiguration anzuzeigen.

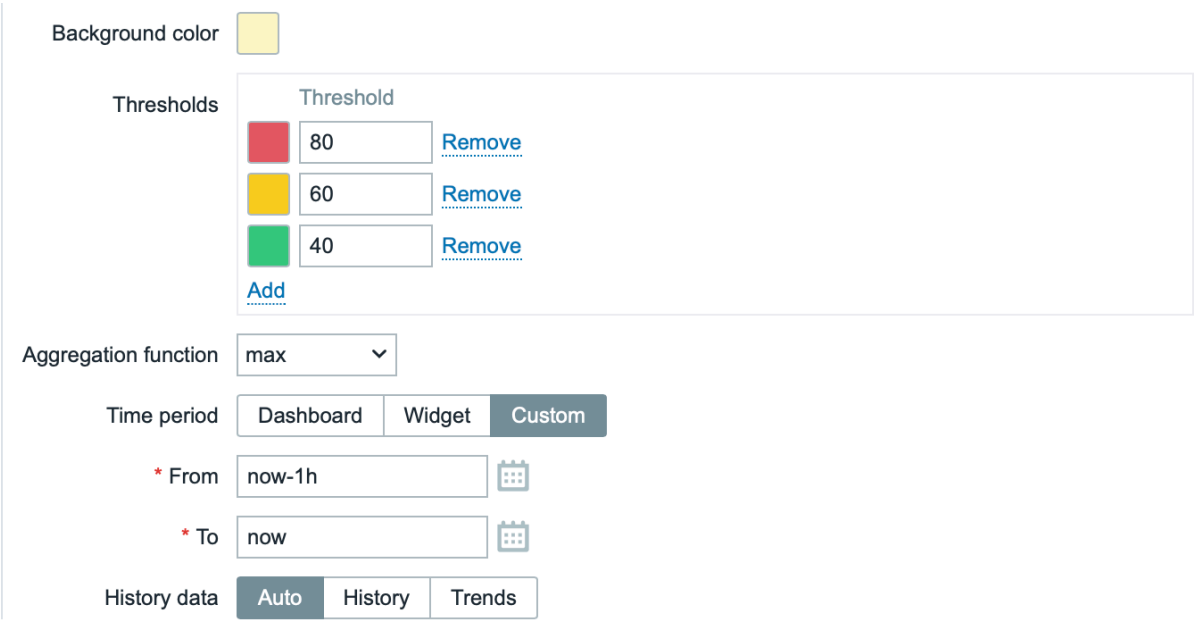
Erweiterte Konfiguration

Erweiterte Konfigurationsoptionen sind im einklappbaren Abschnitt *Erweiterte Konfiguration* verfügbar und nur für die Elemente, die im Feld *Anzeigen* ausgewählt sind (siehe oben).

Beachten Sie, dass mehrere Elemente nicht denselben Platz einnehmen können; wenn sie im selben Bereich platziert werden, wird eine Fehlermeldung angezeigt.

Allgemeine Widget-Parameter

Diese Parameter bestimmen die Hintergrundfarbe (statisch oder dynamisch) für das gesamte Widget sowie eine Aggregationsfunktion zur Anzeige von Werten.

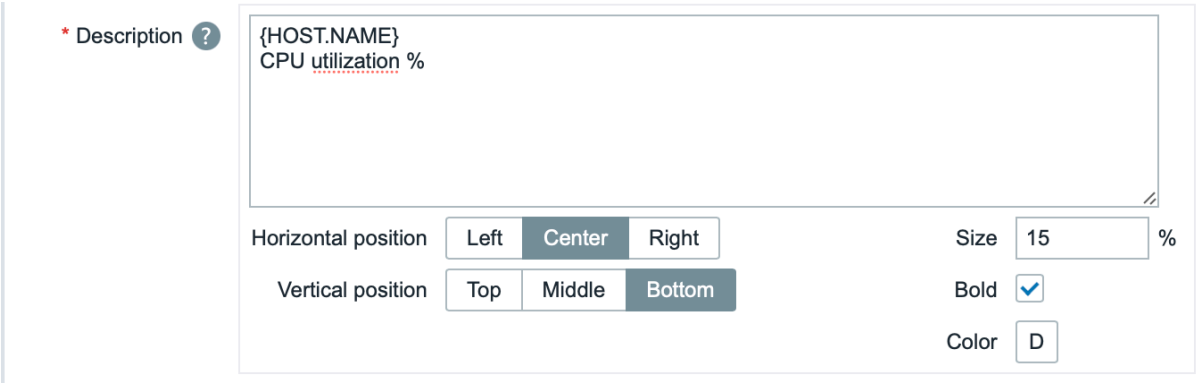


Hintergrundfarbe	Wählen Sie die Hintergrundfarbe für das gesamte Widget aus der Farbpalette aus. D steht für die Standardfarbe (abhängig vom Frontend-Theme). Um den Standardwert wiederherzustellen, klicken Sie in der Farbpalette auf die Schaltfläche *Standard verwenden*.
Schwellenwerte	Konfigurieren Sie die dynamische Hintergrundfarbe für das gesamte Widget. Klicken Sie auf *Hinzufügen*, um einen Schwellenwert hinzuzufügen, wählen Sie die Hintergrundfarbe aus der Farbpalette aus und geben Sie einen numerischen Wert an. Sobald der Wert des Datenpunkts dem Schwellenwert entspricht oder größer ist, ändert sich die Hintergrundfarbe. Die Liste wird beim Speichern in aufsteigender Reihenfolge sortiert. Beachten Sie, dass die dynamische Hintergrundfarbe nur für numerische Datenpunkte korrekt angezeigt wird.
Aggregationsfunktion	Geben Sie an, welche Aggregationsfunktion innerhalb des ausgewählten *Zeitraums* verwendet werden soll: **min** - den kleinsten Wert anzeigen; **max** - den größten Wert anzeigen; **avg** - den Durchschnittswert anzeigen; **count** - die Anzahl der Werte anzeigen; **sum** - die Summe der Werte anzeigen; **first** - den ersten Wert anzeigen; **last** - den letzten Wert anzeigen. Wenn **nicht verwendet** ausgewählt ist, wird der aktuellste Wert angezeigt (keine Aggregation). Durch die Aggregation kann für das angegebene Intervall (5 Minuten, eine Stunde, ein Tag) ein aggregierter Wert anstelle des aktuellsten Werts angezeigt werden. Für *min*, *max*, *avg* und *sum* können nur numerische Daten angezeigt werden. Bei *count* werden nicht numerische Daten in numerische Daten umgewandelt.
Zeitraum	Zeitraum, der zur Aggregation von Werten verwendet werden soll. Wählen Sie die Datenquelle für den Zeitraum aus: **Dashboard** - den Zeitraumauswahlschalter des Dashboards verwenden; **Widget** - ein kompatibles Widget verwenden (im Parameter *Widget* festgelegt); **Benutzerdefiniert** - einen benutzerdefinierten Zeitraum verwenden, der in den Parametern *Von* und *Bis* festgelegt ist; wenn dieser festgelegt ist, wird in der oberen rechten Ecke des Widgets ein Uhrsymbol angezeigt, das beim Überfahren mit dem Mauszeiger die festgelegte Zeit angibt. Beachten Sie, dass kompatible Widgets unabhängig von der Konfiguration des Parameters *Zeitraum* des Widgets weiterhin als Datenquelle für den Zeitraum verwendet werden können. Dieser Parameter ist ausgeblendet, wenn für *Aggregationsfunktion* „nicht verwendet“ festgelegt ist.
Widget	Geben Sie ein kompatibles Widget als Datenquelle für den Zeitraum ein oder wählen Sie es aus. Dieser Parameter ist verfügbar, wenn für *Zeitraum* „Widget“ festgelegt ist.
Von	Geben Sie den Beginn des Zeitraums ein oder wählen Sie ihn aus. Die Syntax für relative Zeitangaben (`now`, `now/d`, `now/w-1w` usw.) wird unterstützt. Dieser Parameter ist verfügbar, wenn für *Zeitraum* „Benutzerdefiniert“ festgelegt ist.

Bis	Geben Sie das Ende des Zeitraums ein oder wählen Sie es aus. Die Syntax für relative Zeitangaben (`now`, `now/d`, `now/w-1w` usw.) wird unterstützt. Dieser Parameter ist verfügbar, wenn für *Zeitraum* „Benutzerdefiniert“ festgelegt ist.
Verlaufsdaten	Daten aus dem Verlauf oder aus Trends verwenden: **Auto** - automatische Auswahl; **Verlauf** - Verlaufsdaten verwenden; **Trends** - Trenddaten verwenden. Diese Einstellung gilt nur für numerische Daten. Nicht numerische Daten werden immer aus dem Verlauf übernommen.

Beschreibung

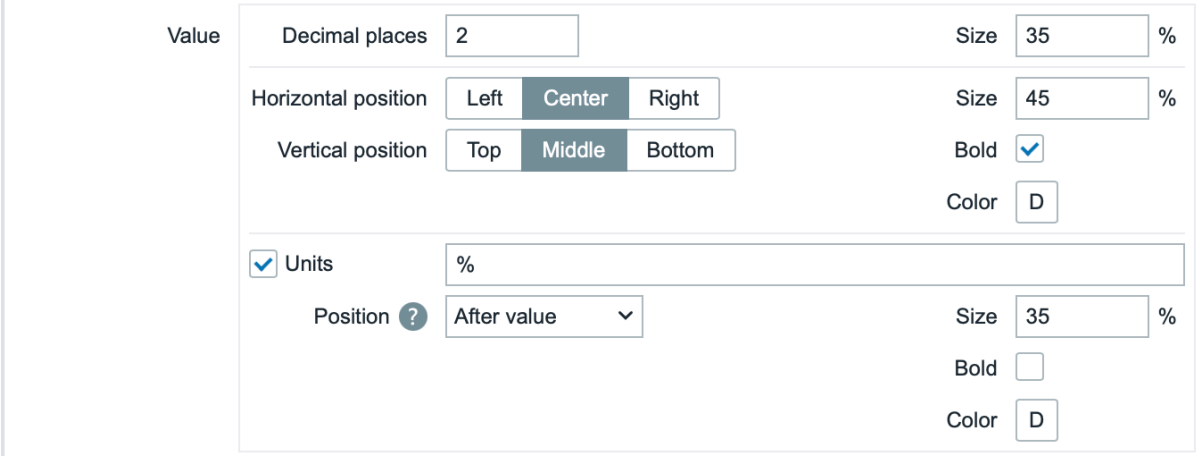
Diese Parameter bestimmen, wie die Beschreibung des Datenpunkts angezeigt werden soll.



Beschreibung	Geben Sie die Beschreibung des Datenpunkts ein. Diese Beschreibung kann den Standardnamen des Datenpunkts überschreiben. Mehrzeilige Beschreibungen werden unterstützt. Eine Kombination aus Text und unterstützten Makros ist möglich. {HOST.*}, {ITEM.*}, {INVENTORY.*} und Benutzermakros werden unterstützt.
Horizontale Position	Wählen Sie die horizontale Position der Beschreibung des Datenpunkts – links, rechts oder zentriert.
Vertikale Position	Wählen Sie die vertikale Position der Beschreibung des Datenpunkts – oben, unten oder mittig.
Größe	Geben Sie die Schriftgröße für die Beschreibung des Datenpunkts ein (in Prozent relativ zur Gesamthöhe des Widgets).
Fett	Aktivieren Sie das Kontrollkästchen, um die Beschreibung des Datenpunkts fett darzustellen.
Farbe	Wählen Sie die Farbe der Beschreibung des Datenpunkts im Farbwähler aus. D steht für die Standardfarbe (abhängig vom Frontend-Theme). Um zum Standardwert zurückzukehren, klicken Sie im Farbwähler auf die Schaltfläche *Standard verwenden*.

Wert

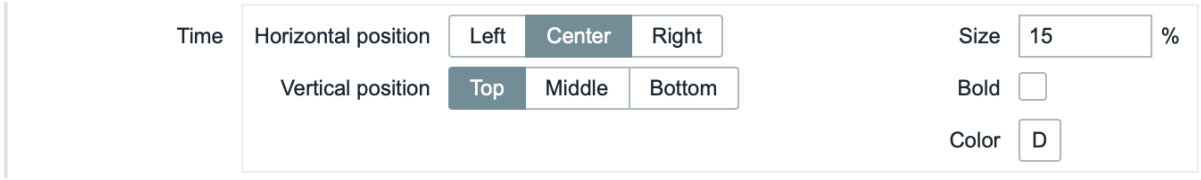
Diese Parameter bestimmen, wie der Datenpunktwert angezeigt werden soll.



Dezimalstellen	Wählen Sie aus, wie viele Dezimalstellen mit dem Wert angezeigt werden sollen. Dieser Wert wirkt sich nur auf Float-Datenpunkte aus. Bei Datenpunkten mit der Einheit „s“ gilt: Wenn dieser Wert größer als 0 ist (Standardwert ist 2), rundet das Widget die höchstwertige Zeiteinheit und zeigt einen numerischen Wert an (z. B. „10.43m“), anstatt den Wert in eine vollständige Zeitzeichenfolge umzuwandeln. Wenn der Wert auf 0 gesetzt ist, werden die Sekunden in eine menschenlesbare Zeitzeichenfolge umgewandelt (zum Beispiel „4h 56m 30s“).
Größe	Geben Sie die Schriftgröße für die Dezimalstellen ein (in Prozent relativ zur Gesamthöhe des Widgets).
Horizontale Position	Wählen Sie die horizontale Position des Datenpunktwerts – links, rechts oder zentriert.
Vertikale Position	Wählen Sie die vertikale Position des Datenpunktwerts – oben, unten oder mittig.
Größe	Geben Sie die Schriftgröße für den Datenpunktwert ein (in Prozent relativ zur Gesamthöhe des Widgets). Beachten Sie, dass die Größe des Datenpunktwerts priorisiert wird; andere Elemente müssen dem Wert Platz einräumen. Beim Änderungsindikator wird der Wert jedoch abgeschnitten, wenn er zu groß ist, damit der Änderungsindikator angezeigt werden kann.
Fett	Aktivieren Sie das Kontrollkästchen, um den Datenpunktwert fett darzustellen.
Farbe	Wählen Sie die Farbe des Datenpunktwerts im Farbwähler aus. D steht für die Standardfarbe (abhängig vom Frontend-Theme). Um zum Standardwert zurückzukehren, klicken Sie im Farbwähler auf die Schaltfläche *Standard verwenden*.
Einheiten	
Einheiten	Aktivieren Sie das Kontrollkästchen, um Einheiten zusammen mit dem Datenpunktwert anzuzeigen. Wenn Sie einen Einheitsnamen eingeben, überschreibt dieser die Einheit aus der Datenpunktkonfiguration.
Position	Wählen Sie die Position der Datenpunkteinheit – oberhalb, unterhalb, vor oder nach dem Wert.
Größe	Geben Sie die Schriftgröße für die Datenpunkteinheit ein (in Prozent relativ zur Gesamthöhe des Widgets).
Fett	Aktivieren Sie das Kontrollkästchen, um die Datenpunkteinheit fett darzustellen.
Farbe	Wählen Sie die Farbe der Datenpunkteinheit im Farbwähler aus. D steht für die Standardfarbe (abhängig vom Frontend-Theme). Um zum Standardwert zurückzukehren, klicken Sie im Farbwähler auf die Schaltfläche *Standard verwenden*.

Zeit

Diese Parameter bestimmen, wie die Zeit (Uhrzeitwert aus der Datenpunkt-Historie) angezeigt werden soll.



Horizontale Position	Wählen Sie die horizontale Position der Zeit – links, rechts oder zentriert.
Vertikale Position	Wählen Sie die vertikale Position der Zeit – oben, unten oder mittig.
Größe	Geben Sie die Schrifthöhe für die Zeit ein (in Prozent relativ zur Gesamthöhe des Widgets).
Fett	Aktivieren Sie das Kontrollkästchen, um die Zeit fett darzustellen.
Farbe	Wählen Sie die Farbe der Zeit im Farbwähler aus. D steht für die Standardfarbe (abhängig vom Frontend-Theme). Um zum Standardwert zurückzukehren, klicken Sie im Farbwähler auf die Schaltfläche *Standard verwenden*.

Änderungsindikator

In diesem Abschnitt können Sie die Farbe der Änderungsindikatoren über die Farbauswahl festlegen.



Die Änderungsindikatoren sind wie folgt:

- ↑ – der Wert des Datenpunkts ist gestiegen (bei numerischen Datenpunkten)
- ↓ – der Wert des Datenpunkts ist gesunken (bei numerischen Datenpunkten)
- ↕ – der Wert des Datenpunkts hat sich geändert (bei Datenpunkten vom Typ Zeichenfolge und Datenpunkten mit Wertzuordnung)

Hinweis:

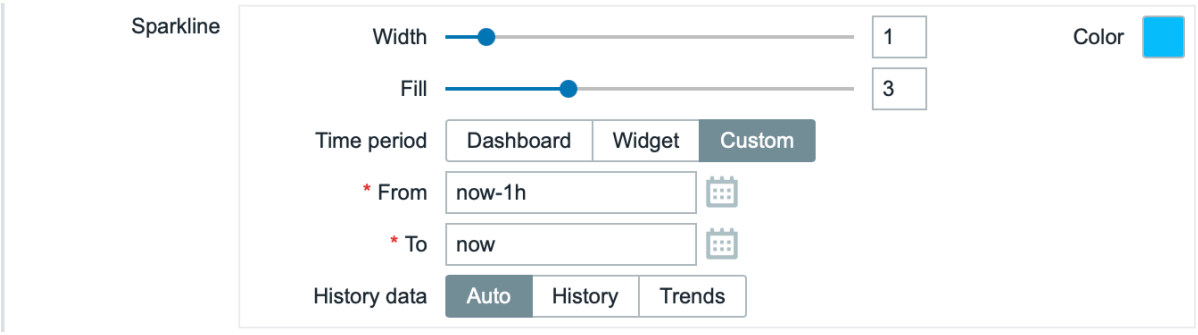
- Die Indikatoren für gestiegene (↑) und gesunkene (↓) Werte werden nicht angezeigt, wenn nur ein Wert vorhanden ist.
- Werte werden immer innerhalb desselben Bereichs verglichen, zum Beispiel:
  - Der neueste Wert wird mit dem vorherigen Wert verglichen.
  - Ein Monatswert wird mit dem vorherigen Monat verglichen.
  - Bei Aggregationen hat der vorherige Zeitraum dieselbe Dauer wie der ausgewählte Zeitraum und endet unmittelbar vor Beginn des ausgewählten Zeitraums.

D steht für die Standardfarbe (abhängig vom Frontend-Theme). Um zum Standardwert zurückzukehren, klicken Sie in der Farbauswahl auf die Schaltfläche *Standard verwenden*.

Die vertikale Größe des Änderungsindikators entspricht der Größe des Werts (bei numerischen Datenpunkten dem ganzzahligen Teil des Werts).

Sparkline

Diese Parameter bestimmen, wie das Sparkline-Diagramm angezeigt werden soll.

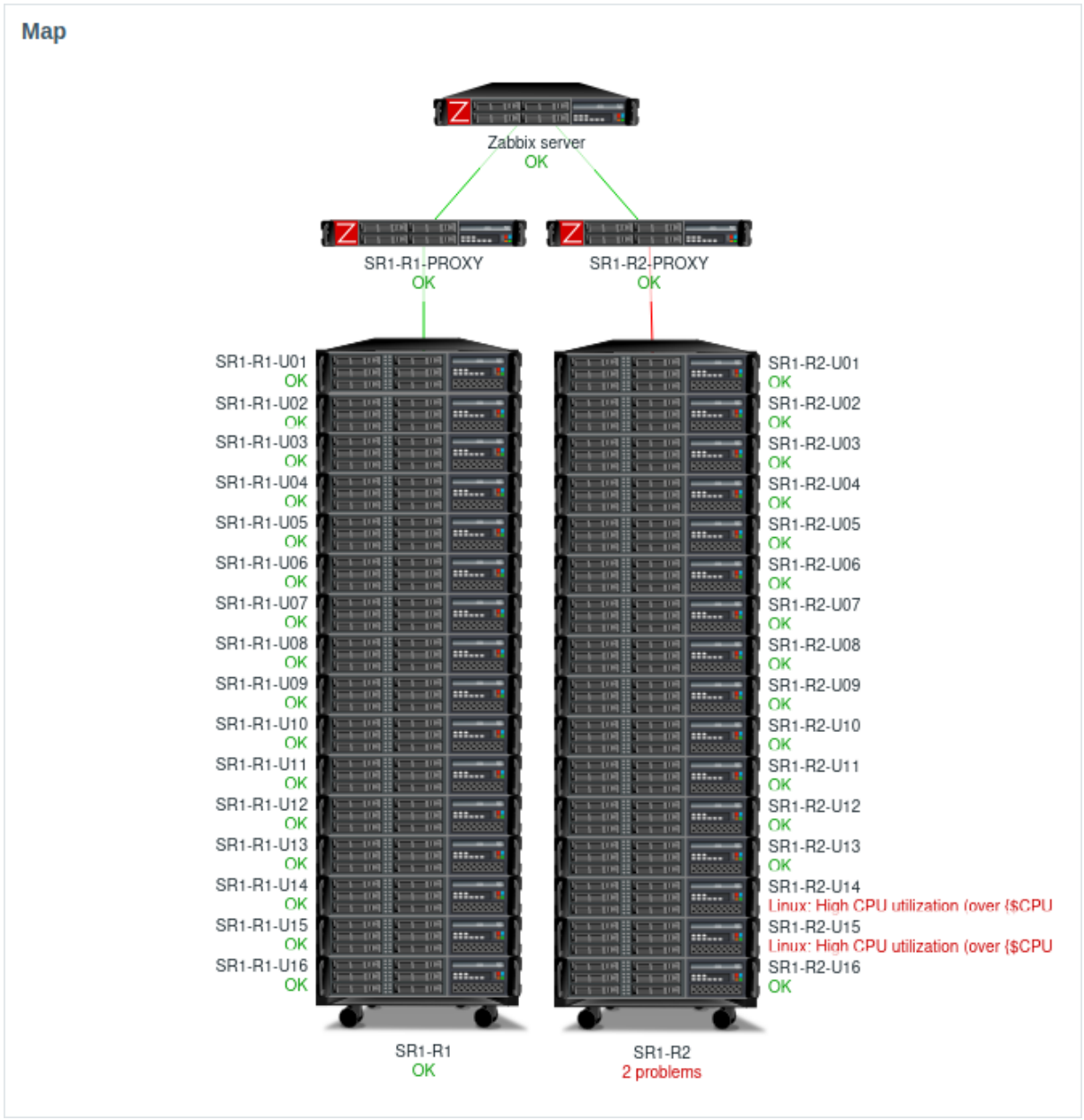


Breite	Legen Sie die Dicke der Diagrammlinie fest, indem Sie den Schieberegler verwenden oder manuell einen Wert im Bereich von 0 bis 10 eingeben.
Farbe	Wählen Sie die Linien- und Füllfarbe aus.
Füllung	Legen Sie die Transparenzstufe der Füllfarbe fest, indem Sie den Schieberegler verwenden oder manuell einen Wert im Bereich von 0 bis 10 eingeben.
Zeitraum	Geben Sie den Zeitraum an, dessen Werte in das Sparkline-Diagramm aufgenommen werden sollen. Wählen Sie die Datenquelle für den Zeitraum aus: **Dashboard** - den Zeitraumauswahl des Dashboards verwenden; **Widget** - ein kompatibles Widget verwenden (im Parameter *Widget* festgelegt); **Benutzerdefiniert** - einen benutzerdefinierten Zeitraum verwenden, der in den Parametern *Von* und *Bis* festgelegt ist; wenn gesetzt, wird in der oberen rechten Ecke des Widgets ein Uhrsymbol angezeigt, das bei Mouseover auf die eingestellte Zeit hinweist. Beachten Sie, dass kompatible Widgets ihn unabhängig von der *Zeitraum*-Konfiguration des Widgets weiterhin als Datenquelle für den Zeitraum verwenden können. Dieser Parameter ist nicht verfügbar, wenn *Aggregationsfunktion* auf „nicht verwendet“ gesetzt ist.
Widget	Geben Sie ein kompatibles Widget als Datenquelle für den Zeitraum ein oder wählen Sie es aus. Dieser Parameter ist verfügbar, wenn *Zeitraum* auf „Widget“ gesetzt ist.
Von	Geben Sie den Beginn des Zeitraums ein oder wählen Sie ihn aus. Die Syntax für relative Zeitangaben (`now`, `now/d`, `now/w-1w` usw.) wird unterstützt. Dieser Parameter ist verfügbar, wenn *Zeitraum* auf „Benutzerdefiniert“ gesetzt ist.
Bis	Geben Sie das Ende des Zeitraums ein oder wählen Sie es aus. Die Syntax für relative Zeitangaben (`now`, `now/d`, `now/w-1w` usw.) wird unterstützt. Dieser Parameter ist verfügbar, wenn *Zeitraum* auf „Benutzerdefiniert“ gesetzt ist.
Verlaufsdaten	Daten aus Verlauf oder Trends verwenden: **Auto** - automatische Auswahl; **Verlauf** - Verlaufsdaten verwenden; **Trends** - Trenddaten verwenden.

Karte

Übersicht

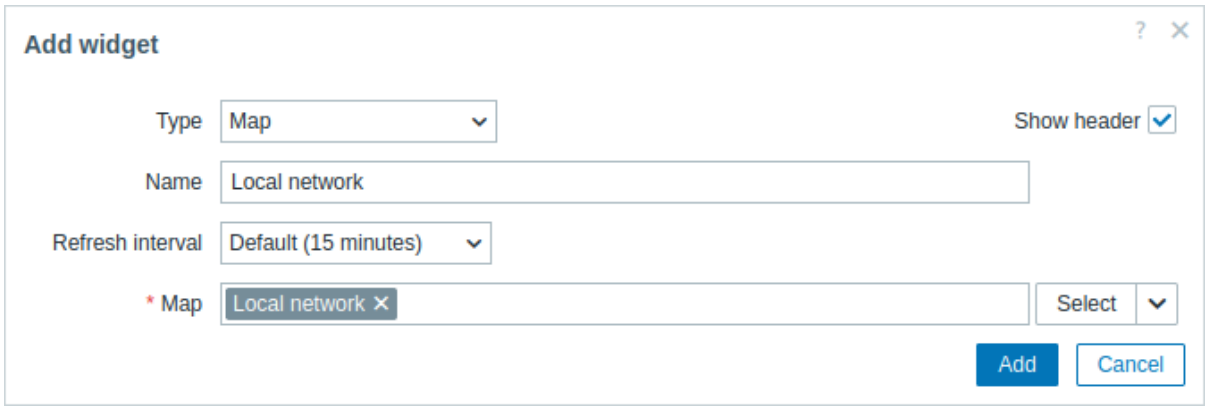
Das Widget *Karte* kann eine Netzwerkkarte oder eine ähnliche Visualisierung anzeigen und Ihnen so eine dynamische Übersicht über Ihr Netzwerk bieten.



Sie können es mit dem Widget Kartennavigationsbaum kombinieren, um dynamisch die im Baum ausgewählte Karte anzuzeigen.

Konfiguration

Um zu konfigurieren, wählen Sie *Karte* als Typ aus:



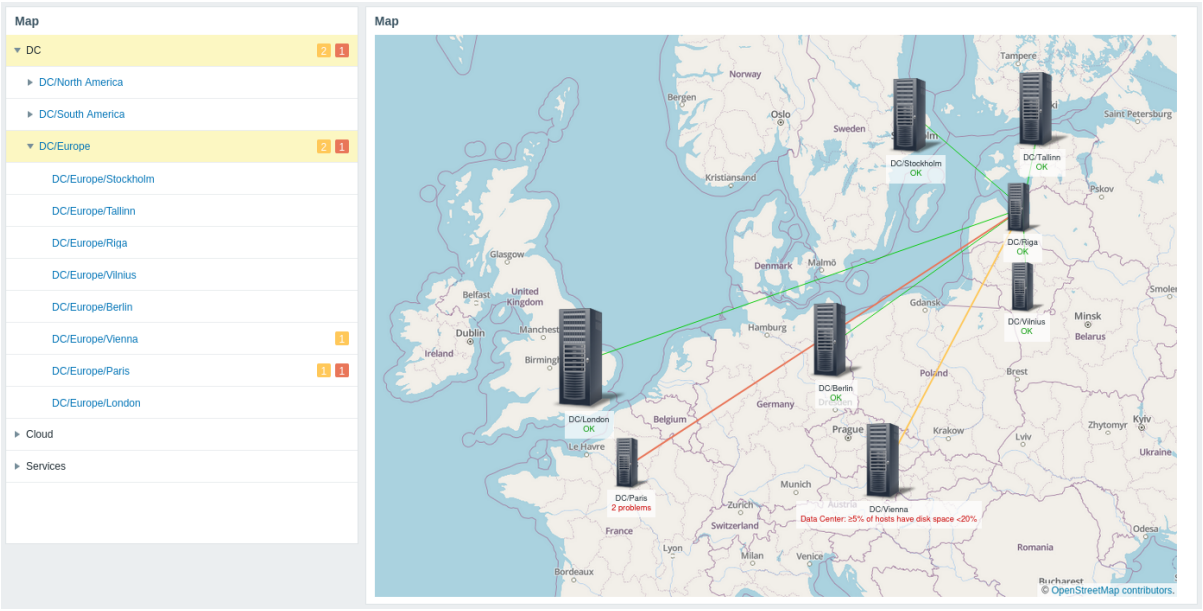
Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

Karte	Legen Sie eine anzuzeigende Karte fest. Alternativ wählen Sie ein kompatibles Widget als Datenquelle für die anzuzeigende Karte aus. Dieses Feld verfügt über eine Autovervollständigung; wenn Sie beginnen, den Namen der Karte oder des Widgets einzugeben, wird eine Dropdown-Liste mit passenden Karten oder Widgets angeboten.

Karten-Navigationsbaum

Übersicht

Das Widget *Map navigation tree* zeigt eine Hierarchie vorhandener Karten sowie die Anzahl der Probleme für jede Karte und Kartengruppe an.

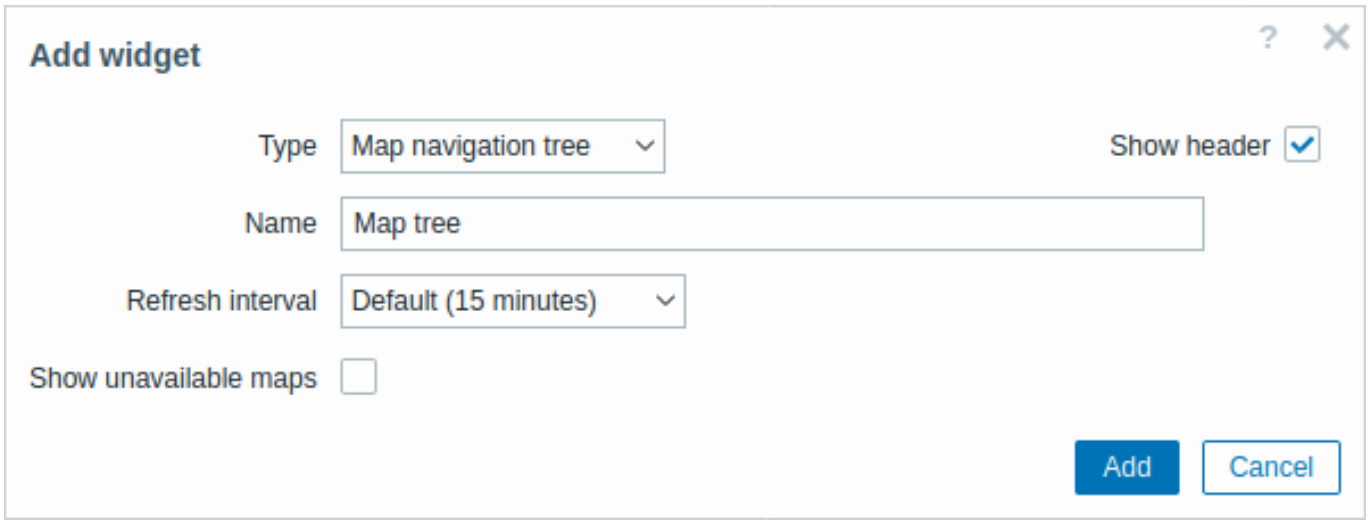


Sie können es mit dem Widget Map kombinieren, um die in der Baumstruktur ausgewählte Karte dynamisch anzuzeigen.

Die Problemanzahl für die Karte der obersten Ebene in der Hierarchie umfasst alle Probleme der Unterkarten sowie ihre eigenen.

Konfiguration

Wählen Sie zur Konfiguration *Map navigation tree* als Typ aus:

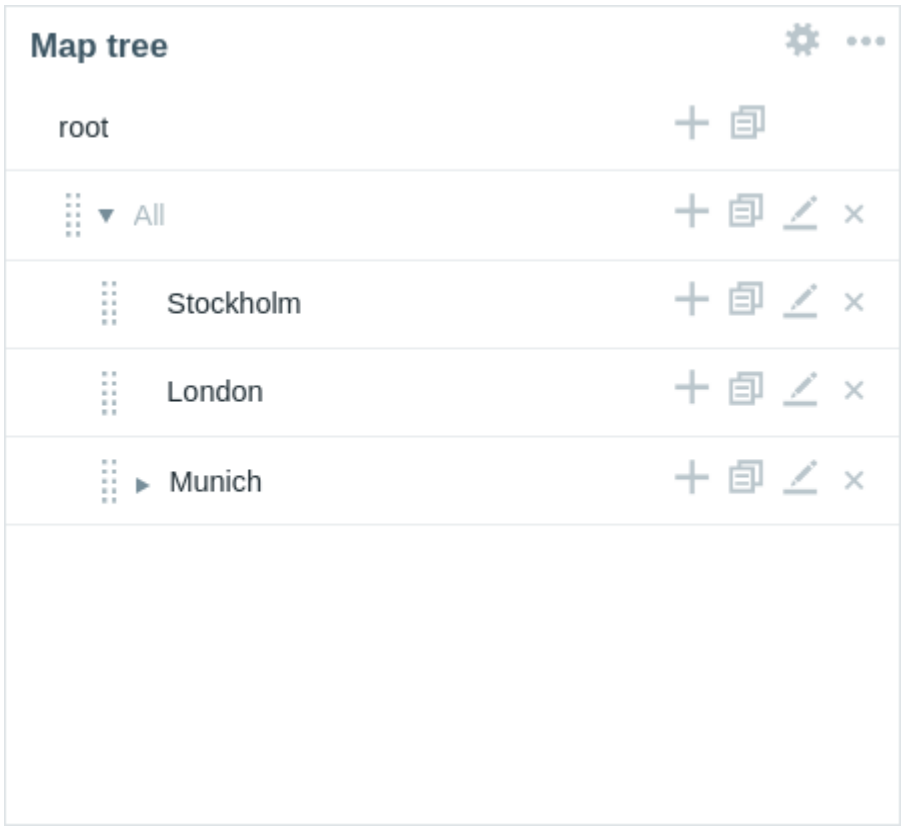


Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

Nicht verfügbare Karten anzeigen	Aktivieren Sie dieses Kontrollkästchen, um Karten anzuzeigen, für die der Benutzer keine Leseberechtigung hat. Nicht verfügbare Karten im Navigationsbaum werden mit einem ausgegrauten Symbol angezeigt. Beachten Sie, dass bei aktiviertem Kontrollkästchen verfügbare Unterkarten auch dann angezeigt werden, wenn die Karte der übergeordneten Ebene nicht verfügbar ist. Ist es nicht aktiviert, werden verfügbare Unterkarten einer nicht verfügbaren übergeordneten Karte überhaupt nicht angezeigt. Die Problemanzahl wird auf Grundlage verfügbarer Karten und verfügbarer Kartenelemente berechnet.

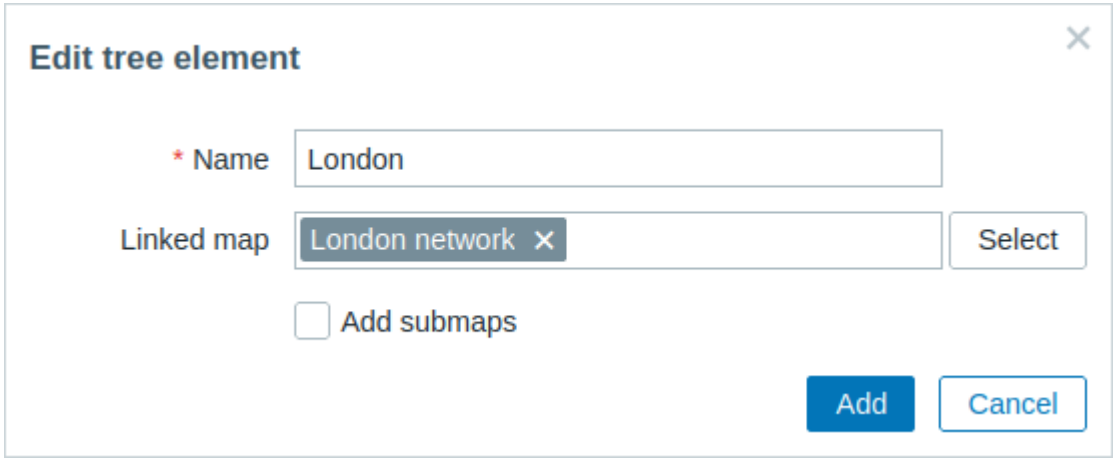
Elemente des Navigationsbaums werden in einer Liste angezeigt. Sie können:

- ein Element (einschließlich seiner untergeordneten Elemente) an eine neue Position in der Liste ziehen;
- ein Element ausklappen oder einklappen, um seine untergeordneten Elemente anzuzeigen oder auszublenden;
- einem Element ein untergeordnetes Element hinzufügen (mit oder ohne verknüpfte Karte);
- einem Element mehrere untergeordnete Elemente hinzufügen (mit verknüpften Karten);
- ein Element bearbeiten;
- ein Element entfernen (einschließlich seiner untergeordneten Elemente).



Element-Konfiguration

Um ein Navigationselement der Baumansicht zu konfigurieren, fügen Sie entweder ein neues Element hinzu oder bearbeiten Sie ein vorhandenes Element.



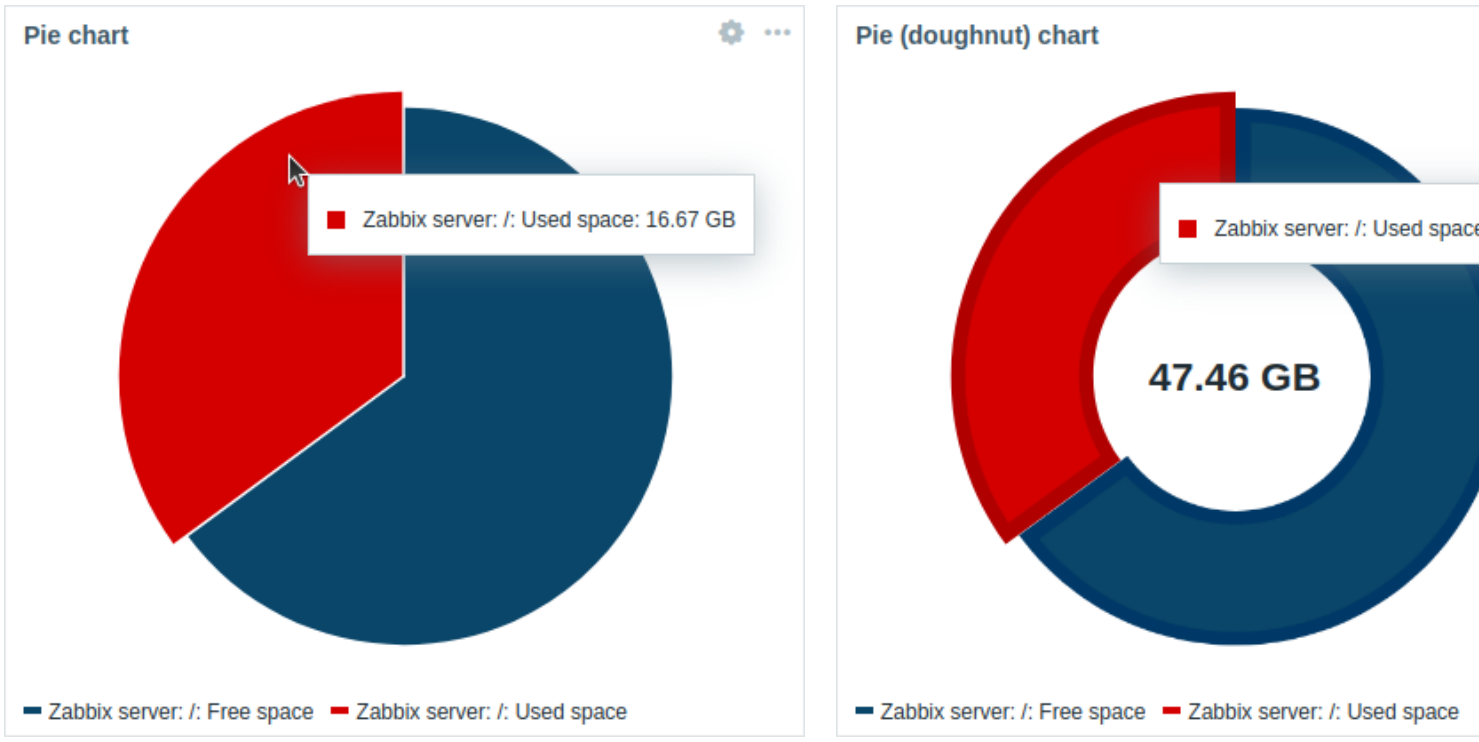
Die folgenden Konfigurationsparameter für Navigationselemente der Baumansicht sind verfügbar:

Name	Geben Sie den Namen des Navigationselements der Baumansicht ein.
Verknüpfte Karte	Wählen Sie die Karte aus, die mit dem Navigationselement der Baumansicht verknüpft werden soll. Dieses Feld verfügt über Autovervollständigung. Wenn Sie beginnen, den Namen einer Karte einzugeben, wird eine Dropdown-Liste mit passenden Karten angezeigt.
Unterkarten hinzufügen	Aktivieren Sie dieses Kontrollkästchen, um die Unterkarten der verknüpften Karte als untergeordnete Elemente zum Navigationselement der Baumansicht hinzuzufügen.

Kreisdiagramm

Übersicht

Das Widget *Kreisdiagramm* zeigt numerische Daten von Datenpunkten als vektorbasiertes Kreis- oder Ringdiagramm an. Mit diesem Widget können Sie visualisieren, welchen Anteil Datenpunkte oder Hosts am Gesamtdatensatz haben.

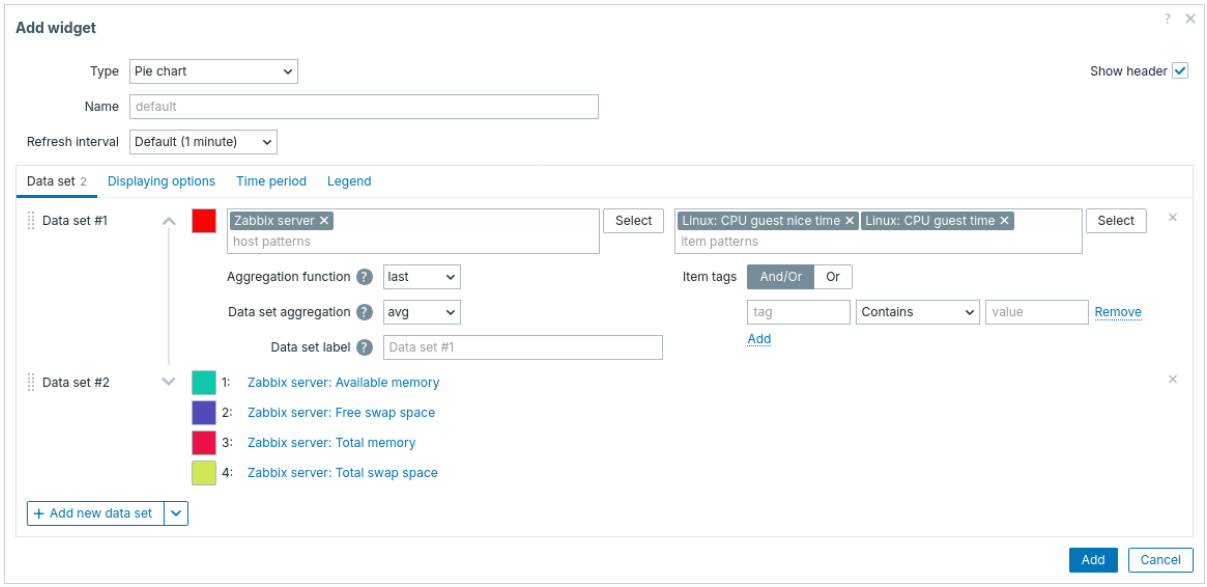


Wenn Sie den Mauszeiger über einen Sektor bewegen, wird der betreffende Sektor erweitert und seine Legende angezeigt. Durch Klicken auf den Sektor bleibt er erweitert und die Legende sichtbar, bis sie geschlossen wird.

Die im Widget *Kreisdiagramm* angezeigten Informationen können als PNG-Bild heruntergeladen werden. Wählen Sie dazu im Menü *Aktionen* des Widgets ... die Option *Bild herunterladen*, die angezeigt wird, wenn Sie den Mauszeiger über das Widget bewegen. Abhängig von Ihren Browsereinstellungen wird das Bild entweder automatisch im Ordner `Downloads` gespeichert oder ein Dialogfeld *Speichern* geöffnet, in dem Sie den Speicherort auswählen können.

Konfiguration

Wählen Sie zur Konfiguration *Kreisdiagramm* als Typ aus:



Datensatz

Auf der Registerkarte **Datensatz** können Sie Daten für das Kreisdiagramm auswählen, indem Sie Datensätze hinzufügen. Es können zwei Arten von Datensätzen hinzugefügt werden:

- *Datenpunktmuster* – Daten aus übereinstimmenden Datenpunkten werden angezeigt. Sie können eine einzelne Grundfarbe auswählen oder eine Palettenzeile auswählen, um jedem übereinstimmenden Datenpunkt eine eigene Farbe zuzuweisen.
- *Datenpunktliste* – Daten aus ausgewählten Datenpunkten werden angezeigt. Sie können die Farbe jedes Datenpunkts einzeln über die Farbauswahl festlegen.

Standardmäßig wird ein Datensatz vom Typ *Datenpunktmuster* hinzugefügt.

Datensatz	Für einen Datensatz vom Typ **Datenpunktmuster**: Wählen Sie Host- und Datenpunktmuster aus oder geben Sie diese ein. Die Daten von Datenpunkten, die den eingegebenen Mustern entsprechen, werden im Kreisdiagramm angezeigt. Es können bis zu 50 Datenpunkte angezeigt werden. Für die Auswahl können Platzhaltermuster verwendet werden (z. B. liefert ∗ Ergebnisse, die null oder mehr Zeichen entsprechen). Um ein Platzhaltermuster anzugeben, geben Sie die Zeichenfolge manuell ein und drücken Sie *Enter*. Das Platzhaltersymbol wird immer interpretiert. Daher ist es beispielsweise nicht möglich, einen Datenpunkt namens *item** einzeln hinzuzufügen, wenn andere übereinstimmende Datenpunkte vorhanden sind (z. B. *item2*, *item3*). Die Angabe von Host- und Datenpunktmustern ist für Datensätze vom Typ „Datenpunktmuster“ erforderlich. Siehe auch: Details zur Datensatzkonfiguration. Für einen Datensatz vom Typ **Datenpunktliste**: Wählen Sie Datenpunkte für das Kreisdiagramm aus, indem Sie auf die Schaltfläche *Datenpunkt hinzufügen* klicken. Über die Dropdown-Liste *Typ* nach dem Namen des Datenpunkts können Sie den Anzeigetyp für jeden Datenpunkt auswählen: **Normal** – der Datenpunktwert wird proportional im Kreisdiagramm dargestellt (Standard); **Gesamt** – der Datenpunktwert nimmt das gesamte Kreisdiagramm ein. Beachten Sie, dass pro Kreisdiagramm nur ein Datenpunkt vom Typ „Gesamt“ vorhanden sein kann und dieser zuerst in der Legende des Kreisdiagramms platziert wird. Wenn ein Datenpunkt auf „Gesamt“ gesetzt ist, wird der Parameter *Datensatzaggregation* (siehe unten) deaktiviert und auf „nicht verwendet“ gesetzt. Sie können auch kompatible Widgets als Datenquelle für Datenpunkte auswählen, indem Sie auf die Schaltfläche *Widget hinzufügen* klicken. Für Widgets gelten dieselben Optionen wie für einzelne Datenpunkte. Die Angabe von Datenpunkten oder Widgets ist für Datensätze vom Typ „Datenpunktliste“ erforderlich. Siehe auch: Details zur Datensatzkonfiguration. Beachten Sie, dass nur numerische Datenpunkttypen zulässig sind. Bei der Konfiguration des Widgets auf einem Vorlagen-Dashboard ist der Parameter zur Angabe von Hostmustern nicht verfügbar. Der Parameter zur Angabe einer Datenpunktliste ermöglicht nur die Auswahl der auf der Vorlage konfigurierten Datenpunkte.
Aggregationsfunktion	Geben Sie an, welche Aggregationsfunktion für jeden Datenpunkt im Datensatz innerhalb des ausgewählten *Zeitraums* verwendet werden soll: **min** – den kleinsten Wert anzeigen; **max** – den größten Wert anzeigen; **avg** – den Durchschnittswert anzeigen; **sum** – die Summe der Werte anzeigen; **count** – die Anzahl der Werte anzeigen; **first** – den ersten Wert anzeigen; **last** – den letzten Wert anzeigen (Standard). Durch die Aggregation kann für das angegebene Intervall (5 Minuten, eine Stunde, ein Tag) ein aggregierter Wert angezeigt werden. Für *min*, *max*, und *sum* können nur numerische Daten angezeigt werden. Für *count* werden nicht numerische Daten in numerische Daten umgewandelt.

Datensatzaggregation	Geben Sie an, welche Aggregationsfunktion für den gesamten Datensatz verwendet werden soll: **nicht verwendet** – keine Aggregation, Datenpunkte werden separat angezeigt (Standard); **min** – den kleinsten Wert anzeigen; **max** – den größten Wert anzeigen; **avg** – den Durchschnittswert anzeigen; **sum** – die Summe der Werte anzeigen; **count** – die Anzahl der Werte anzeigen. Durch die Aggregation kann ein aggregierter Wert für das angegebene Intervall (5 Minuten, eine Stunde, ein Tag) angezeigt werden, das auf der Registerkarte *Zeitraum* ausgewählt oder für das gesamte Dashboard verwendet wird.
Datensatzbezeichnung	Geben Sie eine benutzerdefinierte Bezeichnung für den Datensatz an. Die Bezeichnung wird in der Datensatzkonfiguration und in der Legende des Kreisdiagramms (für aggregierte Datensätze) angezeigt. Alle Datensätze werden nummeriert, einschließlich der Datensätze mit einer angegebenen *Datensatzbezeichnung*. Wenn keine Bezeichnung angegeben ist, wird der Datensatz automatisch entsprechend seiner Nummer bezeichnet (z. B. „Datensatz Nr. 2“, „Datensatz Nr. 3“ usw.). Die Nummerierung der Datensätze wird nach dem Neuanordnen/Verschieben von Datensätzen neu berechnet. Zu lange Datensatzbezeichnungen werden gekürzt, damit sie in den Anzeigebereich passen (z. B. „Anzahl der Proz...“).
Datenpunkt-Tags	Geben Sie Tags an, nach denen die im Widget angezeigten Datenpunkte gefiltert werden sollen. Es ist möglich, bestimmte Tags und Tagwerte einzuschließen oder auszuschließen. Es können mehrere Bedingungen festgelegt werden. Beim Abgleich von Tagnamen wird immer zwischen Groß- und Kleinschreibung unterschieden. Für jede Bedingung stehen mehrere Operatoren zur Verfügung: **Existiert** – die angegebenen Tagnamen einschließen; **Ist gleich** – die angegebenen Tagnamen und Tagwerte einschließen (Groß-/Kleinschreibung wird berücksichtigt); **Enthält** – die angegebenen Tagnamen einschließen, deren Tagwerte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich ohne Berücksichtigung der Groß-/Kleinschreibung); **Existiert nicht** – die angegebenen Tagnamen ausschließen; **Ist nicht gleich** – die angegebenen Tagnamen und Tagwerte ausschließen (Groß-/Kleinschreibung wird berücksichtigt); **Enthält nicht** – die angegebenen Tagnamen ausschließen, deren Tagwerte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich ohne Berücksichtigung der Groß-/Kleinschreibung). Für Bedingungen stehen zwei Berechnungstypen zur Verfügung: **Und/Oder** – alle Bedingungen müssen erfüllt sein; Bedingungen mit demselben Tagnamen werden durch die Oder-Bedingung gruppiert; **Oder** – es genügt, wenn eine Bedingung erfüllt ist.

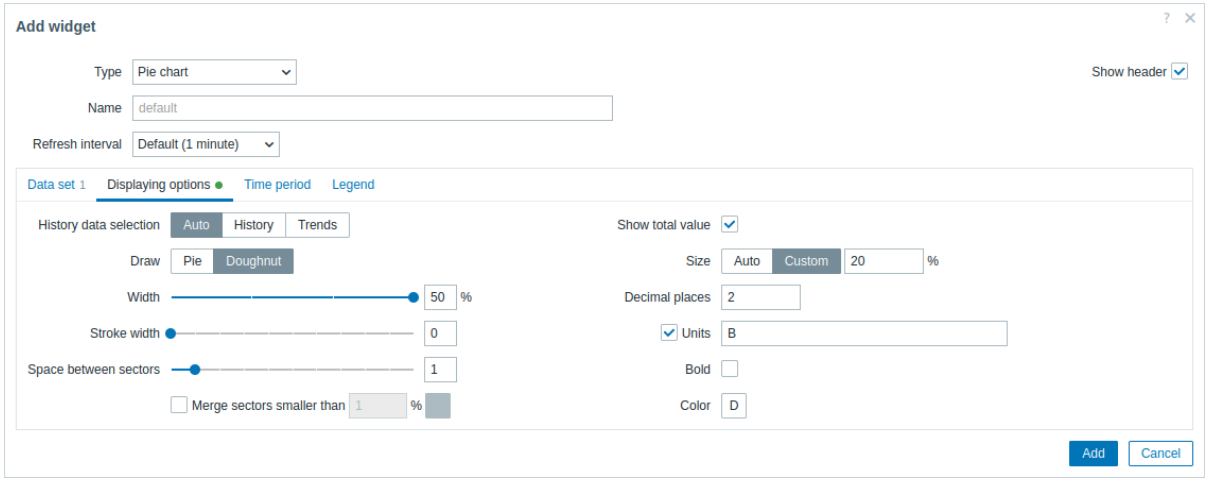
Details zur Datenpunkt-Konfiguration

Vorhandene Datensätze werden in einer Liste angezeigt. Sie können diese Datensätze neu anordnen, aus-/einklappen, Farben ändern und klonen.

Weitere Informationen finden Sie unter Details zur Datensatzkonfiguration im Widget *Graph*. Diese Details gelten auch für das Widget *Pie chart*.

Anzeigeoptionen

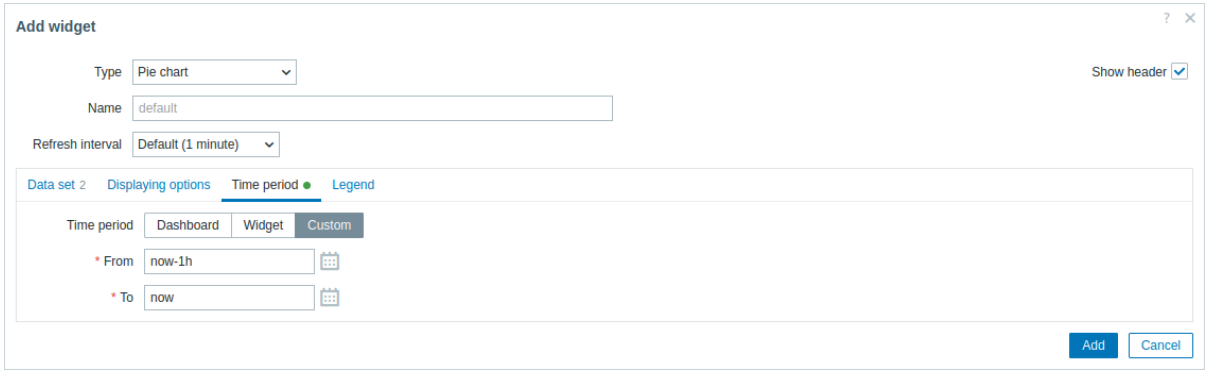
Auf der Registerkarte **Anzeigeoptionen** können Sie die Auswahl- und Visualisierungsoptionen für Verlaufsdaten im Kreisdiagramm festlegen:



Auswahl der Verlaufsdaten	Wählen Sie die Datenquelle aus: **Auto** – Daten werden gemäß dem klassischen Algorithmus bezogen (Standard); **Verlauf** – Daten aus dem Verlauf; **Trends** – Daten aus Trends.
Darstellung	Wählen Sie den Visualisierungsstil des Kreisdiagramms aus: **Kreis** – ein vollständiger Kreis (Sektoren nehmen 100 % des Radius ein); **Ring** – ein Kreis mit einem leeren Kreis in der Mitte (Sektoren nehmen bis zu 50 % des Radius ein).
Abstand zwischen Sektoren	Wählen Sie die Größe des Abstands (in Einheiten von 0 bis 10) zwischen den Sektoren aus (Standard ist „1“).
Sektoren kleiner als N % zusammenführen	Aktivieren Sie das Kontrollkästchen, um Sektoren zusammenzuführen, die kleiner als N % sind. Wenn diese Option aktiviert ist, wählen Sie die Farbe für die zusammengeführten Sektoren und den prozentualen Schwellenwert (N) für das Zusammenführen kleiner Sektoren aus.
Darstellungsstil: ***Ring***	
Breite	Wählen Sie die Ringbreite aus: 20, 30, 40 oder 50 % (Standard) des Radius.
Linienbreite	Wählen Sie die Breite des Randes des Ringsektors aus (0–10).
Gesamtwert anzeigen	Aktivieren Sie das Kontrollkästchen, um den Gesamtwert in der Mitte des Ringdiagramms anzuzeigen.
Größe	Wählen Sie die Größenoption für den Gesamtwert aus: **Auto** – die Textgröße wird automatisch so angepasst, dass der Text in der Mitte des Rings gut lesbar ist; **Benutzerdefiniert** – geben Sie die Textgröße als prozentuale Höhe bezogen auf die Gesamthöhe des Widgets an.
Dezimalstellen	Geben Sie die Anzahl der Dezimalstellen für den Gesamtwert an (0–6).
Einheiten	Geben Sie die Einheiten für den Gesamtwert an.
Fett	Aktivieren Sie das Kontrollkästchen, um den Gesamtwert fett darzustellen.
Farbe	Wählen Sie die Farbe für den Gesamtwert aus.

Zeitraum

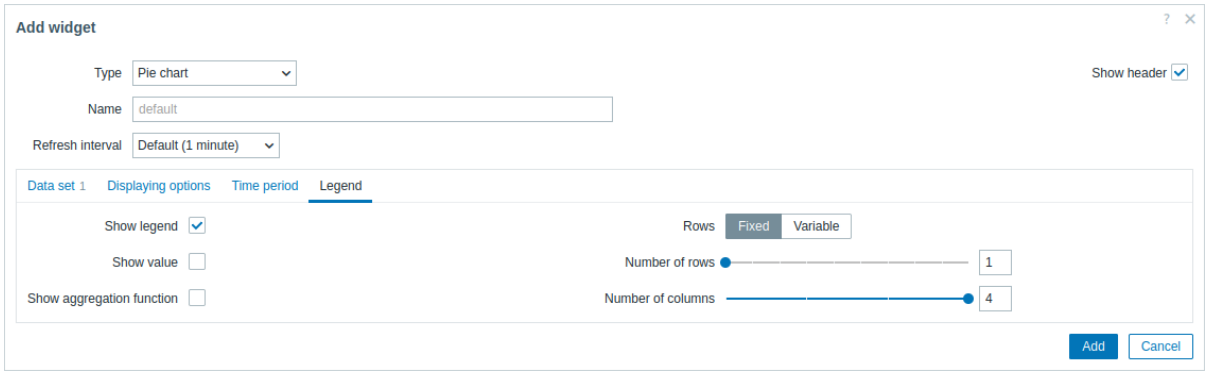
Auf der Registerkarte **Zeitraum** können Sie einen benutzerdefinierten Zeitraum für die Aggregationseinstellungen des Kreisdiagramms festlegen:



Zeitraum	Wählen Sie die Datenquelle für den Zeitraum aus: **Dashboard** – verwendet den Zeitraumauswahlschalter des Dashboards; **Widget** – verwendet ein kompatibles Widget (im Parameter *Widget* festgelegt); **Benutzerdefiniert** – verwendet einen benutzerdefinierten Zeitraum, der in den Parametern *Von* und *Bis* angegeben wird; wenn dieser festgelegt ist, wird in der oberen rechten Ecke des Widgets ein Uhrsymbol angezeigt, das bei einem Mouseover den festgelegten Zeitpunkt angibt. Beachten Sie, dass kompatible Widgets unabhängig von der Konfiguration *Zeitraum* des Widgets weiterhin als Datenquelle für den Zeitraum verwendet werden können.
Widget	Geben Sie ein kompatibles Widget (*Graph*, *Graph (classic)*, *Graph prototype*) als Datenquelle für den Zeitraum ein oder wählen Sie es aus. Dieser Parameter ist verfügbar, wenn *Zeitraum* auf „Widget“ gesetzt ist.
Von	Geben Sie den Beginn des Zeitraums ein oder wählen Sie ihn aus. Die Syntax für relative Zeitangaben (`now`, `now/d`, `now/w-1w` usw.) wird unterstützt. Dieser Parameter ist verfügbar, wenn *Zeitraum* auf „Benutzerdefiniert“ gesetzt ist.
Bis	Geben Sie das Ende des Zeitraums ein oder wählen Sie es aus. Die Syntax für relative Zeitangaben (`now`, `now/d`, `now/w-1w` usw.) wird unterstützt. Dieser Parameter ist verfügbar, wenn *Zeitraum* auf „Benutzerdefiniert“ gesetzt ist.

Legende

Auf der Registerkarte **Legende** können Sie die Legende des Kreisdiagramms anpassen:



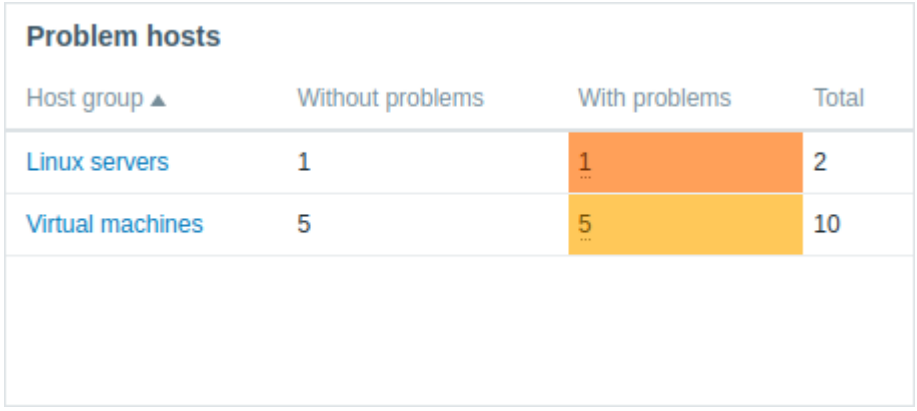
Legende anzeigen	Deaktivieren Sie dieses Kontrollkästchen, um die Legende im Kreisdiagramm auszublenden (standardmäßig aktiviert).
Wert anzeigen	Aktivieren Sie dieses Kontrollkästchen, um den Wert des Datenpunkts in der Legende anzuzeigen.
Aggregationsfunktion anzeigen	Aktivieren Sie dieses Kontrollkästchen, um die Aggregationsfunktion in der Legende anzuzeigen.
Zeilen	Wählen Sie den Anzeigemodus für die Zeilen der Legende aus: **Fest** – die Anzahl der angezeigten Zeilen wird durch den Wert des Parameters *Anzahl der Zeilen* bestimmt; **Variabel** – die Anzahl der angezeigten Zeilen wird durch die Anzahl der konfigurierten Datenpunkte bestimmt, darf jedoch den Wert des Parameters *Maximale Anzahl der Zeilen* nicht überschreiten.
Anzahl der Zeilen/ Maximale Anzahl der Zeilen	Wenn *Zeilen* auf „Fest“ gesetzt ist, legen Sie die Anzahl der anzuzeigenden Legendenzeilen fest (1–10). Wenn *Zeilen* auf „Variabel“ gesetzt ist, legen Sie die maximale Anzahl der anzuzeigenden Legendenzeilen fest (1–10).

Anzahl der Spalten	Legen Sie die Anzahl der anzuzeigenden Legendenspalten fest (1–4). Dieser Parameter ist verfügbar, wenn *Wert anzeigen* deaktiviert ist.

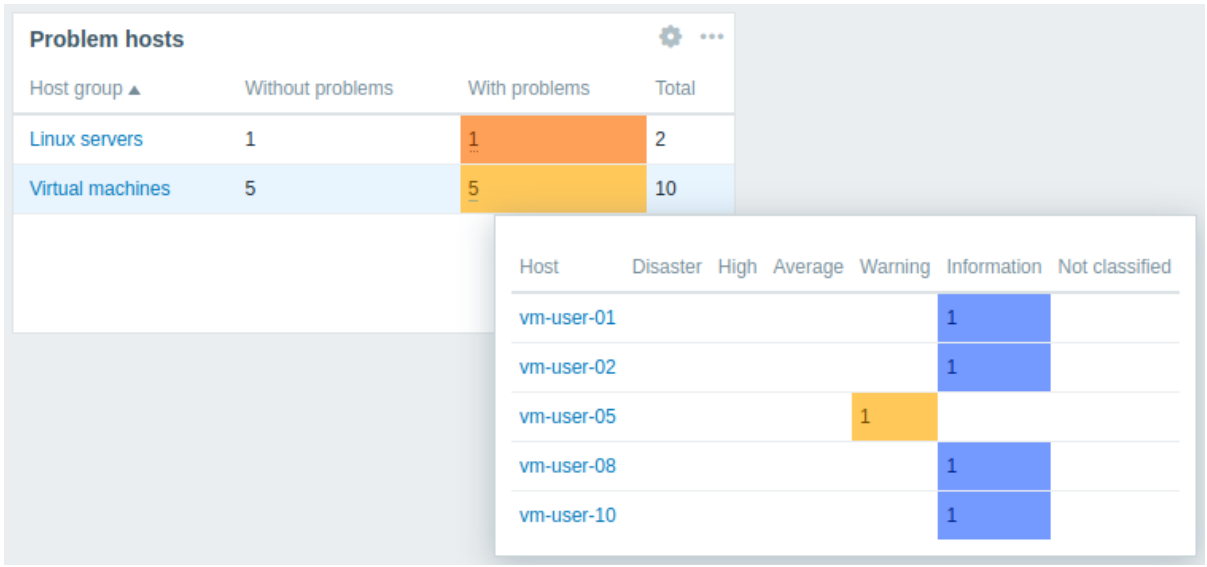
Problem-Hosts

Übersicht

Das Widget *Problem-Hosts* zeigt die Anzahl der Ursachenprobleme pro Host-Gruppe an und zeigt den höchsten Problemschweregrad in jeder Gruppe. Es ermöglicht das Filtern von Problemen nach verschiedenen Parametern (Host-Gruppen, Hosts, Problemnamen usw.).

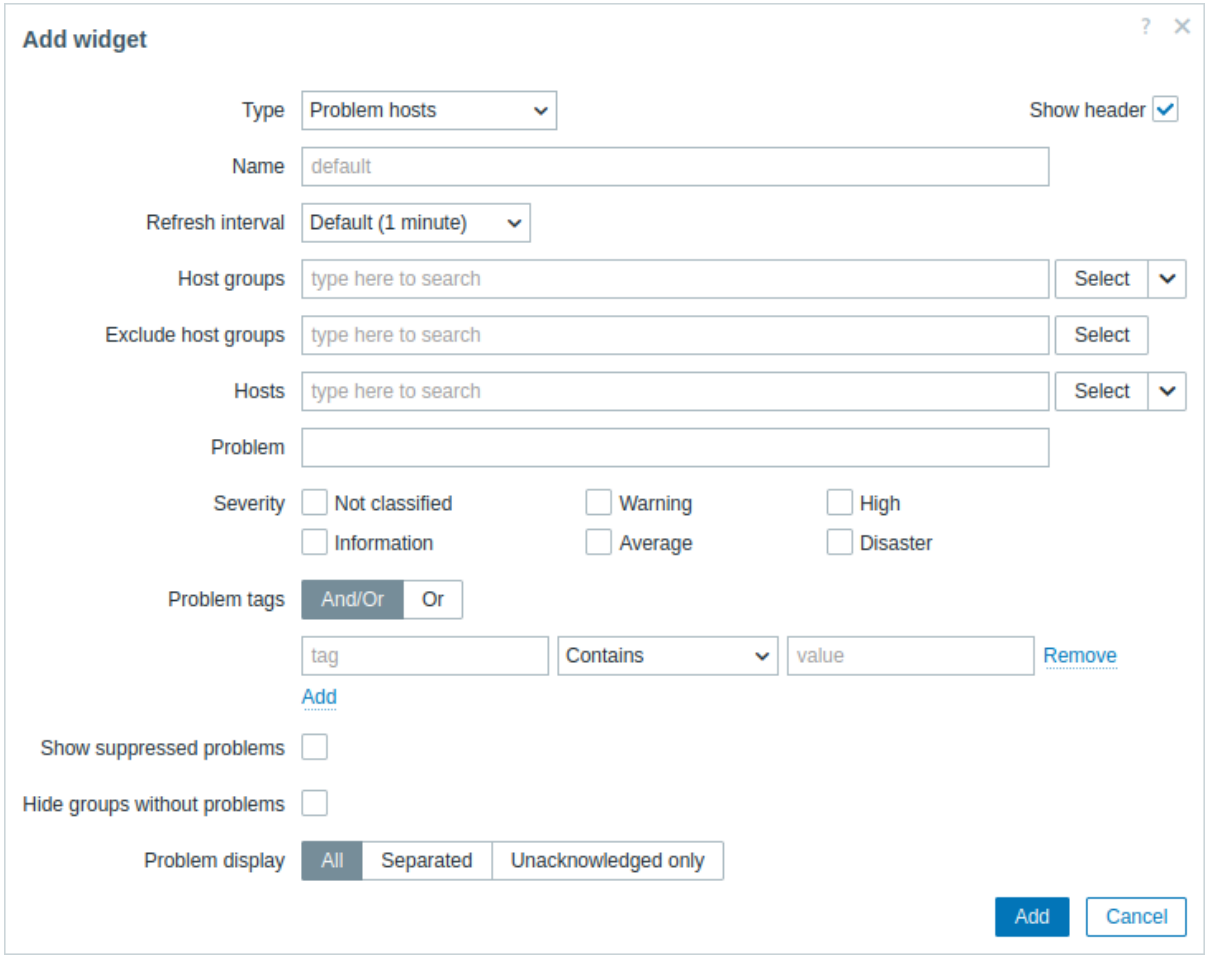


Wenn Sie mit der Maus über eine Problemanzahl fahren, wird die Liste der Hosts mit Problemen angezeigt:



Konfiguration

Um zu konfigurieren, wählen Sie *Problem-Hosts* als Typ aus:



Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

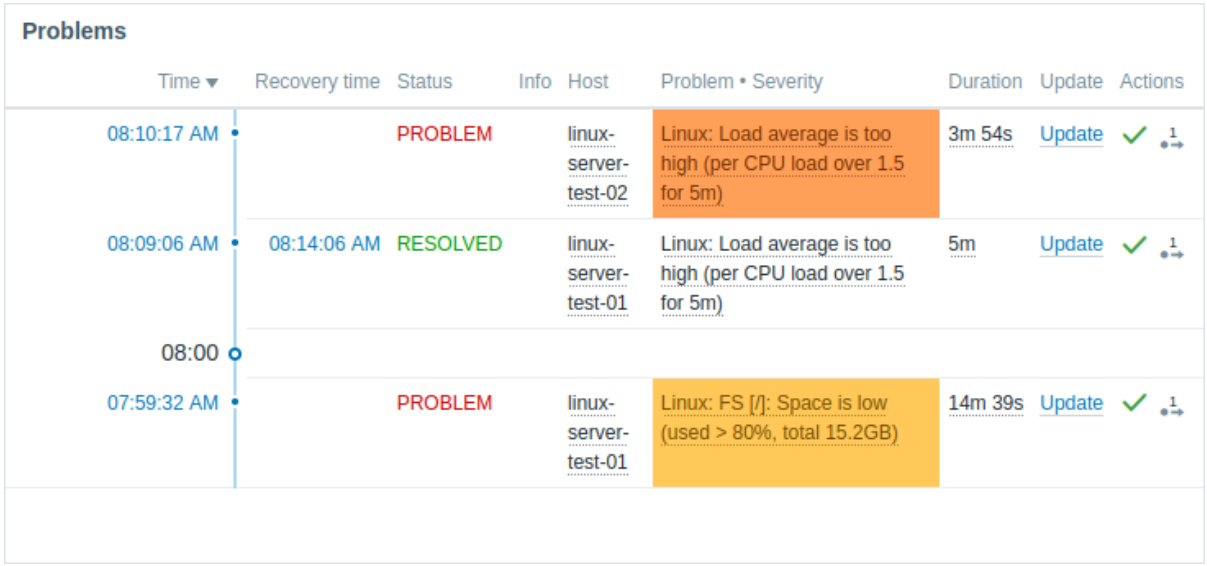
Host-Gruppen	Wählen Sie Host-Gruppen aus, die im Widget angezeigt werden sollen. Alternativ können Sie ein kompatibles Widget als Datenquelle für Host-Gruppen auswählen. Dieses Feld unterstützt Autovervollständigung. Wenn Sie also beginnen, den Namen einer Gruppe einzugeben, wird eine Dropdown-Liste mit passenden Gruppen angeboten. Wenn Sie eine übergeordnete Host-Gruppe angeben, werden implizit auch alle untergeordneten Host-Gruppen ausgewählt. Host-Daten aus diesen Host-Gruppen werden im Widget angezeigt; wenn keine Host-Gruppen eingegeben werden, werden alle Host-Gruppen angezeigt. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Host-Gruppen ausschließen	Wählen Sie Host-Gruppen aus, die im Widget ausgeblendet werden sollen. Dieses Feld unterstützt Autovervollständigung. Wenn Sie also beginnen, den Namen einer Gruppe einzugeben, wird eine Dropdown-Liste mit passenden Gruppen angeboten. Wenn Sie eine übergeordnete Host-Gruppe angeben, werden implizit auch alle untergeordneten Host-Gruppen ausgewählt. Host-Daten aus diesen Host-Gruppen werden im Widget nicht angezeigt. Zum Beispiel können sich die Hosts 001, 002, 003 in Gruppe A befinden und die Hosts 002, 003 zusätzlich in Gruppe B. Wenn wir gleichzeitig Gruppe A *anzeigen* und Gruppe B *ausschließen*, werden im Dashboard nur Daten von Host 001 angezeigt. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Hosts	Wählen Sie Hosts aus, die im Widget angezeigt werden sollen. Alternativ können Sie ein kompatibles Widget oder das Dashboard als Datenquelle für Hosts auswählen. Dieses Feld unterstützt Autovervollständigung. Wenn Sie also beginnen, den Namen eines Hosts einzugeben, wird eine Dropdown-Liste mit passenden Hosts angeboten. Wenn keine Hosts eingegeben werden, werden alle Hosts angezeigt. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.

Problem	Sie können die Anzahl der angezeigten Problem-Hosts anhand des Problemnamens begrenzen. Wenn Sie hier eine Zeichenfolge eingeben, werden nur die Hosts angezeigt, deren Probleme einen Namen enthalten, der die eingegebene Zeichenfolge enthält. Makros werden nicht erweitert.
Schweregrad	Markieren Sie Problem-Schweregrade, um die im Widget anzuzeigenden Probleme zu filtern. Wenn keine Schweregrade markiert sind, werden alle Probleme angezeigt.
Problem-Tags	Geben Sie Problem-Tags an, um die Anzahl der im Widget angezeigten Probleme zu begrenzen. Es ist möglich, bestimmte Tags und Tag-Werte sowohl einzuschließen als auch auszuschließen. Es können mehrere Bedingungen festgelegt werden. Der Abgleich von Tag-Namen ist immer groß-/kleinschreibungssensitiv. Für jede Bedingung stehen mehrere Operatoren zur Verfügung: **Existiert** - die angegebenen Tag-Namen einschließen; **Gleich** - die angegebenen Tag-Namen und -Werte einschließen (groß-/kleinschreibungssensitiv); **Enthält** - die angegebenen Tag-Namen einschließen, bei denen die Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv); **Existiert nicht** - die angegebenen Tag-Namen ausschließen; **Ungleich** - die angegebenen Tag-Namen und -Werte ausschließen (groß-/kleinschreibungssensitiv); **Enthält nicht** - die angegebenen Tag-Namen ausschließen, bei denen die Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv). Für Bedingungen gibt es zwei Berechnungstypen: **Und/Oder** - alle Bedingungen müssen erfüllt sein; Bedingungen mit demselben Tag-Namen werden durch die *Oder*-Bedingung gruppiert; **Oder** - es genügt, wenn eine Bedingung erfüllt ist.
Unterdrückte Probleme anzeigen	Aktivieren Sie das Kontrollkästchen, um Probleme anzuzeigen, die andernfalls aufgrund von Host-Wartung unterdrückt (nicht angezeigt) würden.
Gruppen ohne Probleme ausblenden	Aktivieren Sie die Option *Gruppen ohne Probleme ausblenden*, um Daten aus Host-Gruppen ohne Probleme im Widget auszublenden. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Problemanzeige	Problemanzahl anzeigen als: **Alle** - die vollständige Problemanzahl wird angezeigt; **Getrennt** - die Anzahl der nicht bestätigten Probleme wird getrennt als Zahl der gesamten Problemanzahl angezeigt; **Nur nicht bestätigte** - es wird nur die Anzahl der nicht bestätigten Probleme angezeigt.

Probleme

Übersicht

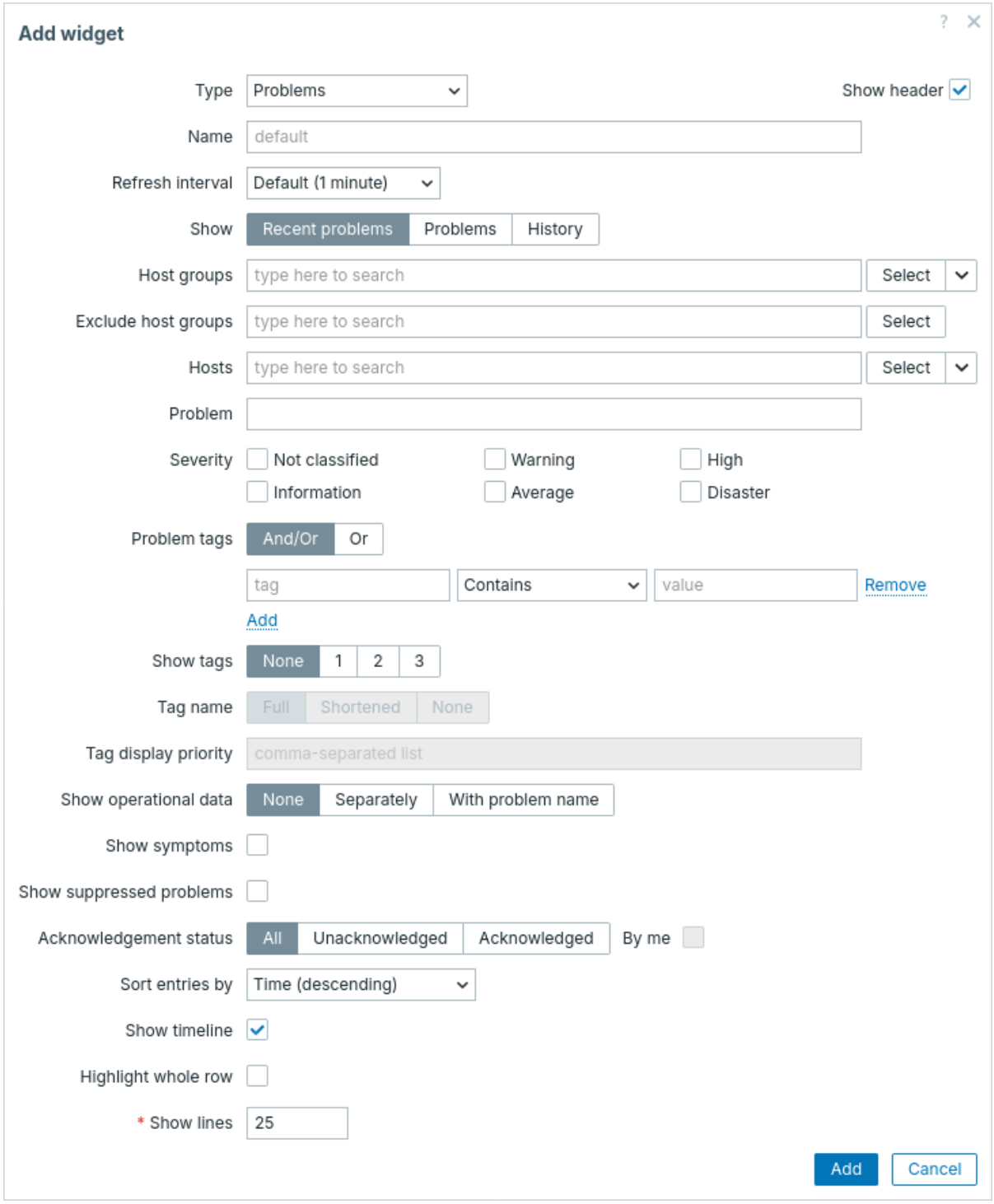
Das Widget *Probleme* zeigt aktuelle Probleme an, gefiltert nach verschiedenen Parametern (Host-Gruppen, Hosts, Problemnamen usw.), und bietet Ihnen so eine klare Übersicht darüber, was Aufmerksamkeit erfordert.



Es zeigt dieselben Daten wie *Monitoring* > *Probleme* an und kann bis zu 1000 Einträge anzeigen.

Konfiguration

Um zu konfigurieren, wählen Sie *Problems* als Typ:



Sie können auf verschiedene Weise begrenzen, wie viele Probleme im Widget angezeigt werden - nach Problemstatus, Problemname, Schweregrad, Hostgruppe, Host, Ereignistag, Bestätigungsstatus usw.

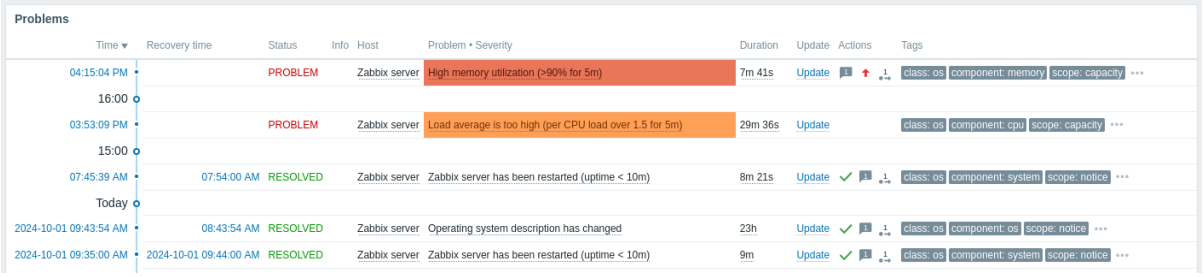
Zusätzlich zu den gemeinsamen Parametern für alle Widgets können Sie die folgenden spezifischen Optionen festlegen:

Show	Nach Problemstatus filtern: **Recent problems** - ungelöste und kürzlich gelöste Probleme werden angezeigt (Standard); **Problems** - ungelöste Probleme werden angezeigt; **History** - der Verlauf aller Ereignisse wird angezeigt.

Host groups	Wählen Sie Hostgruppen aus, deren Probleme im Widget angezeigt werden sollen. Alternativ können Sie ein kompatibles Widget als Datenquelle für Hostgruppen auswählen. Dieses Feld unterstützt Auto-Vervollständigung. Wenn Sie mit der Eingabe des Namens einer Gruppe beginnen, wird eine Dropdown-Liste mit passenden Gruppen angezeigt. Wenn Sie eine übergeordnete Hostgruppe angeben, werden implizit alle verschachtelten Hostgruppen ausgewählt. Probleme aus diesen Hostgruppen werden im Widget angezeigt; wenn keine Hostgruppen angegeben werden, werden Probleme aus allen Hostgruppen angezeigt. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlage-Dashboard konfiguriert wird.
Exclude host groups	Wählen Sie Hostgruppen aus, deren Probleme im Widget ausgeblendet werden sollen. Dieses Feld unterstützt Auto-Vervollständigung. Wenn Sie mit der Eingabe des Namens einer Gruppe beginnen, wird eine Dropdown-Liste mit passenden Gruppen angezeigt. Wenn Sie eine übergeordnete Hostgruppe angeben, werden implizit alle verschachtelten Hostgruppen ausgewählt. Probleme aus diesen Hostgruppen werden nicht angezeigt. Wenn beispielsweise die Hosts 001, 002, 003 in Gruppe A und die Hosts 002, 003 zusätzlich in Gruppe B sind, werden bei gleichzeitiger Auswahl von *show* Gruppe A und *exclude* Gruppe B nur Probleme von Host 001 im Widget angezeigt. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlage-Dashboard konfiguriert wird.
Hosts	Wählen Sie Hosts aus, deren Probleme im Widget angezeigt werden sollen. Alternativ können Sie ein kompatibles Widget oder das Dashboard als Datenquelle für Hosts auswählen. Dieses Feld unterstützt Auto-Vervollständigung. Wenn Sie mit der Eingabe des Namens eines Hosts beginnen, wird eine Dropdown-Liste mit passenden Hosts angezeigt. Wenn keine Hosts angegeben werden, werden Probleme aller Hosts angezeigt. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlage-Dashboard konfiguriert wird.
Problem	Sie können die Anzahl der angezeigten Probleme nach ihrem Namen begrenzen. Wenn Sie hier eine Zeichenfolge eingeben, werden nur die Probleme angezeigt, deren Name die eingegebene Zeichenfolge enthält. Makros werden nicht erweitert.
Severity	Markieren Sie Problem-Schweregrade, um die im Widget angezeigten Probleme zu filtern. Wenn keine Schweregrade markiert sind, werden alle Probleme angezeigt.
Problem tags	Geben Sie Problem-Tags an, um die Anzahl der im Widget angezeigten Probleme zu begrenzen. Es ist möglich, bestimmte Tags und Tag-Werte einzuschließen oder auszuschließen. Es können mehrere Bedingungen festgelegt werden. Die Übereinstimmung des Tag-Namens ist immer groß-/kleinschreibungssensitiv. Für jede Bedingung stehen mehrere Operatoren zur Verfügung: **Exists** - die angegebenen Tag-Namen einschließen; **Equals** - die angegebenen Tag-Namen und Werte einschließen (groß-/kleinschreibungssensitiv); **Contains** - die angegebenen Tag-Namen einschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv); **Does not exist** - die angegebenen Tag-Namen ausschließen; **Does not equal** - die angegebenen Tag-Namen und Werte ausschließen (groß-/kleinschreibungssensitiv); **Does not contain** - die angegebenen Tag-Namen ausschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv). Für Bedingungen gibt es zwei Berechnungstypen: **And/Or** - alle Bedingungen müssen erfüllt sein, Bedingungen mit demselben Tag-Namen werden durch die *Or*-Bedingung gruppiert; **Or** - es genügt, wenn eine Bedingung erfüllt ist. Bei einer Filterung werden die hier angegebenen Tags zusammen mit dem Problem zuerst angezeigt, sofern sie nicht durch die Liste *Tag display priority* (siehe unten) überschrieben werden.

Show tags	Wählen Sie die Anzahl der angezeigten Tags: **None** - keine Spalte *Tags*; **1** - die Spalte *Tags* enthält einen Tag; **2** - die Spalte *Tags* enthält zwei Tags; **3** - die Spalte *Tags* enthält drei Tags. Um alle Tags für das Problem anzuzeigen, bewegen Sie den Mauszeiger über das Symbol mit den drei Punkten.
Tag name	Wählen Sie den Anzeigemodus für Tag-Namen: **Full** - Tag-Namen und Werte werden vollständig angezeigt; **Shortened** - Tag-Namen werden auf 3 Zeichen gekürzt, die Tag-Werte werden jedoch vollständig angezeigt; **None** - es werden nur Tag-Werte angezeigt; keine Namen.
Tag display priority	Geben Sie die Anzeigereihenfolge für Tags eines Problems als durch Kommas getrennte Liste von Tags ein. Es dürfen nur Tag-Namen verwendet werden, keine Werte. Beispiel: `Services,Applications,Application` Die Tags dieser Liste werden immer zuerst angezeigt und überschreiben die natürliche alphabetische Reihenfolge.
Show operational data	Wählen Sie den Modus für die Anzeige von Betriebsdaten: **None** - es werden keine Betriebsdaten angezeigt; **Separately** - Betriebsdaten werden in einer separaten Spalte angezeigt; **With problem name** - Betriebsdaten werden an den Problemnamen angehängt, wobei die Betriebsdaten in Klammern gesetzt werden. Weitere Informationen finden Sie in der Beschreibung des Feldes *Operational data* in der Auslöser-Konfiguration.
Show symptoms	Markieren Sie das Kontrollkästchen, um als Symptome klassifizierte Probleme in einer eigenen Zeile anzuzeigen.
Show suppressed problems	Markieren Sie das Kontrollkästchen, um Probleme anzuzeigen, die andernfalls aufgrund von Host-Wartung oder einzelner Problemunterdrückung unterdrückt (nicht angezeigt) würden.
Acknowledgement status	Filtern Sie, um alle Probleme, nur nicht bestätigte Probleme oder nur bestätigte Probleme anzuzeigen. Markieren Sie zusätzlich das Kontrollkästchen, um Probleme herauszufiltern, die jemals von Ihnen bestätigt wurden.
Sort entries by	Sortieren Sie Einträge nach: **Time** (absteigend oder aufsteigend); **Severity** (absteigend oder aufsteigend); **Problem name** (absteigend oder aufsteigend); **Host** (absteigend oder aufsteigend). Das Sortieren nach **Host** (absteigend oder aufsteigend) ist nicht verfügbar, wenn das Widget auf einem Vorlage-Dashboard konfiguriert wird.
Show timeline	Markieren Sie das Kontrollkästchen, um eine visuelle Zeitachse anzuzeigen.
Highlight whole row	Markieren Sie das Kontrollkästchen, um die gesamte Zeile für ungelöste Probleme hervorzuheben. Für die Hervorhebung wird die Farbe des Problem-Schweregrads verwendet. *Highlight whole row* ist in den Designs mit hohem Kontrast nicht verfügbar.
Show lines	Geben Sie die Anzahl der anzuzeigenden Problemzeilen an.

Verwendung des Widgets



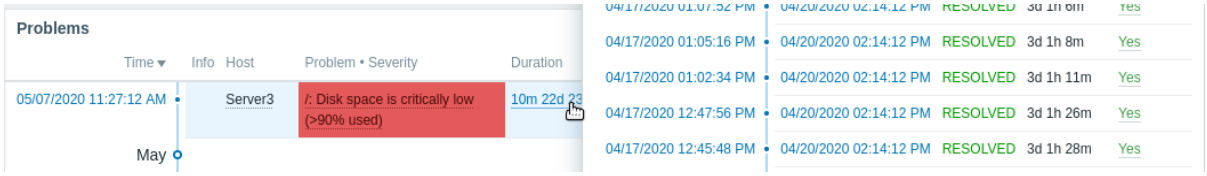
Das Problems-Widget bietet schnellen Zugriff auf zusätzliche Informationen:

- Klicken Sie auf das Problemdatum (in der Spalte *Time*) oder das Wiederherstellungsdatum (in der Spalte *Recovery Time*), um Ereignisdetails anzuzeigen.

- Wenn das Feld *Info* nicht leer ist, bewegen Sie den Mauszeiger über das angezeigte Symbol, um zusätzliche Details anzuzeigen.
- Klicken Sie auf den Hostnamen, um das Host-Menü zu öffnen.
- Klicken Sie auf den Problemnamen, um das Ereignis-Menü zu öffnen.
- Bewegen Sie den Mauszeiger über die Problemdauer oder klicken Sie darauf, um das Problemereignis-Popup anzuzeigen.
- Durch Klicken auf *Update* wird ein Fenster Problem aktualisieren geöffnet.
- Bewegen Sie den Mauszeiger über das graue Pfeilsymbol in der Spalte *Actions* oder klicken Sie darauf, um die Liste der ausgeführten Aktionen anzuzeigen.

Problemereignis-Popup

Das Problemereignis-Popup enthält die Liste der Problemereignisse für diesen Auslöser und, sofern definiert, die Auslöserbeschreibung sowie eine anklickbare URL.



So rufen Sie das Problemereignis-Popup auf:

- Bewegen Sie den Mauszeiger über die Problemdauer in der Spalte *Duration* des *Problems*-Widgets. Das Popup verschwindet, sobald Sie den Mauszeiger von der Dauer wegbewegen.
- Klicken Sie auf die Dauer in der Spalte *Duration* des *Problems*-Widgets. Das Popup verschwindet nur, wenn Sie erneut auf die Dauer klicken.

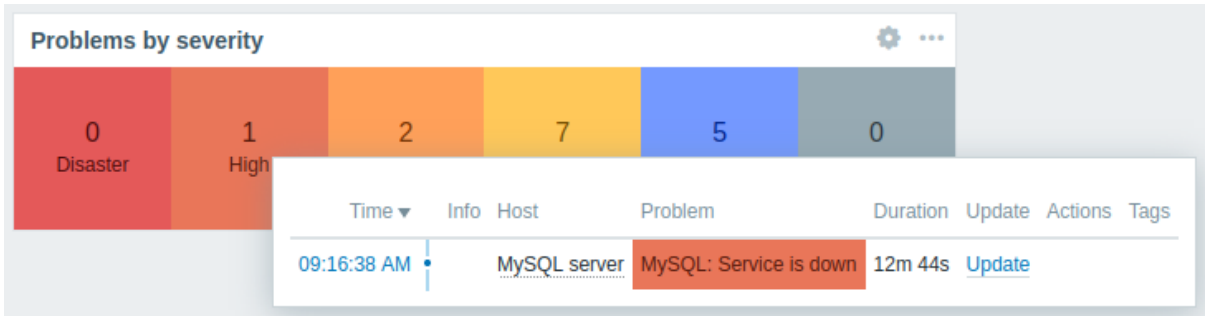
Probleme nach Schweregrad

Übersicht

Das Widget *Probleme nach Schweregrad* zeigt die Anzahl der nach Schweregrad gruppierten Ursachenprobleme an. Es ermöglicht das Filtern von Problemen nach verschiedenen Parametern (Host-Gruppen, Hosts, Problemnamen usw.).

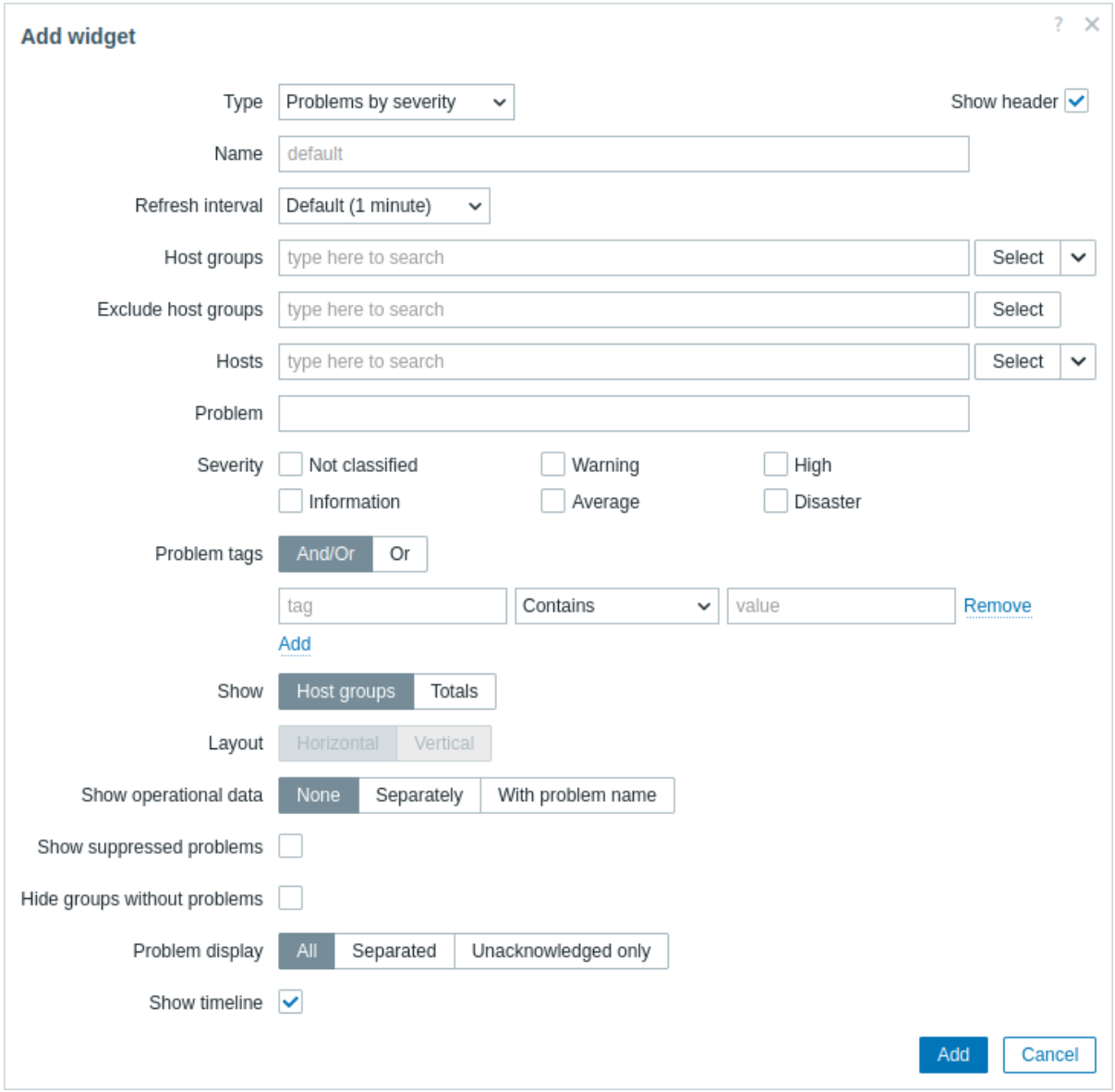


Wenn Sie den Mauszeiger über eine Problemanzahl bewegen, wird die Liste der zugehörigen Probleme angezeigt:



Konfiguration

Um zu konfigurieren, wählen Sie *Problems by severity* als Typ:



Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

Host groups	Wählen Sie Host groups aus, die im Widget angezeigt werden sollen. Alternativ können Sie ein kompatibles Widget als Datenquelle für Host groups auswählen. Dieses Feld unterstützt Auto-Vervollständigung. Wenn Sie mit der Eingabe des Namens einer Gruppe beginnen, wird eine Dropdown-Liste mit passenden Gruppen angezeigt. Wenn Sie eine übergeordnete Host group angeben, werden implizit alle verschachtelten Host groups ausgewählt. Host-Daten aus diesen Host groups werden im Widget angezeigt; wenn keine Host groups angegeben werden, werden alle Host groups angezeigt. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlage-Dashboard konfiguriert wird.
Exclude host groups	Wählen Sie Host groups aus, die im Widget ausgeblendet werden sollen. Dieses Feld unterstützt Auto-Vervollständigung. Wenn Sie mit der Eingabe des Namens einer Gruppe beginnen, wird eine Dropdown-Liste mit passenden Gruppen angezeigt. Wenn Sie eine übergeordnete Host group angeben, werden implizit alle verschachtelten Host groups ausgewählt. Host-Daten aus diesen Host groups werden nicht angezeigt. Beispielsweise können sich die Hosts 001, 002, 003 in Gruppe A und die Hosts 002, 003 zusätzlich in Gruppe B befinden. Wenn wir gleichzeitig Gruppe A *anzeigen* und Gruppe B *ausschließen*, werden im Dashboard nur Daten von Host 001 angezeigt. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlage-Dashboard konfiguriert wird.

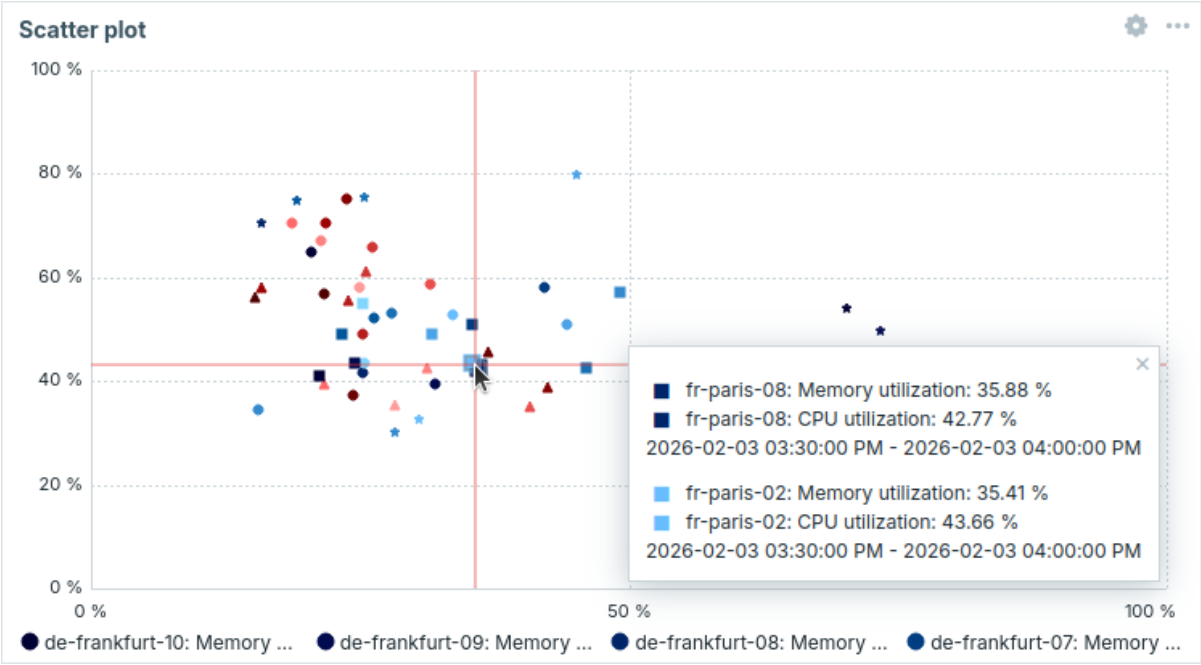
Hosts	Wählen Sie Hosts aus, die im Widget angezeigt werden sollen. Alternativ können Sie ein kompatibles Widget oder das Dashboard als Datenquelle für Hosts auswählen. Dieses Feld unterstützt Auto-Vervollständigung. Wenn Sie mit der Eingabe des Namens eines Hosts beginnen, wird eine Dropdown-Liste mit passenden Hosts angezeigt. Wenn keine Hosts angegeben werden, werden alle Hosts angezeigt. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlage-Dashboard konfiguriert wird.
Problem	Sie können die Anzahl der angezeigten Problem-Hosts nach dem Problemnamen begrenzen. Wenn Sie hier eine Zeichenfolge eingeben, werden nur die Hosts mit Problemen angezeigt, deren Name die eingegebene Zeichenfolge enthält. Makros werden nicht erweitert.
Severity	Markieren Sie Problem-Schweregrade, um die im Widget angezeigten Probleme zu filtern. Wenn keine Schweregrade markiert sind, werden alle Probleme angezeigt.
Problem tags	Geben Sie Problem-Tags an, um die Anzahl der im Widget angezeigten Probleme zu begrenzen. Es ist möglich, bestimmte Tags und Tag-Werte sowohl einzuschließen als auch auszuschließen. Es können mehrere Bedingungen festgelegt werden. Die Übereinstimmung des Tag-Namens ist immer groß-/kleinschreibungssensitiv. Für jede Bedingung stehen mehrere Operatoren zur Verfügung: **Exists** - die angegebenen Tag-Namen einschließen; **Equals** - die angegebenen Tag-Namen und Werte einschließen (groß-/kleinschreibungssensitiv); **Contains** - die angegebenen Tag-Namen einschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv); **Does not exist** - die angegebenen Tag-Namen ausschließen; **Does not equal** - die angegebenen Tag-Namen und Werte ausschließen (groß-/kleinschreibungssensitiv); **Does not contain** - die angegebenen Tag-Namen ausschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv). Für Bedingungen gibt es zwei Berechnungstypen: **And/Or** - alle Bedingungen müssen erfüllt sein, Bedingungen mit demselben Tag-Namen werden durch die *Or*-Bedingung gruppiert; **Or** - es genügt, wenn eine Bedingung erfüllt ist.
Show	Wählen Sie die Anzeigeoption: **Host groups** - Probleme pro Host group anzeigen; **Totals** - eine Gesamtzahl der Probleme für alle ausgewählten Host groups in farbigen Blöcken anzeigen, entsprechend dem Schweregrad des Problems. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlage-Dashboard konfiguriert wird; in diesem Fall wird nur eine Gesamtzahl der Probleme angezeigt.
Layout	Wählen Sie die Layout-Option: **Horizontal** - farbige Blöcke mit Gesamtzahlen werden horizontal angezeigt; **Vertical** - farbige Blöcke mit Gesamtzahlen werden vertikal angezeigt. Dieser Parameter ist verfügbar, wenn *Show* auf ”Totals” gesetzt ist.
Show operational data	Operative Daten anzeigen als: **None** - operative Daten werden nicht angezeigt **Separately** - operative Daten werden als separate Zeile angezeigt **With problem name** - operative Daten werden zusammen mit dem Problemnamen angezeigt. Weitere Informationen finden Sie in der Beschreibung des Feldes *Operational data* in der Trigger-Konfiguration.
Show suppressed problems	Aktivieren Sie das Kontrollkästchen, um Probleme anzuzeigen, die andernfalls aufgrund von Host-Wartung unterdrückt (nicht angezeigt) würden.
Hide groups without problems	Aktivieren Sie die Option *Hide groups without problems*, um Daten von Host groups ohne Probleme im Widget auszublenden. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlage-Dashboard konfiguriert wird.
Problem display	Problemanzahl anzeigen als: **All** - die vollständige Problemanzahl wird angezeigt; **Separated** - die Anzahl nicht bestätigter Probleme wird getrennt von der Gesamtzahl der Probleme angezeigt; **Unacknowledged only** - es wird nur die Anzahl nicht bestätigter Probleme angezeigt.

Show timeline	Aktivieren Sie das Kontrollkästchen, um eine visuelle Zeitleiste anzuzeigen.

Streudiagramm

Übersicht

Das Widget *Streudiagramm* zeigt die Beziehung zwischen zwei Metriken, indem einzelne Datenpunkte entlang einer X- und Y-Achse dargestellt werden. Dadurch lassen sich Muster, Cluster, Korrelationen und Ausreißer im Datensatz erkennen.



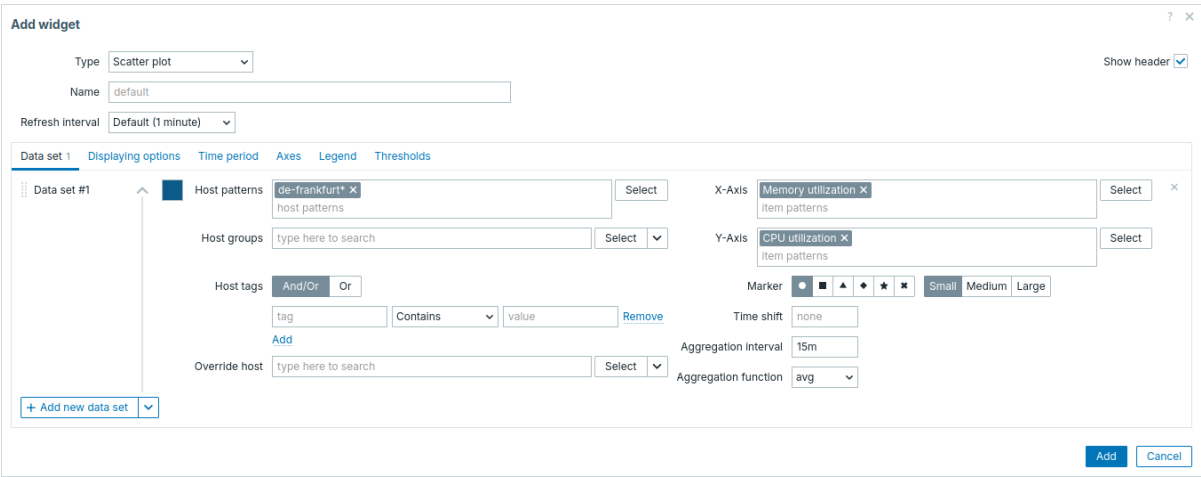
Wenn Sie den Mauszeiger über einen Datenpunkt bewegen, wird ein Tooltip mit einer Liste der Datenpunkte aus diesem Datenpunkt sowie aus anderen Punkten innerhalb eines Radius von 6 px angezeigt. Die Sortierung erfolgt zuerst nach den Werten der X-Achse des Streudiagramms in aufsteigender Reihenfolge und anschließend nach den Y-Achsenwerten in absteigender Reihenfolge.

Durch die Auswahl eines Datenpunkts im Tooltip werden dessen Daten an andere Widgets übertragen.

Die im Widget *Streudiagramm* angezeigten Informationen können als PNG-Bild heruntergeladen werden. Wählen Sie dazu die Option *Bild herunterladen* im Menü *Aktionen* ... des Widgets, das angezeigt wird, wenn Sie den Mauszeiger über das Widget bewegen. Abhängig von Ihren Browsereinstellungen wird das Bild entweder automatisch im Ordner `Downloads` gespeichert oder ein Dialogfeld *Speichern* geöffnet, in dem Sie den Speicherort auswählen können.

Konfiguration

Wählen Sie zum Konfigurieren *Scatter plot* als Typ aus:



Datensatz

Auf der Registerkarte **Datensatz** können Sie Daten für das Streudiagramm auswählen, indem Sie Datensätze hinzufügen. Es können zwei Arten von Datensätzen hinzugefügt werden:

- *Datenpunktmuster* – Hosts mit übereinstimmenden Datenpunkten werden als Datenpunkte angezeigt. Sie können eine einzelne Grundfarbe auswählen oder eine Palettenzeile wählen, um jedem Host mit den übereinstimmenden Datenpunkten eine eigene Farbe zuzuweisen.
- *Datenpunktliste* – Ausgewählte Datenpunkte werden als Datenpunkte angezeigt, wobei jeder Punkt ein Paar von Datenpunkten (X und Y) darstellt. Sie können die Farbe jedes Datensatzes einzeln über die Farbauswahl festlegen.

Datenpunkte werden nur für Host-/Datenpunktpaare erstellt, die auf beiden Achsen mindestens einen Wert haben.

Standardmäßig wird ein Datensatz vom Typ *Datenpunktmuster* hinzugefügt.

Datensatz	Für den Datensatz **Datenpunktmuster**: Wählen Sie Host- und Datenpunktmuster aus oder geben Sie diese ein. Die Daten von Datenpunkten, die diesen Mustern entsprechen, werden im Streudiagramm angezeigt. Beachten Sie, dass `memory_limit` in der Konfigurationsdatei `php.ini` erhöht werden sollte, wenn die Anzahl der Datenpunkte 8.000 überschreitet. Das empfohlene Speicherlimit für 20.000 Datenpunkte beträgt 512 MB. Für die Auswahl können Platzhaltermuster verwendet werden (z. B. gibt ∗ Ergebnisse zurück, die null oder mehr Zeichen entsprechen). Um ein Platzhaltermuster anzugeben, geben Sie die Zeichenfolge manuell ein und drücken Sie die Eingabetaste. Das Platzhaltersymbol wird immer interpretiert. Daher ist es beispielsweise nicht möglich, einen Datenpunkt namens *Datenpunkt** einzeln hinzuzufügen, wenn andere übereinstimmende Datenpunkte vorhanden sind (z. B. *Datenpunkt2*, *Datenpunkt3*). Die Angabe von Host- und Datenpunktmustern ist für Datensätze vom Typ „Datenpunktmuster“ erforderlich. Siehe auch: Details zur Datensatzkonfiguration. Für den Datensatz **Datenpunktliste**: Wählen Sie Datenpunkte für das Streudiagramm aus, indem Sie auf die Schaltfläche *Datenpunkt hinzufügen* klicken. Sie können auch kompatible Widgets als Datenquelle für Datenpunkte auswählen, indem Sie auf die Schaltfläche *Widget hinzufügen* klicken. Die Angabe von Datenpunkten oder Widgets ist für Datensätze vom Typ „Datenpunktliste“ erforderlich. Siehe auch: Details zur Datensatzkonfiguration. Beachten Sie, dass nur numerische Datenpunkttypen zulässig sind. Bei der Konfiguration des Widgets auf einem Vorlagen-Dashboard ist der Parameter zur Angabe von Hostmustern nicht verfügbar. Der Parameter zur Angabe einer Datenpunktliste ermöglicht nur die Auswahl der auf der Vorlage konfigurierten Datenpunkte.
Hostgruppen	Wählen Sie Hostgruppen aus. Alternativ können Sie ein kompatibles Widget als Datenquelle für Hostgruppen auswählen. Dieses Feld unterstützt die automatische Vervollständigung. Wenn Sie beginnen, den Namen einer Gruppe einzugeben, wird eine Dropdown-Liste mit übereinstimmenden Gruppen angezeigt. Durch die Angabe einer übergeordneten Hostgruppe werden implizit alle verschachtelten Hostgruppen ausgewählt. Hostdaten aus diesen Hostgruppen werden im Widget angezeigt. Wenn keine Hostgruppen eingegeben werden, werden alle Hostgruppen angezeigt. Dieser Parameter ist nur für Datensätze vom Typ *Datenpunktmuster* verfügbar und nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.

Host-Tags	Geben Sie Tags an, um die im Widget angezeigten Hosts zu filtern. Sie können bestimmte Tags und Tagwerte sowohl einschließen als auch ausschließen. Es können mehrere Bedingungen festgelegt werden. Beim Abgleich von Tagnamen wird immer zwischen Groß- und Kleinschreibung unterschieden. Für jede Bedingung stehen mehrere Operatoren zur Verfügung: **Existiert** – die angegebenen Tagnamen einschließen; **Ist gleich** – die angegebenen Tagnamen und Tagwerte einschließen (Groß-/Kleinschreibung wird berücksichtigt); **Enthält** – die angegebenen Tagnamen einschließen, deren Tagwerte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich ohne Berücksichtigung der Groß-/Kleinschreibung); **Existiert nicht** – die angegebenen Tagnamen ausschließen; **Ist nicht gleich** – die angegebenen Tagnamen und Tagwerte ausschließen (Groß-/Kleinschreibung wird berücksichtigt); **Enthält nicht** – die angegebenen Tagnamen ausschließen, deren Tagwerte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich ohne Berücksichtigung der Groß-/Kleinschreibung). Für Bedingungen gibt es zwei Berechnungstypen: **Und/Oder** – alle Bedingungen müssen erfüllt sein; Bedingungen mit demselben Tagnamen werden durch die Oder-Bedingung gruppiert; **Oder** – es reicht aus, wenn eine Bedingung erfüllt ist. Dieser Parameter ist nur für Datensätze vom Typ *Datenpunktmuster* verfügbar und nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Host überschreiben	Wählen Sie ein kompatibles Widget oder die Hostauswahl des Dashboards als Datenquelle für Hosts aus. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Markierung	Wählen Sie Typ und Größe der Markierung für Hostdatenpunkte im Datensatz aus.
Zeitverschiebung	Geben Sie bei Bedarf eine Zeitverschiebung an. In diesem Feld können Sie Zeitsuffixe verwenden. Negative Werte sind zulässig.
Aggregationsfunktion	Geben Sie an, welche Aggregationsfunktion für jeden Datenpunkt im Datensatz innerhalb des ausgewählten *Aggregationsintervalls* verwendet werden soll: **min** – den kleinsten Wert anzeigen; **max** – den größten Wert anzeigen; **avg** – den Durchschnittswert anzeigen; **sum** – die Summe der Werte anzeigen; **count** – die Anzahl der Werte anzeigen (wenn auf beiden Achsen mindestens ein Wert vorhanden ist); **first** – den ersten Wert anzeigen; **last** – den letzten Wert anzeigen. Durch die Aggregation kann für das angegebene Intervall (5 Minuten, eine Stunde, ein Tag) ein aggregierter Wert anstelle aller Werte angezeigt werden. Für *min*, *max*, *avg* und *sum* können nur numerische Daten angezeigt werden. Für *count* werden nicht numerische Daten in numerische Daten umgewandelt.

Aggregationsintervall	Geben Sie das Intervall für die Aggregation von Werten an. In diesem Feld können Sie Zeitsuffixe verwenden. Ein numerischer Wert ohne Suffix wird als Sekunden interpretiert. Beachten Sie, dass empfohlen wird, ein Aggregationsintervall zu verwenden, das ein Vielfaches von 1 Stunde ist (z. B. 3600, 60m, 1h, 3h usw.), wenn das Widget so konfiguriert ist, dass historische Daten auf Grundlage von Trends angezeigt werden (*Auswahl der Verlaufsdaten* ist auf *Trends* oder *Automatisch* gesetzt). Trends speichern stündlich aggregierte Werte. Daher kann die Verwendung eines Aggregationsintervalls, das kein Vielfaches von 1 Stunde ist (z. B. 100s, 7min, 15min, 90min usw.), zu schwer interpretierbaren Ergebnissen führen. Die Angabe eines Aggregationsintervalls ist erforderlich. Das Aggregationsintervall bestimmt das Zeitintervall und die Koordinaten eines Datenpunkts. Wenn beispielsweise die Datenpunkte der X-Achse 4 Werte und die Datenpunkte der Y-Achse 1 Wert haben, entsprechen die Koordinaten der Aggregation von 4 Werten auf der X-Achse und 1 Wert auf der Y-Achse innerhalb des angegebenen Intervalls.

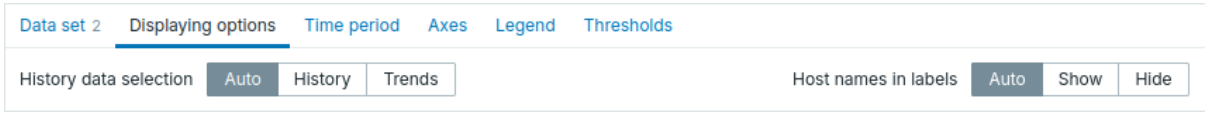
Details zur Datenmengenkonfiguration

Vorhandene Datenmengen werden in einer Liste angezeigt. Sie können diese Datenmengen neu anordnen, aus-/einklappen, Farben ändern und klonen.

Weitere Informationen finden Sie unter Details zur Datenmengenkonfiguration im Widget *Graph*. Diese Details gelten auch für das Widget *Scatter plot*.

Anzeigeoptionen

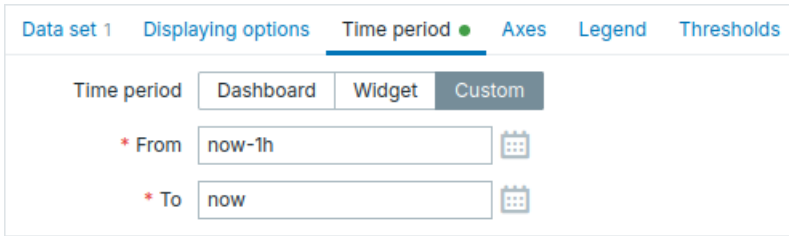
Auf der Registerkarte **Anzeigeoptionen** können Sie die Auswahl der Verlaufsdaten festlegen:



Auswahl der Verlaufsdaten	Legen Sie die Quelle der Daten für das Streudiagramm fest: **Auto** – Daten werden gemäß dem Algorithmus des klassischen Graphen bezogen (Standard). **Verlauf** – Daten aus dem Verlauf. **Trends** – Daten aus Trends.
Host-Namen in Beschriftungen	Wählen Sie aus, ob Host-Namen in der Legende des Graphen und im Tooltip angezeigt werden sollen, der beim Bewegen des Mauszeigers über die Graphdaten erscheint: **Auto** – Host-Namen werden nur angezeigt, wenn in den Datensätzen mehr als ein Host vorhanden ist (Standard). **Anzeigen** – Host-Namen werden angezeigt. **Ausblenden** – Host-Namen werden ausgeblendet. Dieser Parameter ist beim Konfigurieren des Widgets auf einem Vorlagen-Dashboard nicht verfügbar.

Zeitraum

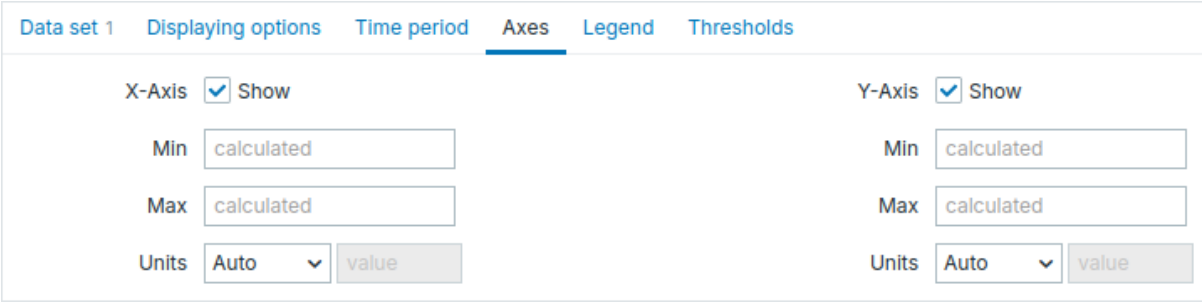
Auf der Registerkarte **Zeitraum** können Sie einen Zeitraum festlegen, für den Daten im Streudiagramm angezeigt werden:



Zeitraum	Wählen Sie die Datenquelle für den Zeitraum aus: **Dashboard** – verwendet die Zeitraumauswahl des Dashboards; **Widget** – verwendet ein kompatibles Widget (im Parameter *Widget* festgelegt); **Benutzerdefiniert** – verwendet einen benutzerdefinierten Zeitraum, der in den Parametern *Von* und *Bis* festgelegt wird; wenn dieser festgelegt ist, wird in der oberen rechten Ecke des Widgets ein Uhrsymbol angezeigt, das bei einem Mouseover die festgelegte Zeit angibt. Beachten Sie, dass kompatible Widgets unabhängig von der Konfiguration *Zeitraum* des Widgets weiterhin als Datenquelle für den Zeitraum verwendet werden können.
Widget	Geben Sie ein kompatibles Widget als Datenquelle für den Zeitraum ein oder wählen Sie es aus. Dieser Parameter ist verfügbar, wenn *Zeitraum* auf „Widget“ gesetzt ist.
Von	Geben Sie den Beginn des Zeitraums ein oder wählen Sie ihn aus. Die Syntax für relative Zeitangaben (`now`, `now/d`, `now/w-1w` usw.) wird unterstützt. Dieser Parameter ist verfügbar, wenn *Zeitraum* auf „Benutzerdefiniert“ gesetzt ist.
Bis	Geben Sie das Ende des Zeitraums ein oder wählen Sie es aus. Die Syntax für relative Zeitangaben (`now`, `now/d`, `now/w-1w` usw.) wird unterstützt. Dieser Parameter ist verfügbar, wenn *Zeitraum* auf „Benutzerdefiniert“ gesetzt ist.

Achsen

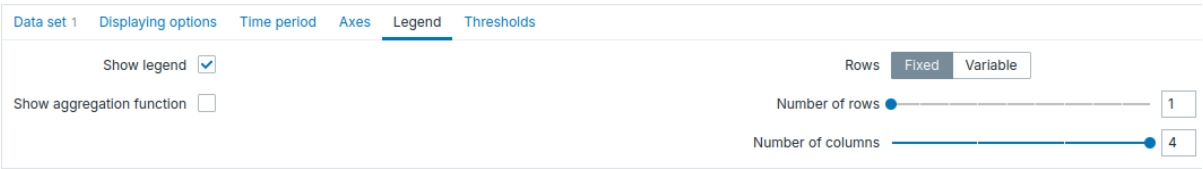
Auf der Registerkarte **Achsen** können Sie festlegen, wie Achsen angezeigt werden:



X-Achse/Y-Achse	Deaktivieren Sie dieses Kontrollkästchen, um die X-Achse/Y-Achse auszublenden (standardmäßig aktiviert).
Min	Legen Sie den Mindestwert der entsprechenden Achse fest.
Max	Legen Sie den Höchstwert der entsprechenden Achse fest.
Einheiten	Wählen Sie die Einheit für die Achsenwerte des Streudiagramms aus der Dropdown-Liste aus: **Auto** – Achsenwerte werden mit der Einheit des ersten Datenpunkts im Datensatz angezeigt; **Statisch** – Achsenwerte werden mit der im Eingabefeld *Wert* angegebenen Einheit angezeigt. Wenn das Feld leer bleibt, werden nur numerische Werte angezeigt.

Legende

Auf der Registerkarte **Legende** können Sie die Legende des Streudiagramms anpassen:

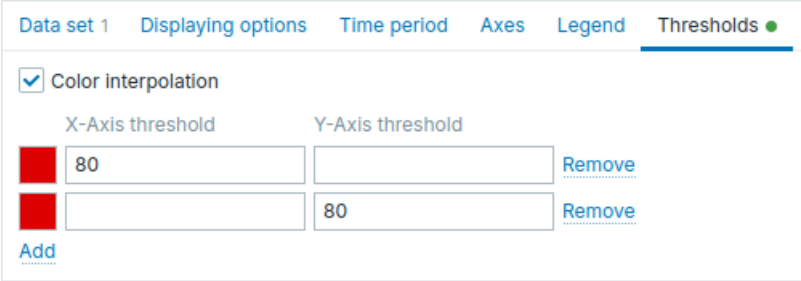


Legende anzeigen	Deaktivieren Sie dieses Kontrollkästchen, um die Legende im Streudiagramm auszublenden (standardmäßig aktiviert).
Aggregationsfunktion anzeigen	Aktivieren Sie dieses Kontrollkästchen, um die Aggregationsfunktion in der Legende anzuzeigen.
Zeilen	Wählen Sie den Anzeigemodus für die Zeilen der Legende aus: **Fest** – Die Anzahl der angezeigten Zeilen wird durch den Wert des Parameters *Anzahl der Zeilen* bestimmt; **Variabel** – Die Anzahl der angezeigten Zeilen wird durch die Anzahl der konfigurierten Datenpunkte bestimmt, darf jedoch den Wert des Parameters *Maximale Anzahl der Zeilen* nicht überschreiten.

Anzahl der Zeilen/ Maximale Anzahl der Zeilen	Wenn *Zeilen* auf „Fest“ gesetzt ist, legen Sie die Anzahl der anzuzeigenden Legendenzeilen fest (1–10). Wenn *Zeilen* auf „Variabel“ gesetzt ist, legen Sie die maximale Anzahl der anzuzeigenden Legendenzeilen fest (1–10).
Anzahl der Spalten	Legen Sie die Anzahl der anzuzeigenden Legendenspalten fest (1–4). Dieser Parameter ist verfügbar, wenn *Min./Durchschn./Max.* nicht aktiviert ist.

Schwellenwerte

Auf der Registerkarte **Schwellenwerte** können Sie Schwellenwerte konfigurieren, die die Farben der Host-Markierungen bestimmen:

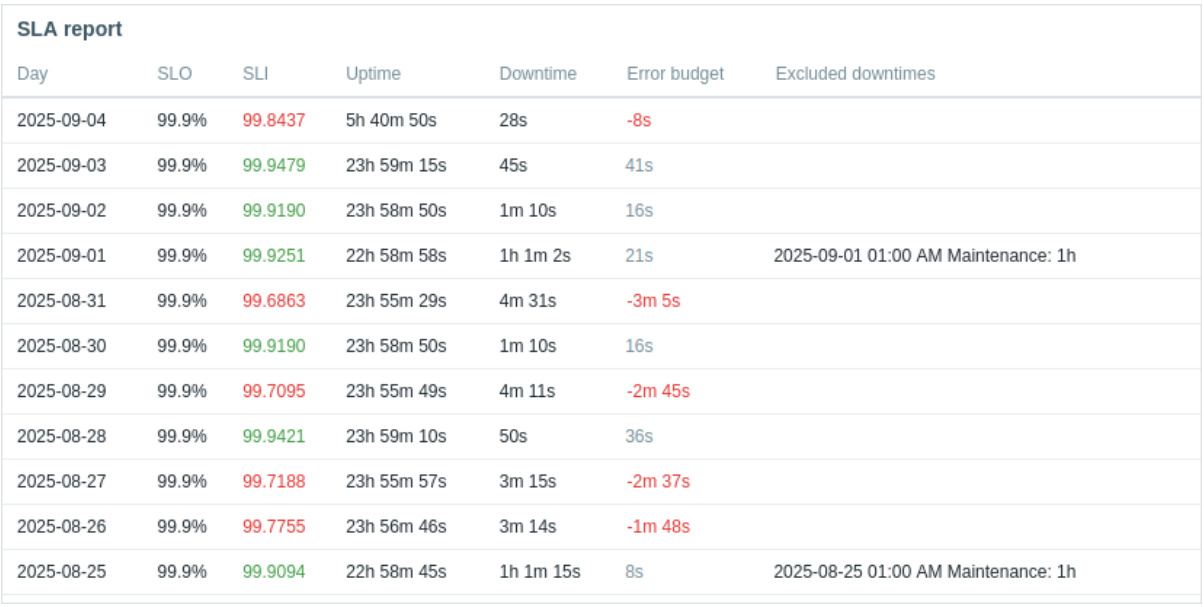


Farbinterpolation	Aktivieren Sie das Kontrollkästchen, um einen fließenden Übergang zwischen den Schwellenwertfarben für Host-Markierungen zu ermöglichen. Dieser Parameter gilt, wenn zwei oder mehr Schwellenwerte für die X-Achse, die Y-Achse oder gleichzeitig für beide Achsen festgelegt sind.
Schwellenwert für X-Achse/Y-Achse	Klicken Sie auf *Hinzufügen*, um einen Schwellenwert hinzuzufügen, wählen Sie im Farbwähler eine Schwellenwertfarbe aus und geben Sie einen numerischen Wert an. Schwellenwerte können auf eine einzelne Achse, separat auf beide Achsen oder gleichzeitig auf beide Achsen angewendet werden. Beispielsweise färbt ein einzelner Schwellenwert mit X=80 und Y=80 die Markierungen, wenn beide Bedingungen erfüllt sind. Die Liste der Schwellenwerte wird beim Speichern in aufsteigender Reihenfolge sortiert. Suffixe (z. B. „1d“, „2w“, „4K“, „8G“) werden unterstützt. Wertzuordnungen werden unterstützt.

SLA-Bericht

Übersicht

Das Widget *SLA-Bericht* zeigt SLA-Berichte an und hilft Ihnen dabei, die Service-Performance im Vergleich zu definierten Zielen zu überwachen.

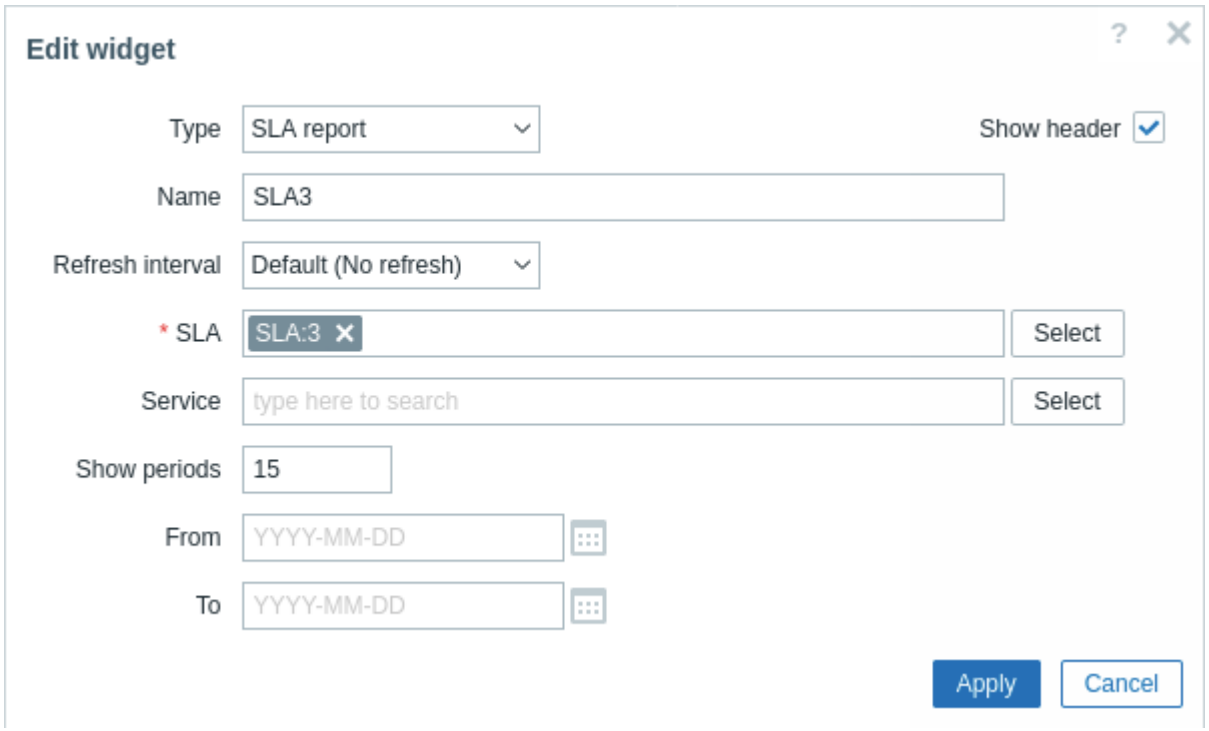

SLA report

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

Es zeigt dieselben Daten an wie *Services* > *SLA-Bericht*.

Konfiguration

Wählen Sie zur Konfiguration *SLA-Bericht* als Typ aus:



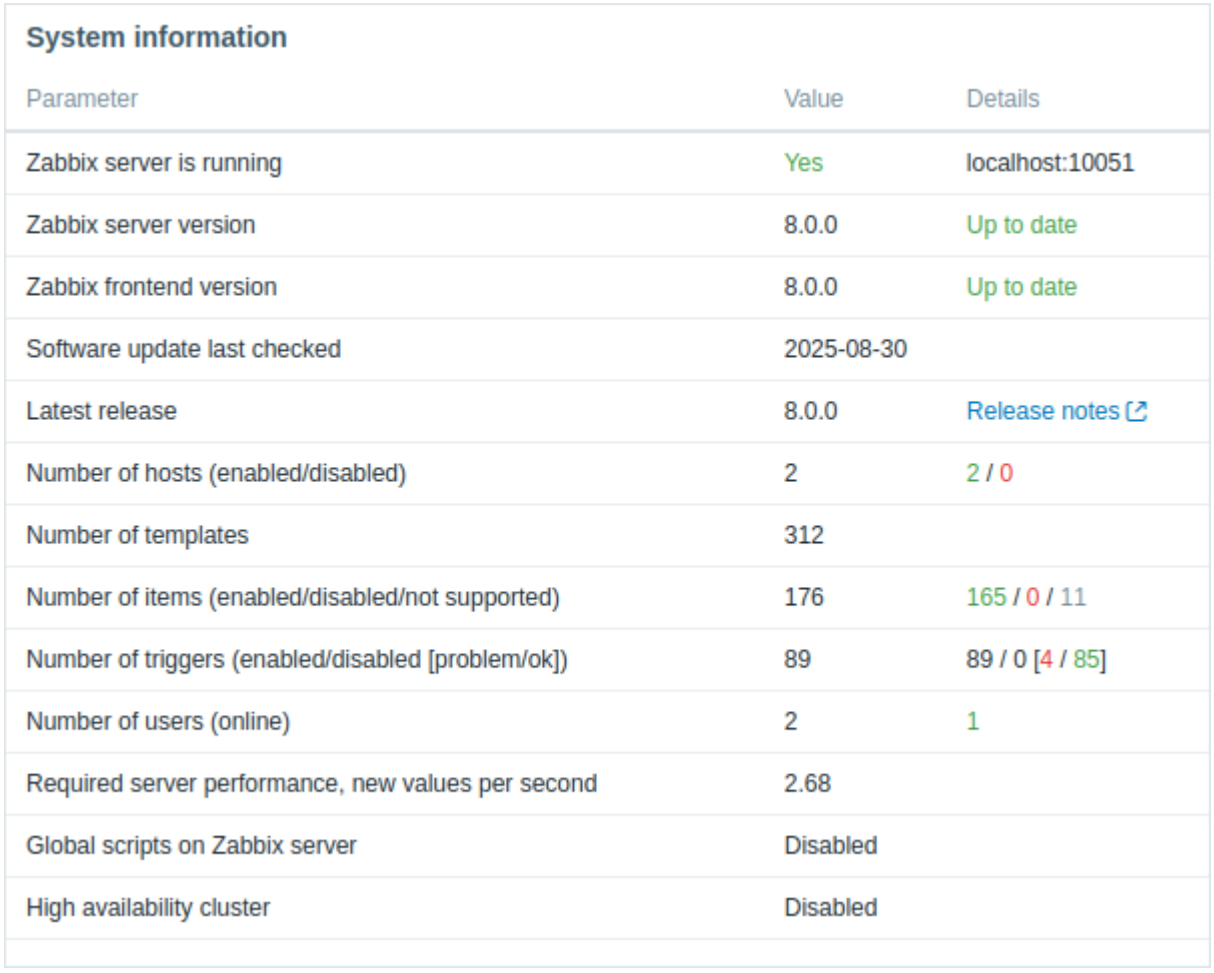
Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

SLA	Wählen Sie das SLA für den Bericht aus.
Service	Wählen Sie den Service für den Bericht aus.
Anzuzeigende Zeiträume	Legen Sie fest, wie viele Zeiträume im Widget angezeigt werden (standardmäßig 20, maximal 100).
Von	Wählen Sie das Anfangsdatum für den Bericht aus. Relative Datumsangaben werden unterstützt: `now`, `now/d`, `now/w-1w` usw.; unterstützte Datumsmodifikatoren: d, w, M, y.
Bis	Wählen Sie das Enddatum für den Bericht aus. Relative Datumsangaben werden unterstützt: `now`, `now/d`, `now/w-1w` usw.; unterstützte Datumsmodifikatoren: d, w, M, y.

Systeminformationen

Übersicht

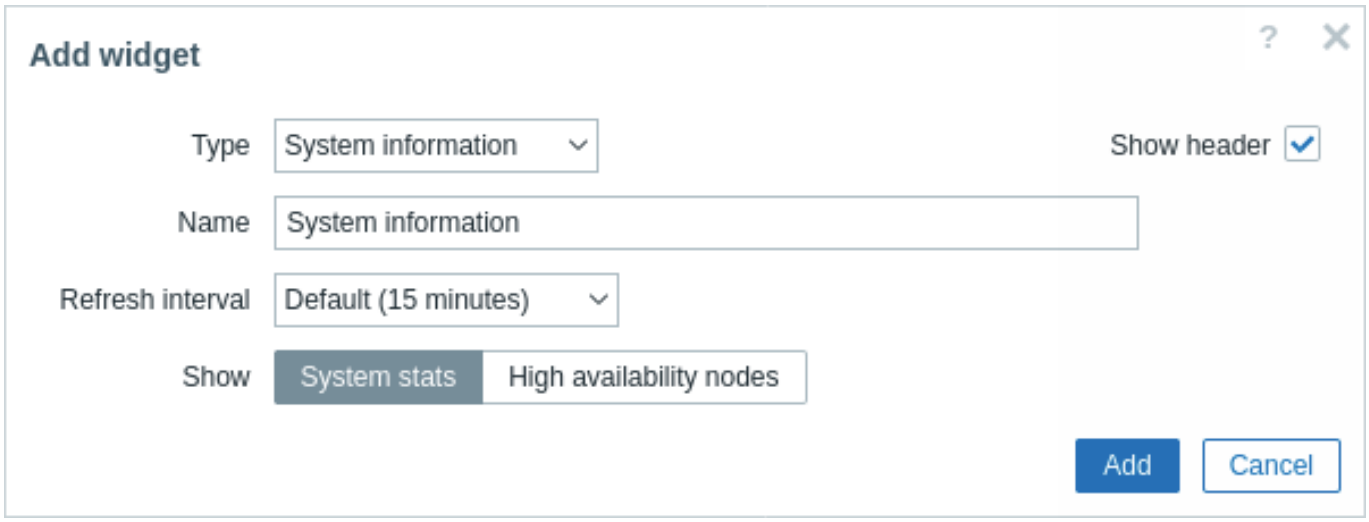
Das Widget *System information* zeigt eine Zusammenfassung wichtiger Zabbix-Server- und Systemdaten sowie, wenn Hochverfügbarkeit aktiviert ist, Details zu den Hochverfügbarkeitsknoten an.



Es zeigt dieselben Daten wie *Reports* > *System information*, mit Ausnahme von *Zabbix server ID*.

Konfiguration

Wählen Sie zur Konfiguration *System information* als Typ aus:



Zusätzlich zu den allgemeinen Parametern, die für alle Widgets gelten, können Sie die folgenden spezifischen Optionen festlegen:

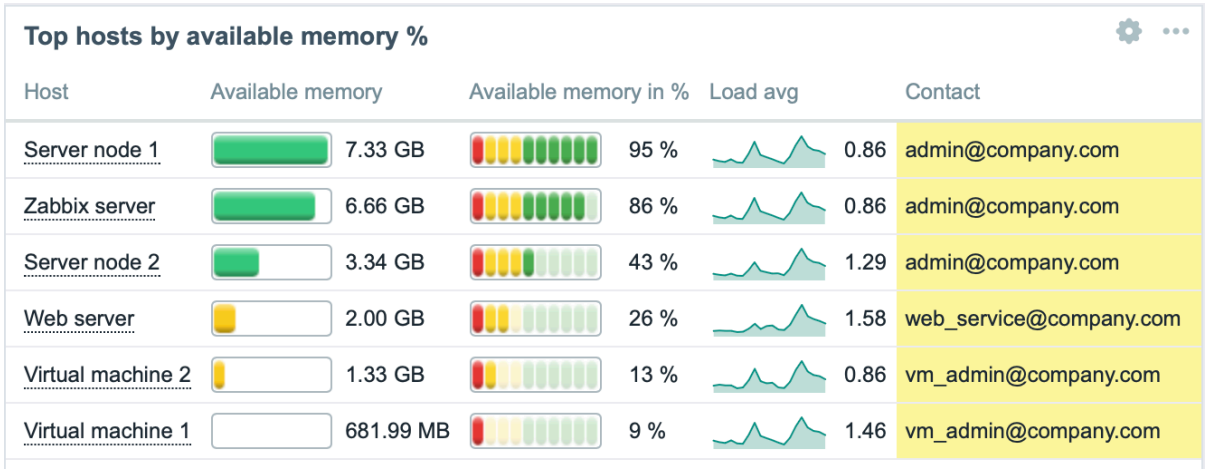
Show	Wählen Sie aus, was angezeigt werden soll: **System stats** – Zeigt eine Zusammenfassung wichtiger Daten des Zabbix-Servers und des Systems an; **High availability nodes** – Zeigt den Status der Hochverfügbarkeitsknoten an (wenn der Hochverfügbarkeitscluster aktiviert ist).

Show software update check details	Aktivieren Sie das Kontrollkästchen, um Details zur Prüfung auf Zabbix-Softwareupdates anzuzeigen. Diese Option ist nur verfügbar, wenn die Prüfung auf Softwareupdates in der Serverkonfiguration von Zabbix aktiviert ist und im Feld *Show* die Option „System stats“ ausgewählt wurde.

Top-Hosts

Übersicht

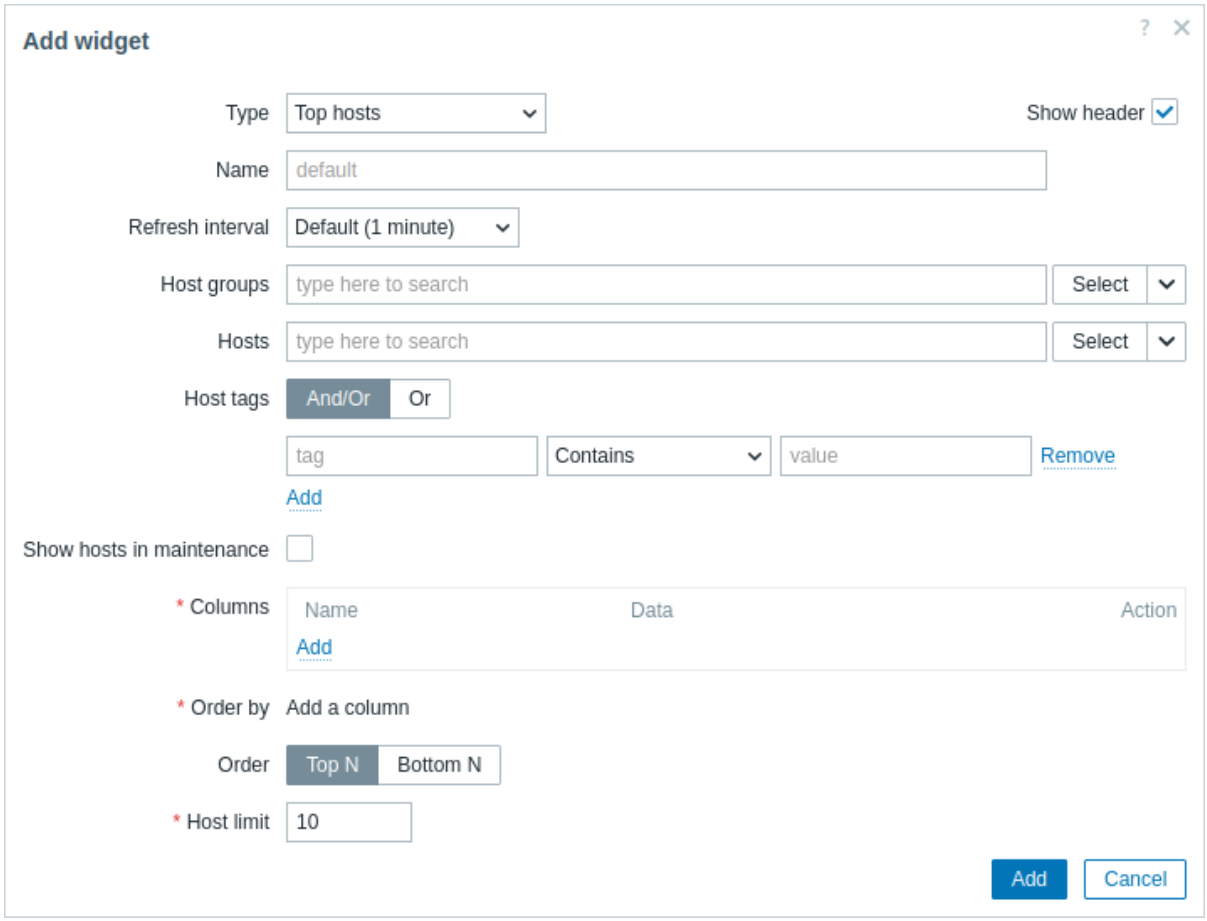
Das Widget *Top hosts* zeigt die höchsten/niedrigsten Werte für ausgewählte Datenpunkte von mehreren Hosts an.



Das Widget unterstützt die gleichzeitige Anzeige von bis zu 1000 Hosts.

Konfiguration

Wählen Sie zur Konfiguration *Top hosts* als Typ aus:



Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

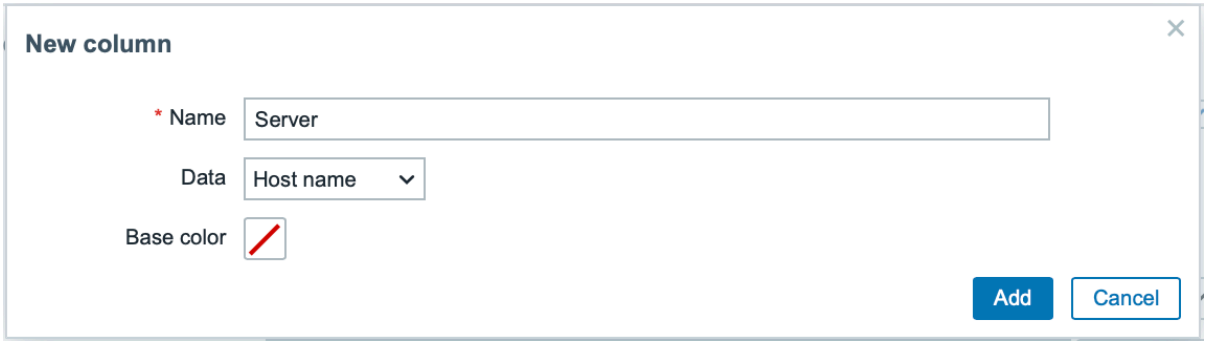
Host-Gruppen	Wählen Sie die Host-Gruppen aus, die im Widget angezeigt werden sollen. Alternativ können Sie ein kompatibles Widget als Datenquelle für Host-Gruppen auswählen. Dieses Feld unterstützt Autovervollständigung; wenn Sie beginnen, den Namen einer Gruppe einzugeben, wird eine Dropdown-Liste mit passenden Gruppen angeboten. Dieser Parameter ist nicht verfügbar, wenn Sie das Widget auf einem Vorlagen-Dashboard konfigurieren.
Hosts	Wählen Sie die Hosts aus, die im Widget angezeigt werden sollen. Alternativ können Sie ein kompatibles Widget oder das Dashboard als Datenquelle für Hosts auswählen. Dieses Feld unterstützt Autovervollständigung; wenn Sie beginnen, den Namen eines Hosts einzugeben, wird eine Dropdown-Liste mit passenden Hosts angeboten. Dieser Parameter ist nicht verfügbar, wenn Sie das Widget auf einem Vorlagen-Dashboard konfigurieren.
Host-Tags	Geben Sie Tags an, um die Anzahl der im Widget angezeigten Hosts zu begrenzen. Es ist möglich, bestimmte Tags und Tag-Werte sowohl einzuschließen als auch auszuschließen. Es können mehrere Bedingungen festgelegt werden. Der Abgleich von Tag-Namen ist immer groß-/kleinschreibungssensitiv. Für jede Bedingung stehen mehrere Operatoren zur Verfügung: **Existiert** - die angegebenen Tag-Namen einschließen; **Gleich** - die angegebenen Tag-Namen und -Werte einschließen (groß-/kleinschreibungssensitiv); **Enthält** - die angegebenen Tag-Namen einschließen, bei denen die Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv); **Existiert nicht** - die angegebenen Tag-Namen ausschließen; **Ungleich** - die angegebenen Tag-Namen und -Werte ausschließen (groß-/kleinschreibungssensitiv); **Enthält nicht** - die angegebenen Tag-Namen ausschließen, bei denen die Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv). Für Bedingungen gibt es zwei Berechnungstypen: **Und/Oder** - alle Bedingungen müssen erfüllt sein; Bedingungen mit demselben Tag-Namen werden durch die Oder-Bedingung gruppiert; **Oder** - es genügt, wenn eine Bedingung erfüllt ist. Dieser Parameter ist nicht verfügbar, wenn Sie das Widget auf einem Vorlagen-Dashboard konfigurieren.
Hosts in Wartung anzeigen	Aktivieren Sie dieses Kontrollkästchen, damit auch Hosts in Wartung angezeigt werden (in diesem Fall wird neben dem Host-Namen ein Wartungssymbol angezeigt). Standardmäßig nicht aktiviert.
Spalten	Fügen Sie Spalten mit anzuzeigenden Daten hinzu. Die Reihenfolge der Spalten bestimmt ihre Anzeige von links nach rechts. Spalten können neu angeordnet werden, indem sie über den Ziehpunkt vor dem Spaltennamen nach oben oder unten gezogen werden.
Sortieren nach	Geben Sie die Spalte aus der definierten Liste *Spalten* an, die für die Sortierung *Top N* oder *Bottom N* verwendet werden soll.
Reihenfolge	Geben Sie die Sortierung der Zeilen an: **Top N** - in absteigender Reihenfolge entsprechend dem aggregierten Wert aus *Sortieren nach*; **Bottom N** - in aufsteigender Reihenfolge entsprechend dem aggregierten Wert aus *Sortieren nach*.
Host-Limit	Anzahl der anzuzeigenden Host-Zeilen (1-1000). Dieser Parameter ist nicht verfügbar, wenn Sie das Widget auf einem Vorlagen-Dashboard konfigurieren.

Spaltenkonfiguration

Es ist möglich, Spalten mit drei Datentypen hinzuzufügen: Host-Name, Datenpunktwert oder Text. Die Liste der verfügbaren Spaltenparameter hängt vom Datentyp der Spalte und beim Typ Datenpunktwert vom Wertformat ab.

Spalte „Host name“

Die Spalte „Host name“ wird verwendet, um den Namen des Hosts anzuzeigen.

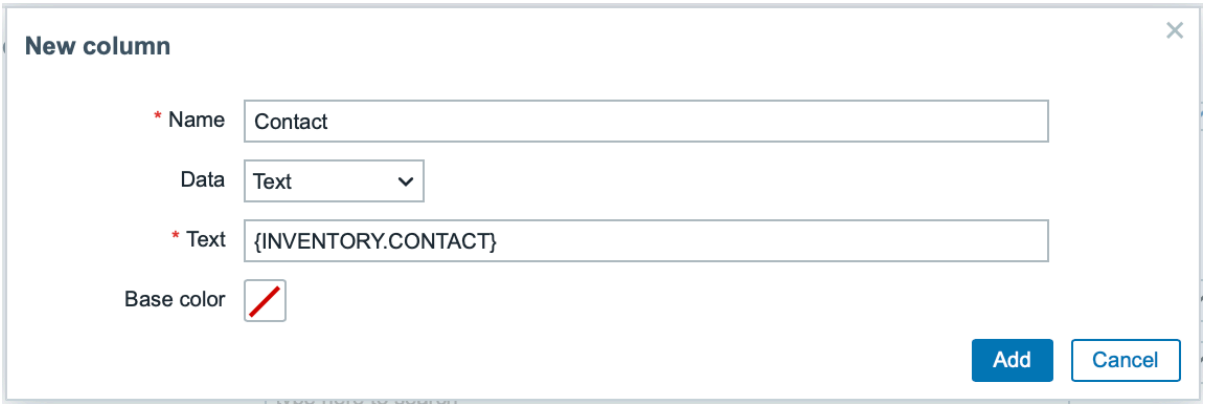


Unterstützte Parameter:

Name	Name der Spalte.
Data	Datentyp, der in der Spalte angezeigt werden soll; wählen Sie **Host name** aus.
Base color	Hintergrundfarbe der Spalte.

Textspalte

Die Textspalte wird verwendet, um eine beliebige angegebene Textzeichenfolge anzuzeigen.

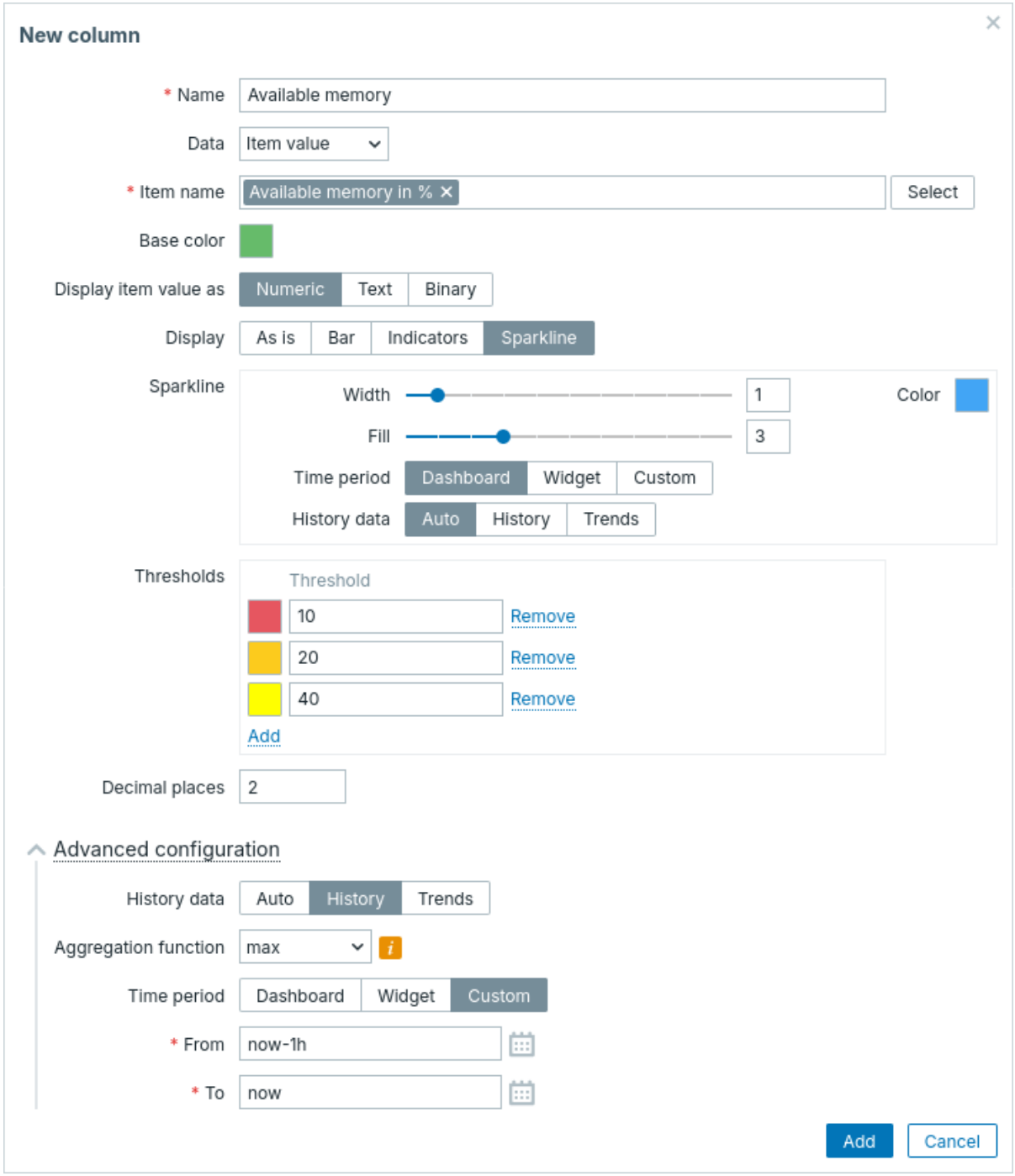


Unterstützte Parameter:

Name	Name der Spalte.
Data	Datentyp, der in der Spalte angezeigt werden soll; wählen Sie **Text** aus.
Base color	Hintergrundfarbe der Spalte.
Text	Geben Sie die anzuzeigende Zeichenfolge ein. Kann Host- und Inventar-Makros enthalten.

Spalte für Datenpunktwert

Die Spalte für den Datenpunktwert wird verwendet, um den Wert des angegebenen Datenpunkts anzuzeigen. Sie unterstützt mehrere Anzeigeformate und Optionen.



Unterstützte Parameter:

Name	Name der Spalte.
Data	Datentyp, der in der Spalte angezeigt werden soll; wählen Sie **Item value**.
Item name	Wählen Sie einen Datenpunkt aus. Sein Name wird verwendet, um Datenpunkte mit demselben Namen auf allen ausgewählten Hosts abzugleichen und anzuzeigen. Wenn ein Host mehrere solche Datenpunkte besitzt, zeigt das Widget den Datenpunkt an, dessen Schlüssel alphabetisch zuerst kommt (z. B. *CPU utilization* mit dem Schlüssel `proc.cpu.util` anstelle von `system.cpu.util`). Beim Konfigurieren des Widgets auf einem Dashboard einer Vorlage können nur auf der Vorlage konfigurierte Datenpunkte ausgewählt werden.
Base color	Hintergrundfarbe der Spalte; Füllfarbe, wenn die Daten *Item value* als Balken/Indikatoren angezeigt werden. Die Standardfarbe kann durch eine benutzerdefinierte Farbe überschrieben werden, wenn der Datenpunktwert einen der angegebenen Schwellenwerte überschreitet.
Display item value as	Format für die Anzeige des Datenpunktwerts: **Numeric**, **Text** oder **Binary**. Die ausgewählte Option bestimmt, welche zusätzlichen Parameter verfügbar sind. Eine Liste der Parameter für jedes Format finden Sie weiter unten.
Advanced configuration	

History data	Daten aus der Historie oder aus Trends verwenden: **Auto** – automatische Auswahl; **History** – Daten aus der Historie verwenden; **Trends** – Trenddaten verwenden. Dieser Parameter ist nur für numerische Datenpunktwerte verfügbar. Bei Text- und Binärwerten werden die Daten immer aus der Historie abgerufen.
Aggregation function	Geben Sie an, welche Aggregationsfunktion innerhalb des ausgewählten *Time period* verwendet werden soll: **min** – den kleinsten Wert anzeigen; **max** – den größten Wert anzeigen; **avg** – den Durchschnittswert anzeigen; **count** – die Anzahl der Werte anzeigen; **sum** – die Summe der Werte anzeigen; **first** – den ersten Wert anzeigen; **last** – den letzten Wert anzeigen. Wenn **not used** ausgewählt ist, wird der aktuellste Wert angezeigt (keine Aggregation). Durch die Aggregation kann für das angegebene Intervall (5 Minuten, eine Stunde, ein Tag) ein aggregierter Wert anstelle des aktuellsten Werts angezeigt werden. Für *min*, *max*, *avg* und *sum* können nur numerische Daten angezeigt werden. Für *count* werden nicht numerische Daten in numerische Daten umgewandelt.
Time period	Geben Sie den Zeitraum an, der für die Aggregation von Werten verwendet werden soll. Wählen Sie die Datenquelle für den Zeitraum aus: **Dashboard** – den Zeitraumauswahlschalter des Dashboards verwenden; **Widget** – ein kompatibles Widget verwenden (im Parameter *Widget* festgelegt); **Custom** – einen benutzerdefinierten Zeitraum verwenden, der in den Parametern *From* und *To* festgelegt ist; wenn dieser festgelegt ist, wird in der oberen rechten Ecke des Widgets ein Uhrsymbol angezeigt, das beim Überfahren mit der Maus die festgelegte Zeit angibt. Beachten Sie, dass kompatible Widgets unabhängig von der Konfiguration *Time period* des Widgets weiterhin als Datenquelle für den Zeitraum verwendet werden können. Dieser Parameter ist nicht verfügbar, wenn *Aggregation function* auf „not used“ gesetzt ist.
Widget	Geben Sie ein kompatibles Widget als Datenquelle für den Zeitraum ein oder wählen Sie es aus. Dieser Parameter ist verfügbar, wenn *Time period* auf „Widget“ gesetzt ist.
From	Geben Sie den Beginn des Zeitraums ein oder wählen Sie ihn aus. Die Syntax für relative Zeitangaben (`now`, `now/d`, `now/w-1w` usw.) wird unterstützt. Dieser Parameter ist verfügbar, wenn *Time period* auf „Custom“ gesetzt ist.
To	Geben Sie das Ende des Zeitraums ein oder wählen Sie es aus. Die Syntax für relative Zeitangaben (`now`, `now/d`, `now/w-1w` usw.) wird unterstützt. Dieser Parameter ist verfügbar, wenn *Time period* auf „Custom“ gesetzt ist.

Parameter für **Numeric item value**:

Display	Legen Sie fest, wie der Wert angezeigt werden soll: **As is** – als normaler Text; **Bar** – als durchgehend farbig gefüllter Balken; **Indicators** – als segmentierter, farbig gefüllter Balken; **Sparkline** – als Mini-Liniendiagramm.
Thresholds	Geben Sie Schwellenwerte an, bei denen sich die Hintergrund-/Füllfarbe ändern soll. Die Liste wird beim Speichern aufsteigend sortiert. Bei Sparklines werden Schwellenwerte nur auf den letzten Wert des Datenpunkts angewendet.
Decimal places	Geben Sie an, wie viele Dezimalstellen mit dem Wert angezeigt werden sollen.
Bar/Indicators	
Min	Minimalwert.
Max	Maximalwert.
Sparkline	
Width	Legen Sie die Stärke der Diagrammlinie fest, indem Sie den Schieberegler verwenden oder manuell einen Wert im Bereich von 0 bis 10 eingeben.
Color	Wählen Sie die Linien- und Füllfarbe aus.
Fill	Legen Sie die Transparenz der Füllfarbe fest, indem Sie den Schieberegler verwenden oder manuell einen Wert im Bereich von 0 bis 10 eingeben.

Time period	Geben Sie den Zeitraum an, dessen Werte in das Sparkline-Diagramm aufgenommen werden sollen. Wählen Sie die Datenquelle für den Zeitraum aus: **Dashboard** – den Zeitraumauswahlschalter des Dashboards verwenden **Widget** – ein kompatibles Widget verwenden (im Parameter *Widget* festgelegt); **Custom** – einen benutzerdefinierten Zeitraum verwenden, der in den Parametern *From* und *To* festgelegt ist; wenn dieser festgelegt ist, wird in der oberen rechten Ecke des Widgets ein Uhrsymbol angezeigt, das beim Überfahren mit der Maus die festgelegte Zeit angibt. Beachten Sie, dass kompatible Widgets unabhängig von der Konfiguration *Time period* des Widgets weiterhin als Datenquelle für den Zeitraum verwendet werden können.
Widget	Geben Sie ein kompatibles Widget als Datenquelle für den Zeitraum ein oder wählen Sie es aus. Dieser Parameter ist verfügbar, wenn *Time period* auf „Widget“ gesetzt ist.
From	Geben Sie den Beginn des Zeitraums ein oder wählen Sie ihn aus. Die Syntax für relative Zeitangaben (`now`, `now/d`, `now/w-1w` usw.) wird unterstützt. Dieser Parameter ist verfügbar, wenn *Time period* auf „Custom“ gesetzt ist.
To	Geben Sie das Ende des Zeitraums ein oder wählen Sie es aus. Die Syntax für relative Zeitangaben (`now`, `now/d`, `now/w-1w` usw.) wird unterstützt. Dieser Parameter ist verfügbar, wenn *Time period* auf „Custom“ gesetzt ist.
History data	Daten aus der Historie oder aus Trends verwenden: **Auto** – automatische Auswahl; **History** – Daten aus der Historie verwenden; **Trends** – Trenddaten verwenden.

Parameter für **Text item value**:

Highlights	Geben Sie die regulären Ausdrücke an, bei deren Übereinstimmung sich die Hintergrund-/Füllfarbe ändern soll.

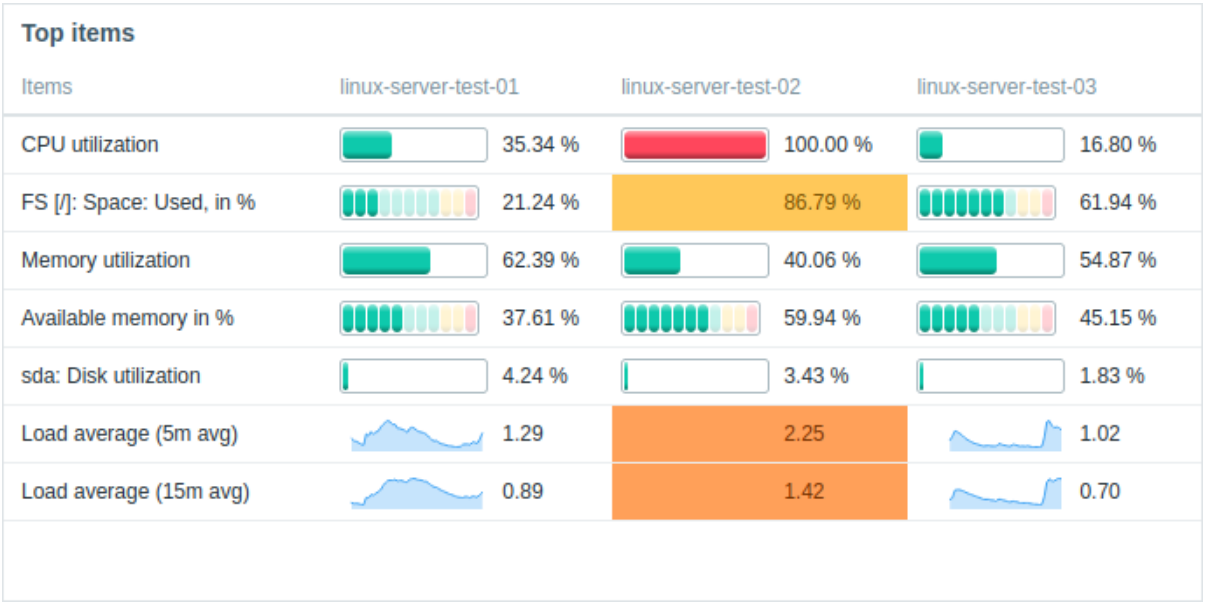
Parameter für **Binary item value**:

Show thumbnail	Geben Sie an, ob für das Bild mit Binärdaten ein Vorschaubild erstellt und angezeigt werden soll oder ob in der Wertespalte ein Hyperlink *Show* angezeigt werden soll, der zum Bild in voller Größe führt.

Top-Datenpunkte

Übersicht

Das Widget *Top items* zeigt die höchsten/niedrigsten Werte für ausgewählte Datenpunkte an und bietet einen schnellen Überblick über deren Leistung.

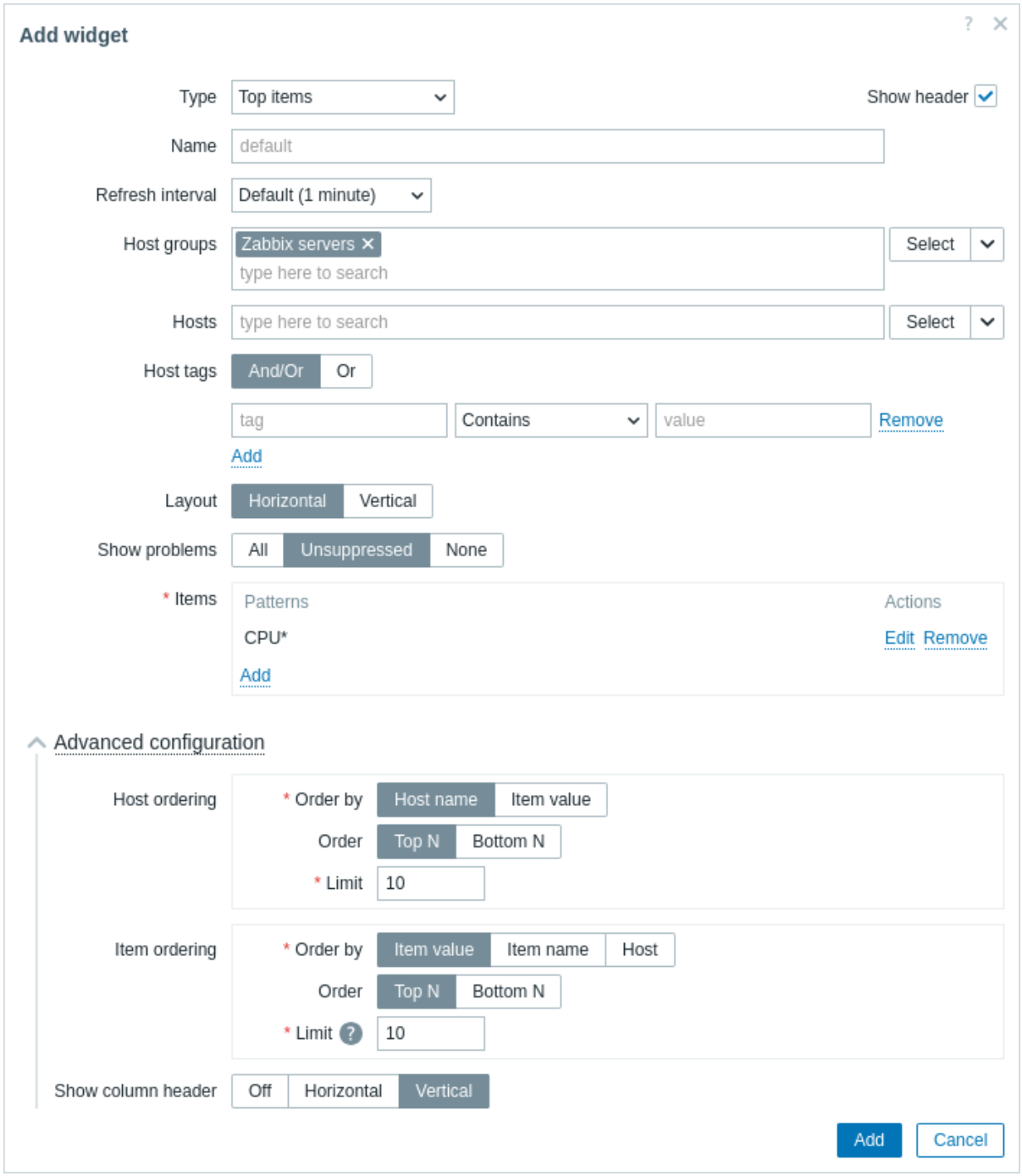


Standardmäßig zeigt das Widget Werte aus den letzten 24 Stunden an. Dieser Zeitraum kann unter *Administration* > *General* > *GUI* (*Maximale Anzeigeperiode des Verlaufs*) geändert werden.

Durch Klicken auf den Datenpunktwert wird das Datenpunktmenü geöffnet.

Konfiguration

Um zu konfigurieren, wählen Sie *Top items* als Typ:



Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

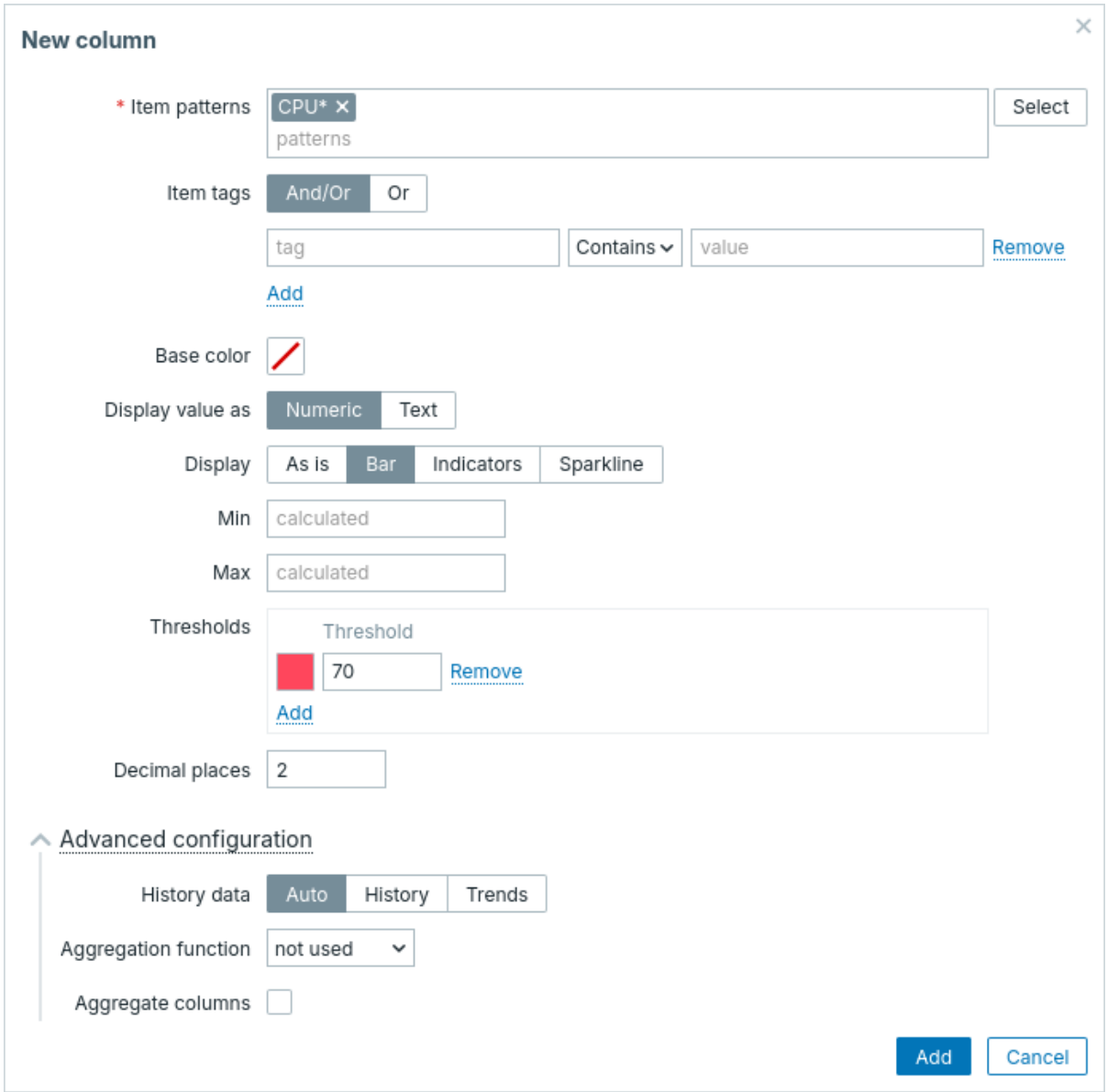
Host groups	Hostgruppen auswählen. Alternativ können Sie ein kompatibles Widget als Datenquelle für Hostgruppen auswählen. Dieses Feld unterstützt Autovervollständigung, sodass beim Eingeben des Namens einer Gruppe ein Dropdown mit passenden Gruppen angezeigt wird. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlage-Dashboard konfiguriert wird.

Hosts	Hosts auswählen. Alternativ können Sie ein kompatibles Widget oder das Dashboard als Datenquelle für Hosts auswählen. Dieses Feld unterstützt Autovervollständigung, sodass beim Eingeben des Namens eines Hosts ein Dropdown mit passenden Hosts angezeigt wird. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlage-Dashboard konfiguriert wird.
Host tags	Tags angeben, um die Menge der im Widget angezeigten Hostdaten zu begrenzen. Es ist möglich, bestimmte Tags und Tag-Werte einzuschließen oder auszuschließen. Es können mehrere Bedingungen festgelegt werden. Die Übereinstimmung des Tag-Namens ist immer groß-/kleinschreibungssensitiv. Für jede Bedingung stehen mehrere Operatoren zur Verfügung: **Exists** - die angegebenen Tag-Namen einschließen; **Equals** - die angegebenen Tag-Namen und Werte einschließen (groß-/kleinschreibungssensitiv); **Contains** - die angegebenen Tag-Namen einschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv); **Does not exist** - die angegebenen Tag-Namen ausschließen; **Does not equal** - die angegebenen Tag-Namen und Werte ausschließen (groß-/kleinschreibungssensitiv); **Does not contain** - die angegebenen Tag-Namen ausschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv). Für Bedingungen gibt es zwei Berechnungstypen: **And/Or** - alle Bedingungen müssen erfüllt sein, Bedingungen mit demselben Tag-Namen werden durch die *Or*-Bedingung gruppiert; **Or** - es genügt, wenn eine Bedingung erfüllt ist.
Layout	Layout-Option auswählen: **Horizontal** - Hostnamen werden links angezeigt; **Vertical** - Hostnamen werden oben angezeigt.
Show problems	Filtern Sie, welche Probleme basierend auf ihrem Status angezeigt werden sollen (alle, nicht unterdrückte, keine). Probleme werden durch farbige Datenpunktwerte angezeigt und überschreiben die Anzeigeeinstellungen der Datenpunktspalte. Die Farben basieren auf der Problem-Schwere, die unter *Administration > Allgemein > Trigger displaying options* angepasst werden kann. Problem-Schweregrade können beim Aktualisieren von Problemen geändert werden.
Items	Datenpunktmuster oder bestimmte Datenpunkte zur Anzeige hinzufügen (siehe Spaltenkonfiguration).
Advanced configuration	
Host ordering	Sortieroptionen für die Host-Spalte/-Zeile auswählen.
Order by	Hosts sortieren nach: **Host name** - Hosts werden nach Hostnamen sortiert; **Item value** - Hosts werden nach dem Wert der ausgewählten Datenpunkte sortiert.
Order	Wählen Sie aus, ob die höchsten oder niedrigsten Werte angezeigt werden sollen: **TopN** - die obersten N Werte; **BottomN** - die untersten N Werte. Der Wert von N wird im Feld *Limit* festgelegt.
Limit	Grenzwert für anzeigbare Hosts eingeben (Bereich 1-1000; Standard 10). Dieser Wert wird zum Wert von N im Feld *Order*.

Item ordering		Sortieroptionen für die Datenpunkt-Spalte/-Zeile auswählen.
	Order by	Datenpunkte sortieren nach: **Item value** - Datenpunkte werden nach dem Datenpunktwert sortiert; **Item name** - Datenpunkte werden nach dem Datenpunktnamen sortiert; **Host** - Datenpunkte werden nach dem ausgewählten Host-Muster sortiert.
	Order	Wählen Sie aus, ob die höchsten oder niedrigsten Werte angezeigt werden sollen: **TopN** - die obersten N Werte; **BottomN** - die untersten N Werte. Der Wert von N wird im Feld *Limit* festgelegt.
	Limit	Grenzwert für anzeigbare Datenpunkte eingeben (Bereich 1-1000; Standard 10). Dieser Wert wird zum Wert von N im Feld *Order*.
Show column header		Optionen für die Anzeige der Spaltenüberschrift auswählen: **Off** - Spaltenüberschrift nicht anzeigen; **Horizontal** - Text in der Überschrift horizontal anzeigen; **Vertical** - Text in der Überschrift vertikal anzeigen.

Spaltenkonfiguration

Um Datenpunkt-Spalten zu konfigurieren, klicken Sie im Parameter *Datenpunkte* auf *Hinzufügen*:



Allgemeine Spaltenparameter:

Datenpunkt-Muster	Geben Sie ein oder mehrere Datenpunkt-Muster unter Verwendung des Platzhalterzeichens an. Alternativ können Sie die Datenpunkte auswählen. Wenn Sie das Widget in einem Vorlagen-Dashboard konfigurieren, können nur in der Vorlage konfigurierte Datenpunkte ausgewählt werden.
Datenpunkt-Tags	Geben Sie Tags an, um die Anzahl der im Widget angezeigten Datenpunktdaten zu begrenzen. Es ist möglich, bestimmte Tags und Tag-Werte sowohl einzuschließen als auch auszuschließen. Es können mehrere Bedingungen festgelegt werden. Der Abgleich von Tag-Namen ist immer groß-/kleinschreibungssensitiv. Für jede Bedingung stehen mehrere Operatoren zur Verfügung: **Existiert** - die angegebenen Tag-Namen einschließen; **Gleich** - die angegebenen Tag-Namen und -Werte einschließen (groß-/kleinschreibungssensitiv); **Enthält** - die angegebenen Tag-Namen einschließen, bei denen die Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv); **Existiert nicht** - die angegebenen Tag-Namen ausschließen; **Ungleich** - die angegebenen Tag-Namen und -Werte ausschließen (groß-/kleinschreibungssensitiv); **Enthält nicht** - die angegebenen Tag-Namen ausschließen, bei denen die Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv). Für Bedingungen gibt es zwei Berechnungstypen: **Und/Oder** - alle Bedingungen müssen erfüllt sein; Bedingungen mit demselben Tag-Namen werden durch die *Oder*-Bedingung gruppiert; **Oder** - es genügt, wenn eine Bedingung erfüllt ist.
Basisfarbe	Wählen Sie die Hintergrundfarbe der Spalte oder die Füllfarbe aus, wenn *Anzeige* auf „Balken“ oder „Indikatoren“ gesetzt ist. Beachten Sie, dass die Basisfarbe durch Schwellenwert- oder Hervorhebungsfarben überschrieben werden kann.
Wert anzeigen als	Format für die Anzeige des Datenpunkt-Werts: **Numerisch** oder **Text**. Die ausgewählte Option bestimmt, welche zusätzlichen Parameter verfügbar sind. Siehe die Parameterliste für jedes Format unten.

Spaltenparameter für numerische Werte:

Anzeige	Legen Sie fest, wie der Wert angezeigt werden soll: **Wie vorhanden** – als regulärer Text; **Balken** – als ausgefüllter Balken mit einer Farbe; **Indikatoren** – als segmentierter, ausgefüllter Balken mit einer Farbe; **Sparkline** – als Mini-Liniendiagramm.
Min	Minimalwert für die Anzeige von Balken/Indikatoren.
Max	Maximalwert für die Anzeige von Balken/Indikatoren.
Breite	Legen Sie die Stärke der Diagrammlinie mithilfe des Schiebereglers fest oder geben Sie manuell einen Wert im Bereich von 0 bis 10 ein. Dieser Parameter gilt nur für die Sparkline-Anzeige.
Füllung	Legen Sie den Transparenzgrad der Füllfarbe mithilfe des Schiebereglers fest oder geben Sie manuell einen Wert im Bereich von 0 bis 10 ein. Dieser Parameter gilt nur für die Sparkline-Anzeige.
Farbe	Wählen Sie die Linien- und Füllfarbe aus. Dieser Parameter gilt nur für die Sparkline-Anzeige.
Eine Beschreibung der Felder für die Auswahl des Zeitraums und der Verlaufsdaten finden Sie unter *Erweiterte Konfiguration*.	
Schwellenwerte	Geben Sie Schwellenwerte an, bei denen sich die Hintergrund-/Füllfarbe ändern soll. Die Liste wird beim Speichern in aufsteigender Reihenfolge sortiert. Bei Sparklines werden Schwellenwerte nur auf den letzten Wert des Datenpunkts angewendet.
Dezimalstellen	Geben Sie an, wie viele Dezimalstellen mit dem Wert angezeigt werden sollen.

Erweiterte Konfiguration	
Verlaufsdaten	Beziehen Sie Daten aus dem Verlauf oder aus Trends: **Automatisch** – automatische Auswahl; **Verlauf** – Verlaufsdaten verwenden; **Trends** – Trenddaten verwenden. Dieser Parameter ist nur für numerische Datenpunktwerte verfügbar. Bei Textwerten werden die Daten immer aus dem Verlauf bezogen.
Aggregationsfunktion	Geben Sie an, welche Aggregationsfunktion innerhalb des ausgewählten *Zeitraums* verwendet werden soll: **min** – den kleinsten Wert anzeigen; **max** – den größten Wert anzeigen; **avg** – den Durchschnittswert anzeigen; **count** – die Anzahl der Werte anzeigen; **sum** – die Summe der Werte anzeigen; **first** – den ersten Wert anzeigen; **last** – den letzten Wert anzeigen. Wenn **nicht verwendet** ausgewählt ist, wird der aktuellste Wert angezeigt (keine Aggregation). Durch die Aggregation kann für das angegebene Intervall (5 Minuten, eine Stunde, ein Tag) ein aggregierter Wert anstelle des aktuellsten Werts angezeigt werden. Für *min*, *max*, *avg* und *sum* können nur numerische Daten angezeigt werden. Für *count* werden nicht numerische Daten in numerische Daten umgewandelt.
Zeitraum	Geben Sie den Zeitraum an, der für die Aggregation von Werten verwendet werden soll. Wählen Sie die Datenquelle für den Zeitraum aus: **Dashboard** – den Zeitraumauswahlschalter des Dashboards verwenden; **Widget** – ein kompatibles Widget verwenden (im Parameter *Widget* festgelegt); **Benutzerdefiniert** – einen benutzerdefinierten Zeitraum verwenden, der in den Parametern *Von* und *Bis* festgelegt ist; wenn dieser festgelegt ist, wird in der oberen rechten Ecke des Widgets ein Uhrsymbol angezeigt, das beim Überfahren mit der Maus die festgelegte Zeit angibt. Beachten Sie, dass kompatible Widgets unabhängig von der Konfiguration des Parameters *Zeitraum* des Widgets weiterhin als Datenquelle für den Zeitraum verwendet werden können. Dieser Parameter ist nicht verfügbar, wenn für *Aggregationsfunktion* „nicht verwendet“ festgelegt ist.
Widget	Geben Sie ein kompatibles Widget als Datenquelle für den Zeitraum ein oder wählen Sie es aus. Dieser Parameter ist verfügbar, wenn für *Zeitraum* „Widget“ festgelegt ist.
Von	Geben Sie den Beginn des Zeitraums ein oder wählen Sie ihn aus. Die Syntax für relative Zeitangaben (`now`, `now/d`, `now/w-1w` usw.) wird unterstützt. Dieser Parameter ist verfügbar, wenn für *Zeitraum* „Benutzerdefiniert“ festgelegt ist.
Bis	Geben Sie das Ende des Zeitraums ein oder wählen Sie es aus. Die Syntax für relative Zeitangaben (`now`, `now/d`, `now/w-1w` usw.) wird unterstützt. Dieser Parameter ist verfügbar, wenn für *Zeitraum* „Benutzerdefiniert“ festgelegt ist.
Spalten aggregieren	Wenn diese Option aktiviert ist, werden alle Datenpunkte, die den Datenpunktmustern der Spalte entsprechen, zu einer einzelnen kombinierten Spalte/Zeile zusammengefasst. Werte aus übereinstimmenden Datenpunkten werden mithilfe der ausgewählten *Spaltenaggregationsfunktion* aggregiert und unter dem *Namen der kombinierten Spalte* angezeigt. Standard: deaktiviert (nicht ausgewählt).Dieser Parameter wird nur angezeigt, wenn *Wert anzeigen als* auf **Numerisch** gesetzt ist.
Spaltenaggregationsfunktion	Wählen Sie aus, wie Werte aus den gruppierten Datenpunktmustern kombiniert werden. Verfügbare Optionen: **min**, **max**, **avg**, **count**, **sum** (Standard). Dieser Parameter wird nur angezeigt, wenn *Spalten aggregieren* aktiviert ist.
Name der kombinierten Spalte	Geben Sie den Anzeigenamen für die aggregierte Spalte ein. Dieses Feld ist erforderlich, wenn *Spalten aggregieren* aktiviert ist, und wird zur Spaltenüberschrift für die kombinierten Werte.

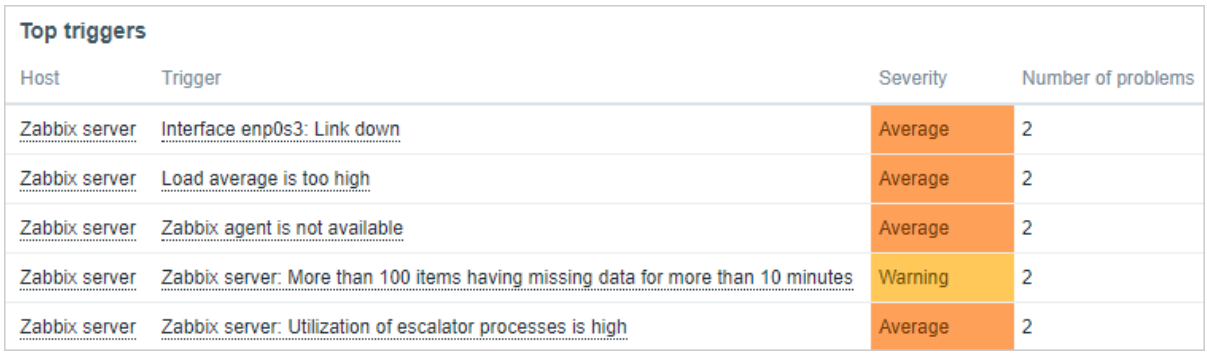
Spaltenparameter für Textwerte:

Hervorhebungen	Geben Sie die regulären Ausdrücke an, bei deren Übereinstimmung sich die Hintergrund-/Füllfarbe ändern soll.

Top-Auslöser

Übersicht

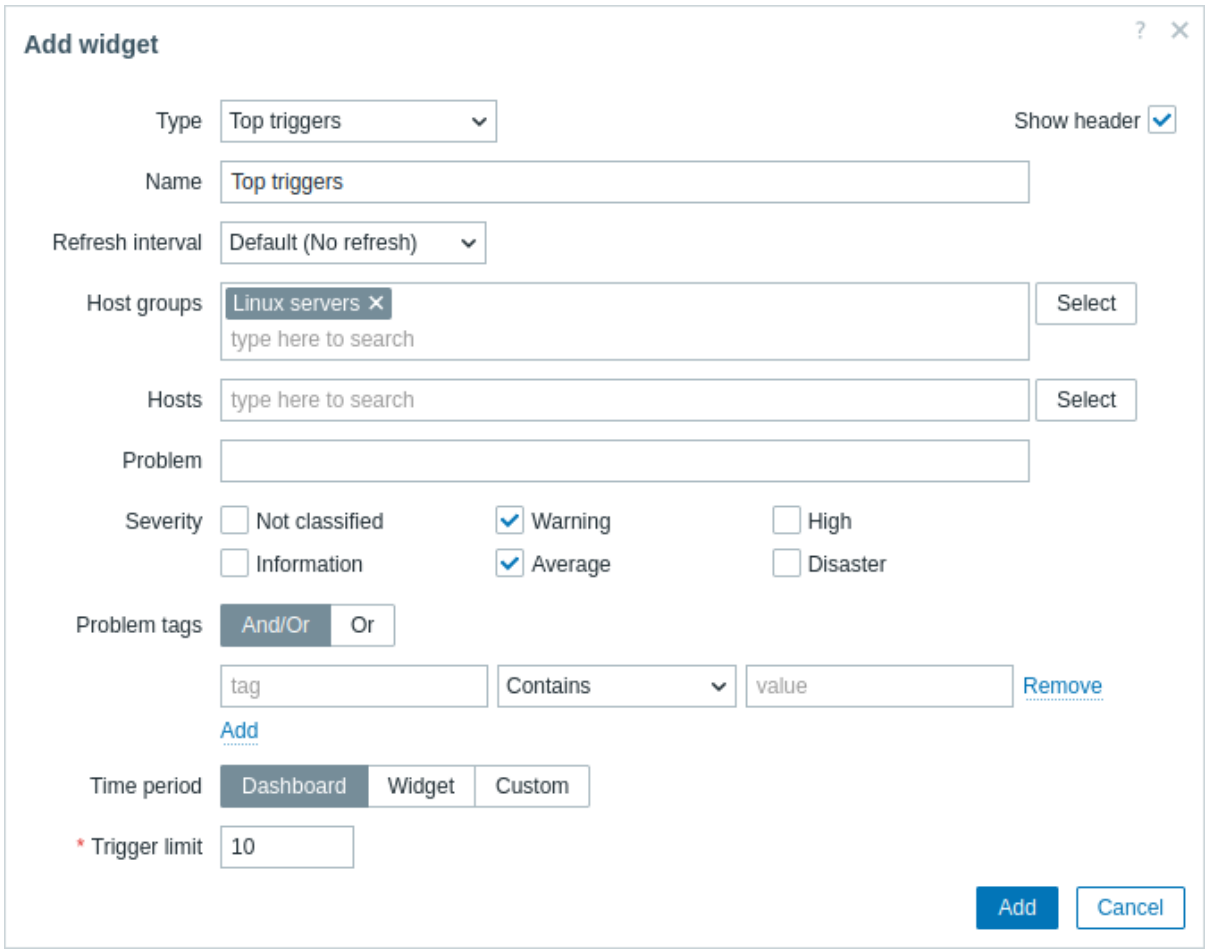
Das Widget *Top-Auslöser* zeigt Auslöser mit der höchsten Anzahl an Problemen an.



Es zeigt dieselben Daten wie *Berichte > Top 100 Auslöser* an und kann bis zu 1000 Einträge anzeigen.

Konfiguration

Um die Konfiguration vorzunehmen, wählen Sie *Top-Auslöser* als Widget-Typ aus:



Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

Host-Gruppen	Wählen Sie die Host-Gruppen aus, deren Probleme im Widget angezeigt werden sollen. Dieses Feld unterstützt die automatische Vervollständigung. Wenn Sie also beginnen, den Namen einer Gruppe einzugeben, wird eine Dropdown-Liste mit passenden Gruppen angeboten. Die Angabe einer übergeordneten Host-Gruppe wählt implizit auch alle darunterliegenden Host-Gruppen aus. Probleme aus diesen Host-Gruppen werden im Widget angezeigt; wenn keine Host-Gruppen eingegeben werden, werden Probleme aus allen Host-Gruppen angezeigt. Dieser Parameter ist nicht verfügbar, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.

Hosts	Wählen Sie die Hosts aus, deren Probleme im Widget angezeigt werden sollen. Dieses Feld unterstützt die automatische Vervollständigung. Wenn Sie also beginnen, den Namen eines Hosts einzugeben, wird eine Dropdown-Liste mit passenden Hosts angeboten. Wenn keine Hosts eingegeben werden, werden Probleme aller Hosts angezeigt. Dieser Parameter ist nicht verfügbar, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Problem	Sie können die Auslöser nur für bestimmte Probleme anzeigen. Geben Sie dazu die Zeichenfolge ein, die im Problemnamen übereinstimmen soll. Makros werden nicht expandiert.
Schweregrad	Markieren Sie Auslöser-Schweregrade, um die im Widget angezeigten Auslöser zu filtern. Wenn keine Schweregrade markiert sind, werden alle Auslöser angezeigt.
Problem-Tags	Geben Sie die Tags der Probleme an, die im Widget angezeigt werden sollen. Es ist möglich, bestimmte Tags und Tag-Werte sowohl einzuschließen als auch auszuschließen. Es können mehrere Bedingungen festgelegt werden. Der Abgleich von Tag-Namen ist immer groß-/kleinschreibungssensitiv. Für jede Bedingung stehen mehrere Operatoren zur Verfügung: **Existiert** - die angegebenen Tag-Namen einschließen; **Gleich** - die angegebenen Tag-Namen und Werte einschließen (groß-/kleinschreibungssensitiv); **Enthält** - die angegebenen Tag-Namen einschließen, bei denen die Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv); **Existiert nicht** - die angegebenen Tag-Namen ausschließen; **Ungleich** - die angegebenen Tag-Namen und Werte ausschließen (groß-/kleinschreibungssensitiv); **Enthält nicht** - die angegebenen Tag-Namen ausschließen, bei denen die Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv). Für Bedingungen gibt es zwei Berechnungstypen: **Und/Oder** - alle Bedingungen müssen erfüllt sein; Bedingungen mit demselben Tag-Namen werden durch die *Oder*-Bedingung gruppiert; **Oder** - es genügt, wenn eine Bedingung erfüllt ist.
Zeitraum	Wählen Sie die Datenquelle für den Zeitraum aus: **Dashboard** - den Zeitraumauswahl des Dashboards verwenden; **Widget** - ein kompatibles Widget verwenden (im Parameter *Widget* festgelegt); **Benutzerdefiniert** - einen benutzerdefinierten Zeitraum verwenden, der in den Parametern *Von* und *Bis* festgelegt ist; falls gesetzt, wird in der oberen rechten Ecke des Widgets ein Uhrsymbol angezeigt, das beim Überfahren mit der Maus auf die eingestellte Zeit hinweist. Beachten Sie, dass kompatible Widgets unabhängig von der *Zeitraum*-Konfiguration des Widgets weiterhin als Datenquelle für den Zeitraum verwendet werden können.
Widget	Geben Sie ein kompatibles Widget als Datenquelle für den Zeitraum ein oder wählen Sie es aus. Dieser Parameter ist verfügbar, wenn *Zeitraum* auf "Widget" gesetzt ist.
Von	Geben Sie den Beginn des Zeitraums ein oder wählen Sie ihn aus. Die Syntax für relative Zeitangaben (`now`, `now/d`, `now/w-1w` usw.) wird unterstützt. Dieser Parameter ist verfügbar, wenn *Zeitraum* auf "Benutzerdefiniert" gesetzt ist.
Bis	Geben Sie das Ende des Zeitraums ein oder wählen Sie es aus. Die Syntax für relative Zeitangaben (`now`, `now/d`, `now/w-1w` usw.) wird unterstützt. Dieser Parameter ist verfügbar, wenn *Zeitraum* auf "Benutzerdefiniert" gesetzt ist.
Auslöser-Limit	Legen Sie die Anzahl der anzuzeigenden Auslöser fest. Möglicher Wertebereich: 1-1000.

Auslöser-Übersicht

Übersicht

Das Widget *Trigger overview* zeigt die aktuellen Zustände von Host-Auslösern als Tabelle farbiger Blöcke an. Es bietet eine schnelle visuelle Momentaufnahme des Zustands und der Aktivität von Auslösern auf verschiedenen Hosts.



Das Widget verwendet mehrere visuelle Elemente, um Details zu Auslösern darzustellen:

- Blockfarben für PROBLEM-Auslöser entsprechen den Schweregraden der Probleme.
- Blinkende Blöcke zeigen kürzliche Zustandsänderungen von Auslösern an (innerhalb der letzten 2 Minuten).
- Graue Pfeile kennzeichnen Auslöser mit Abhängigkeiten (bewegen Sie den Mauszeiger darüber, um Details anzuzeigen).
- Kontrollkästchen-Symbole kennzeichnen bestätigte Probleme; alle aktiven oder behobenen Probleme für den Auslöser müssen bestätigt sein, damit dieses Symbol angezeigt wird.

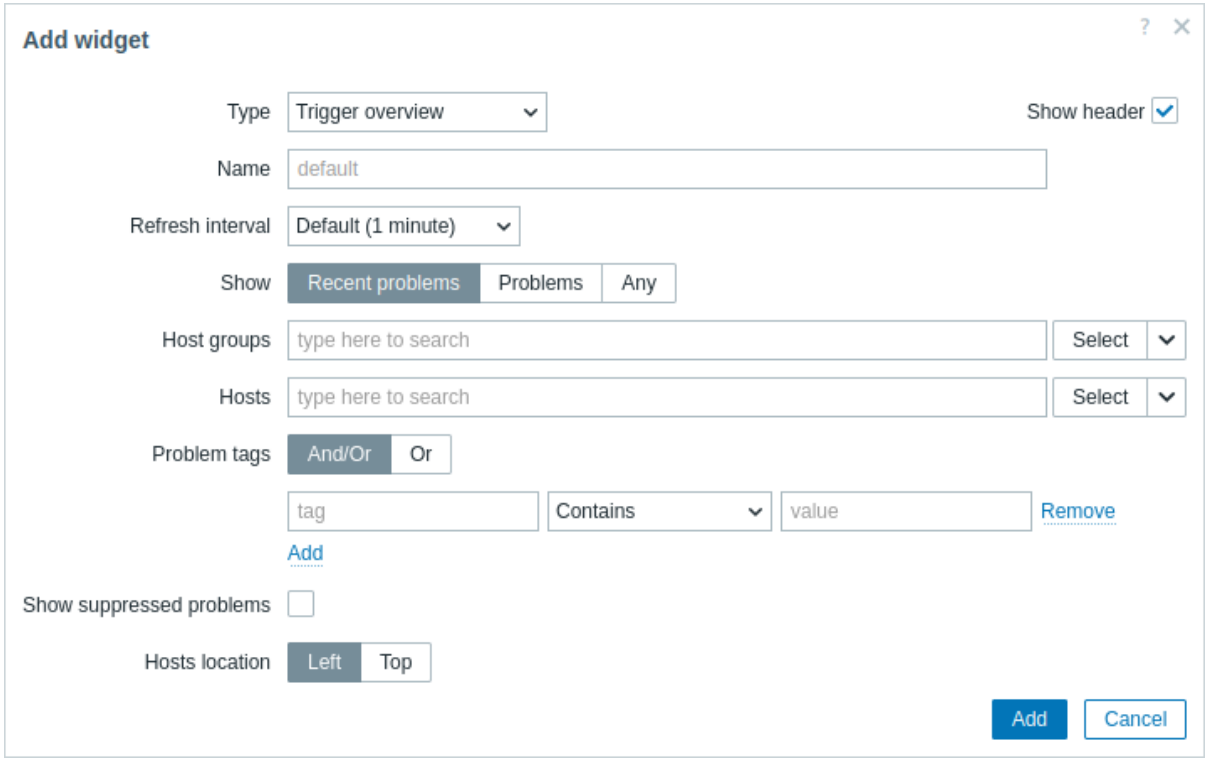
Note:
Die Farben der Auslöser-Schweregrade können unter *Administration* > *General* angepasst werden. Die Schweregrade von Auslösern können beim Aktualisieren von Problemen geändert werden.

Wenn Sie auf einen Block klicken, werden Links zu auslöserbezogenen Informationen angezeigt (Probleme, Auslöser-Konfiguration usw.).

Standardmäßig werden bis zu 50 Datensätze angezeigt (konfigurierbar unter *Administration* > *General* > *GUI* im Abschnitt *Max number of columns and rows in overview tables*). Wenn mehr Auslöser vorhanden sind, fordert eine Meldung Sie auf, die Filterkriterien zu verfeinern. Diese Begrenzung wird vor jeder zusätzlichen Filterung angewendet (zum Beispiel nach Tags).

Konfiguration

Wählen Sie zur Konfiguration *Auslöserübersicht* als Typ aus:



Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

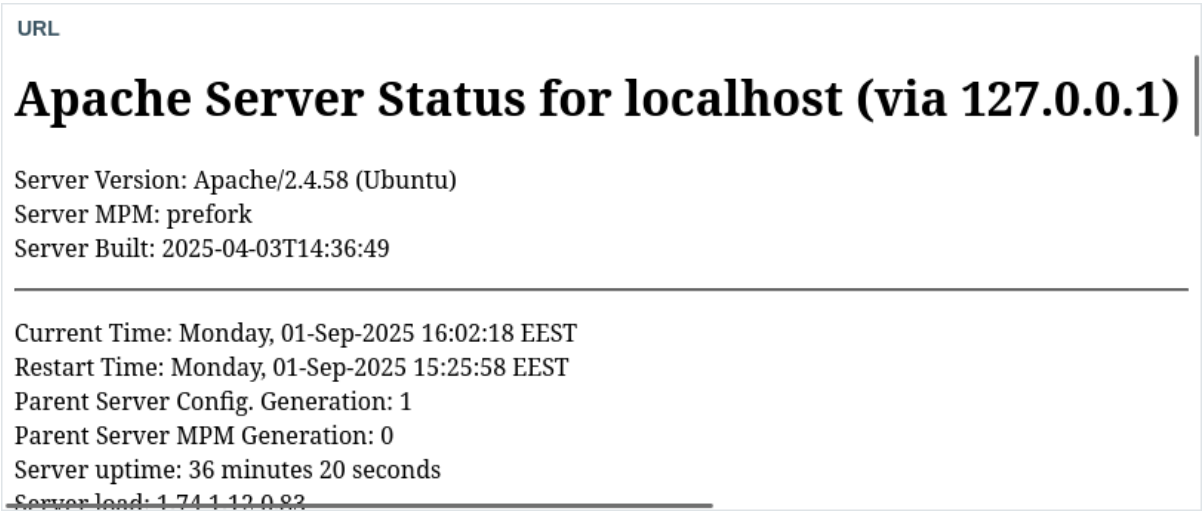
Anzeigen	Filtern Sie Auslöser nach Auslöserstatus: **Aktuelle Probleme** - *(Standard)* zeigt Auslöser an, die vor Kurzem in einem PROBLEM-Status waren oder sich noch darin befinden (gelöst und ungelöst); **Probleme** - zeigt Auslöser an, die sich in einem PROBLEM-Status befinden (ungelöst); **Beliebig** - zeigt alle Auslöser an.
Host-Gruppen	Wählen Sie Host-Gruppen aus. Alternativ können Sie ein kompatibles Widget als Datenquelle für Host-Gruppen auswählen. Dieses Feld unterstützt Autovervollständigung; wenn Sie beginnen, den Namen einer Gruppe einzugeben, wird eine Dropdown-Liste mit passenden Gruppen angeboten. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Hosts	Wählen Sie Hosts aus. Alternativ können Sie ein kompatibles Widget oder das Dashboard als Datenquelle für Hosts auswählen. Dieses Feld unterstützt Autovervollständigung; wenn Sie beginnen, den Namen eines Hosts einzugeben, wird eine Dropdown-Liste mit passenden Hosts angeboten. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.

Problem-Tags	Geben Sie Tags an, um die im Widget angezeigten Auslöser zu filtern. Es ist möglich, bestimmte Tags und Tag-Werte sowohl einzuschließen als auch auszuschließen. Es können mehrere Bedingungen festgelegt werden. Der Abgleich von Tag-Namen ist immer groß-/kleinschreibungssensitiv. **Hinweis:** Wenn der Parameter *Anzeigen* auf „Beliebig“ gesetzt ist, werden alle Auslöser angezeigt, auch wenn Tags angegeben sind. Während jedoch aktuelle Änderungen des Auslöserstatus (als blinkende Blöcke angezeigt) für alle Auslöser aktualisiert werden, werden die Details des Auslöserstatus (Farbe des Problemschweregrads und ob das Problem bestätigt wurde) nur für Auslöser aktualisiert, die den angegebenen Tags entsprechen. Für jede Bedingung sind mehrere Operatoren verfügbar: **Existiert** - die angegebenen Tag-Namen einschließen; **Gleich** - die angegebenen Tag-Namen und -Werte einschließen (groß-/kleinschreibungssensitiv); **Enthält** - die angegebenen Tag-Namen einschließen, bei denen die Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv); **Existiert nicht** - die angegebenen Tag-Namen ausschließen; **Ungleich** - die angegebenen Tag-Namen und -Werte ausschließen (groß-/kleinschreibungssensitiv); **Enthält nicht** - die angegebenen Tag-Namen ausschließen, bei denen die Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv). Für Bedingungen gibt es zwei Berechnungstypen: **Und/Oder** - alle Bedingungen müssen erfüllt sein; Bedingungen mit demselben Tag-Namen werden durch die *Oder*-Bedingung gruppiert; **Oder** - es genügt, wenn eine Bedingung erfüllt ist.
Unterdrückte Probleme anzeigen	Aktivieren Sie das Kontrollkästchen, um Probleme anzuzeigen, die andernfalls aufgrund von Host-Wartung unterdrückt (nicht angezeigt) würden.
Layout	Wählen Sie die Layout-Option aus: **Horizontal** - Host-Namen werden links angezeigt; **Vertikal** - Host-Namen werden oben angezeigt.

URL

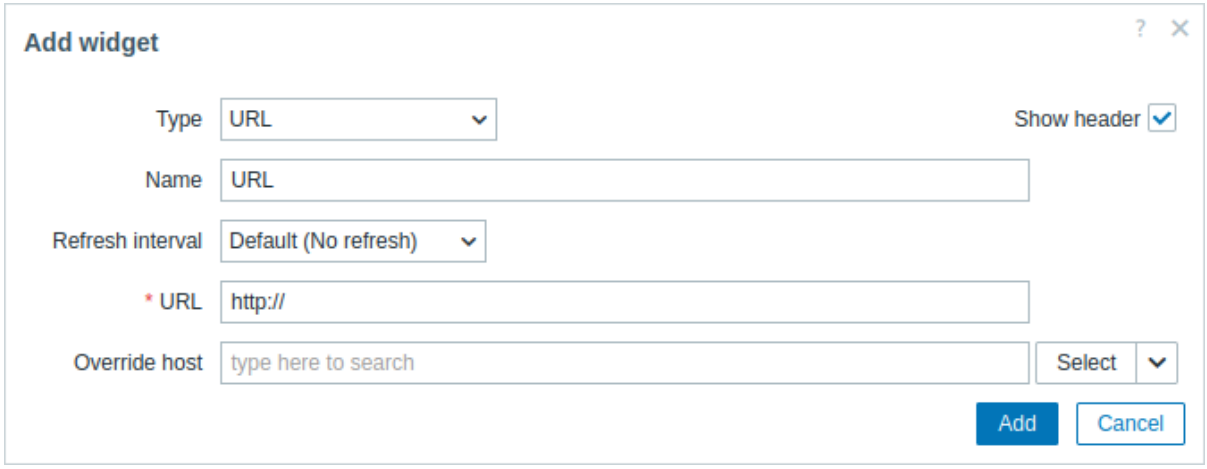
Übersicht

Das *URL*-Widget zeigt den von einer URL abgerufenen Inhalt an.



Konfiguration

Um zu konfigurieren, wählen Sie *URL* als Typ:



Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

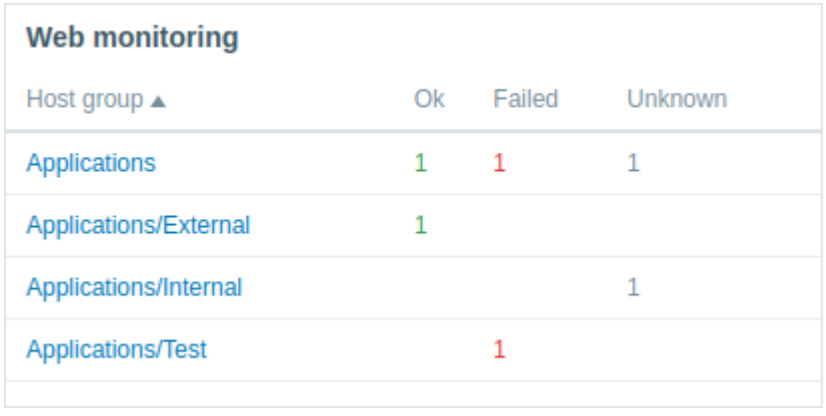
URL	Geben Sie die URL ein, die angezeigt werden soll (bis zu 2048 Zeichen). Externe URLs müssen mit `http://` oder `https://` beginnen. Interne URLs unterstützen relative Pfade (zum Beispiel `zabbix.php?action=report.status`). `{HOST.*}`-Makros werden unterstützt.
Override host	Wählen Sie ein kompatibles Widget oder den Dashboard-Hostauswähler als Datenquelle für Hosts aus. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einer Vorlage-Dashboard konfiguriert wird.

Attention:
Browser laden möglicherweise eine im Widget konfigurierte HTTP-Seite nicht, wenn auf das Zabbix-Frontend über HTTPS zugegriffen wird. Außerdem müssen Sie möglicherweise den X-Frame-Options HTTP-Antwortheader anpassen, um Inhalte anzuzeigen, die außerhalb des Zabbix-Frontend liegen.

Webüberwachung

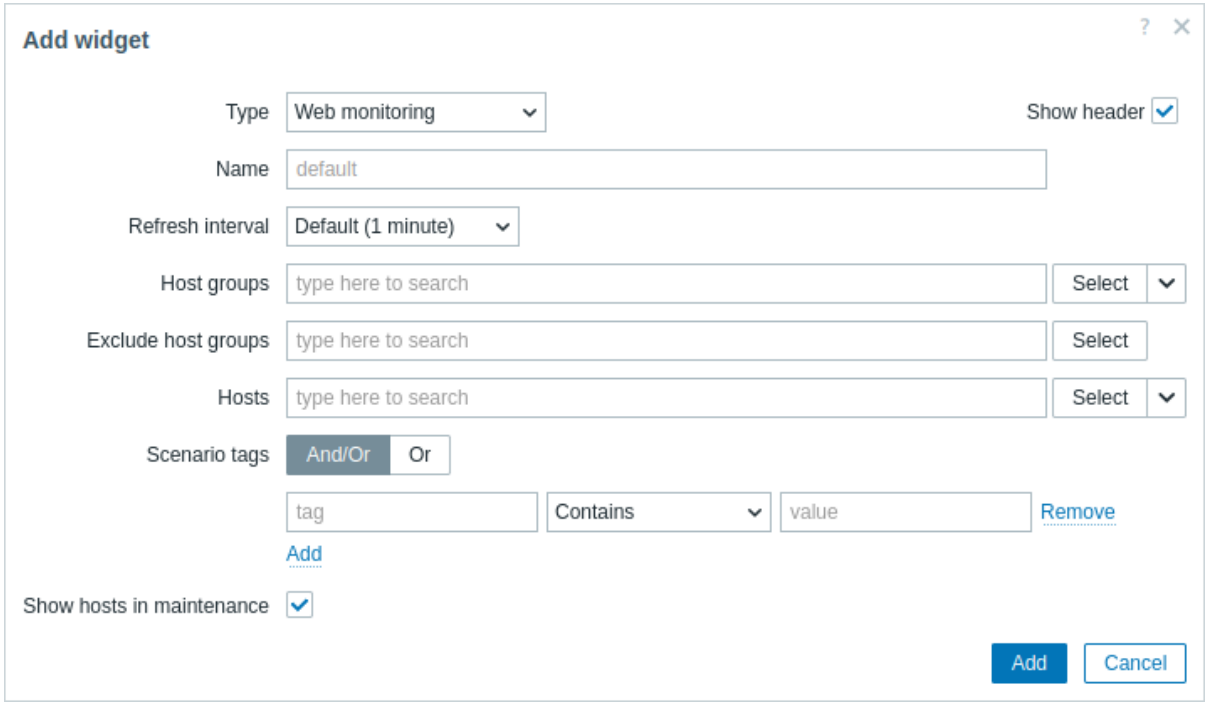
Übersicht

Das Widget *Web monitoring* zeigt den Status aktiver Webszenarien an.



Ein Klick auf den Namen einer Hostgruppe öffnet *Monitoring* > *Hosts* > *Web scenarios*, wo Sie Details zum Szenario anzeigen können.

Konfiguration



Zusätzlich zu den Parametern, die für alle Widgets gemeinsam sind, können Sie die folgenden spezifischen Optionen festlegen:

Host-Gruppen	Wählen Sie Host-Gruppen aus, die im Widget angezeigt werden sollen. Alternativ können Sie ein kompatibles Widget als Datenquelle für Host-Gruppen auswählen. Dieses Feld unterstützt Autovervollständigung; wenn Sie beginnen, den Namen einer Gruppe einzugeben, wird eine Dropdown-Liste mit passenden Gruppen angeboten. Die Angabe einer übergeordneten Host-Gruppe wählt implizit auch alle untergeordneten Host-Gruppen aus. Host-Daten aus diesen Host-Gruppen werden im Widget angezeigt; wenn keine Host-Gruppen eingegeben werden, werden alle Host-Gruppen angezeigt. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Host-Gruppen ausschließen	Wählen Sie Host-Gruppen aus, die im Widget ausgeblendet werden sollen. Dieses Feld unterstützt Autovervollständigung; wenn Sie beginnen, den Namen einer Gruppe einzugeben, wird eine Dropdown-Liste mit passenden Gruppen angeboten. Die Angabe einer übergeordneten Host-Gruppe wählt implizit auch alle untergeordneten Host-Gruppen aus. Host-Daten aus diesen Host-Gruppen werden im Widget nicht angezeigt. Zum Beispiel können sich die Hosts 001, 002, 003 in Gruppe A befinden und die Hosts 002, 003 zusätzlich in Gruppe B. Wenn wir gleichzeitig Gruppe A *anzeigen* und Gruppe B *ausschließen*, werden im Dashboard nur Daten von Host 001 angezeigt. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Hosts	Wählen Sie Hosts aus, die im Widget angezeigt werden sollen. Alternativ können Sie ein kompatibles Widget oder das Dashboard als Datenquelle für Hosts auswählen. Dieses Feld unterstützt Autovervollständigung; wenn Sie beginnen, den Namen eines Hosts einzugeben, wird eine Dropdown-Liste mit passenden Hosts angeboten. Wenn keine Hosts eingegeben werden, werden alle Hosts angezeigt. Dieser Parameter ist nicht verfügbar, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.

Szenario-Tags	Geben Sie Tags an, um die Anzahl der im Widget angezeigten Webszenarien zu begrenzen. Es ist möglich, bestimmte Tags und Tag-Werte sowohl einzuschließen als auch auszuschließen. Es können mehrere Bedingungen festgelegt werden. Der Abgleich von Tag-Namen ist immer groß-/kleinschreibungssensitiv. Für jede Bedingung stehen mehrere Operatoren zur Verfügung: **Existiert** - die angegebenen Tag-Namen einschließen; **Gleich** - die angegebenen Tag-Namen und -Werte einschließen (groß-/kleinschreibungssensitiv); **Enthält** - die angegebenen Tag-Namen einschließen, bei denen die Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv); **Existiert nicht** - die angegebenen Tag-Namen ausschließen; **Ungleich** - die angegebenen Tag-Namen und -Werte ausschließen (groß-/kleinschreibungssensitiv); **Enthält nicht** - die angegebenen Tag-Namen ausschließen, bei denen die Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv). Für Bedingungen gibt es zwei Berechnungstypen: **Und/Oder** - alle Bedingungen müssen erfüllt sein; Bedingungen mit demselben Tag-Namen werden durch die Bedingung *Oder* gruppiert; **Oder** - es genügt, wenn eine Bedingung erfüllt ist.
Hosts in Wartung anzeigen	Hosts, die sich in Wartung befinden, in die Statistik einbeziehen. Dieser Parameter ist mit *Daten in Wartung anzeigen* beschriftet, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.

Überwachung

Übersicht

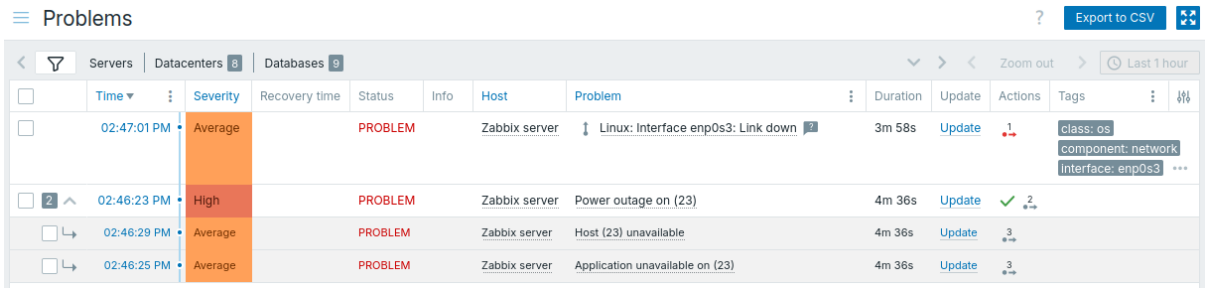
Im Menü *Monitoring* dreht sich alles um die Anzeige von Daten.
Alle Informationen, die Zabbix erfassen, visualisieren und für die es Aktionen ausführen soll, werden in den verschiedenen Bereichen des Menüs *Monitoring* angezeigt.

Probleme

Übersicht

In *Monitoring* > *Probleme* können Sie sehen, welche Probleme Sie derzeit haben. Probleme sind jene Auslöser, die sich im Status "Problem" befinden.

Standardmäßig werden alle neuen Probleme als Ursachenprobleme klassifiziert. Es ist möglich, bestimmte Probleme manuell als Symptomenprobleme des Ursachenproblems neu zu klassifizieren. Weitere Details finden Sie unter Ursachen- und Symptomenereignisse.

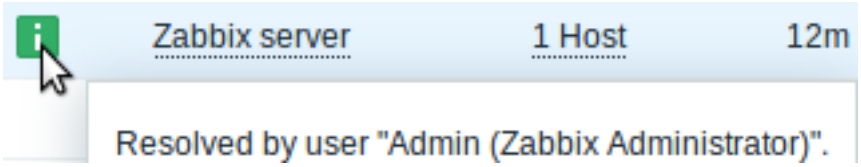
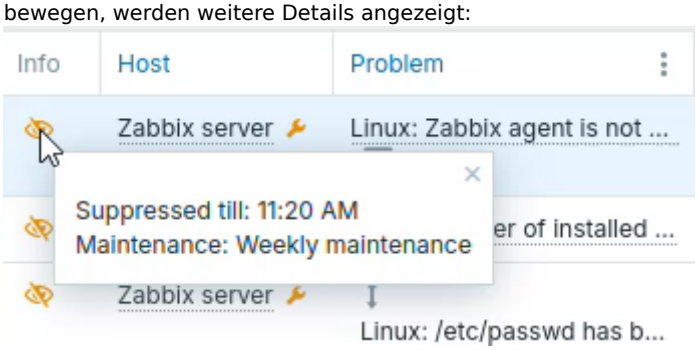


Problemdetails werden in einer anpassbaren Tabelle angezeigt. Die Schaltfläche *Tabelle anpassen* in der Tabellenkopfzeile ermöglicht es Ihnen, Spalten ein- oder auszublenden, ihre Reihenfolge zu ändern und das Layout auf die Standardwerte zurückzusetzen. Diese Optionen sind ebenfalls verfügbar:

- *Kompakte Ansicht* - reduziert die Zeilenhöhe, damit mehr Probleme auf dem Bildschirm angezeigt werden.
- *Gesamte Zeile hervorheben* - verwendet die Schweregradfarbe des Problems, um ungelöste Probleme hervorzuheben (in kontrastreichen Designs nicht verfügbar).

Sie können außerdem jede Spalte in der Breite anpassen, indem Sie ihren rechten Rand ziehen; doppelklicken Sie auf den Rand, um die Einstellung zurückzusetzen. Alle Anpassungen werden sofort übernommen und für den aktuellen Benutzer innerhalb des aktiven Filters-Tabs (Standard oder gespeichert) immer beibehalten.

Die folgenden Spalten sind verfügbar:

Column	Description
Checkbox	Checkboxen zur Auswahl von Problemen werden angezeigt. Symbole neben den Checkboxen haben die folgende Bedeutung:  - die Anzahl der Symptomenereignisse für das Ursachenproblem;  - aufklappen, um Symptomenereignisse anzuzeigen;  - zuklappen, um Symptomenereignisse auszublenden;  - dies ist ein Symptomenereignis.
Time	Die Startzeit des Problems wird angezeigt. Wenn Sie auf die Schaltfläche *Optionen* in der Spaltenkopfzeile klicken, erhalten Sie Zugriff auf die Option *Zeitachse anzeigen*, mit der die visuelle Zeitachse und die Problemgruppierung umgeschaltet werden können (nicht verfügbar, wenn *Kompakte Ansicht* aktiviert ist).
Severity	Der Schweregrad des Problems wird angezeigt. Der Schweregrad des Problems basiert ursprünglich auf dem Schweregrad des zugrunde liegenden Problem-Auslösers, kann jedoch nach dem Auftreten des Ereignisses über den Bildschirm *Problem aktualisieren* geändert werden. Die Farbe des Problem-Schweregrads wird während der Problemzeit als Zellenhintergrund verwendet.
Recovery time	Die Behebungszeit des Problems wird angezeigt Diese Spalte ist nicht verfügbar, wenn *Anzeigen* im Filter auf *Probleme* gesetzt ist.
Status	Der Problemstatus wird angezeigt: **Problem** - ungelöstes Problem **Resolved** - kürzlich gelöstes Problem. Sie können kürzlich gelöste Probleme mithilfe des Filters ausblenden. Neue und kürzlich gelöste Probleme blinken 2 Minuten lang. Gelöste Probleme werden insgesamt 5 Minuten lang angezeigt. Beide Werte sind unter *Administration* > *Allgemein* > *Optionen zur Auslöseranzeige* konfigurierbar. Diese Spalte ist nicht verfügbar, wenn *Anzeigen* im Filter auf *Probleme* gesetzt ist.
Info	Ein grünes Informationssymbol wird angezeigt, wenn ein Problem durch globale Korrelation oder manuell beim Aktualisieren des Problems geschlossen wurde. Wenn Sie den Mauszeiger über das Symbol bewegen, werden weitere Details angezeigt:  Wenn ein unterdrücktes Problem angezeigt wird, wird das folgende Symbol angezeigt:  . Wenn das Ereignis durch Host-Wartung unterdrückt wird, wird das Symbol orange angezeigt:  . Wenn sowohl Wartung als auch manuelle Unterdrückung zutreffen, hat die manuelle Unterdrückung bei der Symbolfarbe Vorrang. Wenn Sie den Mauszeiger über das Symbol bewegen, werden weitere Details angezeigt: 
Host	Der Problem-Host wird angezeigt. Wenn Sie auf den Host-Namen klicken, wird das Host-Menü geöffnet.

Column	Description
Problem	Der Problemname wird angezeigt. Der Problemname basiert auf dem Namen des zugrunde liegenden Problem-Auslösers. Makros im Namen des Auslösers werden zum Zeitpunkt des Auftretens des Problems aufgelöst, und die aufgelösten Werte werden danach nicht mehr aktualisiert. Wenn Sie auf den Problemnamen klicken, wird das Ereignis-Menü geöffnet.  Wenn Sie den Mauszeiger über das Symbol nach dem Problemnamen bewegen, wird die Auslöserbeschreibung angezeigt (für jene Probleme, die eine haben). Wenn Sie auf die Schaltfläche *Optionen* in der Spaltenkopfzeile klicken, erhalten Sie Zugriff auf zwei Optionen: - *Betriebsdaten anzeigen*, wodurch Betriebsdaten nach dem Problemnamen in Klammern angezeigt werden (nur wenn das Feld *Betriebsdaten* in der Auslöserkonfiguration nicht leer ist). - *Auslöserausdruck anzeigen*, wodurch der Auslöserausdruck des Problems unter dem Problemnamen angezeigt wird (nicht verfügbar, wenn *Kompakte Ansicht* aktiviert ist).
Operational data	Dynamische Echtzeitdaten für das Problem werden angezeigt. Der Inhalt hängt vom Feld *Betriebsdaten* in der Auslöserkonfiguration ab: Wenn eine Zeichenfolge mit Makros definiert ist, wird sie hier angezeigt; wenn das Feld leer gelassen wird, werden stattdessen die neuesten Werte aller Datenpunkte im Auslöserausdruck angezeigt.
Duration	Die Problemdauer wird angezeigt. Siehe auch: Negative Problemdauer
Update	Klicken Sie auf den Link *Aktualisieren*, um zum Bildschirm Problem aktualisieren zu gelangen, auf dem verschiedene Aktionen für das Problem durchgeführt werden können, einschließlich Kommentieren und Bestätigen des Problems.
Actions	Der Verlauf der Aktivitäten zum Problem wird mithilfe symbolischer Symbole angezeigt:  - das Problem wurde bestätigt. Dieses Symbol wird immer zuerst angezeigt.  - es wurden Kommentare hinzugefügt. Die Anzahl der Kommentare wird ebenfalls angezeigt.  - der Schweregrad des Problems wurde erhöht (z. B. Information > Warning).  - der Schweregrad des Problems wurde verringert (z. B. Warning > Information).  - der Schweregrad des Problems wurde geändert, ist jedoch auf den ursprünglichen Wert zurückgekehrt (z. B. Warning > Information > Warning).  - das Problem wurde unterdrückt.  - die Unterdrückung des Problems wurde aufgehoben.  - das Problem wurde aufgrund von Host-Wartung unterdrückt.  - die Unterdrückung des Problems wurde nach der Host-Wartung aufgehoben.  - es wurden Aktionen ausgeführt. Die Anzahl der Aktionen wird ebenfalls angezeigt.  - es wurden Aktionen ausgeführt, mindestens eine davon ist noch in Bearbeitung. Die Anzahl der Aktionen wird ebenfalls angezeigt.  - es wurden Aktionen ausgeführt, mindestens eine davon ist fehlgeschlagen. Die Anzahl der Aktionen wird ebenfalls angezeigt. Wenn Sie den Mauszeiger über die Symbole bewegen, werden Pop-ups mit Details zur Aktivität angezeigt. Weitere Informationen zu den im Pop-up für ausgeführte Aktionen verwendeten Symbolen finden Sie unter Details anzeigen.
Tags	Problem-Tags werden angezeigt. Zusätzlich können auch Tags aus einem externen Ticketsystem angezeigt werden (siehe die Option *Tags verarbeiten* bei der Konfiguration von webhooks). Wenn Sie den Mauszeiger über das Drei-Punkte-Symbol nach den angezeigten Tags bewegen oder darauf klicken, wird ein Pop-up mit allen Tags geöffnet. Wenn Sie auf die Schaltfläche *Optionen* in der Spaltenkopfzeile klicken, können Sie die Anzahl der angezeigten Tags (1-3) festlegen, auswählen, wie Tag-Namen angezeigt werden (vollständig, gekürzt oder gar nicht), die Anzeigepriorität der Tags festlegen oder die Spalte duplizieren (duplizierte Spalten können umbenannt oder gelöscht werden).

Column	Description
Tag value	Problem-Tag-Werte werden angezeigt. Wenn Sie auf die Schaltfläche ⋮ *Optionen* in der Spaltenkopfzeile klicken, können Sie die Spalte umbenennen, auswählen, welche Tag-Werte angezeigt werden sollen, oder die Spalte duplizieren (duplizierte Spalten können umbenannt oder gelöscht werden).
Custom text	Benutzerdefinierter Text, Makrowerte oder eine Kombination aus beidem werden angezeigt. Wenn Sie auf die Schaltfläche ⋮ *Optionen* in der Spaltenkopfzeile klicken, können Sie die Spalte umbenennen, den benutzerdefinierten Text oder Makros eingeben oder die Spalte duplizieren (duplizierte Spalten können umbenannt oder gelöscht werden). Unterstützte Makros: `{HOST.*}` (außer `{HOST.METADATA}`), `{INVENTORY.*}`, `{ITEM.*}` (außer `{ITEM.STATE.ERROR}`), `{TRIGGER.ID}`, `{EVENT.ID}` und Benutzermakros.

Negative Problemdauer

In einigen häufigen Situationen ist es tatsächlich möglich, dass eine negative Problemdauer auftritt, d. h. wenn der Zeitpunkt der Problembehebung vor dem Zeitpunkt der Problemerstellung liegt, z. B.:

- Wenn ein Host von einem Proxy überwacht wird und ein Netzwerkfehler auftritt, sodass eine Zeit lang keine Daten vom Proxy empfangen werden, wird der nodata(/host/key)-Auslöser vom Server ausgelöst. Wenn die Verbindung wiederhergestellt wird, empfängt der Server Datenpunkte vom Proxy mit einem Zeitstempel aus der Vergangenheit. Dann wird das nodata(/host/key)-Problem behoben und weist eine negative Problemdauer auf;
- Wenn Datenpunkte, die das Problemereignis beheben, von Zabbix sender gesendet werden und einen Zeitstempel enthalten, der vor dem Zeitpunkt der Problemerstellung liegt, wird ebenfalls eine negative Problemdauer angezeigt.

Note:
Eine negative Problemdauer wirkt sich in keiner Weise auf die SLA-Berechnung oder den Verfügbarkeitsbericht eines bestimmten Auslösers aus; sie verkürzt oder verlängert die Problemzeit weder.

Optionen für die Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen für die Massenbearbeitung:

- *Massenaktualisierung* - aktualisiert die ausgewählten Probleme, indem zur Seite Problemaktualisierung navigiert wird

Um diese Option zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Problemen und klicken Sie dann auf die Schaltfläche *Massenaktualisierung*.

Schaltflächen

Die Schaltfläche rechts bietet die folgende Option:

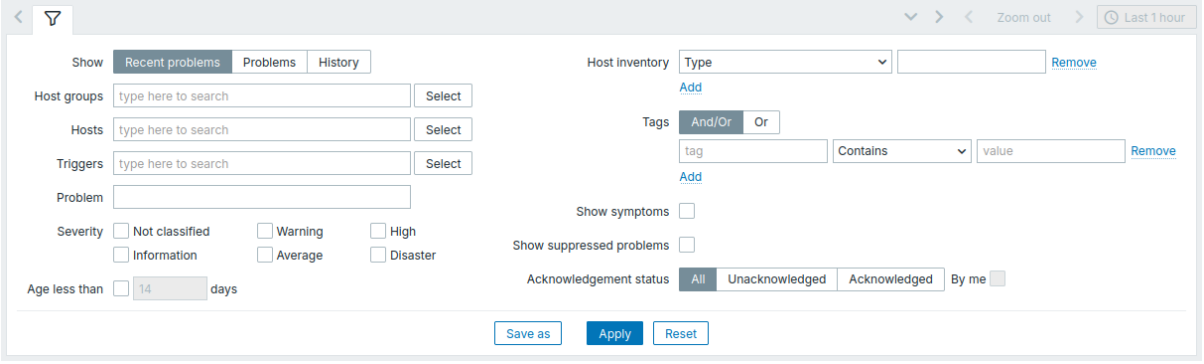
Export to CSV	Inhalt von allen Seiten in eine CSV-Datei exportieren.

Schaltflächen für den Ansichtsmodus, die in allen Bereichen gleich sind, werden auf der Seite Monitoring beschrieben.

Verwendung des Filters

Sie können den Filter verwenden, um nur die Probleme anzuzeigen, die Sie interessieren. Für eine bessere Suchleistung werden die Daten mit nicht aufgelösten Makros durchsucht.

Der Filter befindet sich oberhalb der Tabelle. Favorisierte Filtereinstellungen können als Tabs gespeichert und dann schnell über die Tabs oberhalb des Filters aufgerufen werden.

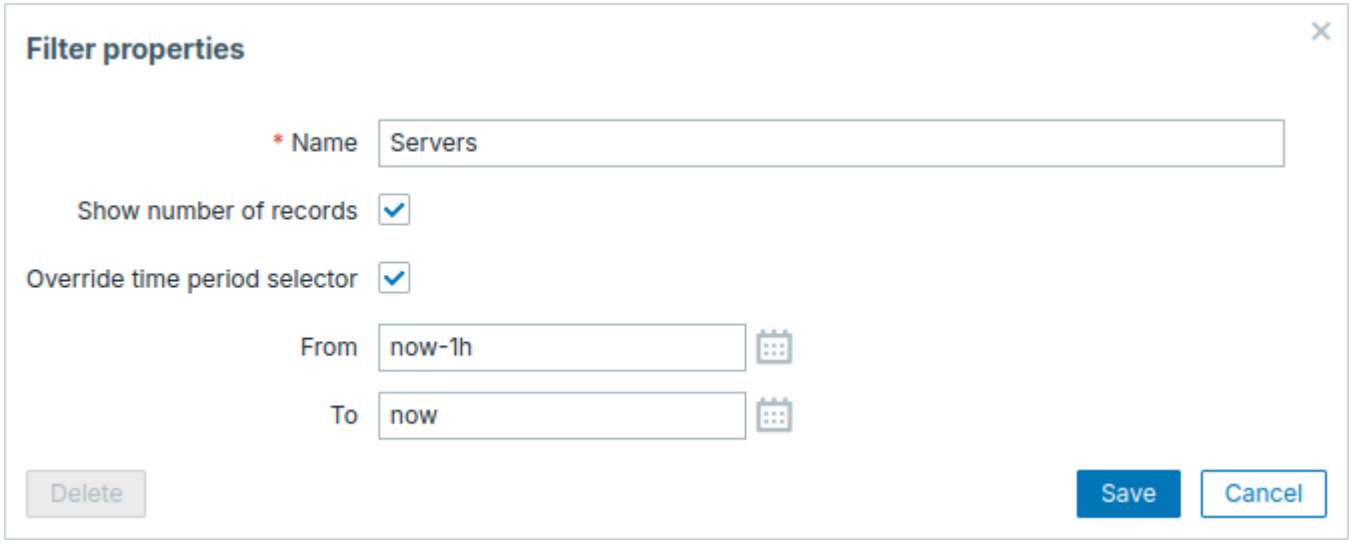


Parameter	Beschreibung
Show	Nach Problemstatus filtern: **Recent problems** - ungelöste und kürzlich gelöste Probleme werden angezeigt (Standard) **Problems** - ungelöste Probleme werden angezeigt **History** - der Verlauf aller Ereignisse wird angezeigt
Host groups	Nach einer oder mehreren Hostgruppen filtern. Die Angabe einer übergeordneten Hostgruppe wählt implizit alle verschachtelten Hostgruppen aus.
Hosts	Nach einem oder mehreren Hosts filtern.
Triggers	Nach einem oder mehreren Auslösern filtern.
Problem	Nach dem Problemnamen filtern.
Severity	Nach der Schwere des Auslösers (Problems) filtern.
Age less than	Nach dem Alter des Problems filtern.
Host inventory	Nach Inventartyp und -wert filtern.
Tags	Nach Ereignistag-Name und -Wert filtern. Es ist möglich, bestimmte Tags und Tag-Werte einzuschließen sowie auszuschließen. Es können mehrere Bedingungen festgelegt werden. Die Übereinstimmung des Tag-Namens ist immer groß-/kleinschreibungssensitiv. Für jede Bedingung stehen mehrere Operatoren zur Verfügung: **Exists** - die angegebenen Tag-Namen einschließen **Equals** - die angegebenen Tag-Namen und -Werte einschließen (groß-/kleinschreibungssensitiv) **Contains** - die angegebenen Tag-Namen einschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv) **Does not exist** - die angegebenen Tag-Namen ausschließen **Does not equal** - die angegebenen Tag-Namen und -Werte ausschließen (groß-/kleinschreibungssensitiv) **Does not contain** - die angegebenen Tag-Namen ausschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv) Für Bedingungen gibt es zwei Berechnungstypen: **And/Or** - alle Bedingungen müssen erfüllt sein, Bedingungen mit demselben Tag-Namen werden durch die Or-Bedingung gruppiert **Or** - es genügt, wenn eine Bedingung erfüllt ist Beim Filtern werden die hier angegebenen Tags zusammen mit dem Problem zuerst angezeigt, sofern sie nicht durch die Liste *Tag display priority* (siehe unten) überschrieben werden.
Show symptoms	Aktivieren Sie das Kontrollkästchen, um als Symptome klassifizierte Probleme in einer eigenen Zeile anzuzeigen.
Show suppressed problems	Aktivieren Sie das Kontrollkästchen, um Probleme anzuzeigen, die andernfalls unterdrückt und ausgeblendet würden, weil sich der Host in Wartung befindet oder sie manuell von einem Benutzer unterdrückt wurden.
Acknowledgement status	Filtern, um alle Probleme, nur nicht bestätigte Probleme oder nur bestätigte Probleme anzuzeigen. Aktivieren Sie das zusätzliche Kontrollkästchen, um Probleme herauszufiltern, die jemals von Ihnen bestätigt wurden.

Tabs für Favoritenfilter

Häufig verwendete Sätze von Filterparametern können in Tabs gespeichert werden.

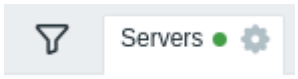
Um einen neuen Satz von Filterparametern zu speichern, öffnen Sie den Haupt-Tab, konfigurieren Sie die Filtereinstellungen und klicken Sie dann auf die Schaltfläche *Save as*. Definieren Sie in einem neuen Popup-Fenster die *Filter properties*.



Parameter	Beschreibung
Name	Der Name des Filters, der in der Tab-Liste angezeigt werden soll.
Show number of records	Aktivieren Sie diese Option, wenn die Anzahl der Probleme neben dem Tab-Namen angezeigt werden soll.
Override time period selector	Aktivieren Sie diese Option, um für diesen Filtersatz einen bestimmten Standard-Zeitraum festzulegen. Wenn diese Option gesetzt ist, können Sie den Zeitraum für diesen Tab nur ändern, indem Sie die Filtereinstellungen aktualisieren. Bei Tabs ohne benutzerdefinierten Zeitraum kann der Zeitbereich über die Schaltfläche zur Zeitauswahl oben rechts geändert werden (der Schaltflächenname hängt vom ausgewählten Zeitintervall ab: Diese Woche, Letzte 30 Minuten, Gestern usw.). Diese Option ist nur für Filter unter *Monitoring* > *Problems* verfügbar.
From/To	Zeitraum Anfang und Ende in absoluter (`Y-m-d H:i:s`) oder relativer Zeitsyntax (`now-1d`). Verfügbar, wenn *Set custom time period* aktiviert ist.

Nach dem Speichern wird der Filter als benannter Filter-Tab erstellt und sofort aktiviert.

Um die Filtereigenschaften eines vorhandenen Filters zu bearbeiten, klicken Sie auf das Zahnradsymbol neben dem Namen des aktiven Tabs.



Hinweise:

- Um den Filterbereich auszublenden, klicken Sie auf den Namen des aktuellen Tabs. Klicken Sie erneut auf den Namen des aktiven Tabs, um den Filterbereich wieder zu öffnen.
- Die Tastaturnavigation wird unterstützt: Verwenden Sie die Pfeiltasten, um zwischen den Tabs zu wechseln, und drücken Sie *Enter*, um zu öffnen.
- Die Schaltflächen links/rechts oberhalb des Filters können verwendet werden, um zwischen gespeicherten Filtern zu wechseln. Alternativ öffnet die nach unten zeigende Schaltfläche ein Dropdown-Menü mit allen gespeicherten Filtern, und Sie können den gewünschten anklicken.
- Wenn die Einstellungen eines gespeicherten Filters geändert wurden (aber nicht gespeichert sind), wird nach dem Filternamen ein grüner Punkt angezeigt. Um den Filter entsprechend den neuen Einstellungen zu aktualisieren, klicken Sie auf die Schaltfläche *Update*, die anstelle der Schaltfläche *Save as* angezeigt wird.
- Das Spaltenlayout der Tabelle wird pro Filter-Tab gespeichert. Um das Layout eines gespeicherten Filters anzupassen, öffnen Sie zuerst diesen Filter-Tab und passen Sie dann die Spalten an.
- Filter-Tabs, Filtereinstellungen und das Spaltenlayout der Tabelle für jeden Tab werden im Benutzerprofil gespeichert und wiederhergestellt, wenn der Benutzer die Seite erneut öffnet.

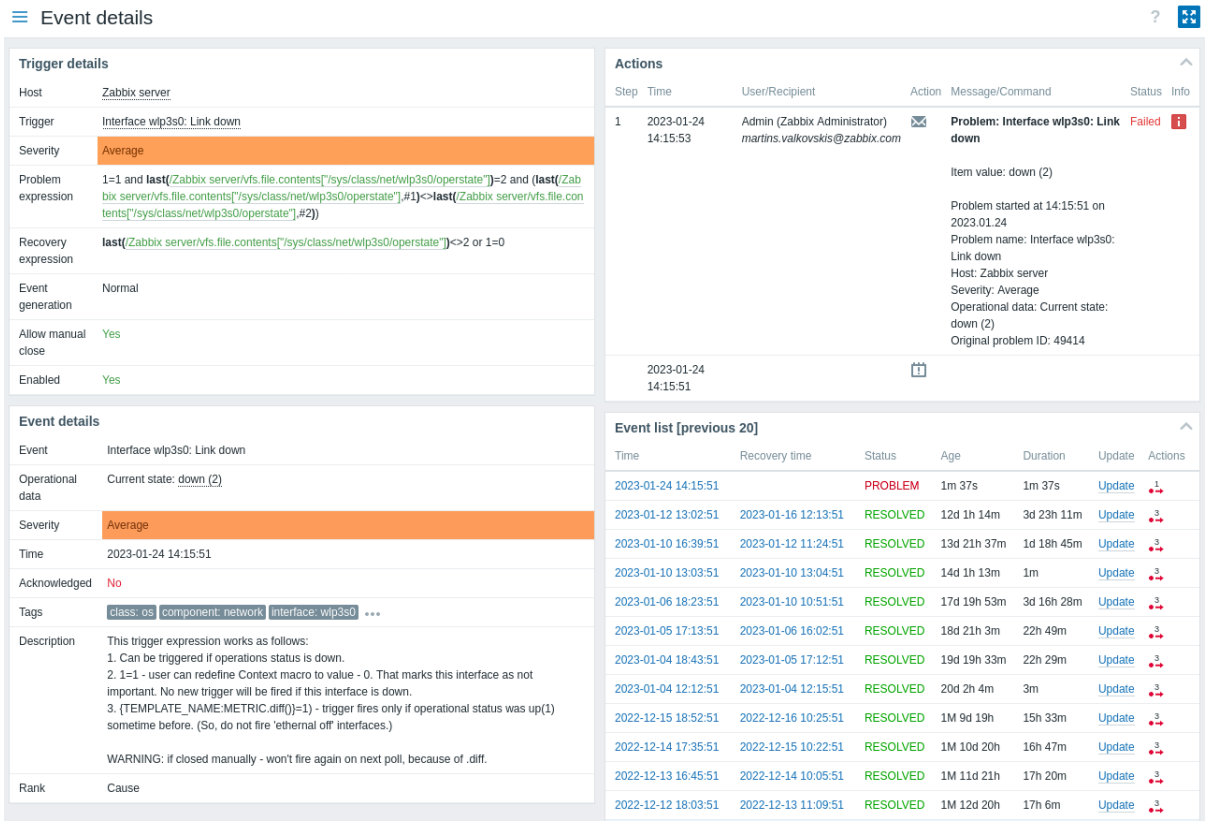
Note:
Um Filter freizugeben, kopieren Sie die URL eines aktiven Filters und senden Sie sie an andere. Nach dem Öffnen dieser URL können andere Benutzer diesen Parametersatz als permanenten Filter in ihrem Zabbix-Konto speichern.
 Siehe auch: Page parameters.

Filterschaltflächen

Apply	Angegebene Filterkriterien anwenden (ohne zu speichern).
Reset	Aktuellen Filter zurücksetzen und zu den gespeicherten Parametern der aktuellen Registerkarte zurückkehren. Auf der Hauptregisterkarte wird dadurch der Filter gelöscht.
Save as	Aktuelle Filterparameter in einer neuen Registerkarte speichern. Nur auf der Hauptregisterkarte verfügbar.
Update	Parameter der Registerkarte durch die aktuell angegebenen Parameter ersetzen. Auf der Hauptregisterkarte nicht verfügbar.

Details anzeigen

Die Zeiten für den Beginn und die Behebung eines Problems unter *Monitoring* > *Probleme* sind Links. Wenn Sie darauf klicken, werden weitere Details zum Ereignis geöffnet.



In den Ereignisdetails können Sie sehen, ob das Ereignis aufgrund von Wartung unterdrückt wurde oder ob es von einem Benutzer manuell unterdrückt wurde.

Beachten Sie, dass sich die Problem-Schwere für den Auslöser und das Problemereignis unterscheiden kann - wenn sie für das Problemereignis über den Bildschirm *Problem aktualisieren* geändert wurde.

In der Aktionsliste werden die folgenden Symbole verwendet, um den Aktivitätstyp zu kennzeichnen:

- - Problemereignis erzeugt.
- - Nachricht wurde gesendet.
- - Problemereignis bestätigt.
- - Problemereignis nicht bestätigt.
- - ein Kommentar wurde hinzugefügt.
- - die Problem-Schwere wurde erhöht (z. B. Information > Warning).
- - die Problem-Schwere wurde verringert (z. B. Warning > Information).

-  - die Problem-Schwere wurde geändert, ist jedoch auf die ursprüngliche Stufe zurückgekehrt (z. B. Warning > Information > Warning).
-  - ein Remote-Befehl wurde ausgeführt.
-  - Problemereignis wurde behoben.
-  - das Problem wurde manuell geschlossen.
-  - das Problem wurde unterdrückt.
-  - die Unterdrückung des Problems wurde aufgehoben.
-  - das Problem wurde aufgrund von Host-Wartung unterdrückt.
-  - die Unterdrückung des Problems wurde nach der Host-Wartung aufgehoben.
-  - das Problem wurde in ein Symptom-Problem umgewandelt.
-  - das Problem wurde in ein Ursache-Problem umgewandelt.

Ursache- und Symptomprobleme

Übersicht

Standardmäßig werden alle neuen Probleme als Ursacheprobleme klassifiziert. Es ist möglich, bestimmte Probleme manuell als Symptomprobleme eines Ursacheproblems neu zu klassifizieren.

Beispielsweise kann ein Stromausfall die eigentliche Grundursache dafür sein, dass ein Host nicht erreichbar ist oder ein Dienst ausgefallen ist. In diesem Fall müssen die Probleme „Host ist nicht erreichbar“ und „Dienst ist ausgefallen“ als Symptomprobleme von „Stromausfall“ – dem Ursacheproblem – klassifiziert werden.

Die Ursache-Symptom-Hierarchie unterstützt nur zwei Ebenen. Ein Problem, das bereits ein Symptom ist, kann keine „untergeordneten“ Symptomprobleme zugewiesen bekommen; alle Probleme, die einem Symptomproblem als Symptome zugewiesen werden, werden zu Symptomen desselben Ursacheproblems.

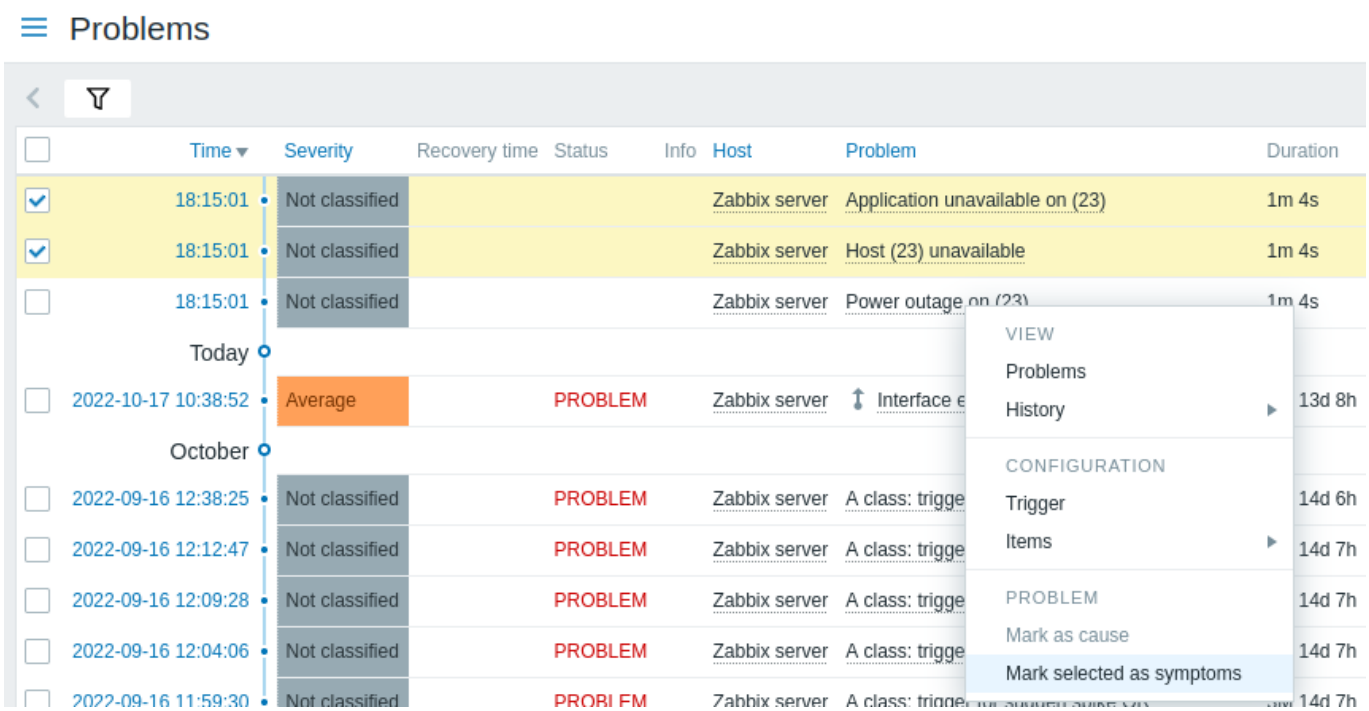
Nur Ursacheprobleme werden in den Problemgesamtzahlen auf Karten, in Dashboard-Widgets wie *Probleme nach Schweregrad* oder *Problem-Hosts* usw. gezählt. Die Problemrangfolge wirkt sich jedoch nicht auf Dienste aus.

Ein Symptomproblem kann nur mit einem Ursacheproblem verknüpft werden. Symptomprobleme werden nicht automatisch gelöst, wenn das Ursacheproblem gelöst oder geschlossen wird.

Konfiguration

Um ein Problem als Symptomproblem neu zu klassifizieren, wählen Sie es zunächst in der Liste der Probleme aus. Es können ein oder mehrere Probleme ausgewählt werden.

Gehen Sie dann zum Ursacheproblem und klicken Sie in dessen Kontextmenü auf die Option *Mark selected as symptoms*.



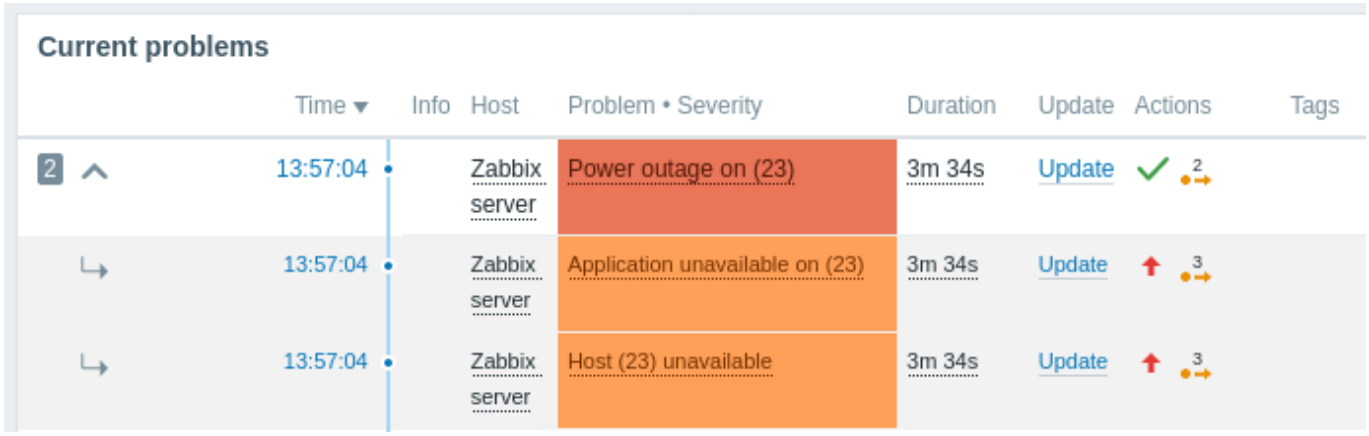
Danach werden die ausgewählten Probleme vom Server zu Symptomproblemen des Ursacheproblems aktualisiert.

Während der Status des Problems aktualisiert wird, wird er auf eine von zwei Arten angezeigt:

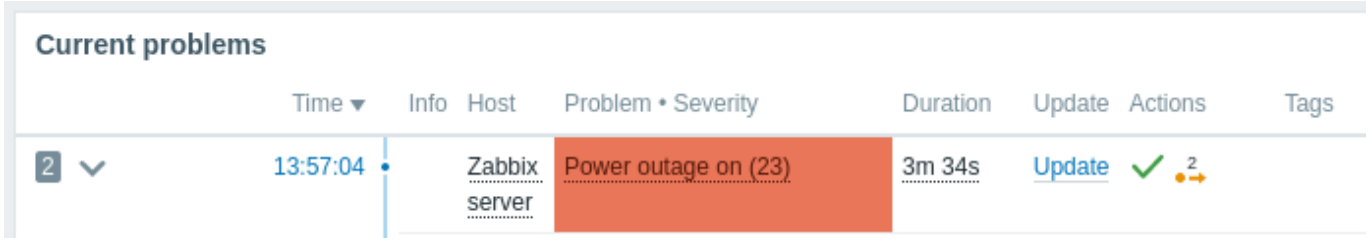
- In der Spalte Status wird ein blinkender Status ”UPDATING” angezeigt;
- In der Spalte Info wird ein blinkendes Symbol oder angezeigt (dies gilt, wenn im Filter nur *Problems* ausgewählt sind und daher die Spalte Status nicht angezeigt wird).

Anzeige

Symptomprobleme werden unterhalb des Ursache-Problems angezeigt und in *Monitoring* > *Probleme* (sowie im *Probleme*-Dashboard-Widget) entsprechend markiert - mit einem Symbol, kleinerer Schrift und anderem Hintergrund.



In der eingeklappten Ansicht ist nur das Ursache-Problem sichtbar; das Vorhandensein von Symptomproblemen wird durch die Zahl am Anfang der Zeile und das Symbol zum Erweitern der Ansicht angezeigt.



Es ist außerdem möglich, Symptomprobleme zusätzlich in normaler Schrift und in einer eigenen Zeile anzuzeigen. Wählen Sie dazu in den Filtereinstellungen oder in der Widget-Konfiguration *Symptome anzeigen* aus.

Rückkehr zum Ursache-Problem

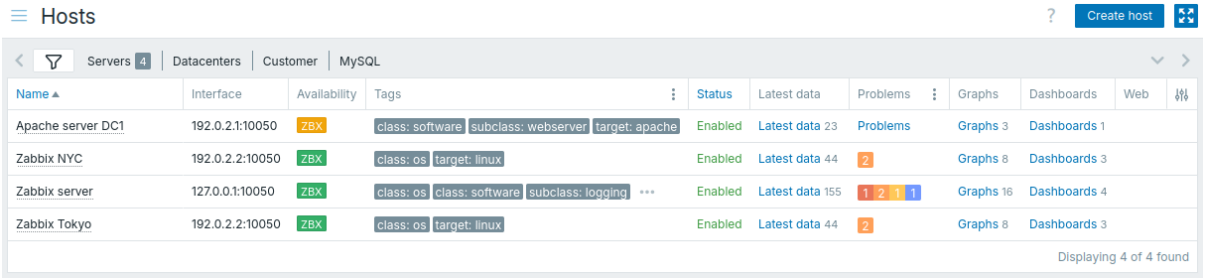
Ein Symptom-Problem kann wieder in ein Ursache-Problem zurückgesetzt werden. Dazu können Sie entweder:

- die Option *Als Ursache markieren* im Kontextmenü des Symptom-Problems anklicken;
- auf dem Bildschirm Problemaktualisierung die Option *In Ursache umwandeln* markieren und auf *Aktualisieren* klicken (diese Option funktioniert auch, wenn mehrere Probleme ausgewählt sind).

Hosts

Übersicht

Der Abschnitt *Monitoring* > *Hosts* zeigt eine vollständige Liste der überwachten Hosts mit detaillierten Informationen über Host-Schnittstelle, Verfügbarkeit, Tags, aktuelle Probleme, Status (aktiviert/deaktiviert) sowie Links, über die Sie einfach zu den neuesten Daten des Hosts, zur Problemhistorie, zu Diagrammen, Dashboards und Web-Szenarien navigieren können.



Die Host-Details werden in einer anpassbaren Tabelle angezeigt. Die Schaltfläche *Tabelle anpassen* in der Tabellenkopfzeile ermöglicht es Ihnen, Spalten ein- oder auszublenden, ihre Reihenfolge zu ändern und das Layout auf den Standard zurückzusetzen. Sie können außerdem jede Spalte in der Breite anpassen, indem Sie ihren rechten Rand ziehen; ein Doppelklick auf den Rand setzt die Breite zurück. Alle Anpassungen werden sofort übernommen und für den aktuellen Benutzer innerhalb der aktiven Filter-Registerkarte (Standard oder gespeichert) dauerhaft gespeichert.

Die folgenden Spalten sind verfügbar:

Column	Description
Name	Der sichtbare Hostname. Ein Klick auf den Namen öffnet das Host-Menü. Ein orangefarbenes Schraubenschlüssel-Symbol hinter dem Namen zeigt an, dass sich dieser Host in Wartung befindet. Klicken Sie auf die Spaltenüberschrift, um Hosts nach Namen auf- oder absteigend zu sortieren.
Interface	Die Hauptschnittstelle des Hosts wird angezeigt.

Column	Description
Availability	Die Verfügbarkeit des Hosts pro konfigurierter Schnittstelle wird angezeigt. Verfügbarkeits-Symbole stellen den aktuellen Status der Host-Schnittstelle auf dem Zabbix Server dar. Wenn Sie daher einen Host im Frontend deaktivieren, wird seine Verfügbarkeit erst aktualisiert, nachdem der Zabbix Server die Konfigurationsänderungen synchronisiert hat. Entsprechend wird bei Aktivierung eines Hosts seine Verfügbarkeit erst aktualisiert, nachdem der Zabbix Server die Konfigurationsänderungen synchronisiert und den Host abgefragt hat. Verfügbarkeits-Symbole stellen nur die konfigurierten Schnittstellentypen (Agent, SNMP, IPMI, JMX) dar. Wenn Sie mit der Maus über das Symbol fahren, wird ein Pop-up mit einer Liste aller Schnittstellen desselben Typs samt Details, Status und Fehlern angezeigt. Für die Agent-Schnittstelle zeigt das Pop-up Schnittstellen (passiv) und aktive Prüfungen an. Wenn ein Host nur aktive Prüfungen hat, wird das Symbol für die Agent-Schnittstelle angezeigt, auch wenn für den Host keine Agent-Schnittstelle konfiguriert ist. Die Spalte ist für Hosts ohne Schnittstellen leer. Der Status einer einzelnen Host-Schnittstelle wird durch die Verbindung zwischen einem aktivierten Datenpunkt, der die Schnittstelle verwendet, und dem Host bestimmt. Der Status kann sein: **Verfügbar** - die Verbindung zum Host war erfolgreich; **Nicht verfügbar** - die Verbindung zum Host war nicht erfolgreich (Timeout, Firewall-Probleme usw.); **Unbekannt** - es wurde kein Verbindungsversuch zum Host unternommen oder das Ergebnis ist unbekannt. Weitere Details dazu, wie der Zabbix Server den Schnittstellenstatus bestimmt, finden Sie unter Unbekannter Schnittstellenstatus und Einstellungen für nicht erreichbare/nicht verfügbare Host-Schnittstellen. Der Status aller Host-Schnittstellen eines Typs (Agent, SNMP, IPMI, JMX) wird durch jene Schnittstellen bestimmt, die von mindestens einem aktivierten Datenpunkt verwendet werden. Der Status wird durch die Symbolfarbe angezeigt: **Grün** - alle Schnittstellen sind verfügbar; **Gelb** - mindestens eine Schnittstelle ist nicht verfügbar und mindestens eine ist verfügbar oder unbekannt; **Rot** - alle Schnittstellen sind nicht verfügbar; **Grau** - mindestens eine Schnittstelle ist unbekannt, aber keine ist nicht verfügbar. **Verfügbarkeit aktiver Prüfungen.** Wenn auf dem Host mindestens eine aktive Prüfung aktiviert ist, wirken sich aktive Prüfungen ebenfalls auf die gesamte Verfügbarkeit der Agent-Schnittstelle aus, wie oben beschrieben. Zur Bestimmung der Verfügbarkeit aktiver Prüfungen werden Heartbeat-Nachrichten im Thread für aktive Prüfungen des Agent gesendet. Die Häufigkeit der Heartbeat-Nachrichten wird durch den Parameter `HeartbeatFrequency` in der Konfiguration von Zabbix Agent oder agent 2 gesteuert (Standard 60 Sekunden, Bereich 0-3600). Aktive Prüfungen gelten als nicht verfügbar, wenn der aktive Prüfungs-Heartbeat älter als 2 x HeartbeatFrequency Sekunden ist. **Hinweis:** Zabbix Agenten älter als Version 6.2.x senden keine Heartbeats für aktive Prüfungen, daher bleibt die Verfügbarkeit ihrer Hosts unbekannt.
Tags	Host-Tags werden angezeigt. Wenn Sie mit der Maus über das Symbol mit den drei Punkten ••• hinter den angezeigten Tags fahren oder darauf klicken, wird ein Pop-up mit allen Tags geöffnet. Wenn Sie auf die Schaltfläche ⋮ *Optionen* in der Spaltenüberschrift klicken, können Sie die Anzahl der angezeigten Tags (1–3) festlegen, auswählen, wie Tag-Namen angezeigt werden (vollständig, gekürzt oder gar nicht), die Anzeigepriorität der Tags festlegen oder die Spalte duplizieren (duplizierte Spalten können umbenannt oder gelöscht werden).
Tag value	Host-Tag-Werte werden angezeigt. Wenn Sie auf die Schaltfläche ⋮ *Optionen* in der Spaltenüberschrift klicken, können Sie die Spalte umbenennen, auswählen, welche Tag-Werte angezeigt werden sollen, oder die Spalte duplizieren (duplizierte Spalten können umbenannt oder gelöscht werden).
Status	Host-Status - *Enabled* oder *Disabled*. Klicken Sie auf die Spaltenüberschrift, um Hosts nach Status auf- oder absteigend zu sortieren.

Column	Description
Latest data	Wenn Sie auf den Link klicken, wird der Abschnitt *Monitoring* > *Latest data* mit allen vom Host gesammelten neuesten Daten geöffnet. Die Anzahl der Datenpunkte mit neuesten Daten wird grau angezeigt.
Problems	Die Anzahl der offenen Probleme für einen Host wird nach Schweregrad sortiert angezeigt. Die Farbe jedes Problemsymbols zeigt den Schweregrad an, während die Zahl die Anzahl der Probleme für diesen Schweregrad angibt. Ein Klick auf ein Symbol öffnet *Monitoring* > *Problems* für diesen Host; wenn ein Host keine Probleme hat, wird stattdessen ein Link *Problems* angezeigt. Wenn Sie auf die Schaltfläche ⋮ *Optionen* in der Spaltenüberschrift klicken, erhalten Sie Zugriff auf die Option *Unterdrückte Probleme anzeigen*, die auch Probleme einschließt, die andernfalls aufgrund von Host-Wartung oder einzelner Problemunterdrückung unterdrückt (ausgeblendet) würden.
Graphs	Wenn Sie auf den Link klicken, werden die für den Host konfigurierten Diagramme angezeigt. Die Anzahl der Diagramme wird grau angezeigt. Wenn ein Host keine Diagramme hat, wird kein Link angezeigt.
Dashboards	Wenn Sie auf den Link klicken, werden die für den Host konfigurierten Dashboards angezeigt. Die Anzahl der Dashboards wird grau angezeigt. Wenn ein Host keine Dashboards hat, wird kein Link angezeigt.
Web	Wenn Sie auf den Link klicken, werden die für den Host konfigurierten Web-Szenarien angezeigt. Die Anzahl der Web-Szenarien wird grau angezeigt. Wenn ein Host keine Web-Szenarien hat, wird kein Link angezeigt.
Custom text	Benutzerdefinierter Text, Makrowerte oder eine Kombination aus beidem wird angezeigt. Wenn Sie auf die Schaltfläche ⋮ *Optionen* in der Spaltenüberschrift klicken, können Sie die Spalte umbenennen, den benutzerdefinierten Text oder Makros eingeben oder die Spalte duplizieren (duplizierte Spalten können umbenannt oder gelöscht werden). Unterstützte Makros: `{HOST.*}` (außer `{HOST.METADATA}`), `{INVENTORY.*}` und Benutzermakros.

Schaltflächen

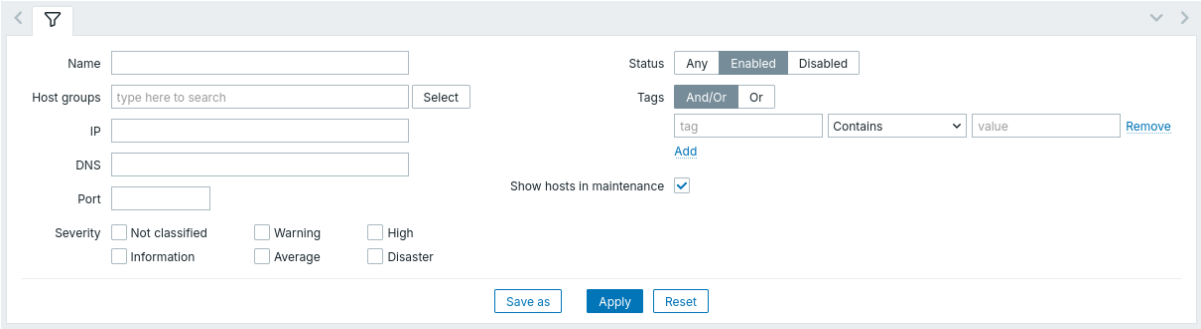
Host erstellen ermöglicht Ihnen, einen neuen Host zu erstellen. Diese Schaltfläche ist nur für Benutzer mit den Rollen „Admin“ und „Super Admin“ verfügbar.

Die Schaltflächen des Ansichtsmodus, die für alle Bereiche gelten, werden auf der Seite Überwachung beschrieben.

Verwendung des Filters

Sie können den Filter verwenden, um nur die Hosts anzuzeigen, die Sie interessieren. Für eine bessere Suchleistung werden Daten mit nicht aufgelösten Makros durchsucht.

Der Filter befindet sich oberhalb der Tabelle. Hosts können nach Name, Host-Gruppe, IP oder DNS, Schnittstellenport, Tags, Problem-Schweregrad, Status (aktiviert/deaktiviert/alle) gefiltert werden; außerdem können Sie auswählen, ob Hosts angezeigt werden sollen, die sich derzeit in Wartung befinden.



Parameter	Beschreibung
Name	Nach dem sichtbaren Host-Namen filtern.
Host groups	Nach einer oder mehreren Host-Gruppen filtern. Die Angabe einer übergeordneten Host-Gruppe wählt implizit alle verschachtelten Host-Gruppen aus.
IP	Nach IP-Adresse filtern.
DNS	Nach DNS-Namen filtern.
Port	Nach Portnummer filtern.

Parameter	Beschreibung
Severity	Nach Problem-Schweregrad filtern. Standardmäßig werden Probleme aller Schweregrade angezeigt. Probleme werden angezeigt, wenn sie nicht unterdrückt sind.
Status	Nach Host-Status filtern.
Tags	Nach Host-Tag-Namen und -Wert filtern. Hosts können sowohl nach Host-Ebene-Tags als auch nach Tags aus allen verknüpften Vorlagen, einschließlich verschachtelter Vorlagen, gefiltert werden. Es ist möglich, bestimmte Tags und Tag-Werte einzuschließen sowie auszuschließen. Es können mehrere Bedingungen festgelegt werden. Die Übereinstimmung des Tag-Namens ist immer groß-/kleinschreibungssensitiv. Für jede Bedingung stehen mehrere Operatoren zur Verfügung: **Exists** - die angegebenen Tag-Namen einschließen **Equals** - die angegebenen Tag-Namen und -Werte einschließen (groß-/kleinschreibungssensitiv) **Contains** - die angegebenen Tag-Namen einschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv) **Does not exist** - die angegebenen Tag-Namen ausschließen **Does not equal** - die angegebenen Tag-Namen und -Werte ausschließen (groß-/kleinschreibungssensitiv) **Does not contain** - die angegebenen Tag-Namen ausschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv) Für Bedingungen gibt es zwei Berechnungstypen: **And/Or** - alle Bedingungen müssen erfüllt sein, Bedingungen mit demselben Tag-Namen werden durch die Or-Bedingung gruppiert **Or** - es genügt, wenn eine Bedingung erfüllt ist
Show hosts in maintenance	Aktivieren Sie das Kontrollkästchen, um Hosts anzuzeigen, die sich in Wartung befinden (standardmäßig angezeigt).

Filter speichern

Favorisierte Filtereinstellungen können als Registerkarten gespeichert und dann schnell über einen Klick auf die jeweilige Registerkarte oberhalb des Filters aufgerufen werden.

Weitere Details zum Speichern von Filtern.

Diagramme

Übersicht

Host-Diagramme können unter *Monitoring* > *Hosts* durch Klicken auf Diagramme für den jeweiligen Host aufgerufen werden.

Jedes für den Host konfigurierte benutzerdefinierte Diagramm kann angezeigt werden, ebenso wie jedes einfache Diagramm.

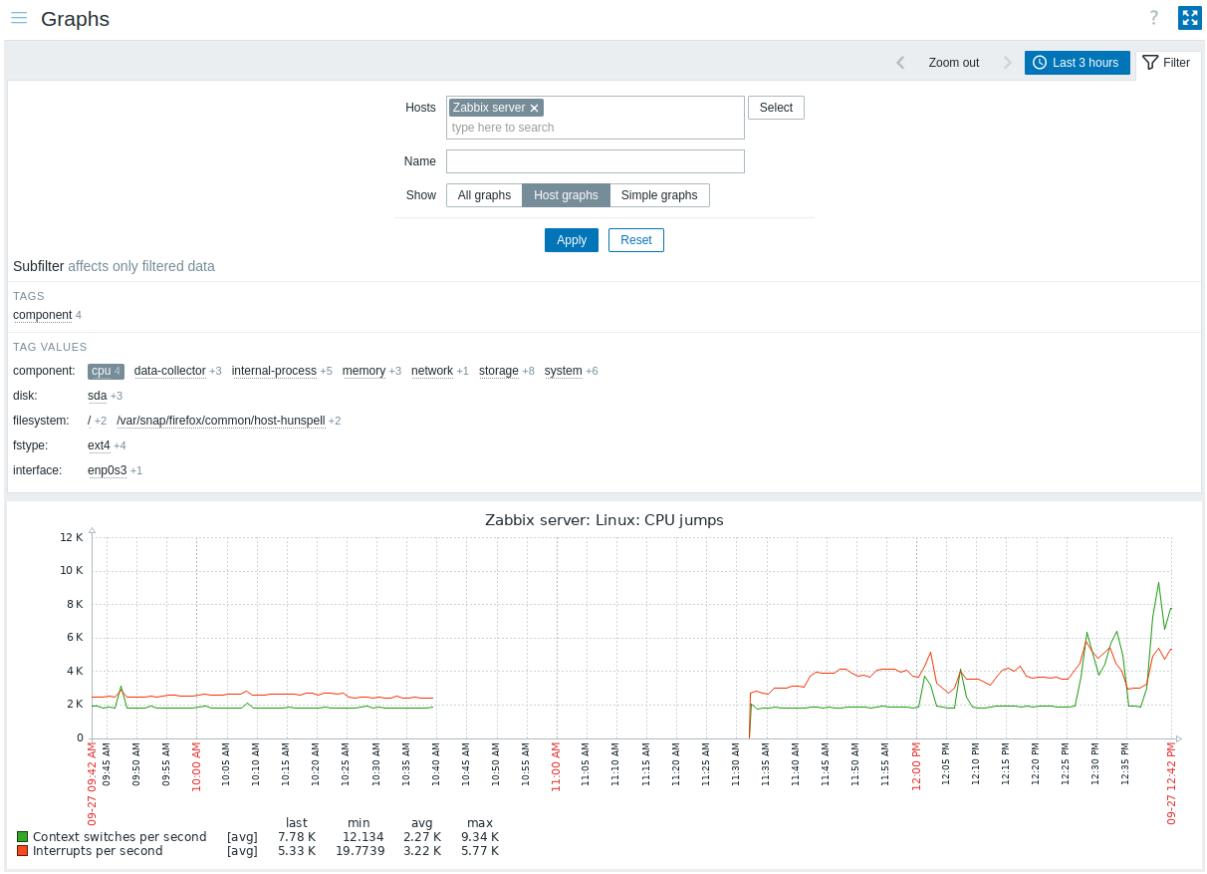


Diagramme werden sortiert nach:

- Diagrammname (benutzerdefinierte Diagramme)
- Datenpunktname (einfache Diagramme)

Diagramme für deaktivierte Hosts sind ebenfalls zugänglich.

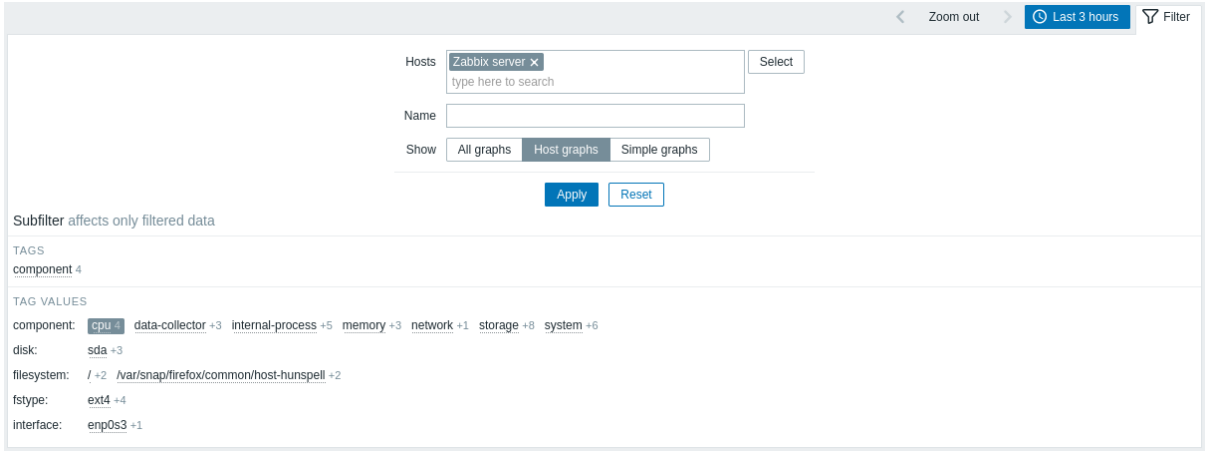
Auswahl des Zeitraums

Beachten Sie die Auswahl des Zeitraums oberhalb des Diagramms. Sie ermöglicht die Auswahl häufig benötigter Zeiträume mit einem Mausklick.

Siehe auch: *Auswahl des Zeitraums*

Filter verwenden

Um ein bestimmtes Diagramm anzuzeigen, wählen Sie es im Filter aus. Im Filter können Sie den Host, den Namen des Diagramms und die Option *Anzeigen* (alle/Host-Diagramme/einfache Diagramme) angeben.



Wenn im Filter kein Host ausgewählt ist, werden keine Diagramme angezeigt.

Subfilter verwenden

Der Subfilter ermöglicht eine weitere Anpassung der durch den Hauptfilter angewendeten Filterung.

Er enthält anklickbare Links für den schnellen Zugriff auf zugehörige Diagramme. Diagramme werden durch eine gemeinsame Entität – den Tag-Namen oder -Wert – miteinander verknüpft. Wenn ein Tag-Name oder -Wert angeklickt wird, wird er mit einem grauen Hintergrund hervorgehoben und die Diagramme werden sofort gefiltert (im Hauptfilter muss nicht auf *Anwenden* geklickt werden). Durch das Anklicken eines weiteren Tag-Namens oder -Werts wird dieser zu den gefilterten Ergebnissen hinzugefügt. Durch erneutes Anklicken des Tag-Namens oder -Werts wird die Filterung entfernt.

Subfilter werden auf Grundlage der gefilterten Daten erstellt, die auf 1000 Datensätze begrenzt sind. Wenn Sie mehr Datensätze im Subfilter anzeigen möchten, müssen Sie den Wert des Parameters *Limit for search and filter results* unter *Administration* > *General* > *GUI* erhöhen.

Im Gegensatz zum Hauptfilter wird der Subfilter bei jeder Aktualisierungsanforderung der Tabelle aktualisiert, damit stets aktuelle Informationen zu den verfügbaren Filteroptionen und deren Anzahl angezeigt werden.

Die Anzahl der horizontal angezeigten Entitäten ist auf 100 begrenzt. Wenn weitere vorhanden sind, wird am Ende ein Symbol mit drei Punkten angezeigt; es ist nicht anklickbar. Vertikale Listen (z. B. Tags mit ihren Werten) sind auf 20 Einträge begrenzt. Wenn weitere vorhanden sind, wird ein Symbol mit drei Punkten angezeigt; es ist nicht anklickbar.

Eine Zahl neben jeder anklickbaren Entität gibt an, wie viele Diagramme sie in den Ergebnissen des Hauptfilters enthält.

Sobald eine Entität ausgewählt wurde, werden die Zahlen neben den anderen verfügbaren Entitäten mit einem Pluszeichen angezeigt. Dieses gibt an, wie viele Diagramme zur aktuellen Auswahl hinzugefügt werden können.

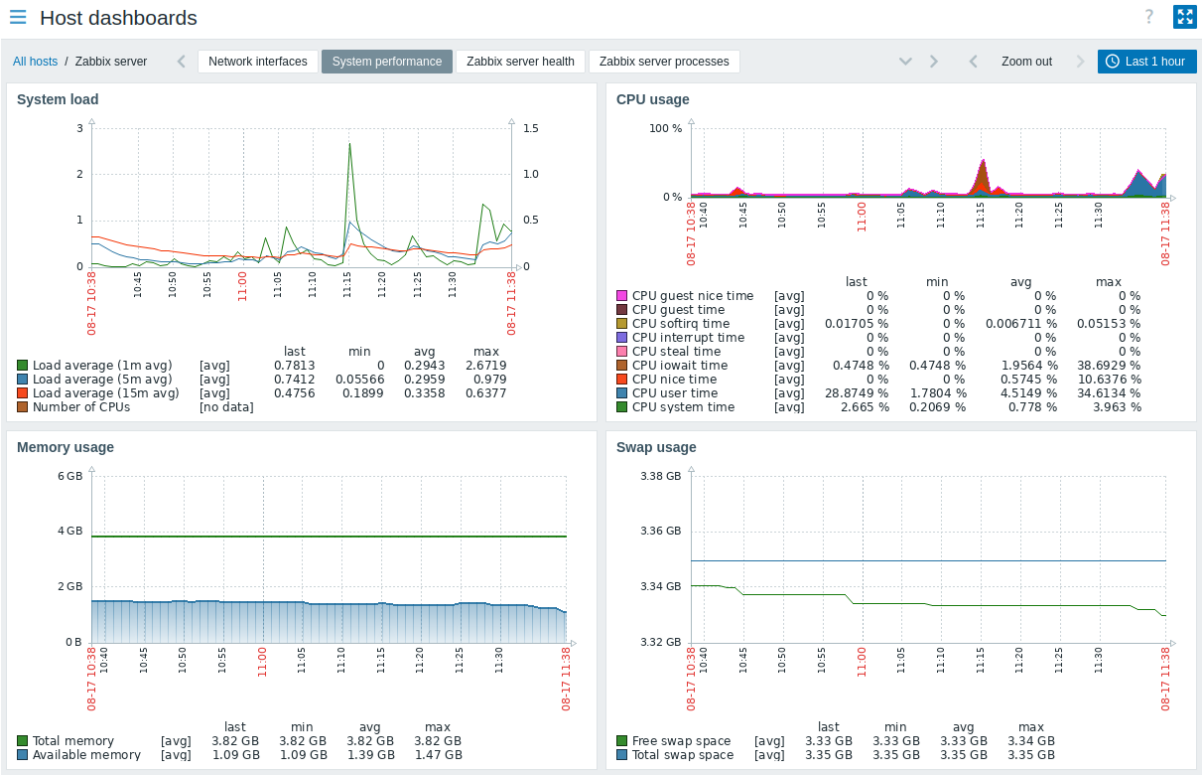
Schaltflächen

Die Schaltflächen des Ansichtsmodus, die in allen Bereichen gemeinsam verwendet werden, sind auf der Seite Monitoring beschrieben.

Host-Dashboards

Übersicht

Host-Dashboards ähneln globalen Dashboards; Host-Dashboards verfügen jedoch über keinen Besitzer und zeigen Daten nur für den ausgewählten Host an.



Beim Anzeigen von Host-Dashboards können Sie zwischen den konfigurierten Dashboards wechseln, indem Sie auf Folgendes klicken:

- die Dashboard-Registerkarten;
- die Pfeilschaltflächen unter der Kopfzeile;

- die Pfeilschaltfläche unter der Kopfzeile, über die die vollständige Liste der verfügbaren Host-Dashboards angezeigt wird.

Um zum Abschnitt *Monitoring* > *Hosts* zu wechseln, klicken Sie auf den Navigationslink *Alle Hosts* unter der Kopfzeile in der oberen linken Ecke.

Konfiguration

Host-Dashboards werden auf der Ebene der Vorlage konfiguriert. Sobald eine Vorlage mit einem Host verknüpft ist, werden für diesen Host Host-Dashboards erzeugt. Beachten Sie, dass Host-Dashboards *nicht* im Abschnitt *Dashboards* konfiguriert werden können, da dieser globalen Dashboards vorbehalten ist.

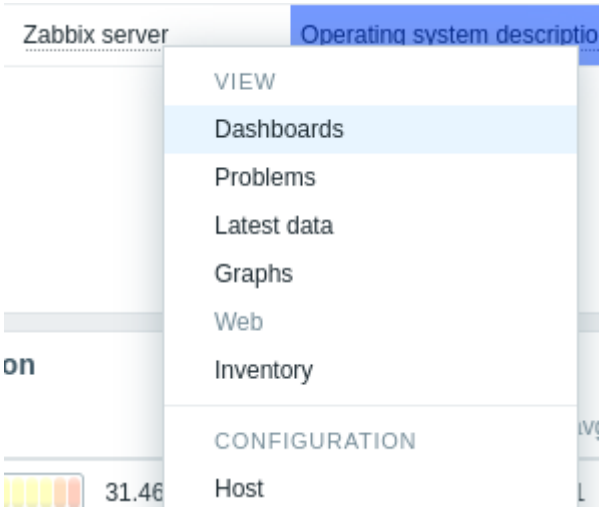
Widgets von Host-Dashboards können ebenfalls nur auf der Ebene der Vorlage konfiguriert werden, mit Ausnahme der Änderung des Aktualisierungsintervalls. Außerdem können Widgets von Host-Dashboards nur in andere Host-Dashboards innerhalb derselben Vorlage kopiert werden. Beachten Sie, dass Widgets aus globalen Dashboards *nicht* in Host-Dashboards kopiert werden können.

Note:
Host-Dashboards waren vor Zabbix 5.2 Host-Bildschirme. Beim Import einer älteren Vorlage, die Bildschirme enthält, wird der Bildschirmimport ignoriert.

Zugriff

Host-Dashboards können wie folgt aufgerufen werden:

- nach der Suche nach einem Hostnamen in der globalen Suche (klicken Sie auf den Link *Dashboards* in den Suchergebnissen);
- nach dem Klicken auf einen Hostnamen in *Inventory* > *Hosts* (klicken Sie auf den Link *Dashboards* in der Host-Übersicht);
- über das Host-Menü, indem Sie auf *Dashboards* klicken.

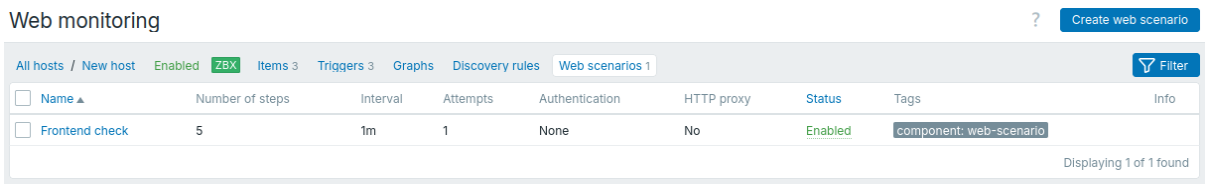


Beachten Sie, dass auf Host-Dashboards im Abschnitt *Dashboards* nicht direkt zugegriffen werden kann, da dieser für globale Dashboards reserviert ist.

Web-Szenarien

Übersicht

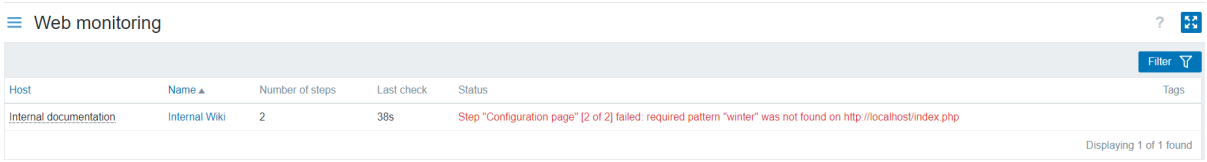
Auf Informationen zu einem Host Web-Szenario kann über *Monitoring* > *Hosts* zugegriffen werden, indem für den jeweiligen Host auf Web geklickt wird.



Ein Klick auf den Host-Namen öffnet das Host-Menü. Daten deaktivierter Hosts sind ebenfalls zugänglich. Der Name eines deaktivierten Hosts wird rot angezeigt.

Der Status eines Web-Szenarios kann sein:

- OK (grün) - alle Schritte waren erfolgreich, und Zabbix hat Daten für das Szenario erfasst.
- Fehler (rot) - ein oder mehrere Schritte sind fehlgeschlagen (siehe die Fehlermeldung, um Hinweise darauf zu erhalten, welche Parameter möglicherweise korrigiert werden müssen).
- Unbekannt (grau) - es ist noch kein Schrittstatus verfügbar oder es wurden noch keine Daten erfasst.



Die maximale Anzahl der pro Seite angezeigten Szenarien hängt von der Benutzereinstellung *Zeilen pro Seite* im Benutzerprofil ab.

Standardmäßig werden nur Werte angezeigt, die innerhalb der letzten 24 Stunden liegen. Diese Begrenzung wurde eingeführt, um die anfänglichen Ladezeiten für große Seiten mit aktuellen Daten zu verbessern. Sie können diesen Zeitraum verlängern, indem Sie den Wert des Parameters *Max history display period* im Menübereich *Administration > Allgemein > GUI* ändern.

Der Name des Szenarios ist ein Link zu detaillierteren Statistiken dazu:



Verwenden des Filters

Die Seite zeigt eine Liste aller Webszenarien des ausgewählten Hosts. Um Webszenarien für einen anderen Host oder eine andere Hostgruppe anzuzeigen, ohne zur Seite *Monitoring > Hosts* zurückzukehren, wählen Sie diesen Host oder diese Gruppe im Filter aus. Sie können Szenarien auch anhand von Tags filtern.

Schaltflächen

Die Schaltflächen des Ansichtsmodus, die in allen Abschnitten gemeinsam verwendet werden, sind auf der Seite Monitoring beschrieben.

Neueste Daten

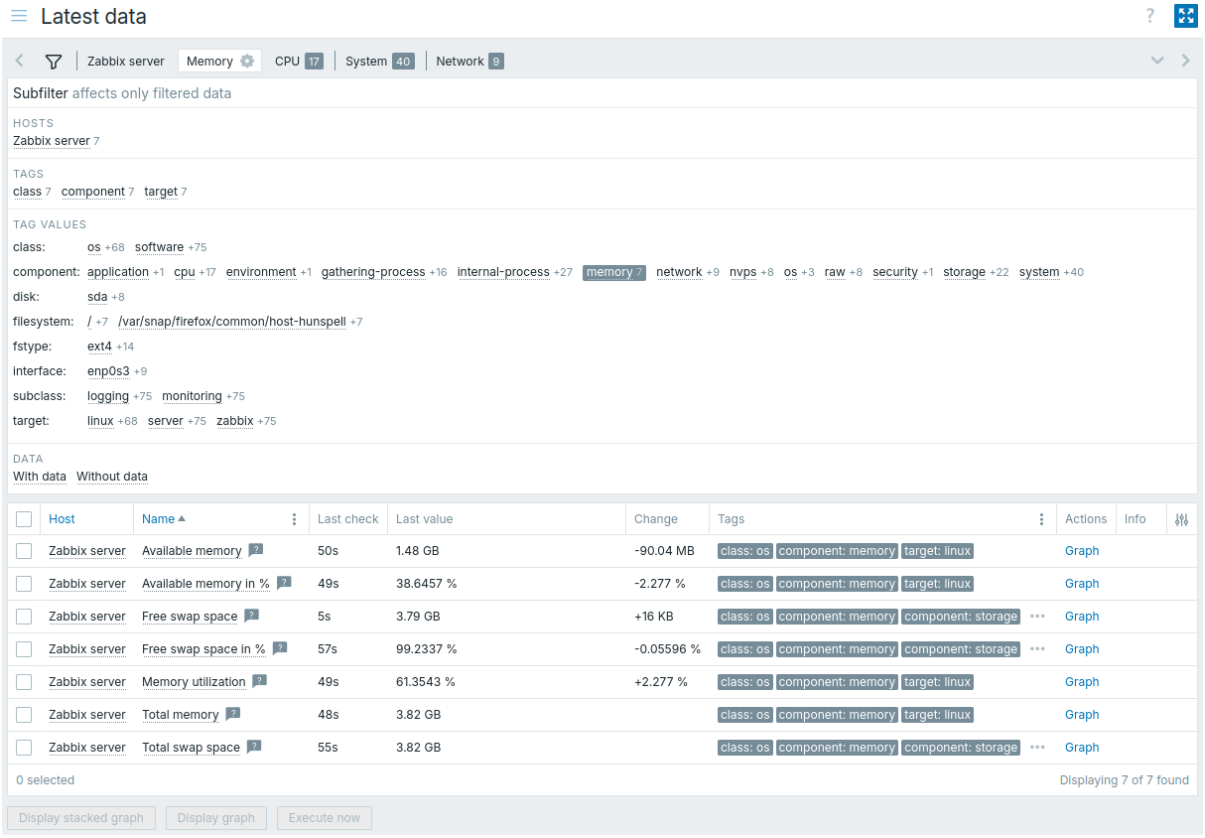
Übersicht

Der Abschnitt *Monitoring* > *Latest data* zeigt die neuesten von Datenpunkten erfassten Werte an.

Dieser Abschnitt enthält die folgenden Elemente:

- Filter
- Subfilter
- Datenpunktliste

Note:
Der Subfilter und die Datenpunktliste werden nur angezeigt, wenn der Filter gesetzt ist und Ergebnisse zur Anzeige vorhanden sind.



Details zum neuesten Wert werden in einer anpassbaren Tabelle angezeigt. Die Schaltfläche *Tabelle anpassen* in der Tabellenkopfzeile ermöglicht es Ihnen, Spalten ein- oder auszublenden, ihre Reihenfolge zu ändern und das Layout auf die Standardwerte zurückzusetzen. Sie können außerdem jede Spalte in der Breite anpassen, indem Sie ihren rechten Rand ziehen; ein Doppelklick auf den Rand setzt die Breite zurück. Alle Anpassungen werden sofort angewendet und für den aktuellen Benutzer innerhalb des aktiven Filter-Tabs (Standard oder gespeichert) dauerhaft gespeichert.

Die folgenden Spalten sind verfügbar:

Column	Description
Host	Name des Hosts, zu dem der Datenpunkt gehört. Durch Klicken auf den Namen wird das Host-Kontextmenü geöffnet. Wenn sich ein Host in Wartung befindet, wird nach dem Hostnamen ein orangefarbenes Schraubenschlüssel-Symbol angezeigt. Wenn ein Host deaktiviert ist, wird der Name des Hosts rot dargestellt. Beachten Sie, dass auf die Daten deaktivierter Hosts (einschließlich Diagramme und Datenpunkt-Wertelisten) im Abschnitt *Latest data* zugegriffen werden kann.

Column	Description
Name	Name des Datenpunkts. Durch Klicken auf den Namen wird das Datenpunktmenü geöffnet. Ein Fragezeichen-Symbol wird neben dem Datenpunktnamen für alle Datenpunkte angezeigt, die eine Beschreibung haben. Bewegen Sie den Mauszeiger über das Symbol, um einen Tooltip mit der Datenpunktbeschreibung anzuzeigen. Durch Klicken auf die Schaltfläche *Optionen* in der Spaltenkopfzeile erhalten Sie Zugriff auf die Option *Datenpunkt-Schlüssel anzeigen*, die den Datenpunkt-Schlüssel unter dem Namen jedes Datenpunkts anzeigt.
Interval	Aktualisierungsintervall des Datenpunkts.
History	Speicherzeitraum für die Historie des Datenpunkts.
Trends	Speicherzeitraum für Trends des Datenpunkts.
Type	Typ des Datenpunkts.
Last check	Zeit seit der letzten Prüfung des Datenpunkts.
Last value	Aktuellster Wert des Datenpunkts. Werte werden mit angewendeter Einheitumrechnung und Wertzuordnung angezeigt. Bewegen Sie den Mauszeiger über den Wert, um die Rohdaten anzuzeigen (gekürzt auf 8192 Zeichen). Binärwerte werden anstelle des tatsächlichen Werts als Platzhalter (*binary value*) angezeigt. Standardmäßig werden nur Werte angezeigt, die in den letzten 24 Stunden empfangen wurden. Diese Begrenzung verbessert die anfänglichen Ladezeiten bei großen *Latest data*-Seiten; um sie zu erweitern, ändern Sie den Wert des Parameters *Max history display period* unter *Administration* > *General* > *GUI*.
Change	Differenz zwischen dem vorherigen Wert und dem aktuellsten Wert. Bei Datenpunkten mit einer Aktualisierungshäufigkeit von 1 Tag oder mehr wird die Änderungsmenge (bei der Standardeinstellung) niemals angezeigt. In diesem Fall wird der letzte Wert überhaupt nicht angezeigt, wenn er vor mehr als 24 Stunden empfangen wurde.
Tags	Mit dem Datenpunkt verknüpfte Tags. Durch Klicken auf einen Tag wird dieser im Subfilter aktiviert, wodurch die Datenpunktliste nach diesem Tag (und allen anderen im Subfilter ausgewählten Tags) gefiltert wird. Nach dem Anklicken sind solche Tags nicht mehr anklickbar; um die Filterung nach Tags anzupassen, muss der Subfilter verwendet werden. Durch Klicken auf die Schaltfläche *Optionen* in der Spaltenkopfzeile können Sie die Anzahl der angezeigten Tags (1-3) festlegen, auswählen, wie Tag-Namen angezeigt werden (vollständig, gekürzt oder gar nicht), die Anzeigepriorität der Tags festlegen oder die Spalte duplizieren (duplizierte Spalten können umbenannt oder gelöscht werden).
Tag value	Werte des Tags des Datenpunkts werden angezeigt. Durch Klicken auf die Schaltfläche *Optionen* in der Spaltenkopfzeile können Sie die Spalte umbenennen, auswählen, welche Tag-Werte angezeigt werden sollen, oder die Spalte duplizieren (duplizierte Spalten können umbenannt oder gelöscht werden).
Actions	Link zu einfachen Diagrammen/Historie der Datenpunktwerte.
Info	Zusätzliche Informationen zum Datenpunkt. Wenn ein Datenpunkt Fehler aufweist (z. B. nicht mehr unterstützt wird), wird ein Informationssymbol angezeigt. Bewegen Sie den Mauszeiger über das Symbol, um Details anzuzeigen.
Custom text	Benutzerdefinierter Text, Makrowerte oder eine Kombination aus beidem wird angezeigt. Durch Klicken auf die Schaltfläche *Optionen* in der Spaltenkopfzeile können Sie die Spalte umbenennen, den benutzerdefinierten Text oder Makros eingeben oder die Spalte duplizieren (duplizierte Spalten können umbenannt oder gelöscht werden). Unterstützte Makros: `{HOST.*}` (außer `{HOST.METADATA}`), `{INVENTORY.*}`, `{ITEM.*}` (außer `{ITEM.STATE.ERROR}`) und Benutzermakros.

Schaltflächen

Die Schaltflächen des Ansichtsmodus, die in allen Abschnitten gemeinsam verwendet werden, sind auf der Seite Monitoring beschrieben.

Massenaktionen

Die Schaltflächen unter der Liste bieten Massenaktionen für einen oder mehrere ausgewählte Einträge:

- *Gestapeltes Diagramm anzeigen* - ein gestapeltes Ad-hoc-Diagramm anzeigen.
- *Diagramm anzeigen* - ein einfaches Ad-hoc-Diagramm anzeigen.

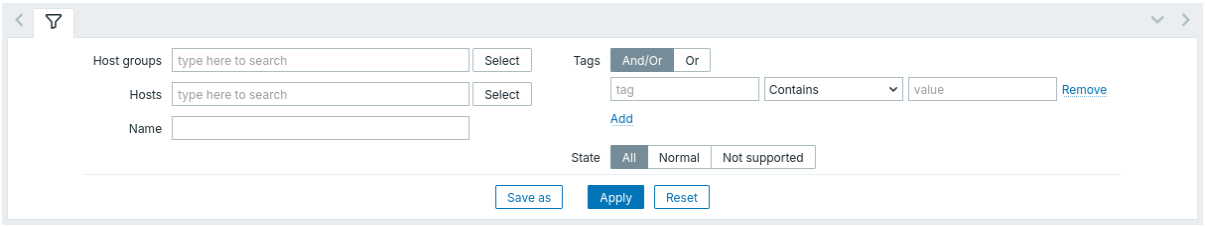
- *Jetzt ausführen* - eine Prüfung auf neue Datenpunktwerte sofort ausführen. Nur für **passive** Prüfungen unterstützt (siehe weitere Details). Diese Option ist nur für Hosts mit Lese-/Schreibzugriff verfügbar. Der Zugriff auf diese Option für Hosts mit schreibgeschützten Berechtigungen hängt von der Benutzerrolle-Option *"Jetzt ausführen" auf Hosts mit schreibgeschütztem Zugriff aufrufen* ab.

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Datenpunkten und klicken Sie dann auf die gewünschte Schaltfläche.

Filter verwenden

Sie können den Filter verwenden, um nur die Datenpunkte anzuzeigen, die Sie interessieren. Für eine bessere Suchleistung werden Daten mit nicht aufgelösten Makros durchsucht.

Das  -Symbol des Filters befindet sich über der Datenpunktliste und dem Unterfilter. Klicken Sie darauf, um den Filter zu erweitern.



Mit dem Filter können Sie die Datenpunktliste nach Hostgruppe, Host, Datenpunktname, Tags und Status eingrenzen. Wenn Sie eine übergeordnete Hostgruppe im Filter angeben, werden implizit alle verschachtelten Hostgruppen ausgewählt. Weitere Informationen zum Filtern nach Tags finden Sie unter *Überwachung* > *Probleme*.

Mit der Filteroption *Details anzeigen* können Sie die für die Datenpunkte angezeigten Informationen erweitern. Aktivieren Sie diese Option, um Details wie das Aktualisierungsintervall des Datenpunkts, die Einstellungen für Verlauf und Trends, den Datenpunkttyp sowie Fehler des Datenpunkts (fehlerhaft/nicht unterstützt) anzuzeigen.

Filter speichern

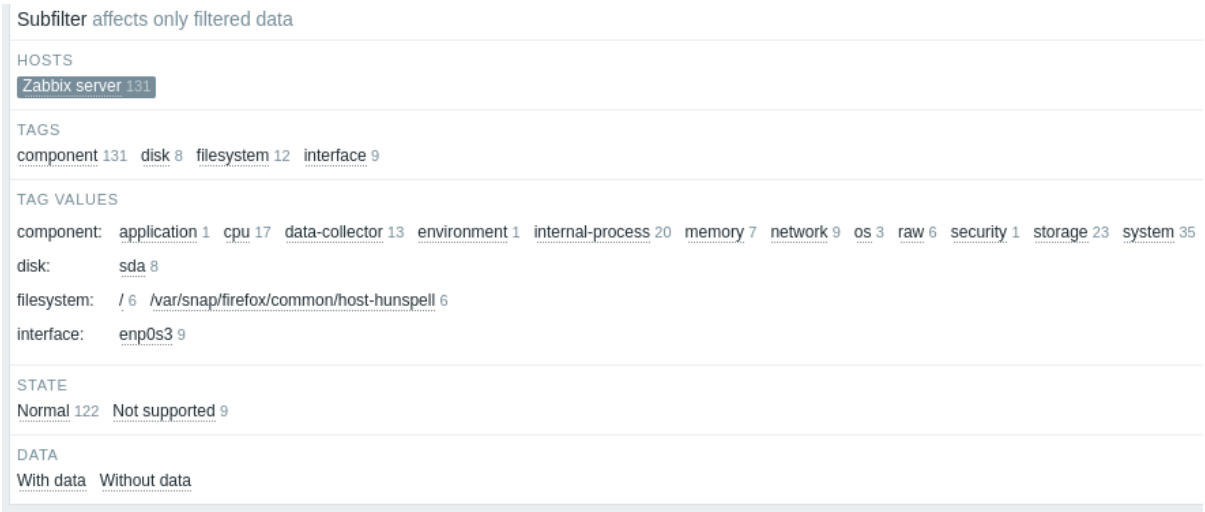
Die Einstellungen eines Favoritenfilters können als Registerkarten gespeichert und dann durch Klicken auf die jeweilige Registerkarte oberhalb des Filters schnell aufgerufen werden.

Weitere Details zum Speichern von Filtern.

Subfilter verwenden

Mit dem Subfilter können Sie die Filterung aus dem Hauptfilter weiter verfeinern.

Er enthält anklickbare Links für den schnellen Zugriff auf verwandte Datenpunkte. Datenpunkte werden nach einer gemeinsamen Entität gruppiert – Host, Tag-Name oder -Wert, Datenpunktstatus oder Datenstatus. Wenn eine Entität angeklickt wird, wird sie mit einem grauen Hintergrund hervorgehoben und die Datenpunkte werden sofort gefiltert (im Hauptfilter muss nicht auf *Anwenden* geklickt werden). Durch Anklicken einer weiteren Entität wird diese zu den gefilterten Ergebnissen hinzugefügt. Durch erneutes Anklicken der Entität wird die Filterung entfernt.



Subfilter werden auf Grundlage der gefilterten Daten erstellt, die auf 1000 Datensätze begrenzt sind. Wenn es 20 Hosts mit jeweils 100 Datenpunkten gibt (also insgesamt 2000 Datensätze), wird nur die Hälfte der Hosts im Subfilter angezeigt. Wenn Sie mehr Datensätze im Subfilter anzeigen möchten, müssen Sie den Wert des Parameters *Limit für Such- und Filterergebnisse* erhöhen (unter *Administration* > *Allgemein* > *GUI*).

Im Gegensatz zum Hauptfilter wird der Subfilter bei jeder Aktualisierung der Tabelle aktualisiert, damit die verfügbaren Filteroptionen und deren Anzahl stets aktuell sind.

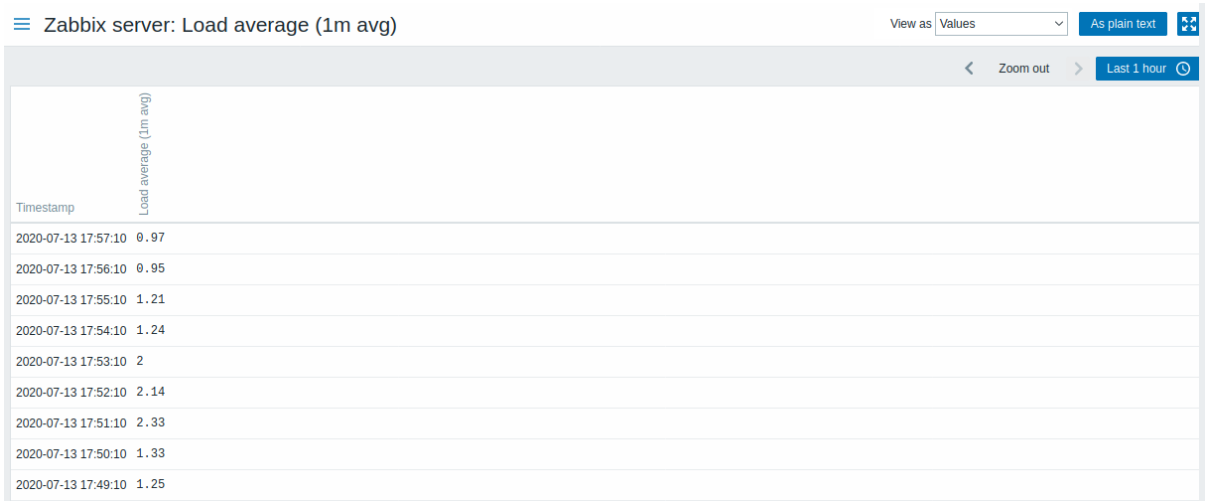
Für jede Entitätsgruppe (Hosts, Tags, Tag-Werte usw.) werden bis zu 10 Zeilen mit Entitäten angezeigt. Wenn es mehr Entitäten gibt, kann diese Liste durch Klicken auf das Drei-Punkte-Symbol ••• am Ende der Liste erweitert werden, um maximal 1000 Einträge anzuzeigen (der Wert von SUBFILTER_VALUES_PER_GROUP in den Frontend-Definitionen). Für *Tag-Werte* kann die Liste erweitert werden, um maximal 200 Tag-Namen mit den entsprechenden Werten anzuzeigen. Beachten Sie, dass die Liste nach vollständiger Erweiterung nicht wieder reduziert werden kann.

Eine Zahl neben jeder anklickbaren Entität gibt die Anzahl der darin gruppierten Datenpunkte an (basierend auf den Ergebnissen des Hauptfilters). Wenn eine Entität angeklickt wird, werden neben den anderen verfügbaren Entitäten Zahlen mit einem Pluszeichen angezeigt. Diese geben an, wie viele Datenpunkte zur aktuellen Auswahl hinzugefügt werden können. Entitäten ohne Datenpunkte werden nicht angezeigt, es sei denn, sie wurden zuvor im Subfilter ausgewählt.

Diagramme und Verlauf

Die Spalte *Graph/History* in der Datenpunktliste bietet die folgenden Links:

- **History** - für alle textbasierten Datenpunkte, führt zu Listen (*Values/500 latest values*), die den Verlauf früherer Datenpunktwerte anzeigen.
- **Graph** - für alle numerischen Datenpunkte, führt zu einem einfachen Diagramm. Beachten Sie, dass beim Anzeigen des Diagramms ein Dropdown-Menü oben rechts auch die Möglichkeit bietet, zu *Values/500 latest values* zu wechseln.



Die in dieser Liste angezeigten Werte sind Rohwerte, das heißt, es wird keine Nachverarbeitung angewendet.

Note:
Die Gesamtzahl der angezeigten Werte wird durch den Wert des Parameters *Limit for search and filter results* festgelegt, der unter *Administration* > *General* > *GUI* gesetzt wird.

Karten

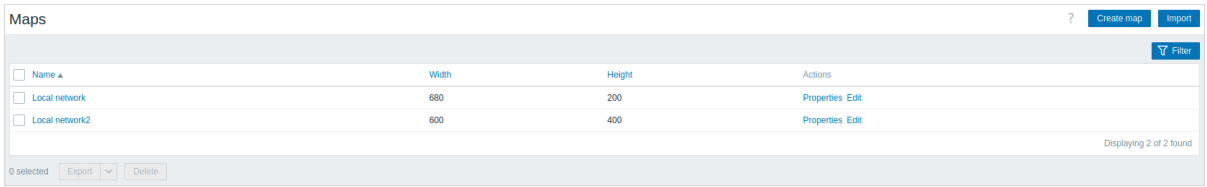
Übersicht

Im Abschnitt *Monitoring* > *Maps* können Sie Netzwerk-Maps konfigurieren, verwalten und anzeigen.

Wenn Sie diesen Abschnitt öffnen, sehen Sie entweder die zuletzt aufgerufene Map oder eine Liste aller Maps, auf die Sie Zugriff haben.

Alle Maps können entweder öffentlich oder privat sein. Öffentliche Maps sind für alle Benutzer verfügbar, während private Maps nur für ihren Besitzer und die Benutzer zugänglich sind, für die die Map freigegeben wurde.

Kartenliste



Angezeigte Daten:

Column	Description
Name	Name der Karte. Klicken Sie auf den Namen, um die Karte anzuzeigen view.
Width	Die Kartenbreite wird angezeigt.
Height	Die Kartenhöhe wird angezeigt.
Actions	Es sind zwei Aktionen verfügbar: **Properties** - allgemeine properties der Karte festlegen **Edit** - auf das Raster zugreifen, um map elements hinzuzufügen

Um eine neue Karte zu configure, klicken Sie auf die Schaltfläche *Create map* in der oberen rechten Ecke. Um eine Karte aus einer YAML-, XML- oder JSON-Datei zu importieren, klicken Sie auf die Schaltfläche *Import* in der oberen rechten Ecke. Der Benutzer, der die Karte importiert, wird als Eigentümer festgelegt.

Zwei Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

- *Export* - die Karten in eine YAML-, XML- oder JSON-Datei exportieren
- *Delete* - die Karten löschen

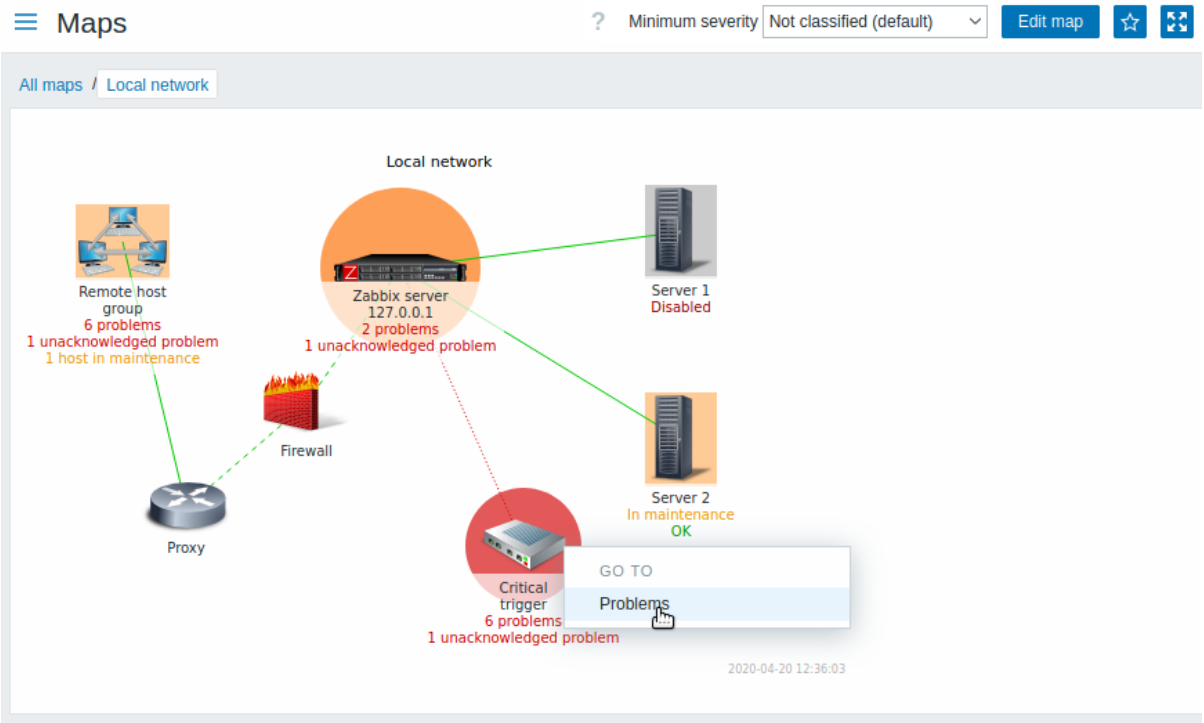
Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Karten und klicken Sie dann auf die gewünschte Schaltfläche.

Filter verwenden

Sie können den Filter verwenden, um nur die Karten anzuzeigen, die Sie interessieren. Für eine bessere Suchleistung werden Daten mit nicht aufgelösten Makros durchsucht.

Anzeigen von Karten

Um eine Karte anzuzeigen, klicken Sie in der Liste aller Karten auf ihren Namen.



Sie können das Dropdown-Menü in der Titelleiste der Karte verwenden, um den niedrigsten Schweregrad der anzuzeigenden Auslöser für Probleme auszuwählen. Der als *default* markierte Schweregrad ist der in der Kartenkonfiguration festgelegte Wert. Wenn die Karte eine Unterkarte enthält, wird beim Navigieren zur Unterkarte der Schweregrad der übergeordneten Karte beibehalten (außer wenn er *Not classified* ist; in diesem Fall wird er nicht an die Unterkarte weitergegeben).

Symbolhervorhebung

Wenn sich ein Map-Element in einem Problemzustand befindet, wird es mit einem runden Kreis hervorgehoben. Die Füllfarbe des Kreises entspricht der Farbe des Schweregrads des Problems. Nur Probleme mit dem ausgewählten Schweregrad oder einem höheren Schweregrad werden zusammen mit dem Element angezeigt. Wenn alle Probleme bestätigt wurden, wird ein dicker grüner Rand um den Kreis angezeigt.

Zusätzlich:

- Ein Host in der Wartung wird mit einem orangefarbenen, ausgefüllten Quadrat hervorgehoben. Beachten Sie, dass die Hervorhebung der Wartung Vorrang vor der Hervorhebung des Problem-Schweregrads hat, wenn das Map-Element ein Host ist.
- Ein deaktivierter (nicht überwachter) Host wird mit einem grauen, ausgefüllten Quadrat hervorgehoben.

Die Hervorhebung wird angezeigt, wenn das Kontrollkästchen *Symbolhervorhebung* in der Map-Konfiguration aktiviert ist.

Markierungen für kürzlich erfolgte Änderungen

Nach innen zeigende rote Dreiecke um ein Element weisen auf eine kürzlich erfolgte Änderung des Auslöserstatus hin – eine Änderung, die innerhalb der letzten 30 Minuten stattgefunden hat.
Diese Dreiecke werden angezeigt, wenn das Kontrollkästchen *Elemente bei Änderung des Auslöserstatus markieren* in der Karteneinstellung aktiviert ist.

Links

Durch Klicken auf ein Kartenelement wird ein Menü mit einigen verfügbaren Links geöffnet. Durch Klicken auf den Hostnamen wird das Host-Menü angezeigt.

Schaltflächen

Die Schaltflächen rechts bieten folgende Optionen:

	Zur Bearbeitung des Karteninhalts wechseln.
	Karte zum Favoriten-Widget in Dashboards hinzufügen.
	Die Karte befindet sich im Favoriten-Widget in Dashboards. Klicken Sie, um die Karte aus dem Favoriten-Widget zu entfernen.

Die Schaltflächen des Ansichtsmodus, die in allen Bereichen gleich sind, werden auf der Seite Monitoring beschrieben.

Lesbare Zusammenfassung in Karten

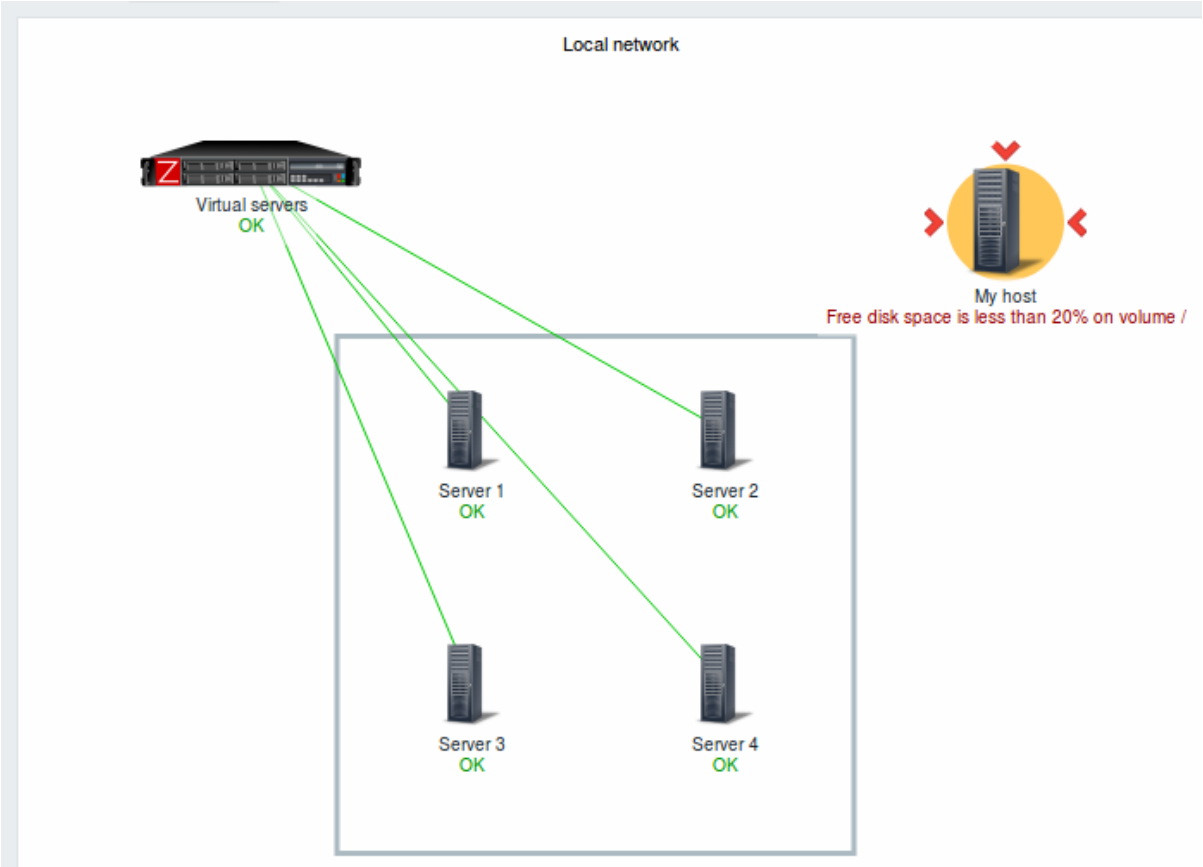
Eine ausgeblendete Eigenschaft „aria-label“ ist verfügbar, sodass Karteninformationen mit einem Screenreader vorgelesen werden können.
Sowohl eine allgemeine Kartenbeschreibung als auch die Beschreibung einzelner Elemente sind im folgenden Format verfügbar:

- für die Kartenbeschreibung: `<Kartenname>, <* von * Elementen mit Problemstatus>, <* Probleme insgesamt>.`
- für die Beschreibung eines Elements mit einem Problem: `<Elementtyp>, Status <Elementstatus>, <Elementname>, <Problembeschreibung>.`
- für die Beschreibung eines Elements mit mehreren Problemen: `<Elementtyp>, Status <Elementstatus>, <Elementname>, <* Probleme>.`
- für die Beschreibung eines Elements ohne Probleme: `<Elementtyp>, Status <Elementstatus>, <Elementname>.`

Beispielsweise ist folgende Beschreibung verfügbar:

```
'Lokales Netzwerk, 1 von 6 Elementen mit Problemstatus, 1 Problem insgesamt. Host, Status Problem, Mein Ho
```

für die folgende Karte:



Referenzieren einer Netzwerkkarte

Netzwerkkarten können sowohl über die GET-Parameter `sysmapid` als auch `mapname` referenziert werden. Zum Beispiel

`http://zabbix/zabbix/zabbix.php?action=map.view&mapname=Local%20network`

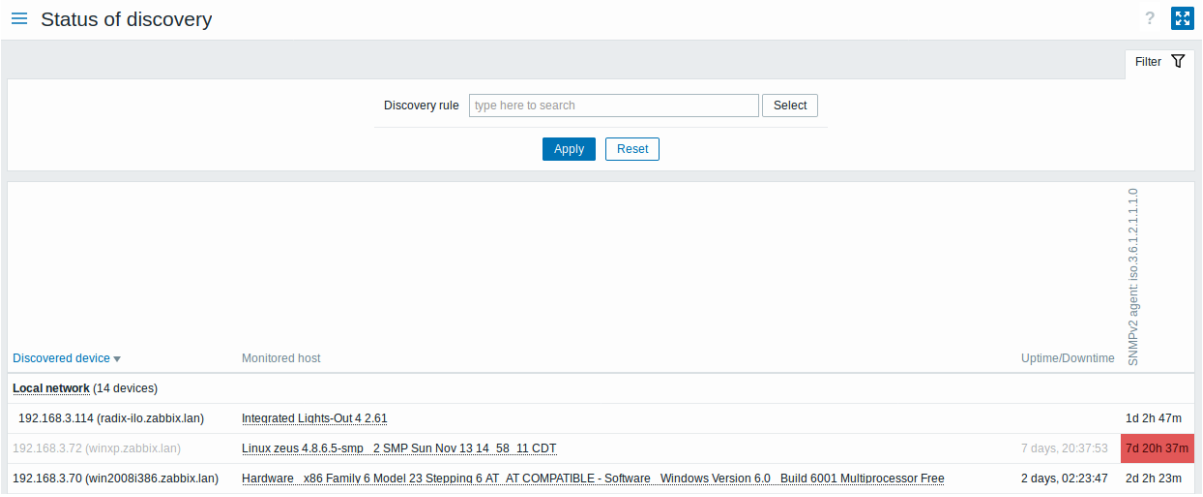
öffnet die Karte mit diesem Namen (Local network).

Wenn sowohl `sysmapid` (Karten-ID) als auch `mapname` (Kartenname) angegeben sind, hat `mapname` eine höhere Priorität.

Erkennung

Übersicht

Im Bereich *Überwachung* > *Erkennung* werden die Ergebnisse der Netzwerkerkennung angezeigt. Erkannte Geräte werden nach der Erkennungsregel sortiert.



Angezeigte Daten:

Spalte	Beschreibung
Erkanntes Gerät	Erkannte Geräte werden nach Erkennungsregel gruppiert aufgelistet. Wenn Sie auf die Erkennungsregel klicken, wird das Regelmenü mit dem Link zum Konfigurationsformular der Erkennungsregel geöffnet. Wenn ein Gerät mehrere IP-Adressen hat (wie durch die *Kriterien für die Eindeutigkeit von Geräten* der Erkennungsregel bestimmt), werden diese zusammen aufgelistet: zuerst die primäre IP-Adresse des Geräts, gefolgt von den anderen IP-Adressen (eingerückt). Die primäre IP-Adresse ist die niedrigste IP-Adresse, für die in der Spalte *Überwachter Host* ein Host aufgeführt ist, oder, falls keine solche Adresse vorhanden ist, die niedrigste IP-Adresse des Geräts. Geräte können nach ihrer primären IP-Adresse in aufsteigender oder absteigender Reihenfolge sortiert werden. Die anderen IP-Adressen eines Geräts werden immer unterhalb seiner primären IP-Adresse in aufsteigender Reihenfolge aufgeführt.
Überwachter Host	Wenn ein Gerät bereits überwacht wird, wird der Hostname in dieser Spalte aufgeführt. Wenn Sie auf den Hostnamen klicken, wird das Hostmenü geöffnet.
Betriebszeit/Ausfallzeit	In dieser Spalte wird die Dauer angezeigt, seit das Gerät nach der vorherigen Erkennung erkannt wurde oder nicht mehr erkannt wird. Bei einem Gerät mit mehreren IP-Adressen wird die Dauer nur für seine primäre IP-Adresse angezeigt.
Erkennungsprüfung	Der Status des einzelnen Dienstes (Erkennungsprüfung) wird für jedes erkannte Gerät angezeigt. Eine rote Zelle zeigt an, dass der Dienst nicht verfügbar ist. Die Betriebs- oder Ausfallzeit des Dienstes wird innerhalb der Zelle angezeigt. Diese Spalte wird nur angezeigt, wenn der Dienst auf mindestens einem erkannten Gerät gefunden wurde.

Note:
Die Anzahl der Erkennungsregeln und die Gesamtzahl der in diesem Bereich angezeigten erkannten Geräte wird durch den Parameter *Limit für Such- und Filterergebnisse* unter *Administration > Allgemein > GUI* begrenzt. Regeln oder Geräte, die das Limit überschreiten, werden nicht angezeigt, ohne dass dies kenntlich gemacht wird.

Schaltflächen

Schaltflächen für den Ansichtsmodus, die für alle Bereiche gemeinsam sind, werden auf der Seite Monitoring beschrieben.

Filter verwenden

Sie können den Filter verwenden, um nur die Discovery-Regeln anzuzeigen, die Sie interessieren. Für eine bessere Suchleistung werden die Daten mit nicht aufgelösten Makros durchsucht.

Wenn im Filter nichts ausgewählt ist, werden alle aktivierten Discovery-Regeln angezeigt. Um eine bestimmte Discovery-Regel zur Anzeige auszuwählen, beginnen Sie, ihren Namen in den Filter einzugeben. Alle passenden aktivierten Discovery-Regeln werden zur Auswahl aufgelistet. Es können mehrere Discovery-Regeln ausgewählt werden.

APM

Übersicht

Das Menü *APM* dient den Funktionen zur Überwachung der Anwendungsleistung (APM) von Zabbix. Es zeigt Telemetriedaten an, die Ihre Anwendungen und Dienste mithilfe von OpenTelemetry, einem offenen Standard zur Erfassung von Telemetriedaten, senden. Diese Daten helfen Ihnen dabei, nachzuvollziehen, wie Anforderungen durch Ihre Anwendungen und Dienste fließen, deren Leistung zu messen und die Ursache für langsame oder fehlgeschlagene Vorgänge zu ermitteln.

Das Menü enthält die folgenden Abschnitte, die den OpenTelemetry-Signaltypen entsprechen:

- Traces
- Metriken
- Logs

Im Gegensatz zu *Monitoring* > *Neueste Daten*, das von Zabbix-Datenpunkten erfasste Werte anzeigt, zeigen die *APM*-Abschnitte Telemetriedaten direkt aus der Datenquelle an. Daher müssen Sie keine Datenpunkte erstellen. So zeigen Sie Daten in den *APM*-Abschnitten an:

1. Geben Sie unter *Administration* > *Datenquelle* eine Telemetriedatenquelle an.
2. Stellen Sie sicher, dass Telemetriedaten von einem Zabbix Proxy mit aktiviertem APM oder von einem OpenTelemetry Collector in die Datenquelle geschrieben werden. Beachten Sie, dass der Zabbix Server keine Telemetriedaten empfangen kann; dies ist nur mit einem Zabbix Proxy möglich.

Um Telemetriedaten in Auslösern oder Diagrammen zu verwenden, nutzen Sie den Telemetrieabfrage-Datenpunkt.

Traces

Übersicht

Der Abschnitt *APM* > *Traces* zeigt die aus Ihren Anwendungen und Diensten erfassten Traces an.

Ein **Trace** stellt den Pfad dar, den eine Anfrage durch Ihre Anwendungen und Dienste nimmt. Wenn ein Benutzer beispielsweise in einem Onlineshop auf die Schaltfläche *Bestellung aufgeben* klickt, kann die Anfrage vom Frontend-Dienst über die Checkout- und Zahlungsdienste bis zur Datenbank weitergeleitet werden.

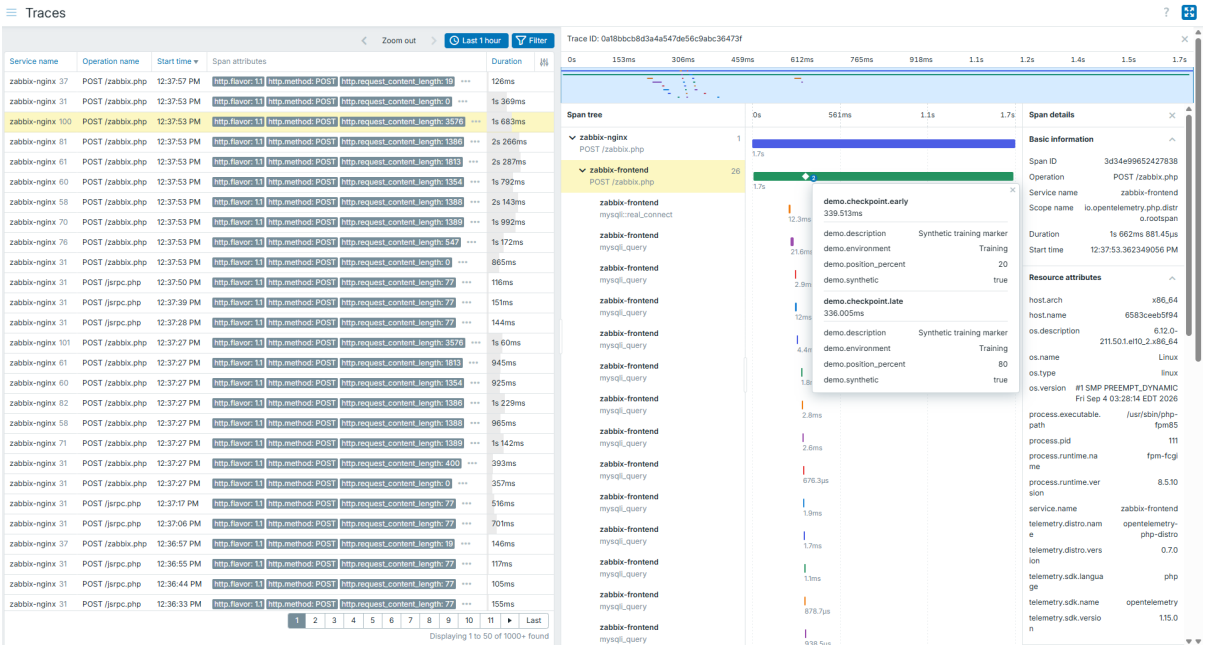
Ein Trace besteht aus **Spans**. Jeder Span stellt eine einzelne Operation dar, z. B. eine HTTP-Anfrage, die der Frontend-Dienst an den Checkout-Dienst sendet, eine Datenbankabfrage zum Speichern der Bestellung oder einen Funktionsaufruf zur Berechnung der Versandkosten.

Spans bilden eine Hierarchie. Der **Root-Span** stellt die erste Operation der Anfrage dar. Jeder andere Span ist ein **untergeordneter Span** des Spans, während dessen Operation er gestartet wurde. Beispielsweise ist der Span, der den Empfang der Anfrage *Bestellung aufgeben* durch den Frontend-Dienst darstellt, der Root-Span. Der Span, der die Verarbeitung der Bestellung durch den Checkout-Dienst darstellt, ist sein untergeordneter Span.

Jeder Span wird von einem **Dienst** erzeugt (z. B. `zabbix-frontend`; benannt durch das OpenTelemetry-Ressourcenattribut `service.name`) sowie von einem Instrumentierungsbereich, also dem Teil des Codes, der ihn erstellt hat (z. B. eine Bibliothek, die HTTP-Anfragen verfolgt).

Um Kontext zu jeder Operation bereitzustellen, verfügen Spans außerdem über **Attribute** in Form von Schlüssel-Wert-Paaren: Span-Attribute beschreiben die Operation selbst (z. B. `http.method: POST`), während Ressourcenattribute den Dienst und die Umgebung beschreiben, in der er ausgeführt wird (z. B. `host.name: eabf8c2a5ddc`).

Weitere Informationen zu Traces finden Sie in der OpenTelemetry-Dokumentation.

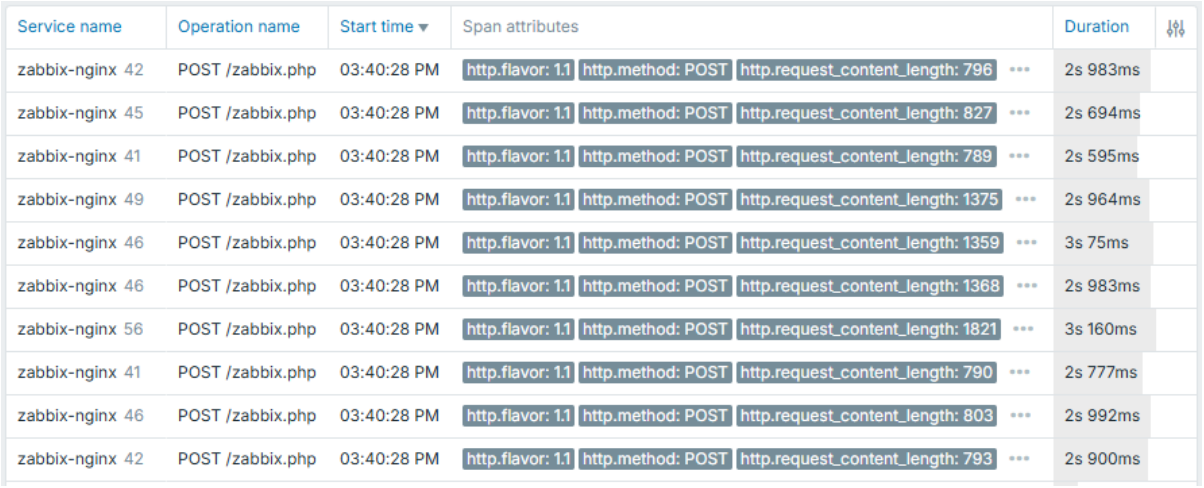


Der Abschnitt *APM* > *Traces* enthält die folgenden Elemente:

- Schaltfläche Kioskmodus – zeigt nur den Seiteninhalt ohne Menü und Kopfzeile an.
- Zeitraumauswahl – wählt den Zeitraum aus, für den Traces angezeigt werden sollen.
- Filter – zeigt nur die Traces an, die Sie interessieren.
- Trace-Liste – zeigt die Traces an, die durch ihre Root-Spans beschrieben werden.
- Trace-Panel – zeigt alle Spans des ausgewählten Traces auf einer Zeitleiste an.

Trace-Liste

Jede Zeile in der Trace-Liste stellt einen Trace dar; die Spalten zeigen die Daten seines Root-Spans.



Service name	Operation name	Start time ▾	Span attributes	Duration
zabbix-nginx 42	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 796 •••	2s 983ms
zabbix-nginx 45	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 827 •••	2s 694ms
zabbix-nginx 41	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 789 •••	2s 595ms
zabbix-nginx 49	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 1375 •••	2s 964ms
zabbix-nginx 46	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 1359 •••	3s 75ms
zabbix-nginx 46	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 1368 •••	2s 983ms
zabbix-nginx 56	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 1821 •••	3s 160ms
zabbix-nginx 41	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 790 •••	2s 777ms
zabbix-nginx 46	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 803 •••	2s 992ms
zabbix-nginx 42	POST /zabbix.php	03:40:28 PM	http.flavor: 1.1 http.method: POST http.request_content_length: 793 •••	2s 900ms

Traces werden in einer anpassbaren Tabelle angezeigt. Mit der Schaltfläche *Tabelle anpassen* in der Tabellenkopfzeile können Sie Spalten ein- und ausblenden, ihre Reihenfolge ändern und das Layout auf die Standardeinstellungen zurücksetzen. Sie können außerdem die Größe jeder Spalte ändern, indem Sie ihren rechten Rand ziehen; durch Doppelklicken auf den Rand wird die ursprüngliche Größe wiederhergestellt. Alle Anpassungen werden sofort angewendet und für den aktuellen Benutzer gespeichert.

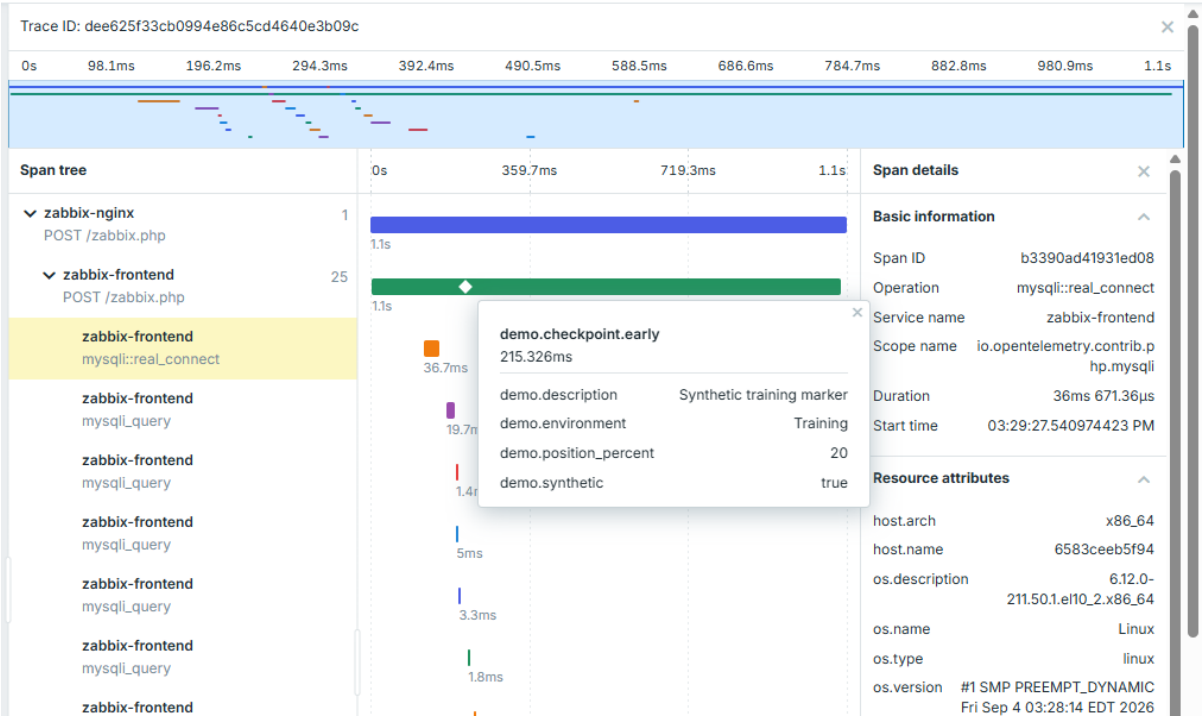
Die Trace-Liste kann die folgenden Spalten enthalten:

Note:
Jede Spalte entspricht einer Spalte in der Tabelle `otel_traces` der Telemetriedatenquelle; eine Liste der Spalten finden Sie unter Unterstützte Telemetriedatenspalten. Die meisten Spalten haben denselben Namen wie in der Datenquelle (z. B. entspricht *Trace-ID* der Spalte `TraceId`). Wenn sich die Namen unterscheiden, wird die Spalte der Datenquelle in der folgenden Tabelle angegeben.

Spalte	Beschreibung
Trace-ID	ID des Traces, die von allen zu diesem Trace gehörenden Spans gemeinsam verwendet wird.
Span-ID	ID des Root-Spans.
Trace-Status	Zusätzliche Daten, sofern vorhanden, die Tracing-Tools an den Trace anhängen (siehe W3C Trace Context).
Servicename	Service, der den Root-Span erzeugt hat, gefolgt von der Anzahl der Spans im Trace. Wenn ein Span den Statuscode *Error* aufweist (siehe unten), zeigt ein rotes Symbol deren Anzahl an.
Operationsname	Operation, die der Root-Span darstellt (z. B. `POST /zabbix.php`). Spalte der Datenquelle: `SpanName`.
Span-Art	Rolle des Root-Spans: **Unspecified** – nicht angegeben. **Internal** – innerhalb einer Anwendung ausgeführte Operation. **Server** – Verarbeitung einer eingehenden Anfrage. **Client** – ausgehende Anfrage. **Producer** – Erstellung einer asynchronen Operation (z. B. Senden einer Nachricht). **Consumer** – Verarbeitung einer asynchronen Operation.
Scope-Name	Instrumentierungs-Scope, der den Root-Span erzeugt hat.
Scope-Version	Version des Instrumentierungs-Scopes.
Statuscode	Status der Operation, die der Root-Span darstellt: **Unset** – dem Root-Span wurde kein expliziter Status zugewiesen. **Ok** – die Operation wurde erfolgreich abgeschlossen. **Error** – die Operation wurde mit einem Fehler abgeschlossen.
Statusmeldung	Zusätzliche Informationen, sofern vorhanden, zum Status.
Startzeit	Datum und Uhrzeit, zu denen der Root-Span gestartet wurde. Spalte der Datenquelle: `Timestamp`.
Span-Attribute	Span-Attribute des Root-Spans. Klicken Sie auf die Schaltfläche mit den drei Punkten ••• , um alle Attribute anzuzeigen.
Ressourcenattribute	Ressourcenattribute des Root-Spans. Klicken Sie auf die Schaltfläche mit den drei Punkten ••• , um alle Attribute anzuzeigen.
Dauer	Dauer des Traces; der graue Balken zeigt sie relativ zu den anderen Traces an.

Trace-Bereich

Klicken Sie in der Trace-Liste auf einen Trace, um den Trace-Bereich zu öffnen. Dieser zeigt alle Spans des Traces auf einer Zeitleiste: welche Vorgänge ausgeführt wurden, in welcher Reihenfolge sie ausgeführt wurden und wie lange jeder Vorgang gedauert hat.

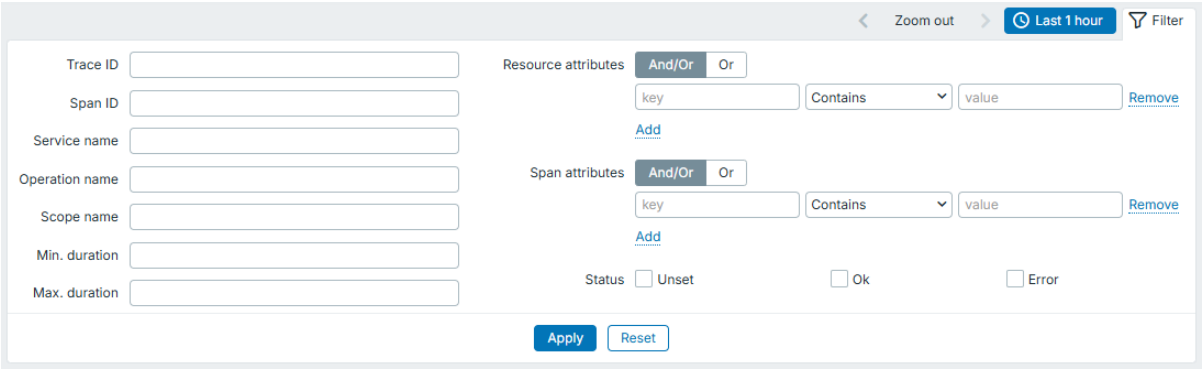


Der Trace-Bereich kann folgende Elemente enthalten:

Element	Beschreibung
Trace ID	ID des ausgewählten Traces.
Übersicht der Trace-Zeitleiste	Der gesamte Trace mit Spans in bis zu 8 Zeilen (mehrere Spans können sich eine Zeile teilen). Ziehen Sie die Ränder des ausgewählten Bereichs oder den Bereich selbst, um die Span-Zeitleiste auf diesen Teil des Traces zu vergrößern.
Span-Baum	Spans in ihrer Hierarchie; jeder Span zeigt seinen Service (fett), den Namen des Vorgangs und die Anzahl der direkten untergeordneten Spans.
Span-Zeitleiste	Spans als Balken: Position und Länge zeigen, wann jeder Span gestartet wurde und wie lange er gedauert hat. Marker zeigen Ereignisse (z. B. Ausnahmen) an. Bewegen Sie den Mauszeiger über einen Marker oder klicken Sie darauf, um Details anzuzeigen.
Span-Details	Klicken Sie im Span-Baum auf einen Span, um dessen Details und Attribute anzuzeigen.

Filter verwenden

Sie können den Filter verwenden, um nur die für Sie relevanten Traces anzuzeigen.



Um nach Attributen zu filtern, geben Sie den Attributschlüssel ein, wählen Sie den Operator aus und geben Sie den Attributwert ein.

Die folgenden Operatoren werden unterstützt:

- **Enthält** - *(Standard)* der Attributwert enthält die eingegebene Zeichenfolge.
- **Ist gleich** - der Attributwert entspricht der eingegebenen Zeichenfolge.

- **Enthält nicht** - der Attributwert enthält die eingegebene Zeichenfolge nicht.
- **Ist nicht gleich** - der Attributwert entspricht nicht der eingegebenen Zeichenfolge.
- **Existiert** - das Attribut mit dem eingegebenen Schlüssel existiert.
- **Existiert nicht** - das Attribut mit dem eingegebenen Schlüssel existiert nicht.

Wenn Sie mehrere Bedingungen hinzufügen, wählen Sie aus, wie diese ausgewertet werden sollen:

- **Und/Oder** - *(Standard)* alle Bedingungen müssen erfüllt sein; Bedingungen mit demselben Attributschlüssel werden mit *Oder* kombiniert.
- **Oder** - mindestens eine Bedingung muss erfüllt sein.

Metriken

Übersicht

Der Abschnitt *APM > Metriken* zeigt Metrikdatenpunkte an, die von Ihren Anwendungen und Diensten erfasst wurden.

Eine **Metrik** ist eine Messung (z. B. Anfragedauer, Anzahl verarbeiteter Bestellungen oder Speichernutzung). Anwendungen senden ihre Werte als **Datenpunkte**, die den zu einem bestimmten Zeitpunkt gemessenen Wert (oder aggregierte Werte) enthalten. Eine Metrik ähnelt einem Zabbix-Datenpunkt, und ein Datenpunkt ähnelt dem Wert dieses Datenpunkts.

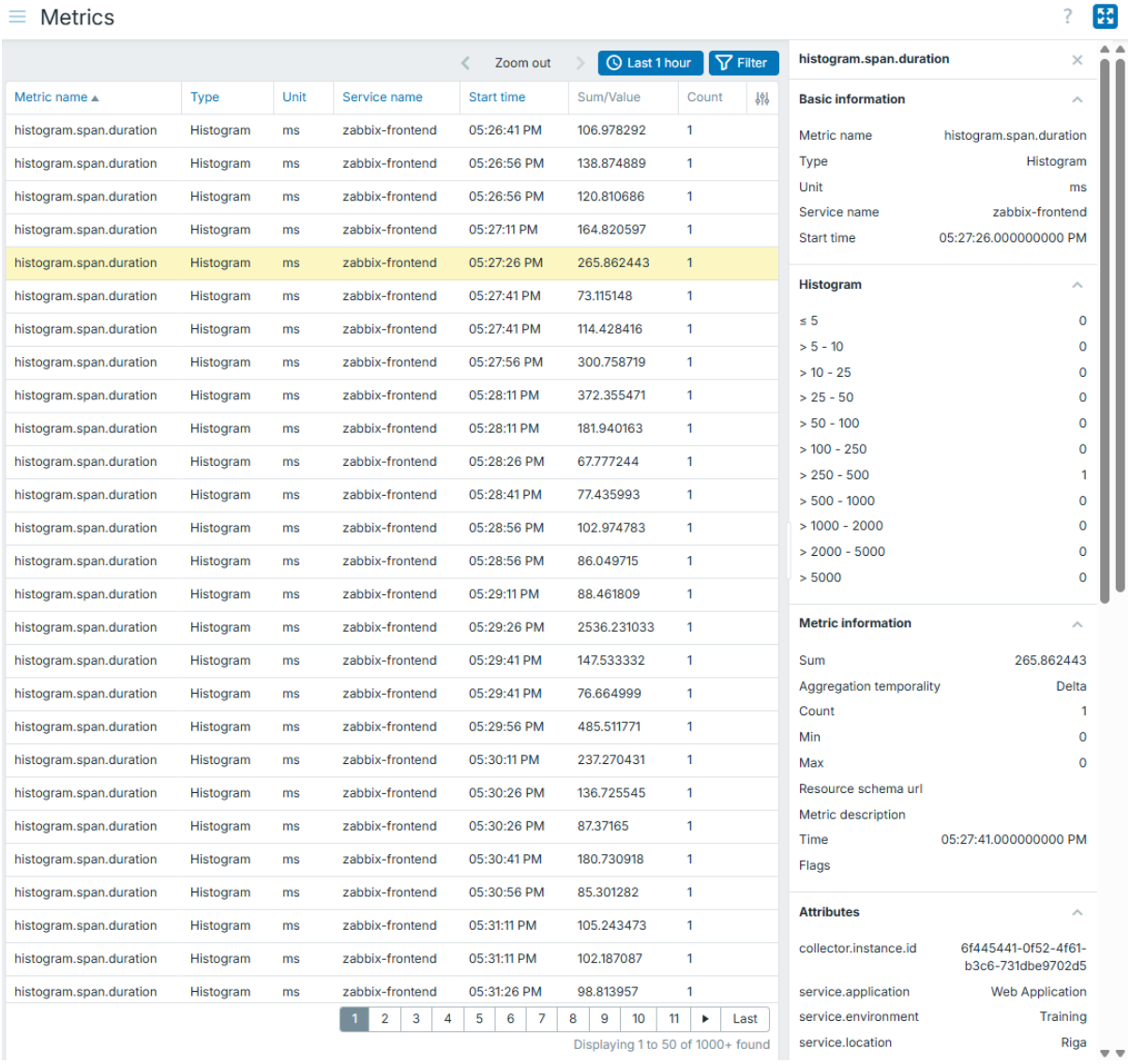
Ein Datenpunkt kann einen der folgenden Typen haben:

- **Gauge** – ein zu einem bestimmten Zeitpunkt gemessener Wert (z. B. aktuelle Speichernutzung).
- **Sum** – eine Summe von Messungen über einen bestimmten Zeitraum (z. B. Gesamtzahl der Anfragen).
- **Histogramm** – die Anzahl der Messungen über einen bestimmten Zeitraum, die in jeden von mehreren festgelegten Wertebereichen fallen, den sogenannten Buckets (z. B. wie viele Anfragen 0–100 ms, 100–250 ms oder 250–500 ms gedauert haben).
- **Exponentielles Histogramm** – ähnlich wie ein Histogramm, wobei jeder Bucket um denselben Faktor größer ist als der vorherige (z. B. 1–2 ms, 2–4 ms, 4–8 ms).

Jeder Datenpunkt wird von einem **Dienst** (z. B. `zabbix-frontend`; benannt durch das OpenTelemetry-Ressourcenattribut `service.name`) und von einem Instrumentierungsbereich erzeugt, also dem Teil des Codes, der ihn erstellt hat (z. B. eine Bibliothek, die HTTP-Anfragedauern misst).

Um Kontext zu jeder Messung bereitzustellen, verfügen Datenpunkte außerdem über **Attribute** aus Schlüssel-Wert-Paaren: Datenpunktattribute beschreiben, was gemessen wurde, und unterscheiden Datenpunkte derselben Metrik (z. B. `http.request.method: POST`). Ressourcenattribute beschreiben den Dienst und die Umgebung, in der er ausgeführt wird (z. B. `host.name: eabf8c2a5ddc`).

Weitere Informationen zu Metriken finden Sie in der OpenTelemetry-Dokumentation.

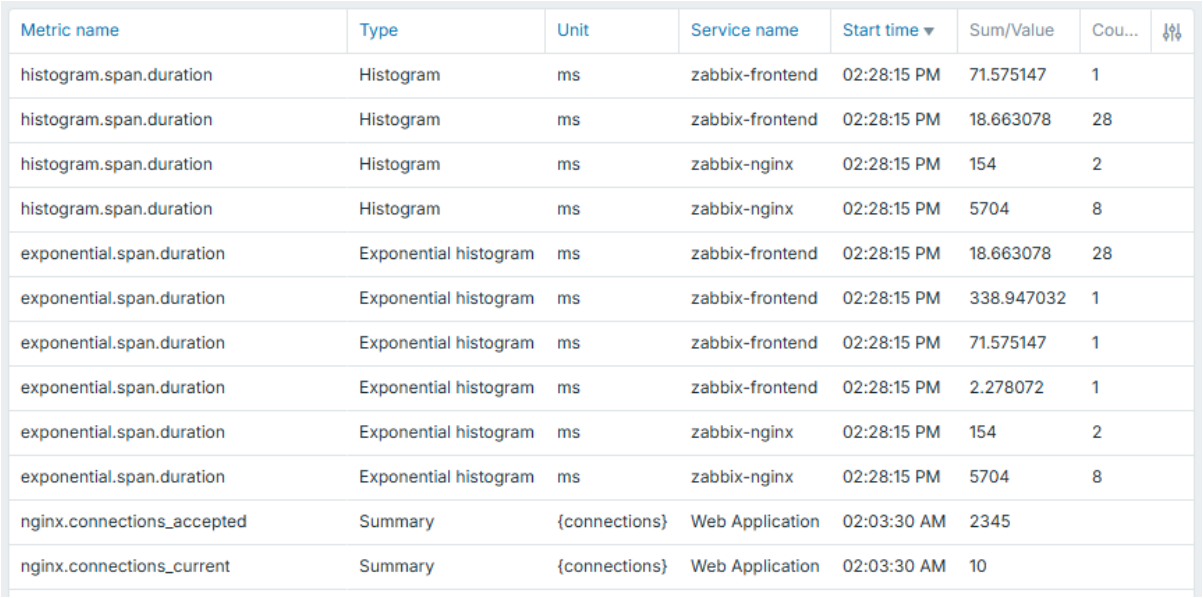


Der Abschnitt *APM* > *Metriken* enthält die folgenden Elemente:

- Schaltfläche Kioskmodus – zeigt nur den Seiteninhalt ohne Menü und Kopfzeile an.
- Zeitraumauswahl – wählt den Zeitraum aus, für den Datenpunkte angezeigt werden sollen.
- Filter – zeigt nur die Datenpunkte an, die Sie interessieren.
- Metrikliste – zeigt die Datenpunkte an.
- Metrikdetails – zeigt die Details des ausgewählten Datenpunkts an.

Metrikliste

Jede Zeile in der Metrikliste stellt einen Datenpunkt dar.



Metric name	Type	Unit	Service name	Start time ▾	Sum/Value	Cou...
histogram.span.duration	Histogram	ms	zabbix-frontend	02:28:15 PM	71.575147	1
histogram.span.duration	Histogram	ms	zabbix-frontend	02:28:15 PM	18.663078	28
histogram.span.duration	Histogram	ms	zabbix-nginx	02:28:15 PM	154	2
histogram.span.duration	Histogram	ms	zabbix-nginx	02:28:15 PM	5704	8
exponential.span.duration	Exponential histogram	ms	zabbix-frontend	02:28:15 PM	18.663078	28
exponential.span.duration	Exponential histogram	ms	zabbix-frontend	02:28:15 PM	338.947032	1
exponential.span.duration	Exponential histogram	ms	zabbix-frontend	02:28:15 PM	71.575147	1
exponential.span.duration	Exponential histogram	ms	zabbix-frontend	02:28:15 PM	2.278072	1
exponential.span.duration	Exponential histogram	ms	zabbix-nginx	02:28:15 PM	154	2
exponential.span.duration	Exponential histogram	ms	zabbix-nginx	02:28:15 PM	5704	8
nginx.connections_accepted	Summary	{connections}	Web Application	02:03:30 AM	2345	
nginx.connections_current	Summary	{connections}	Web Application	02:03:30 AM	10	

Datenpunkte werden in einer anpassbaren Tabelle angezeigt. Mit der Schaltfläche *Tabelle anpassen* in der Tabellenkopfzeile können Sie Spalten ein- und ausblenden, ihre Reihenfolge ändern und das Layout auf die Standardeinstellungen zurücksetzen. Sie können außerdem die Größe jeder Spalte ändern, indem Sie ihren rechten Rand ziehen. Doppelklicken Sie auf den Rand, um die Größe zurückzusetzen. Alle Anpassungen werden sofort angewendet und für den aktuellen Benutzer gespeichert.

Die Metrikliste kann folgende Spalten enthalten:

Note:
Jede Spalte außer *Typ* entspricht einer Spalte in den Metriktabellen (`otel_metrics_gauge`, `otel_metrics_sum`, `otel_metrics_histogram` und `otel_metrics_exponential_histogram`) der Telemetriedatenquelle. Eine Liste der Spalten finden Sie unter Unterstützte Telemetriedatenspalten. Die meisten Spalten haben denselben Namen wie in der Datenquelle (z. B. entspricht *Metrikname* der Spalte `MetricName`). Wenn sich die Namen unterscheiden, wird die Spalte der Datenquelle in der folgenden Tabelle angegeben.

Spalte	Beschreibung
Metrikname	Metrik, zu der der Datenpunkt gehört (z. B. `http.server.duration`).
Typ	Typ des Datenpunkts: *Gauge*, *Summe*, *Histogramm* oder *Exponentielles Histogramm*. Wird anhand der Metriktabelle bestimmt, in der sich der Datenpunkt befindet.
Einheit	Einheit der Metrikwerte, sofern vorhanden (z. B. `s` oder `By`; siehe UCUM). Spalte der Datenquelle: `MetricUnit`.
Servicename	Service, der den Datenpunkt erzeugt hat.
Bereichsname	Instrumentierungsbereich, der den Datenpunkt erzeugt hat.
Startzeit	Datum und Uhrzeit, zu denen der Messzeitraum des Datenpunkts begonnen hat. Spalte der Datenquelle: `StartTimeUnix`.
Summe/Wert	Summe aller Messungen (*Histogramm*, *Exponentielles Histogramm*) oder der Wert (*Gauge*, *Summe*) des Datenpunkts.
Anzahl	Anzahl der Messungen in einem Datenpunkt vom Typ *Histogramm* oder *Exponentielles Histogramm*.
Zeit	Datum und Uhrzeit des Datenpunkts; bei Datenpunkten vom Typ *Summe*, *Histogramm* und *Exponentielles Histogramm* das Ende des Messzeitraums. Spalte der Datenquelle: `TimeUnix`.
Flags	Flags des Datenpunkts: **0** - Es sind keine Flags gesetzt. **1** - Für den Datenpunkt wurde kein Wert aufgezeichnet.
Aggregationszeitlichkeit	Gibt an, wie die Werte eines Datenpunkts vom Typ *Summe*, *Histogramm* oder *Exponentielles Histogramm* im Zeitverlauf aggregiert werden: **Nicht angegeben** - nicht angegeben. **Delta** - Aggregation über den Zeitraum seit dem vorherigen Datenpunkt. **Kumulativ** - Aggregation über den Zeitraum seit der Startzeit.
Metrikbeschreibung	Beschreibung der Metrik, sofern vorhanden.
Attribute	Attribute des Datenpunkts. Klicken Sie auf die Schaltfläche mit den drei Punkten ••• , um alle Attribute anzuzeigen.

Spalte	Beschreibung
Ressourcenattribute	Ressourcenattribute des Datenpunkts. Klicken Sie auf die Schaltfläche mit den drei Punkten ••• , um alle Attribute anzuzeigen.
Bereichsattribute	Attribute des Instrumentierungsbereichs. Klicken Sie auf die Schaltfläche mit den drei Punkten ••• , um alle Attribute anzuzeigen.

Metrikdetails

Klicken Sie in der Metrikliste auf einen Datenpunkt, um den Bereich mit den Metrikdetails zu öffnen. Dieser zeigt alle Informationen zum Datenpunkt an.

exponential.span.calls

Basic information

Metric name	exponential.span.calls
Type	Summary
Unit	
Service name	zabbix-frontend
Start time	02:22:15.000000000 PM

Metric information

Value	1
Aggregation temporality	Delta
Is monotonic	true
Resource schema url	
Metric description	
Time	02:22:30.000000000 PM
Flags	

Attributes

Resource attributes

Scope attributes

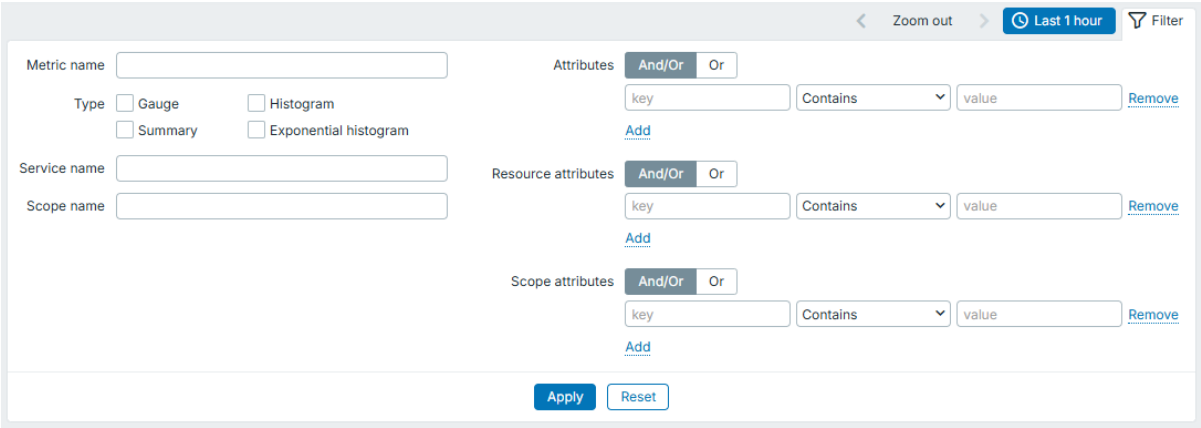
Scope

Der Bereich mit den Metrikdetails kann die folgenden Abschnitte enthalten:

Abschnitt	Beschreibung
Grundlegende Informationen	Metrikname, Typ, Einheit, Servicename und Startzeit.
Histogramm	Buckets (z. B. `> 5 – 10`), jeweils mit der Anzahl der darin enthaltenen Messungen. Wird nur für Datenpunkte vom Typ *Histogramm* angezeigt.
Exponentielles Histogramm	Buckets (z. B. `>= 249.7885 – 249.7887`), jeweils mit der Anzahl der darin enthaltenen Messungen. Wird nur für Datenpunkte vom Typ *Exponentielles Histogramm* angezeigt.
Metrikinformationen	Werte des Datenpunkts, die von seinem Typ abhängen, sowie die Metrikbeschreibung, die Zeit, Flags und die URL des Ressourcenschemas.
Attribute	Attribute des Datenpunkts.
Ressourcenattribute	Ressourcenattribute.
Bereichsattribute	Attribute des Instrumentierungsbereichs, sofern vorhanden.
Bereich	Name, Version und Schema-URL des Instrumentierungsbereichs.

Filter verwenden

Sie können den Filter verwenden, um nur die Datenpunkte anzuzeigen, die Sie interessieren.



Um nach Attributen zu filtern, geben Sie den Attributschlüssel ein, wählen Sie den Operator aus und geben Sie den Attributwert ein.

Die folgenden Operatoren werden unterstützt:

- **Enthält** - *(Standard)* der Attributwert enthält die eingegebene Zeichenfolge.
- **Ist gleich** - der Attributwert entspricht der eingegebenen Zeichenfolge.
- **Enthält nicht** - der Attributwert enthält die eingegebene Zeichenfolge nicht.
- **Ist nicht gleich** - der Attributwert entspricht nicht der eingegebenen Zeichenfolge.
- **Existiert** - das Attribut mit dem eingegebenen Schlüssel existiert.
- **Existiert nicht** - das Attribut mit dem eingegebenen Schlüssel existiert nicht.

Wenn Sie mehrere Bedingungen hinzufügen, wählen Sie aus, wie diese ausgewertet werden sollen:

- **Und/Oder** - *(Standard)* alle Bedingungen müssen erfüllt sein; Bedingungen mit demselben Attributschlüssel werden mit *Oder* kombiniert.
- **Oder** - mindestens eine Bedingung muss erfüllt sein.

Protokolle

Übersicht

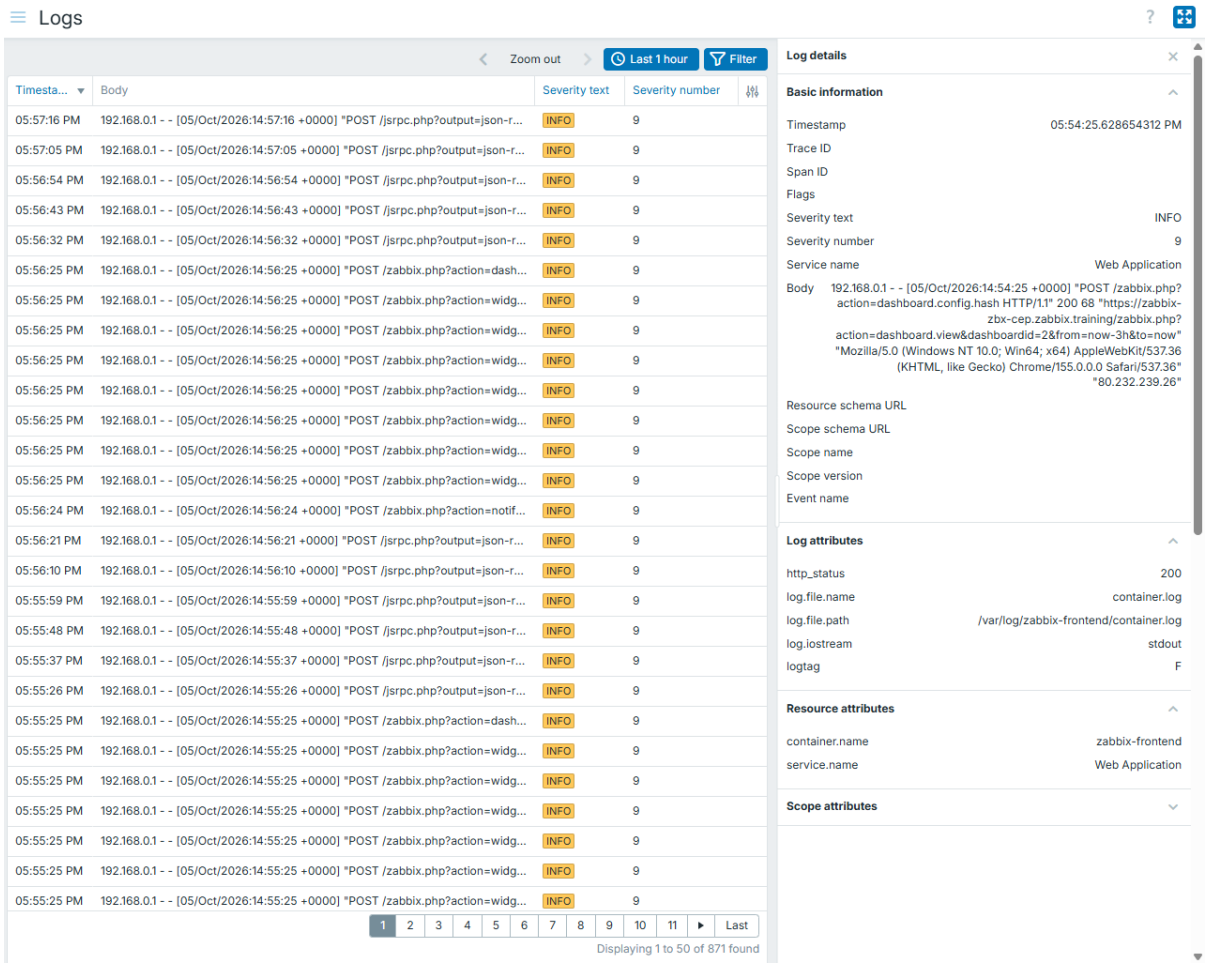
Der Abschnitt *APM* > *Logs* zeigt die aus Ihren Anwendungen und Diensten erfassten Protokolleinträge an.

Ein **Protokolleintrag** ist ein Texteintrag zu einem Ereignis (z. B. einer fehlgeschlagenen Zahlung oder einer Zeitüberschreitung bei einer Datenbank). Jeder Protokolleintrag verfügt über einen Schweregrad und eine Nachricht (Text). Wenn ein Protokolleintrag während eines Vorgangs erzeugt wurde, der Teil eines Trace ist, enthält er außerdem die IDs dieses Traces und Spans.

Jeder Protokolleintrag wird von einem **Dienst** erzeugt (z. B. `zabbix-frontend`; benannt durch das OpenTelemetry-Ressourcenattribut `service.name`) sowie von einem Instrumentierungsbereich, also dem Teil des Codes, der den Eintrag erstellt hat (z. B. einer Protokollierungsbibliothek).

Um Kontext zu jedem Ereignis bereitzustellen, enthalten Protokolleinträge außerdem **Attribute** in Form von Schlüssel-Wert-Paaren: Protokollattribute beschreiben das Ereignis selbst (z. B. `log.file.path: /var/log/zabbix/zabbix_frontend.log`), während Ressourcenattribute den Dienst und die Umgebung beschreiben, in der er ausgeführt wird (z. B. `host.name: eabf8c2a5ddc`).

Weitere Informationen zu Protokollen finden Sie in der OpenTelemetry-Dokumentation.

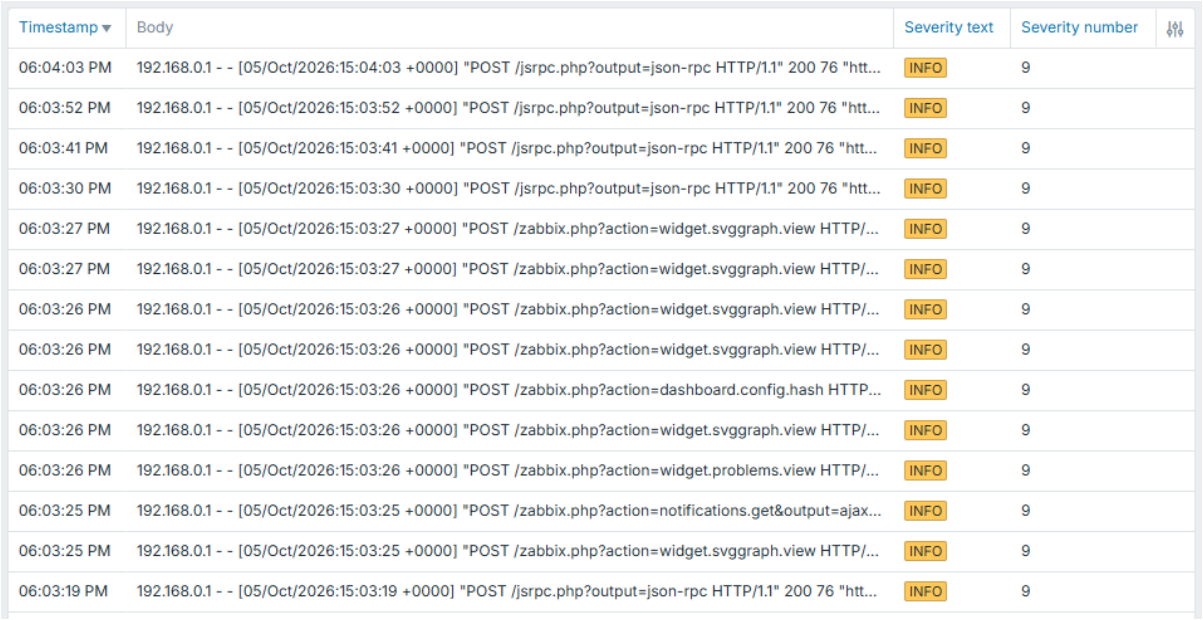


Der Abschnitt *APM* > *Logs* enthält die folgenden Elemente:

- Schaltfläche Kioskmodus – zeigt nur den Seiteninhalt ohne Menü und Kopfzeile an.
- Zeitraumauswahl – wählt den Zeitraum aus, für den Protokolleinträge angezeigt werden sollen.
- Filter – zeigt nur die für Sie relevanten Protokolleinträge an.
- Protokollliste – zeigt die Protokolleinträge an.
- Protokolldetails – zeigt die Details des ausgewählten Protokolleintrags an.

Protokollliste

Jede Zeile in der Protokollliste stellt einen Protokolldatensatz dar.



Timestamp ▾	Body	Severity text	Severity number
06:04:03 PM	192.168.0.1 - - [05/Oct/2026:15:04:03 +0000] "POST /jsrpc.php?output=json-rpc HTTP/1.1" 200 76 "htt...	INFO	9
06:03:52 PM	192.168.0.1 - - [05/Oct/2026:15:03:52 +0000] "POST /jsrpc.php?output=json-rpc HTTP/1.1" 200 76 "htt...	INFO	9
06:03:41 PM	192.168.0.1 - - [05/Oct/2026:15:03:41 +0000] "POST /jsrpc.php?output=json-rpc HTTP/1.1" 200 76 "htt...	INFO	9
06:03:30 PM	192.168.0.1 - - [05/Oct/2026:15:03:30 +0000] "POST /jsrpc.php?output=json-rpc HTTP/1.1" 200 76 "htt...	INFO	9
06:03:27 PM	192.168.0.1 - - [05/Oct/2026:15:03:27 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:27 PM	192.168.0.1 - - [05/Oct/2026:15:03:27 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:26 PM	192.168.0.1 - - [05/Oct/2026:15:03:26 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:26 PM	192.168.0.1 - - [05/Oct/2026:15:03:26 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:26 PM	192.168.0.1 - - [05/Oct/2026:15:03:26 +0000] "POST /zabbix.php?action=dashboard.config.hash HTTP...	INFO	9
06:03:26 PM	192.168.0.1 - - [05/Oct/2026:15:03:26 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:26 PM	192.168.0.1 - - [05/Oct/2026:15:03:26 +0000] "POST /zabbix.php?action=widget.problems.view HTTP/...	INFO	9
06:03:25 PM	192.168.0.1 - - [05/Oct/2026:15:03:25 +0000] "POST /zabbix.php?action=notifications.get&output=ajax...	INFO	9
06:03:25 PM	192.168.0.1 - - [05/Oct/2026:15:03:25 +0000] "POST /zabbix.php?action=widget.svggraph.view HTTP/...	INFO	9
06:03:19 PM	192.168.0.1 - - [05/Oct/2026:15:03:19 +0000] "POST /jsrpc.php?output=json-rpc HTTP/1.1" 200 76 "htt...	INFO	9

Protokolldatensätze werden in einer anpassbaren Tabelle angezeigt. Mit der Schaltfläche *Tabelle anpassen* in der Tabellenkopfzeile können Sie Spalten ein- und ausblenden, ihre Reihenfolge ändern und das Layout auf die Standardeinstellungen zurücksetzen. Sie können außerdem die Größe jeder Spalte ändern, indem Sie ihren rechten Rand ziehen. Doppelklicken Sie auf den Rand, um die Größe zurückzusetzen. Alle Anpassungen werden sofort angewendet und für den aktuellen Benutzer gespeichert.

Die Protokollliste kann folgende Spalten enthalten:

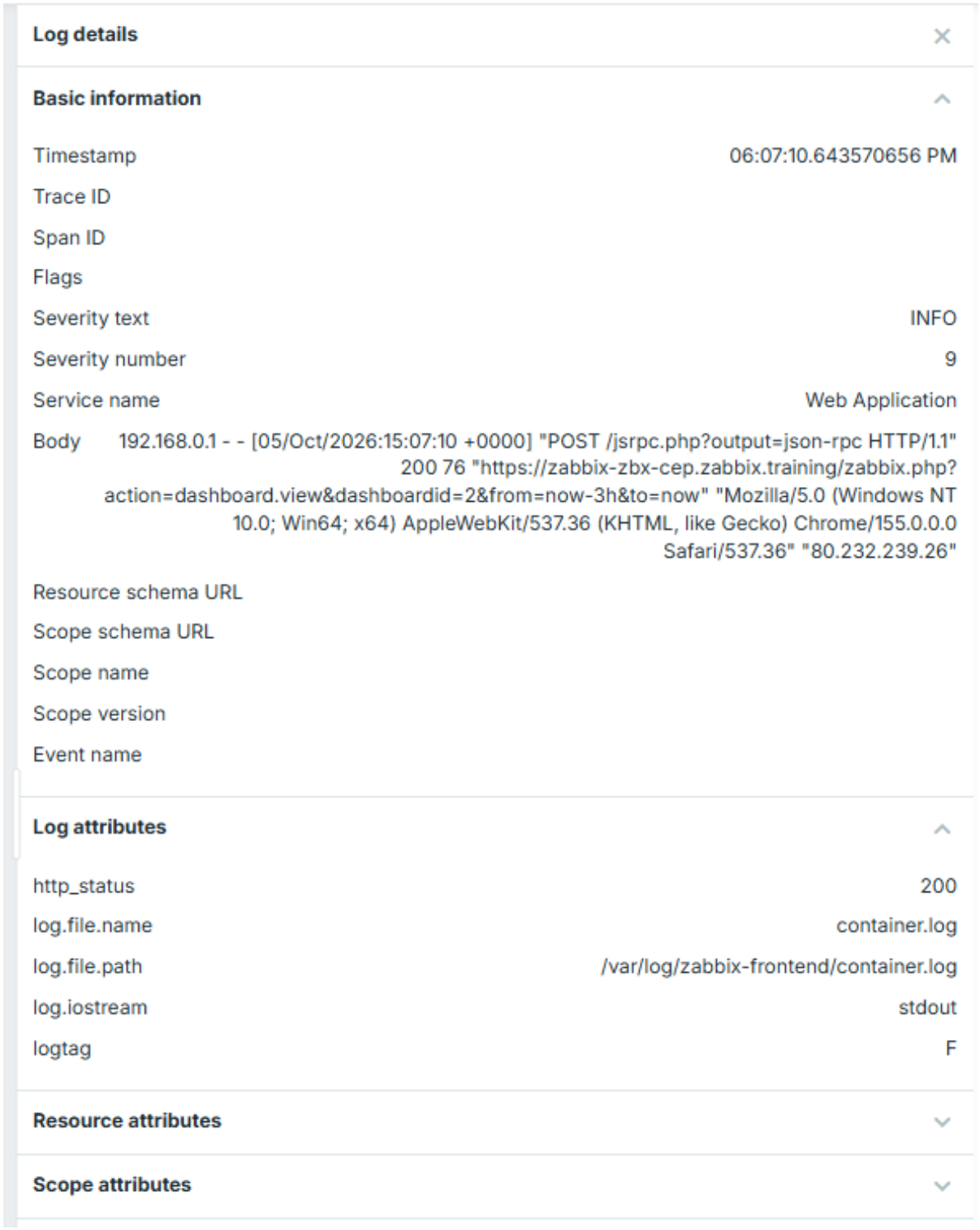
Note:
Jede Spalte entspricht einer Spalte in der Tabelle `otel_logs` der Telemetriedatenquelle. Eine Liste der Spalten finden Sie unter Unterstützte Telemetriedatenspalten. Die meisten Spalten haben denselben Namen wie in der Datenquelle (z. B. entspricht *Schweregradtext* der Spalte `SeverityText`). Wenn sich die Namen unterscheiden, wird die Spalte der Datenquelle in der folgenden Tabelle angegeben.

Spalte	Beschreibung
Zeitstempel	Datum und Uhrzeit, zu denen das Ereignis aufgetreten ist.
Textkörper	Nachricht des Protokolldatensatzes.
Schweregradtext	Von der Anwendung bereitgestellte Schweregradstufe (z. B. `INFO`, `WARN` oder `ERROR`). Die Farbe der Bezeichnung hängt vom Schweregrad ab.
Schweregradnummer	Schweregradstufe als Zahl: **0** - UNSPECIFIED. **1-4** - TRACE. **5-8** - DEBUG. **9-12** - INFO. **13-16** - WARN. **17-20** - ERROR. **21-24** - FATAL.
Trace-ID	ID des Trace, während dessen der Protokolldatensatz erzeugt wurde, sofern vorhanden.
Span-ID	ID des Spans, während dessen der Protokolldatensatz erzeugt wurde, sofern vorhanden.
Flags	Trace-Flags des Protokolldatensatzes: **0** - Der Trace wird nicht erfasst. **1** - Der Trace wird erfasst (aufgezeichnet). Spalte der Datenquelle: `TraceFlags`.
Servicename	Service, der den Protokolldatensatz erzeugt hat.
Scope-Name	Instrumentierungs-Scope, der den Protokolldatensatz erzeugt hat.
Scope-Version	Version des Instrumentierungs-Scopes.
Ereignisname	Name des Ereignistyps, den der Protokolldatensatz darstellt, sofern vorhanden.
Protokollattribute	Protokollattribute des Protokolldatensatzes. Klicken Sie auf die Drei-Punkte-Schaltfläche ••• , um alle Attribute anzuzeigen.
Ressourcenattribute	Ressourcenattribute des Protokolldatensatzes. Klicken Sie auf die Drei-Punkte-Schaltfläche ••• , um alle Attribute anzuzeigen.

Spalte	Beschreibung
Scope-Attribute	Attribute des Instrumentierungs-Scopes. Klicken Sie auf die Drei-Punkte-Schaltfläche ••• , um alle Attribute anzuzeigen.

Protokolldetails

Klicken Sie in der Protokollliste auf einen Protokolldatensatz, um das Fenster *Protokolldetails* zu öffnen, in dem alle Informationen zum Protokolldatensatz angezeigt werden.

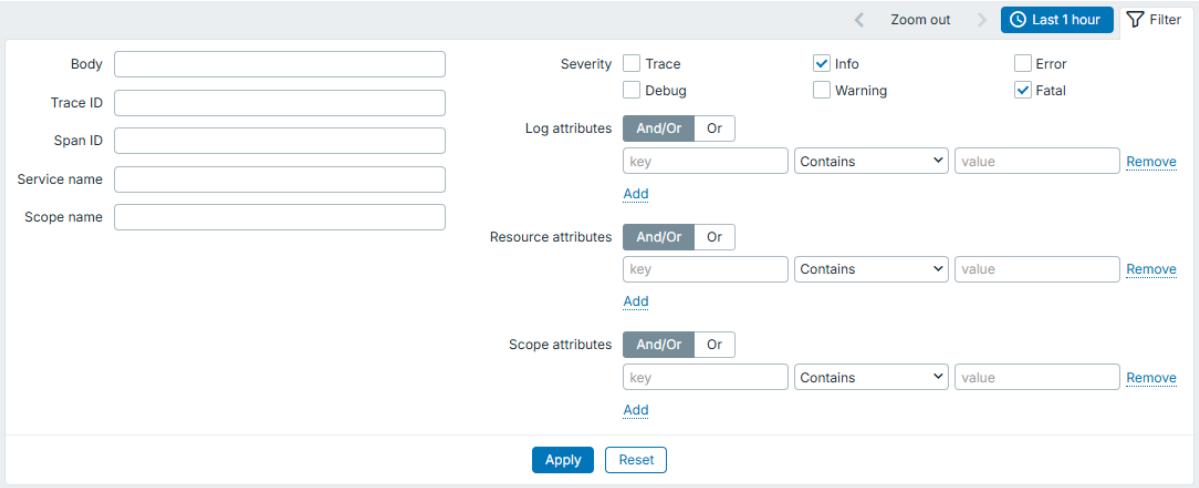


Das Fenster *Protokolldetails* kann die folgenden Abschnitte enthalten:

Abschnitt	Beschreibung
Grundlegende Informationen	Trace-ID, Span-ID und Trace-Flags des Protokolldatensatzes.
Protokollattribute	Protokollattribute des Protokolldatensatzes.
Ressourcenattribute	Schema-URL und Ressourcenattribute.
Bereichsattribute	Schema-URL, Name, Version und Attribute des Instrumentierungsbereichs.

Filter verwenden

Sie können den Filter verwenden, um nur die Protokolldatensätze anzuzeigen, die Sie interessieren.



Um nach Attributen zu filtern, geben Sie den Attributschlüssel ein, wählen Sie den Operator aus und geben Sie den Attributwert ein.

Die folgenden Operatoren werden unterstützt:

- **Enthält** - *(Standard)* der Attributwert enthält die eingegebene Zeichenfolge.
- **Ist gleich** - der Attributwert entspricht der eingegebenen Zeichenfolge.
- **Enthält nicht** - der Attributwert enthält die eingegebene Zeichenfolge nicht.
- **Ist nicht gleich** - der Attributwert entspricht nicht der eingegebenen Zeichenfolge.
- **Existiert** - das Attribut mit dem eingegebenen Schlüssel existiert.
- **Existiert nicht** - das Attribut mit dem eingegebenen Schlüssel existiert nicht.

Wenn Sie mehrere Bedingungen hinzufügen, wählen Sie aus, wie diese ausgewertet werden sollen:

- **Und/Oder** - *(Standard)* alle Bedingungen müssen erfüllt sein; Bedingungen mit demselben Attributschlüssel werden mit *Oder* verknüpft.
- **Oder** - mindestens eine Bedingung muss erfüllt sein.

Dienste

Übersicht

Das Menü **Services** ist für die Funktionen der Service-Überwachung von Zabbix vorgesehen.

Dienste

Übersicht

In diesem Abschnitt sehen Sie den übergeordneten Status aller in Zabbix konfigurierten Services, basierend auf Ihrer Infrastruktur.

Ein Service kann eine Hierarchie bilden, die aus mehreren Ebenen anderer Services besteht, den sogenannten „untergeordneten“ Services. Diese beeinflussen den Gesamtstatus des Services (siehe auch eine Übersicht über die Funktionalität der Serviceüberwachung).

Die Hauptkategorien des Servicestatus sind *OK* oder *Problem*. Der Status *Problem* wird durch den entsprechenden Namen und die Farbe des Problem-Schweregrads dargestellt.

Im Ansichtsmodus können Services zusammen mit ihrem Status und anderen Details überwacht werden. Sie können in diesem Abschnitt jedoch auch die Servicehierarchie konfigurieren (Services und untergeordnete Services hinzufügen/bearbeiten), indem Sie in den Bearbeitungsmodus wechseln.

Um vom Ansichtsmodus in den Bearbeitungsmodus (und zurück) zu wechseln, klicken Sie auf die entsprechende Schaltfläche in der oberen rechten Ecke:

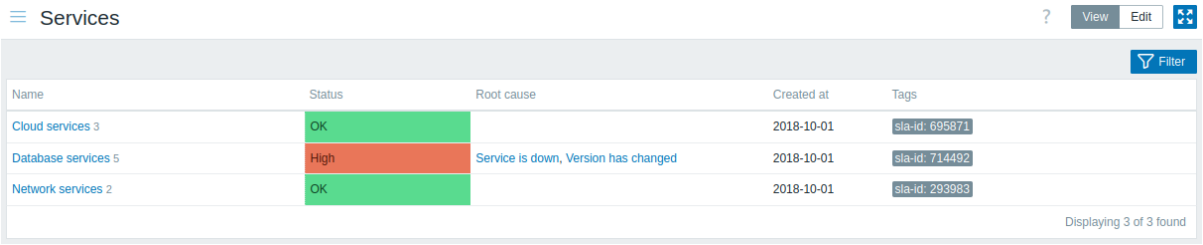
- 

- Services anzeigen
- 

- Services und untergeordnete Services hinzufügen/bearbeiten

Beachten Sie, dass der Zugriff auf die Bearbeitung von den Einstellungen der Benutzerrolle abhängt.

Anzeigen von Diensten



Eine Liste der vorhandenen Dienste wird angezeigt.

Angezeigte Daten:

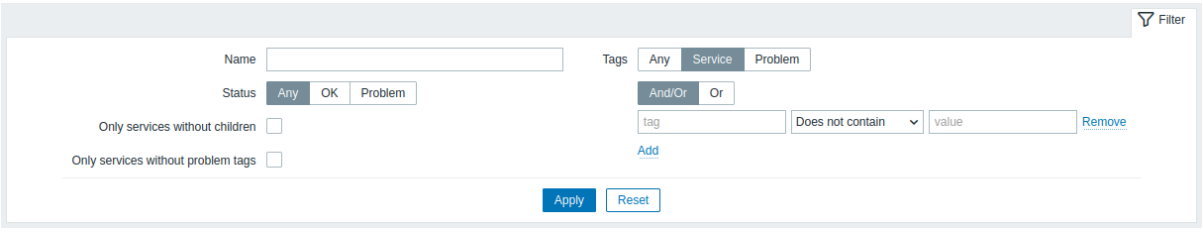
Parameter	Beschreibung
Name	Dienstname. Der Dienstname ist ein Link zu den Dienstdetails. Die Zahl hinter dem Namen gibt an, wie viele Kinddienste der Dienst hat.
Status	Dienststatus: **OK** - keine Probleme **<problem color and severity>** - weist auf ein Problem und dessen Schweregrad hin. Bei mehreren Problemen werden die Farbe und der Schweregrad des kritischsten Problems angezeigt.
Root cause	Es werden zugrunde liegende Probleme aufgelistet, die den Dienststatus direkt oder indirekt beeinflussen. Dieselben Probleme werden auch so aufgelistet, wie sie von der {SERVICE.ROOTCAUSE} macro zurückgegeben werden. Klicken Sie auf den Problemnamen, um weitere Details dazu unter *Monitoring* > *Probleme* anzuzeigen. Probleme, die den Dienststatus nicht beeinflussen, sind nicht in der Liste enthalten.
Created at	Die Zeit, zu der der Dienst erstellt wurde, wird angezeigt.
Tags	Tags des Dienstes werden angezeigt. Tags werden verwendet, um einen Dienst in Dienst-Aktionen und SLAs zu identifizieren.

Schaltflächen

Die Schaltflächen des Ansichtsmodus, die in allen Abschnitten gemeinsam verwendet werden, sind auf der Seite Monitoring beschrieben.

Verwendung des Filters

Sie können den Filter verwenden, um nur die Services anzuzeigen, die Sie interessieren.

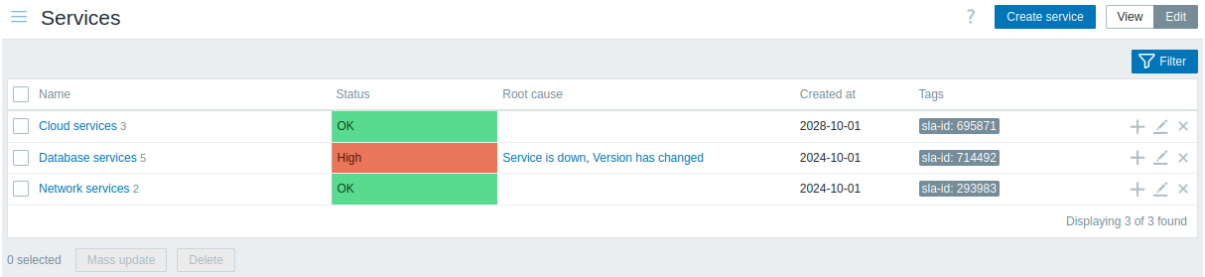


Parameter	Beschreibung
Name	Nach Servicenamen filtern.
Status	Nach Servicestatus filtern.
Only services without children	Aktivieren Sie das Kontrollkästchen, um nur Services ohne untergeordnete Services anzuzeigen. Dieser Parameter ist nur im Service-Bearbeitungsmodus verfügbar.
Only services without problem tags	Aktivieren Sie das Kontrollkästchen, um nur Services ohne Problem-Tags anzuzeigen. Dieser Parameter ist nur im Service-Bearbeitungsmodus verfügbar.
Tags	Nach Service-Tag-Name und -Wert oder nach Name und Wert des Service-Problem-Tags filtern (im Service-Bearbeitungsmodus). Es ist möglich, bestimmte Tags und Tag-Werte sowohl einzuschließen als auch auszuschließen. Es können mehrere Bedingungen festgelegt werden. Der Abgleich von Tag-Namen ist immer groß-/kleinschreibungssensitiv. Für jede Bedingung stehen mehrere Operatoren zur Verfügung: **Exists** - die angegebenen Tag-Namen einschließen; **Equals** - die angegebenen Tag-Namen und -Werte einschließen (groß-/kleinschreibungssensitiv); **Contains** - die angegebenen Tag-Namen einschließen, bei denen die Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv); **Does not exist** - die angegebenen Tag-Namen ausschließen; **Does not equal** - die angegebenen Tag-Namen und -Werte ausschließen (groß-/kleinschreibungssensitiv); **Does not contain** - die angegebenen Tag-Namen ausschließen, bei denen die Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv). Für Bedingungen gibt es zwei Berechnungstypen: **And/Or** - alle Bedingungen müssen erfüllt sein; Bedingungen mit demselben Tag-Namen werden durch die Oder-Bedingung gruppiert; **Or** - es genügt, wenn eine Bedingung erfüllt ist.

Bearbeiten von Services

Klicken Sie auf die Schaltfläche *Bearbeiten*, um den Bearbeitungsmodus aufzurufen. Im Bearbeitungsmodus wird die Liste durch Kontrollkästchen vor den Einträgen ergänzt und außerdem um diese zusätzlichen Optionen erweitert:

-  - einen untergeordneten Service zu diesem Service hinzufügen
-  - diesen Service bearbeiten
-  - diesen Service löschen

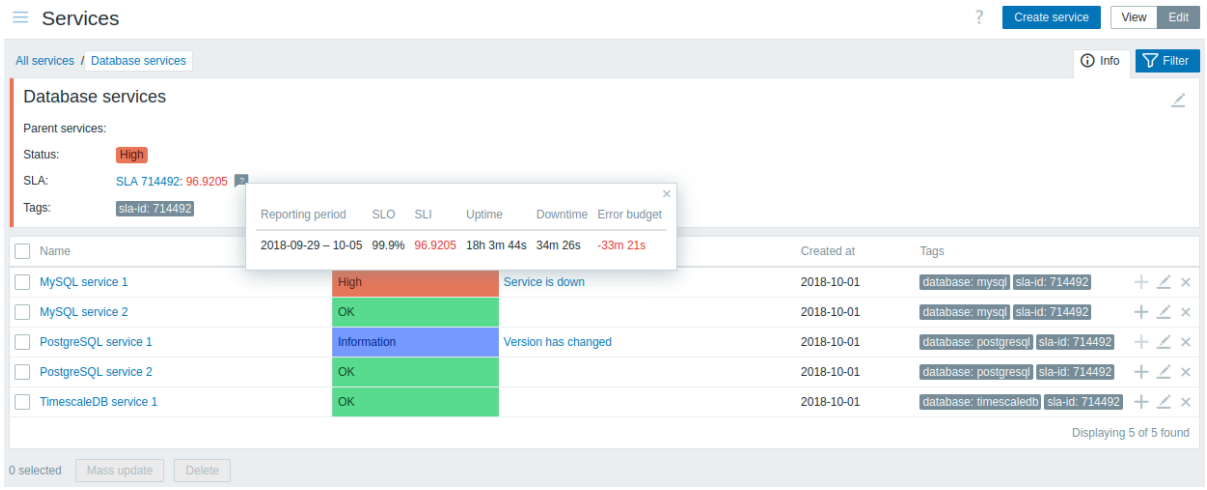


Um einen neuen Service zu konfigurieren, klicken Sie auf die Schaltfläche *Service erstellen* in der oberen rechten Ecke.

Servicedetails

Um auf die Servicedetails zuzugreifen, klicken Sie auf den Servicenamen. Um zur Liste aller Services zurückzukehren, klicken Sie auf *Alle Services*.

Die Servicedetails umfassen die Infobox und die Liste der untergeordneten Services.



Um auf die Infobox zuzugreifen, klicken Sie auf die Registerkarte *Info*. Die Infobox enthält die folgenden Einträge:

- Namen der übergeordneten Services (falls vorhanden)
- Aktueller Status dieses Services
- Aktuelle SLA(s) dieses Services im Format `SLA name:service level indicator`. 'SLA name' ist außerdem ein Link zum SLA-Bericht für diesen Service. Wenn Sie den Mauszeiger in der Infobox neben den Service-Level-Indikator (SLI) bewegen, wird eine Popup-Infoliste mit SLI-Details angezeigt. Der Service-Level-Indikator zeigt den aktuellen Service-Level in Prozent an.
- Service-Tags

Die Infobox enthält außerdem einen Link zur Servicekonfiguration.

Um den Filter für untergeordnete Services zu verwenden, klicken Sie auf die Registerkarte *Filter*.

Im Bearbeitungsmodus wird die Liste der untergeordneten Services um zusätzliche Bearbeitungsoptionen ergänzt:

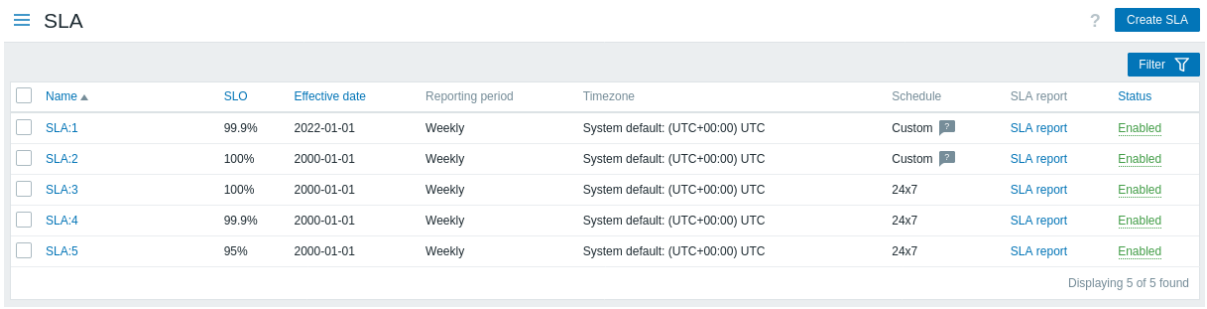
-  - einen untergeordneten Service zu diesem Service hinzufügen
-  - diesen Service bearbeiten
-  - diesen Service löschen

SLA

Übersicht

In diesem Abschnitt können SLAs angezeigt und konfiguriert werden.

SLAs



Eine Liste der konfigurierten SLAs wird angezeigt. *Hinweis*: Nur die SLAs, die sich auf für den Benutzer zugängliche Services beziehen, werden angezeigt (schreibgeschützt, sofern für die Benutzerrolle nicht *Manage SLA* aktiviert ist).

Angezeigte Daten:

Parameter	Beschreibung
Name	Der SLA-Name wird angezeigt. Der Name ist ein Link zur SLA-Konfiguration.
SLO	Das Service-Level-Ziel (SLO) wird angezeigt.
Effective date	Das Datum des Beginns der SLA-Berechnung wird angezeigt.

Parameter	Beschreibung
Reporting period	Der im SLA-Bericht verwendete Zeitraum wird angezeigt - *täglich*, *wöchentlich*, *monatlich*, *vierteljährlich* oder *jährlich*.
Time zone	Die Zeitzone des SLA wird angezeigt.
Schedule	Der SLA-Zeitplan wird angezeigt - 24x7 oder benutzerdefiniert.
SLA report	Klicken Sie auf den Link, um den SLA-Bericht für dieses SLA anzuzeigen.
Status	Der SLA-Status wird angezeigt - aktiviert oder deaktiviert.

SLA-Bericht

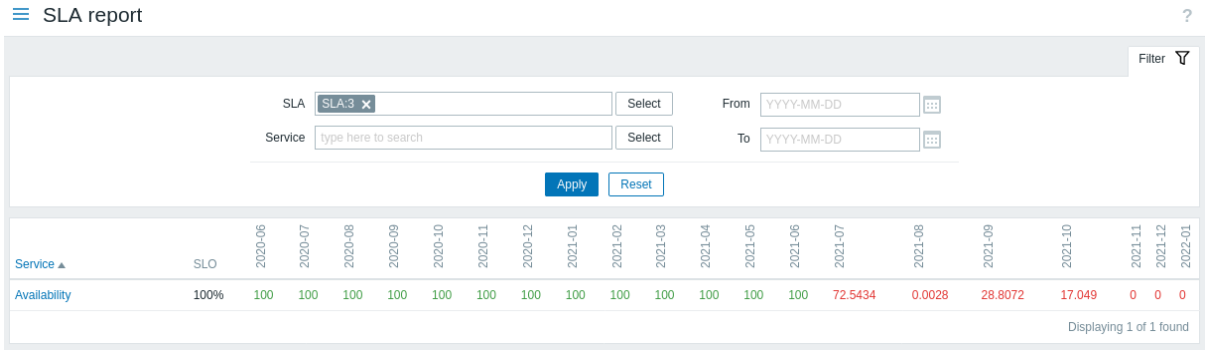
Übersicht

In diesem Abschnitt können SLA-Berichte basierend auf den im Filter ausgewählten Kriterien angezeigt werden.

SLA-Berichte können auch als Dashboard-Widget angezeigt werden.

Bericht

Mit dem Filter können Sie den Bericht anhand des SLA-Namens sowie des Servicenamens auswählen. Außerdem kann der angezeigte Zeitraum eingeschränkt werden.

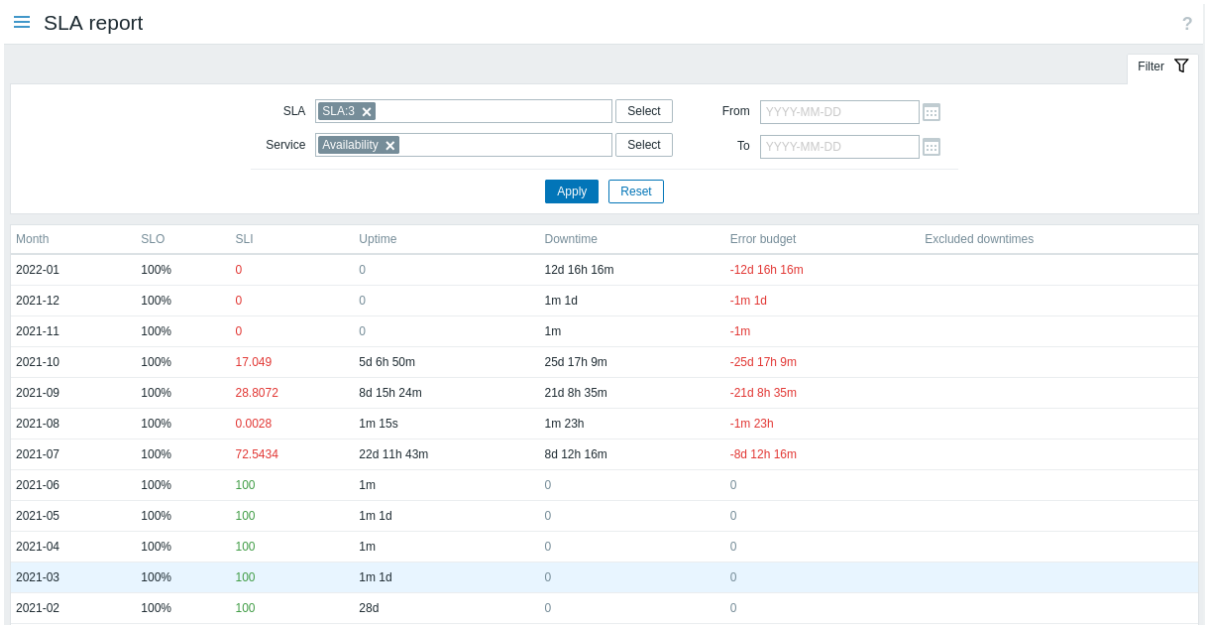


Jede Spalte (Zeitraum) zeigt den SLI für diesen Zeitraum an. SLIs, die das festgelegte SLO verletzen, werden rot hervorgehoben.

Im Bericht werden 20 Zeiträume angezeigt. Es können maximal 100 Zeiträume angezeigt werden, wenn sowohl das Datum *Von* als auch das Datum *Bis* angegeben sind.

Berichtdetails

Wenn Sie im Bericht auf den Servicenamen klicken, können Sie einen weiteren Bericht aufrufen, der eine detailliertere Ansicht anzeigt.



Beachten Sie, dass sich negative Problemdauer nicht auf die SLA-Berechnung oder die Berichterstellung auswirkt.

Inventar

Übersicht

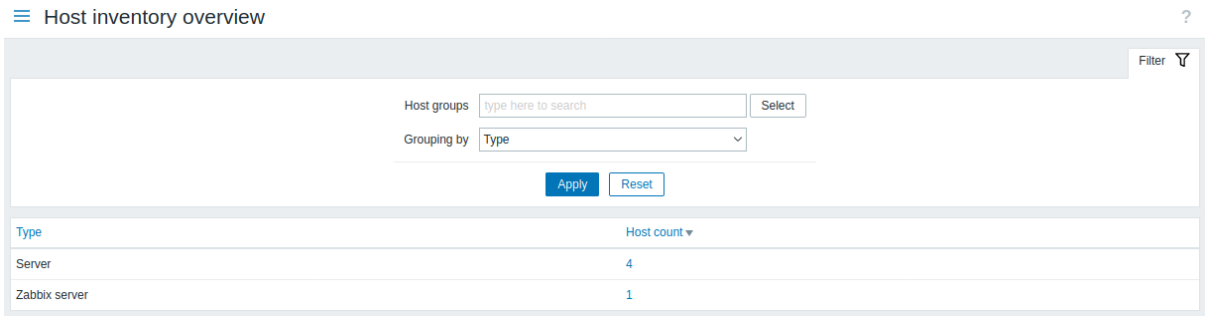
Das Menü „Inventar“ enthält Abschnitte, die eine Übersicht über die Inventardaten von Hosts nach einem ausgewählten Parameter bieten, sowie die Möglichkeit, Details zum Host-Inventar anzuzeigen.

Übersicht

Übersicht

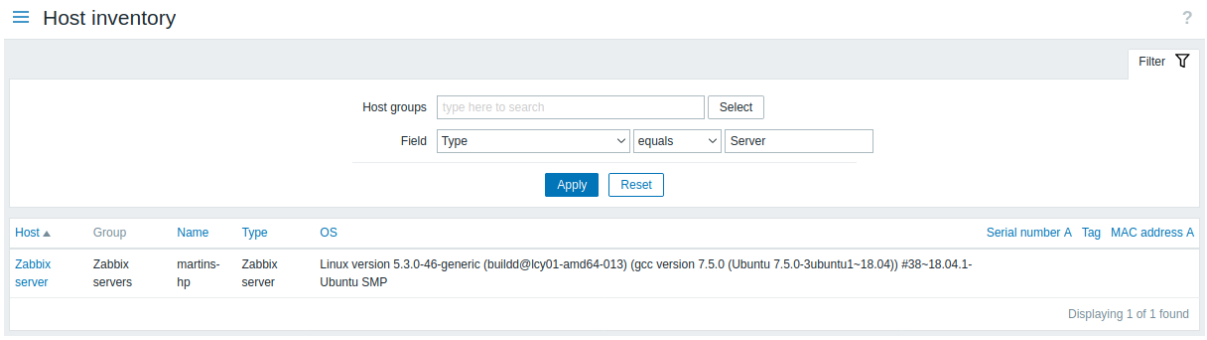
Der Abschnitt *Inventory* > *Overview* bietet Möglichkeiten, eine Übersicht über die Daten des Host-Inventars zu erhalten.

Damit eine Übersicht angezeigt wird, wählen Sie Host-Gruppen (oder keine) und das Inventarfeld aus, nach dem die Daten angezeigt werden sollen. Die Anzahl der Hosts, die jedem Eintrag des ausgewählten Feldes entsprechen, wird angezeigt.



Die Vollständigkeit einer Übersicht hängt davon ab, wie viele Inventarinformationen bei den Hosts gepflegt werden.

Die Zahlen in der Spalte *Host count* sind Links; sie führen dazu, dass diese Hosts in der Tabelle *Host Inventories* gefiltert angezeigt werden.

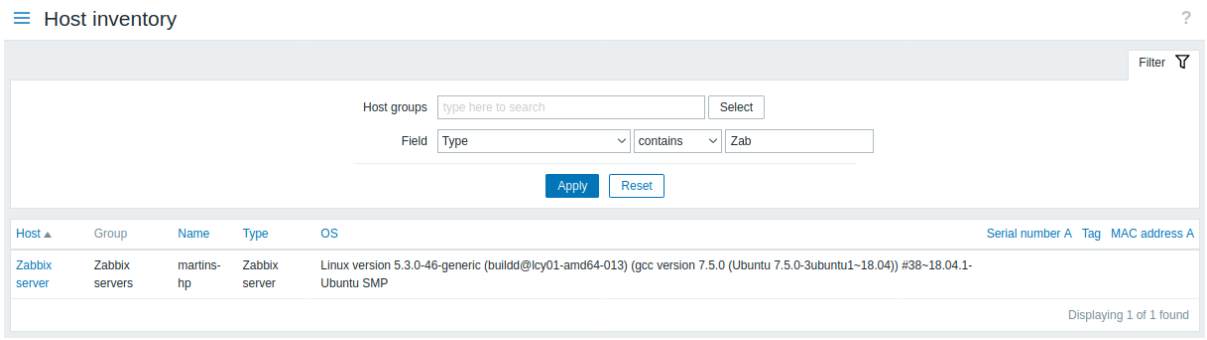


Hosts

Übersicht

Im Abschnitt *Inventory* > *Hosts* werden Inventardaten von Hosts angezeigt.

Sie können die Hosts nach Hostgruppen und nach beliebigen Inventarfeldern filtern, um nur die Hosts anzuzeigen, die Sie interessieren.

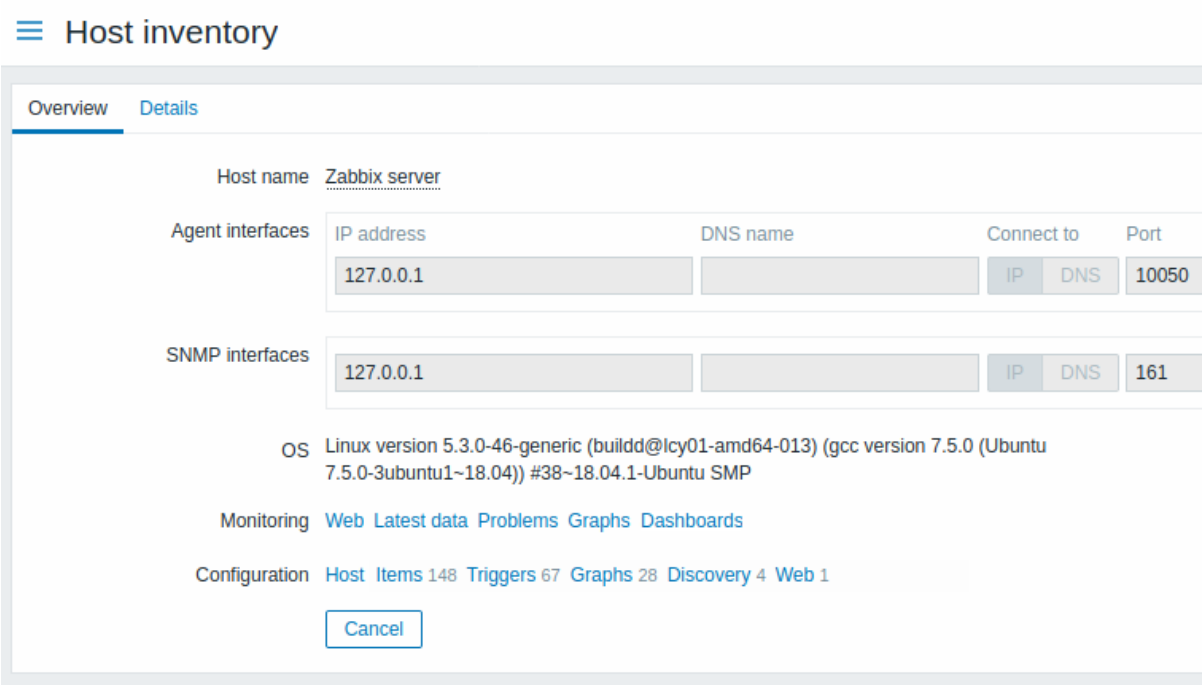


Um alle Host-Inventare anzuzeigen, klicken Sie auf die Schaltfläche *Reset*.

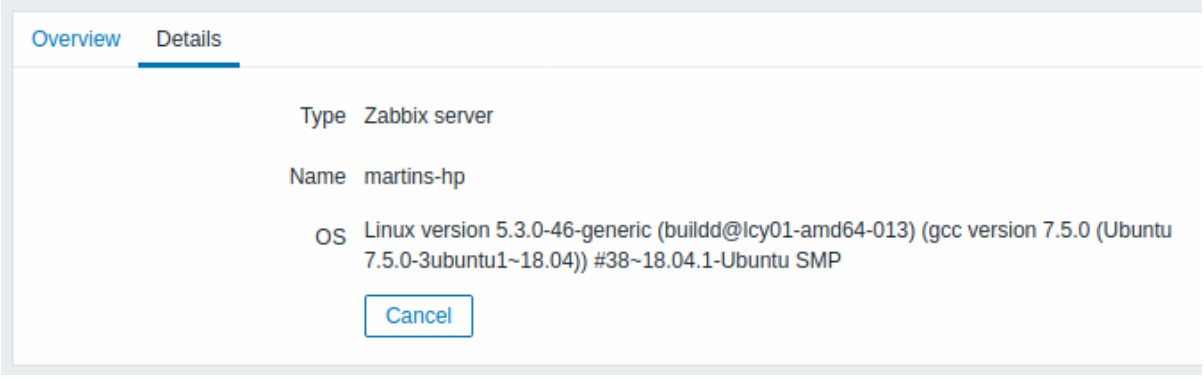
Während in der Tabelle nur einige wichtige Inventarfelder angezeigt werden, können Sie auch alle verfügbaren Inventarinformationen für diesen Host anzeigen. Klicken Sie dazu in der ersten Spalte auf den Hostnamen.

Inventardetails

Die Registerkarte **Übersicht** enthält einige allgemeine Informationen über den Host, die neuesten Monitoring-Daten und Konfigurationsoptionen des Hosts:



Die Registerkarte **Details** enthält alle verfügbaren Inventardetails für den Host:



Die Vollständigkeit der Inventardaten hängt davon ab, wie viele Inventarinformationen für den Host gepflegt werden. Wenn keine Informationen gepflegt werden, ist die Registerkarte *Details* deaktiviert.

Berichte

Übersicht

Das Menü „Berichte“ umfasst mehrere Abschnitte mit verschiedenen vordefinierten und vom Benutzer anpassbaren Berichten, die darauf ausgerichtet sind, eine Übersicht über Parameter wie Systeminformationen, Auslöser und erfasste Daten anzuzeigen.

Systeminformationen

Übersicht

Unter *Berichte* > *Systeminformationen* wird eine Zusammenfassung wichtiger Daten des Zabbix-Servers und des Systems angezeigt. Systemdaten werden mithilfe von internen Datenpunkten erfasst.

Der Bericht mit den Systeminformationen ist nur für Zabbix-*Super-Admins* und Zabbix-*Admins* sichtbar.

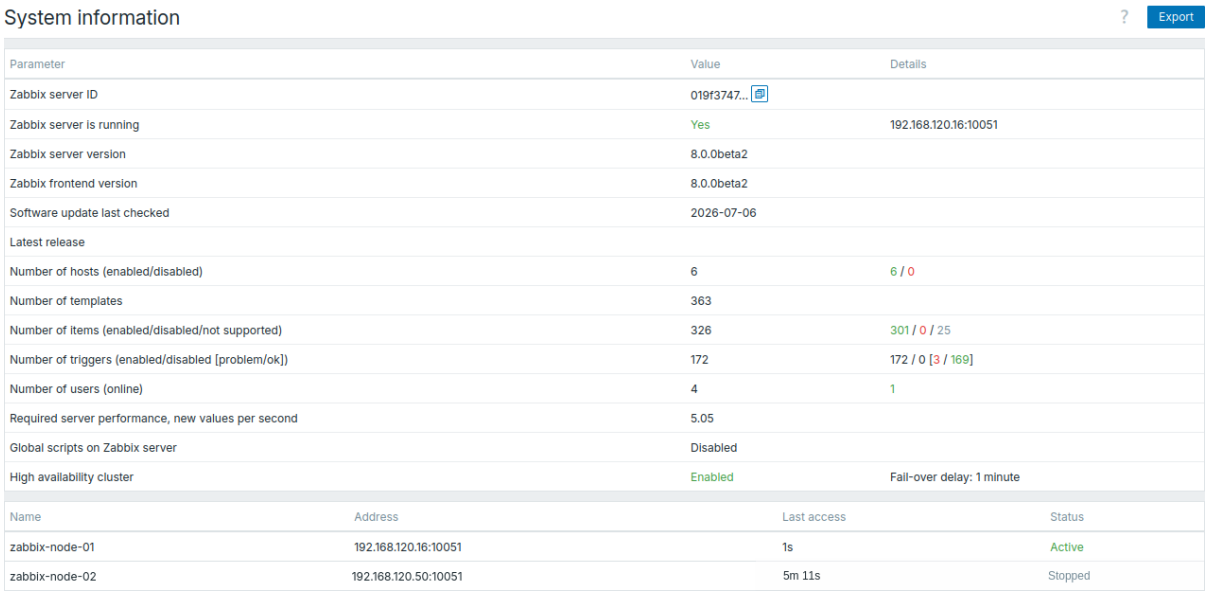
Beachten Sie, dass in einer Hochverfügbarkeitskonfiguration die Quelle der Systeminformationen (Serverinstanz) umgeleitet werden kann. Bearbeiten Sie dazu die Datei Frontend-Konfigurationsdatei - entfernen Sie die Kommentarzeichen vor `$ZBX_SERVER`

oder sowohl vor $ZBX_SERVER als auch vor $ZBX_SERVER_PORT und setzen Sie den Wert auf einen anderen Server als den aktuell angezeigten. Wenn nur $ZBX_SERVER festgelegt wird, wird der Standardwert (10051) für $ZBX_SERVER_PORT verwendet.

Wenn die Hochverfügbarkeitskonfiguration aktiviert ist, wird unterhalb der Systemstatistiken ein separater Bereich mit Details zu den Hochverfügbarkeitsknoten angezeigt.

Systeminformationen sind auch als Widget in einem Dashboard verfügbar.

Systemstatistiken



Angezeigte Daten:

Parameter	Wert	Details
Zabbix-Server-ID	Eindeutiger Bezeichner des Zabbix-Servers (UUIDv7).	Die Server-ID wird beim ersten Start des Servers generiert. Die ersten 8 Zeichen werden gefolgt von einer Auslassung angezeigt. Wenn Sie den Mauszeiger über einen Server-ID-Wert bewegen oder darauf klicken, wird ein Tooltip mit dem vollständigen Wert angezeigt. Durch Klicken auf das Symbol neben dem Wert wird die ID in die Zwischenablage kopiert. Wenn Hochverfügbarkeit aktiviert ist, verfügt jeder Knoten über eine eigene Knoten-ID für die Adressierung über die Befehlszeile. Alle Knoten werden jedoch unter einer einzigen Server-ID gruppiert, die von dem zuerst gestarteten Knoten (oder einem zuvor eigenständigen Server) generiert wurde, unabhängig davon, welcher Knoten derzeit aktiv ist.
Zabbix-Server läuft	Status des Zabbix-Servers: **Ja** - Server läuft **Nein** - Server läuft nicht *Hinweis:* Damit die übrigen Informationen angezeigt werden können, muss der Server laufen und mindestens ein Trapper-Prozess auf dem Server gestartet sein (der Parameter StartTrappers in der Datei `zabbix_server.conf` > 0).	Standort und Port des Zabbix-Servers.

Parameter	Wert	Details
Zabbix-Serverversion	Die aktuelle Versionsnummer des Servers wird angezeigt. *Hinweis:* Sie wird nur angezeigt, wenn der Zabbix-Server läuft.	Der Status der Serverversion wird angezeigt: **Aktuell** - die neueste Version wird verwendet; **Neues Update verfügbar** - eine aktuellere Version ist verfügbar; **Veraltet** - der vollständige Supportzeitraum für diese Version ist abgelaufen. Diese Informationen sind nur verfügbar, wenn die Prüfung auf Softwareupdates in der Serverkonfiguration von Zabbix aktiviert ist. Wenn die letzte Prüfung auf Softwareupdates länger als eine Woche zurückliegt oder keine Daten zur aktuellen Version vorhanden sind, wird nichts angezeigt.
Zabbix-Frontend-Version	Die Versionsnummer des Zabbix-Frontends wird angezeigt.	Der Status der Frontend-Version von Zabbix wird angezeigt: **Aktuell** - die neueste Version wird verwendet; **Neues Update verfügbar** - eine aktuellere Version ist verfügbar; **Veraltet** - der vollständige Supportzeitraum für diese Version ist abgelaufen. Diese Informationen sind nur verfügbar, wenn die Prüfung auf Softwareupdates in der Serverkonfiguration von Zabbix aktiviert ist. Wenn die letzte Prüfung auf Softwareupdates länger als eine Woche zurückliegt oder keine Daten zur aktuellen Version vorhanden sind, wird nichts angezeigt.
Letzte Prüfung auf Softwareupdates	Das Datum der letzten Prüfung auf Zabbix-Softwareupdates wird angezeigt. Diese Informationen sind nur verfügbar, wenn die Prüfung auf Softwareupdates in der Serverkonfiguration von Zabbix aktiviert ist.	
Neueste Version	Die Nummer einer neueren Version (falls verfügbar) für die aktuelle Zabbix-Version wird angezeigt. Diese Informationen sind nur verfügbar, wenn die Prüfung auf Softwareupdates in der Serverkonfiguration von Zabbix aktiviert ist. Wenn die letzte Prüfung auf Softwareupdates länger als eine Woche zurückliegt oder keine Daten zur aktuellen Version vorhanden sind, wird nichts angezeigt.	Ein Link zu den Versionshinweisen der neuesten verfügbaren Zabbix-Version wird angezeigt.
Anzahl der Hosts	Die Gesamtzahl der konfigurierten Hosts wird angezeigt.	Anzahl der überwachten/nicht überwachten Hosts.
Anzahl der Vorlagen	Die Gesamtzahl der Vorlagen wird angezeigt.	
Anzahl der Datenpunkte	Die Gesamtzahl der Datenpunkte wird angezeigt.	Anzahl der überwachten/deaktivierten/nicht unterstützten Datenpunkte auf Host-Ebene. Datenpunkte auf deaktivierten Hosts werden als deaktiviert gezählt.
Anzahl der Auslöser	Die Gesamtzahl der Auslöser wird angezeigt.	Anzahl der aktivierten/deaktivierten Auslöser auf Host-Ebene; Aufteilung der aktivierten Auslöser nach den Zuständen „Problem“/„OK“. Unter dem Zustand „OK“ aufgeführte Auslöser umfassen Auslöser mit dem Status „Unbekannt“. Auslöser, die von deaktivierten Datenpunkten abhängig oder deaktivierten Hosts zugewiesen sind, werden als deaktiviert gezählt.
Anzahl der Benutzer	Die Gesamtzahl der konfigurierten Benutzer wird angezeigt.	Anzahl der Benutzer, die online sind.

Parameter	Wert	Details
Erforderliche Serverleistung, neue Werte pro Sekunde	Die erwartete Anzahl neuer Werte, die der Zabbix-Server pro Sekunde verarbeitet, wird angezeigt.	*Erforderliche Serverleistung* ist eine Schätzung und kann als Richtwert dienen. Für genaue Zahlen der verarbeiteten Werte verwenden Sie den internen Datenpunkt `zabbix[wcache,values,all]`. Aktivierte Datenpunkte überwachter Hosts werden in die Berechnung einbezogen. Protokolldatenpunkte werden pro Aktualisierungsintervall des Datenpunkts als ein Wert gezählt. Werte mit regulären Intervallen werden gezählt; Werte mit flexiblen und geplanten Intervallen werden nicht gezählt. Während eines Wartungszeitraums mit „nodata“ wird die Berechnung nicht angepasst. Trapper-Datenpunkte werden nicht gezählt.
Globale Skripte auf dem Zabbix-Server	**Deaktiviert** wird in diesem Feld angezeigt, wenn globale Skripte auf dem Zabbix-Server deaktiviert sind, indem EnableGlobalScripts=0 in der Serverkonfiguration festgelegt wird.	
Hochverfügbarkeitsstatus	Status des Hochverfügbarkeitsclusters für den Zabbix-Server: **Deaktiviert** - eigenständiger Server **Aktiviert** - mindestens ein Hochverfügbarkeitsknoten ist vorhanden	Wenn aktiviert, wird die Failover-Verzögerung angezeigt.

Systeminformationen zeigen außerdem unter den folgenden Bedingungen eine Fehlermeldung an:

- Die verwendete Datenbank verfügt nicht über den erforderlichen Zeichensatz oder die erforderliche Sortierung (UTF-8).
- Die Version der Datenbank liegt unter- oder oberhalb des unterstützten Bereichs (nur für Benutzer mit dem Rollentyp *Super-Admin* verfügbar).
- Housekeeping für TimescaleDB ist falsch konfiguriert (Verlaufs- oder Trendtabellen enthalten komprimierte Chunks, aber die Optionen *Datenpunkt-Verlaufszeitraum überschreiben* oder *Datenpunkt-Trendzeitraum überschreiben* sind deaktiviert).

Schaltflächen

Die Schaltfläche in der oberen rechten Ecke der Seite bietet die folgende Option:

Export	Systeminformationen in eine JSON-Datei exportieren. Die Datei wird als `Zabbix-system-information-{time}.json` gespeichert, wobei `{time}` der Unix-Zeitstempel ist, der angibt, wann die Daten auf dem Server vorbereitet wurden. Wenn Hochverfügbarkeit aktiviert ist, werden auch Details zu den Hochverfügbarkeitsknoten exportiert.

Hochverfügbarkeitsknoten

Wenn High availability cluster aktiviert ist, wird ein zusätzlicher Datenblock angezeigt, der den Status jedes Hochverfügbarkeitsknotens zeigt.



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	Standby

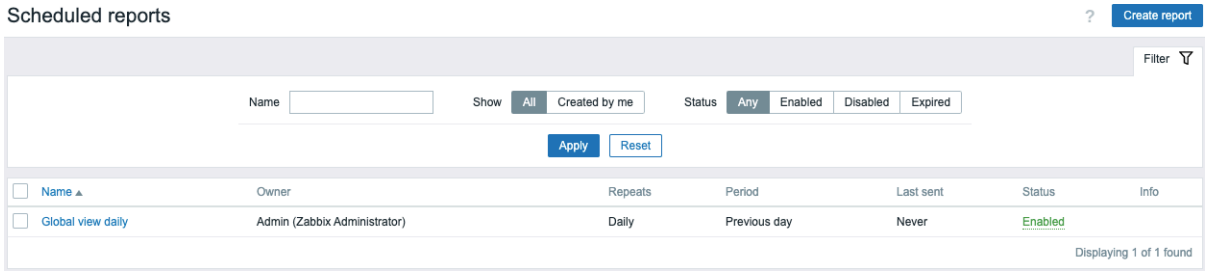
Angezeigte Daten:

Column	Description
Name	Knotenname, wie in der Serverkonfiguration definiert.
Address	IP-Adresse und Port des Knotens.
Last access	Zeit des letzten Zugriffs auf den Knoten. Wenn Sie den Mauszeiger über die Zelle bewegen, wird der Zeitstempel des letzten Zugriffs im Langformat angezeigt.
Status	Knotenstatus. Die Tabellenzeilen werden nach diesen Status in folgender Prioritätsreihenfolge sortiert: **Active** - Knoten ist aktiv und arbeitet **Unavailable** - Knoten wurde länger als die Failover-Verzögerung nicht gesehen (Sie sollten möglicherweise herausfinden, warum) **Stopped** - Knoten wurde gestoppt oder konnte nicht gestartet werden (Sie sollten ihn möglicherweise starten oder löschen) **Standby** - Knoten ist aktiv und wartet

Geplante Berichte

Übersicht

Unter *Berichte* > *Geplante Berichte* können Benutzer mit ausreichenden Berechtigungen die geplante Erstellung von PDF-Versionen der Dashboards konfigurieren, die per E-Mail an angegebene Empfänger gesendet werden.



Der Startbildschirm zeigt Informationen zu geplanten Berichten an. Diese können zur einfacheren Navigation gefiltert werden – siehe den Abschnitt Filter verwenden weiter unten.

Angezeigte Daten:

Spalte	Beschreibung
Name	Name des Berichts. Durch Klicken wird das Formular zur Berichtskonfiguration geöffnet.
Besitzer	Benutzer, der den Bericht erstellt hat.
Wiederholungen	Häufigkeit der Berichtserstellung (täglich/wöchentlich/monatlich/jährlich).
Zeitraum	Zeitraum, für den der Bericht erstellt wird (vorheriger Tag, vorherige Woche, vorheriger Monat oder vorheriges Jahr).
Zuletzt gesendet	Datum und Uhrzeit, zu denen der letzte Bericht gesendet wurde.
Status	Aktueller Status des Berichts (aktiviert/deaktiviert/abgelaufen). Benutzer mit ausreichenden Berechtigungen können den Status durch Klicken ändern – von „Aktiviert“ zu „Deaktiviert“ (und zurück) sowie von „Abgelaufen“ zu „Deaktiviert“ (und zurück). Für Benutzer ohne ausreichende Berechtigungen ist der Status nicht anklickbar.
Info	Zeigt informative Symbole an: Ein rotes Symbol weist darauf hin, dass die Berichtserstellung fehlgeschlagen ist. Wenn Sie den Mauszeiger darüber bewegen, wird eine QuickInfo mit Informationen zum Fehler angezeigt. Ein gelbes Symbol weist darauf hin, dass ein Bericht erstellt wurde, das Senden an einige oder alle Empfänger jedoch fehlgeschlagen ist oder dass ein Bericht abgelaufen ist. Wenn Sie den Mauszeiger darüber bewegen, wird eine QuickInfo mit zusätzlichen Informationen angezeigt.

Verwendung des Filters

Sie können den Filter verwenden, um die Liste der Berichte einzugrenzen. Für eine bessere Suchleistung werden die Daten mit nicht aufgelösten Makros durchsucht.

Die folgenden Filteroptionen sind verfügbar:

- *Name* - teilweise Übereinstimmung des Namens ist zulässig
- *Anzeigen* - vom aktuellen Benutzer erstellt oder alle Berichte

- *Status* - wählen Sie zwischen „Beliebig“ (alle Berichte anzeigen), „Aktiviert“, „Deaktiviert“ oder „Abgelaufen“

Der Filter befindet sich unter dem Abschnittsnamen *Geplante Berichte*. Er kann durch Klicken auf die Registerkarte *Filter* in der oberen rechten Ecke geöffnet und eingeklappt werden.

Massenaktualisierung

Manchmal möchten Sie den Status mehrerer Berichte auf einmal löschen oder ändern. Anstatt jeden einzelnen Bericht zur Bearbeitung zu öffnen, können Sie dafür die Funktion zur Massenaktualisierung verwenden.

Gehen Sie wie folgt vor, um mehrere Berichte per Massenaktualisierung zu aktualisieren:

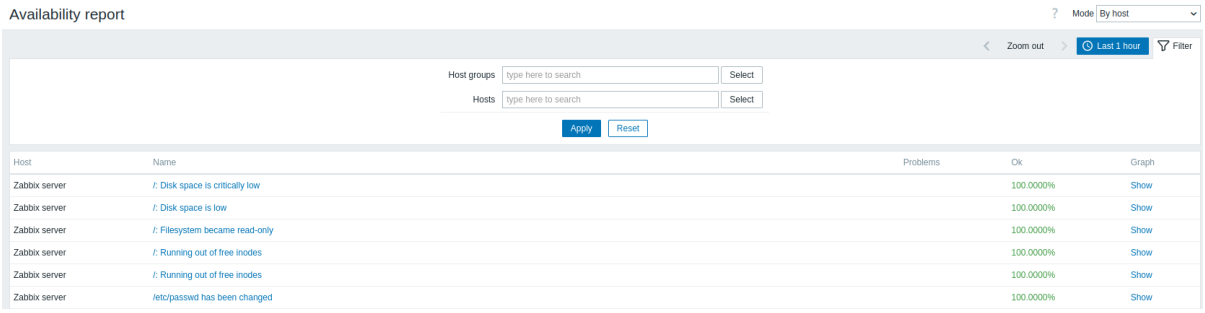
- Aktivieren Sie in der Liste die Kontrollkästchen der zu aktualisierenden Berichte
- Klicken Sie unter der Liste auf die gewünschte Schaltfläche, um die Änderungen vorzunehmen (*Aktivieren*, *Deaktivieren* oder *Löschen*)

Verfügbarkeitsbericht

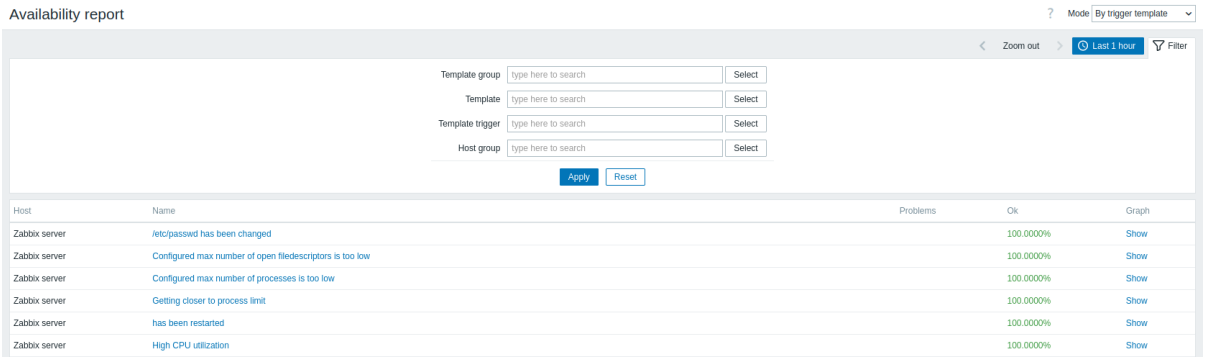
Übersicht

Unter *Berichte* > *Verfügbarkeitsbericht* können Sie sehen, während welchen Anteils der Zeit sich jeder Auslöser im Problem-/OK-Status befunden hat.

Für jeden Status wird ein prozentualer Zeitanteil angezeigt, sodass sich die Verfügbarkeit verschiedener Elemente in Ihrem System leicht bestimmen lässt.



Im Dropdown-Menü in der oberen rechten Ecke können Sie den Auswahlmodus festlegen – ob Auslöser nach Hosts oder nach Auslösern angezeigt werden sollen, die zu einer Vorlage gehören.



Der Name des Auslösers ist ein Link zu den neuesten Ereignissen dieses Auslösers.

Filter verwenden

Der Filter kann dabei helfen, die Anzahl der angezeigten Hosts und/oder Auslöser einzugrenzen. Für eine bessere Suchleistung werden die Daten mit nicht aufgelösten Makros durchsucht.

Der Filter befindet sich unter dem Abschnittsnamen *Verfügbarkeitsbericht*. Er kann durch Klicken auf die Registerkarte *Filter* rechts geöffnet und eingeklappt werden.

Filtern nach Auslöser-Vorlage

Im Modus *Nach Auslöser-Vorlage* können Ergebnisse anhand eines oder mehrerer der unten aufgeführten Parameter gefiltert werden.

Parameter	Beschreibung
Vorlagengruppe	Filtert Hosts nach Auslösern, die von Vorlagen geerbt werden, die zur ausgewählten Vorlagengruppe gehören. Die Angabe einer übergeordneten Vorlagengruppe wählt implizit auch alle verschachtelten Vorlagengruppen aus.
Vorlage	Filtert Hosts nach Auslösern, die von der ausgewählten Vorlage geerbt werden, einschließlich verschachtelter Vorlagen. Wenn eine verschachtelte Vorlage eigene Auslöser hat, werden diese Auslöser nicht angezeigt.
Vorlagen-Auslöser	Filtert Hosts nach dem ausgewählten Auslöser. Andere Auslöser der gefilterten Hosts werden nicht angezeigt.
Hostgruppe	Filtert Hosts, die zur ausgewählten Hostgruppe gehören.

Filtern nach Host

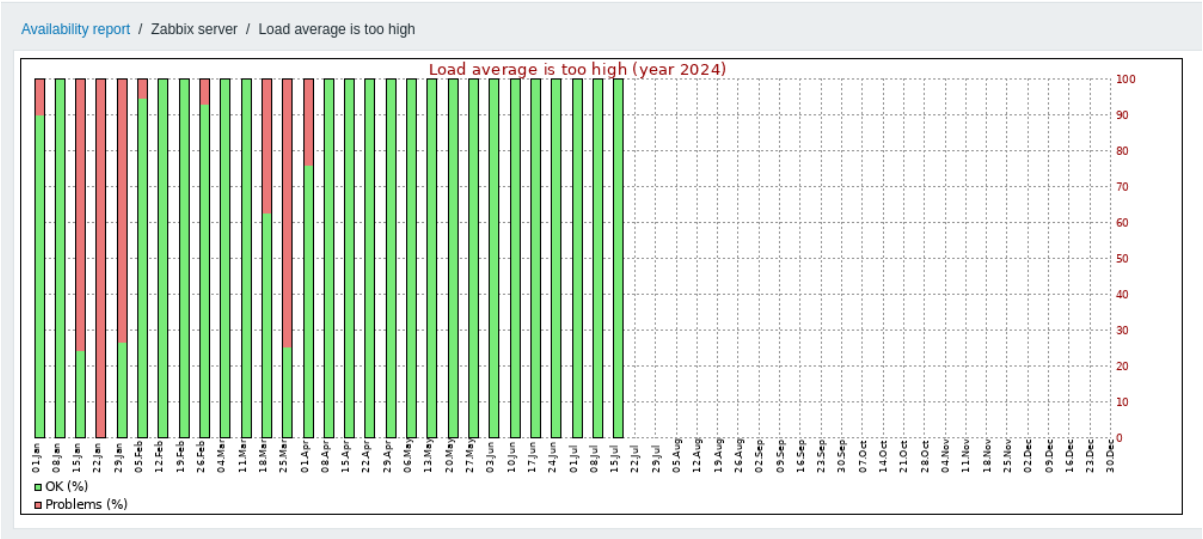
Im Modus *Nach Host* können Ergebnisse nach Host oder Hostgruppe gefiltert werden. Die Angabe einer übergeordneten Hostgruppe wählt implizit alle verschachtelten Hostgruppen aus.

Zeitraum-Auswahl

Mit der Zeitraum-Auswahl können Sie häufig verwendete Zeiträume mit einem Klick auswählen. Die Auswahl kann durch Klicken auf die Registerkarte *Zeitraum* neben dem Filter ein- und ausgeblendet werden.

Wenn Sie in der Spalte *Graph* auf *Anzeigen* klicken, werden Verfügbarkeitsinformationen in einem Balkendiagramm angezeigt, in dem jeder Balken eine vergangene Woche des aktuellen Jahres darstellt.

Availability report graph



Die grüne Farbe eines Balkens steht für die OK-Zeit, die rote für die Problemzeit.

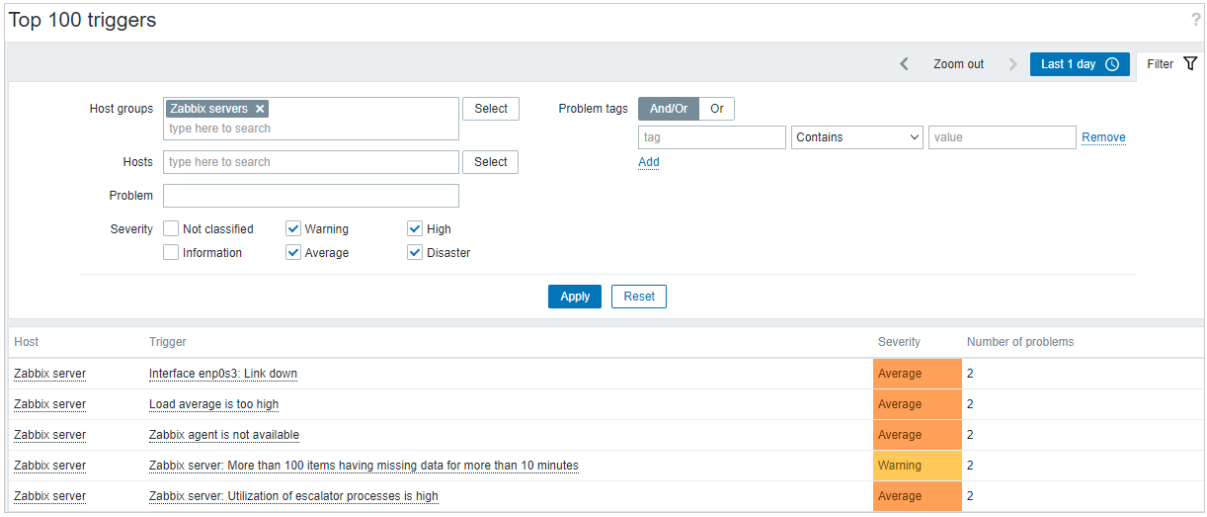
Wirkung von Wartungszeiträumen

Wartung schließt Zeit nicht automatisch aus dem Verfügbarkeitsbericht aus. Nur Wartung, die konfiguriert ist als *Keine Datenerfassung*, stoppt das Abfragen (sodass keine Probleme erzeugt werden) und lässt daher den Verfügbarkeitsprozentsatz für betroffene Auslöser unverändert.

Top 100 Auslöser

Übersicht

Unter *Berichte* > *Top 100 Auslöser* können Sie die Auslöser mit der höchsten Anzahl an Problemen anzeigen, die während des ausgewählten Zeitraums erkannt wurden.



Sowohl die Einträge in den Host- als auch in den Auslöser-Spalten sind Links, die nützliche Optionen bieten:

- für den Host - durch Klicken auf den Hostnamen wird das Host-Menü geöffnet
- für den Auslöser - durch Klicken auf den Auslösernamen werden Links zu den neuesten Ereignissen, zum einfachen Graphen für jeden Auslöser-Datenpunkt sowie zu den Konfigurationsformularen des Auslösers selbst und jedes Auslöser-Datenpunkts angezeigt

Filter verwenden

Sie können den Filter verwenden, um Auslöser nach Hostgruppe, Host, Problemname, Tags oder Auslöser-Schweregrad anzuzeigen. Durch die Angabe einer übergeordneten Hostgruppe werden implizit alle verschachtelten Hostgruppen ausgewählt. Für eine bessere Suchleistung werden die Daten mit nicht aufgelösten Makros durchsucht.

Der Filter befindet sich unterhalb des Abschnittsnamens *Top 100 Auslöser*. Er kann durch Klicken auf die Registerkarte *Filter* auf der rechten Seite geöffnet und eingeklappt werden.

Auswahl des Zeitraums

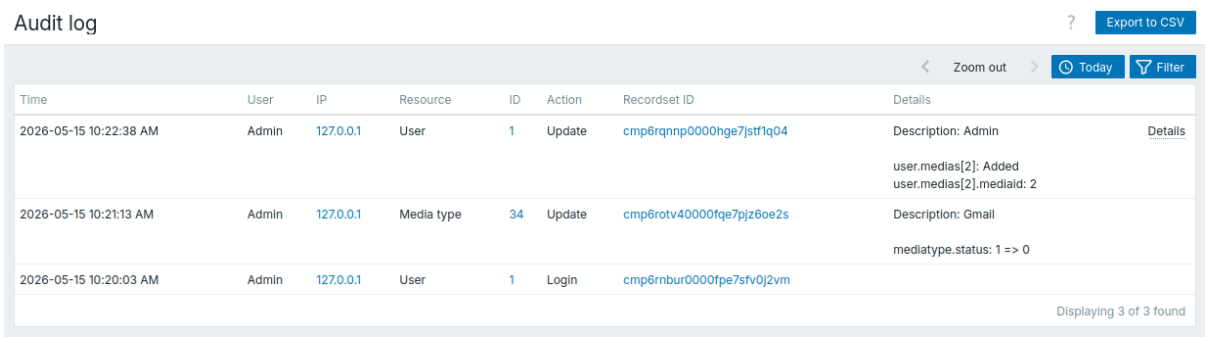
Mit der Auswahl des Zeitraums neben dem Filter können Sie häufig benötigte Zeiträume mit einem Mausklick auswählen. Weitere Informationen finden Sie unter Auswahl von Zeitraum und Host.

Auditprotokoll

Übersicht

Im Abschnitt *Reports* > *Audit log* können die Einträge der Benutzer- und Systemaktivitäten angezeigt werden.

Note:
Damit Audit-Einträge erfasst und angezeigt werden, muss das Kontrollkästchen *Enable audit logging* im Abschnitt *Administration* > *Audit log* aktiviert sein. Ohne diese Einstellung wird die Historie der Aktivitäten nicht in der Datenbank gespeichert und nicht im Audit-Log angezeigt.



Das Audit-Log zeigt die folgenden Daten an:

Column	Description
Time	Zeitstempel des Audit-Eintrags.
User	Benutzer, der die Aktivität ausgeführt hat.

Column	Description
IP	IP, von der aus die Aktivität initiiert wurde. Durch Klicken auf den Hyperlink werden die Audit-Log-Einträge nach dieser IP gefiltert.
Resource	Typ der betroffenen Ressource (*API token*, *Action*, *Authentication*, *Autoregistration* usw.).
ID	ID der betroffenen Ressource. Durch Klicken auf den Hyperlink werden die Audit-Log-Einträge nach dieser Ressourcen-ID gefiltert.
Action	Typ der Aktivität (*Add*, *Configuration refresh*, *Delete*, *Execute*, *Failed login*, *History clear*, *Login*, *Logout*, *Push*, *Update*).
Recordset ID	Gemeinsame ID für alle Audit-Log-Einträge, die als Ergebnis derselben Operation erstellt wurden. Wenn beispielsweise eine Vorlage mit einem Host verknüpft wird, wird für jede geerbte Vorlagen-Entität (Datenpunkt, Auslöser usw.) ein separater Audit-Log-Eintrag erstellt - alle diese Einträge haben dieselbe *Recordset ID*. Durch Klicken auf den Hyperlink werden die Audit-Log-Einträge nach dieser *Recordset ID* gefiltert.
Details	Beschreibung der Ressource und detaillierte Informationen über die ausgeführte Aktivität. Wenn ein Eintrag mehr als zwei Zeilen enthält, wird zusätzlich ein Link *Details* angezeigt. Klicken Sie auf diesen Link, um die vollständige Liste der Änderungen anzuzeigen.

Note:
Wenn ein trapper item oder ein HTTP agent item (mit aktiviertem Trapping) Daten empfangen hat, wird ein Eintrag im Audit-Log nur dann hinzugefügt, wenn die Daten mit der API-Methode `history.push` und nicht mit dem Dienstprogramm Zabbix sender gesendet wurden.

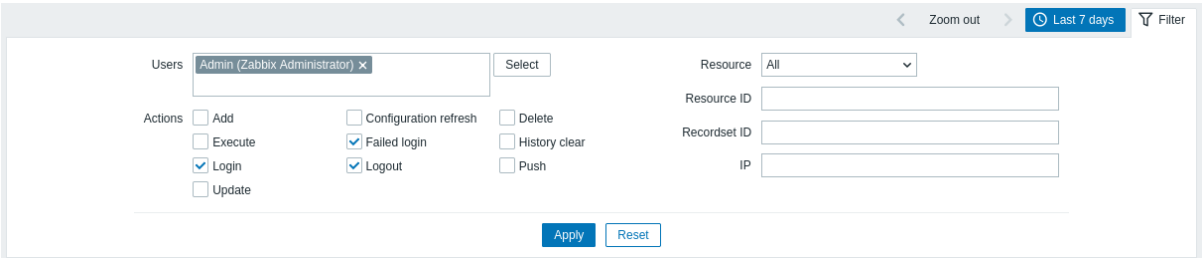
Schaltflächen

Die Schaltfläche in der oberen rechten Ecke der Seite bietet die folgende Option:

Export to CSV	Audit-Protokoll-Einträge von allen Seiten in eine CSV-Datei exportieren. Wenn ein Filter angewendet wird, werden nur die gefilterten Einträge exportiert.

Filter verwenden

Der Filter befindet sich unterhalb der Leiste *Audit log*. Er kann durch Klicken auf die Registerkarte *Filter* in der oberen rechten Ecke geöffnet und eingeklappt werden.



Sie können den Filter verwenden, um die Einträge nach Benutzer, betroffener Ressource, Ressourcen-ID, durchgeführter Operation (*Recordset ID*) und IP einzugrenzen. Je nach Ressource können im Filter eine oder mehrere spezifische Aktionen ausgewählt werden.

Für eine bessere Suchleistung werden alle Daten mit nicht aufgelösten Makros durchsucht.

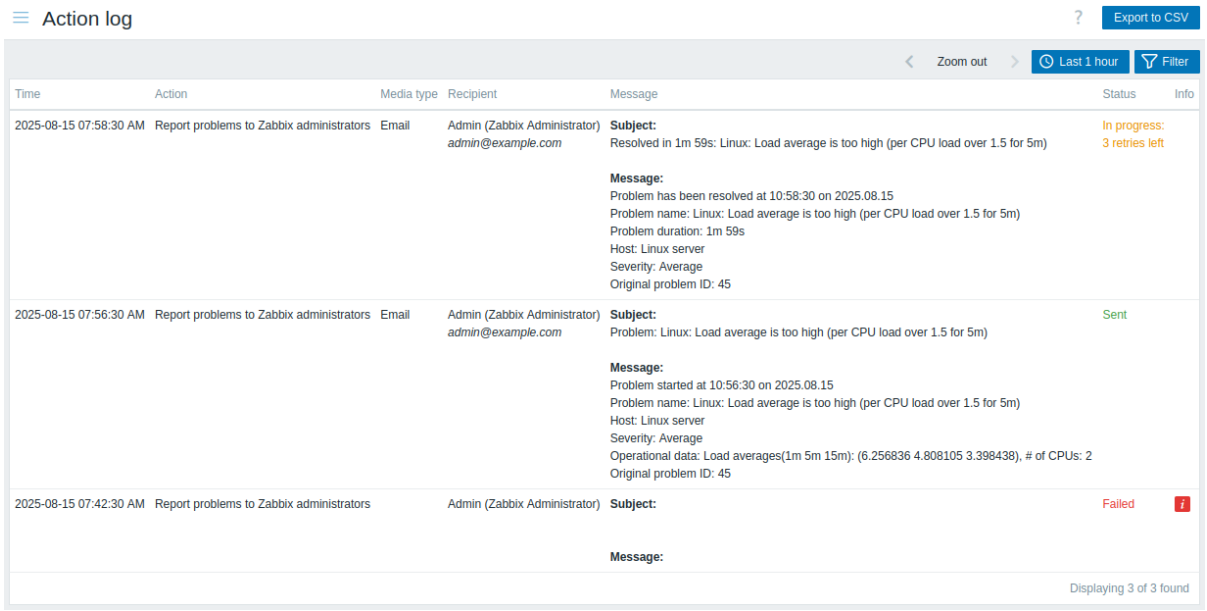
Zeitraum-Auswahl

Mit der Zeitraum-Auswahl neben dem Filter können Sie häufig benötigte Zeiträume mit einem Mausklick auswählen. Weitere Informationen finden Sie unter Zeitraum- und Host-Auswahl.

Aktionsprotokoll

Übersicht

In *Reports* > *Action log* können Sie Details zu Operationen (Benachrichtigungen, Remote-Befehle) anzeigen, die innerhalb einer Aktion ausgeführt wurden.



Angezeigte Daten:

Column	Description
Time	Zeitstempel der Operation.
Action	Name der Aktion, die die Operationen auslöst.
Media type	Medientyp (z. B. E-Mail, Jira usw.), der zum Senden einer Benachrichtigung verwendet wurde. Bei Operationen, die Remote-Befehle ausgeführt haben, bleibt diese Spalte leer.
Recipient	Informationen zum Empfänger der Benachrichtigung - Benutzername, Vorname und Nachname (in Klammern) sowie zusätzliche Informationen je nach Medientyp (E-Mail, Benutzername usw.). Bei Operationen, die Remote-Befehle ausgeführt haben, bleibt diese Spalte leer.
Message	Der Inhalt der Nachricht, des Remote-Befehls oder des Namens des globalen Skripts. Ein Remote-Befehl wird durch ein Doppelpunktzeichen vom Ziel-Host getrennt: `<host>:<command>`. Wenn der Remote-Befehl beispielsweise auf dem Zabbix Server ausgeführt wurde, hat die Information das folgende Format: `Zabbix Server:<command>`.
Status	Status der Operation: *In progress* - die Operation zum Senden einer Benachrichtigung läuft (die verbleibende Anzahl der Versuche des Servers, die Benachrichtigung zu senden, wird ebenfalls angezeigt) *Sent* - die Benachrichtigung wurde gesendet *Executed* - der Remote-Befehl wurde ausgeführt *Failed* - die Operation wurde nicht abgeschlossen
Info	Fehlerinformationen (falls vorhanden) zur Ausführung der Operation.

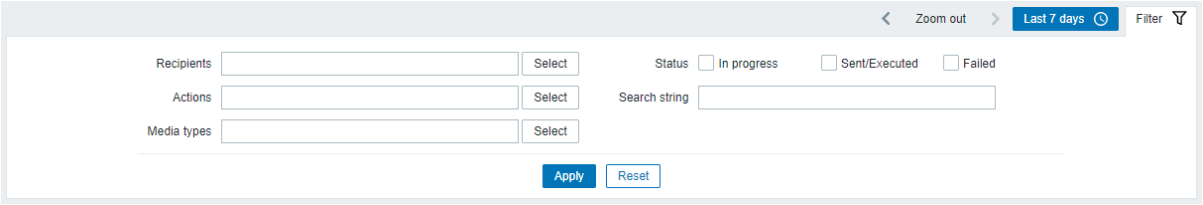
Schaltflächen

Die Schaltfläche in der oberen rechten Ecke der Seite bietet die folgende Option:

Export to CSV	Exportiert Aktionsprotokolleinträge von der aktuellen Seite in eine CSV-Datei. Wenn ein Filter angewendet wird, werden nur die gefilterten Einträge exportiert. In der exportierten CSV-Datei werden die Spalten „Empfänger“ und „Nachricht“ in mehrere Spalten aufgeteilt – „Zabbix-Benutzername des Empfängers“, „Vorname des Empfängers“, „Nachname des Empfängers“, „Empfänger“ sowie „Betreff“, „Nachricht“, „Befehl“.

Verwenden des Filters

Der Filter befindet sich unterhalb der Leiste *Action log*. Er kann durch Klicken auf die Registerkarte *Filter* in der oberen rechten Ecke der Seite geöffnet und eingeklappt werden.



Sie können den Filter verwenden, um die Einträge nach Benachrichtigungsempfängern, Aktionen, Medientypen, Status oder nach dem Inhalt der Nachricht/des Remote-Befehls (*Search string*) einzugrenzen. Für eine bessere Suchleistung werden die Daten mit nicht aufgelösten Makros durchsucht.

Zeitraum-Auswahl

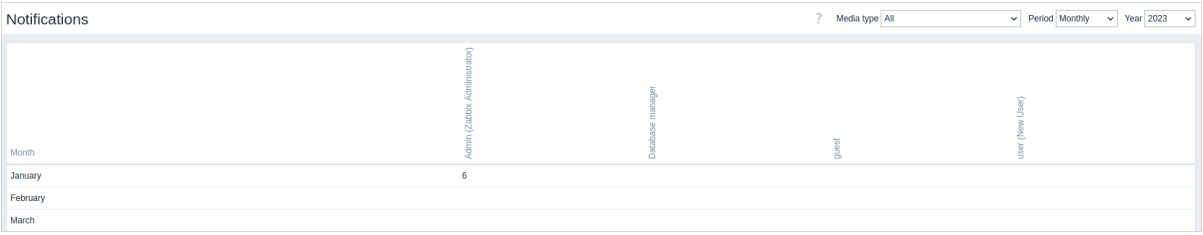
Mit der Zeitraum-Auswahl neben dem Filter können Sie häufig benötigte Zeiträume mit einem Mausklick auswählen.
Weitere Informationen finden Sie unter Zeitraum- und Host-Auswahl.

Benachrichtigungen

Übersicht

Im Abschnitt *Reports* > *Notifications* wird ein Bericht über die Anzahl der an jeden Benutzer gesendeten Benachrichtigungen angezeigt.

Über die Dropdown-Listen in der oberen rechten Ecke können Sie den Medientyp (oder alle), den Zeitraum (Daten für jeden Tag/jede Woche/jeden Monat/jedes Jahr) und das Jahr für die gesendeten Benachrichtigungen auswählen.



Jede Spalte zeigt die Gesamtsummen pro Systembenutzer an.

Datenerfassung

Übersicht

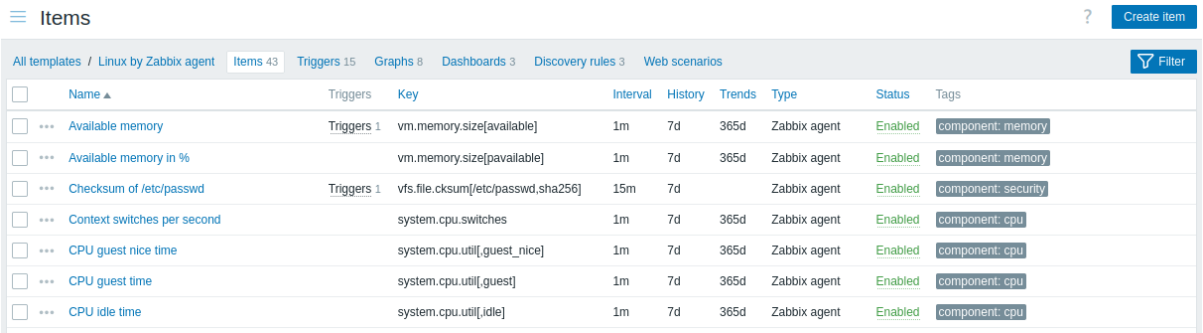
Dieses Menü enthält Abschnitte, die mit der Konfiguration der Datenerfassung zusammenhängen.

Datenpunkte

Übersicht

Die Datenpunktliste für eine Vorlage kann über *Datensammlung* > *Vorlagen* aufgerufen werden, indem Sie bei der entsprechenden Vorlage auf *Datenpunkte* klicken.

Eine Liste der vorhandenen Datenpunkte wird angezeigt.



Angezeigte Daten:

Spalte	Beschreibung
Datenpunktmenü	Klicken Sie auf das Drei-Punkte-Symbol, um das Datenpunktmenü zu öffnen.
Vorlage	Vorlage, zu der der Datenpunkt gehört. Durch Klicken auf den Vorlagennamen wird das Vorlagenkonfigurationsformular geöffnet. Diese Spalte wird nur angezeigt, wenn im Filter mehrere Vorlagen oder keine Vorlage ausgewählt sind.
Name	Name des Datenpunkts, der als blauer Link zu den Datenpunktdetails angezeigt wird. Durch Klicken auf den Link mit dem Datenpunktnamen wird das Datenpunktkonfigurationsformular geöffnet. Wenn der Datenpunkt von einer anderen Vorlage geerbt wurde, wird der Vorlagenname vor dem Datenpunktnamen als grauer Link angezeigt. Durch Klicken auf den Vorlagenlink wird die Datenpunktliste auf dieser Vorlagenebene geöffnet.
Auslöser	Wenn Sie den Mauszeiger über *Auslöser* bewegen, wird ein Infokasten mit den Auslösern angezeigt, die dem Datenpunkt zugeordnet sind. Die Anzahl der Auslöser wird in Grau angezeigt.
Schlüssel	Der Schlüssel des Datenpunkts wird angezeigt.
Intervall	Die Häufigkeit der Prüfung wird angezeigt.
Historie	Die Anzahl der Tage, für die die Datenpunkthistorie aufbewahrt wird, wird angezeigt.
Trends	Die Anzahl der Tage, für die die Trendhistorie des Datenpunkts aufbewahrt wird, wird angezeigt.
Typ	Der Datenpunkttyp wird angezeigt (Zabbix Agent, SNMP Agent, einfache Prüfung usw.).
Status	Der Status des Datenpunkts wird angezeigt – *Aktiviert* oder *Deaktiviert*. Durch Klicken auf den Status können Sie ihn ändern – von „Aktiviert“ zu „Deaktiviert“ und umgekehrt.
Tags	Die Tags des Datenpunkts werden angezeigt. Bis zu drei Tags (Name-Wert-Paare) können angezeigt werden. Wenn weitere Tags vorhanden sind, wird ein Link „...“ angezeigt, über den beim Bewegen des Mauszeigers alle Tags angezeigt werden können.

Um einen neuen Datenpunkt zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Datenpunkt erstellen*.

Optionen für die Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen für die Massenbearbeitung:

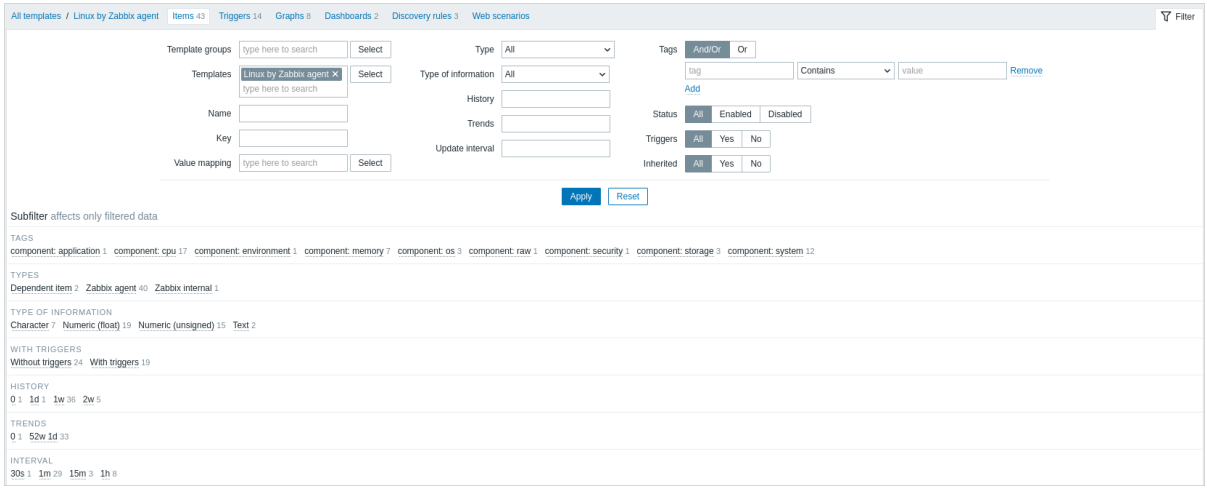
- *Enable* - den Status des Datenpunkts auf *Enabled* ändern.
- *Disable* - den Status des Datenpunkts auf *Disabled* ändern.
- *Copy* - die Datenpunkte auf andere Hosts oder Vorlagen kopieren.
- *Mass update* - mehrere Eigenschaften für eine Anzahl von Datenpunkten gleichzeitig aktualisieren.
- *Delete* - die Datenpunkte löschen.

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Datenpunkten und klicken Sie dann auf die gewünschte Schaltfläche.

Verwendung des Filters

Die Datenpunktliste kann viele Datenpunkte enthalten. Mithilfe des Filters können Sie einige davon herausfiltern, um die gesuchten Datenpunkte schnell zu finden. Für eine bessere Suchleistung werden die Daten mit nicht aufgelösten Makros durchsucht.

Das Symbol *Filter* ist oben rechts verfügbar. Wenn Sie darauf klicken, wird ein Filter geöffnet, in dem Sie die gewünschten Filterkriterien angeben können.

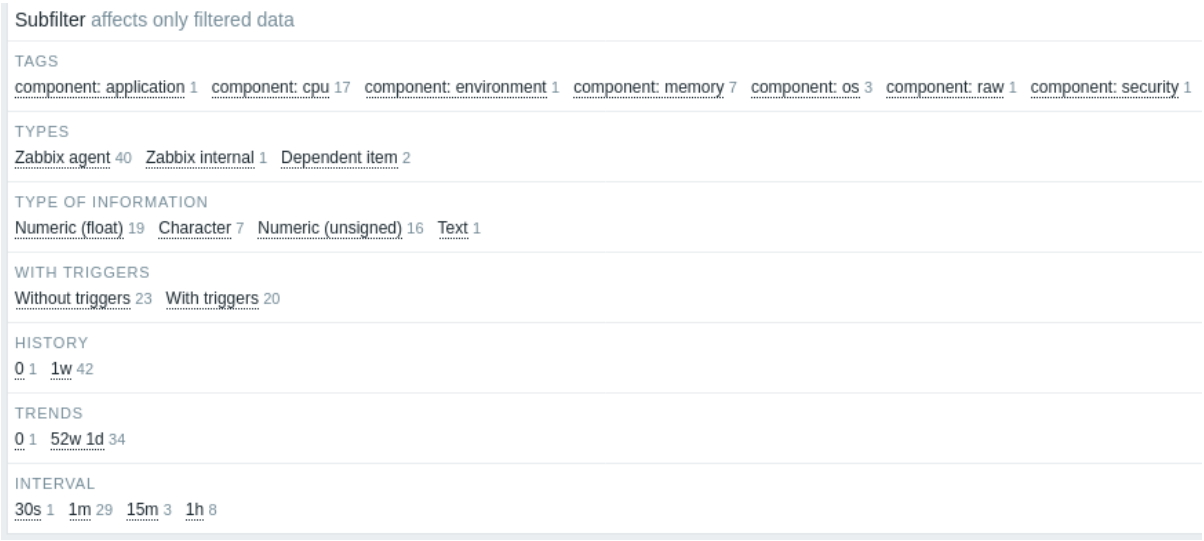


Parameter	Beschreibung
Template groups	Nach einem oder mehreren Template groups filtern. Die Angabe einer übergeordneten Template group wählt implizit alle verschachtelten Gruppen aus.
Templates	Nach einem oder mehreren Vorlagen filtern.
Name	Nach dem Namen des Datenpunkts filtern.
Key	Nach dem Schlüssel des Datenpunkts filtern.
Value mapping	Nach der verwendeten Wertzuordnung filtern. Dieser Parameter wird nicht angezeigt, wenn die Option *Templates* leer ist.
Type	Nach dem Typ des Datenpunkts filtern (Zabbix agent, SNMP agent usw.).
Type of information	Nach dem Informationstyp filtern (Unsigned numeric, float usw.).
History	Nach der Dauer filtern, für die der Verlauf des Datenpunkts aufbewahrt wird.
Trends	Nach der Dauer filtern, für die die Trends des Datenpunkts aufbewahrt werden.
Update interval	Nach dem Aktualisierungsintervall des Datenpunkts filtern.
Tags	Geben Sie Tags an, um die Anzahl der angezeigten Datenpunkte zu begrenzen. Es ist möglich, bestimmte Tags und Tag-Werte einzuschließen sowie auszuschließen. Es können mehrere Bedingungen festgelegt werden. Die Übereinstimmung des Tag-Namens ist immer groß-/kleinschreibungssensitiv. Für jede Bedingung stehen mehrere Operatoren zur Verfügung: **Exists** - die angegebenen Tag-Namen einschließen **Equals** - die angegebenen Tag-Namen und Werte einschließen (groß-/kleinschreibungssensitiv) **Contains** - die angegebenen Tag-Namen einschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv) **Does not exist** - die angegebenen Tag-Namen ausschließen **Does not equal** - die angegebenen Tag-Namen und Werte ausschließen (groß-/kleinschreibungssensitiv) **Does not contain** - die angegebenen Tag-Namen ausschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv) Für Bedingungen gibt es zwei Berechnungstypen: **And/Or** - alle Bedingungen müssen erfüllt sein, Bedingungen mit demselben Tag-Namen werden durch die Or-Bedingung gruppiert **Or** - es genügt, wenn eine Bedingung erfüllt ist
Status	Nach dem Status des Datenpunkts filtern - *Enabled* oder *Disabled*.
Triggers	Datenpunkte mit (oder ohne) Auslösern filtern.
Inherited	Nach Datenpunkten filtern, die von verknüpften Vorlagen geerbt wurden (oder nicht geerbt wurden).

Verwendung des Unterfilters

Der Unterfilter ermöglicht eine weitere Anpassung der durch den Hauptfilter angewendeten Filterung.

Er enthält anklickbare Links für den schnellen Zugriff auf verwandte Datenpunkte. Datenpunkte sind durch eine gemeinsame Entität miteinander verbunden – beispielsweise Tag, Datenpunkttyp, Aktualisierungsintervall des Datenpunkts usw. Wenn eine Entität angeklickt wird, wird sie mit einem grauen Hintergrund hervorgehoben und die Datenpunkte werden sofort gefiltert (im Hauptfilter muss nicht auf *Anwenden* geklickt werden). Durch das Anklicken einer weiteren Entität wird diese zu den gefilterten Ergebnissen hinzugefügt. Durch erneutes Anklicken der Entität wird die Filterung entfernt.



Unterfilter werden auf Grundlage der gefilterten Daten erstellt, die auf 1000 Datensätze begrenzt sind. Wenn Sie mehr Datensätze im Unterfilter anzeigen möchten, müssen Sie den Wert des Parameters *Limit für Such- und Filterergebnisse* erhöhen (unter *Administration* > *Allgemein* > *GUI*).

Im Gegensatz zum Hauptfilter wird der Unterfilter bei jeder Aktualisierungsanforderung der Tabelle aktualisiert, damit die verfügbaren Filteroptionen und deren Anzahl stets aktuell sind.

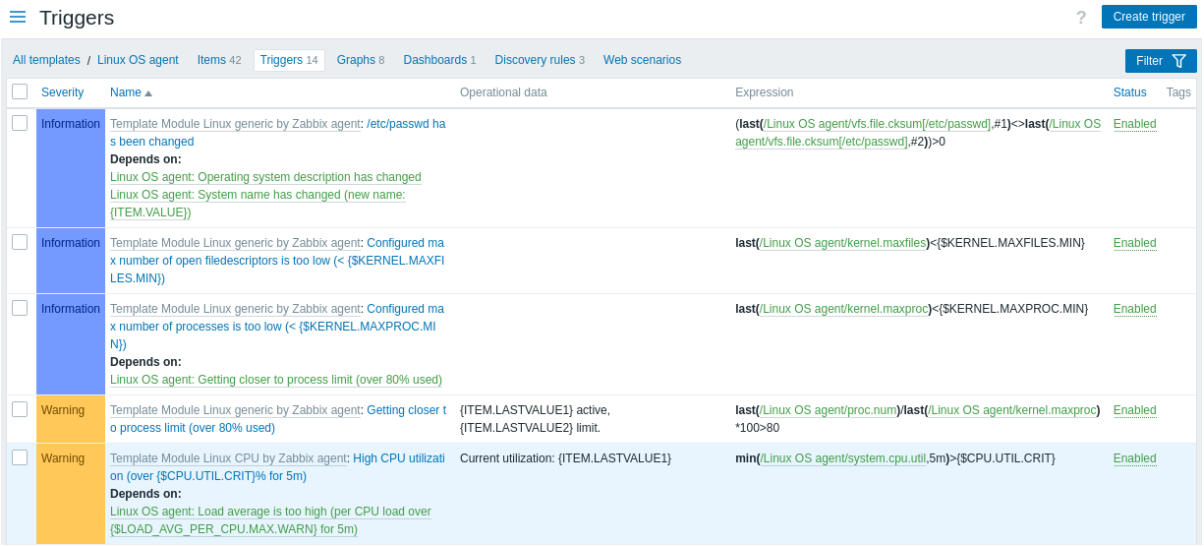
Die Anzahl der horizontal angezeigten Entitäten ist auf 100 begrenzt. Wenn es mehr gibt, wird am Ende ein Symbol mit drei Punkten angezeigt; es ist nicht anklickbar.

Eine Zahl neben jeder anklickbaren Entität gibt die Anzahl der darin gruppierten Datenpunkte an (basierend auf den Ergebnissen des Hauptfilters). Wenn eine Entität angeklickt wird, werden neben den anderen verfügbaren Entitäten Zahlen mit einem Pluszeichen angezeigt. Diese geben an, wie viele Datenpunkte zur aktuellen Auswahl hinzugefügt werden können. Entitäten ohne Datenpunkte werden nicht angezeigt, es sei denn, sie wurden zuvor im Unterfilter ausgewählt.

Auslöser

Übersicht

Die Auslöserliste für eine Vorlage kann unter *Datensammlung* > *Vorlagen* aufgerufen werden, indem Sie bei der entsprechenden Vorlage auf *Auslöser* klicken.



Angezeigte Daten:

Spalte	Beschreibung
Schweregrad	Der Schweregrad des Auslösers wird sowohl durch den Namen als auch durch die Hintergrundfarbe der Zelle angezeigt.

Spalte	Beschreibung
Vorlage	Vorlage, zu der der Auslöser gehört. Durch Klicken auf den Vorlagennamen wird das Vorlagenkonfigurationsformular geöffnet. Diese Spalte wird nur angezeigt, wenn im Filter mehrere Vorlagen oder keine Vorlage ausgewählt sind.
Name	Name des Auslösers, der als blauer Link zu den Auslöserdetails angezeigt wird. Durch Klicken auf den Link mit dem Auslösernamen wird das Auslöserkonfigurationsformular geöffnet. Wenn der Auslöser von einer anderen Vorlage geerbt wurde, wird der Vorlagenname vor dem Auslösernamen als grauer Link angezeigt. Durch Klicken auf den Vorlagenlink wird die Auslöserliste auf dieser Vorlagenebene geöffnet.
Betriebsdaten	Definition der Betriebsdaten des Auslösers, die beliebige Zeichenfolgen und Makros enthält, die unter *Überwachung* > *Probleme* dynamisch aufgelöst werden.
Ausdruck	Der Auslöserausdruck wird angezeigt. Der Teil des Ausdrucks, der sich auf den Datenpunkt der Vorlage bezieht, wird als Link angezeigt und führt zum Datenpunktkonfigurationsformular.
Status	Der Status des Auslösers wird angezeigt – *Aktiviert* oder *Deaktiviert*. Durch Klicken auf den Status können Sie ihn ändern – von „Aktiviert“ zu „Deaktiviert“ und umgekehrt.
Tags	Wenn ein Auslöser Tags enthält, werden Tagname und -wert in dieser Spalte angezeigt.

Um einen neuen Auslöser zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Auslöser erstellen*.

Optionen für die Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen für die Massenbearbeitung:

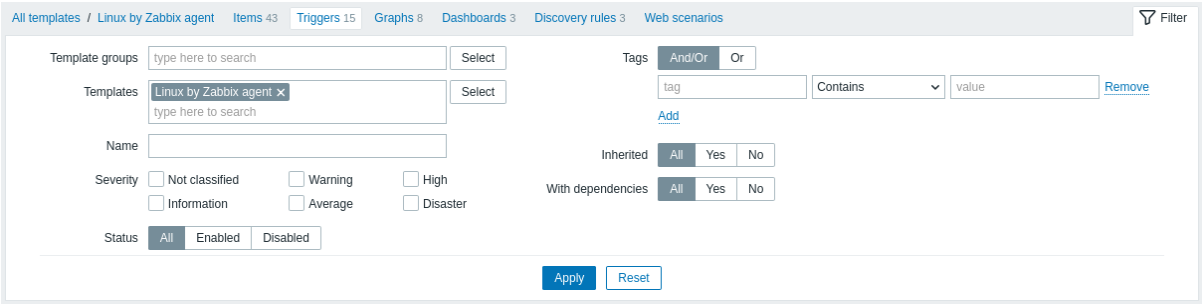
- *Enable* - den Auslöserstatus auf *Enabled* ändern
- *Disable* - den Auslöserstatus auf *Disabled* ändern
- *Copy* - die Auslöser auf andere Hosts oder Vorlagen kopieren
- *Mass update* - mehrere Eigenschaften für eine Anzahl von Auslösern gleichzeitig aktualisieren
- *Delete* - die Auslöser löschen

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Auslösern und klicken Sie dann auf die gewünschte Schaltfläche.

Verwendung des Filters

Sie können den Filter verwenden, um nur die Auslöser anzuzeigen, die Sie interessieren. Für eine bessere Suchleistung werden die Daten mit nicht aufgelösten Makros durchsucht.

Das Symbol *Filter* ist in der oberen rechten Ecke verfügbar. Wenn Sie darauf klicken, wird ein Filter geöffnet, in dem Sie die gewünschten Filterkriterien angeben können.



Parameter	Beschreibung
Vorlagegruppen	Nach einer oder mehreren Vorlagegruppen filtern. Wenn eine übergeordnete Vorlagegruppe angegeben wird, werden implizit alle verschachtelten Gruppen ausgewählt.
Vorlagen	Nach einer oder mehreren Vorlagen filtern. Wenn oben bereits Vorlagegruppen ausgewählt wurden, ist die Vorlagenauswahl auf diese Gruppen beschränkt.
Name	Nach dem Namen des Auslösers filtern.
Schweregrad	Auswählen, um nach einem oder mehreren Schweregraden von Auslösern zu filtern.
Status	Nach dem Aktivierungsstatus des Auslösers filtern (Enabled/Disabled).

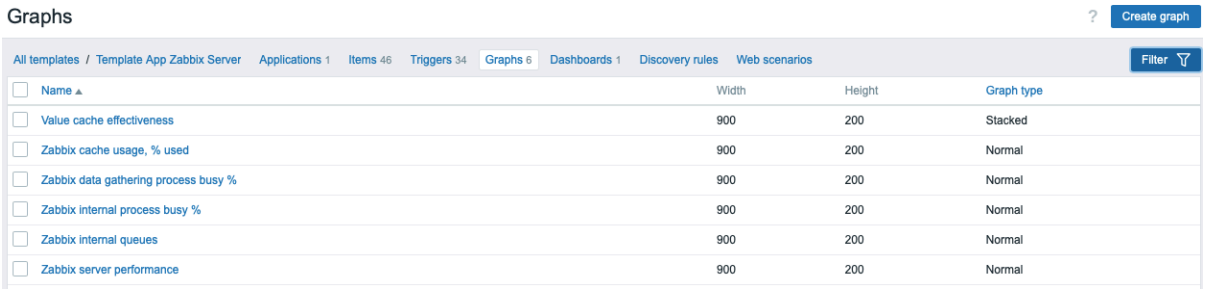
Parameter	Beschreibung
Tags	Nach Tag-Namen und -Wert des Auslösers filtern. Es ist möglich, bestimmte Tags und Tag-Werte einzuschließen oder auszuschließen. Es können mehrere Bedingungen festgelegt werden. Die Übereinstimmung des Tag-Namens ist immer groß-/kleinschreibungssensitiv. Für jede Bedingung stehen mehrere Operatoren zur Verfügung: **Exists** - die angegebenen Tag-Namen einschließen **Equals** - die angegebenen Tag-Namen und -Werte einschließen (groß-/kleinschreibungssensitiv) **Contains** - die angegebenen Tag-Namen einschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv) **Does not exist** - die angegebenen Tag-Namen ausschließen **Does not equal** - die angegebenen Tag-Namen und -Werte ausschließen (groß-/kleinschreibungssensitiv) **Does not contain** - die angegebenen Tag-Namen ausschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv) Es gibt zwei Berechnungstypen für Bedingungen: **And/Or** - alle Bedingungen müssen erfüllt sein, Bedingungen mit demselben Tag-Namen werden durch die Or-Bedingung gruppiert **Or** - es reicht aus, wenn eine Bedingung erfüllt ist Makros und Makrofunktionen werden in den Feldern für Tag-Namen und Tag-Wert unterstützt.
Geerbt	Auslöser filtern, die von verknüpften Vorlagen geerbt wurden (oder nicht geerbt wurden).
Mit Abhängigkeiten	Auslöser mit (oder ohne) Abhängigkeiten filtern.

Diagramme

Übersicht

Die Liste der benutzerdefinierten Graphen für eine Vorlage kann unter *Datensammlung* > *Vorlagen* aufgerufen werden, indem Sie bei der entsprechenden Vorlage auf *Graphen* klicken.

Eine Liste der vorhandenen Graphen wird angezeigt.



Angezeigte Daten:

Spalte	Beschreibung
Vorlage	Vorlage, zu der der Graph gehört. Durch Klicken auf den Namen der Vorlage wird das Konfigurationsformular für Vorlagen geöffnet. Diese Spalte wird nur angezeigt, wenn im Filter mehrere oder keine Vorlagen ausgewählt sind.
Name	Name des benutzerdefinierten Graphen, der als blauer Link zu den Graphendetails angezeigt wird. Durch Klicken auf den Link mit dem Graphennamen wird das Konfigurationsformular für Graphen geöffnet. Wenn der Graph von einer anderen Vorlage übernommen wurde, wird der Vorlagenname vor dem Graphennamen als grauer Link angezeigt. Durch Klicken auf den Vorlagenlink wird die Graphenliste auf dieser Vorlagenebene geöffnet.
Breite	Die Breite des Graphen wird angezeigt.
Höhe	Die Höhe des Graphen wird angezeigt.
Graphentyp	Der Graphentyp wird angezeigt – *Normal*, *Gestapelt*, *Kreis* oder *Explodiert*.

Um einen neuen Graphen zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Graph erstellen*.

Optionen für die Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen für die Massenbearbeitung:

- *Kopieren* - die Diagramme auf andere Hosts oder Vorlagen kopieren
- *Löschen* - die Diagramme löschen

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Diagrammen und klicken Sie dann auf die gewünschte Schaltfläche.

Filter verwenden

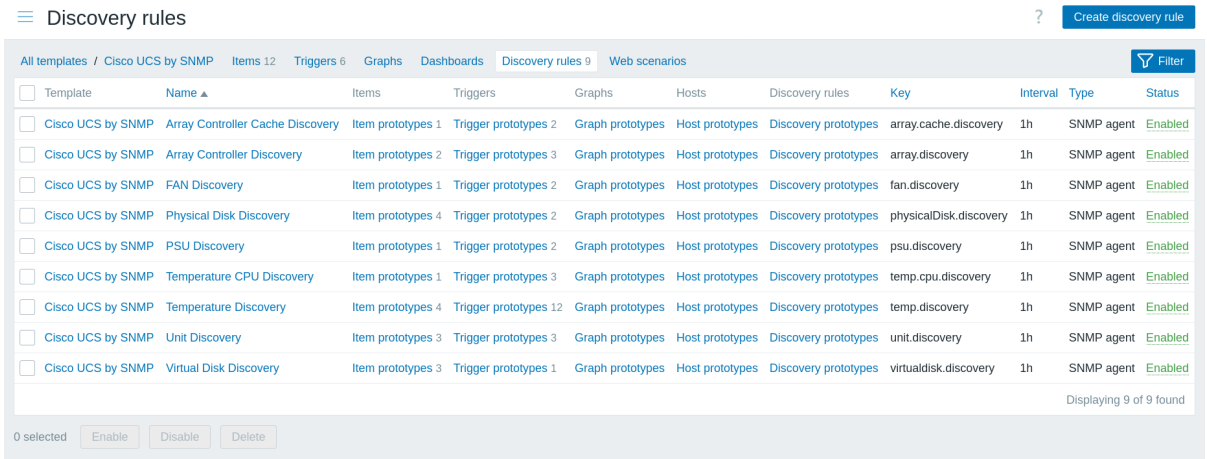
Sie können Graphen nach Vorlagengruppe und Vorlage filtern. Für eine bessere Suchleistung werden Daten mit nicht aufgelösten Makros durchsucht.

Erkennungsregeln

Übersicht

Die Liste der Low-Level-Discovery-Regeln für eine Vorlage kann über *Datensammlung* > *Vorlagen* aufgerufen werden, indem Sie bei der entsprechenden Vorlage auf *Discovery* klicken.

Eine Liste der vorhandenen Low-Level-Discovery-Regeln wird angezeigt. Sie können auch alle Discovery-Regeln unabhängig von der Vorlage oder alle Discovery-Regeln einer bestimmten Vorlagengruppe anzeigen, indem Sie die Filtereinstellungen ändern.



Angezeigte Daten:

Spalte	Beschreibung
Vorlage	Die Vorlage, zu der die Discovery-Regel gehört. Durch Klicken auf den Namen der Vorlage wird das Konfigurationsformular der Vorlage geöffnet.
Name	Name der Regel, der als blauer Link angezeigt wird. Durch Klicken auf den Namen der Regel wird das Konfigurationsformular der Low-Level-Discovery-Regel geöffnet. Wenn die Discovery-Regel von einer anderen Vorlage übernommen wurde, wird der Name der Vorlage vor dem Regelnamen als grauer Link angezeigt. Durch Klicken auf den Vorlagenlink wird die Liste der Discovery-Regeln auf dieser Vorlagenebene geöffnet.
Datenpunkte	Ein Link zur Liste der Datenpunkt-Prototypen wird angezeigt. Die Anzahl der vorhandenen Datenpunkt-Prototypen wird grau angezeigt.
Auslöser	Ein Link zur Liste der Auslöser-Prototypen wird angezeigt. Die Anzahl der vorhandenen Auslöser-Prototypen wird grau angezeigt.
Graphen	Ein Link zur Liste der Graph-Prototypen wird angezeigt. Die Anzahl der vorhandenen Graph-Prototypen wird grau angezeigt.
Hosts	Ein Link zur Liste der Host-Prototypen wird angezeigt. Die Anzahl der vorhandenen Host-Prototypen wird grau angezeigt.
Schlüssel	Der für die Discovery verwendete Datenpunktschlüssel wird angezeigt.
Intervall	Die Häufigkeit der Durchführung der Discovery wird angezeigt.
Typ	Der für die Discovery verwendete Datenpunkttyp wird angezeigt (Zabbix Agent, SNMP-Agent usw.).

Spalte	Beschreibung
Status	Der Status der Discovery-Regel wird angezeigt - *Aktiviert* oder *Deaktiviert*. Durch Klicken auf den Status können Sie ihn ändern - von „Aktiviert“ zu „Deaktiviert“ (und umgekehrt).

Um eine neue Low-Level-Discovery-Regel zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Discovery-Regel erstellen*.

Optionen für die Massenbearbeitung

Die Schaltflächen unter der Liste bieten einige Optionen für die Massenbearbeitung:

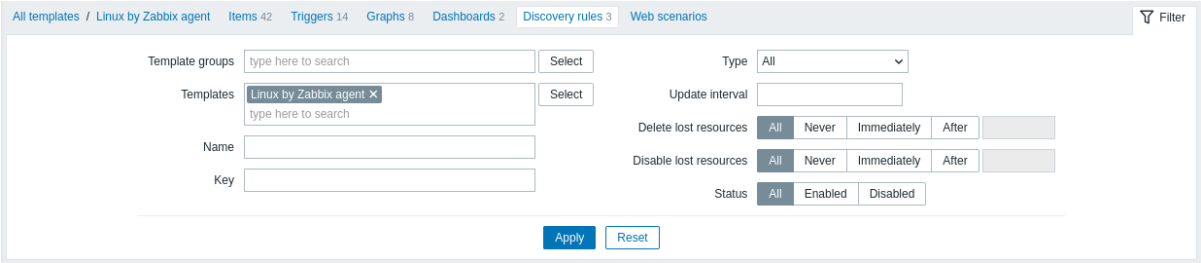
- *Aktivieren* - den Status der Low-Level-Discovery-Regel auf *Aktiviert* ändern
- *Deaktivieren* - den Status der Low-Level-Discovery-Regel auf *Deaktiviert* ändern
- *Löschen* - die Low-Level-Discovery-Regeln löschen

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Discovery-Regeln und klicken Sie dann auf die gewünschte Schaltfläche.

Verwendung des Filters

Sie können den Filter verwenden, um nur die Erkennungsregeln anzuzeigen, die Sie interessieren. Für eine bessere Suchleistung werden die Daten mit nicht aufgelösten Makros durchsucht.

Das Symbol *Filter* ist in der oberen rechten Ecke verfügbar. Wenn Sie darauf klicken, wird ein Filter geöffnet, in dem Sie die gewünschten Filterkriterien wie Vorlage, Name der Erkennungsregel, Datenpunktschlüssel, Datenpunkttyp usw. angeben können.



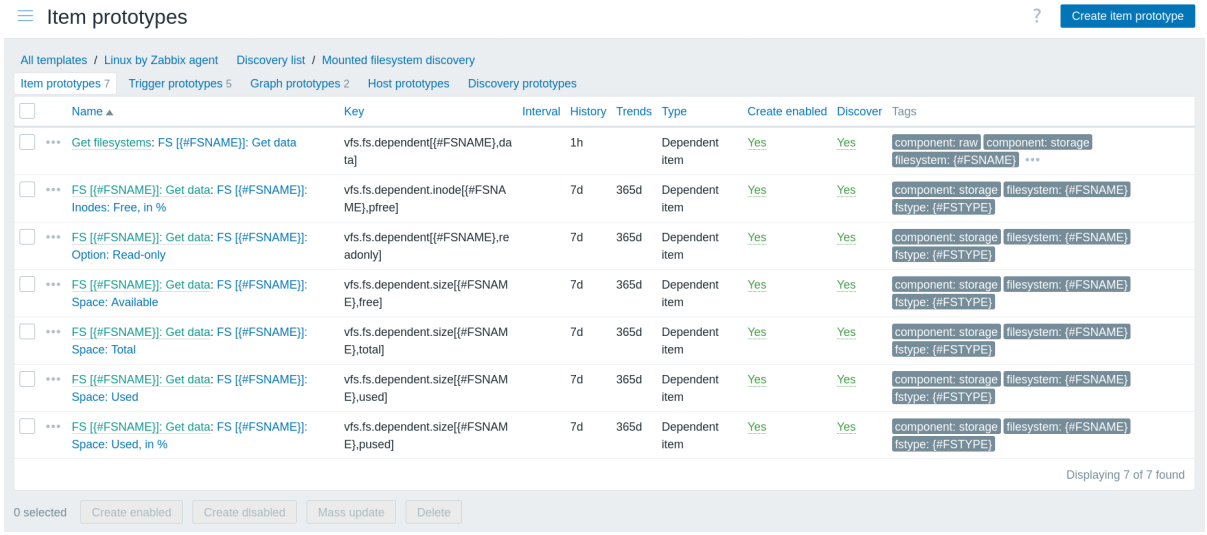
Parameter	Beschreibung
Template groups	Nach einem oder mehreren Vorlagengruppen filtern. Durch die Angabe einer übergeordneten Vorlagengruppe werden implizit alle untergeordneten Gruppen ausgewählt.
Templates	Nach einer oder mehreren Vorlagen filtern.
Name	Nach dem Namen der Erkennungsregel filtern.
Key	Nach dem Schlüssel des Erkennungsdatenpunkts filtern.
Type	Nach dem Typ des Erkennungsdatenpunkts filtern.
Update interval	Nach dem Aktualisierungsintervall filtern. Für Zabbix trapper und abhängige Datenpunkte nicht verfügbar.
Delete lost resources	Nach dem Zeitraum *Delete lost resources* filtern.
Disable lost resources	Nach dem Zeitraum *Disable lost resources* filtern.
Status	Nach dem Status der Erkennungsregel filtern (All/Enabled/Disabled).

Datenpunkt-Prototypen

Übersicht

In diesem Abschnitt werden die konfigurierten Datenpunkt-Prototypen einer Low-Level-Discovery-Regel auf der Vorlage angezeigt.

Wenn die Vorlage mit dem Host verknüpft ist, bilden die Datenpunkt-Prototypen die Grundlage für die Erstellung tatsächlicher Host-Datenpunkte während der Low-Level-Discovery.



Angezeigte Daten:

Column	Description
Name	Name des Datenpunkt-Prototyps, als blauer Link angezeigt. Durch Klicken auf den Namen wird das Konfigurationsformular für den Datenpunkt-Prototyp geöffnet. Wenn der Datenpunkt-Prototyp zu einer verknüpften Vorlage gehört, wird der Name der Vorlage vor dem Namen des Datenpunkt-Prototyps als grauer Link angezeigt. Durch Klicken auf den Vorlagenlink wird die Liste der Datenpunkt-Prototypen auf der Ebene der verknüpften Vorlage geöffnet. Wenn der Datenpunkt-Prototyp vom Typ „Abhängiger Datenpunkt“ ist, wird der Name des Master-Datenpunkts vor dem Namen des Datenpunkt-Prototyps grün angezeigt (wie bei Datenpunkt-Prototypen auf Host-Ebene).
Key	Der Schlüssel des Datenpunkt-Prototyps wird angezeigt.
Interval	Das Prüfintervall wird angezeigt.
History	Die Anzahl der Tage, für die der Datenpunktverlauf aufbewahrt wird, wird angezeigt.
Trends	Die Anzahl der Tage, für die der Trendverlauf des Datenpunkts aufbewahrt wird, wird angezeigt.
Type	Der Typ des Datenpunkt-Prototyps wird angezeigt (Zabbix agent, SNMP agent, einfache Prüfung usw.).
Create enabled	Erstellung des auf diesem Prototyp basierenden Datenpunkts als: **Yes** - aktiviert **No** - deaktiviert. Durch Klicken auf „Yes“ und „No“ können Sie zwischen den Optionen wechseln.
Discover	Erkennung des auf diesem Prototyp basierenden Datenpunkts: **Yes** - erkennen **No** - nicht erkennen. Durch Klicken auf „Yes“ und „No“ können Sie zwischen den Optionen wechseln.
Tags	Die Tags des Datenpunkt-Prototyps werden angezeigt.

Um einen neuen Datenpunkt-Prototyp zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Datenpunkt-Prototyp erstellen*.

Optionen für die Massenbearbeitung

Die Schaltflächen unter der Liste bieten einige Optionen für die Massenbearbeitung:

- *Aktiviert erstellen* - diese Datenpunkte als *Aktiviert* erstellen
- *Deaktiviert erstellen* - diese Datenpunkte als *Deaktiviert* erstellen
- *Massenaktualisierung* - diese Datenpunktvorlagen massenhaft aktualisieren
- *Löschen* - diese Datenpunktvorlagen löschen

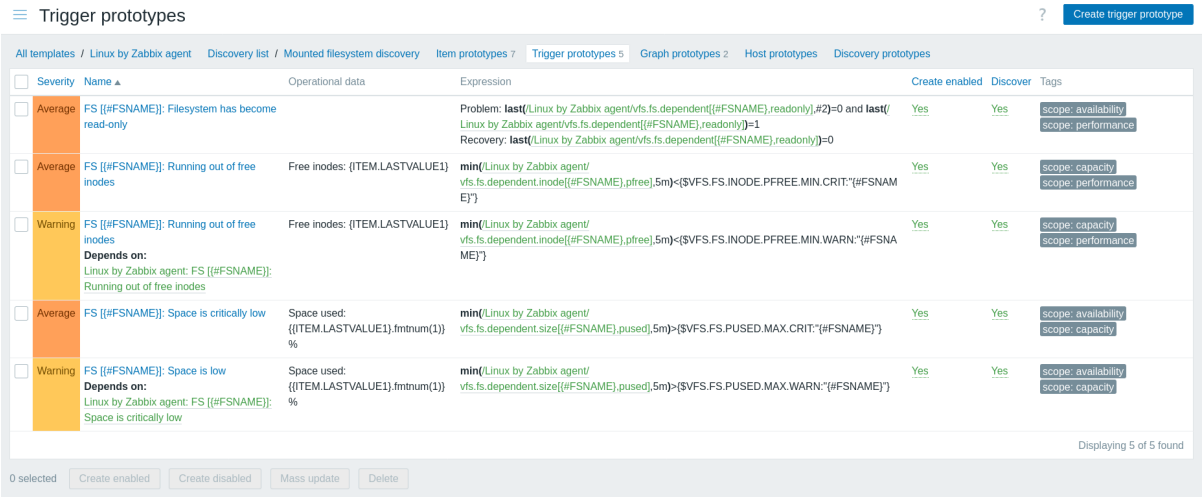
Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Datenpunktvorlagen und klicken Sie dann auf die gewünschte Schaltfläche.

Auslöser-Prototypen

Übersicht

In diesem Abschnitt werden die konfigurierten Auslöser-Prototypen einer Low-Level-Discovery-Regel auf der Vorlage angezeigt.

Wenn die Vorlage mit dem Host verknüpft ist, bilden die Auslöser-Prototypen die Grundlage für die Erstellung echter Auslöser auf dem Host während der Low-Level-Discovery.



Angezeigte Daten:

Column	Description
Severity	Der Schweregrad des Auslösers wird sowohl durch den Namen als auch durch die Hintergrundfarbe der Zelle angezeigt.
Name	Der Name des Auslöser-Prototyps wird als blauer Link angezeigt. Durch Klicken auf den Namen wird das Konfigurationsformular für Auslöser-Prototypen geöffnet. Wenn der Auslöser-Prototyp zu einer verknüpften Vorlage gehört, wird der Name der Vorlage vor dem Namen des Auslösers als grauer Link angezeigt. Durch Klicken auf den Vorlagen-Link wird die Liste der Auslöser-Prototypen auf der Ebene der verknüpften Vorlage geöffnet.
Expression	Der Ausdruck des Auslösers wird angezeigt. Der Teil des Ausdrucks, der sich auf den Datenpunkt der Vorlage bezieht, wird als Link angezeigt und führt zum Konfigurationsformular für Datenpunkte.
Operational data	Das Format der Betriebsdaten des Auslösers wird angezeigt. Es kann beliebige Zeichenfolgen und Makros enthalten, die unter *Monitoring > Problems* dynamisch aufgelöst werden.
Create enabled	Erstellung des Auslösers auf Grundlage dieses Prototyps als: **Yes** – aktiviert **No** – deaktiviert. Durch Klicken auf „Yes“ oder „No“ können Sie zwischen den Optionen wechseln.
Discover	Erkennung des Auslösers auf Grundlage dieses Prototyps: **Yes** – erkennen **No** – nicht erkennen. Durch Klicken auf „Yes“ oder „No“ können Sie zwischen den Optionen wechseln.
Tags	Die Tags des Auslöser-Prototyps werden angezeigt.

Um einen neuen Auslöser-Prototyp zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Create trigger prototype*.

Optionen für die Massenbearbeitung

Die Schaltflächen unter der Liste bieten einige Optionen für die Massenbearbeitung:

- *Aktiviert erstellen* - diese Auslöser als *Aktiviert* erstellen
- *Deaktiviert erstellen* - diese Auslöser als *Deaktiviert* erstellen
- *Massenaktualisierung* - diese Auslöserprototypen massenweise aktualisieren
- *Löschen* - diese Auslöserprototypen löschen

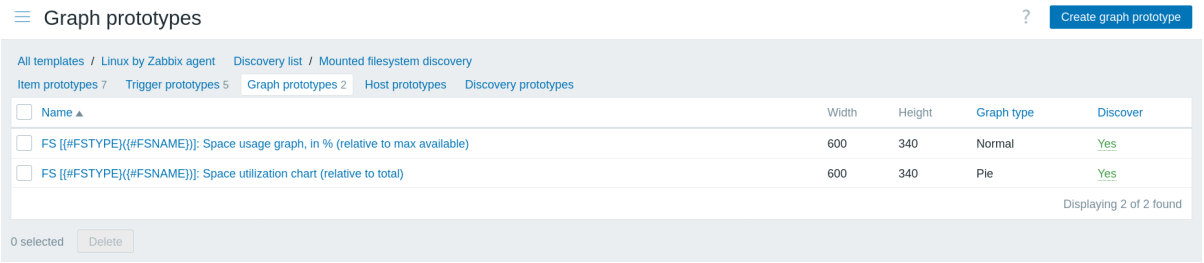
Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Auslöserprototypen und klicken Sie dann auf die gewünschte Schaltfläche.

Diagrammvorlagen

Übersicht

In diesem Abschnitt werden die konfigurierten Diagrammprototypen einer Low-Level-Discovery-Regel auf der Vorlage angezeigt.

Wenn die Vorlage mit dem Host verknüpft ist, bilden die Diagrammprototypen die Grundlage für die Erstellung echter Diagramme des Hosts während der Low-Level-Discovery.



Angezeigte Daten:

Spalte	Beschreibung
Name	Name des Diagrammprototyps, angezeigt als blauer Link. Durch Klicken auf den Namen wird das Konfigurationsformular für den Diagrammprototyp geöffnet. Wenn der Diagrammprototyp zu einer verknüpften Vorlage gehört, wird der Name der Vorlage vor dem Diagrammnamen als grauer Link angezeigt. Durch Klicken auf den Vorlagenlink wird die Liste der Diagrammprototypen auf der Ebene der verknüpften Vorlage geöffnet.
Breite	Die Breite des Diagrammprototyps wird angezeigt.
Höhe	Die Höhe des Diagrammprototyps wird angezeigt.
Typ	Der Typ des Diagrammprototyps wird angezeigt – *Normal*, *Gestapelt*, *Kreis* oder *Explodiert*.
Ermitteln	Ermittelt das Diagramm auf Grundlage dieses Prototyps: **Ja** – ermitteln **Nein** – nicht ermitteln. Durch Klicken auf *Ja* oder *Nein* können Sie zwischen den Optionen wechseln.

Um einen neuen Diagrammprototyp zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Diagrammprototyp erstellen*.

Optionen für die Massenbearbeitung

Die Schaltflächen unter der Liste bieten einige Optionen für die Massenbearbeitung:

- *Löschen* - diese Graph-Prototypen löschen

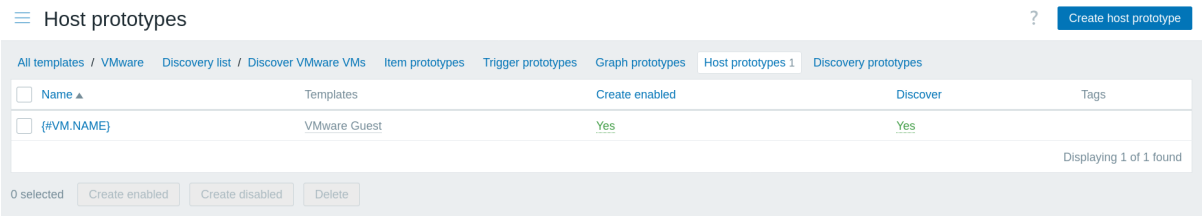
Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Graph-Prototypen und klicken Sie dann auf die gewünschte Schaltfläche.

Hostvorlagen

Übersicht

In diesem Abschnitt werden die konfigurierten Host-Prototypen einer Low-Level-Discovery-Regel auf der Vorlage angezeigt.

Wenn die Vorlage mit dem Host verknüpft ist, bilden die Host-Prototypen die Grundlage für die Erstellung echter Hosts während der Low-Level-Discovery.



Angezeigte Daten:

Spalte	Beschreibung
Name	Name des Host-Prototyps, als blauer Link angezeigt. Durch Klicken auf den Namen wird das Konfigurationsformular des Host-Prototyps geöffnet. Wenn der Host-Prototyp zu einer verknüpften Vorlage gehört, wird der Vorlagenname vor dem Hostnamen als grauer Link angezeigt. Durch Klicken auf den Vorlagenlink wird die Liste der Host-Prototypen auf der Ebene der verknüpften Vorlage geöffnet.
Vorlagen	Die Vorlagen des Host-Prototyps werden angezeigt.
Erstellung aktiviert	Den Host basierend auf diesem Prototyp erstellen als: **Ja** – aktiviert **Nein** – deaktiviert. Sie können durch Klicken zwischen „Ja“ und „Nein“ wechseln.
Ermitteln	Den Host basierend auf diesem Prototyp ermitteln: **Ja** – ermitteln **Nein** – nicht ermitteln. Sie können durch Klicken zwischen „Ja“ und „Nein“ wechseln.
Tags	Die Tags des Host-Prototyps werden angezeigt.

Um einen neuen Host-Prototyp zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Host-Prototyp erstellen*.

Optionen für die Massenbearbeitung

Die Schaltflächen unter der Liste bieten einige Optionen für die Massenbearbeitung:

- *Aktiviert erstellen* - diese Hosts als *Aktiviert* erstellen
- *Deaktiviert erstellen* - diese Hosts als *Deaktiviert* erstellen
- *Löschen* - diese Host-Prototypen löschen

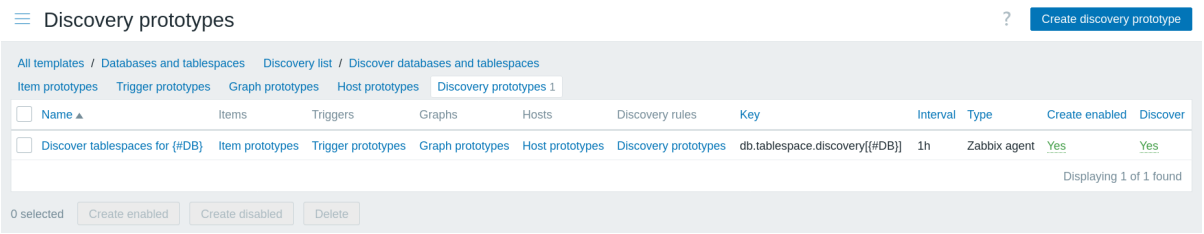
Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Host-Prototypen und klicken Sie dann auf die gewünschte Schaltfläche.

Discovery-Prototypen

Übersicht

In diesem Abschnitt werden die konfigurierten Discovery-Prototypen einer Low-Level-Discovery-Regel auf der Vorlage angezeigt.

Wenn die Vorlage mit dem Host verknüpft ist, bilden die Discovery-Prototypen die Grundlage für die Erstellung echter Discovery-Prototypen der übergeordneten Discovery-Regel.



Angezeigte Daten:

Column	Description
Name	Name des Discovery-Prototyps, als blauer Link angezeigt. Durch Klicken auf den Namen wird das Konfigurationsformular des Discovery-Prototyps geöffnet. Wenn der Discovery-Prototyp zu einer verknüpften Vorlage gehört, wird der Name der Vorlage vor dem Namen des Discovery-Prototyps als grauer Link angezeigt. Durch Klicken auf den Vorlagenlink wird die Liste der Discovery-Prototypen auf der Ebene der verknüpften Vorlage geöffnet.
Items	Ein Link zur Liste der Datenpunkt-Prototypen wird angezeigt. Die Anzahl der vorhandenen Datenpunkt-Prototypen wird grau angezeigt.
Triggers	Ein Link zur Liste der Auslöser-Prototypen wird angezeigt. Die Anzahl der vorhandenen Auslöser-Prototypen wird grau angezeigt.
Graphs	Ein Link zur Liste der Graph-Prototypen wird angezeigt. Die Anzahl der vorhandenen Graph-Prototypen wird grau angezeigt.
Hosts	Ein Link zur Liste der Host-Prototypen wird angezeigt. Die Anzahl der vorhandenen Host-Prototypen wird grau angezeigt.

Column	Description
Discovery rules	Ein Link zur Liste der Discovery-Prototypen wird angezeigt. Die Anzahl der vorhandenen Discovery-Prototypen wird grau angezeigt.
Key	Der für die Discovery verwendete Datenpunktschlüssel wird angezeigt.
Interval	Die Häufigkeit der Durchführung der Discovery wird angezeigt.
Type	Der für die Discovery verwendete Datenpunkttyp wird angezeigt (Zabbix Agent, SNMP Agent usw.).
Create enabled	Der Status der Erstellung des Discovery-Prototyps wird angezeigt – Erstellung aktiviert (*Yes*) oder Erstellung deaktiviert (*No*). Durch Klicken auf den Status können Sie ihn ändern – von Yes zu No (und umgekehrt).
Discover	Der Discovery-Status des Discovery-Prototyps wird angezeigt – Discovery durchführen (*Yes*) oder keine Discovery durchführen (*No*). Durch Klicken auf den Status können Sie ihn ändern – von Yes zu No (und umgekehrt).

Um einen neuen Discovery-Prototyp zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Create discovery prototype*.

Optionen für die Massenbearbeitung

Die Schaltflächen unter der Liste bieten einige Optionen für die Massenbearbeitung:

- *Aktiviert erstellen* - diese Erkennungsprototypen als *Aktiviert* erstellen
- *Deaktiviert erstellen* - diese Erkennungsprototypen als *Deaktiviert* erstellen
- *Löschen* - diese Erkennungsprototypen löschen

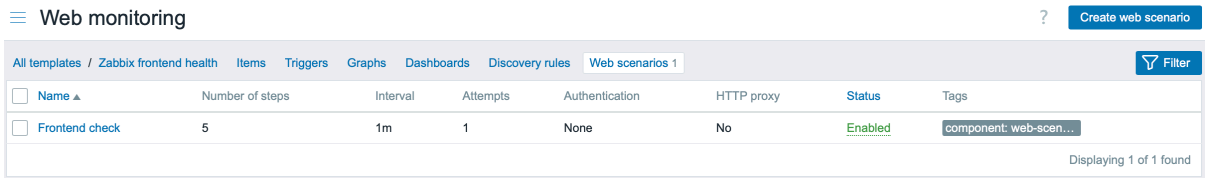
Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Erkennungsprototypen und klicken Sie dann auf die gewünschte Schaltfläche.

Web-Szenarien

Übersicht

Die Liste der Web-Szenarien für eine Vorlage kann unter *Datensammlung* > *Vorlagen* aufgerufen werden, indem Sie bei der entsprechenden Vorlage auf *Web* klicken.

Eine Liste der vorhandenen Web-Szenarien wird angezeigt.



Angezeigte Daten:

Spalte	Beschreibung
Name	Name des Web-Szenarios. Durch Klicken auf den Namen des Web-Szenarios wird das Konfigurationsformular für Web-Szenarien geöffnet. Wenn das Web-Szenario von einer anderen Vorlage übernommen wurde, wird der Vorlagenname vor dem Namen des Web-Szenarios als grauer Link angezeigt. Durch Klicken auf den Vorlagenlink wird die Liste der Web-Szenarien auf dieser Vorlagenebene geöffnet.
Anzahl der Schritte	Die Anzahl der Schritte, die das Szenario enthält.
Aktualisierungsintervall	Wie häufig das Szenario ausgeführt wird.
Versuche	Wie viele Versuche zur Ausführung der Schritte des Web-Szenarios unternommen werden.
Authentifizierung	Die Authentifizierungsmethode wird angezeigt: Basic, NTLM, Kerberos, Digest oder Keine.
HTTP-Proxy	Zeigt den HTTP-Proxy oder „Nein“ an, wenn kein Proxy verwendet wird.
Status	Der Status des Web-Szenarios wird angezeigt: *Aktiviert* oder *Deaktiviert*. Durch Klicken auf den Status können Sie ihn ändern.
Tags	Die Tags des Web-Szenarios werden angezeigt. Bis zu drei Tags (Name-Wert-Paare) können angezeigt werden. Wenn weitere Tags vorhanden sind, wird ein Link „...“ angezeigt, über den beim Überfahren mit der Maus alle Tags angezeigt werden können.

Um ein neues Web-Szenario zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Web-Szenario erstellen*.

Optionen für die Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen für die Massenbearbeitung:

- *Aktivieren* - den Status des Szenarios auf *Aktiviert* ändern
- *Deaktivieren* - den Status des Szenarios auf *Deaktiviert* ändern
- *Löschen* - die Webszenarien löschen

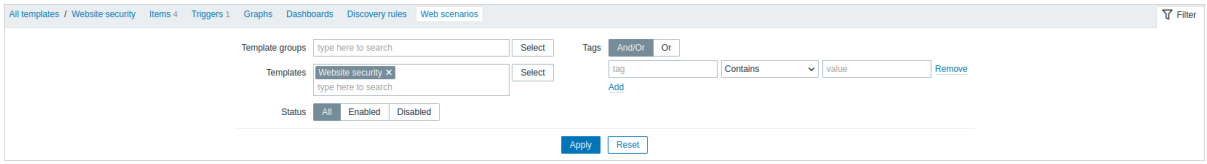
Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Webszenarien und klicken Sie dann auf die gewünschte Schaltfläche.

Filter verwenden

Sie können den Filter verwenden, um nur die Szenarien anzuzeigen, die Sie interessieren.

Für eine bessere Suchleistung werden die Daten mit nicht aufgelösten Makros durchsucht.

Der Link *Filter* ist oberhalb der Liste der Webszenarien verfügbar. Wenn Sie darauf klicken, wird ein Filter angezeigt, mit dem Sie Szenarien nach Vorlagengruppe, Vorlage, Status und Tags filtern können.

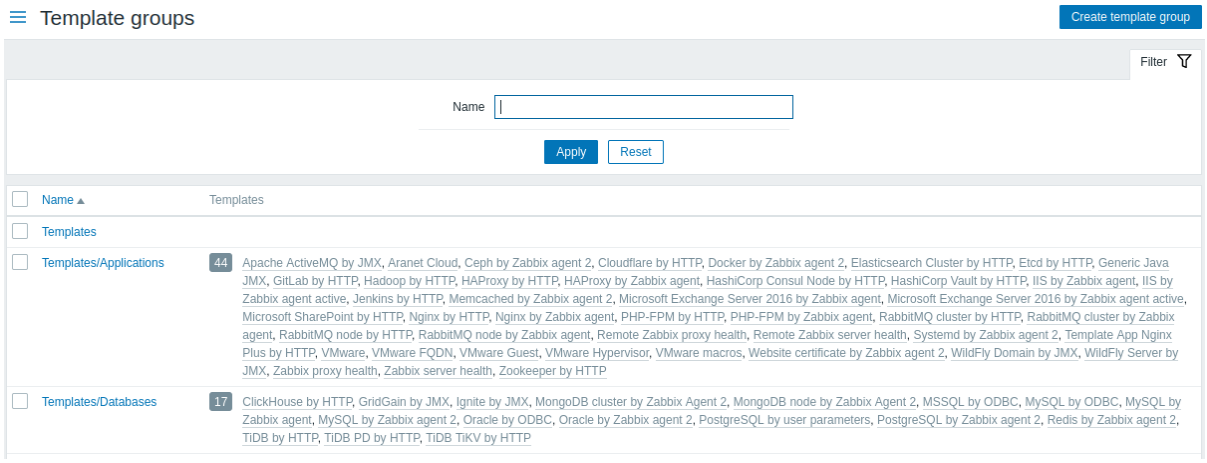


Vorlagengruppen

Übersicht

Im Bereich *Datensammlung* > *Vorlagengruppen* können Benutzer Vorlagengruppen konfigurieren und verwalten.

Eine Liste der vorhandenen Vorlagengruppen mit ihren Details wird angezeigt. Sie können Vorlagengruppen nach Namen suchen und filtern.



Angezeigte Daten:

Spalte	Beschreibung
Name	Name der Vorlagengruppe. Durch Klicken auf den Namen der Gruppe wird das Konfigurationsformular für Vorlagengruppen geöffnet.
Vorlagen	Anzahl der Vorlagen in der Gruppe (grau angezeigt), gefolgt von der Liste der Gruppenmitglieder. Durch Klicken auf den Namen einer Vorlage wird das Konfigurationsformular für Vorlagen geöffnet. Durch Klicken auf die Anzahl wird die Liste der Vorlagen in dieser Gruppe geöffnet.

Optionen für die Massenbearbeitung

Um mehrere Vorlagengruppen auf einmal zu löschen, markieren Sie die Kontrollkästchen vor den jeweiligen Gruppen und klicken Sie dann auf die Schaltfläche Löschen unterhalb der Liste.

Filter verwenden

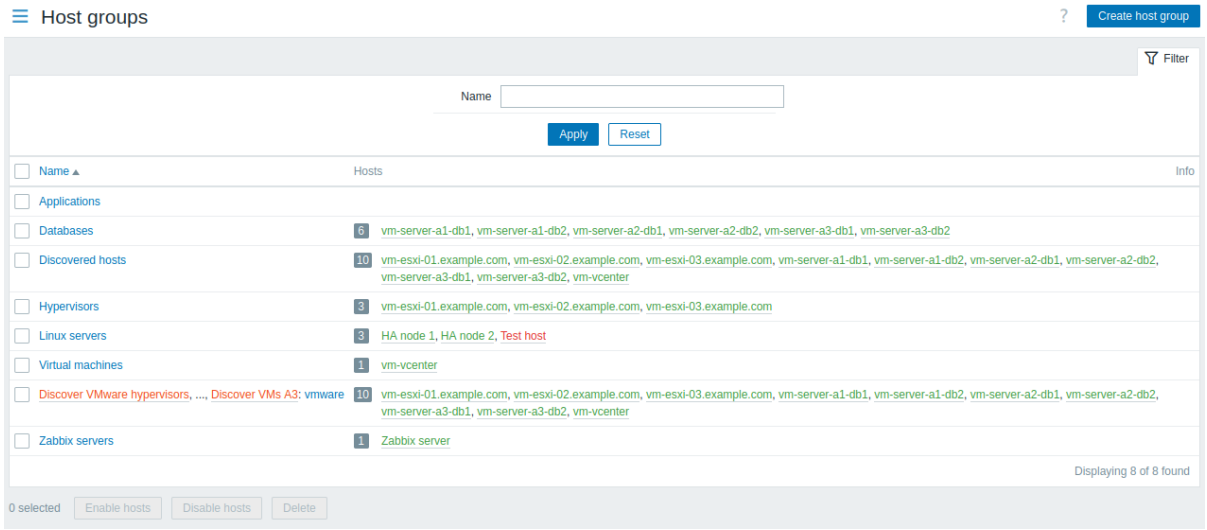
Sie können den Filter verwenden, um nur die Vorlagengruppen anzuzeigen, die Sie interessieren. Für eine bessere Suchleistung werden die Daten mit nicht aufgelösten Makros durchsucht.

Hostgruppen

Übersicht

Im Bereich *Datenerfassung* > *Host-Gruppen* können Benutzer Host-Gruppen konfigurieren und verwalten.

Eine Liste der vorhandenen Host-Gruppen mit ihren Details wird angezeigt. Sie können Host-Gruppen nach Namen suchen und filtern.



Angezeigte Daten:

Spalte	Beschreibung
Name	Name der Host-Gruppe. Wenn Sie auf den Namen der Gruppe klicken, wird das Konfigurationsformular für Host-Gruppen geöffnet. Ermittelte Host-Gruppen werden mit den Namen der Low-Level-Discovery-Regeln als Präfix angezeigt. Wenn Sie auf den Namen der LLD-Regel klicken, wird das Konfigurationsformular für Host-Prototypen geöffnet. Beachten Sie, dass ermittelte Host-Gruppen gelöscht werden, wenn alle LLD-Regeln, durch die sie ermittelt wurden, gelöscht werden.
Hosts	Anzahl der Hosts in der Gruppe (grau angezeigt), gefolgt von der Liste der Gruppenmitglieder. Wenn Sie auf den Namen eines Hosts klicken, wird das Konfigurationsformular für Hosts geöffnet. Wenn Sie auf die Anzahl klicken, werden in der gesamten Host-Liste diejenigen herausgefiltert, die zu dieser Gruppe gehören.
Info	Fehlerinformationen zur Host-Gruppe (falls vorhanden) werden angezeigt.

Optionen für die Massenbearbeitung

Die Schaltflächen unter der Liste bieten einige Optionen für die Massenbearbeitung:

- *Hosts aktivieren* - den Status aller Hosts in der Gruppe auf ”Überwacht” ändern
- *Hosts deaktivieren* - den Status aller Hosts in der Gruppe auf ”Nicht überwacht” ändern
- *Löschen* - die Hostgruppen löschen

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Hostgruppen und klicken Sie dann auf die gewünschte Schaltfläche.

Filter verwenden

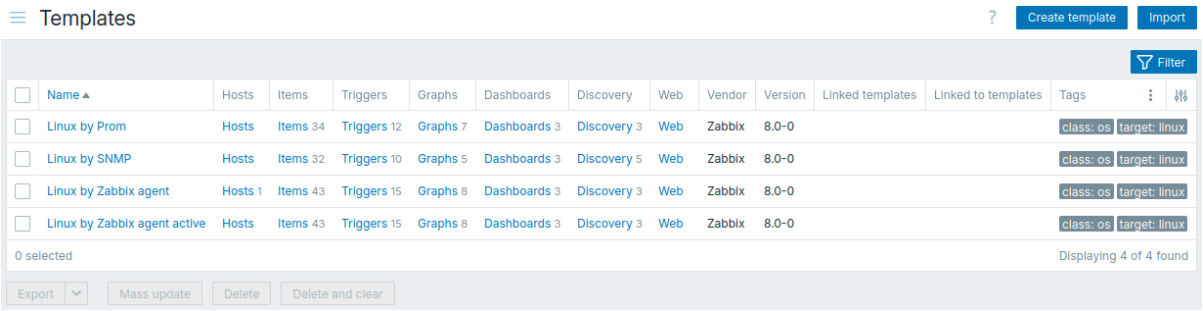
Sie können den Filter verwenden, um nur die Hostgruppen anzuzeigen, an denen Sie interessiert sind. Für eine bessere Suchleistung werden die Daten mit nicht aufgelösten Makros durchsucht.

Vorlagen

Übersicht

Im Abschnitt *Datensammlung* > *Vorlagen* können Benutzer Vorlagen konfigurieren und verwalten.

Eine Liste der vorhandenen Vorlagen mit ihren Details wird angezeigt.



Die Vorlagendetails werden in einer anpassbaren Tabelle angezeigt. Mit der Schaltfläche *Tabelle anpassen* in der Tabellenkopfzeile können Sie Spalten ein- oder ausblenden, ihre Reihenfolge ändern und das Layout auf die Standardeinstellungen zurücksetzen. Sie können außerdem die Größe jeder Spalte ändern, indem Sie ihren rechten Rand ziehen; durch Doppelklicken auf den Rand wird die ursprüngliche Größe wiederhergestellt. Alle Anpassungen werden sofort angewendet und für den aktuellen Benutzer dauerhaft gespeichert.

Die folgenden Spalten sind verfügbar:

Spalte	Beschreibung
Name	Name der Vorlage. Durch Klicken auf den Vorlagennamen wird das Konfigurationsformular der Vorlage geöffnet.
Hosts	Anzahl der bearbeitbaren Hosts, mit denen die Vorlage verknüpft ist; schreibgeschützte Hosts werden nicht berücksichtigt. Durch Klicken auf *Hosts* wird die Host-Liste geöffnet, wobei nur die Hosts gefiltert werden, die mit der Vorlage verknüpft sind.
Entitäten (Datenpunkte, Auslöser, Graphen, Dashboards, Discovery, Web)	Anzahl der jeweiligen Entitäten in der Vorlage (grau angezeigt). Durch Klicken auf den Namen der Entität werden in der vollständigen Liste dieser Entität diejenigen herausgefiltert, die zur Vorlage gehören.
Verknüpfte Vorlagen	Vorlagen, die mit der Vorlage verknüpft sind.
Verknüpft mit Vorlagen	Vorlagen, mit denen die Vorlage verknüpft ist.
Hersteller, *Version*	Hersteller und Version der Vorlage; werden angezeigt, wenn die Vorlagenkonfiguration diese Informationen enthält, und nur für vorkonfigurierte Vorlagen, importierte Vorlagen oder Vorlagen, die über die Template API geändert wurden. Bei vorkonfigurierten Vorlagen wird die Version wie folgt angezeigt: Hauptversion von Zabbix, Trennzeichen ("-"), Revisionsnummer (wird mit jeder neuen Version der Vorlage erhöht und mit jeder Hauptversion von Zabbix zurückgesetzt). Beispiele: 7.0-0, 7.0-3, 8.0-0, 8.0-3.
Tags	Vorlagen-Tags werden mit nicht aufgelösten Makros angezeigt. Wenn Sie den Mauszeiger über das Symbol ••• mit den drei Punkten nach den angezeigten Tags bewegen oder darauf klicken, wird ein Pop-up mit allen Tags geöffnet. Durch Klicken auf die Schaltfläche *Optionen* in der Spaltenkopfzeile können Sie die Anzahl der angezeigten Tags (1–3) festlegen, auswählen, wie Tag-Namen angezeigt werden (vollständig, gekürzt oder gar nicht), die Anzeigepriorität der Tags festlegen oder die Spalte duplizieren (duplizierte Spalten können umbenannt oder gelöscht werden).
Tag-Wert	Werte der Vorlagen-Tags werden mit nicht aufgelösten Makros angezeigt. Durch Klicken auf die Schaltfläche *Optionen* in der Spaltenkopfzeile können Sie die Spalte umbenennen, auswählen, welche Tag-Werte angezeigt werden, oder die Spalte duplizieren (duplizierte Spalten können umbenannt oder gelöscht werden).
Benutzerdefinierter Text	Benutzerdefinierter Text, Makrowerte oder eine Kombination aus beidem werden angezeigt. Durch Klicken auf die Schaltfläche *Optionen* in der Spaltenkopfzeile können Sie die Spalte umbenennen, den benutzerdefinierten Text oder Makros eingeben oder die Spalte duplizieren (duplizierte Spalten können umbenannt oder gelöscht werden). Unterstützte Makros: `{HOST.ID}`, `{HOST.HOST}`, `{HOST.NAME}`, `{HOST.DESCRIPTION}` und Benutzermakros.

Um eine neue Vorlage zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Vorlage erstellen*.

Um eine Vorlage aus einer YAML-, XML- oder JSON-Datei zu importieren, klicken Sie oben rechts auf die Schaltfläche *Importieren*.

Optionen für die Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen für die Massenbearbeitung:

- *Export* - die Vorlage in eine YAML-, XML- oder JSON-Datei exportieren;
- *Massenaktualisierung* - mehrere Eigenschaften aktualisieren für mehrere Vorlagen gleichzeitig;
- *Löschen* - die Vorlage löschen, wobei die verknüpften Entitäten (Datenpunkte, Auslöser usw.) bei den Hosts verbleiben;
- *Löschen und bereinigen* - die Vorlage und ihre verknüpften Entitäten von den Hosts löschen.

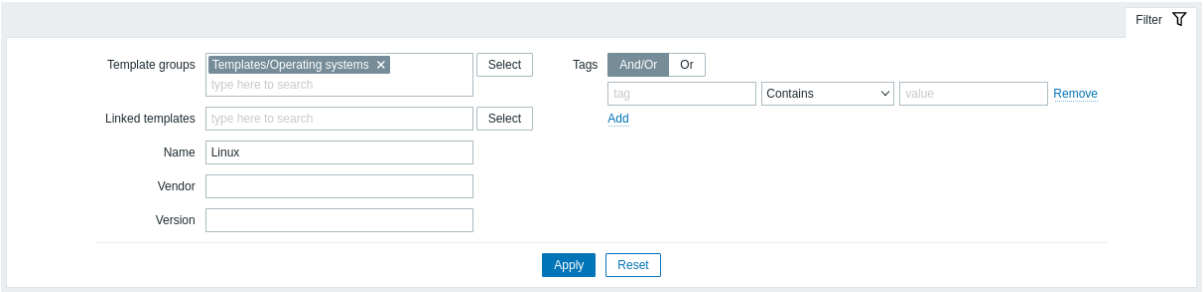
Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Vorlagen und klicken Sie dann auf die gewünschte Schaltfläche.

Verwendung des Filters

Sie können den Filter verwenden, um nur die Vorlagen anzuzeigen, die Sie interessieren.

Für eine bessere Suchleistung werden die Daten mit nicht aufgelösten Makros durchsucht.

Der Link *Filter* ist unter den Schaltflächen *Vorlage erstellen* und *Import* verfügbar. Wenn Sie darauf klicken, wird ein Filter verfügbar, mit dem Sie Vorlagen nach Vorlagengruppe, verknüpften Vorlagen, Name und Tags filtern können.



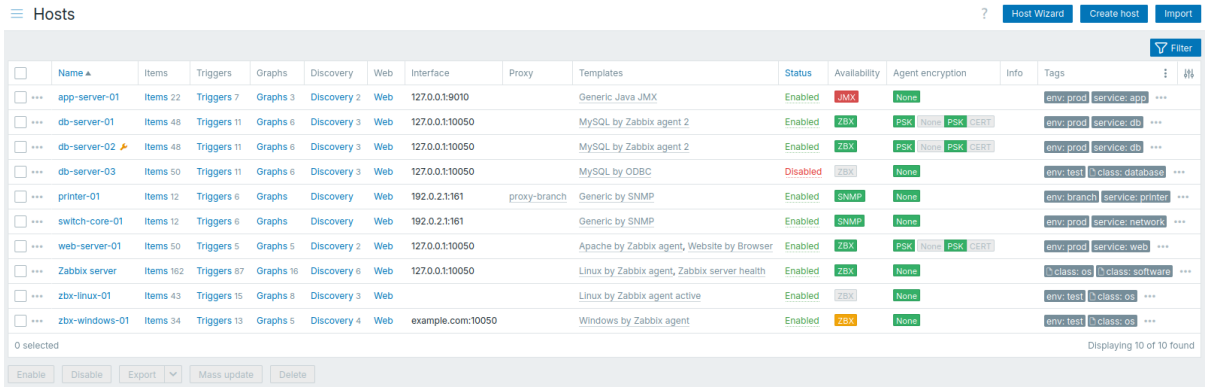
Parameter	Beschreibung
Vorlagengruppen	Nach einer oder mehreren Vorlagengruppen filtern. Die Angabe einer übergeordneten Vorlagengruppe wählt implizit alle verschachtelten Gruppen aus.
Verknüpfte Vorlagen	Nach direkt verknüpften Vorlagen filtern.
Name	Nach Vorlagennamen filtern.
Vendor	Nach Vorlagenhersteller filtern.
Version	Nach Vorlagenversion filtern.
Tags	Nach Vorlagen-Tag-Namen und -Wert filtern. Das Filtern ist nur nach Tags auf Vorlagenebene möglich (nicht nach geerbten Tags). Es ist möglich, bestimmte Tags und Tag-Werte einzuschließen sowie auszuschließen. Es können mehrere Bedingungen festgelegt werden. Die Übereinstimmung des Tag-Namens ist immer groß-/kleinschreibungssensitiv. Für jede Bedingung stehen mehrere Operatoren zur Verfügung: **Exists** - die angegebenen Tag-Namen einschließen; **Equals** - die angegebenen Tag-Namen und -Werte einschließen (groß-/kleinschreibungssensitiv); **Contains** - die angegebenen Tag-Namen einschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv); **Does not exist** - die angegebenen Tag-Namen ausschließen; **Does not equal** - die angegebenen Tag-Namen und -Werte ausschließen (groß-/kleinschreibungssensitiv); **Does not contain** - die angegebenen Tag-Namen ausschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv). Für Bedingungen gibt es zwei Berechnungstypen: **And/Or** - alle Bedingungen müssen erfüllt sein, Bedingungen mit demselben Tag-Namen werden durch die Or-Bedingung gruppiert; **Or** - es genügt, wenn eine Bedingung erfüllt ist.

Hosts

Übersicht

Im Abschnitt *Datensammlung* > *Hosts* können Benutzer Hosts konfigurieren und verwalten.

Eine Liste der vorhandenen Hosts mit ihren Details wird angezeigt.



Hostdetails werden in einer anpassbaren Tabelle angezeigt. Mit der Schaltfläche *Tabelle anpassen* in der Tabellenkopfzeile können Sie Spalten ein- oder ausblenden, neu anordnen und das Layout auf die Standardeinstellungen zurücksetzen. Sie können außerdem die Größe jeder Spalte ändern, indem Sie ihren rechten Rand ziehen. Durch einen Doppelklick auf den Rand wird die ursprüngliche Größe wiederhergestellt. Alle Anpassungen werden sofort angewendet und für den aktuellen Benutzer dauerhaft gespeichert.

Die folgenden Spalten sind verfügbar:

Spalte	Beschreibung
Name	Name des Hosts. Durch Klicken auf den Hostnamen wird das Host-Konfigurationsformular geöffnet. Ein orangefarbenes Schraubenschlüsselsymbol hinter dem Hostnamen zeigt an, dass sich dieser Host im Wartungsmodus befindet. Die Wartungsdetails werden angezeigt, wenn sich der Mauszeiger über dem Symbol befindet.
Entitäten (Datenpunkte, Auslöser, Graphen, Discovery, Web)	Durch Klicken auf den Namen der Entität werden die Datenpunkte, Auslöser usw. des Hosts angezeigt. Die Anzahl der jeweiligen Entitäten wird grau angezeigt.
Schnittstelle	Die Hauptschnittstelle des Hosts wird angezeigt.
Proxy	In dieser Spalte werden zugewiesene Proxys angezeigt: <Proxy name> - der Host wird von einem eigenständigen Proxy überwacht (auch wenn der Proxy Teil einer Proxy-Gruppe ist); <Proxy group name: proxy name> - der Host wird von einer Proxy-Gruppe überwacht und der Zabbix-Server hat einen Proxy zur Überwachung des Hosts zugewiesen; <Proxy group name> - der Host wird von einer Proxy-Gruppe ohne Proxys überwacht oder der Zabbix-Server hat keinen Proxy zur Überwachung des Hosts zugewiesen; Nichts - der Host wird weder von einem Proxy noch von einer Proxy-Gruppe überwacht. Diese Spalte wird nur angezeigt, wenn die Filteroption *Überwacht von* auf „Beliebig“, „Proxy“ oder „Proxy-Gruppe“ gesetzt ist.
Vorlagen	Die mit dem Host verknüpften Vorlagen werden angezeigt. Wenn die verknüpfte Vorlage weitere Vorlagen enthält, werden diese in Klammern und durch Kommas getrennt angezeigt. Durch Klicken auf den Namen einer Vorlage wird deren Konfigurationsformular geöffnet.
Status	Der Hoststatus wird angezeigt - *Aktiviert* oder *Deaktiviert*. Durch Klicken auf den Status können Sie ihn manuell ändern. Entdeckte Hosts, die nicht mehr vorhanden sind, werden mit einem Infosymbol gekennzeichnet. Der Tooltip enthält Details zu ihrem Status.

Spalte	Beschreibung
Verfügbarkeit	Die Hostverfügbarkeit für jede konfigurierte Schnittstelle wird angezeigt. Verfügbarkeitssymbole stellen den aktuellen Status der Host-Schnittstelle auf dem Zabbix-Server dar. Wenn Sie einen Host im Frontend deaktivieren, wird seine Verfügbarkeit daher erst aktualisiert, nachdem der Zabbix-Server die Konfigurationsänderungen synchronisiert hat. Wenn Sie einen Host aktivieren, wird seine Verfügbarkeit ebenfalls erst aktualisiert, nachdem der Zabbix-Server die Konfigurationsänderungen synchronisiert und den Host abgefragt hat. Verfügbarkeitssymbole werden nur für konfigurierte Schnittstellentypen (Agent, SNMP, IPMI, JMX) angezeigt. Wenn Sie den Mauszeiger über das Symbol bewegen, wird ein Pop-up mit einer Liste aller Schnittstellen desselben Typs einschließlich Details, Status und Fehlern angezeigt. Bei einer Agent-Schnittstelle zeigt das Pop-up passive Schnittstellen und aktive Prüfungen an. Wenn ein Host nur aktive Prüfungen hat, wird das Agent-Schnittstellensymbol angezeigt, auch wenn für den Host keine Agent-Schnittstelle konfiguriert ist. Bei Hosts ohne Schnittstellen bleibt die Spalte leer. Der Status einer einzelnen Host-Schnittstelle wird anhand der Verbindung zwischen einem aktivierten Datenpunkt, der die Schnittstelle verwendet, und dem Host bestimmt. Der Status kann sein: **Verfügbar** - die Verbindung zum Host war erfolgreich; **Nicht verfügbar** - die Verbindung zum Host war nicht erfolgreich (Timeout, Firewall-Probleme usw.); **Unbekannt** - es wurde noch keine Verbindung zum Host versucht oder das Ergebnis ist unbekannt. Weitere Informationen dazu, wie der Zabbix-Server den Schnittstellenstatus bestimmt, finden Sie unter Unbekannter Schnittstellenstatus und Einstellungen für nicht erreichbare/nicht verfügbare Host-Schnittstellen. Der Status aller Host-Schnittstellen eines einzelnen Typs (Agent, SNMP, IPMI, JMX) wird anhand der Schnittstellen bestimmt, die von mindestens einem aktivierten Datenpunkt verwendet werden. Der Status wird durch die Symbolfarbe angezeigt: **Grün** - alle Schnittstellen sind verfügbar; **Gelb** - mindestens eine Schnittstelle ist nicht verfügbar und mindestens eine ist verfügbar oder unbekannt; **Rot** - alle Schnittstellen sind nicht verfügbar; **Grau** - mindestens eine Schnittstelle ist unbekannt, aber keine ist nicht verfügbar. **Verfügbarkeit aktiver Prüfungen.** Wenn auf dem Host mindestens eine aktive Prüfung aktiviert ist, beeinflussen aktive Prüfungen ebenfalls die Gesamtverfügbarkeit der Agent-Schnittstelle wie oben beschrieben. Zur Bestimmung der Verfügbarkeit aktiver Prüfungen werden im Thread für aktive Prüfungen Heartbeat-Nachrichten gesendet. Die Häufigkeit der Heartbeat-Nachrichten wird durch den Parameter `HeartbeatFrequency` in der Konfiguration des Zabbix-Agents oder Agents 2 gesteuert (Standardwert 60 Sekunden, Bereich 0-3600). Aktive Prüfungen gelten als nicht verfügbar, wenn der Heartbeat der aktiven Prüfung älter als 2 x HeartbeatFrequency Sekunden ist. **Hinweis:** Zabbix Agents älter als Version 6.2.x senden keine Heartbeats für aktive Prüfungen. Daher bleibt die Verfügbarkeit ihrer Hosts unbekannt.
Agent-Verschlüsselung	Der Verschlüsselungsstatus für Verbindungen zum Host und vom Host wird angezeigt: **Keine** - keine Verschlüsselung; **PSK** - vorinstallierter Schlüssel wird verwendet; **Zertifikat** - Zertifikat wird verwendet.
Info	Fehlerinformationen (falls vorhanden) zum Host werden angezeigt.
Tags	Host-Tags werden mit nicht aufgelösten Makros angezeigt. Wenn Sie den Mauszeiger über das Symbol ••• mit den drei Punkten nach den angezeigten Tags bewegen oder darauf klicken, wird ein Pop-up mit allen Tags geöffnet. Durch Klicken auf die Schaltfläche ⋮ *Optionen* in der Spaltenkopfzeile können Sie die Anzahl der angezeigten Tags (1–3) festlegen, auswählen, wie Tag-Namen angezeigt werden (vollständig, gekürzt oder gar nicht), die Priorität der Tag-Anzeige festlegen oder die Spalte duplizieren (duplizierte Spalten können umbenannt oder gelöscht werden).

Spalte	Beschreibung
Tag-Wert	Werte von Host-Tags werden mit nicht aufgelösten Makros angezeigt. Durch Klicken auf die Schaltfläche ⋮ *Optionen* in der Spaltenkopfzeile können Sie die Spalte umbenennen, auswählen, welche Tag-Werte angezeigt werden, oder die Spalte duplizieren (duplizierte Spalten können umbenannt oder gelöscht werden).
Benutzerdefinierter Text	Benutzerdefinierter Text, Makrowerte oder eine Kombination aus beidem wird angezeigt. Durch Klicken auf die Schaltfläche ⋮ *Optionen* in der Spaltenkopfzeile können Sie die Spalte umbenennen, den benutzerdefinierten Text oder Makros eingeben oder die Spalte duplizieren (duplizierte Spalten können umbenannt oder gelöscht werden). Unterstützte Makros: `{HOST.*}` (außer `{HOST.METADATA}`), `{INVENTORY.*}` und Benutzermakros.

Um das Host-Menü zu öffnen, klicken Sie auf die Schaltfläche ⋮ *Optionen* vor dem Hostnamen.

Um einen neuen Host zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Host-Assistent* oder *Host erstellen*.

Um einen Host aus einer YAML-, XML- oder JSON-Datei zu importieren, klicken Sie oben rechts auf die Schaltfläche *Importieren*.

Optionen für die Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen für die Massenbearbeitung:

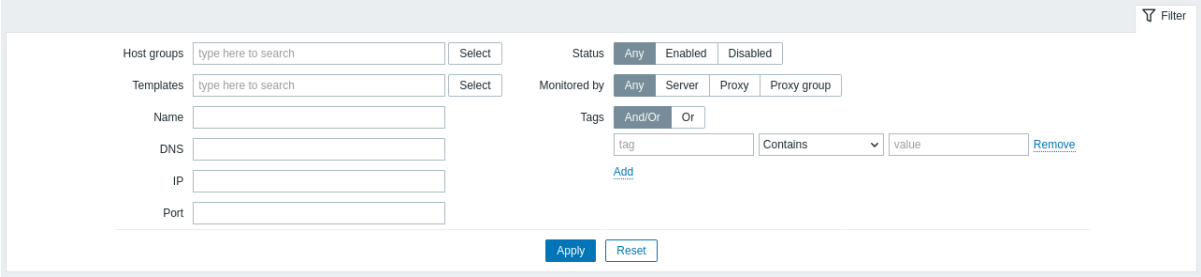
- *Enable* - den Host-Status auf *Überwacht* ändern;
- *Disable* - den Host-Status auf *Nicht überwacht* ändern;
- *Export* - die Hosts in eine YAML-, XML- oder JSON-Datei exportieren;
- *Mass update* - mehrere Eigenschaften aktualisieren für mehrere Hosts gleichzeitig;
- *Delete* - die Hosts löschen.

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Hosts und klicken Sie dann auf die gewünschte Schaltfläche.

Filter verwenden

Sie können den Filter verwenden, um nur die Hosts anzuzeigen, die Sie interessieren. Für eine bessere Suchleistung werden Daten mit nicht aufgelösten Makros durchsucht.

Das Symbol *Filter* befindet sich in der oberen rechten Ecke. Wenn Sie darauf klicken, wird ein Filter geöffnet, in dem Sie die gewünschten Filterkriterien angeben können.



Parameter	Beschreibung
Host groups	Nach einem oder mehreren Host groups filtern. Wenn eine übergeordnete Host group angegeben wird, werden implizit alle verschachtelten Host groups ausgewählt.
Templates	Nach verknüpften Vorlagen filtern.
Name	Nach dem sichtbaren Hostnamen filtern.
DNS	Nach dem DNS-Namen filtern.
IP	Nach der IP-Adresse filtern.
Port	Nach der Portnummer filtern.
Status	Nach dem Host-Status filtern.
Monitored by	Hosts filtern, die von Zabbix Server, Proxy oder Proxy group überwacht werden.
Proxies	Hosts filtern, die von den hier angegebenen Proxies überwacht werden. Dieses Feld ist nur verfügbar, wenn im Feld *Monitored by* "Proxy" ausgewählt ist.
Proxy groups	Hosts filtern, die von den hier angegebenen Proxy groups überwacht werden. Dieses Feld ist nur verfügbar, wenn im Feld *Monitored by* "Proxy group" ausgewählt ist.

Parameter	Beschreibung
Tags	Nach Host-Tag-Namen und -Werten filtern. Es ist möglich, bestimmte Tags und Tag-Werte einzuschließen sowie auszuschließen. Es können mehrere Bedingungen festgelegt werden. Die Übereinstimmung des Tag-Namens ist immer groß-/kleinschreibungssensitiv. Für jede Bedingung stehen mehrere Operatoren zur Verfügung: **Exists** - die angegebenen Tag-Namen einschließen; **Equals** - die angegebenen Tag-Namen und -Werte einschließen (groß-/kleinschreibungssensitiv); **Contains** - die angegebenen Tag-Namen einschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv); **Does not exist** - die angegebenen Tag-Namen ausschließen; **Does not equal** - die angegebenen Tag-Namen und -Werte ausschließen (groß-/kleinschreibungssensitiv); **Does not contain** - die angegebenen Tag-Namen ausschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv). Für Bedingungen gibt es zwei Berechnungstypen: **And/Or** - alle Bedingungen müssen erfüllt sein, Bedingungen mit demselben Tag-Namen werden durch die Or-Bedingung gruppiert; **Or** - es genügt, wenn eine Bedingung erfüllt ist.

Unbekannter Schnittstellenstatus

Der Zabbix-Server zeigt den Status ”Unknown” für eine Host-Schnittstelle (Agent, SNMP, IPMI, JMX) in den folgenden Fällen an:

- Host ist deaktiviert.
- Host ist so eingestellt, dass er von einem Proxy, einem anderen Proxy oder vom Server überwacht wird, wenn er zuvor von einem Proxy überwacht wurde.
- Host wird von einem Proxy überwacht, der offenbar offline ist (keine Aktualisierungen vom Proxy während des maximalen Heartbeat-Intervalls - 1 Stunde empfangen).
- Alle Datenpunkte des Hosts mit diesem Schnittstellentyp sind deaktiviert.
- Für diesen Schnittstellentyp wurden keine Poller konfiguriert (z. B. ist der Server-Konfigurationsparameter `StartAgentPollers` oder `StartSNMPPollers` auf 0 gesetzt).

Die Verfügbarkeit der Schnittstelle wird nach der Synchronisierung des Konfigurationscaches des Zabbix-Servers auf ”Unknown” gesetzt.

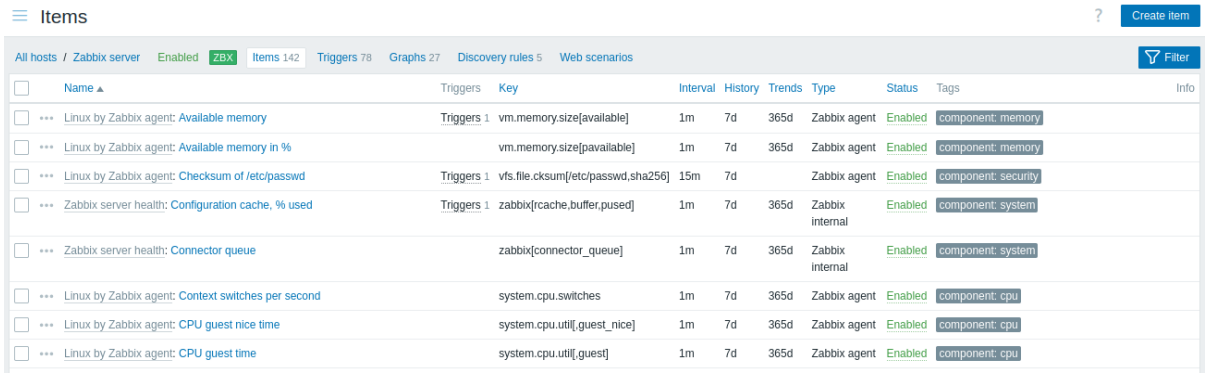
Die Verfügbarkeit der Schnittstelle (Verfügbar/Nicht verfügbar) auf Hosts, die von Proxys überwacht werden, wird nach der Synchronisierung des Proxy-Konfigurationscaches wiederhergestellt.

Datenpunkte

Übersicht

Die Liste der Datenpunkte für einen Host kann über *Datensammlung* > *Hosts* aufgerufen werden, indem Sie beim entsprechenden Host auf *Datenpunkte* klicken.

Eine Liste der vorhandenen Datenpunkte wird angezeigt.



Angezeigte Daten:

Spalte	Beschreibung
Datenpunktmenü	Klicken Sie auf das Drei-Punkte-Symbol, um das Datenpunktmenü zu öffnen.
Host	Host des Datenpunkts. Durch Klicken auf den Hostnamen wird das Host-Konfigurationsformular geöffnet. Diese Spalte wird nur angezeigt, wenn im Filter mehrere Hosts oder keine Hosts ausgewählt sind.
Name	Name des Datenpunkts, der als blauer Link zu den Datenpunktdetails angezeigt wird. Durch Klicken auf den Link mit dem Datenpunktnamen wird das Datenpunkt-Konfigurationsformular geöffnet. Wenn der Datenpunkt zu einer Vorlage gehört, wird der Vorlagenname vor dem Datenpunktnamen als grauer Link angezeigt. Durch Klicken auf den Vorlagenlink wird die Datenpunktliste auf Vorlagenebene geöffnet. Wenn der Datenpunkt aus einem Datenpunktprototyp erstellt wurde, wird seinem Namen der Name der Low-Level-Discovery-Regel in Orange vorangestellt. Durch Klicken auf den Namen der Discovery-Regel wird die Liste der Datenpunktprototypen geöffnet.
Auslöser	Wenn Sie den Mauszeiger über *Auslöser* bewegen, wird ein Infokasten mit den mit dem Datenpunkt verknüpften Auslösern angezeigt. Die Anzahl der Auslöser wird grau angezeigt.
Schlüssel	Der Schlüssel des Datenpunkts wird angezeigt.
Intervall	Die Häufigkeit der Prüfung wird angezeigt. *Hinweis*: Passive Datenpunkte können auch sofort geprüft werden, indem Sie auf die Schaltfläche *Jetzt ausführen* klicken.
Historie	Die Anzahl der Tage, für die die Datenpunkthistorie aufbewahrt wird, wird angezeigt.
Trends	Die Anzahl der Tage, für die die Trendhistorie des Datenpunkts aufbewahrt wird, wird angezeigt.
Typ	Der Typ des Datenpunkts wird angezeigt (Zabbix Agent, SNMP Agent, einfache Prüfung usw.).
Status	Der Status des Datenpunkts wird angezeigt – *Aktiviert*, *Deaktiviert* oder *Nicht unterstützt*. Sie können den Status manuell ändern, indem Sie darauf klicken: von „Aktiviert“ zu „Deaktiviert“ (und zurück) sowie von „Nicht unterstützt“ zu „Deaktiviert“ (und zurück). Entdeckte Datenpunkte, die verloren gegangen sind, werden mit einem Infosymbol gekennzeichnet. Der Tooltip-Text enthält Details zu ihrem Status.
Tags	Die Tags des Datenpunkts werden angezeigt. Bis zu drei Tags (Name-Wert-Paare) können angezeigt werden. Wenn weitere Tags vorhanden sind, wird ein Link „...“ angezeigt, über den beim Überfahren mit dem Mauszeiger alle Tags angezeigt werden können.
Info	Wenn der Datenpunkt ordnungsgemäß funktioniert, wird in dieser Spalte kein Symbol angezeigt. Bei Fehlern wird ein quadratisches Symbol mit dem Buchstaben „i“ angezeigt. Bewegen Sie den Mauszeiger über das Symbol, um einen Tooltip mit der Fehlerbeschreibung anzuzeigen.

Um einen neuen Datenpunkt zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Datenpunkt erstellen*.

Optionen für die Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen für die Massenbearbeitung:

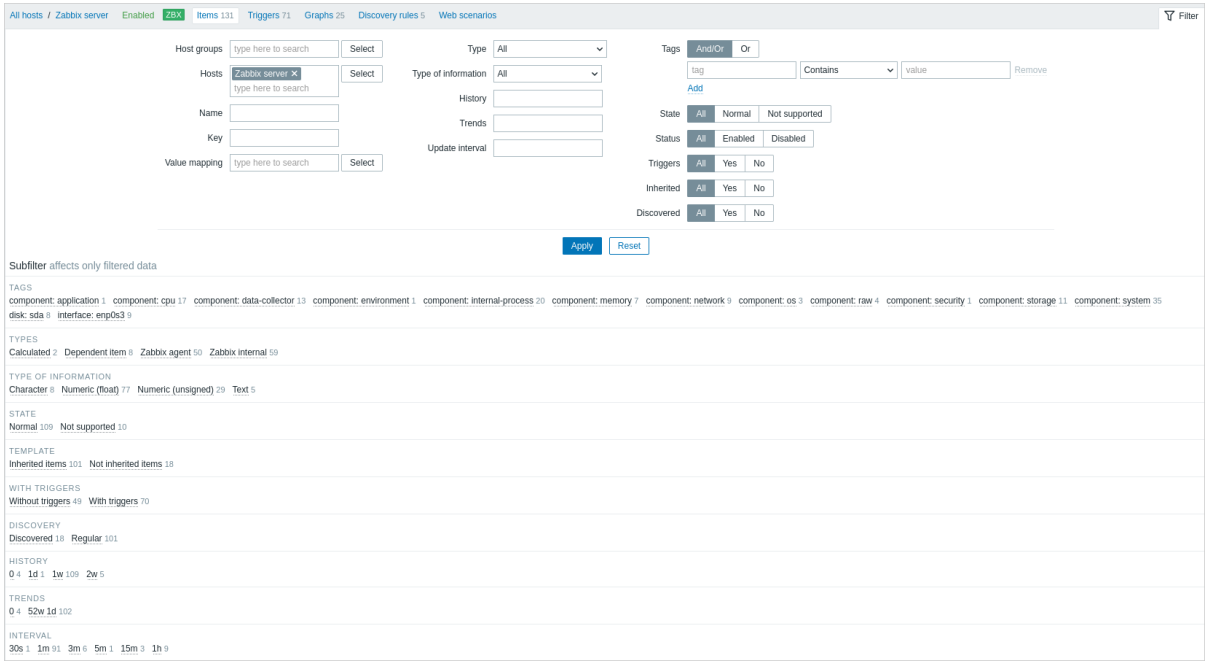
- *Enable* - den Status des Datenpunkts auf *Enabled* ändern
- *Disable* - den Status des Datenpunkts auf *Disabled* ändern
- *Execute now* - eine Prüfung auf neue Datenpunktwerte sofort ausführen. Nur für **passive** Prüfungen unterstützt (siehe weitere Details). Beachten Sie, dass bei der sofortigen Prüfung von Werten der Konfigurations-Cache nicht aktualisiert wird und die Werte daher nicht die sehr aktuellen Änderungen an der Datenpunktkonfiguration widerspiegeln.
- *Clear history and trends* - Verlaufs- und Trenddaten für Datenpunkte löschen.
- *Copy* - die Datenpunkte auf andere Hosts oder Vorlagen kopieren.
- *Mass update* - mehrere Eigenschaften für eine Anzahl von Datenpunkten gleichzeitig aktualisieren.
- *Delete* - die Datenpunkte löschen.

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Datenpunkten und klicken Sie dann auf die gewünschte Schaltfläche.

Filter verwenden

Sie können den Filter verwenden, um nur die Datenpunkte anzuzeigen, die Sie interessieren. Für eine bessere Suchleistung werden die Daten mit nicht aufgelösten Makros durchsucht.

Das Symbol *Filter* ist oben rechts verfügbar. Wenn Sie darauf klicken, wird ein Filter geöffnet, in dem Sie die gewünschten Filterkriterien angeben können.

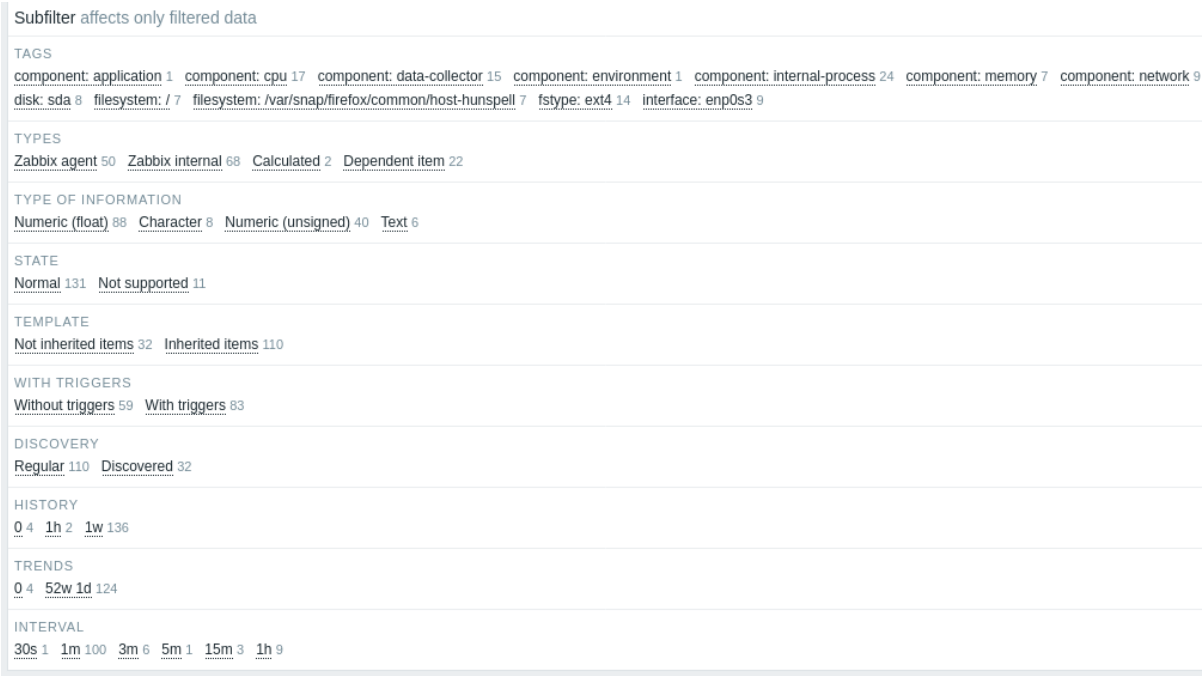


Parameter	Beschreibung
Host groups	Nach einer oder mehreren Hostgruppen filtern. Durch das Angeben einer übergeordneten Hostgruppe werden implizit alle verschachtelten Hostgruppen ausgewählt. Hostgruppen, die nur Vorlagen enthalten, können nicht ausgewählt werden.
Hosts	Nach einem oder mehreren Hosts filtern.
Name	Nach dem Namen des Datenpunkts filtern.
Key	Nach dem Schlüssel des Datenpunkts filtern.
Value mapping	Nach der verwendeten Wertzuordnung filtern. Dieser Parameter wird nicht angezeigt, wenn die Option *Hosts* leer ist.
Type	Nach dem Typ des Datenpunkts filtern (Zabbix agent, SNMP agent usw.).
Type of information	Nach dem Informationstyp filtern (Unsigned numeric, float usw.).
History	Nach der Aufbewahrungsdauer des Datenpunktverlaufs filtern.
Trends	Nach der Aufbewahrungsdauer der Datenpunkt-Trends filtern.
Update interval	Nach dem Aktualisierungsintervall des Datenpunkts filtern.
Tags	Geben Sie Tags an, um die Anzahl der angezeigten Datenpunkte einzuschränken. Es ist möglich, bestimmte Tags und Tag-Werte einzuschließen oder auszuschließen. Es können mehrere Bedingungen festgelegt werden. Die Übereinstimmung des Tag-Namens ist immer groß-/kleinschreibungssensitiv. Für jede Bedingung stehen mehrere Operatoren zur Verfügung: **Exists** - die angegebenen Tag-Namen einschließen **Equals** - die angegebenen Tag-Namen und Werte einschließen (groß-/kleinschreibungssensitiv) **Contains** - die angegebenen Tag-Namen einschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv) **Does not exist** - die angegebenen Tag-Namen ausschließen **Does not equal** - die angegebenen Tag-Namen und Werte ausschließen (groß-/kleinschreibungssensitiv) **Does not contain** - die angegebenen Tag-Namen ausschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv) Für Bedingungen gibt es zwei Berechnungstypen: **And/Or** - alle Bedingungen müssen erfüllt sein, Bedingungen mit demselben Tag-Namen werden durch die Or-Bedingung gruppiert **Or** - es genügt, wenn eine Bedingung erfüllt ist
State	Nach dem Zustand des Datenpunkts filtern - *Normal* oder *Not supported*.
Status	Nach dem Status des Datenpunkts filtern - *Enabled* oder *Disabled*.
Triggers	Datenpunkte mit (oder ohne) Auslösern filtern.
Inherited	Geerbte (oder nicht geerbte) Datenpunkte aus einer Vorlage filtern.
Discovery	Datenpunkte filtern, die durch Low-Level-Discovery erkannt wurden (oder nicht erkannt wurden).

Subfilter verwenden

Der Subfilter ermöglicht eine weitere Anpassung der Filterung, die durch den Hauptfilter angewendet wird.

Er enthält anklickbare Links für den schnellen Zugriff auf verwandte Datenpunkte. Datenpunkte werden nach einer gemeinsamen Entität gruppiert – Tag, Datentyp, Datenpunktstatus, Aktualisierungsintervall des Datenpunkts usw. Wenn eine Entität angeklickt wird, wird sie mit einem grauen Hintergrund hervorgehoben und die Datenpunkte werden sofort gefiltert (im Hauptfilter muss nicht auf *Anwenden* geklickt werden). Durch das Anklicken einer weiteren Entität wird diese zu den gefilterten Ergebnissen hinzugefügt. Durch erneutes Anklicken der Entität wird die Filterung aufgehoben.



Subfilter werden auf Grundlage der gefilterten Daten generiert, die auf 1000 Datensätze begrenzt sind. Wenn Sie mehr Datensätze im Subfilter anzeigen möchten, müssen Sie den Wert des Parameters *Limit für Such- und Filterergebnisse* erhöhen (unter *Administration* > *Allgemein* > *GUI*).

Im Gegensatz zum Hauptfilter wird der Subfilter bei jeder Aktualisierungsanforderung der Tabelle aktualisiert, damit die verfügbaren Filteroptionen und deren Anzahl stets aktuell sind.

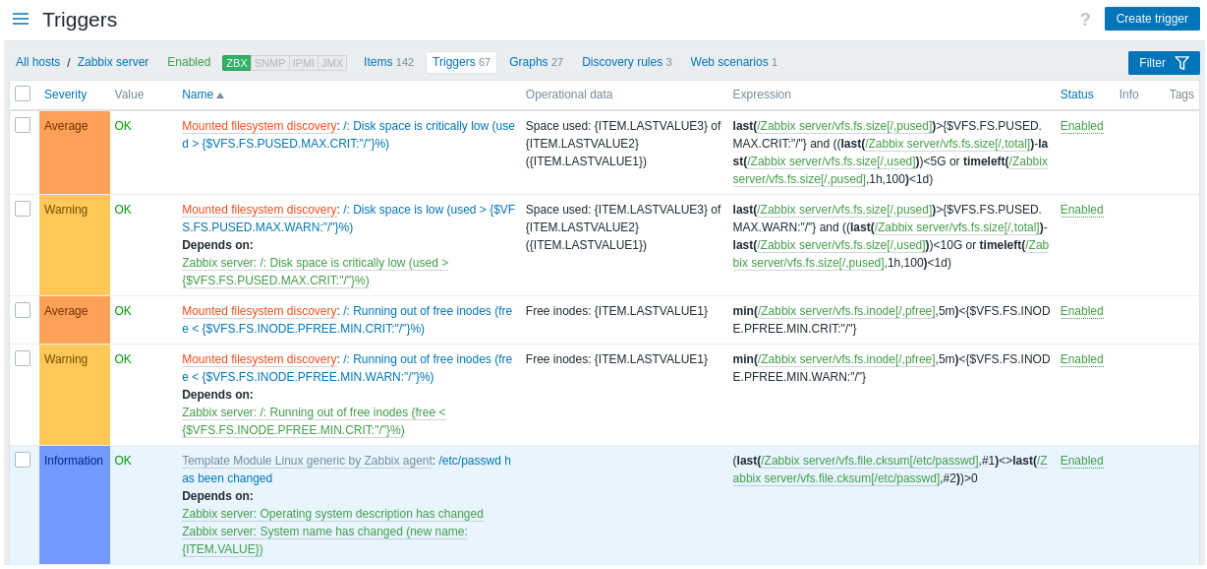
Die Anzahl der horizontal angezeigten Entitäten ist auf 100 begrenzt. Wenn es mehr Entitäten gibt, wird am Ende ein Symbol mit drei Punkten angezeigt; es ist nicht anklickbar.

Eine Zahl neben jeder anklickbaren Entität gibt die Anzahl der darin gruppierten Datenpunkte an (basierend auf den Ergebnissen des Hauptfilters). Wenn eine Entität angeklickt wird, werden neben den anderen verfügbaren Entitäten Zahlen mit einem Pluszeichen angezeigt. Diese geben an, wie viele Datenpunkte zur aktuellen Auswahl hinzugefügt werden können. Entitäten ohne Datenpunkte werden nicht angezeigt, es sei denn, sie wurden zuvor im Subfilter ausgewählt.

Auslöser

Übersicht

Die Auslöserliste für einen Host kann unter *Datensammlung* > *Hosts* aufgerufen werden, indem Sie beim entsprechenden Host auf *Auslöser* klicken.



Angezeigte Daten:

Spalte	Beschreibung
Schweregrad	Der Schweregrad des Auslösers wird sowohl durch den Namen als auch durch die Hintergrundfarbe der Zelle angezeigt.
Wert	Der Status des Auslösers wird angezeigt: **OK** – OK-Status **PROBLEM** – Problemstatus
Host	Host des Auslösers. Durch Klicken auf den Hostnamen wird das Host-Konfigurationsformular geöffnet. Diese Spalte wird nur angezeigt, wenn im Filter mehrere Hosts oder keine Hosts ausgewählt sind.
Name	Name des Auslösers, angezeigt als blauer Link zu den Auslöserdetails. Durch Klicken auf den Link mit dem Auslösernamen wird das Auslöser-Konfigurationsformular geöffnet. Wenn der Host-Auslöser zu einer Vorlage gehört, wird der Vorlagenname vor dem Auslösernamen als grauer Link angezeigt. Durch Klicken auf den Vorlagenlink wird die Auslöserliste auf Vorlagenebene geöffnet. Wenn der Auslöser aus einem Auslöserprototyp erstellt wurde, wird seinem Namen der Name der Low-Level-Discovery-Regel in Orange vorangestellt. Durch Klicken auf den Namen der Discovery-Regel wird die Liste der Auslöserprototypen geöffnet.
Betriebsdaten	Definition der Betriebsdaten des Auslösers, die beliebige Zeichenfolgen und Makros enthalten kann, die unter *Überwachung* > *Probleme* dynamisch aufgelöst werden.
Ausdruck	Der Ausdruck des Auslösers wird angezeigt. Der Host-Datenpunkt-Teil des Ausdrucks wird als Link angezeigt, der zum Datenpunkt-Konfigurationsformular führt.
Status	Der Status des Auslösers wird angezeigt – *Aktiviert*, *Deaktiviert* oder *Unbekannt*. Durch Klicken auf den Status können Sie ihn manuell ändern – von „Aktiviert“ zu „Deaktiviert“ (und zurück) sowie von „Unbekannt“ zu „Deaktiviert“ (und zurück). Probleme eines deaktivierten Auslösers werden im Frontend nicht mehr angezeigt, aber nicht gelöscht. Entdeckte Auslöser, die nicht mehr vorhanden sind, werden mit einem Infosymbol gekennzeichnet. Der Tooltip-Text enthält Details zu ihrem Status.
Info	Wenn alles ordnungsgemäß funktioniert, wird in dieser Spalte kein Symbol angezeigt. Bei Fehlern wird ein quadratisches Symbol mit dem Buchstaben „i“ angezeigt. Bewegen Sie den Mauszeiger über das Symbol, um einen Tooltip mit der Fehlerbeschreibung anzuzeigen.
Tags	Wenn ein Auslöser Tags enthält, werden der Tagname und der Wert in dieser Spalte angezeigt.

Um einen neuen Auslöser zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Auslöser erstellen*.

Optionen für die Massenbearbeitung

Die Schaltflächen unter der Liste bieten einige Optionen für die Massenbearbeitung:

- *Aktivieren* - den Status des Auslösers auf *Aktiviert* ändern.
- *Deaktivieren* - den Status des Auslösers auf *Deaktiviert* ändern.
- *Kopieren* - die Auslöser auf andere Hosts oder Vorlagen kopieren.

- *Massenaktualisierung* - mehrere Eigenschaften für eine Anzahl von Auslösern gleichzeitig aktualisieren.
- *Löschen* - die Auslöser löschen.

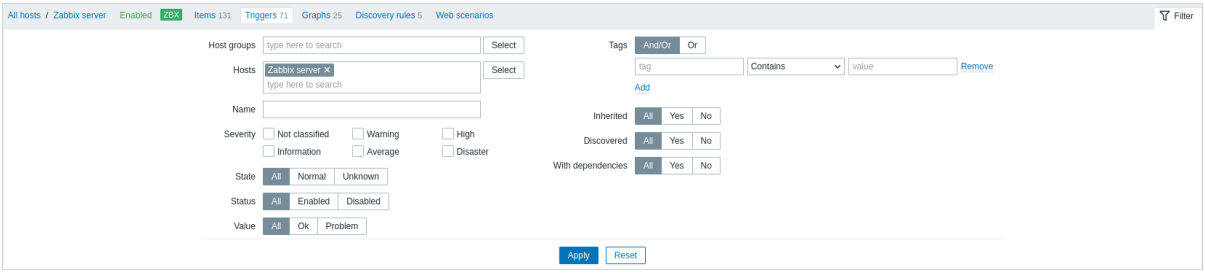
Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Auslösern und klicken Sie dann auf die gewünschte Schaltfläche.

Filter verwenden

Sie können den Filter verwenden, um nur die Auslöser anzuzeigen, die Sie interessieren.

Für eine bessere Suchleistung werden die Daten mit nicht aufgelösten Makros durchsucht.

Das Symbol *Filter* ist oben rechts verfügbar. Wenn Sie darauf klicken, wird ein Filter geöffnet, in dem Sie die gewünschten Filterkriterien festlegen können.



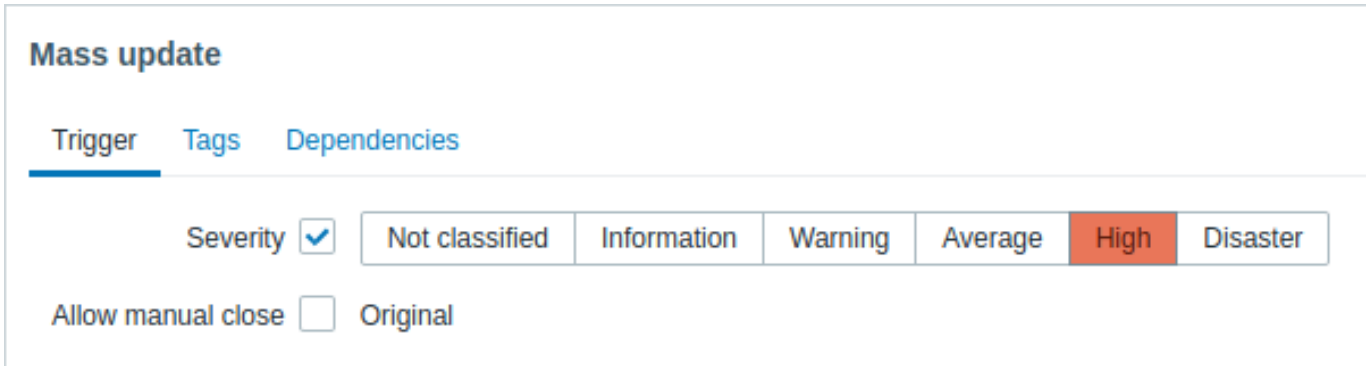
Parameter	Beschreibung
Host groups	Nach einer oder mehreren Hostgruppen filtern. Wenn eine übergeordnete Hostgruppe angegeben wird, werden implizit alle verschachtelten Hostgruppen ausgewählt. Hostgruppen, die nur Vorlagen enthalten, können nicht ausgewählt werden.
Hosts	Nach einem oder mehreren Hosts filtern. Wenn oben bereits Hostgruppen ausgewählt sind, ist die Hostauswahl auf diese Gruppen beschränkt.
Name	Nach dem Namen des Auslösers filtern.
Severity	Auswählen, um nach einer oder mehreren Auslöser-Schweregraden zu filtern.
State	Nach dem Ausdruckszustand des Auslösers filtern (Normal/Unknown).
Status	Nach dem Aktivierungsstatus des Auslösers filtern (Enabled/Disabled).
Value	Nach dem Wert des Auslösers filtern.
Tags	Nach Tag-Namen und -Wert des Auslösers filtern. Es ist möglich, bestimmte Tags und Tag-Werte einzuschließen sowie auszuschließen. Es können mehrere Bedingungen festgelegt werden. Die Übereinstimmung des Tag-Namens ist immer groß-/kleinschreibungssensitiv. Für jede Bedingung stehen mehrere Operatoren zur Verfügung: **Exists** - die angegebenen Tag-Namen einschließen **Equals** - die angegebenen Tag-Namen und -Werte einschließen (groß-/kleinschreibungssensitiv) **Contains** - die angegebenen Tag-Namen einschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv) **Does not exist** - die angegebenen Tag-Namen ausschließen **Does not equal** - die angegebenen Tag-Namen und -Werte ausschließen (groß-/kleinschreibungssensitiv) **Does not contain** - die angegebenen Tag-Namen ausschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv) Für Bedingungen gibt es zwei Berechnungstypen: **And/Or** - alle Bedingungen müssen erfüllt sein, Bedingungen mit demselben Tag-Namen werden durch die Or-Bedingung gruppiert **Or** - es genügt, wenn eine Bedingung erfüllt ist Makros und Makrofunktionen werden sowohl in den Feldern für den Tag-Namen als auch für den Tag-Wert unterstützt.
Inherited	Auslöser filtern, die von einer Vorlage geerbt wurden (oder nicht geerbt wurden).
Discovered	Auslöser filtern, die durch Low-Level-Discovery entdeckt wurden (oder nicht entdeckt wurden).
With dependencies	Auslöser mit (oder ohne) Abhängigkeiten filtern.

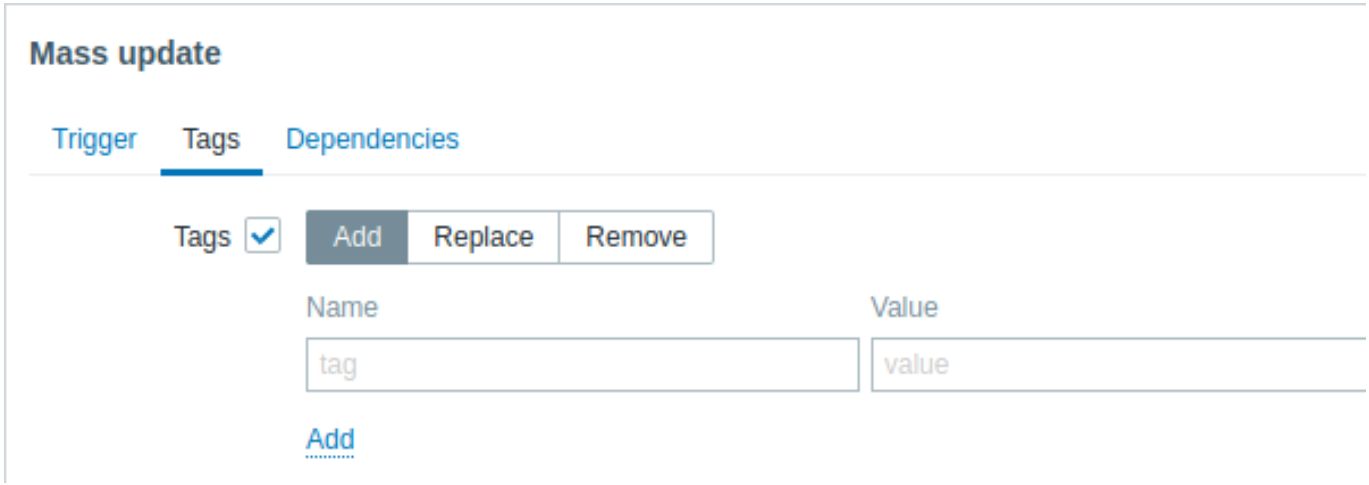
Massenaktualisierung verwenden

Mit der Massenaktualisierung können Sie ein bestimmtes Attribut für mehrere Auslöser gleichzeitig ändern. Dadurch müssen Sie nicht jeden einzelnen Auslöser zur Bearbeitung öffnen.

Gehen Sie wie folgt vor, um mehrere Auslöser gleichzeitig zu aktualisieren:

- Aktivieren Sie in der Liste die Kontrollkästchen der Auslöser, die Sie aktualisieren möchten.
- Klicken Sie unterhalb der Liste auf *Massenaktualisierung*.
- Wechseln Sie zu der Registerkarte mit den erforderlichen Attributen (*Auslöser*, *Tags* oder *Abhängigkeiten*).
- Aktivieren Sie die Kontrollkästchen der zu aktualisierenden Attribute.

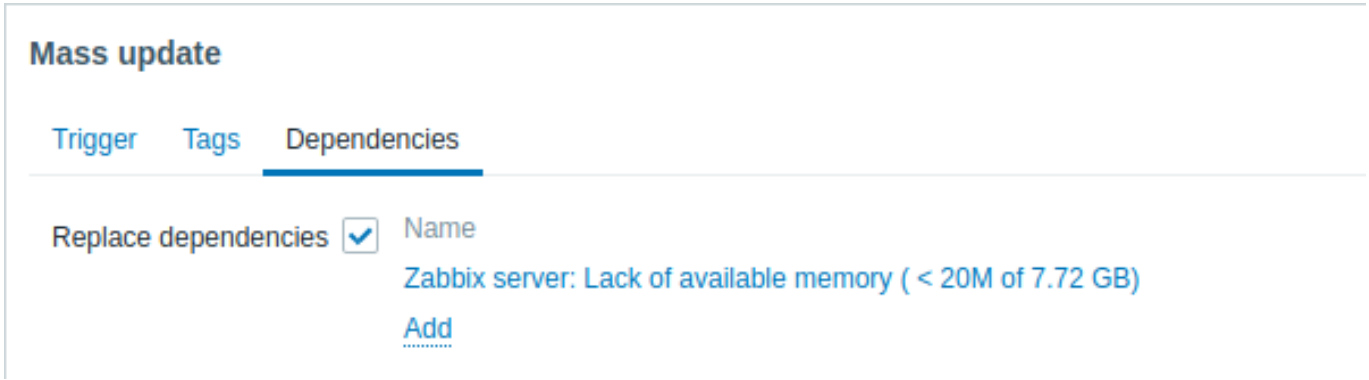




Bei der Auswahl der entsprechenden Schaltfläche für die Tag-Aktualisierung stehen folgende Optionen zur Verfügung:

- *Hinzufügen* – neue Tags für die Auslöser hinzufügen;
- *Ersetzen* – alle vorhandenen Tags aus dem Auslöser entfernen und durch die unten angegebenen Tags ersetzen;
- *Entfernen* – die angegebenen Tags aus den Auslösern entfernen.

Beachten Sie, dass Tags mit demselben Namen, aber unterschiedlichen Werten nicht als „Duplikate“ betrachtet werden und demselben Auslöser hinzugefügt werden können.



Abhängigkeiten ersetzen – entfernt alle vorhandenen Abhängigkeiten aus dem Auslöser und ersetzt sie durch die angegebenen Abhängigkeiten.

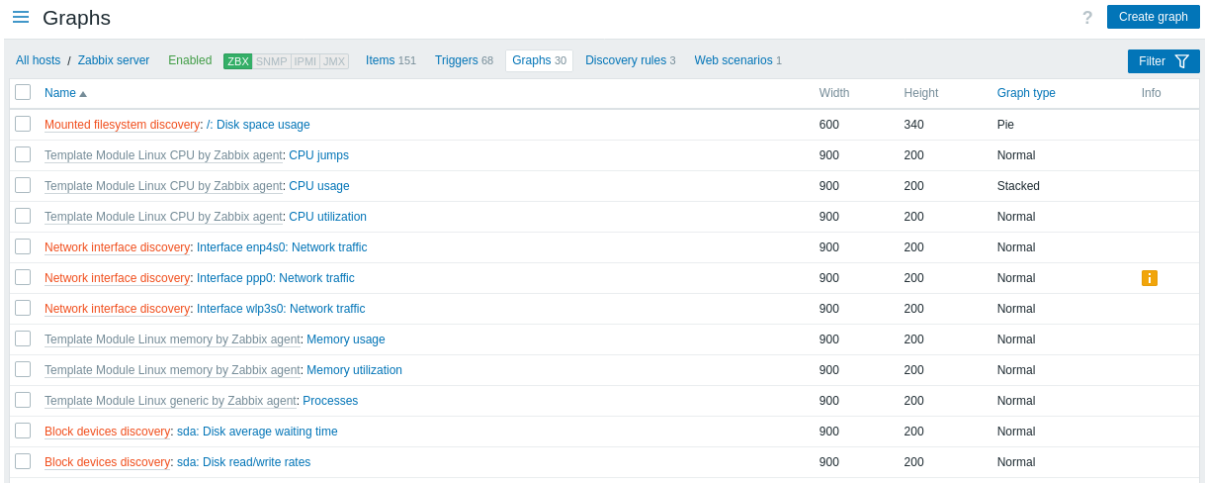
Klicken Sie auf *Aktualisieren*, um die Änderungen anzuwenden.

Diagramme

Übersicht

Die Liste der benutzerdefinierten Graphen für einen Host kann unter *Datensammlung* > *Hosts* aufgerufen werden, indem Sie beim entsprechenden Host auf *Graphen* klicken.

Eine Liste der vorhandenen Graphen wird angezeigt.



Angezeigte Daten:

Spalte	Beschreibung
Host	Host des Graphen. Durch Klicken auf den Hostnamen wird das Host-Konfigurationsformular geöffnet. Diese Spalte wird nur angezeigt, wenn im Filter mehrere Hosts oder keine Hosts ausgewählt sind.
Name	Name des benutzerdefinierten Graphen, der als blauer Link zu den Graphendetails angezeigt wird. Durch Klicken auf den Link mit dem Graphennamen wird das Graphen-Konfigurationsformular geöffnet. Wenn der Host-Graph zu einer Vorlage gehört, wird der Name der Vorlage vor dem Graphennamen als grauer Link angezeigt. Durch Klicken auf den Vorlagenlink wird die Graphenliste auf Vorlagenebene geöffnet. Wenn der Graph aus einem Graph-Prototyp erstellt wurde, wird seinem Namen der Name der Low-Level-Discovery-Regel in Orange vorangestellt. Durch Klicken auf den Namen der Discovery-Regel wird die Liste der Graph-Prototypen geöffnet.
Breite	Die Breite des Graphen wird angezeigt.
Höhe	Die Höhe des Graphen wird angezeigt.
Graphentyp	Der Graphentyp wird angezeigt – *Normal*, *Gestapelt*, *Kreis* oder *Explodiert*.
Info	Wenn der Graph ordnungsgemäß funktioniert, wird in dieser Spalte kein Symbol angezeigt. Bei Fehlern wird ein quadratisches Symbol mit dem Buchstaben „i“ angezeigt. Bewegen Sie den Mauszeiger über das Symbol, um eine QuickInfo mit der Fehlerbeschreibung anzuzeigen.

Um einen neuen Graphen zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Graph erstellen*.

Optionen für die Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen für die Massenbearbeitung:

- *Kopieren* - die Diagramme auf andere Hosts oder Vorlagen kopieren
- *Löschen* - die Diagramme löschen

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Diagrammen und klicken Sie dann auf die gewünschte Schaltfläche.

Filter verwenden

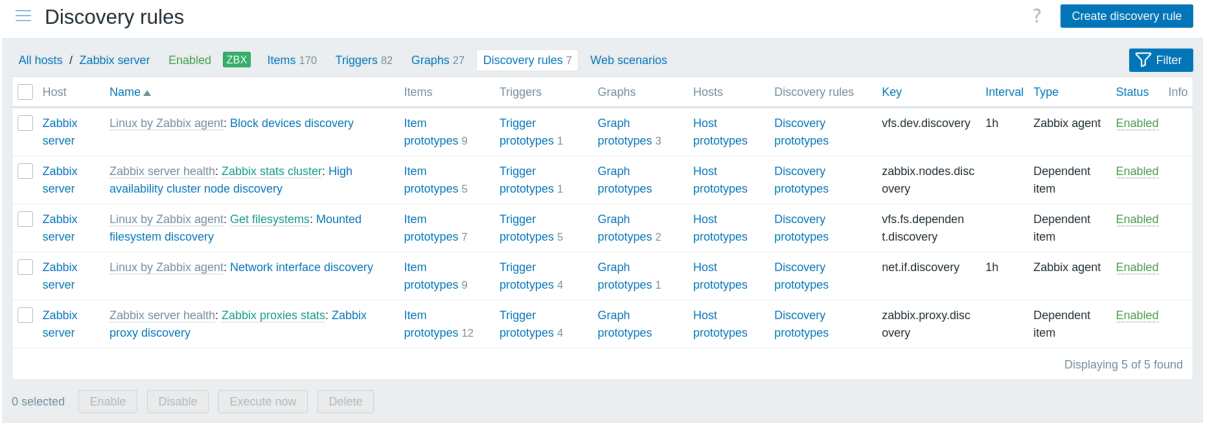
Sie können Graphen nach Host-Gruppe und Host filtern. Für eine bessere Suchleistung werden Daten mit nicht aufgelösten Makros durchsucht.

Erkennungsregeln

Übersicht

Die Liste der Low-Level-Discovery-Regeln für einen Host kann unter *Datensammlung* > *Hosts* aufgerufen werden, indem Sie bei dem entsprechenden Host auf *Discovery* klicken.

Eine Liste der vorhandenen Low-Level-Discovery-Regeln wird angezeigt. Es ist auch möglich, alle Discovery-Regeln unabhängig vom Host oder alle Discovery-Regeln einer bestimmten Hostgruppe anzuzeigen, indem Sie die Filtereinstellungen ändern.



Angezeigte Daten:

Spalte	Beschreibung
Host	Der sichtbare Hostname wird angezeigt. Durch Klicken auf den Hostnamen wird das Host-Konfigurationsformular geöffnet. Wenn kein sichtbarer Hostname vorhanden ist, wird der technische Hostname angezeigt.
Name	Der Name der Regel wird als blauer Link angezeigt. Durch Klicken auf den Regelnamen wird das Konfigurationsformular der Low-Level-Discovery-Regel geöffnet. Wenn die Discovery-Regel zu einer Vorlage gehört, wird der Name der Vorlage vor dem Regelnamen als grauer Link angezeigt. Durch Klicken auf den Vorlagenlink wird die Regelliste auf Vorlagenebene geöffnet.
Datenpunkte	Ein Link zur Liste der Datenpunkt-Prototypen wird angezeigt. Die Anzahl der vorhandenen Datenpunkt-Prototypen wird grau angezeigt.
Auslöser	Ein Link zur Liste der Auslöser-Prototypen wird angezeigt. Die Anzahl der vorhandenen Auslöser-Prototypen wird grau angezeigt.
Graphen	Ein Link zur Liste der Graph-Prototypen wird angezeigt. Die Anzahl der vorhandenen Graph-Prototypen wird grau angezeigt.
Hosts	Ein Link zur Liste der Host-Prototypen wird angezeigt. Die Anzahl der vorhandenen Host-Prototypen wird grau angezeigt.
Schlüssel	Der für die Discovery verwendete Datenpunktschlüssel wird angezeigt.
Intervall	Die Häufigkeit der Durchführung der Discovery wird angezeigt. *Hinweis*: Die Discovery kann auch sofort durchgeführt werden, indem Sie unterhalb der Liste auf die Schaltfläche *Jetzt ausführen* klicken.
Typ	Der für die Discovery verwendete Datenpunkttyp wird angezeigt (Zabbix agent, SNMP agent usw.).
Status	Der Status der Discovery-Regel wird angezeigt – *Aktiviert*, *Deaktiviert* oder *Nicht unterstützt*. Durch Klicken auf den Status können Sie ihn ändern – von „Aktiviert“ zu „Deaktiviert“ (und umgekehrt) sowie von „Nicht unterstützt“ zu „Deaktiviert“ (und umgekehrt).
Info	Wenn alles in Ordnung ist, wird in dieser Spalte kein Symbol angezeigt. Bei Fehlern wird ein quadratisches Symbol mit dem Buchstaben „i“ angezeigt. Bewegen Sie den Mauszeiger über das Symbol, um eine QuickInfo mit der Fehlerbeschreibung anzuzeigen.

Um eine neue Low-Level-Discovery-Regel zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Discovery-Regel erstellen*.

Optionen für die Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen für die Massenbearbeitung:

- *Aktivieren* - den Status der Low-Level-Discovery-Regel auf *Aktiviert* ändern.
- *Deaktivieren* - den Status der Low-Level-Discovery-Regel auf *Deaktiviert* ändern.
- *Jetzt ausführen* - die Discovery basierend auf den Discovery-Regeln sofort ausführen. Siehe weitere Details. Beachten Sie, dass bei der sofortigen Ausführung der Discovery der Konfigurations-Cache nicht aktualisiert wird und das Ergebnis daher keine sehr aktuellen Änderungen an der Konfiguration der Discovery-Regel widerspiegelt.

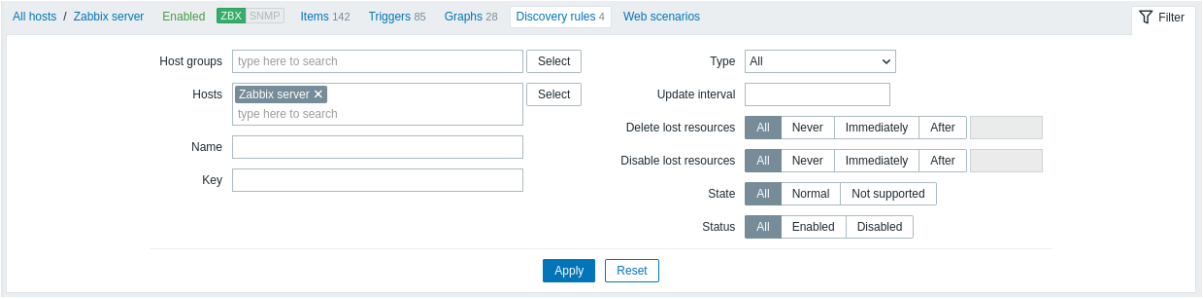
- *Löschen* - die Low-Level-Discovery-Regeln löschen.

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Discovery-Regeln und klicken Sie dann auf die gewünschte Schaltfläche.

Verwendung des Filters

Sie können den Filter verwenden, um nur die Erkennungsregeln anzuzeigen, die Sie interessieren. Für eine bessere Suchleistung werden Daten mit nicht aufgelösten Makros durchsucht.

Der Link *Filter* ist oberhalb der Liste der Erkennungsregeln verfügbar. Wenn Sie darauf klicken, wird ein Filter angezeigt, mit dem Sie Erkennungsregeln nach Hostgruppe, Host, Name, Datenpunkt-Schlüssel, Datenpunkt-Typ und anderen Parametern filtern können.

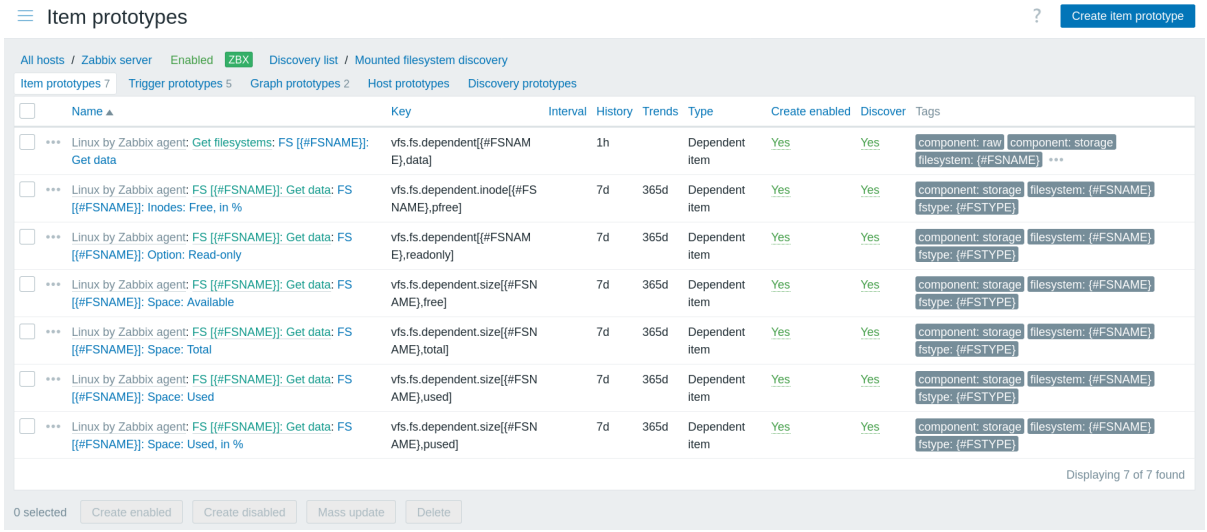


Parameter	Beschreibung
Host groups	Nach einer oder mehreren Hostgruppen filtern. Durch die Angabe einer übergeordneten Hostgruppe werden implizit alle verschachtelten Hostgruppen ausgewählt.
Hosts	Nach einem oder mehreren Hosts filtern.
Name	Nach dem Namen der Erkennungsregel filtern.
Key	Nach dem Schlüssel des Erkennungs-Datenpunkts filtern.
Type	Nach dem Typ des Erkennungs-Datenpunkts filtern.
Update interval	Nach dem Aktualisierungsintervall filtern. Für Zabbix trapper und abhängige Datenpunkte nicht verfügbar.
Delete lost resources	Nach dem Zeitraum *Delete lost resources* filtern.
Disable lost resources	Nach dem Zeitraum *Disable lost resources* filtern.
SNMP OID	Nach der SNMP OID filtern. Nur verfügbar, wenn *SNMP agent* als Typ ausgewählt ist.
State	Nach dem Zustand der Erkennungsregel filtern (All/Normal/Not supported).
Status	Nach dem Status der Erkennungsregel filtern (All/Enabled/Disabled).

Datenpunkt-Prototypen

Übersicht

In diesem Abschnitt werden die Datenpunkt-Prototypen einer Low-Level-Discovery-Regel auf dem Host angezeigt. Datenpunkt-Prototypen bilden die Grundlage für echte Datenpunkte auf dem Host, die während der Low-Level-Discovery erstellt werden.



Angezeigte Daten:

Column	Description
Name	Name des Datenpunkt-Prototyps, als blauer Link angezeigt. Durch Klicken auf den Namen wird das Konfigurationsformular für den Datenpunkt-Prototyp geöffnet. Wenn der Datenpunkt-Prototyp zu einer Vorlage gehört, wird der Name der Vorlage vor dem Namen der Regel als grauer Link angezeigt. Durch Klicken auf den Vorlagenlink wird die Liste der Datenpunkt-Prototypen auf Vorlagenebene geöffnet. Wenn der Datenpunkt-Prototyp vom Typ „Abhängiger Datenpunkt“ ist, wird der Name des Master-Datenpunkts vor dem Namen des Datenpunkt-Prototyps in Grün angezeigt.
Key	Der Schlüssel des Datenpunkt-Prototyps wird angezeigt.
Interval	Die Häufigkeit der Prüfung wird angezeigt.
History	Die Anzahl der Tage, für die der Datenpunktverlauf aufbewahrt wird, wird angezeigt.
Trends	Die Anzahl der Tage, für die der Verlauf der Datenpunkt-Trends aufbewahrt wird, wird angezeigt.
Type	Der Typ des Datenpunkt-Prototyps wird angezeigt (Zabbix agent, SNMP agent, einfache Prüfung usw.).
Create enabled	Erstellung des Datenpunkts auf Grundlage dieses Prototyps: **Yes** – aktiviert **No** – deaktiviert. Durch Klicken auf „Yes“ oder „No“ können Sie zwischen den Optionen wechseln.
Discover	Erkennung des Datenpunkts auf Grundlage dieses Prototyps: **Yes** – erkennen **No** – nicht erkennen. Durch Klicken auf „Yes“ oder „No“ können Sie zwischen den Optionen wechseln.
Tags	Die Tags des Datenpunkt-Prototyps werden angezeigt.

Um einen neuen Datenpunkt-Prototyp zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Create item prototype*.

Optionen für die Massenbearbeitung

Die Schaltflächen unter der Liste bieten einige Optionen für die Massenbearbeitung:

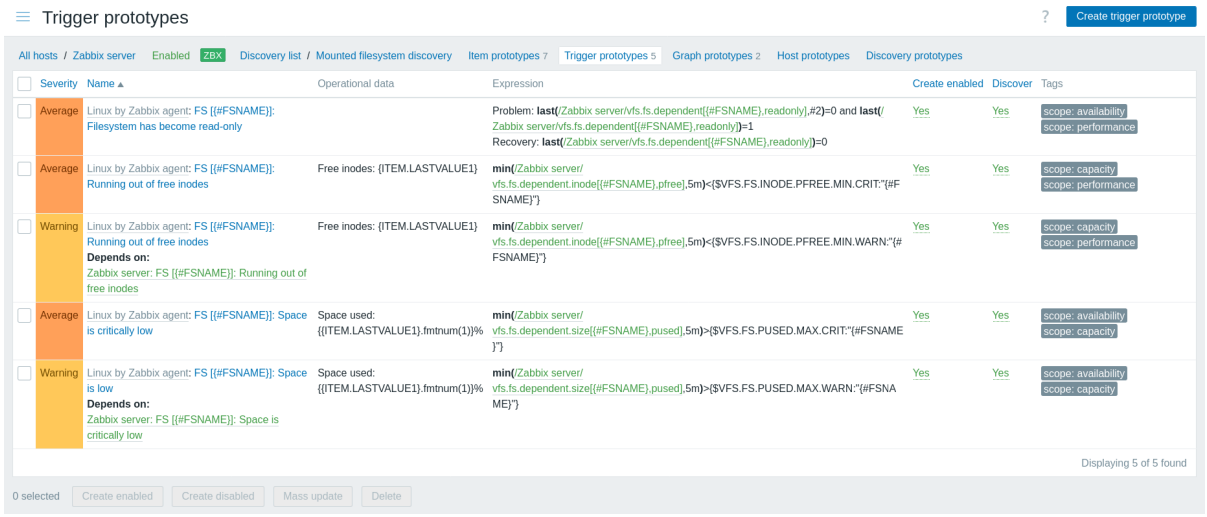
- *Aktiviert erstellen* - diese Datenpunkte als *Aktiviert* erstellen
- *Deaktiviert erstellen* - diese Datenpunkte als *Deaktiviert* erstellen
- *Massenaktualisierung* - diese Datenpunktvorlagen massenhaft aktualisieren
- *Löschen* - diese Datenpunktvorlagen löschen

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Datenpunktvorlagen und klicken Sie dann auf die gewünschte Schaltfläche.

Auslöser-Prototypen

Übersicht

In diesem Abschnitt werden die Auslöser-Prototypen einer Low-Level-Discovery-Regel auf dem Host angezeigt. Auslöser-Prototypen bilden die Grundlage für echte Auslöser auf Hosts, die während der Low-Level-Discovery erstellt werden.



Angezeigte Daten:

Column	Description
Name	Name des Auslöser-Prototyps, als blauer Link angezeigt. Durch Klicken auf den Namen wird das Konfigurationsformular für den Auslöser-Prototyp geöffnet. Wenn der Auslöser-Prototyp zu einer verknüpften Vorlage gehört, wird der Vorlagenname vor dem Namen des Auslösers als grauer Link angezeigt. Durch Klicken auf den Vorlagenlink wird die Liste der Auslöser-Prototypen auf der Ebene der verknüpften Vorlage geöffnet.
Operational data	Das Format der Betriebsdaten des Auslösers wird angezeigt. Es kann beliebige Zeichenfolgen und Makros enthalten, die unter *Monitoring > Problems* dynamisch aufgelöst werden.
Create enabled	Erstellung des Auslösers auf Grundlage dieses Prototyps als: **Yes** – aktiviert **No** – deaktiviert. Durch Klicken auf **Yes** bzw. **No** können Sie zwischen den Optionen wechseln.
Discover	Erkennung des Auslösers auf Grundlage dieses Prototyps: **Yes** – erkennen **No** – nicht erkennen. Durch Klicken auf **Yes** bzw. **No** können Sie zwischen den Optionen wechseln.
Tags	Tags des Auslöser-Prototyps werden angezeigt.

Um einen neuen Auslöser-Prototyp zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Create trigger prototype*.

Optionen für die Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen für die Massenbearbeitung:

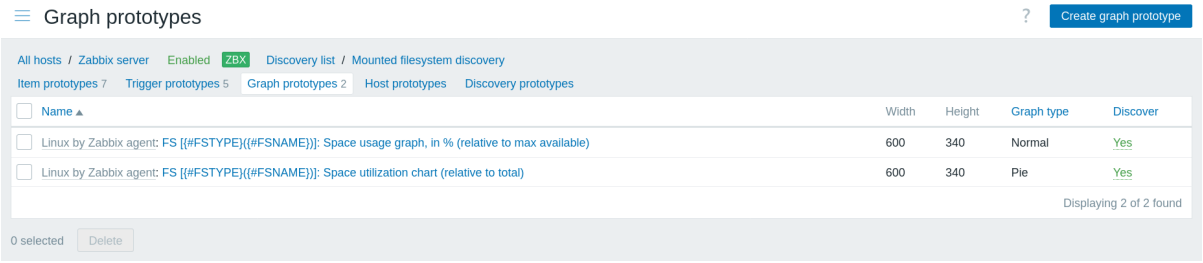
- *Aktiviert erstellen* - diese Auslöser als *Aktiviert* erstellen
- *Deaktiviert erstellen* - diese Auslöser als *Deaktiviert* erstellen
- *Massenaktualisierung* - diese Auslöservorlagen massenhaft aktualisieren
- *Löschen* - diese Auslöservorlagen löschen

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Auslöservorlagen und klicken Sie dann auf die gewünschte Schaltfläche.

Diagrammvorlagen

Übersicht

In diesem Abschnitt werden die Diagrammprototypen einer Low-Level-Discovery-Regel auf dem Host angezeigt. Diagrammprototypen bilden die Grundlage für echte Diagramme des Hosts, die während der Low-Level-Discovery erstellt werden.



Angezeigte Daten:

Column	Description
Name	Name des Diagrammprototyps, angezeigt als blauer Link. Durch Klicken auf den Namen wird das Konfigurationsformular für den Diagrammprototyp geöffnet. Wenn der Diagrammprototyp zu einer verknüpften Vorlage gehört, wird der Vorlagenname vor dem Diagrammnamen als grauer Link angezeigt. Durch Klicken auf den Vorlagenlink wird die Liste der Diagrammprototypen auf der Ebene der verknüpften Vorlage geöffnet.
Width	Die Breite des Diagrammprototyps wird angezeigt.
Height	Die Höhe des Diagrammprototyps wird angezeigt.
Type	Der Typ des Diagrammprototyps wird angezeigt – *Normal*, *Gestapelt*, *Kreis* oder *Explodiert*.
Discover	Das Diagramm basierend auf diesem Prototyp erkennen: **Yes** – erkennen **No** – nicht erkennen. Durch Klicken auf *Yes* oder *No* können Sie zwischen den Optionen wechseln.

Um einen neuen Diagrammprototyp zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Diagrammprototyp erstellen*.

Optionen für die Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen für die Massenbearbeitung:

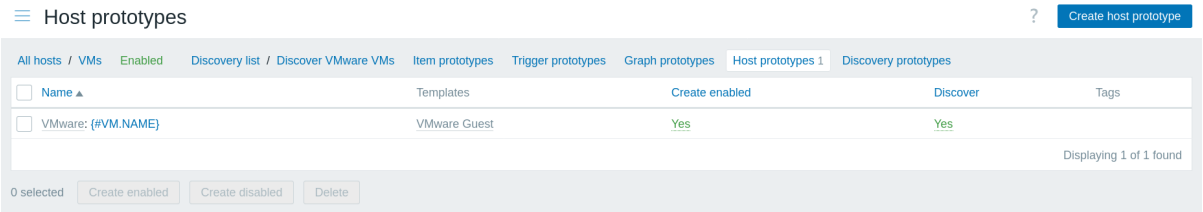
- *Löschen* - diese Graphprototypen löschen

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Graphprototypen und klicken Sie dann auf die gewünschte Schaltfläche.

Host-Vorlagen

Übersicht

In diesem Abschnitt werden die Host-Prototypen einer Low-Level-Discovery-Regel auf dem Host angezeigt. Host-Prototypen bilden die Grundlage für reale Hosts, die während der Low-Level-Discovery erstellt werden.



Angezeigte Daten:

Spalte	Beschreibung
Name	Name des Host-Prototyps, als blauer Link angezeigt. Durch Klicken auf den Namen wird das Konfigurationsformular des Host-Prototyps geöffnet. Wenn der Host-Prototyp zu einer verknüpften Vorlage gehört, wird der Vorlagenname vor dem Hostnamen als grauer Link angezeigt. Durch Klicken auf den Vorlagenlink wird die Liste der Host-Prototypen auf der Ebene der verknüpften Vorlage geöffnet.
Vorlagen	Die Vorlagen des Host-Prototyps werden angezeigt.
Erstellung aktiviert	Den Host basierend auf diesem Prototyp erstellen als: **Ja** – aktiviert **Nein** – deaktiviert. Sie können durch Klicken zwischen „Ja“ und „Nein“ wechseln.

Spalte	Beschreibung
Ermitteln	Den Host basierend auf diesem Prototyp ermitteln: **Ja** – ermitteln **Nein** – nicht ermitteln. Sie können durch Klicken zwischen „Ja“ und „Nein“ wechseln.
Tags	Die Tags des Host-Prototyps werden angezeigt.

Um einen neuen Host-Prototyp zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Host-Prototyp erstellen*.

Optionen zur Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

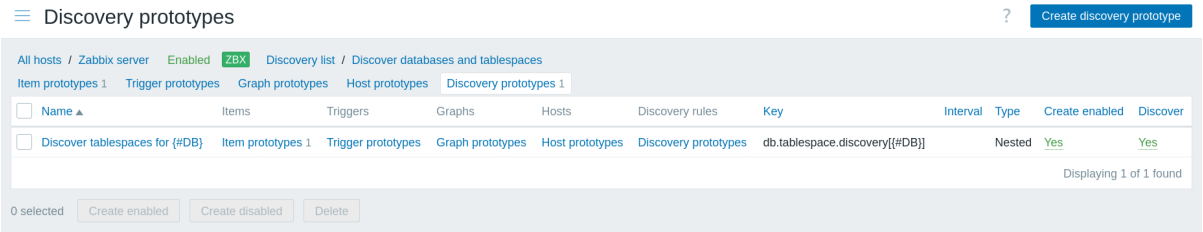
- *Aktiviert erstellen* - diese Hosts als *Aktiviert* erstellen
- *Deaktiviert erstellen* - diese Hosts als *Deaktiviert* erstellen
- *Löschen* - diese Host-Prototypen löschen

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Host-Prototypen und klicken Sie dann auf die gewünschte Schaltfläche.

Erkennungsprototypen

Übersicht

In diesem Abschnitt werden die Discovery-Prototypen einer Low-Level-Discovery-Regel auf dem Host angezeigt. Discovery-Prototypen sind verschachtelte Discovery-Regeln der übergeordneten Discovery-Regel.



Angezeigte Daten:

Spalte	Beschreibung
Name	Der Name des Discovery-Prototyps wird als blauer Link angezeigt. Durch Klicken auf den Namen wird das Konfigurationsformular des Discovery-Prototyps geöffnet. Wenn der Discovery-Prototyp zu einer verknüpften Vorlage gehört, wird der Vorlagenname vor dem Hostnamen als grauer Link angezeigt. Durch Klicken auf den Vorlagenlink wird die Liste der Discovery-Prototypen auf der Ebene der verknüpften Vorlage geöffnet.
Datenpunkte	Ein Link zur Liste der Datenpunkt-Prototypen wird angezeigt. Die Anzahl der vorhandenen Datenpunkt-Prototypen wird grau angezeigt.
Auslöser	Ein Link zur Liste der Auslöser-Prototypen wird angezeigt. Die Anzahl der vorhandenen Auslöser-Prototypen wird grau angezeigt.
Graphen	Ein Link zur Liste der Graph-Prototypen wird angezeigt. Die Anzahl der vorhandenen Graph-Prototypen wird grau angezeigt.
Hosts	Ein Link zur Liste der Host-Prototypen wird angezeigt. Die Anzahl der vorhandenen Host-Prototypen wird grau angezeigt.
Discovery-Regeln	Ein Link zur Liste der Discovery-Prototypen wird angezeigt. Die Anzahl der vorhandenen Discovery-Prototypen wird grau angezeigt.
Schlüssel	Der für die Discovery verwendete Datenpunktschlüssel wird angezeigt.
Intervall	Die Häufigkeit der Durchführung der Discovery wird angezeigt. *Hinweis*: Die Discovery kann auch sofort durchgeführt werden, indem Sie unterhalb der Liste auf die Schaltfläche *Jetzt ausführen* klicken.
Typ	Der für die Discovery verwendete Datenpunkttyp wird angezeigt (Zabbix agent, SNMP agent usw.).
Erstellung aktiviert	Der Status der Erstellung des Discovery-Prototyps wird angezeigt – Erstellung aktiviert (*Ja*) oder Erstellung deaktiviert (*Nein*). Durch Klicken auf den Status können Sie ihn ändern – von Ja zu Nein (und zurück).

Spalte	Beschreibung
Ermitteln	Der Discovery-Status des Discovery-Prototyps wird angezeigt – ermitteln (*Ja*) oder nicht ermitteln (*Nein*). Durch Klicken auf den Status können Sie ihn ändern – von Ja zu Nein (und zurück).

Um einen neuen Discovery-Prototyp zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Discovery-Prototyp erstellen*.

Optionen für die Massenbearbeitung

Die Schaltflächen unter der Liste bieten einige Optionen für die Massenbearbeitung:

- *Aktiviert erstellen* - diese Erkennungsprototypen als *Aktiviert* erstellen
- *Deaktiviert erstellen* - diese Erkennungsprototypen als *Deaktiviert* erstellen
- *Löschen* - diese Erkennungsprototypen löschen

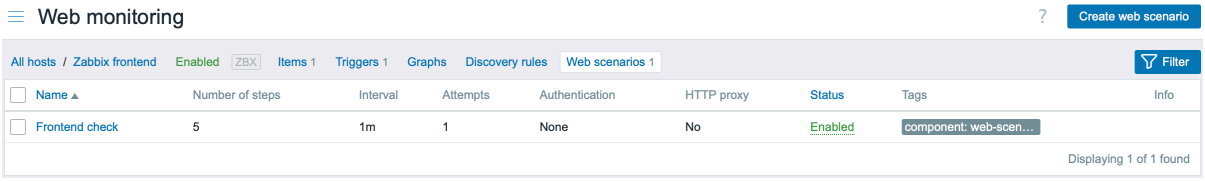
Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Erkennungsprototypen und klicken Sie dann auf die gewünschte Schaltfläche.

Web-Szenarien

Übersicht

Die Liste der Web-Szenarien für einen Host kann unter *Datensammlung* > *Hosts* aufgerufen werden, indem Sie beim entsprechenden Host auf *Web* klicken.

Eine Liste der vorhandenen Web-Szenarien wird angezeigt.



Angezeigte Daten:

Spalte	Beschreibung
Name	Name des Web-Szenarios. Durch Klicken auf den Namen des Web-Szenarios wird das Konfigurationsformular für Web-Szenarien geöffnet. Wenn das Web-Szenario des Hosts zu einer Vorlage gehört, wird der Name der Vorlage vor dem Namen des Web-Szenarios als grauer Link angezeigt. Durch Klicken auf den Vorlagenlink wird die Liste der Web-Szenarien auf Vorlagenebene geöffnet.
Anzahl der Schritte	Die Anzahl der Schritte, die das Szenario enthält.
Aktualisierungsintervall	Wie oft das Szenario ausgeführt wird.
Versuche	Die Anzahl der Versuche zur Ausführung der Schritte des Web-Szenarios.
Authentifizierung	Die Authentifizierungsmethode wird angezeigt: Basic, NTLM, Kerberos, Digest oder Keine.
HTTP-Proxy	Zeigt den HTTP-Proxy oder „Nein“ an, wenn kein Proxy verwendet wird.
Status	Der Status des Web-Szenarios wird angezeigt: *Aktiviert* oder *Deaktiviert*. Durch Klicken auf den Status können Sie ihn ändern.
Tags	Die Tags des Web-Szenarios werden angezeigt. Bis zu drei Tags (Name-Wert-Paare) können angezeigt werden. Wenn weitere Tags vorhanden sind, wird ein Link „...“ angezeigt, über den alle Tags beim Überfahren mit dem Mauszeiger angezeigt werden können.
Info	Wenn alles ordnungsgemäß funktioniert, wird in dieser Spalte kein Symbol angezeigt. Bei Fehlern wird ein quadratisches Symbol mit dem Buchstaben „i“ angezeigt. Bewegen Sie den Mauszeiger über das Symbol, um eine Kurzinfo mit der Fehlerbeschreibung anzuzeigen.

Um ein neues Web-Szenario zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Web-Szenario erstellen*.

Optionen für die Massenbearbeitung

Die Schaltflächen unter der Liste bieten einige Optionen für die Massenbearbeitung:

- *Aktivieren* - den Status des Szenarios auf *Aktiviert* ändern
- *Deaktivieren* - den Status des Szenarios auf *Deaktiviert* ändern

- *Verlauf und Trends löschen* - Verlauf und Trenddaten für die Szenarien löschen
- *Löschen* - die Webszenarien löschen

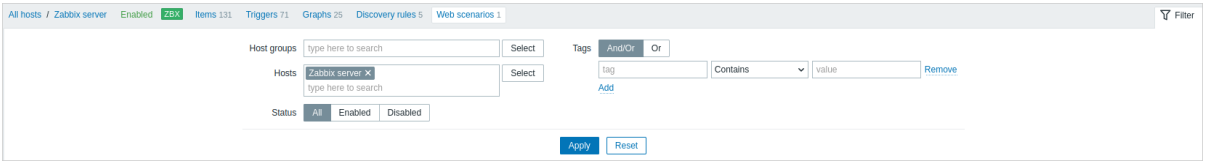
Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Webszenarien und klicken Sie dann auf die gewünschte Schaltfläche.

Verwendung des Filters

Sie können den Filter verwenden, um nur die Szenarien anzuzeigen, die Sie interessieren.

Für eine bessere Suchleistung werden die Daten mit nicht aufgelösten Makros durchsucht.

Der Link *Filter* ist oberhalb der Liste der Webszenarien verfügbar. Wenn Sie darauf klicken, wird ein Filter geöffnet, in dem Sie die gewünschten Filterkriterien angeben können.



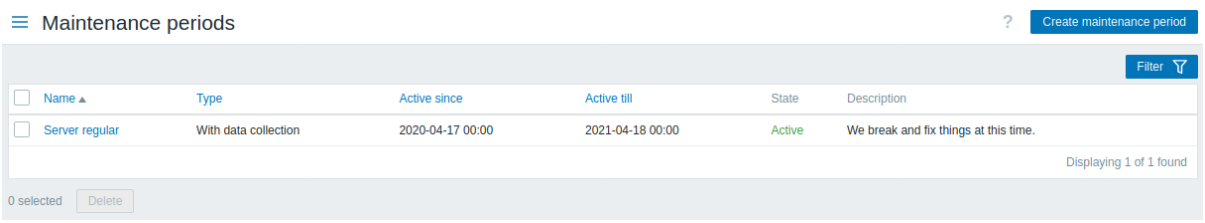
Parameter	Beschreibung
Host groups	Nach einer oder mehreren Hostgruppen filtern. Wenn eine übergeordnete Hostgruppe angegeben wird, werden alle verschachtelten Hostgruppen implizit ausgewählt.
Hosts	Nach einem oder mehreren Hosts filtern. Wenn oben bereits Hostgruppen ausgewählt wurden, ist die Host-Auswahl auf diese Gruppen beschränkt.
Status	Nach dem Status des Webszenarios filtern.
Tags	Nach Tag-Namen und -Wert des Webszenarios filtern. Es ist möglich, bestimmte Tags und Tag-Werte einzuschließen sowie auszuschließen. Es können mehrere Bedingungen festgelegt werden. Die Übereinstimmung des Tag-Namens ist immer groß-/kleinschreibungssensitiv. Für jede Bedingung stehen mehrere Operatoren zur Verfügung: **Exists** - die angegebenen Tag-Namen einschließen; **Equals** - die angegebenen Tag-Namen und -Werte einschließen (groß-/kleinschreibungssensitiv); **Contains** - die angegebenen Tag-Namen einschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv); **Does not exist** - die angegebenen Tag-Namen ausschließen; **Does not equal** - die angegebenen Tag-Namen und -Werte ausschließen (groß-/kleinschreibungssensitiv); **Does not contain** - die angegebenen Tag-Namen ausschließen, deren Tag-Werte die eingegebene Zeichenfolge enthalten (Teilzeichenfolgenabgleich, nicht groß-/kleinschreibungssensitiv). Es gibt zwei Berechnungstypen für Bedingungen: **And/Or** - alle Bedingungen müssen erfüllt sein, Bedingungen mit demselben Tag-Namen werden durch die Oder-Bedingung gruppiert; **Or** - es genügt, wenn eine Bedingung erfüllt ist.

Wartung

Übersicht

Im Bereich *Datenerfassung* > *Wartung* können Benutzer Wartungszeiträume für Hosts konfigurieren und verwalten.

Eine Liste der vorhandenen Wartungszeiträume mit ihren Details wird angezeigt.



Angezeigte Daten:

Spalte	Beschreibung
Name	Name des Wartungszeitraums. Durch Klicken auf den Namen des Wartungszeitraums wird das Konfigurationsformular für den Wartungszeitraum geöffnet.
Typ	Der Wartungstyp wird angezeigt: *Mit Datenerfassung* oder *Keine Datenerfassung*
Aktiv seit	Datum und Uhrzeit, ab denen die Ausführung von Wartungszeiträumen aktiv wird. Hinweis: Diese Zeit aktiviert keinen Wartungszeitraum; Wartungszeiträume müssen separat festgelegt werden.
Aktiv bis	Datum und Uhrzeit, bis zu denen die Ausführung von Wartungszeiträumen aktiv bleibt.
Status	Der Status des Wartungszeitraums: **Anstehend** – wird bald aktiv **Aktiv** – ist aktiv **Abgelaufen** – ist nicht mehr aktiv
Beschreibung	Die Beschreibung des Wartungszeitraums wird angezeigt.

Um einen neuen Wartungszeitraum zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Wartungszeitraum erstellen*.

Optionen für die Massenbearbeitung

Eine Schaltfläche unterhalb der Liste bietet eine Option zur Massenbearbeitung:

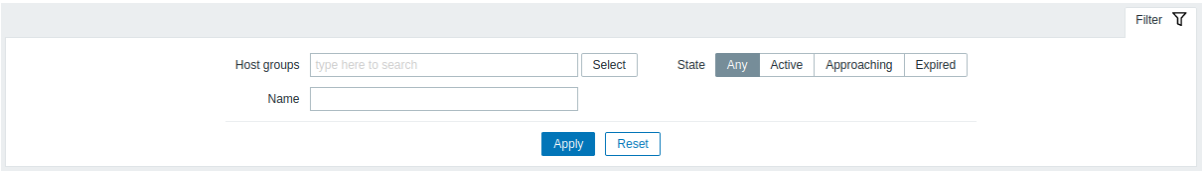
- *Löschen* - die Wartungszeiträume löschen

Um diese Option zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Wartungszeiträumen und klicken Sie auf *Löschen*.

Verwenden des Filters

Sie können den Filter verwenden, um nur die Wartungszeiträume anzuzeigen, die Sie interessieren. Für eine bessere Suchleistung werden die Daten mit nicht aufgelösten Makros durchsucht.

Der Link *Filter* ist oberhalb der Liste der Wartungszeiträume verfügbar. Wenn Sie darauf klicken, wird ein Filter angezeigt, mit dem Sie Wartungszeiträume nach Hostgruppe, Name und Status filtern können.



Berechnung von Warteschlangen während der Wartung

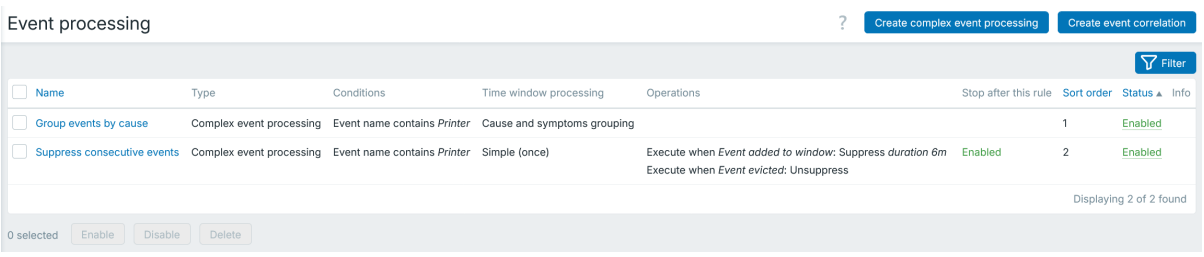
Attention:
Der Zabbix Proxy ist sich von Wartungszeiträumen nicht bewusst; siehe Berechnung von Warteschlangen während der Wartung für Details.

Event processing

Overview

In the *Data collection* > *Event processing* section users can configure and maintain:

- Complex event processing rules (CEP rules)
- Global event correlation rules (deprecated)



Displayed data:

Column	Description
Name	Name of the rule. Clicking the name opens the rule configuration form - for CEP rule or global correlation rule respectively.
Type	Type of the rule - *Complex event processing* or *Correlation*.
Conditions	Rule conditions are displayed.
Time window processing	Time window type is displayed.
Operations	Rule operations are displayed.
Stop after this rule	Stop execution after this rule - *Enabled* or *Disabled*.
Sort order	Sort order of the CEP rule.
Status	Rule status is displayed - *Enabled* or *Disabled*. By clicking the status you can change it.
Info	If everything is working correctly, no icon is displayed in this column. In case of errors, a square icon with the letter "i" is displayed. Hover over the icon to see a tooltip with the error description.

To configure a new CEP rule, click the *Create complex event processing* button in the upper-right corner.

To configure a new correlation rule, click the *Create event correlation* button in the upper-right corner.

Mass editing options

Buttons below the list offer some mass-editing options:

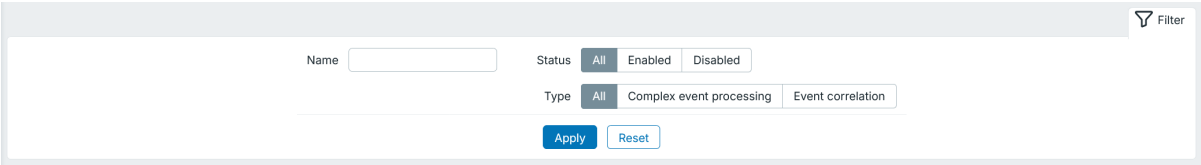
- *Enable* - change the rule status to *Enabled*
- *Disable* - change the rule status to *Disabled*
- *Delete* - delete the rules

To use these options, mark the checkboxes before the respective rules, then click the required button.

Using filter

You can use the filter to display only the rules you are interested in. For better search performance, data is searched with macros unresolved.

The *Filter* link is available above the list of CEP/correlation rules. If you click it, a filter becomes available where you can filter rules by name and status.

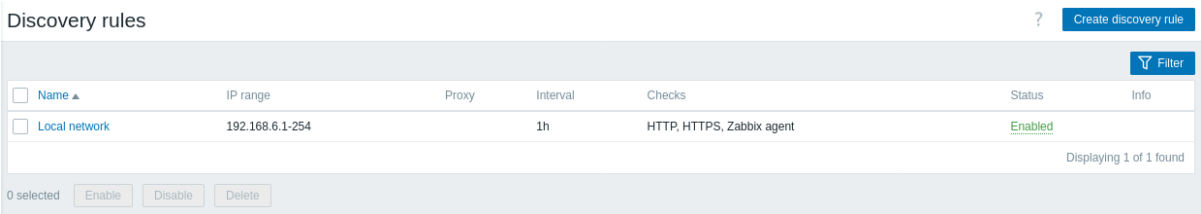


Erkennung

Übersicht

Im Bereich *Datensammlung* > *Erkennung* können Benutzer Erkennungsregeln konfigurieren und verwalten.

Eine Liste der vorhandenen Erkennungsregeln mit ihren Details wird angezeigt.



Angezeigte Daten:

Spalte	Beschreibung
Name	Name der Erkennungsregel. Durch Klicken auf den Namen der Erkennungsregel wird das Konfigurationsformular für Erkennungsregeln geöffnet.
IP-Bereich	Der Bereich der IP-Adressen, der für die Netzwerkerkennung verwendet werden soll, wird angezeigt.
Proxy	Der Name des Proxy wird angezeigt, wenn die Erkennung durch den Proxy durchgeführt wird.

Spalte	Beschreibung
Intervall	Die Häufigkeit, mit der die Erkennung durchgeführt wird, wird angezeigt.
Prüfungen	Die für die Erkennung verwendeten Prüfungstypen werden angezeigt.
Status	Der Status der Erkennungsregel wird angezeigt – *Aktiviert* oder *Deaktiviert*. Durch Klicken auf den Status können Sie ihn ändern.
Info	Wenn alles ordnungsgemäß funktioniert, wird in dieser Spalte nichts angezeigt. Bei Fehlern wird ein rotes Infosymbol mit dem Buchstaben „i“ angezeigt. Bewegen Sie den Mauszeiger über das Symbol, um eine QuickInfo mit der Fehlerbeschreibung anzuzeigen.

Um eine neue Erkennungsregel zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Erkennungsregel erstellen*.

Optionen für die Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen für die Massenbearbeitung:

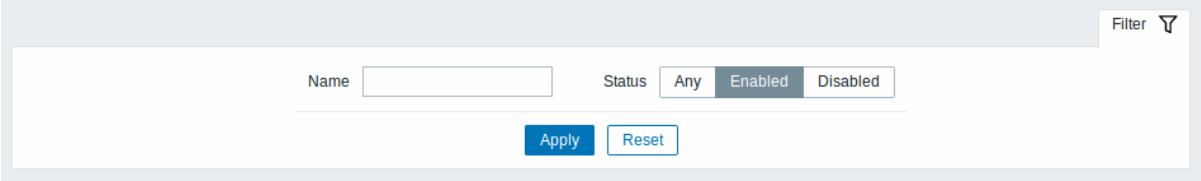
- *Aktivieren* - den Status der Erkennungsregel auf *Aktiviert* ändern
- *Deaktivieren* - den Status der Erkennungsregel auf *Deaktiviert* ändern
- *Löschen* - die Erkennungsregeln löschen

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Erkennungsregeln und klicken Sie dann auf die gewünschte Schaltfläche.

Filter verwenden

Sie können den Filter verwenden, um nur die Erkennungsregeln anzuzeigen, die Sie interessieren. Für eine bessere Suchleistung werden die Daten mit nicht aufgelösten Makros durchsucht.

Der Link *Filter* ist oberhalb der Liste der Erkennungsregeln verfügbar. Wenn Sie darauf klicken, wird ein Filter angezeigt, mit dem Sie Erkennungsregeln nach Name und Status filtern können.



Benachrichtigungen

Übersicht

Dieses Menü enthält Abschnitte, die mit der Konfiguration von Warnmeldungen in Zabbix zusammenhängen.

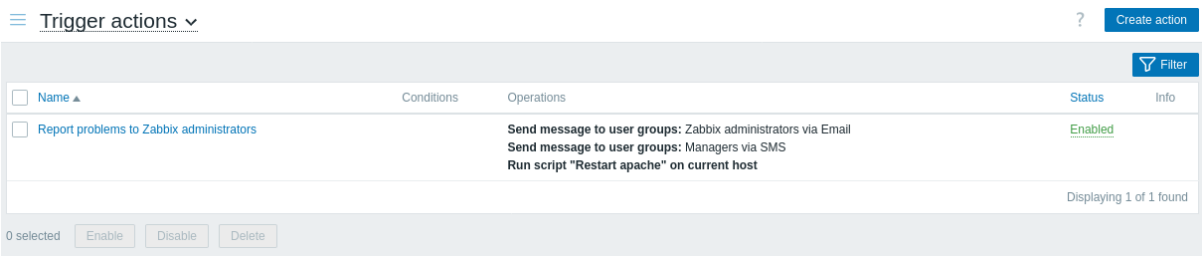
Aktionen

Übersicht

Im Abschnitt *Alarme* > *Aktionen* können Benutzer Aktionen konfigurieren und verwalten.

Die angezeigten Aktionen sind Aktionen, die der ausgewählten Ereignisquelle zugewiesen sind (Auslöser, Services, Discovery, automatische Registrierung, interne Aktionen).

Um zu einer anderen Ereignisquelle zu wechseln, klicken Sie im Menüabschnitt *Alarme* auf *Aktionen*. Sie können auch über das Titelauswahlfeld in der oberen linken Ecke zwischen den Quellen wechseln.



Angezeigte Daten:

Spalte	Beschreibung
Name	Name der Aktion. Durch Klicken auf den Aktionsnamen wird das Aktionskonfigurationsformular geöffnet.
Bedingungen	Die Aktionsbedingungen werden angezeigt.
Operationen	Die Aktionsoperationen werden angezeigt. Die Operationsliste zeigt außerdem den für die Benachrichtigung verwendeten Medientyp (E-Mail, SMS oder Skript) sowie den Vor- und Nachnamen (in Klammern nach dem Benutzernamen) eines Benachrichtigungsempfängers an. Eine Aktionsoperation kann je nach ausgewähltem Operationstyp entweder eine Benachrichtigung oder ein Remote-Befehl sein.
Status	Der Aktionsstatus wird angezeigt – *Aktiviert* oder *Deaktiviert*. Durch Klicken auf den Status können Sie ihn ändern. Weitere Informationen dazu, was geschieht, wenn eine Aktion während einer laufenden Eskalation deaktiviert wird, finden Sie im Abschnitt Eskalationen.
Info	Wenn alles ordnungsgemäß funktioniert, wird in dieser Spalte kein Symbol angezeigt. Bei Fehlern, beispielsweise fehlenden Aktionsoperationen oder -bedingungen nach einem Upgrade/einer Low-Level-Discovery, wird ein Warnsymbol angezeigt. Bewegen Sie den Mauszeiger über das Symbol, um eine QuickInfo mit der Fehlerbeschreibung anzuzeigen.

Um eine neue Aktion zu konfigurieren, klicken Sie in der oberen rechten Ecke auf die Schaltfläche *Aktion erstellen*.

Für Benutzer ohne Super-Admin-Rechte werden Aktionen entsprechend den Berechtigungseinstellungen angezeigt. Das bedeutet, dass ein Benutzer ohne Super-Admin-Rechte aufgrund bestimmter Berechtigungseinschränkungen in einigen Fällen nicht die vollständige Aktionsliste anzeigen kann. Eine Aktion wird einem Benutzer ohne Super-Admin-Rechte angezeigt, wenn die folgenden Bedingungen erfüllt sind:

- Der Benutzer verfügt in den Aktionsbedingungen über Lese-/Schreibzugriff auf Hostgruppen, Hosts, Vorlagen und Auslöser.
- Der Benutzer verfügt in den Aktionsoperationen, Wiederherstellungsoperationen und Aktualisierungsoperationen über Lese-/Schreibzugriff auf Hostgruppen, Hosts und Vorlagen.
- Der Benutzer verfügt in den Aktionsoperationen, Wiederherstellungsoperationen und Aktualisierungsoperationen über Lesezugriff auf Benutzergruppen und Benutzer.

Optionen für die Massenbearbeitung

Die Schaltflächen unter der Liste bieten einige Optionen für die Massenbearbeitung:

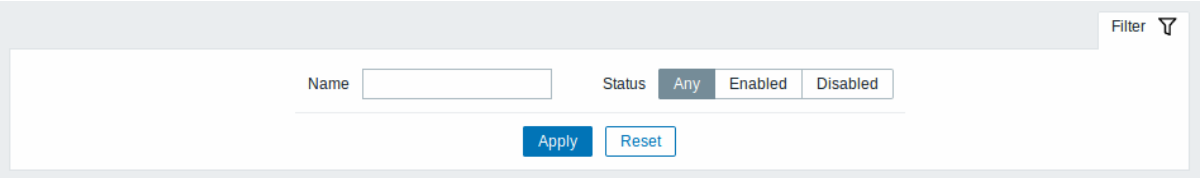
- *Aktivieren* - den Aktionsstatus auf *Aktiviert* ändern
- *Deaktivieren* - den Aktionsstatus auf *Deaktiviert* ändern
- *Löschen* - die Aktionen löschen

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Aktionen und klicken Sie dann auf die gewünschte Schaltfläche.

Filter verwenden

Sie können den Filter verwenden, um nur die Aktionen anzuzeigen, die Sie interessieren. Für eine bessere Suchleistung werden die Daten mit nicht aufgelösten Makros durchsucht.

Der Link *Filter* ist oberhalb der Aktionsliste verfügbar. Wenn Sie darauf klicken, wird ein Filter eingeblendet, mit dem Sie Aktionen nach Name und Status filtern können.



Medientypen

Übersicht

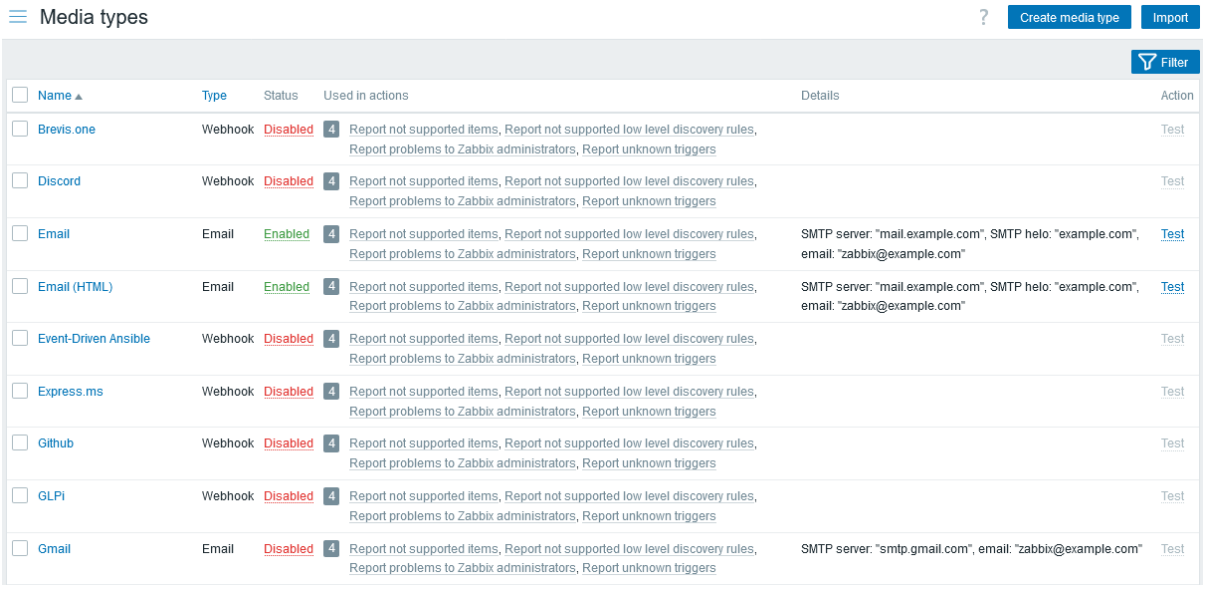
Im Abschnitt *Alarme* > *Medientypen* können Benutzer Informationen zu Medientypen konfigurieren und verwalten.

Dieses Menü ist nur für Benutzer des Benutzertyps SuperAdmin verfügbar.

Die Informationen zu einem Medientyp enthalten allgemeine Anweisungen zur Verwendung eines Mediums als Zustellkanal für Benachrichtigungen. Spezifische Details, z. B. die einzelnen E-Mail-Adressen, an die eine Benachrichtigung gesendet werden soll, werden bei den einzelnen Benutzern hinterlegt.

Eine Liste der vorhandenen Medientypen mit ihren Details wird angezeigt.

Note:
Wenn Zustellmethoden (*E-Mail*, *SMS*, *Webhook*, *Skript* oder *Push*) im Feature-Flag `$ZBX_FEATURE_FLAGS['media_type_denylist']` in der Frontend-Konfigurationsdatei angegeben sind, werden die entsprechenden vorkonfigurierten Medientypen in diesem Abschnitt deaktiviert.



Angezeigte Daten:

Spalte	Beschreibung
Name	Name des Medientyps. Durch Klicken auf den Namen wird das Konfigurationsformular für Medientypen geöffnet.
Typ	Der Typ des Mediums (E-Mail, SMS usw.) wird angezeigt.
Status	Der Status des Medientyps wird angezeigt – *Aktiviert* oder *Deaktiviert*. Durch Klicken auf den Status können Sie ihn ändern.
In Aktionen verwendet	Die Aktionen, in denen der Medientyp verwendet wird, werden angezeigt, jeweils vorangestellt von der Gesamtzahl dieser Aktionen. Durch Klicken auf den Aktionsnamen wird das Konfigurationsformular für Aktionen geöffnet. Wenn der Benutzer keine Berechtigungen für die Aktion besitzt, ist der Name nicht anklickbar.
Details	Detaillierte Informationen zum Medientyp werden angezeigt.
Aktionen	Die folgende Aktion ist verfügbar: **Testen** – Klicken Sie hier, um ein Testformular zu öffnen, in dem Sie Parameter des Medientyps eingeben (z. B. eine Empfängeradresse mit Testbetreff und -text) und eine Testnachricht senden können, um zu überprüfen, ob der konfigurierte Medientyp funktioniert. Siehe auch: Testen von Medientypen.

Um einen neuen Medientyp zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Medientyp erstellen*.

Um einen Medientyp zu importieren, klicken Sie oben rechts auf die Schaltfläche *Importieren*.

Optionen für die Massenbearbeitung

Die Schaltflächen unter der Liste bieten einige Optionen für die Massenbearbeitung:

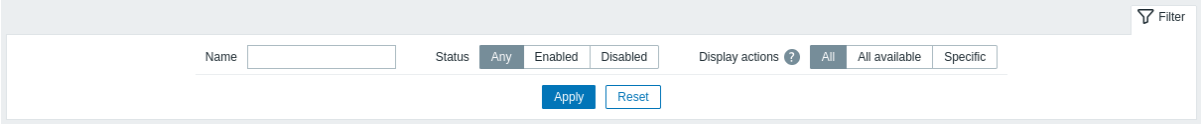
- *Aktivieren* - den Status des Medientyps auf *Aktiviert* ändern
- *Deaktivieren* - den Status des Medientyps auf *Deaktiviert* ändern
- *Exportieren* - die Medientypen in eine YAML-, XML- oder JSON-Datei exportieren
- *Löschen* - die Medientypen löschen

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Medientypen und klicken Sie dann auf die gewünschte Schaltfläche.

Filter verwenden

Sie können den Filter verwenden, um nur die Medienarten anzuzeigen, die Sie interessieren. Für eine bessere Suchleistung werden die Daten mit nicht aufgelösten Makros durchsucht.

Der Link *Filter* ist oberhalb der Liste der Medienarten verfügbar. Wenn Sie darauf klicken, wird ein Filter eingeblendet, mit dem Sie Medienarten nach Name und Status filtern können. Zusätzlich können Sie den Filter verwenden, um Aktionen in der Spalte *Verwendet in Aktionen* basierend auf dem Umfang ihrer Medienartverwendung anzuzeigen (definiert durch den Parameter *An Medienart senden* in den Aktionsdetails der Aktion).



Skripte

Übersicht

Im Abschnitt *Alarme* > *Skripte* können benutzerdefinierte globale Skripte konfiguriert und verwaltet werden.

Dieses Menü ist nur für Benutzer des Benutzertyps SuperAdmin verfügbar.

Globale Skripte stehen abhängig vom konfigurierten Geltungsbereich und den Benutzerberechtigungen zur Ausführung zur Verfügung:

- Über das Host-Menü an verschiedenen Stellen im Frontend (*Dashboard*, *Probleme*, *Neueste Daten*, *Karten* usw.).
- Über das Ereignismenü.
- Als Operation einer Aktion.

Die Skripte werden nur vom Zabbix Agent, Zabbix Server (Proxy) oder Zabbix Server ausgeführt. Siehe auch Befehlsausführung.

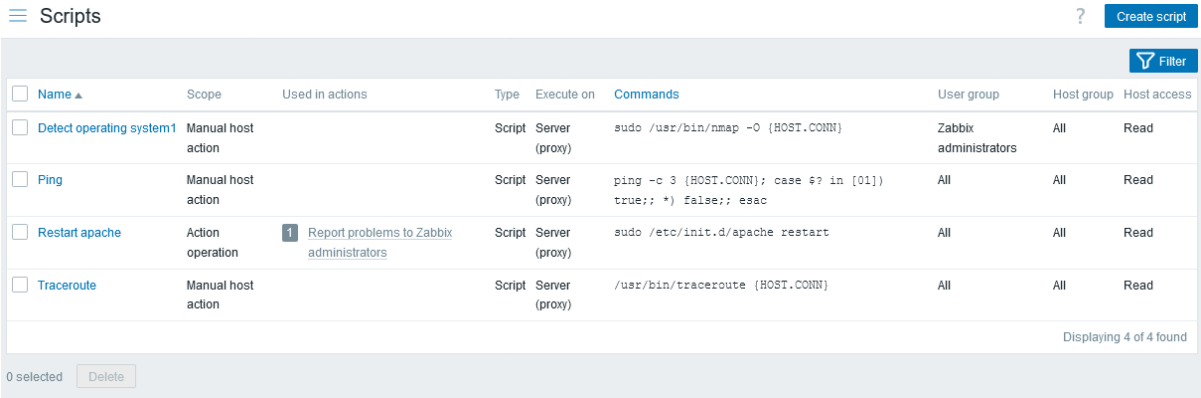
Remote-Befehle sind sowohl im Zabbix Agent als auch im Zabbix Proxy standardmäßig deaktiviert. Sie können wie folgt aktiviert werden:

- Für Remote-Befehle, die vom Zabbix Agent ausgeführt werden, fügen Sie für jeden zulässigen Befehl in der Agent-Konfiguration den Parameter `AllowKey=system.run[<command>,*]` oder `AllowKeyRegexp=^system\.run\[<command>,.*\]$` hinzu, wobei `*` entweder den Ausführungsmodus (`wait` oder `nowait`) bezeichnet.
- Für Remote-Befehle, die vom Zabbix Proxy ausgeführt werden, setzen Sie den Parameter `EnableRemoteCommands` in der Proxy-Konfiguration auf 1. Beachten Sie, dass Remote-Befehle im Zabbix Proxy nicht aktiviert werden müssen, wenn sie vom Zabbix Agent auf einem vom Proxy überwachten Host ausgeführt werden.

Attention:
Verwenden Sie `AllowKey=system.run[*]` oder `AllowKeyRegexp=^system\.run\[.*\]$` nur, wenn dies erforderlich ist, da dadurch die Ausführung beliebiger Befehle auf dem Host ermöglicht wird. Aus Sicherheitsgründen wird empfohlen, nur exakt festgelegte Befehle zuzulassen, während `system.run[*]` standardmäßig abgelehnt wird (`DenyKey=system.run[*]`).

Die Ausführung globaler Skripte auf dem Zabbix Server kann deaktiviert werden, indem `EnableGlobalScripts=0` in der Server-Konfiguration festgelegt wird. Bei Neuinstallationen ab Zabbix 7.0 ist die Ausführung globaler Skripte auf dem Zabbix Server standardmäßig deaktiviert.

Eine Liste der vorhandenen Skripte mit ihren Details wird angezeigt.



Angezeigte Daten:

Spalte	Beschreibung
Name	Name des Skripts. Durch Klicken auf den Skriptnamen wird das Konfigurationsformular für Skripte geöffnet.
Geltungsbereich	Geltungsbereich des Skripts – Aktionsoperation, manuelle Host-Aktion oder manuelle Ereignisaktion. Diese Einstellung bestimmt, wo das Skript verfügbar ist – in Remote-Befehlen von Aktionsoperationen, über das Host-Menü oder über das Ereignismenü.
In Aktionen verwendet	Alle Aktionen, in denen das Skript verwendet wird, werden angezeigt, vorangestellt durch die Gesamtzahl dieser Aktionen. Durch Klicken auf den Aktionsnamen wird das Aktionskonfigurationsformular geöffnet. Wenn der Benutzer keine Berechtigungen für die Aktion besitzt, ist der Name nicht anklickbar.
Typ	Der Skripttyp wird angezeigt – Befehl *URL*, *Webhook*, *Skript*, *SSH*, *Telnet* oder *IPMI*.
Ausführen auf	Es wird angezeigt, ob das Skript auf dem Zabbix Agent, dem Zabbix Proxy oder dem Server oder nur auf dem Zabbix Server ausgeführt wird.
Befehle	Alle innerhalb des Skripts auszuführenden Befehle werden angezeigt. Für Webhooks wird hier nichts angezeigt.
Benutzergruppe	Die Benutzergruppe, für die das Skript verfügbar ist, wird angezeigt (oder *Alle* für alle Benutzergruppen).
Hostgruppe	Die Hostgruppe, für die das Skript verfügbar ist, wird angezeigt (oder *Alle* für alle Hostgruppen).
Hostzugriff	Die Berechtigungsstufe für die Hostgruppe wird angezeigt – *Lesen* oder *Schreiben*. Nur Benutzer mit der erforderlichen Berechtigungsstufe können das Skript ausführen.

Um ein neues Skript zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Skript erstellen*.

Optionen für die Massenbearbeitung

Eine Schaltfläche unter der Liste bietet eine Massenbearbeitungsoption:

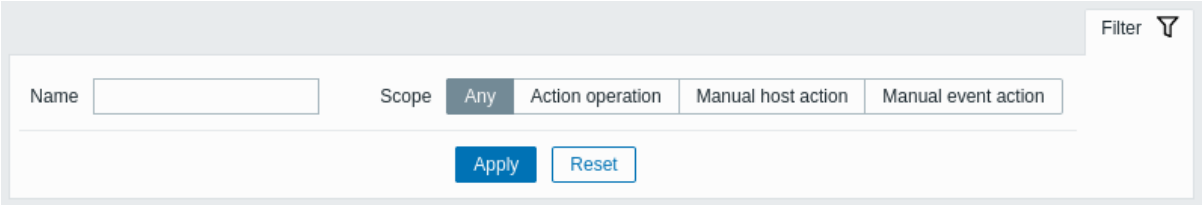
- *Löschen* - die Skripte löschen

Um diese Option zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Skripten und klicken Sie auf *Löschen*.

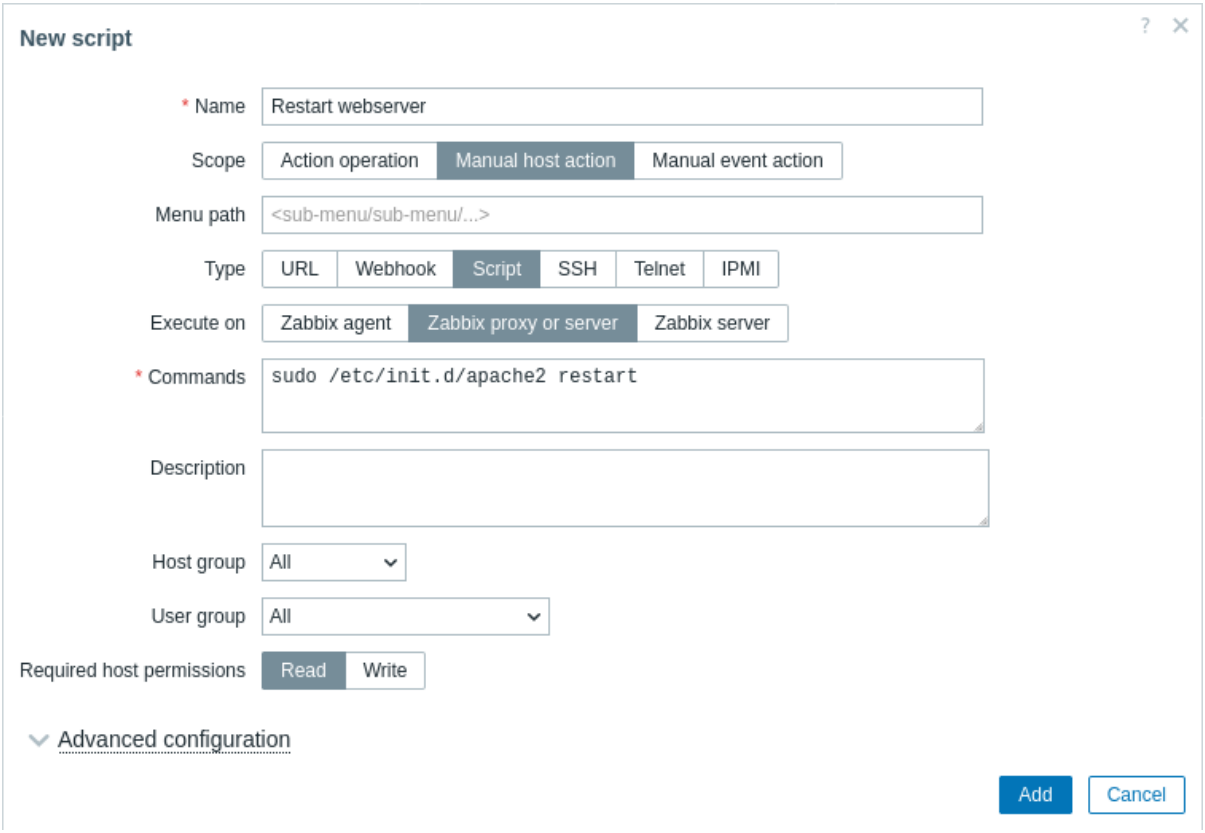
Filter verwenden

Sie können den Filter verwenden, um nur die Skripte anzuzeigen, die Sie interessieren. Für eine bessere Suchleistung werden die Daten mit nicht aufgelösten Makros durchsucht.

Der Link *Filter* ist oberhalb der Skriptliste verfügbar. Wenn Sie darauf klicken, wird ein Filter angezeigt, mit dem Sie Skripte nach Name und Geltungsbereich filtern können.



Konfigurieren eines globalen Skripts



Note:
Es wird empfohlen, in globalen Skripten Makrofunktionen anstelle einfacher Makros zu verwenden, um die Sicherheit zu erhöhen, da Makros nicht automatisch maskiert werden.

Skriptattribute:

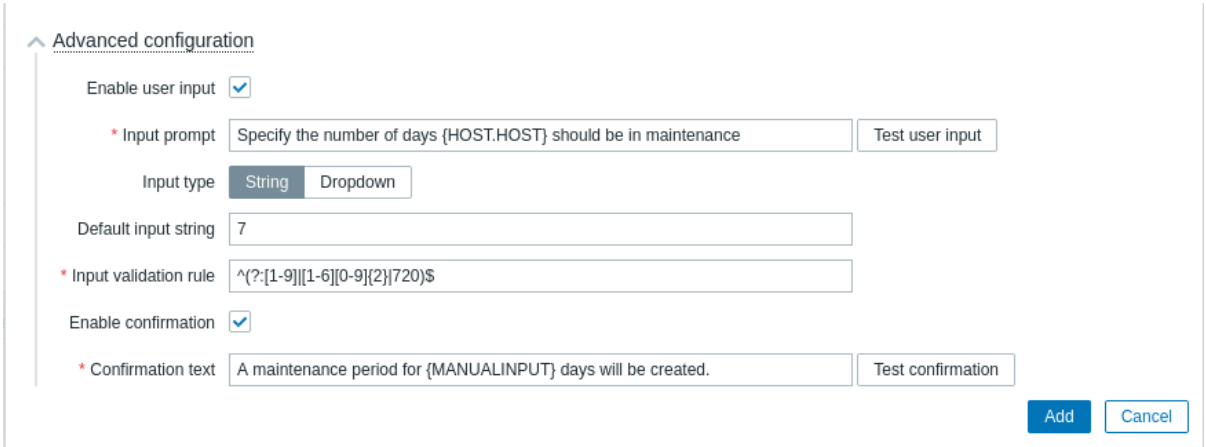
Parameter	Beschreibung
Name	Eindeutiger Name des Skripts. Beispiel: `Clear /tmp filesystem`
Geltungsbereich	Geltungsbereich des Skripts - Aktionsvorgang, manuelle Host-Aktion oder manuelle Ereignisaktion. Diese Einstellung bestimmt, wo das Skript verwendet werden kann - in Remote-Befehlen von Vorgängen einer Aktion, über das Hostmenü oder über das Ereignismenü. Wenn der Geltungsbereich auf „Aktionsvorgang“ gesetzt wird, ist das Skript für alle Benutzer mit Zugriff auf *Alarme* > *Aktionen* verfügbar. Wenn ein Skript tatsächlich in einer Aktion verwendet wird, kann sein Geltungsbereich nicht von „Aktionsvorgang“ weg geändert werden. **Makrounterstützung** Der Geltungsbereich beeinflusst den Bereich der verfügbaren Makros. Beispielsweise werden benutzerbezogene Makros (`{USER.*}`) in Skripten unterstützt, damit Informationen über den Benutzer übergeben werden können, der das Skript gestartet hat. Sie werden jedoch nicht unterstützt, wenn der Geltungsbereich des Skripts „Aktionsvorgang“ ist, da Aktionsvorgänge automatisch ausgeführt werden. Mit einem `{MANUALINPUT}`-Makro können Sie eine manuelle Eingabe zum Zeitpunkt der Skriptausführung angeben. Es wird für Skripte für manuelle Host-Aktionen und manuelle Ereignisaktionen unterstützt. Um herauszufinden, welche anderen Makros unterstützt werden, suchen Sie in der Tabelle Unterstützte Makros nach „Trigger-based notifications and commands/Trigger-based commands“, „Manual host action scripts“ und „Manual event action scripts“. Beachten Sie, dass Sie Makros, die zu einem Wert mit Leerzeichen aufgelöst werden können (z. B. Hostname), bei Bedarf in Anführungszeichen setzen müssen.

Parameter	Beschreibung
Menüpfad	Der gewünschte Menüpfad zum Skript. Beispielsweise wird das Skript mit `Default` oder `Default/` im jeweiligen Verzeichnis angezeigt. Menüs können verschachtelt werden, z. B. `Main menu/Sub menu1/Sub menu2`. Beim Zugriff auf Skripte über das Host- oder Ereignismenü in den Überwachungsbereichen werden sie entsprechend den angegebenen Verzeichnissen organisiert. Dieses Feld wird nur angezeigt, wenn „Manuelle Host-Aktion“ oder „Manuelle Ereignisaktion“ als *Geltungsbereich* ausgewählt ist.
Typ	Klicken Sie auf die entsprechende Schaltfläche, um den Skripttyp auszuwählen: **URL**, **Webhook**, **Skript**, **SSH**, **Telnet** oder **IPMI**-Befehl. Der Typ **URL** ist nur verfügbar, wenn „Manuelle Host-Aktion“ oder „Manuelle Ereignisaktion“ als *Geltungsbereich* ausgewählt ist.
Skripttyp: **URL**	
URL	Geben Sie die URL für den Schnellzugriff über das Hostmenü oder Ereignismenü an. Makros und benutzerdefinierte Benutzermakros werden unterstützt. Die Makrounterstützung hängt vom Geltungsbereich des Skripts ab (siehe *Geltungsbereich* oben). Verwenden Sie das Makro {MANUALINPUT} in diesem Feld, um eine manuelle Eingabe zum Zeitpunkt der Skriptausführung angeben zu können, zum Beispiel: `http://{MANUALINPUT}/zabbix/zabbix.php?action=dashboard.view` Makrowerte dürfen nicht URL-codiert werden.
In neuem Fenster öffnen	Legt fest, ob die URL in einem neuen oder im selben Browser-Tab geöffnet werden soll.
Skripttyp: **Webhook**	
Parameter	Geben Sie die Webhook-Variablen als Attribut-Wert-Paare an. Siehe auch: Medienkonfiguration Webhook. Makros und benutzerdefinierte Benutzermakros werden in Parameterwerten unterstützt. Die Makrounterstützung hängt vom Geltungsbereich des Skripts ab (siehe *Geltungsbereich* oben).
Skript	Geben Sie den JavaScript-Code im modalen Editor ein, der geöffnet wird, wenn Sie in das Parameterfeld oder auf das Stiftsymbol daneben klicken. Die Makrounterstützung hängt vom Geltungsbereich des Skripts ab (siehe *Geltungsbereich* oben). Siehe auch: Medienkonfiguration Webhook, Zusätzliche JavaScript-Objekte.
Zeitüberschreitung	Zeitüberschreitung für die JavaScript-Ausführung (1-60 s, Standard: 30 s). Zeitsuffixe werden unterstützt (z. B. `30s`, `1m`).
Skripttyp: **Skript**	
Ausführen auf	Klicken Sie auf die entsprechende Schaltfläche, um das Shell-Skript auszuführen auf: **Zabbix Agent** - das Skript wird vom Zabbix Agent auf dem Host ausgeführt (wenn der system.run-Datenpunkt zugelassen ist) **Zabbix Proxy oder Server** - das Skript wird vom Zabbix Proxy oder Server ausgeführt, abhängig davon, ob der Host von einem Proxy oder Server überwacht wird. Es wird auf dem Proxy ausgeführt, wenn dies durch EnableRemoteCommands aktiviert ist. Es wird auf dem Server ausgeführt, wenn globale Skripte durch den Serverparameter EnableGlobalScripts aktiviert sind. **Zabbix Server** - das Skript wird ausschließlich vom Zabbix Server ausgeführt. Diese Option ist nicht verfügbar, wenn globale Skripte durch den Serverparameter EnableGlobalScripts deaktiviert sind.
Befehle	Geben Sie den vollständigen Pfad zu den Befehlen ein, die innerhalb des Skripts ausgeführt werden sollen. Die Makrounterstützung hängt vom Geltungsbereich des Skripts ab (siehe *Geltungsbereich* oben). Benutzerdefinierte Benutzermakros werden unterstützt.
Skripttyp: **SSH**	
Authentifizierungsmethode	Wählen Sie die Authentifizierungsmethode aus - Passwort oder öffentlicher Schlüssel.
Benutzername	Geben Sie den Benutzernamen ein.
Passwort	Geben Sie das Passwort ein. Dieses Feld ist verfügbar, wenn „Passwort“ als Authentifizierungsmethode ausgewählt ist.
Datei mit öffentlichem Schlüssel	Geben Sie den Pfad zur Datei mit dem öffentlichen Schlüssel ein. Dieses Feld ist verfügbar, wenn „Öffentlicher Schlüssel“ als Authentifizierungsmethode ausgewählt ist.
Datei mit privatem Schlüssel	Geben Sie den Pfad zur Datei mit dem privaten Schlüssel ein. Dieses Feld ist verfügbar, wenn „Öffentlicher Schlüssel“ als Authentifizierungsmethode ausgewählt ist.

Parameter	Beschreibung
Passphrase	Geben Sie die Passphrase ein. Dieses Feld ist verfügbar, wenn „Öffentlicher Schlüssel“ als Authentifizierungsmethode ausgewählt ist.
Port	Geben Sie den Port des Remote-SSH-Dienstes auf dem Zielhost ein, mit dem Zabbix eine Verbindung herstellt.
Befehle	Geben Sie die Befehle ein. Die Makrounterstützung hängt vom Geltungsbereich des Skripts ab (siehe *Geltungsbereich* oben). Benutzerdefinierte Benutzermakros werden unterstützt.
Skripttyp: **Telnet**	
Benutzername	Geben Sie den Benutzernamen ein.
Passwort	Geben Sie das Passwort ein.
Port	Geben Sie den Port des Remote-Telnet-Dienstes auf dem Zielhost ein, mit dem Zabbix eine Verbindung herstellt.
Befehle	Geben Sie die Befehle ein. Die Makrounterstützung hängt vom Geltungsbereich des Skripts ab (siehe *Geltungsbereich* oben). Benutzerdefinierte Benutzermakros werden unterstützt.
Skripttyp: **IPMI**	
Befehl	Geben Sie den IPMI-Befehl ein. Die Makrounterstützung hängt vom Geltungsbereich des Skripts ab (siehe *Geltungsbereich* oben). Benutzerdefinierte Benutzermakros werden unterstützt.
Beschreibung	Geben Sie eine Beschreibung für das Skript ein.
Hostgruppe	Wählen Sie die Hostgruppe aus, für die das Skript verfügbar sein soll (oder *Alle* für alle Hostgruppen).
Benutzergruppe	Wählen Sie die Benutzergruppe aus, für die das Skript verfügbar sein soll (oder *Alle* für alle Benutzergruppen). Dieses Feld wird nur angezeigt, wenn „Manuelle Host-Aktion“ oder „Manuelle Ereignisaktion“ als *Geltungsbereich* ausgewählt ist.
Erforderliche Host-berech-ti-gun-gen	Wählen Sie die Berechtigungsstufe für die Hostgruppe aus - *Lesen* oder *Schreiben*. Nur Benutzer mit der erforderlichen Berechtigungsstufe können das Skript ausführen. Dieses Feld wird nur angezeigt, wenn „Manuelle Host-Aktion“ oder „Manuelle Ereignisaktion“ als *Geltungsbereich* ausgewählt ist.
Erweiterte Kon-fig-u-ra-tion	Klicken Sie auf die Überschrift *Erweiterte Konfiguration*, um Optionen für die erweiterte Konfiguration anzuzeigen. Dieses Feld wird nur angezeigt, wenn „Manuelle Host-Aktion“ oder „Manuelle Ereignisaktion“ als *Geltungsbereich* ausgewählt ist.

Erweiterte Konfiguration

Erweiterte Konfigurationsoptionen sind im aufklappbaren Abschnitt *Erweiterte Konfiguration* verfügbar:



Parameter	Beschreibung
Benutzereingabe aktivieren	Aktivieren Sie das Kontrollkästchen, um vor der Ausführung des Skripts eine manuelle Benutzereingabe zu ermöglichen. Die manuelle Benutzereingabe ersetzt den Makrowert {MANUALINPUT} im Skript. Siehe auch: Manuelle Benutzereingabe.
Eingabeaufforderung	Geben Sie einen benutzerdefinierten Text ein, der zur Eingabe auffordert. Dieser Text wird über dem Eingabefeld im Popup *Manuelle Eingabe* angezeigt. Um eine Vorschau des Popups *Manuelle Eingabe* anzuzeigen, klicken Sie auf *Benutzereingabe testen*. In der Vorschau können Sie außerdem testen, ob die Eingabezeichenfolge der Eingabevalidierungsregel entspricht (siehe die folgenden Parameter). Die Unterstützung von Makros und Benutzermakros hängt vom Geltungsbereich des Skripts ab (siehe *Geltungsbereich* in den allgemeinen Konfigurationsparametern für Skripte).
Eingabetyp	Wählen Sie den Typ der manuellen Eingabe aus: **Zeichenfolge** – einzelne Zeichenfolge; **Dropdown** – der Wert wird aus mehreren Dropdown-Optionen ausgewählt.
Dropdown-Optionen	Geben Sie eindeutige Werte für das Dropdown-Menü der Benutzereingabe als durch Kommas getrennte Liste ein. Um eine leere Option in das Dropdown-Menü aufzunehmen, fügen Sie am Anfang, in der Mitte oder am Ende der Liste ein zusätzliches Komma ein. Dieses Feld wird nur angezeigt, wenn *Dropdown* als *Eingabetyp* ausgewählt ist.
Standard-Eingabezeichenfolge	Geben Sie die Standardzeichenfolge für die Benutzereingabe ein (oder keine). Dieses Feld wird anhand des regulären Ausdrucks validiert, der im Feld *Eingabevalidierungsregel* angegeben ist. Der hier eingegebene Wert wird standardmäßig im Popup *Manuelle Eingabe* angezeigt. Dieses Feld wird nur angezeigt, wenn *Zeichenfolge* als *Eingabetyp* ausgewählt ist.
Eingabevalidierungsregel	Geben Sie einen regulären Ausdruck zur Validierung der Benutzereingabezeichenfolge ein. Globale reguläre Ausdrücke werden unterstützt. Dieses Feld wird nur angezeigt, wenn *Zeichenfolge* als *Eingabetyp* ausgewählt ist.
Bestätigung aktivieren	Aktivieren Sie das Kontrollkästchen, um vor der Ausführung des Skripts eine Bestätigungsmeldung anzuzeigen. Diese Funktion kann besonders bei potenziell gefährlichen Vorgängen (z. B. einem Neustartskript) oder bei Vorgängen nützlich sein, die längere Zeit dauern können.
Bestätigungstext	Geben Sie einen benutzerdefinierten Bestätigungstext für das mit dem obigen Kontrollkästchen aktivierte Bestätigungs-Popup ein (z. B. *Das Remotesystem wird neu gestartet. Sind Sie sicher?*). Um zu sehen, wie der Text angezeigt wird, klicken Sie neben dem Feld auf *Bestätigung testen*. Makros und benutzerdefinierte Benutzermakros werden unterstützt. *Hinweis:* Die Makros werden beim Testen der Bestätigungsmeldung nicht aufgelöst.

Wenn sowohl eine manuelle Benutzereingabe als auch eine Bestätigungsmeldung konfiguriert sind, werden sie in aufeinanderfolgenden Popup-Fenstern angezeigt.

Manuelle Benutzereingabe

Mit der manuellen Benutzereingabe kann bei jeder Ausführung des Skripts ein benutzerdefinierter Parameter angegeben werden. Dadurch entfällt die Notwendigkeit, mehrere ähnliche Benutzerskripte zu erstellen, die sich nur durch einen einzelnen Parameter unterscheiden.

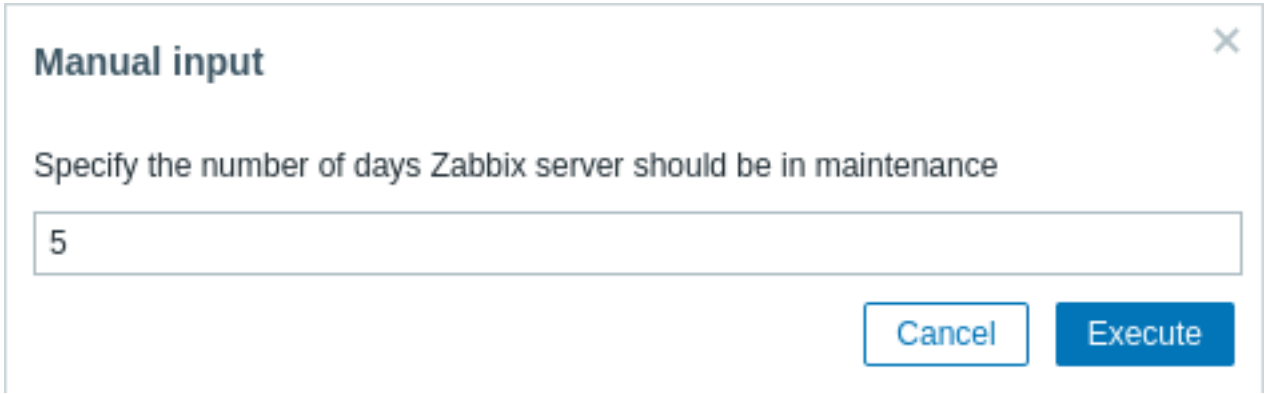
Beispielsweise können Sie dem Skript bei der Ausführung eine andere Ganzzahl oder eine andere URL-Adresse übergeben.

So aktivieren Sie die manuelle Benutzereingabe:

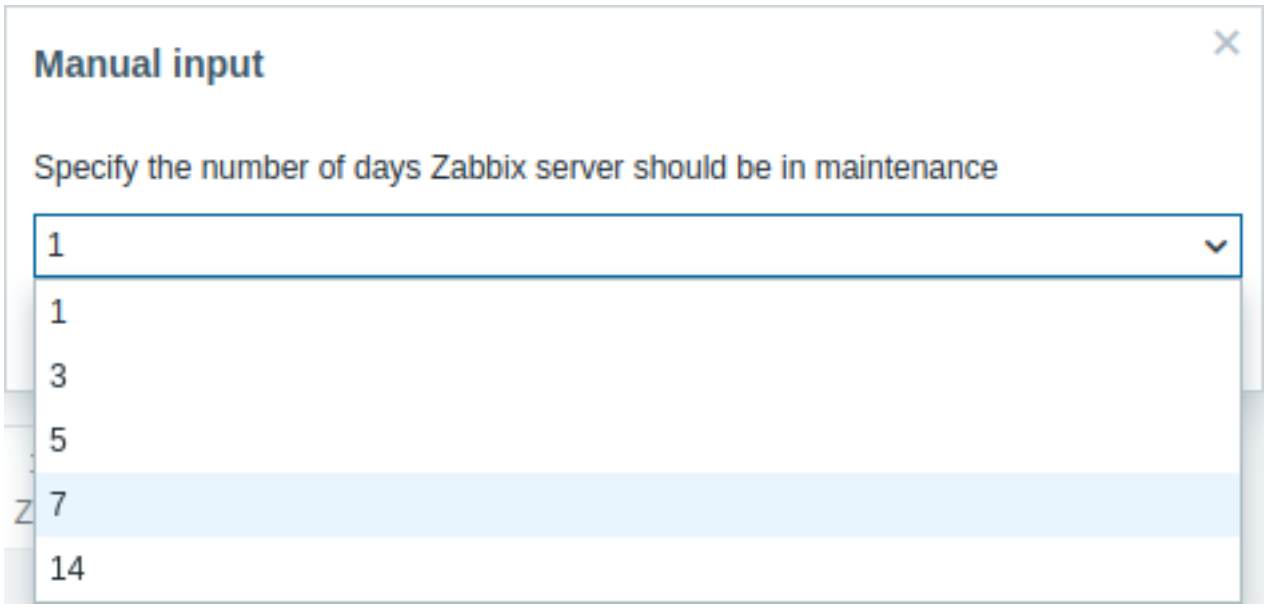
- Verwenden Sie das Makro {MANUALINPUT} im Skript (Befehle, Skript, Skriptparameter) an den erforderlichen Stellen oder im URL-Feld von URL-Skripten.
- Aktivieren Sie in der erweiterten Skriptkonfiguration die manuelle Benutzereingabe und konfigurieren Sie die Eingabeoptionen.

Wenn die Benutzereingabe aktiviert ist, wird dem Benutzer vor der Skriptausführung ein Popup-Fenster *Manuelle Eingabe* angezeigt, in dem er zur Eingabe eines benutzerdefinierten Werts aufgefordert wird. Der eingegebene Wert ersetzt {MANUALINPUT} im Skript.

Je nach Konfiguration wird der Benutzer aufgefordert, einen Zeichenfolgenwert einzugeben:



Oder den Wert aus einer Dropdown-Liste mit vordefinierten Optionen auszuwählen:



Die manuelle Benutzereingabe ist nur für Skripte verfügbar, deren Geltungsbereich „Manuelle Host-Aktion“ oder „Manuelle Ereignisaktion“ lautet.

Skriptausführung und Ergebnis

Vom Zabbix Server ausgeführte Skripte werden in der auf der Seite Befehlsausführung beschriebenen Reihenfolge ausgeführt.

Das Skriptergebnis wird in einem Pop-up-Fenster angezeigt, das nach der Ausführung des Skripts erscheint. Der Rückgabewert des Skripts ist eine Standardausgabe:

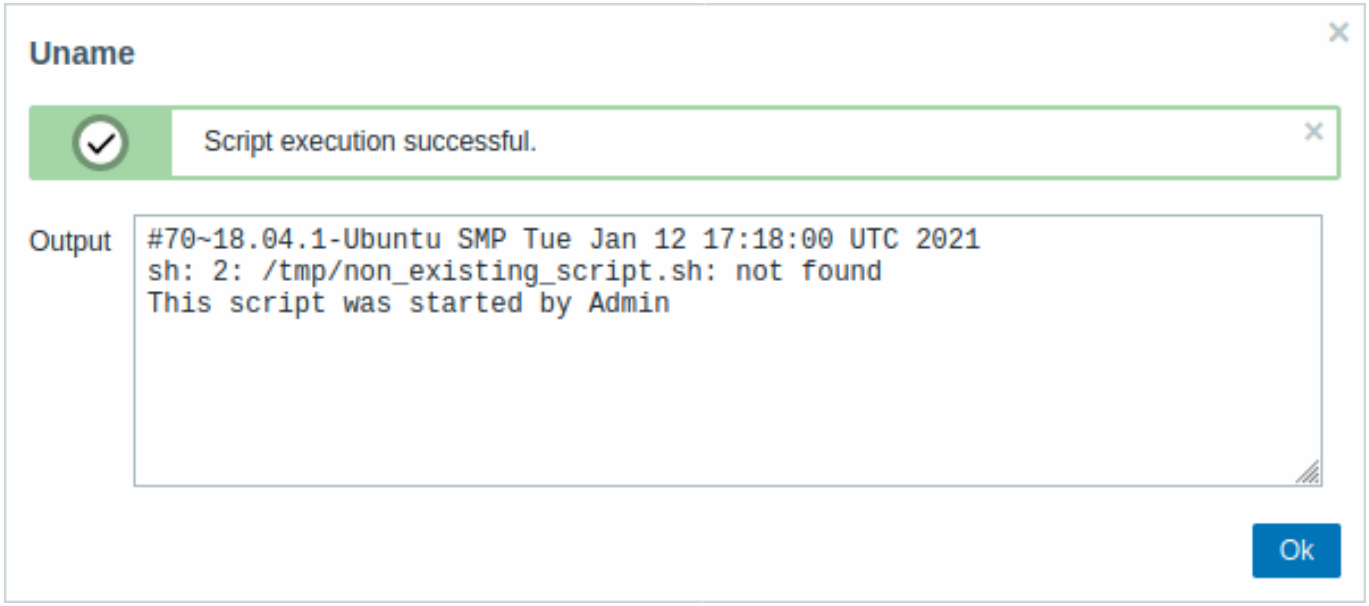
- Wenn das Skript erfolgreich abgeschlossen wird (Exit-Code 0), ist der Rückgabewert auf 16 MB begrenzt (einschließlich nachgestellter Leerzeichen, die abgeschnitten werden).
- Wenn das Skript mit einem Fehler beendet wird (Exit-Code ungleich null), ist der Rückgabewert eine Standardfehlerausgabe, die auf 2 KB begrenzt ist.

Zabbix speichert eine erweiterte Skriptausgabe standardmäßig nicht. Um die vollständigen Ausgabedetails beizubehalten, können Sie die Protokollierung im Skript selbst implementieren (z. B. durch Umleitung der Ausgabe in eine lokale Protokolldatei).

Beachten Sie, dass für Skripte, die entweder auf dem Zabbix Server oder dem Zabbix Proxy ausgeführt werden, auch Datenbankgrenzwerte gelten.

Nachfolgend sehen Sie ein Beispiel für ein Skript und das Ergebnisfenster:

```
uname -v
/tmp/non_existing_script.sh
echo "This script was started by {USER.USERNAME}"
```



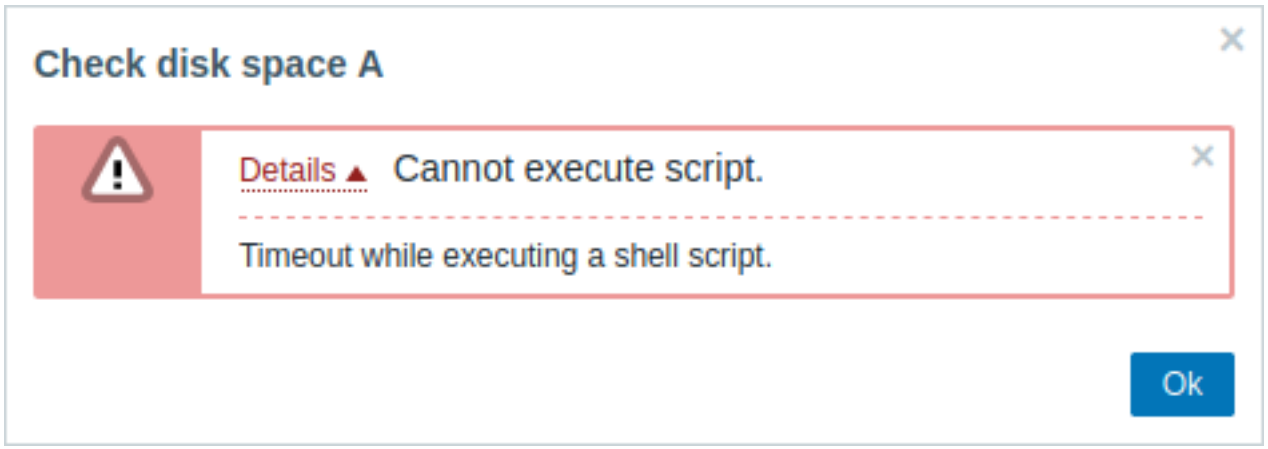
Das Skriptergebnis zeigt nicht das Skript selbst an.

Script-Timeout beim Zabbix Agent

Es kann vorkommen, dass beim Ausführen eines Scripts ein Timeout auftritt.

Nachfolgend sehen Sie ein Beispiel für ein Script, das auf dem Zabbix Agent ausgeführt wird, sowie das Ergebnisfenster:

```
sleep 5
df -h
```



Die Fehlermeldung lautet in diesem Fall:

```
Timeout while executing a shell script.
```

Um solche Situationen zu vermeiden, wird empfohlen, das Script selbst zu optimieren (im obigen Beispiel den Wert „5“), anstatt den Parameter `Timeout` in der Agent-Konfiguration und der Server-Konfiguration von Zabbix anzupassen.

Wenn der Parameter `Timeout` in der Agent-Konfiguration von Zabbix geändert wurde, wird die folgende Fehlermeldung angezeigt:

```
Get value from agent failed: ZBX_TCP_READ() timed out.
```

Das bedeutet, dass die Änderung in der Agent-Konfiguration von Zabbix vorgenommen wurde, der Parameter `Timeout` jedoch auch in der Server-Konfiguration von Zabbix geändert werden muss.

Skript-Timeout auf aktiven Agents

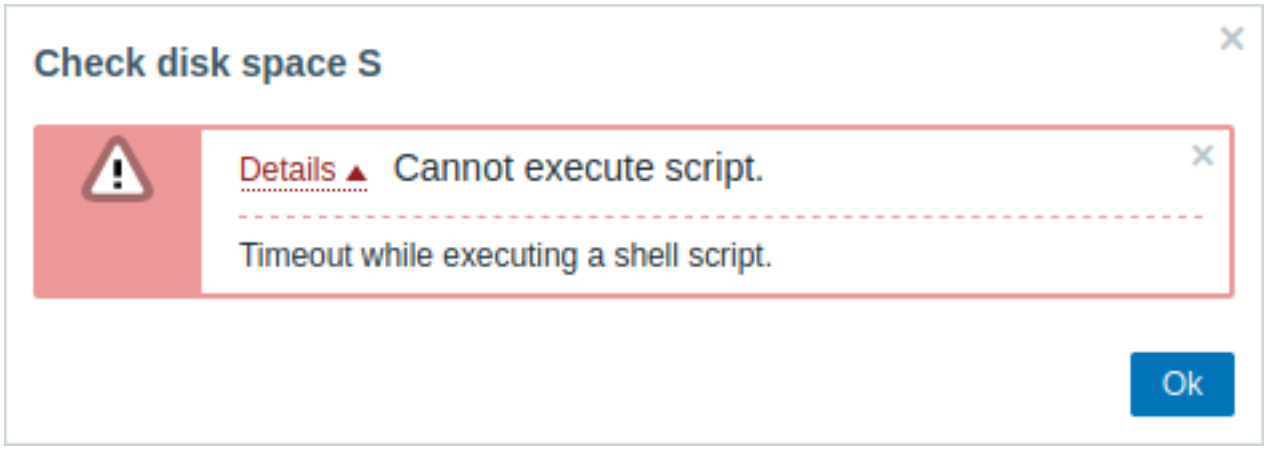
Es ist nur möglich, Skripte auf Agents auszuführen, die im aktiven Modus betrieben werden. Sobald die Skriptausführung durch eine Aktions-Operation oder eine manuelle Skriptausführung ausgelöst wird, wird der Befehl in die Konfiguration der aktiven Prüfungen aufgenommen und ausgeführt, sobald der aktive Agent ihn empfängt.

Manuelle Skripte werden zusammen mit dem Server-/Proxy-Timeout für die Skriptausführung an den aktiven Agent gesendet. Bitte erhöhen Sie das standardmäßige Server/Proxy-Timeout für die Skriptausführung. Das Timeout muss höher sein als die Aktualisierungsfrequenz der aktiven Prüfung, andernfalls wird das Timeout überschritten, bevor der aktive Agent das Skript empfängt und das Ergebnis zurückgeben kann.

Skript-Timeout auf dem Zabbix Server/Proxy

Nachfolgend sehen Sie ein Beispiel für ein Skript, das auf dem Zabbix Server ausgeführt wird, sowie das Ergebnisfenster:

```
sleep 11
df -h
```



Es wird außerdem empfohlen, das Skript selbst zu optimieren, anstatt den Parameter `TrapperTimeout` auf einen entsprechenden Wert (in unserem Fall > 11) anzupassen, indem Sie die Zabbix-Serverkonfiguration ändern.

Benutzer

Übersicht

Dieses Menü enthält Abschnitte, die sich auf die Konfiguration von Benutzern in Zabbix beziehen. Dieses Menü ist nur für Benutzer des Benutzertyps SuperAdmin verfügbar.

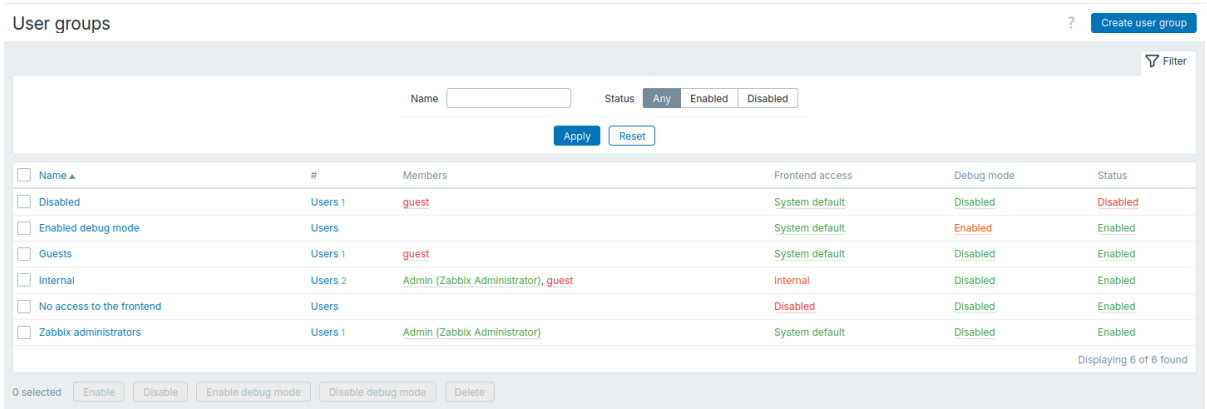
Benutzergruppen

Übersicht

Im Abschnitt *Users* > *User groups* werden die Benutzergruppen des Systems verwaltet.

Benutzergruppen

Eine Liste der vorhandenen Benutzergruppen mit ihren Details wird angezeigt.



Angezeigte Daten:

Spalte	Beschreibung
Name	Name der Benutzergruppe. Wenn Sie auf den Namen der Benutzergruppe klicken, wird das Konfigurationsformular für Benutzergruppen geöffnet.
#	Die Anzahl der Benutzer in der Gruppe. Wenn Sie auf *Benutzer* klicken, werden die entsprechenden Benutzer gefiltert in der Benutzerliste angezeigt.
Mitglieder	Benutzernamen der einzelnen Benutzer in der Benutzergruppe (mit Vor- und Nachnamen in Klammern). Wenn Sie auf den Benutzernamen klicken, wird das Konfigurationsformular für Benutzer geöffnet. Benutzer aus deaktivierten Gruppen werden rot angezeigt.

Spalte	Beschreibung
Frontend-Zugriff	Die Zugriffsebene für das Frontend wird angezeigt: **Systemstandard** – Benutzer werden von Zabbix, LDAP oder HTTP authentifiziert (abhängig von der global festgelegten Authentifizierungsmethode); **Intern** – Benutzer werden von Zabbix authentifiziert; wird ignoriert, wenn die HTTP-Authentifizierung der globale Standard ist; **LDAP** – Benutzer werden von LDAP authentifiziert; wird ignoriert, wenn die HTTP-Authentifizierung der globale Standard ist; **Deaktiviert** – Der Zugriff auf das Zabbix-Frontend ist für diese Gruppe verboten. Durch Klicken auf die aktuelle Ebene können Sie sie ändern.
Debug-Modus	Der Status des Debug-Modus wird angezeigt – *Aktiviert* oder *Deaktiviert*. Durch Klicken auf den Status können Sie ihn ändern.
Status	Der Status der Benutzergruppe wird angezeigt – *Aktiviert* oder *Deaktiviert*. Durch Klicken auf den Status können Sie ihn ändern.

Um eine neue Benutzergruppe zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Benutzergruppe erstellen*.

Optionen für die Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen für die Massenbearbeitung:

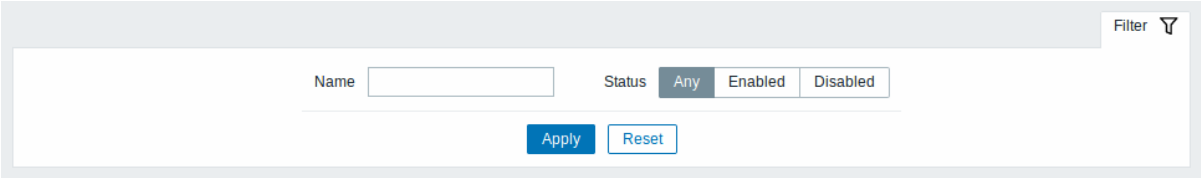
- *Aktivieren* - den Status der Benutzergruppe auf *Aktiviert* ändern
- *Deaktivieren* - den Status der Benutzergruppe auf *Deaktiviert* ändern
- *Debugmodus aktivieren* - den Debugmodus für die Benutzergruppen aktivieren
- *Debugmodus deaktivieren* - den Debugmodus für die Benutzergruppen deaktivieren
- *Löschen* - die Benutzergruppen löschen

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Benutzergruppen und klicken Sie dann auf die gewünschte Schaltfläche.

Verwenden des Filters

Sie können den Filter verwenden, um nur die Benutzergruppen anzuzeigen, an denen Sie interessiert sind. Für eine bessere Suchleistung werden die Daten mit nicht aufgelösten Makros durchsucht.

Der Link *Filter* ist oberhalb der Liste der Benutzergruppen verfügbar. Wenn Sie darauf klicken, wird ein Filter angezeigt, mit dem Sie Benutzergruppen nach Name und Status filtern können.



Benutzerrollen

Übersicht

Im Abschnitt *Benutzer* > *Benutzerrollen* können Sie Benutzerrollen erstellen.

Mit Benutzerrollen können Sie detaillierte Berechtigungen erstellen, die auf dem zunächst ausgewählten Benutzertyp (*Benutzer*, *Administrator*, *Super-Administrator*) basieren.

Nach der Auswahl eines Benutzertyps werden alle für diesen Benutzertyp verfügbaren Berechtigungen gewährt (standardmäßig aktiviert).

Berechtigungen können nur aus der für den Benutzertyp verfügbaren Teilmenge entzogen werden. Sie können nicht über die für den Benutzertyp verfügbaren Berechtigungen hinaus erweitert werden.

Kontrollkästchen für nicht verfügbare Berechtigungen sind ausgegraut. Benutzer können auch dann nicht auf das Element zugreifen, wenn sie eine direkte URL zu diesem Element in den Browser eingeben.

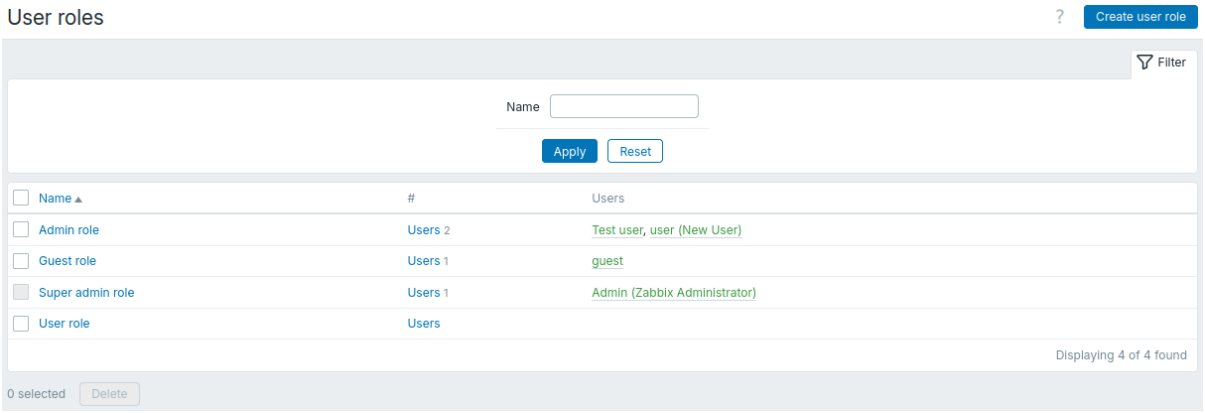
Note:
Die Einschränkung des Zugriffs auf bestimmte UI-Elemente verhindert lediglich das Öffnen dieser Seite. Sie entfernt nicht die Möglichkeit, auf die zugrunde liegenden Daten in anderen Teilen der Oberfläche zuzugreifen.

Benutzerrollen können Systembenutzern zugewiesen werden. Jedem Benutzer kann nur eine Rolle zugewiesen werden.

Standardbenutzerrollen

Standardmäßig ist Zabbix mit vier Benutzerrollen konfiguriert, die über einen vordefinierten Berechtigungssatz verfügen:

- Gastrolle
- Benutzerrolle
- Administratorrolle
- Superadministratorrolle



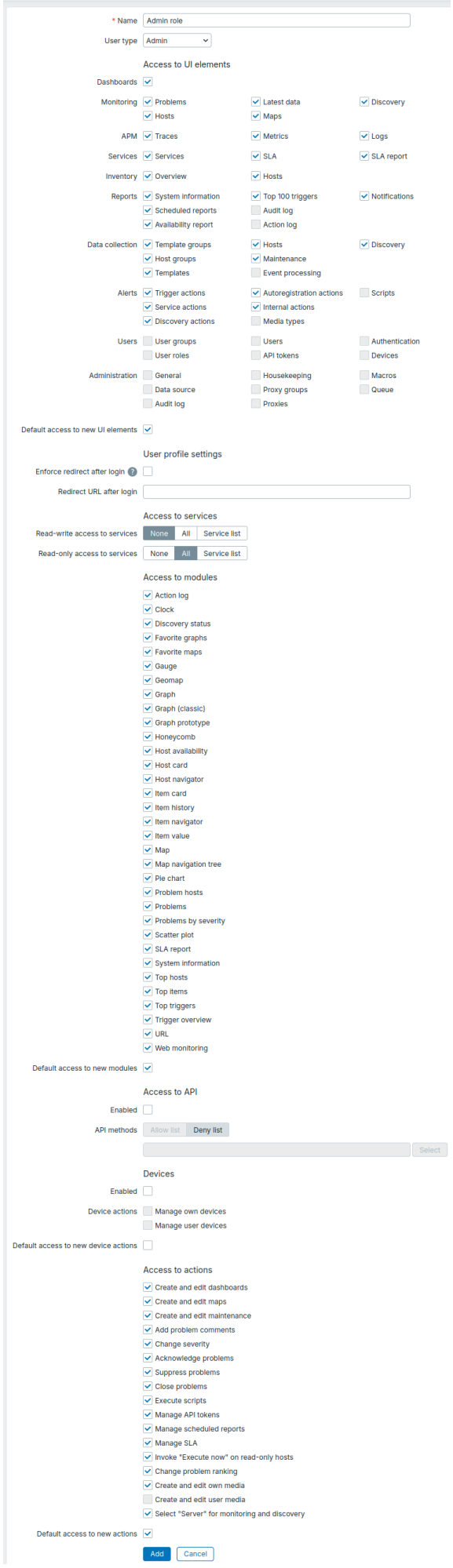
Diese basieren auf den wichtigsten Benutzertypen in Zabbix. Die Liste aller Benutzer, denen die jeweilige Rolle zugewiesen ist, wird angezeigt. Benutzer, die in deaktivierten Gruppen enthalten sind, werden rot angezeigt. Die *Gastrolle* ist eine Benutzerrollenart, die nur über Berechtigungen zum Anzeigen einiger Frontend-Bereiche verfügt.

Note:
Die standardmäßige *Superadministratorrolle* kann nicht geändert oder gelöscht werden, da in Zabbix mindestens ein Superadministratorbenutzer mit uneingeschränkten Berechtigungen vorhanden sein muss. Benutzer vom Typ *Superadministrator* können die Einstellungen ihrer eigenen Rolle ändern, nicht jedoch den Benutzertyp.

Konfiguration

Um eine neue Rolle zu erstellen, klicken Sie oben rechts auf die Schaltfläche *Benutzerrolle erstellen*. Um eine vorhandene Rolle zu aktualisieren, klicken Sie auf den Rollennamen, um das Konfigurationsformular zu öffnen.

User roles


* Name
Admin role
User type
Admin
Access to UI elements
Dashboards
Monitoring
Problems
Latest data
Discovery
Hosts
Maps
APM
Traces
Metrics
Logs
Services
Services
SLA
SLA report
Inventory
Overview
Hosts
Reports
System information
Top 100 triggers
Notifications
Scheduled reports
Audit log
Availability report
Action log
Data collection
Template groups
Hosts
Discovery
Host groups
Maintenance
Templates
Event processing
Alerts
Trigger actions
Autoregistration actions
Scripts
Service actions
Internal actions
Discovery actions
Media types
Users
User groups
Users
Authentication
User roles
API tokens
Devices
Administration
General
Housekeeping
Macros
Data source
Proxy groups
Queue
Audit log
Proxies
Default access to new UI elements
User profile settings
Enforce redirect after login
Redirect URL after login
Access to services
Read-write access to services
None
All
Service list
Read-only access to services
None
All
Service list
Access to modules
Action log
Clock
Discovery status
Favorite graphs
Favorite maps
Gauge
Geomap
Graph
Graph (classic)
Graph prototype
Honeycomb
Host availability
Host card
Host navigator
Item card
Item history
Item navigator
Item value
Map
Map navigation tree
Pie chart
Problem hosts
Problems
Problems by severity
Scatter plot
SLA report
System information
Top hosts
Top items
Top triggers
Trigger overview
URL
Web monitoring
Default access to new modules
Access to API
Enabled
API methods
Allow list
Deny list
Select
Devices
Enabled
Device actions
Manage own devices
Manage user devices
Default access to new device actions
Access to actions
Create and edit dashboards
Create and edit maps
Create and edit maintenance
Add problem comments
Change severity
Acknowledge problems
Suppress problems
Close problems
Execute scripts
Manage API tokens
Manage scheduled reports
Manage SLA
Invoke "Execute now" on read-only hosts
Change problem ranking
Create and edit own media
Create and edit user media
Select "Server" for monitoring and discovery
Default access to new actions
Add
Cancel

Die verfügbaren Berechtigungen werden angezeigt. Um eine bestimmte Berechtigung zu entziehen, deaktivieren Sie das entsprechende Kontrollkästchen.

Die verfügbaren Berechtigungen sowie die Standardeinstellungen für jede vorkonfigurierte Benutzerrolle in Zabbix werden im Folgenden beschrieben.

Standardberechtigungen

Zugriff auf UI-Elemente

Der standardmäßige Zugriff auf Menüabschnitte hängt vom Benutzertyp ab.
Weitere Details finden Sie auf der Seite „Berechtigungen“.

Zugriff auf andere Optionen

Einige der in der folgenden Tabelle beschriebenen Abschnitte können ausgeblendet sein:

- *Zugriff auf Module* ist ausgeblendet, wenn `$ZBX_FEATURE_FLAGS['modules_config_enabled']` in der Frontend-Konfigurationsdatei auf `false` gesetzt ist.
- *Geräte* ist ausgeblendet, wenn `EnableMobileDevices` deaktiviert ist.

Parameter	**Beschreibung**	**Standardbenutzerrollen**			
		Super-Admin-Rolle	*Admin-Rolle*	*Benutzerrolle*	*Gastrolle*
Standardzugriff auf neue UI-Elemente	Diese Option legt fest, wie nach einem Zabbix-Upgrade auf neue Menüabschnitte zugegriffen werden kann. Bestehende Menüabschnitte von Modulen bleiben davon unberührt.	Ja	Ja	Ja	Ja
Benutzerprofileinstellungen					
Weiterleitung nach der Anmeldung erzwingen	Aktivieren Sie das Kontrollkästchen, um Benutzer auf die Weiterleitungs-URL nach der Anmeldung zu beschränken. Wenn das Kontrollkästchen nicht aktiviert ist, können Benutzer ihre persönliche URL nach der Anmeldung in ihrem Profil festlegen; die Weiterleitungs-URL nach der Anmeldung wird nur als Fallback verwendet.	Ja	Nein	Nein	Nein
Weiterleitungs-URL nach der Anmeldung	Geben Sie die Weiterleitungs-URL nach der Anmeldung an.				
Zugriff auf Services					
Lese-/Schreibzugriff auf Services	Wählen Sie den Lese-/Schreibzugriff auf Services aus: **Keine** – überhaupt kein Zugriff **Alle** – Lese-/Schreibzugriff auf alle Services **Serviceliste** – Services für den Lese-/Schreibzugriff auswählen Der gewährte Lese-/Schreibzugriff hat Vorrang vor den Einstellungen für den Nur-Lese-Zugriff und wird dynamisch an untergeordnete Services vererbt.	Alle	Alle	Keine	Keine
Lese-/Schreibzugriff auf Services mit Tag	Geben Sie einen Tagnamen und optional einen Wert an, um zusätzlich Lese-/Schreibzugriff auf Services zu gewähren, die dem Tag entsprechen. Diese Option ist verfügbar, wenn im Parameter *Lese-/Schreibzugriff auf Services* die Option „Serviceliste“ ausgewählt ist. Der gewährte Lese-/Schreibzugriff hat Vorrang vor den Einstellungen für den Nur-Lese-Zugriff und wird dynamisch an untergeordnete Services vererbt.				

Nur-Lese-Zugriff auf Services	Wählen Sie den Nur-Lese-Zugriff auf Services aus: **Keine** – überhaupt kein Zugriff **Alle** – Nur-Lese-Zugriff auf alle Services **Serviceliste** – Services für den Nur-Lese-Zugriff auswählen Der Nur-Lese-Zugriff hat keinen Vorrang vor dem Lese-/Schreibzugriff und wird dynamisch an untergeordnete Services vererbt.			Alle	Alle
Nur-Lese-Zugriff auf Services mit Tag	Geben Sie einen Tagnamen und optional einen Wert an, um zusätzlich Nur-Lese-Zugriff auf Services zu gewähren, die dem Tag entsprechen. Diese Option ist verfügbar, wenn im Parameter *Nur-Lese-Zugriff auf Services* die Option „Serviceliste“ ausgewählt ist. Der Nur-Lese-Zugriff hat keinen Vorrang vor dem Lese-/Schreibzugriff und wird dynamisch an untergeordnete Services vererbt.				
Zugriff auf Mod-ule					
<Modulname>	Zugriff auf ein bestimmtes Modul erlauben oder verweigern. In diesem Abschnitt werden nur aktivierte Module angezeigt. Es ist nicht möglich, Zugriff auf ein derzeit deaktiviertes Modul zu gewähren oder einzuschränken.	Ja	Ja	Ja	Ja
Standardzugriff auf neue Module	Diese Option legt fest, wie nach einem Zabbix-Upgrade auf neue Module und Widgets zugegriffen werden kann. Sie gilt auch für Module und Widgets, die im Abschnitt *Administration* > *Allgemein* > *Module* hinzugefügt werden.				
Zugriff auf die API					
Aktiviert	Zugriff auf die API aktivieren/deaktivieren.	Ja	Nein	Nein	Nein

API-Methoden	Wählen Sie entweder *Zulassungsliste*, um die im Suchfeld angegebenen API-Methoden zuzulassen, oder *Sperrliste*, um sie zu verweigern. Beachten Sie, dass es nicht möglich ist, einige API-Methoden zuzulassen und andere zu verweigern. Beginnen Sie im Suchfeld mit der Eingabe des Methodennamens und wählen Sie die Methode aus der Autovervollständigungsliste aus. Sie können auch auf die Schaltfläche *Auswählen* klicken und Methoden aus der vollständigen Liste auswählen, die für diesen Benutzertyp verfügbar ist. Beachten Sie, dass Benutzer API-Methoden, die sich auf eine Aktion beziehen, nicht verwenden können, wenn eine entsprechende Aktion im Block „Zugriff auf Aktionen“ deaktiviert ist. Platzhalter werden unterstützt. Beispiele: `dashboard.*` (alle Methoden des API-Service „dashboard.“), `*` (jede Methode), `*.export` (Methoden mit dem Namen „.export“ aus allen API-Services). Wenn keine Methoden angegeben sind, verweigert die *Zulassungsliste* den Zugriff auf alle API-Methoden, während die *Sperrliste* den Zugriff auf alle für den Benutzertyp verfügbaren API-Methoden erlaubt. Um mit der *Zulassungsliste* alle Methoden zu erlauben, geben Sie `*` an. Um den API-Zugriff vollständig zu sperren, deaktivieren Sie das Kontrollkästchen *Aktiviert*, anstatt die *Zulassungsliste* leer zu lassen. Die Liste der API-Methoden gilt nicht für Methoden, die keine Authentifizierung erfordern, wie `apiinfo.version` und `user.login`.				
Zugriff auf Aktionen					
Dashboards erstellen und bearbeiten	Wenn Sie dieses Kontrollkästchen deaktivieren, werden auch die Berechtigungen zur Verwendung der API-Methoden `.create`, `.update` und `.delete` für die entsprechenden Elemente entzogen.	Ja	Ja	Ja	Nein
Karten erstellen und bearbeiten					
Wartungen erstellen und bearbeiten				Nein	
Problemkommentare hinzufügen	Wenn Sie dieses Kontrollkästchen deaktivieren, wird auch die Berechtigung entzogen, die entsprechende Aktion über die API-Methode `event.acknowledge` auszuführen.			Ja	
Schweregrad ändern					
Probleme bestätigen					
Probleme unterdrücken					
Probleme schließen					

Skripte ausführen	Wenn Sie dieses Kontrollkästchen deaktivieren, wird auch die Berechtigung zur Verwendung der API-Methode `script.execute` entzogen.				
API-Token verwalten	Wenn Sie dieses Kontrollkästchen deaktivieren, wird auch die Berechtigung zur Verwendung aller `token.`-API-Methoden entzogen.				
Geplante Berichte verwalten	Wenn Sie dieses Kontrollkästchen deaktivieren, wird auch die Berechtigung zur Verwendung aller `report.`-API-Methoden entzogen.			Nein	
SLA verwalten	Berechtigung zum Verwalten von SLA aktivieren/deaktivieren.				
„Jetzt ausführen“ auf Hosts mit Nur-Lese-Zugriff aufrufen	Benutzern erlauben, in den aktuellen Daten die Option „Jetzt ausführen“ für Datenpunkte auf Hosts mit Nur-Lese-Zugriff zu verwenden.			Ja	
Problemrangfolge ändern	Benutzern erlauben, die Problemrangfolge von Ursache zu Symptom und umgekehrt zu ändern.				
Eigene Medien erstellen und bearbeiten	Benutzern erlauben, eigene Medien zu erstellen und zu bearbeiten.				
Benutzermedien erstellen und bearbeiten	Benutzern erlauben, Medien für andere Benutzer zu erstellen und zu bearbeiten. Diese Option ist nur für Super-Admin-Benutzer verfügbar.		Nein	Nein	
„Server“ für Überwachung und Erkennung auswählen	Wenn Sie dieses Kontrollkästchen deaktivieren, wird die Erstellung von Hosts, die vom Server überwacht werden, sowie von Netzwerkerkennungsregeln, die vom Server ausgeführt werden, eingeschränkt. Außerdem wird das Testen von Datenpunkten mit dem Server eingeschränkt. Die Option *Server* wird in der Host-Konfiguration für den Parameter *Überwacht von*, in der Konfiguration der Netzwerkerkennungsregel für den Parameter *Erkannt von* und beim Testen von Datenpunkten für den Parameter *Testen mit* deaktiviert.		Ja	Nein	
Standardzugriff auf neue Aktionen	Diese Option legt fest, wie nach einem Zabbix-Upgrade auf neue Aktionen zugegriffen werden kann.		Ja	Ja	
Geräte					
Aktiviert	Zugriff auf *Benutzereinstellungen* > *Geräte* (für alle Benutzer zugänglich) und *Benutzer* > *Geräte* (nur für Super-Admins zugänglich) aktivieren/deaktivieren.	Ja	Nein	Nein	Nein
Eigene Geräte verwalten	Benutzern erlauben, ihre eigenen Geräte unter *Benutzereinstellungen* > *Geräte* zu verwalten.				
Benutzergeräte verwalten	Benutzern erlauben, Geräte anderer Benutzer unter *Benutzer* > *Geräte* zu verwalten.				
Standardzugriff auf neue Aktionen	Diese Option legt fest, wie nach einem Zabbix-Upgrade auf neue Aktionen zugegriffen werden kann.				

Siehe auch:

- Einen Benutzer konfigurieren

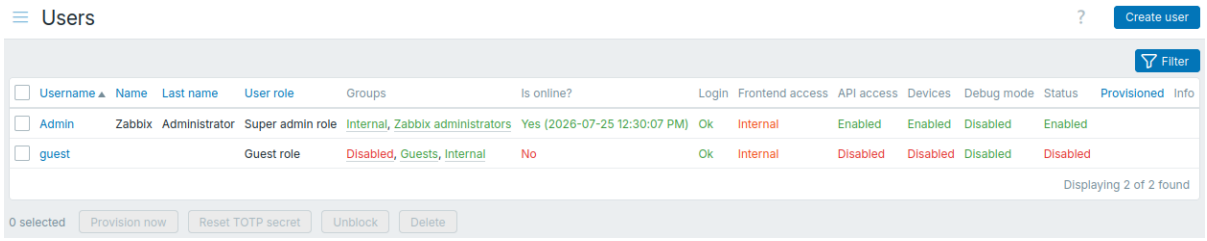
Benutzer

Übersicht

Im Bereich *Benutzer* > *Benutzer* werden die Benutzer des Systems verwaltet.

Benutzer

Eine Liste der vorhandenen Benutzer mit ihren Details wird angezeigt.



Angezeigte Daten:

Spalte	Beschreibung
Benutzername	Benutzername für die Anmeldung bei Zabbix. Wenn Sie auf den Benutzernamen klicken, wird das Benutzerkonfigurationsformular geöffnet.
Vorname	Vorname des Benutzers.
Nachname	Nachname des Benutzers.
Benutzerrolle	Die Benutzerrolle wird angezeigt.
Gruppen	Die Gruppen, deren Mitglied der Benutzer ist, werden aufgelistet. Wenn Sie auf den Namen der Benutzergruppe klicken, wird das Konfigurationsformular für Benutzergruppen geöffnet. Deaktivierte Gruppen werden rot angezeigt.
Ist online?	Der Online-Status des Benutzers wird angezeigt - *Ja* oder *Nein*. Die Zeit der letzten Benutzeraktivität wird in Klammern angezeigt.
Anmeldung	Der Anmeldestatus des Benutzers wird angezeigt - *OK* oder *Gesperrt*. Ein Benutzer kann vorübergehend gesperrt werden, wenn die in *Administration > Allgemein > Sonstige* festgelegte Anzahl erfolgloser Anmeldeversuche überschritten wird (standardmäßig fünf). Durch Klicken auf *Gesperrt* können Sie die Sperre des Benutzers aufheben.
Frontend-Zugriff	Die Zugriffsebene auf das Frontend wird angezeigt - *Systemstandard*, *Intern*, *LDAP* oder *Deaktiviert*, abhängig von der Einstellung für die gesamte Benutzergruppe.
API-Zugriff	Der Status des API-Zugriffs wird angezeigt - *Aktiviert* oder *Deaktiviert*, abhängig von der Einstellung für die Benutzerrolle. Wenn der API-Zugriff aktiviert ist, werden beim Bewegen des Mauszeigers über *Aktiviert* die API-Methoden der Benutzerrolle unter *Zulässige Methoden* oder *Verweigerte Methoden* angezeigt, abhängig vom Modus der API-Methoden. *Keine* wird angezeigt, wenn keine Methoden angegeben sind.
Geräte	Der Status des Gerätezugriffs wird angezeigt - *Aktiviert* oder *Deaktiviert*, abhängig von der Einstellung für die Benutzerrolle. Diese Spalte wird angezeigt, wenn `EnableMobileDevices` aktiviert ist.
Debug-Modus	Der Status des Debug-Modus wird angezeigt - *Aktiviert* oder *Deaktiviert*, abhängig von der Einstellung für die gesamte Benutzergruppe.
Status	Der Benutzerstatus wird angezeigt - *Aktiviert* oder *Deaktiviert*, abhängig von der Einstellung für die gesamte Benutzergruppe.
Bereitgestellt	Das Datum, an dem der Benutzer zuletzt bereitgestellt wurde, wird angezeigt. Wird für Benutzer verwendet, die durch JIT-Bereitstellung aus LDAP/SAML erstellt wurden.
Info	Informationen zu Fehlern werden angezeigt. Für Benutzer ohne Benutzergruppen wird eine gelbe Warnung angezeigt. Für Benutzer ohne Rollen sowie für Benutzer ohne Rollen und Benutzergruppen wird eine rote Warnung angezeigt.

Um einen neuen Benutzer zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Benutzer erstellen*.

Optionen zur Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen zur Massenbearbeitung:

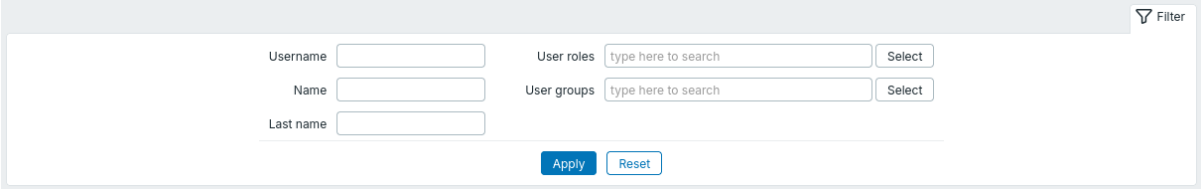
- *Jetzt bereitstellen* – Benutzerinformationen aus LDAP aktualisieren (diese Option ist nur aktiviert, wenn ein LDAP-Benutzer ausgewählt ist)
- *TOTP-Geheimnis zurücksetzen* – TOTP-Geheimnisse der Benutzer für alle TOTP-Methoden zurücksetzen und die Benutzersitzung löschen (diese Option ist nur aktiviert, wenn MFA aktiviert ist; bei Benutzern ohne TOTP-Geheimnisse wird ihre Sitzung nicht gelöscht)
- *Entsperren* – Systemzugriff für gesperrte Benutzer wieder aktivieren
- *Löschen* – Benutzer löschen

Um diese Optionen zu verwenden, aktivieren Sie die Kontrollkästchen vor den entsprechenden Benutzern und klicken Sie anschließend auf die gewünschte Schaltfläche.

Filter verwenden

Sie können den Filter verwenden, um nur die Benutzer anzuzeigen, die Sie interessieren. Für eine bessere Suchleistung werden die Daten mit nicht aufgelösten Makros durchsucht.

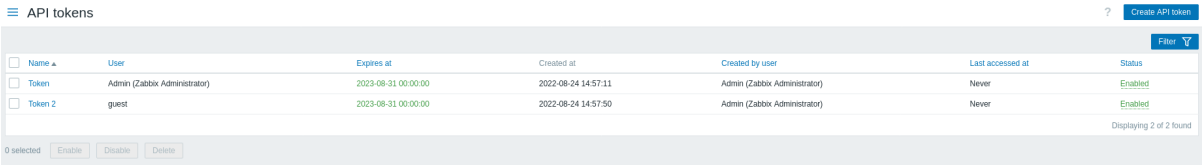
Der Link *Filter* ist oberhalb der Benutzerliste verfügbar. Wenn Sie darauf klicken, wird ein Filter eingeblendet, mit dem Sie Benutzer nach Benutzername, Vorname, Nachname, Benutzerrolle und Benutzergruppe filtern können.



API-Tokens

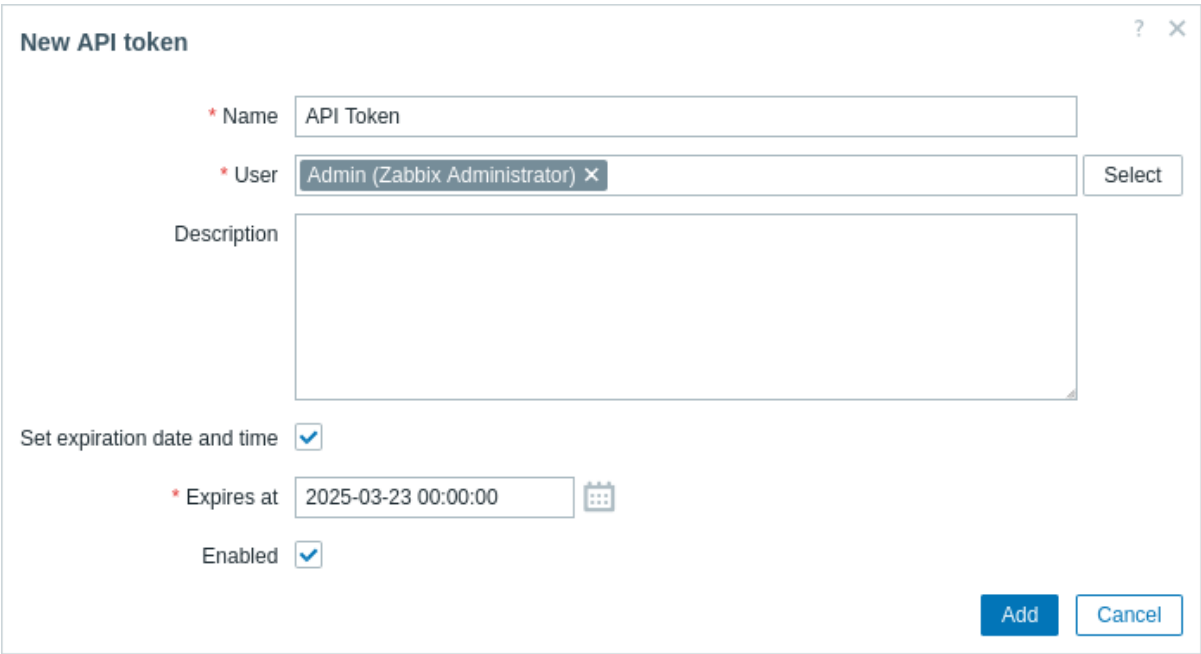
Übersicht

In diesem Abschnitt können Sie API-Token erstellen und verwalten.
 API-Token werden zur Authentifizierung in der Zabbix-API verwendet.
 Jeder API-Token ist einem einzelnen Benutzer zugewiesen und übernimmt die Berechtigungen dieses Benutzers. Mit einem API-Token authentifizierte Anfragen werden entsprechend der Rolle und den Berechtigungen des zugehörigen Benutzers autorisiert.



Sie können API-Token nach Name, Benutzern, denen die Token zugewiesen sind, Ablaufdatum, Benutzern, die Token erstellt haben, oder Status (aktiviert/deaktiviert) filtern. Klicken Sie in der Liste auf den Token-Status, um einen Token schnell zu aktivieren/deaktivieren. Sie können Token auch gesammelt aktivieren/deaktivieren, indem Sie sie in der Liste auswählen und anschließend auf die Schaltflächen *Enable/Disable* unterhalb der Liste klicken.

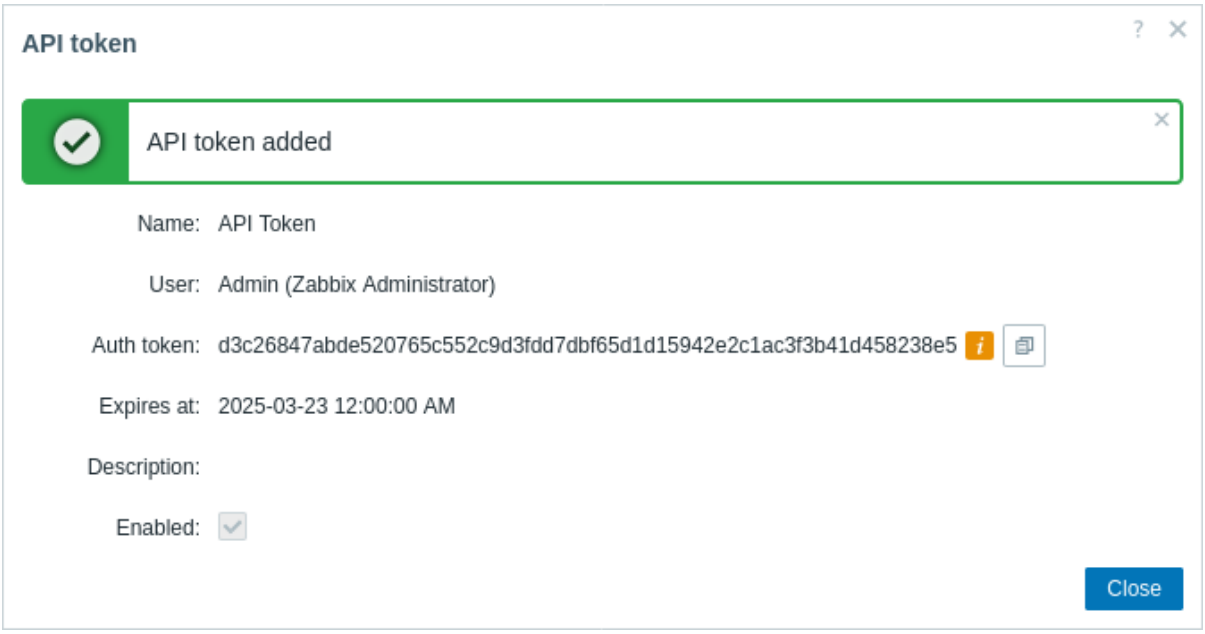
Um einen neuen Token zu erstellen, klicken Sie oben rechts auf die Schaltfläche *Create API token* und füllen Sie anschließend die erforderlichen Felder im Konfigurationsbildschirm des Tokens aus:



Parameter	Beschreibung
Name	Sichtbarer Name des Tokens.

Parameter	Beschreibung
User	Benutzer, dem der Token zugewiesen werden soll. Um schnell einen Benutzer auszuwählen, beginnen Sie mit der Eingabe des Benutzernamens sowie des Vor- oder Nachnamens und wählen Sie anschließend den gewünschten Benutzer aus der Autovervollständigungsliste aus. Alternativ können Sie auf die Schaltfläche *Select* klicken und einen Benutzer aus der vollständigen Benutzerliste auswählen. Ein Token kann nur einem Benutzer zugewiesen werden.
Description	Optionale Beschreibung des Tokens.
Set expiration date and time	Deaktivieren Sie dieses Kontrollkästchen, wenn der Token kein Ablaufdatum haben soll.
Expiry date	Klicken Sie auf das Kalendersymbol, um das Ablaufdatum des Tokens auszuwählen, oder geben Sie das Datum manuell im Format YYYY-MM-DD hh:mm:ss ein.
Enabled	Deaktivieren Sie dieses Kontrollkästchen, wenn Sie den Token im deaktivierten Zustand erstellen möchten.

Klicken Sie auf *Add*, um einen Token zu erstellen.



Kopieren Sie den Wert *Auth token* und speichern Sie ihn **vor dem Schließen der Seite** an einem sicheren Ort. Klicken Sie anschließend auf *Close*. Der Token wird in der Liste angezeigt.

Warning:
Der Wert *Auth token* kann später nicht erneut angezeigt werden. Er ist nur unmittelbar nach der Erstellung eines Tokens verfügbar. Wenn Sie einen gespeicherten Token verlieren, müssen Sie ihn neu generieren. Dadurch wird eine neue Autorisierungszeichenfolge erstellt.

Klicken Sie auf den Namen des Tokens, um den Namen, die Beschreibung, die Einstellungen für das Ablaufdatum oder den Token-Status zu bearbeiten. Beachten Sie, dass nicht geändert werden kann, welchem Benutzer der Token zugewiesen ist. Klicken Sie auf die Schaltfläche *Update*, um die Änderungen zu speichern. Wenn ein Token verloren gegangen ist oder offengelegt wurde, können Sie auf die Schaltfläche *Regenerate* klicken, um einen neuen Token-Wert zu generieren. Ein Bestätigungsdialog wird angezeigt, in dem Sie diesen Vorgang bestätigen müssen, da der zuvor generierte Token nach der Fortsetzung ungültig wird.

Benutzer ohne Zugriff auf den Menüabschnitt *Administration* können Details der ihnen zugewiesenen Token nur unter *User profile* > *API tokens* Abschnitt anzeigen und ändern, wenn *Manage API tokens* in den Berechtigungen ihrer Benutzerrolle erlaubt ist.

Authentifizierung

Übersicht

Im Abschnitt *Benutzer* > *Authentifizierung* können Sie die Benutzerauthentifizierungsmethode für Zabbix und die Anforderungen an interne Passwörter festlegen.

Die verfügbaren Authentifizierungsmethoden sind die interne Authentifizierung sowie HTTP-, LDAP-, SAML- und MFA-Authentifizierung.

Standardauthentifizierung

Standardmäßig verwendet Zabbix für alle Benutzer die **interne** Zabbix-Authentifizierung.

Es ist möglich, die Standardauthentifizierung systemweit auf das LDAP-System umzustellen. Navigieren Sie dazu zur Registerkarte *LDAP* und konfigurieren Sie die LDAP-Parameter. Kehren Sie anschließend zur Registerkarte *Authentifizierung* zurück und stellen Sie den Auswahlbereich *Standardauthentifizierung* auf LDAP um.

Beachten Sie, dass die Authentifizierungsmethode auf Ebene der Benutzergruppe angepasst werden kann. Auch wenn die LDAP-Authentifizierung global festgelegt ist, können einige Benutzergruppen weiterhin über Zabbix authentifiziert werden. Für diese Gruppen muss der Frontend-Zugriff auf „Intern“ gesetzt sein.

Es ist auch möglich, die LDAP-Authentifizierung nur für bestimmte Benutzergruppen zu aktivieren, wenn global die interne Authentifizierung verwendet wird. In diesem Fall können LDAP-Authentifizierungsdetails angegeben und für bestimmte Benutzergruppen verwendet werden. Für diese Gruppen muss der Frontend-Zugriff anschließend auf LDAP gesetzt werden. Wenn ein Benutzer mindestens einer Benutzergruppe mit LDAP-Authentifizierung angehört, kann dieser Benutzer die interne Authentifizierungsmethode nicht verwenden.

Die Authentifizierungsmethoden HTTP, SAML 2.0 und MFA können zusätzlich zur Standardauthentifizierung verwendet werden.

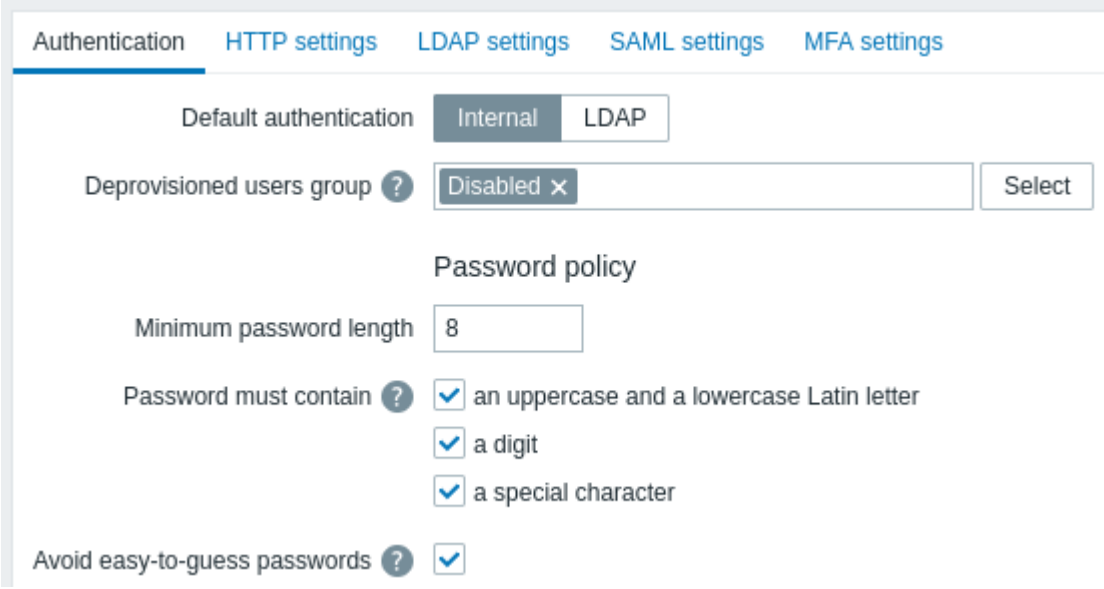
Zabbix unterstützt die Just-in-Time-Bereitstellung (JIT), mit der Sie Benutzerkonten in Zabbix erstellen und bereitstellen können, wenn sich ein externer Benutzer zum ersten Mal authentifiziert. Die JIT-Bereitstellung wird für LDAP und SAML unterstützt.

Siehe auch:

- HTTP-Authentifizierung
- LDAP-Authentifizierung
- SAML-Authentifizierung
- MFA-Authentifizierung

Konfiguration

Auf der Registerkarte *Authentifizierung* können Sie die standardmäßige Authentifizierungsmethode festlegen, eine Gruppe für Benutzer mit aufgehobener Bereitstellung angeben und Anforderungen an die Passwortkomplexität für Zabbix-Benutzer festlegen.



Konfigurationsparameter:

Parameter	Beschreibung
Standardauthentifizierung	Wählen Sie die standardmäßige Authentifizierungsmethode für Zabbix aus – *Intern* oder *LDAP*.
Gruppe für Benutzer mit aufgehobener Bereitstellung	Geben Sie eine Benutzergruppe für Benutzer mit aufgehobener Bereitstellung an. Diese Einstellung ist nur für die JIT-Bereitstellung erforderlich und betrifft Benutzer, die in Zabbix aus LDAP- oder SAML-Systemen erstellt wurden, aber nicht mehr bereitgestellt werden müssen. Es muss eine deaktivierte Benutzergruppe angegeben werden.
Mindestpasswortlänge	Standardmäßig ist die Mindestpasswortlänge auf 8 festgelegt. Unterstützter Bereich: 1–70. Beachten Sie, dass Passwörter mit mehr als 72 Zeichen abgeschnitten werden.

Parameter	Beschreibung
Passwort muss enthalten	Aktivieren Sie ein oder mehrere Kontrollkästchen, um die Verwendung der angegebenen Zeichen in einem Passwort zu verlangen: - einen lateinischen Groß- und einen Kleinbuchstaben - eine Ziffer - ein Sonderzeichen Bewegen Sie den Mauszeiger über das Fragezeichen, um einen Hinweis mit der Liste der Zeichen für jede Option anzuzeigen.
Leicht zu erratende Passwörter vermeiden	Wenn diese Option aktiviert ist, wird ein Passwort anhand der folgenden Anforderungen geprüft: - darf nicht den Namen, Nachnamen oder Benutzernamen des Benutzers enthalten - darf keines der häufig verwendeten oder kontextspezifischen Passwörter sein. Die Liste der häufig verwendeten und kontextspezifischen Passwörter wird automatisch aus der Liste der „Top 100k passwords“ des NCSC, der Liste der „Top 1M passwords“ von SecLists und der Liste der kontextspezifischen Zabbix-Passwörter generiert. Interne Benutzer dürfen keine in dieser Liste enthaltenen Passwörter festlegen, da solche Passwörter aufgrund ihrer häufigen Verwendung als schwach gelten.

Änderungen an den Anforderungen an die Passwortkomplexität wirken sich nicht auf bestehende Benutzerpasswörter aus. Wenn ein bestehender Benutzer jedoch ein Passwort ändern möchte, muss das neue Passwort die aktuellen Anforderungen erfüllen. Ein Hinweis mit der Liste der Anforderungen wird neben dem Feld *Passwort* im Benutzerprofil und im Benutzerkonfigurationsformular angezeigt, das über das Menü *Benutzer* > *Benutzer* erreichbar ist.

HTTP

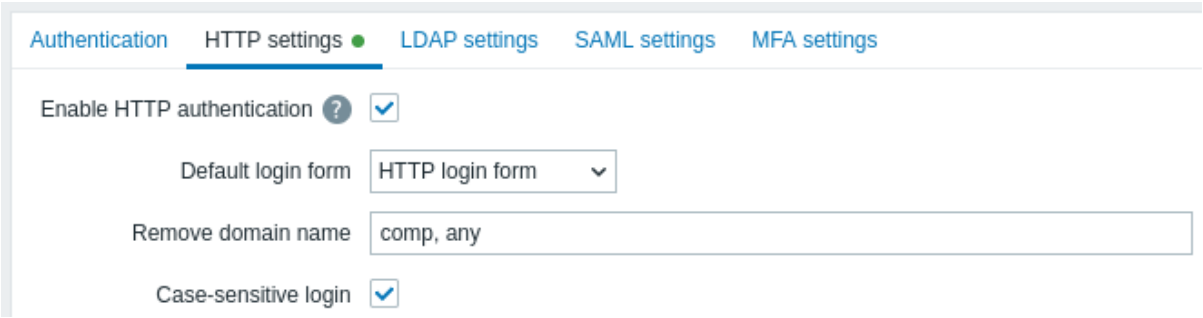
Übersicht

Die Authentifizierung über einen HTTP- oder webbasierten Server (zum Beispiel: BasicAuthentication, NTLM/Kerberos) kann verwendet werden, um Benutzernamen und Passwörter zu überprüfen. Beachten Sie jedoch, dass der Benutzer auch in Zabbix vorhanden sein muss. Sein Zabbix-Passwort wird allerdings nicht verwendet.

Attention:
Vorsicht! Stellen Sie sicher, dass die Authentifizierung des Webservers konfiguriert ist und ordnungsgemäß funktioniert, bevor Sie sie aktivieren.

Note:
Die HTTP-Authentifizierung kann deaktiviert werden, indem Sie `$ZBX_FEATURE_FLAGS['http_auth_enabled'] = false` in der Frontend-Konfigurationsdatei (`zabbix.conf.php`) festlegen. In diesem Fall wird die Registerkarte *HTTP-Einstellungen* mit den Optionen für die HTTP-Authentifizierung ausgeblendet.

Konfiguration



Konfigurationsparameter:

Parameter	Beschreibung
HTTP-Authentifizierung aktivieren	Aktivieren Sie das Kontrollkästchen, um die HTTP-Authentifizierung zu aktivieren. Wenn Sie den Mauszeiger über bewegen, wird ein Hinweisfeld angezeigt, das darauf aufmerksam macht, dass bei einer Authentifizierung durch den Webserver alle Benutzer (auch Benutzer mit auf LDAP/Intern gesetztem Frontend-Zugriff) durch den Webserver und nicht durch Zabbix authentifiziert werden.
Standard-Anmeldeformular	Geben Sie an, wohin nicht authentifizierte Benutzer weitergeleitet werden sollen: **Zabbix-Anmeldeformular** – Standard-Anmeldeseite von Zabbix. **HTTP-Anmeldeformular** – HTTP-Anmeldeseite. Es wird empfohlen, die webserverbasierte Authentifizierung nur für die Seite `index_http.php` zu aktivieren. Wenn *Standard-Anmeldeformular* auf „HTTP-Anmeldeseite“ gesetzt ist, wird der Benutzer automatisch angemeldet, wenn das Authentifizierungsmodul des Webservers eine gültige Benutzeranmeldung in der Variable `$_SERVER` setzt. Unterstützte Schlüssel von `$_SERVER` sind `PHP_AUTH_USER`, `REMOTE_USER`, `AUTH_USER`.
Domänennamen entfernen	Eine durch Kommas getrennte Liste von Domänennamen, die aus dem Benutzernamen entfernt werden sollen. Beispiel: `comp,any` – wenn der Benutzername „Admin@any“ oder „comp\Admin“ lautet, wird der Benutzer als „Admin“ angemeldet; bei „notacompany\Admin“ wird die Anmeldung verweigert.
Groß-/Kleinschreibung bei der Anmeldung berücksichtigen	Deaktivieren Sie das Kontrollkästchen, um die Berücksichtigung der Groß-/Kleinschreibung bei der Anmeldung für Benutzernamen zu deaktivieren (standardmäßig aktiviert). Wenn die Berücksichtigung der Groß-/Kleinschreibung deaktiviert ist, ist beispielsweise eine Anmeldung als „admin“ möglich, auch wenn der Zabbix-Benutzer „Admin“ oder „ADMIN“ lautet. Beachten Sie, dass bei deaktivierter Berücksichtigung der Groß-/Kleinschreibung und mehreren Zabbix-Benutzern mit ähnlichen Benutzernamen (z. B. Admin und admin) die Anmeldung für diese Benutzer immer mit der folgenden Fehlermeldung verweigert wird: „Authentifizierung fehlgeschlagen: Die angegebenen Anmeldedaten sind nicht eindeutig.“

Note:
Für interne Benutzer, die sich nicht mit HTTP-Anmeldedaten anmelden können (wenn das HTTP-Anmeldeformular als Standard festgelegt ist) und dadurch den 401-Fehler erhalten, können Sie der Direktive für die Basisauthentifizierung die Zeile `ErrorDocument 401 /index.php?form=default` hinzufügen. Dadurch wird das reguläre Zabbix-Anmeldeformular geöffnet.

LDAP

Übersicht

Externe LDAP-Authentifizierung kann verwendet werden, um Benutzernamen und Passwörter zu überprüfen.

Die LDAP-Authentifizierung von Zabbix funktioniert mindestens mit Microsoft Active Directory und OpenLDAP.

Wenn nur die LDAP-Anmeldung konfiguriert ist, muss der Benutzer auch in Zabbix vorhanden sein, sein Zabbix-Passwort wird jedoch nicht verwendet. Wenn die Authentifizierung erfolgreich ist, gleicht Zabbix einen lokalen Benutzernamen mit dem von LDAP zurückgegebenen Attribut für den Benutzernamen ab.

Benutzerbereitstellung

Es ist möglich, die JIT-Benutzerbereitstellung (Just-in-Time) für LDAP-Benutzer zu konfigurieren. In diesem Fall muss ein Benutzer nicht bereits in Zabbix vorhanden sein. Das Benutzerkonto kann erstellt werden, wenn sich der Benutzer zum ersten Mal bei Zabbix anmeldet.

Wenn ein LDAP-Benutzer seinen LDAP-Benutzernamen und sein LDAP-Passwort eingibt, prüft Zabbix, ob dieser Benutzer auf dem *Standard*-LDAP-Server vorhanden ist. Wenn der Benutzer vorhanden ist und noch kein Konto in Zabbix besitzt, wird ein neuer Benutzer in Zabbix erstellt und der Benutzer kann sich anmelden.

Ein über JIT-Bereitstellung erstellter Benutzer wird dem LDAP-Server (Verzeichnis) zugeordnet, der zum Zeitpunkt der Erstellung als Standard festgelegt ist. Wenn der Standard-LDAP-Server später geändert wird, wird der mit bereits bereitgestellten Benutzern verknüpfte LDAP-Server nicht geändert oder aktualisiert.

Attention:
Wenn die JIT-Bereitstellung aktiviert ist, muss auf der Registerkarte *Authentifizierung* eine Benutzergruppe für Benutzer angegeben werden, deren Bereitstellung aufgehoben wurde.

Die JIT-Bereitstellung ermöglicht außerdem, bereitgestellte Benutzerkonten auf Grundlage von Änderungen in LDAP zu aktualisieren. Wenn ein Benutzer beispielsweise von einer LDAP-Gruppe in eine andere verschoben wird, wird der Benutzer auch in Zabbix von einer Gruppe in eine andere verschoben. Wenn ein Benutzer aus einer LDAP-Gruppe entfernt wird, wird er auch in Zabbix aus der Gruppe entfernt und, sofern er keiner anderen Gruppe angehört, der Benutzergruppe für Benutzer hinzugefügt, deren Bereitstellung aufgehoben wurde. Bereitgestellte Benutzerkonten werden entsprechend dem konfigurierten Bereitstellungsintervall oder bei der Anmeldung des Benutzers bei Zabbix aktualisiert.

Beachten Sie, dass die Hintergrundbereitstellung vom Zabbix Frontend durchgeführt wird, wenn der Benutzer damit interagiert oder zumindest eine Frontend-Seite im Browser geöffnet hat. Für die Benutzerbereitstellung gibt es keine dedizierten Hintergrundprozesse.

LDAP unterstützt drei Möglichkeiten, eine Bindung an das Verzeichnis für die Authentifizierung und für Suchvorgänge herzustellen:

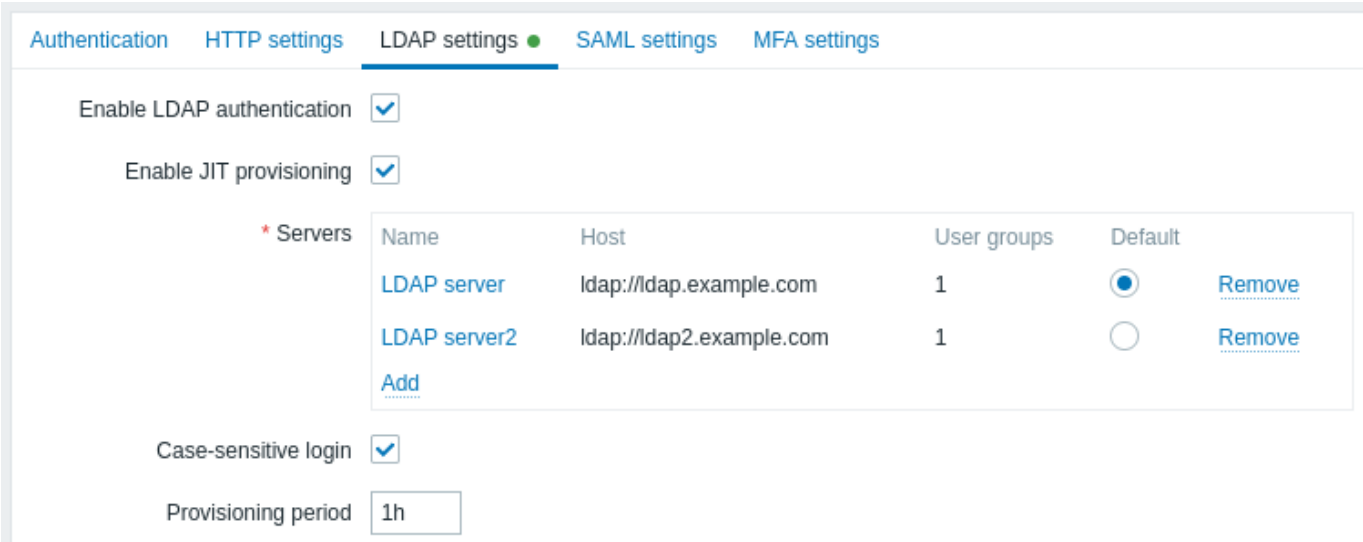
- Anonyme Bindung – es werden kein *Bind DN* und kein *Bind-Passwort* angegeben und der LDAP-Server erlaubt anonyme Abfragen.
- Dedizierter Bind-Benutzer (Dienstkonto) – ein bestimmtes LDAP-Konto wird unter *Bind DN* / *Bind-Passwort* festgelegt und von Zabbix für Suchvorgänge und die Bereitstellung verwendet. Dies ist die empfohlene und flexibelste Option, da Zabbix Suchvorgänge und Hintergrundbereitstellungen ohne die Anmeldedaten des Endbenutzers durchführen kann.
- Direkte Benutzerbindung – Zabbix stellt die Bindung mit den Anmeldedaten her, die der Benutzer bei der Anmeldung eingibt (kein *Bind DN* / *Bind-Passwort* konfiguriert). Dies wird konfiguriert, indem ein Platzhalter wie `uid=%{user}` in der Basis-DN angegeben wird. In diesem Modus hat Zabbix nur während der interaktiven Anmeldung Zugriff auf das Passwort des Benutzers. Daher können Bereitstellungsaktionen, die eine Authentifizierung bei LDAP außerhalb der Anmeldesitzung des Benutzers erfordern (beispielsweise über die Schaltfläche *Jetzt bereitstellen* oder beim Ausführen einer Hintergrundbereitstellung, wenn der Benutzer nicht aktiv angemeldet ist), keine Authentifizierung durchführen und funktionieren daher nicht. Die Bereitstellung und Aktualisierung bei direkter Benutzerbindung erfolgt nur zu dem Zeitpunkt, an dem sich der Benutzer anmeldet.

Mehrere Server

Bei Bedarf können mehrere LDAP-Server definiert werden. So kann beispielsweise ein anderer Server verwendet werden, um eine andere Benutzergruppe zu authentifizieren. Sobald LDAP-Server konfiguriert sind, kann in der Konfiguration der Benutzergruppe der erforderliche LDAP-Server für die jeweilige Benutzergruppe ausgewählt werden.

Wenn ein Benutzer mehreren Benutzergruppen und mehreren LDAP-Servern angehört, wird für die Authentifizierung der erste Server in der nach Namen aufsteigend sortierten Liste der LDAP-Server verwendet.

Konfiguration

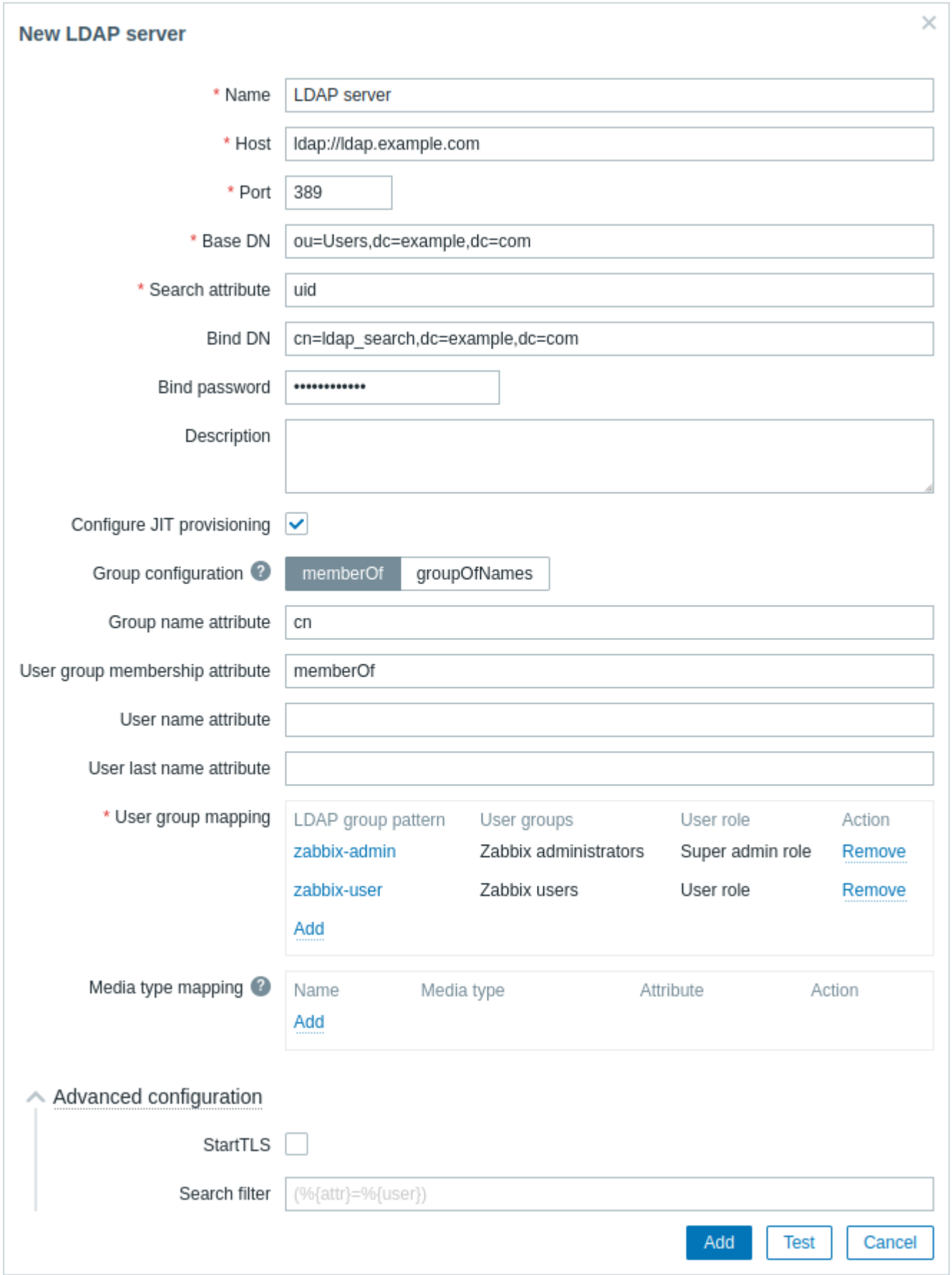


Konfigurationsparameter:

Parameter	Beschreibung
LDAP-Authentifizierung aktivieren	Aktivieren Sie das Kontrollkästchen, um die LDAP-Authentifizierung zu aktivieren.

Parameter	Beschreibung
JIT-Bereitstellung aktivieren	Aktivieren Sie das Kontrollkästchen, um die JIT-Bereitstellung zu aktivieren.
Server	Klicken Sie auf *Hinzufügen*, um einen LDAP-Server zu konfigurieren (siehe LDAP-Serverkonfiguration weiter unten).
Anmeldung unter Beachtung der Groß-/Kleinschreibung	Deaktivieren Sie das Kontrollkästchen, um die Beachtung der Groß-/Kleinschreibung bei Benutzernamen zu deaktivieren (standardmäßig aktiviert). Wenn die Beachtung der Groß-/Kleinschreibung deaktiviert ist, ist beispielsweise eine Anmeldung als „admin“ möglich, auch wenn der Zabbix-Benutzer „Admin“ oder „ADMIN“ lautet. Beachten Sie, dass die Anmeldung für Benutzer immer mit der folgenden Fehlermeldung abgelehnt wird, wenn die Beachtung der Groß-/Kleinschreibung deaktiviert ist und mehrere Zabbix-Benutzer mit ähnlichen Benutzernamen vorhanden sind (z. B. Admin und admin): „Authentication failed: supplied credentials are not unique.“
Bereitstellungszeitraum	Legen Sie den Bereitstellungszeitraum fest, d. h. wie oft der angemeldete Benutzer während der Arbeit mit dem Frontend bereitgestellt wird.

LDAP-Server-Konfiguration



LDAP-Server-Konfigurationsparameter:

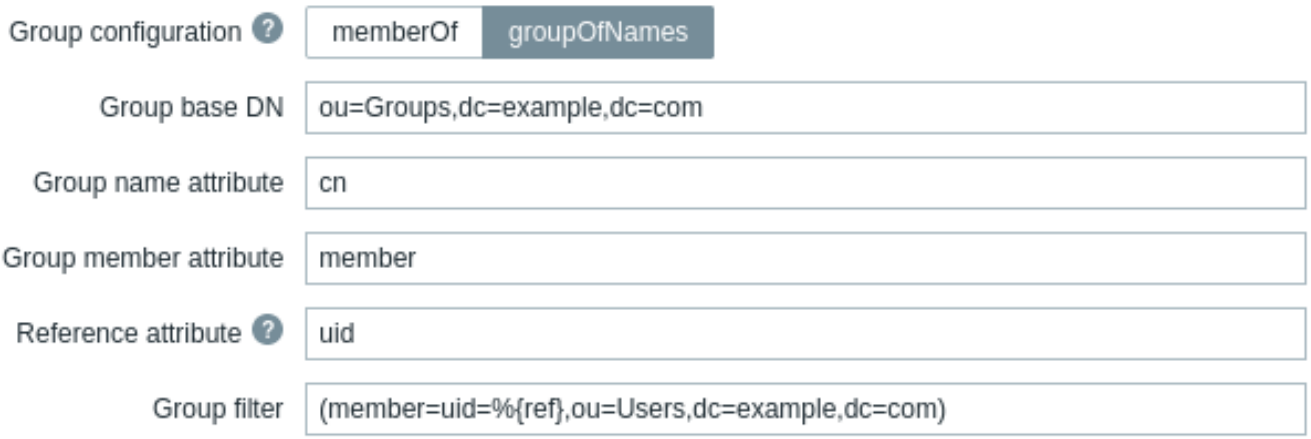
Parameter	Beschreibung
Name	Name des LDAP-Servers in der Zabbix-Konfiguration.

Parameter	Beschreibung
Host	Hostname, IP oder URI des LDAP-Servers. Beispiele: ldap.example.com, 127.0.0.1, ldap://ldap.example.com Für einen sicheren LDAP-Server verwenden Sie das Protokoll *ldaps* und einen Hostnamen. Beispiel: ldaps://ldap.example.com Mit OpenLDAP 2.x.x und höher kann eine vollständige LDAP-URI der Form ldap://hostname:port oder ldaps://hostname:port verwendet werden.
Port	Port des LDAP-Servers. Standard ist 389. Für eine sichere LDAP-Verbindung ist die Portnummer normalerweise 636. Wird bei Verwendung vollständiger LDAP-URIs nicht verwendet.
Base DN	Basis-Pfad zu Benutzerkonten auf dem LDAP-Server: ou=Users,ou=system (für OpenLDAP), DC=company,DC=com (für Microsoft Active Directory) uid=%{user},dc=example,dc=com (für direkte Benutzerbindung, siehe Hinweis unten)
Search attribute	LDAP-Kontoattribut, das für die Suche verwendet wird: uid (für OpenLDAP), sAMAccountName (für Microsoft Active Directory)
Bind DN	LDAP-Konto für die Bindung und Suche auf dem LDAP-Server, Beispiele: uid=ldap_search,ou=system (für OpenLDAP), CN=ldap_search,OU=user_group,DC=company,DC=com (für Microsoft Active Directory) Anonyme Bindung wird ebenfalls unterstützt. Beachten Sie, dass anonyme Bindung potenziell die Domänenkonfiguration für unbefugte Benutzer öffnet (Informationen über Benutzer, Computer, Server, Gruppen, Dienste usw.). Deaktivieren Sie aus Sicherheitsgründen anonyme Bindungen auf LDAP-Hosts und verwenden Sie stattdessen authentifizierten Zugriff.
Bind password	LDAP-Passwort des Kontos für die Bindung und Suche auf dem LDAP-Server.
Description	Beschreibung des LDAP-Servers.
Configure JIT provisioning	Aktivieren Sie dieses Kontrollkästchen, um Optionen für JIT-Provisioning anzuzeigen.
Group configuration	Wählen Sie die Methode zur Gruppenkonfiguration: **memberOf** - durch Suche nach Benutzern und ihrem Gruppenmitgliedschaftsattribut **groupOfNames** - durch Suche nach Gruppen über das member-Attribut Beachten Sie, dass memberOf vorzuziehen ist, da es schneller ist; verwenden Sie groupOfNames, wenn Ihr LDAP-Server `memberOf` nicht unterstützt oder eine Gruppenfilterung erforderlich ist.
Group name attribute	Geben Sie das Attribut an, aus dem der Gruppenname aus allen Objekten im Attribut `memberOf` ermittelt wird (siehe Feld *User group membership attribute*) Der Gruppenname ist für das Mapping von Benutzergruppen erforderlich.
User group membership attribute	Geben Sie das Attribut an, das Informationen über die Gruppen enthält, denen der Benutzer angehört (z. B. `memberOf`). Zum Beispiel kann das Attribut memberOf Informationen wie diese enthalten: `memberOf=cn=zabbix-admin,ou=Groups,dc=example,dc=com` Dieses Feld ist nur für die Methode memberOf verfügbar.
User name attribute	Geben Sie das Attribut an, das den Vornamen des Benutzers enthält.
User last name attribute	Geben Sie das Attribut an, das den Nachnamen des Benutzers enthält.
User group mapping	Ordnen Sie ein LDAP-Benutzergruppenmuster einer Zabbix-Benutzergruppe und einer Benutzerrolle zu. Dies ist erforderlich, um festzulegen, welche Benutzergruppe/Rolle der bereitgestellte Benutzer in Zabbix erhält. Klicken Sie auf *Add*, um ein Mapping hinzuzufügen. Das Feld *LDAP group pattern* unterstützt Platzhalter. Der Gruppenname muss mit einer vorhandenen Gruppe übereinstimmen. Wenn ein LDAP-Benutzer mehreren Zabbix-Benutzergruppen entspricht, wird der Benutzer Mitglied aller dieser Gruppen. Wenn ein Benutzer mehreren Zabbix-Benutzerrollen entspricht, erhält er die Rolle mit dem höchsten Berechtigungsniveau unter ihnen.
Media type mapping	Ordnen Sie die LDAP-Medien-Attribute des Benutzers (z. B. E-Mail) den Zabbix-Benutzermedien für den Versand von Benachrichtigungen zu (der Attributwert wird als Feld *Send to* des Mediums verwendet).
Advanced configuration	Klicken Sie auf die Überschrift *Advanced configuration*, um erweiterte Konfigurationsoptionen anzuzeigen (siehe unten).
StartTLS	Aktivieren Sie das Kontrollkästchen, um beim Verbinden mit dem LDAP-Server die StartTLS-Operation zu verwenden. Die Verbindung schlägt fehl, wenn der Server StartTLS nicht unterstützt. StartTLS kann nicht mit Servern verwendet werden, die das Protokoll *ldaps* verwenden.

Parameter	Beschreibung
Search filter	Definieren Sie beim Authentifizieren eines Benutzers in LDAP eine benutzerdefinierte Zeichenfolge. Die folgenden Platzhalter werden unterstützt: `%{attr}` - Name des Suchattributs (uid, sAMAccountName) `%{user}` - Benutzername-Wert des zu authentifizierenden Benutzers Um beispielsweise eine case-sensitive Suche in einer case-insensitive LDAP- oder Microsoft-Active-Directory-Umgebung durchzuführen, kann die Zeichenfolge wie folgt definiert werden: `(%{attr}:caseExactMatch:=%{user})`. Wenn der Filter nicht angepasst wird, verwendet LDAP standardmäßig: `(%{attr}=%{user})`.

Note:
Um einen LDAP-Server für die **direkte Benutzerbindung** zu konfigurieren, fügen Sie dem Parameter *Base DN* ein Attribut uid=%{user} hinzu (zum Beispiel *uid=%{user},dc=example,dc=com*) und lassen Sie die Parameter *BindDN* und *Bind password* leer. Beim Authentifizieren wird der Platzhalter %{user} durch den beim Anmelden eingegebenen Benutzernamen ersetzt.
 Bei direkter Benutzerbindung hat Zabbix nur während der interaktiven Anmeldung Zugriff auf die Anmeldedaten des Benutzers. Daher ignorieren Bereitstellungsaufgaben, die ohne interaktive Anmeldung ausgeführt werden (zum Beispiel die Schaltfläche *Provision now* oder die API-Methode `user.provision`), Benutzer, die sich über direkte Benutzerbindung authentifizieren, da Zabbix nicht in ihrem Namen binden kann. Verwenden Sie anonyme Bindung oder einen dedizierten Bind-Benutzer (Servicekonto), wenn Sie benötigen, dass Provisioning und LDAP-Suchen im Frontend funktionieren, ohne dass der Endbenutzer angemeldet sein muss.

Die folgenden Felder sind spezifisch für "groupOfNames" als Methode *Group configuration*:



Parameter	Beschreibung
Group base DN	Basis-Pfad zu den Gruppen auf dem LDAP-Server.
Group name attribute	Geben Sie das Attribut an, aus dem der Gruppenname im angegebenen Basis-Pfad zu den Gruppen ermittelt wird. Der Gruppenname ist für das Mapping von Benutzergruppen erforderlich.
Group member attribute	Geben Sie das Attribut an, das Informationen über die Mitglieder der Gruppe in LDAP enthält (z. B. `member`).
Reference attribute	Geben Sie das Referenzattribut für den Gruppenfilter an (siehe Feld *Group filter*). Verwenden Sie dann `%{ref}` im Gruppenfilter, um Werte für das hier angegebene Attribut zu erhalten.
Group filter	Geben Sie den Filter an, um die Gruppe abzurufen, der der Benutzer angehört. Zum Beispiel stimmt `(member=uid=%{ref},ou=Users,dc=example,dc=com)` mit "User1" überein, wenn das member-Attribut der Gruppe `uid=User1,ou=Users,dc=example,dc=com` ist, und gibt die Gruppe zurück, der "User1" angehört.

Warning:
Falls Probleme mit Zertifikaten auftreten, müssen Sie möglicherweise eine Zeile `TLS_REQCERT allow` zur Konfigurationsdatei `/etc/openldap/ldap.conf` hinzufügen, damit eine sichere LDAP-Verbindung (ldaps) funktioniert. Dies kann die Sicherheit der Verbindung zum LDAP-Katalog verringern.

Note:
Es wird empfohlen, ein separates LDAP-Konto (*Bind DN*) für die Bindung und Suche auf dem LDAP-Server mit minimalen Berechtigungen in LDAP zu erstellen, anstatt echte Benutzerkonten zu verwenden (die für die Anmeldung im Zabbix-Frontend genutzt werden).
 Ein solcher Ansatz bietet mehr Sicherheit und erfordert keine Änderung des *Bind password*, wenn der Benutzer sein eigenes Passwort auf dem LDAP-Server ändert.
 In der obigen Tabelle ist dies das Konto *ldap_search*.

Zugriff testen

Mit der Schaltfläche *Test* können Sie den Benutzerzugriff testen:

Parameter	Beschreibung
Login	Zu testender LDAP-Benutzername (mit dem aktuellen Benutzernamen aus dem Zabbix Frontend vorausgefüllt). Dieser Benutzername muss auf dem LDAP-Server vorhanden sein. Zabbix aktiviert die LDAP-Authentifizierung nicht, wenn der Testbenutzer nicht authentifiziert werden kann.
User password	Zu testendes LDAP-Benutzerpasswort.

SAML

Übersicht

SAML 2.0 Authentifizierung kann verwendet werden, um sich bei Zabbix anzumelden.

Wenn die SAML-Anmeldung ohne JIT-Benutzerbereitstellung konfiguriert ist, muss der Benutzer bereits in Zabbix vorhanden sein; das Zabbix-Passwort des Benutzers wird nicht für die Authentifizierung verwendet. Nach erfolgreicher SAML-Authentifizierung gleicht Zabbix das von SAML empfangene Attribut für den Benutzernamen mit einem vorhandenen lokalen Zabbix-Benutzernamen ab.

Benutzerbereitstellung

Es ist möglich, JIT-**Benutzerbereitstellung** (just-in-time) für SAML-Benutzer zu konfigurieren. In diesem Fall ist es nicht erforderlich, dass ein Benutzer bereits in Zabbix existiert. Das Benutzerkonto kann erstellt werden, wenn sich der Benutzer zum ersten Mal bei Zabbix anmeldet.

Attention:
Wenn die JIT-Bereitstellung aktiviert ist, muss im Reiter *Authentifizierung* eine Benutzergruppe für deprovisionierte Benutzer angegeben werden.

Zusätzlich zur JIT-Bereitstellung ist es auch möglich, die SCIM-Bereitstellung (System for Cross-domain Identity Management) zu aktivieren und zu konfigurieren – eine *kontinuierliche* Benutzerkontoverwaltung für Benutzer, die durch die Benutzerbereitstellung erstellt wurden. Die SCIM-Bereitstellung erfordert zur Authentifizierung bei Zabbix ein Zabbix-API-Token (mit Super-Admin-Berechtigungen).

Wenn ein Benutzer beispielsweise von einer SAML-Gruppe in eine andere verschoben wird, wird der Benutzer auch in Zabbix von einer Gruppe in eine andere verschoben; wenn ein Benutzer aus einer SAML-Gruppe entfernt wird, wird der Benutzer auch in Zabbix aus der Gruppe entfernt und, falls er keiner anderen Gruppe angehört, der Benutzergruppe für deprovisionierte Benutzer hinzugefügt.

Wenn SCIM aktiviert und konfiguriert ist, wird ein SAML-Benutzer in dem Moment bereitgestellt, in dem sich der Benutzer bei Zabbix anmeldet, und anschließend kontinuierlich auf Grundlage von Änderungen in SAML aktualisiert. Bereits vorhandene SAML-Benutzer werden nicht bereitgestellt, und nur bereitgestellte Benutzer werden aktualisiert. Beachten Sie, dass einem Benutzer bei der Bereitstellung oder Aktualisierung nur gültige Medien hinzugefügt werden.

Wenn SCIM nicht aktiviert ist, wird ein SAML-Benutzer in dem Moment bereitgestellt (und später aktualisiert), in dem sich der Benutzer bei Zabbix anmeldet.

Note:
Wenn die SAML-Authentifizierung aktiviert ist, können Benutzer zwischen lokaler Anmeldung und Anmeldung über SAML Single Sign-on wählen. Wenn JIT-Bereitstellung verwendet wird, ist nur Single Sign-on möglich.

Einrichten des Identitätsanbieters

Um mit Zabbix zu arbeiten, muss ein SAML-Identitätsanbieter (onelogin.com, auth0.com, okta.com usw.) wie folgt konfiguriert werden:

- *Assertion Consumer URL* sollte auf `<path_to_zabbix_ui>/index_sso.php?acs` gesetzt werden
- *Single Logout URL* sollte auf `<path_to_zabbix_ui>/index_sso.php?sls` gesetzt werden

Beispiele für `<path_to_zabbix_ui>`: `https://example.com/zabbix/ui`, `http://another.example.com/zabbix`, `http://<any_public_ip_address>/zabbix`

Zabbix einrichten

Attention:
Wenn Sie die SAML-Authentifizierung im Frontend verwenden möchten, muss php-openssl installiert werden.

Für die Verwendung der SAML-Authentifizierung sollte Zabbix wie folgt konfiguriert werden:

1. Stellen Sie den privaten SP-Schlüssel und die Zertifikate bereit. Ein IdP-Zertifikat muss bereitgestellt werden (Dateitext einfügen oder Datei auswählen), um SAML über das Frontend zu aktivieren. Ein SP-Zertifikat und ein privater SP-Schlüssel müssen bereitgestellt werden, wenn die Optionen *Sign* oder *Encrypt* ausgewählt sind.

Wenn `$SSO['CERT_STORAGE'] = 'database'` in der Frontend-Konfigurationsdatei (`zabbix.conf.php`) festgelegt ist, können Sie den Zertifikatstext einfügen oder die Zertifikatsdatei während der SAML-Konfiguration im Frontend hochladen – Dateien im Dateisystem sind nicht erforderlich.

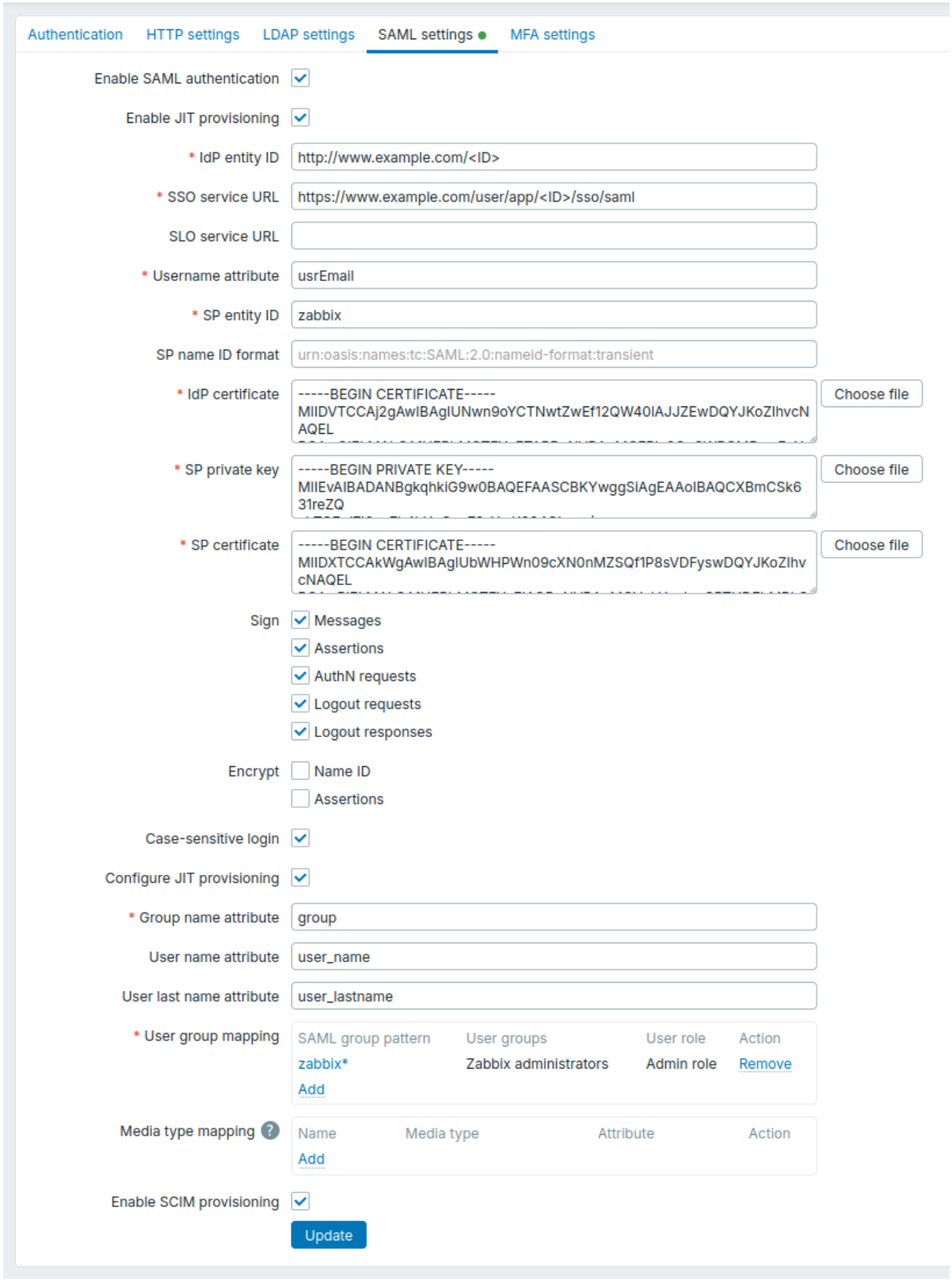
Wenn `$SSO['CERT_STORAGE'] = 'file'` in der Konfigurationsdatei festgelegt ist, muss das Zertifikat im Dateisystem verfügbar sein (standardmäßig in `ui/conf/certs` oder unter dem in der Konfigurationsdatei festgelegten Pfad), und das Frontend speichert keine Zertifikate in der Datenbank. Beachten Sie, dass der Dateispeicher verwendet wird, wenn `$SSO['CERT_STORAGE']` nicht festgelegt oder auskommentiert ist.

Standardmäßig sucht Zabbix an den folgenden Speicherorten:

- ui/conf/certs/sp.key - Datei mit dem privaten SP-Schlüssel
- ui/conf/certs/sp.crt - SP-Zertifikatsdatei
- ui/conf/certs/idp.crt - IDP-Zertifikatsdatei

Attention:
Zertifikatsmaterial kann über das Frontend importiert werden, indem Text eingefügt oder Dateien hochgeladen werden. Obwohl unverschlüsselte Importe in einigen Umgebungen aus Gründen der Benutzerfreundlichkeit zulässig sind, wird die Verwendung eines verschlüsselten Transports dringend empfohlen (aktivieren Sie beispielsweise HTTPS/TLS für das Zabbix-Frontend). Wenn SAML-Zertifikate/-Schlüssel in der Datenbank gespeichert werden, aktivieren Sie TLS/SSL für Datenbankverbindungen, um den Datenverkehr zwischen der Anwendung und der Datenbank zu schützen.
 Verschlüsseln Sie Datenbanksicherungen, die SAML-Zertifikate oder private Schlüssel enthalten können, und beschränken Sie den Zugriff auf Sicherungsdateien sowie auf die Datenbanktabellen, in denen SAML-Material gespeichert wird. Beschränken Sie die Berechtigungen des Datenbankbenutzers auf das erforderliche Minimum.

2. Alle wichtigen Einstellungen können im Zabbix-Frontend konfiguriert werden. Es ist jedoch möglich, zusätzliche Einstellungen in der Konfigurationsdatei festzulegen.



Im Zabbix-Frontend verfügbare Konfigurationsparameter:

Parameter	Beschreibung
Enable SAML authentication	Aktivieren Sie das Kontrollkästchen, um die SAML-Authentifizierung zu aktivieren.
Enable JIT provisioning	Aktivieren Sie das Kontrollkästchen, um die JIT-Benutzerbereitstellung zu aktivieren.
IDP entity ID	Die eindeutige Entitätskennung innerhalb des SAML-Identitätsanbieters.
SSO service URL	Die URL, zu der Benutzer bei der Anmeldung weitergeleitet werden.
SLO service URL	Die URL, zu der Benutzer bei der Abmeldung weitergeleitet werden. Wenn das Feld leer bleibt, wird der SLO-Dienst nicht verwendet.

Parameter	Beschreibung
Username attribute	SAML-Attribut, das bei der Anmeldung bei Zabbix als Benutzername verwendet werden soll. Die Liste der unterstützten Werte wird vom Identitätsanbieter bestimmt. Beispiele: 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	Die eindeutige Kennung des Dienstanbieters (bei fehlender Übereinstimmung wird der Vorgang abgelehnt). Es kann eine URL oder eine beliebige Zeichenfolge angegeben werden.
SP name ID format	Fordert ein bestimmtes Name-ID-Format in der Antwort an. Beispiele: 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	Inhalt des Dienstanbieterzertifikats (SP) für die Einrichtung des SAML-Single-Sign-on-Dienstes (SSO).
SP private key	Inhalt des privaten Schlüssels des Dienstanbieters (SP) für die Einrichtung des SAML-Single-Sign-on-Dienstes (SSO). Ermöglicht eine sichere Authentifizierung und einen sicheren Datenaustausch mit dem Identitätsanbieter (IdP).
SP certificate	Inhalt des Dienstanbieterzertifikats (SP) für die Einrichtung des SAML-Single-Sign-on-Dienstes (SSO).
Sign	Deaktivieren Sie die Kontrollkästchen, um SAML-Signaturen für die entsprechenden Entitäten zu deaktivieren: *Messages* *Assertions* *AuthN requests* *Logout requests* *Logout responses*
Encrypt	Aktivieren Sie die Kontrollkästchen, um die Entitäten auszuwählen, für die die SAML-Verschlüsselung aktiviert werden soll: *Name ID* *Assertions*
Case-sensitive login	Deaktivieren Sie das Kontrollkästchen, um die Groß-/Kleinschreibung bei der Anmeldung für Benutzernamen zu deaktivieren (standardmäßig aktiviert). Durch die Deaktivierung der Groß-/Kleinschreibung bei der Anmeldung ist beispielsweise eine Anmeldung als „admin“ möglich, auch wenn der Zabbix-Benutzer „Admin“ oder „ADMIN“ lautet. Beachten Sie, dass die Anmeldung für diese Benutzer immer mit der folgenden Fehlermeldung abgelehnt wird, wenn die Groß-/Kleinschreibung bei der Anmeldung deaktiviert ist und mehrere Zabbix-Benutzer mit ähnlichen Benutzernamen vorhanden sind (z. B. Admin und admin): „Authentication failed: supplied credentials are not unique.“
Configure JIT provisioning	Aktivieren Sie dieses Kontrollkästchen, um Optionen für die JIT-Benutzerbereitstellung anzuzeigen.
Group name attribute	Geben Sie das Attribut für den Gruppennamen der JIT-Benutzerbereitstellung an.
User name attribute	Geben Sie das Attribut für den Benutzernamen der JIT-Benutzerbereitstellung an.
User last name attribute	Geben Sie das Attribut für den Nachnamen des Benutzers der JIT-Benutzerbereitstellung an.

Parameter	Beschreibung
User group mapping	Ordnen Sie ein SAML-Benutzergruppenmuster einer Zabbix-Benutzergruppe und einer Benutzerrolle zu. Dies ist erforderlich, um zu bestimmen, welche Benutzergruppe bzw. Rolle der bereitgestellte Benutzer in Zabbix erhält. Klicken Sie auf *Add*, um eine Zuordnung hinzuzufügen. Das Feld *SAML group pattern* unterstützt Platzhalter. Der Gruppenname muss mit einer vorhandenen Gruppe übereinstimmen. Wenn ein SAML-Benutzer mehreren Zabbix-Benutzergruppen entspricht, wird der Benutzer Mitglied aller dieser Gruppen. Wenn ein Benutzer mehreren Zabbix-Benutzerrollen entspricht, erhält er die höchste Berechtigungsstufe unter diesen Rollen.
Media type mapping	Ordnen Sie die SAML-Medienattribute des Benutzers (z. B. E-Mail) den Zabbix-Benutzermedien zum Senden von Benachrichtigungen zu.
Enable SCIM provisioning	Aktivieren Sie dieses Kontrollkästchen, um die SCIM-2.0-Bereitstellung zu aktivieren.

Beispiele für die Konfiguration von SAML-Identitätsanbietern für die Anmeldung und Benutzerbereitstellung in Zabbix finden Sie unter:

- Microsoft Azure AD
- Okta
- Onelogin

Hinweise zur SCIM-Bereitstellung

Geben Sie für die SCIM-Bereitstellung auf der Seite des Identitätsanbieters den Pfad zum Zabbix-Frontend an und hängen Sie `api_scim.php` daran an, also:

`https://<path-to-zabbix-ui>/api_scim.php`

Benutzerattribute, die in Zabbix verwendet werden (Benutzername, Vorname, Nachname und Medienattribute), müssen als benutzerdefinierte Attribute hinzugefügt werden. Falls erforderlich, sollte der externe Namespace mit dem Benutzerschema übereinstimmen: `urn:ietf:params:scim:schemas:core:2.0:User`.

Erweiterte Einstellungen

Zusätzliche SAML-Parameter können in der Frontend-Konfigurationsdatei von Zabbix (`zabbix.conf.php`) konfiguriert werden:

- `$SSO['SP_KEY']`
- `$SSO['SP_CERT']`
- `$SSO['IDP_CERT']`
- `$SSO['SETTINGS']`

Konfigurationsbeispiele

Beispiel 1: Verbindung zu einer Zabbix-Instanz mit einem Load Balancer über TLS/SSL

Wenn ein Load Balancer für die Verbindung zu einer Zabbix-Instanz verwendet wird und der Load Balancer TLS/SSL verwendet, Zabbix jedoch nicht, geben Sie die Parameter `baseurl`, `strict` und `use_proxy_headers` wie folgt an:

```
$SSO['SETTINGS'] = [
    'strict' => false,
    'baseurl' => 'https://zabbix.example.com/zabbix/',
    'use_proxy_headers' => true
];
```

Beispiel 2: Konfigurieren benutzerdefinierter SAML-Signatur- und Digest-Algorithmen

```
$SSO['SETTINGS'] = [
    'security' => [
        'signatureAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#rsa-sha384'
        'digestAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#sha384',
        // ...
    ],
    // ...
];
```

Beispiel 3: Konfigurieren von SSO mit Kerberos-Authentifizierung über ADFS

```
$SSO['SETTINGS'] = [
    'security' => [
        'requestedAuthnContext' => [
            'urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos',
        ],
        'requestedAuthnContextComparison' => 'exact'
    ]
];
```

In diesem Fall legen Sie im Feld *SP name ID* der SAML-Konfiguration Folgendes fest:

```
urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified
```

MFA

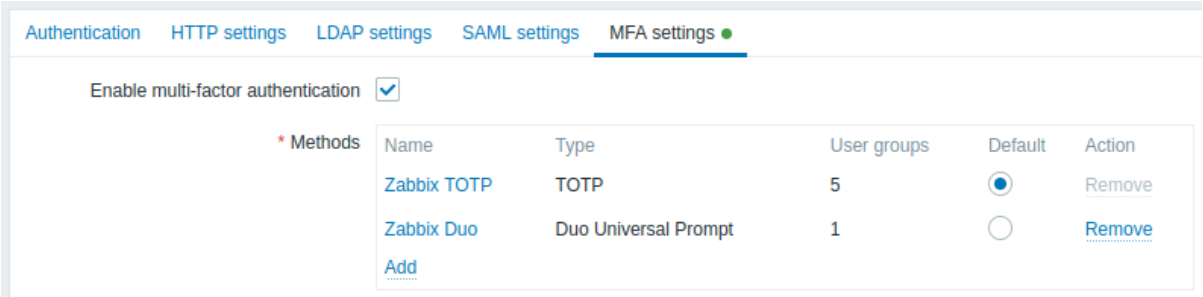
Übersicht

Die Multi-Faktor-Authentifizierung (MFA) kann verwendet werden, um sich bei Zabbix anzumelden, und bietet eine zusätzliche Sicherheitsebene über Benutzername und Passwort hinaus.

Mit MFA muss der Benutzer in Zabbix vorhanden sein, beim Anmelden Zabbix-Zugangsdaten angeben und seine Identität zusätzlich auf andere Weise nachweisen, in der Regel mit einem Code, der von einer Authenticator-App auf dem Telefon des Benutzers generiert wird.

Es stehen mehrere MFA-Methoden zur Verfügung, sodass Benutzer die Option wählen können, die ihren Sicherheitsanforderungen und Präferenzen am besten entspricht. Diese Methoden sind zeitbasierte Einmalpasswörter (TOTP) und Duo Universal Prompt.

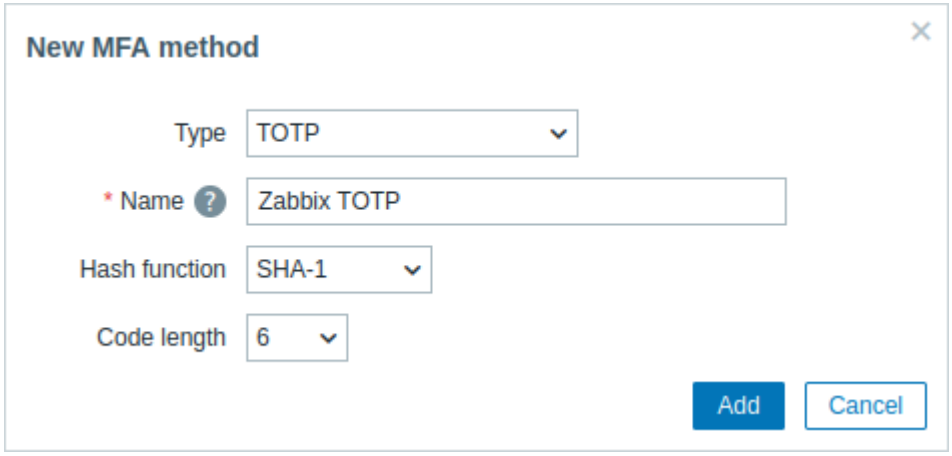
Konfiguration



Konfigurationsparameter:

Parameter	Beschreibung
Multi-Faktor-Authentifizierung aktivieren	Aktivieren Sie das Kontrollkästchen, um die Multi-Faktor-Authentifizierung zu aktivieren.
Methoden	Klicken Sie auf *Hinzufügen*, um eine MFA-Methode zu konfigurieren (siehe Methodenkonfiguration unten).

Methodenkonfiguration



Parameter der Methodenkonfiguration:

Parameter	Beschreibung
Type	Wählen Sie den Typ der MFA-Methode aus: **TOTP** - verwenden Sie eine Authenticator-App, um zeitbasierte Einmalpasswörter zu generieren; **Duo Universal Prompt** - verwenden Sie den Authentifizierungsdienst Duo, um Multi-Faktor-Authentifizierung bereitzustellen.
Name	Geben Sie einen Namen ein, der in Authenticator-Apps allen MFA-Benutzern als Kontoname angezeigt wird (zum Beispiel „Zabbix“).
Hash function	Wählen Sie die Hash-Funktion (SHA-1, SHA-256 oder SHA-512) zur Generierung von TOTP-Codes aus. Dieser Parameter ist verfügbar, wenn der Typ der MFA-Methode auf „TOTP“ gesetzt ist. Beachten Sie, dass die Wahl von SHA-256 oder SHA-512 die Kompatibilität erheblich einschränken kann, da viele Anwendungen diese Funktionen derzeit nicht unterstützen.
Code length	Wählen Sie die Länge des Bestätigungscodes (6 oder 8) aus. Dieser Parameter ist verfügbar, wenn der Typ der MFA-Methode auf „TOTP“ gesetzt ist.
API hostname	Geben Sie den vom Duo-Authentifizierungsdienst bereitgestellten API-Hostnamen ein. Dieser Parameter ist verfügbar, wenn der Typ der MFA-Methode auf „Duo Universal Prompt“ gesetzt ist.
Client ID	Geben Sie die vom Duo-Authentifizierungsdienst bereitgestellte Client-ID ein. Dieser Parameter ist verfügbar, wenn der Typ der MFA-Methode auf „Duo Universal Prompt“ gesetzt ist.
Client secret	Geben Sie das vom Duo-Authentifizierungsdienst bereitgestellte Client-Secret ein. Dieser Parameter ist verfügbar, wenn der Typ der MFA-Methode auf „Duo Universal Prompt“ gesetzt ist.

Konfigurationsbeispiele

Dieser Abschnitt enthält Beispiele für die Konfiguration von MFA mit Time-Based One-Time Password (TOTP) und Duo Universal Prompt.

TOTP

Für TOTP müssen Benutzer ihre Identität mithilfe einer Authenticator-App verifizieren (zum Beispiel der App Google Authenticator).

1. Gehen Sie in Zabbix unter *Users* > *Authentication* zu den MFA-Einstellungen und aktivieren Sie die Multi-Faktor-Authentifizierung.

2. Fügen Sie eine neue MFA-Methode mit der folgenden Konfiguration hinzu:

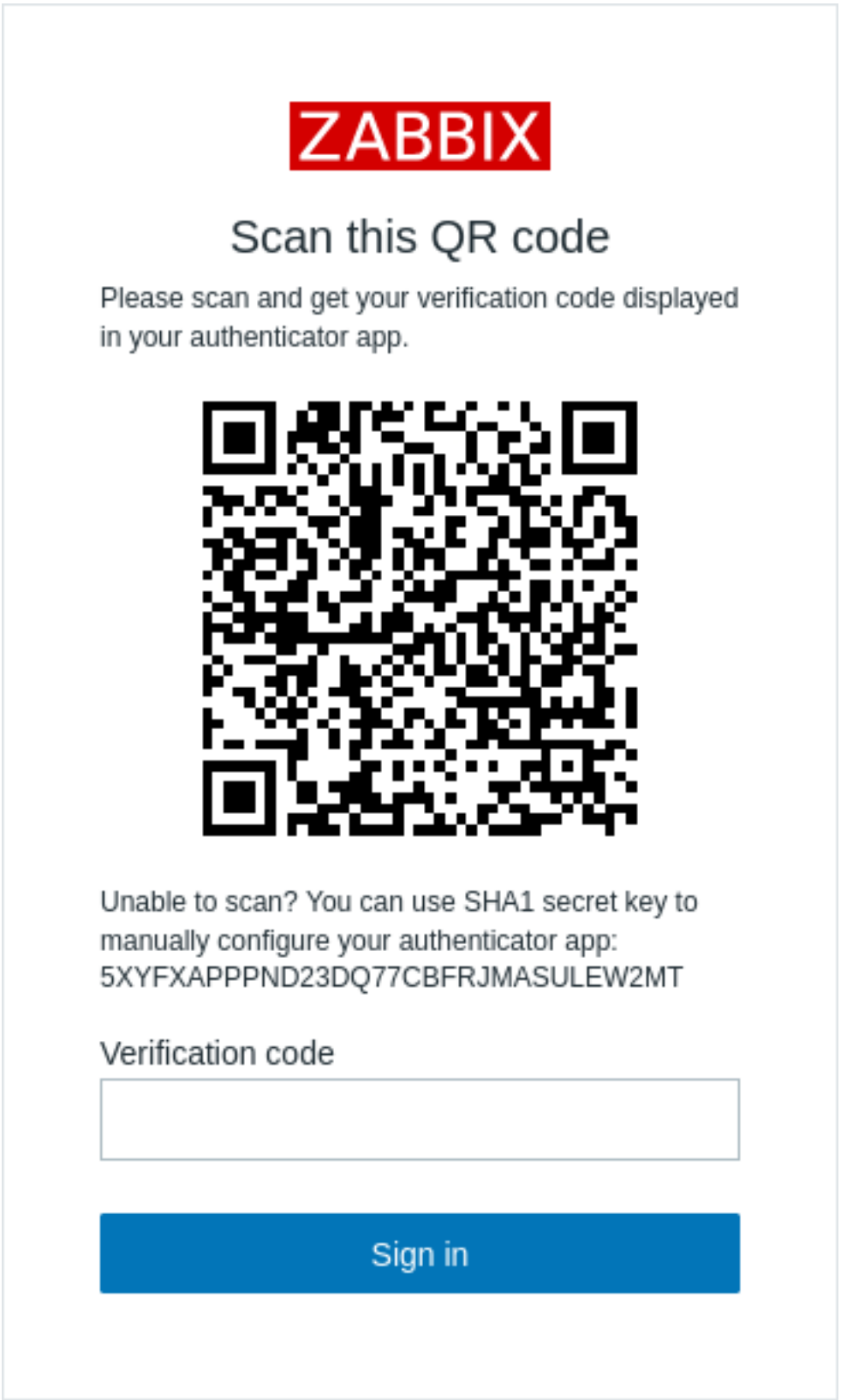
   - Typ: TOTP
   - Name: Zabbix TOTP
   - Hash-Funktion: SHA-1
   - Code-Länge: 6

3. Klicken Sie auf die Schaltfläche *Add* und anschließend auf *Update*.

4. Gehen Sie zu *Users* > *User groups* und erstellen Sie eine neue Benutzergruppe mit der folgenden Konfiguration:

   - Gruppenname: TOTP group
   - Benutzer: Admin
   - Multi-Faktor-Authentifizierung: Default (oder ”Zabbix TOTP”, wenn es nicht als Standard festgelegt ist)

5. Melden Sie sich von Zabbix ab und anschließend mit Ihren Zugangsdaten wieder an. Nach erfolgreicher Anmeldung werden Sie aufgefordert, sich für MFA zu registrieren; dabei werden ein QR-Code und ein geheimer Schlüssel angezeigt.



6. Scannen Sie den QR-Code oder geben Sie den geheimen Schlüssel in die Google Authenticator-App ein. Die App generiert einen Verifizierungscode, den Sie eingeben müssen, um den Anmeldevorgang abzuschließen.

7. Für spätere Anmeldungen rufen Sie den Verifizierungscode aus der Google Authenticator-App ab und geben Sie ihn während der Anmeldung ein.

Duo Universal Prompt

Für Duo Universal Prompt müssen Benutzer ihre Identität mit der Authenticator-App Duo Mobile verifizieren.

Attention:
Die MFA-Methode Duo Universal Prompt erfordert die Installation der Erweiterung php-curl, Zugriff auf Zabbix über HTTPS und die Berechtigung für ausgehende Verbindungen zu Duo-Servern. Wenn Sie außerdem Content Security Policy (CSP) auf dem Webserver aktiviert haben, stellen Sie sicher, dass Sie "duo.com" zur CSP-Direktive in der Konfigurationsdatei Ihres virtuellen Hosts hinzufügen.

1. Registrieren Sie sich für ein kostenloses Duo-Administratorkonto unter Duo Signup.

2. Öffnen Sie das Duo Admin Panel, gehen Sie zu *Applications* > *Protect an Application*, suchen Sie nach der Anwendung *Web SDK* und klicken Sie auf *Protect*.

3. Notieren Sie sich die Anmeldedaten (Client ID, Client secret, API hostname), die für die Konfiguration der MFA-Methode in Zabbix erforderlich sind.

4. Gehen Sie in Zabbix unter *Users* > *Authentication* zu den MFA-Einstellungen und aktivieren Sie die Multi-Faktor-Authentifizierung.

5. Fügen Sie eine neue MFA-Methode mit der folgenden Konfiguration hinzu:

- Type: Duo Universal Prompt
- Name: Zabbix Duo
- API hostname: (use API hostname from Duo)
- Client ID: (use Client ID from Duo)
- Client secret: (use Client secret from Duo)

6. Klicken Sie auf die Schaltfläche *Add* und anschließend auf *Update*.

7. Gehen Sie zu *Users* > *User groups* und erstellen Sie eine neue Benutzergruppe mit der folgenden Konfiguration:

- Group name: Duo group
- Users: Admin
- Multi-factor authentication: Default (or "Zabbix Duo" if it is not set as default)

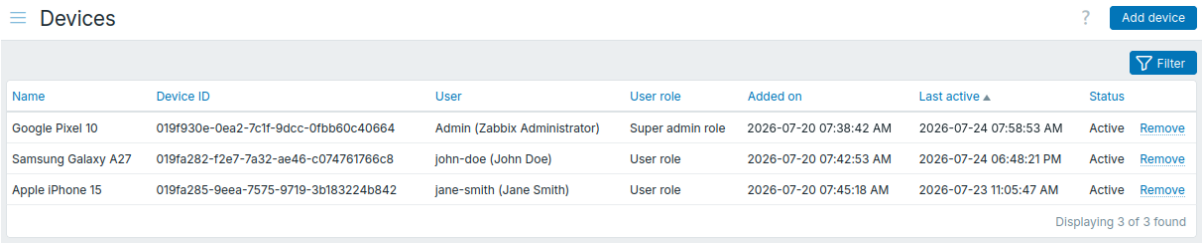
8. Melden Sie sich von Zabbix ab und anschließend mit Ihren Zugangsdaten wieder an. Nach erfolgreicher Anmeldung werden Sie aufgefordert, sich für MFA zu registrieren, und zu Duo weitergeleitet. Schließen Sie die Duo-Einrichtung ab und verifizieren Sie Ihren Benutzer mit der Duo-App auf Ihrem Telefon, um sich anzumelden.

9. Verwenden Sie bei späteren Anmeldungen die passende von der Duo-App bereitgestellte MFA-Methode (z. B. Abrufen eines Verifizierungscodes, Bestätigen von Push-Benachrichtigungen oder Verwendung von Hard Keys) und geben Sie die erforderlichen Informationen während der Anmeldung ein.

Geräte

In diesem Bereich können Super-Admins mobile Geräte verwalten, die mit der Zabbix Mobile App verwendet werden sollen.

Dieser Bereich wird angezeigt, wenn `EnableMobileDevices` aktiviert ist.



So fügen Sie ein Gerät hinzu:

1. Klicken Sie auf *Gerät hinzufügen*.
2. Wählen Sie einen Benutzer aus und klicken Sie anschließend auf *Hinzufügen*. Für den Benutzer muss der Zugriff auf Geräte aktiviert sein.
3. Ein QR-Code wird angezeigt. Der Benutzer muss diesen Code in seiner Zabbix Mobile App scannen.

Note:
Wenn Sie ein Gerät für einen Benutzer hinzufügen, fügt Zabbix außerdem den Medientyp Push-Benachrichtigung zu den Benutzermedien des Benutzers hinzu. Wenn Sie das Gerät entfernen, wird dieses Benutzermedium ebenfalls entfernt. Wenn mehrere Medientypen für Push-Benachrichtigungen konfiguriert sind, fügt Zabbix den aktivierten Medientyp hinzu, der alphabetisch an erster Stelle steht.

Hinzugefügte Geräte werden in einer Liste angezeigt, die Sie filtern können.

Sie können Geräte auch entfernen. Wenn Sie ein Gerät entfernen, werden seine Zugangsdaten widerrufen und aus Zabbix entfernt. Danach kann das Gerät nicht mehr auf Zabbix zugreifen.

Die Liste zeigt folgende Details:

Spalte	Beschreibung
Name	Gerätename.
Geräte-ID	Geräte-ID.
Benutzer	Benutzer, für den das Gerät hinzugefügt wurde, oder *Nicht zugänglicher Benutzer*, wenn dieser Benutzer gelöscht wurde.
Benutzerrolle	Benutzerrolle.
Hinzugefügt am	Datum und Uhrzeit, zu denen das Gerät hinzugefügt wurde.
Zuletzt aktiv	Datum und Uhrzeit, zu denen das Gerät zuletzt aktiv war.
Status	Gerätestatus: **Neu** – Das Gerät wurde hinzugefügt, ohne den QR-Code zu scannen, und muss erneut hinzugefügt werden. **Neu (Token abgelaufen)** – Das Gerät wurde hinzugefügt, ohne den QR-Code zu scannen, und das Token im QR-Code ist abgelaufen; das Gerät muss erneut hinzugefügt werden. **Aktiv** – Das Gerät wurde hinzugefügt, der QR-Code wurde gescannt und es kann die Zabbix Mobile App verwenden. **Nicht zugewiesen** – Das Gerät wurde hinzugefügt, aber sein Benutzer wurde gelöscht; es kann die Zabbix Mobile App nicht verwenden.

Benutzer mit ausreichenden Berechtigungen können ihre eigenen Geräte auch unter *Benutzereinstellungen* > *Geräte* verwalten.

Administration

Übersicht

Das Menü **Administration** ist für administrative Funktionen von Zabbix vorgesehen. Dieses Menü ist nur für Benutzer des Benutzertyps SuperAdmin verfügbar.

Datenquelle

Übersicht

Im Abschnitt *Administration* > *Datenquelle* können Sie die Telemetriedatenquelle angeben, also die Datenbank, in der Telemetriedaten (Traces, Metriken und Protokolle) gespeichert werden. Telemetriedaten werden für das Application Performance Monitoring (APM) verwendet. Damit können Sie nachvollziehen, wie Anforderungen durch Ihre Anwendungen und Services fließen, ihre Leistung messen und die Ursache für langsame oder fehlgeschlagene Vorgänge ermitteln.

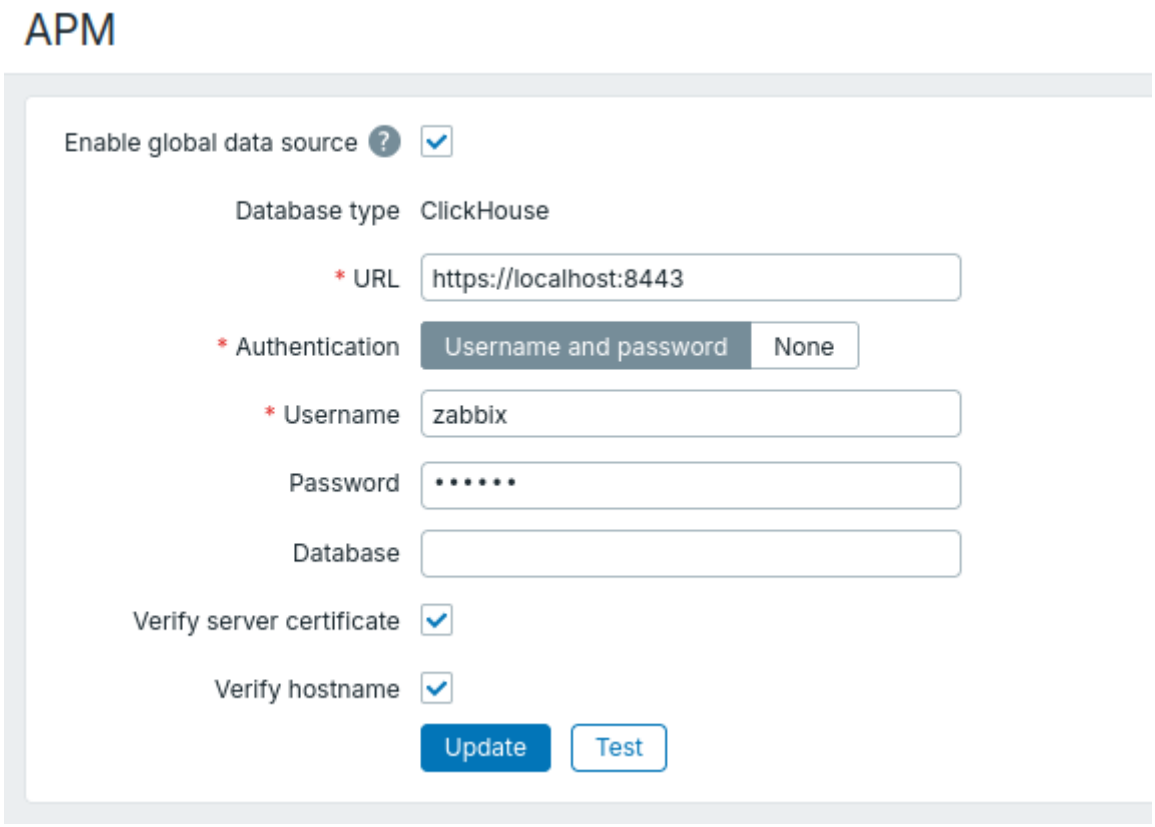
Die Telemetriedatenquelle muss OpenTelemetry unterstützen.

Die in diesem Abschnitt angegebene Telemetriedatenquelle wird als globale Datenquelle bezeichnet, da sie von Zabbix Server, Proxy und Frontend gemeinsam verwendet wird. Sie können sie jedoch für jede dieser Komponenten überschreiben, indem Sie eine andere Telemetriedatenquelle in der Konfiguration von Server, Proxy oder Frontend angeben.

Der Datenpunkt Telemetrieabfrage fragt diese Datenquelle ab, sofern in der Konfiguration der Komponente keine andere Datenquelle angegeben ist.

APM

Um die globale Datenquelle anzugeben, aktivieren Sie das Kontrollkästchen *Globale Datenquelle aktivieren* und geben Sie die unten beschriebenen Parameter ein. Vor dem Speichern können Sie auf *Testen* klicken, um die Verbindung zu dieser Datenquelle zu überprüfen (erfordert die Erweiterung php-curl).



Parameter	Beschreibung
Datenbanktyp	Der Datenbanktyp ist fest auf *ClickHouse* eingestellt und kann nicht geändert werden. Siehe auch: Anforderungen und ClickHouse-Einrichtung.
URL	Geben Sie die HTTP- oder HTTPS-URL ein, die für die Verbindung zur Datenbank verwendet wird.
Authentifizierung	Wählen Sie die Authentifizierungsmethode für die Anmeldung an der Datenbank aus: *Benutzername und Passwort* oder *Keine*.
Benutzername	Geben Sie den Benutzernamen für die Anmeldung an der Datenbank an. Dieser Parameter ist verfügbar, wenn *Authentifizierung* auf *Benutzername und Passwort* gesetzt ist.
Passwort	Geben Sie das Passwort für die Anmeldung an der Datenbank an oder lassen Sie das Feld leer, wenn der Datenbankbenutzer kein Passwort hat. Das Passwort wird zurückgesetzt, wenn Sie die URL ändern. Dieser Parameter ist verfügbar, wenn *Authentifizierung* auf *Benutzername und Passwort* gesetzt ist.
Datenbank	Geben Sie den Namen der Datenbank ein, in der die Telemetriedaten gespeichert werden. Wenn das Feld leer bleibt, verwendet Zabbix die in ClickHouse für den Benutzer konfigurierte Standarddatenbank.
Serverzertifikat überprüfen	Aktivieren Sie das Kontrollkästchen, um das Serverzertifikat zu überprüfen. Wenn diese Option aktiviert ist, müssen die zugehörigen TLS-Optionen in der Konfiguration von Server, Proxy oder Frontend festgelegt werden. Dieser Parameter ist verfügbar, wenn *URL* eine HTTPS-URL enthält.
Hostnamen überprüfen	Aktivieren Sie das Kontrollkästchen, um zu überprüfen, ob der Hostname in der *URL* mit dem Hostnamen im Serverzertifikat übereinstimmt. Wenn diese Option aktiviert ist, müssen die zugehörigen TLS-Optionen in der Konfiguration von Server, Proxy oder Frontend festgelegt werden. Dieser Parameter ist verfügbar, wenn *URL* eine HTTPS-URL enthält.

Allgemein

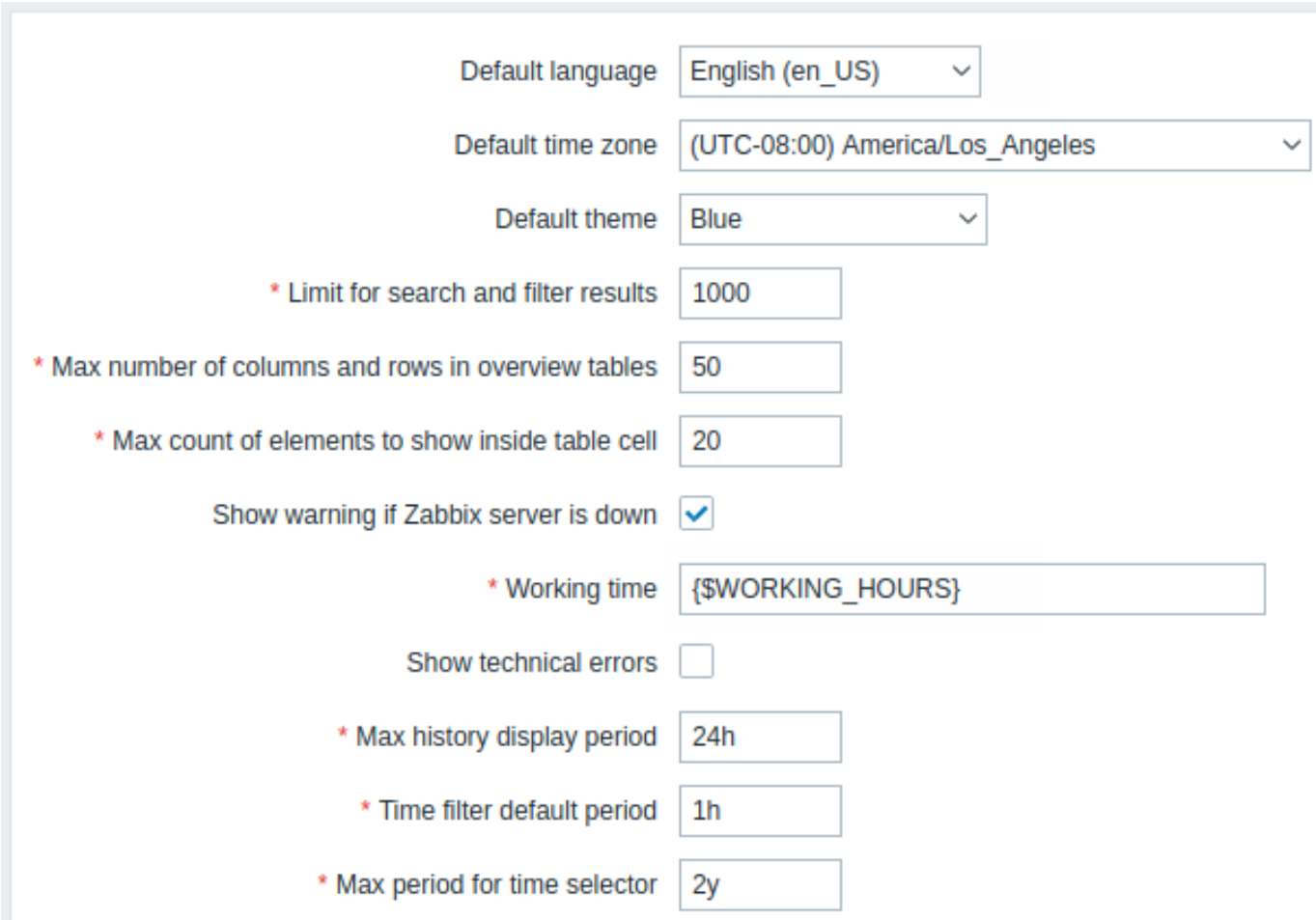
Übersicht

Der Abschnitt *Administration* > *General* enthält eine Reihe von Unterabschnitten zum Festlegen von Standardwerten für das Frontend und zum Anpassen von Zabbix.

Die Liste der verfügbaren Unterabschnitte wird angezeigt, wenn Sie im Menüabschnitt *Administration* auf *General* klicken. Es ist auch möglich, zwischen den Unterabschnitten über das Titeldropdown in der oberen linken Ecke zu wechseln.

GUI

Dieser Abschnitt bietet die Anpassung mehrerer Frontend-bezogener Standardwerte.



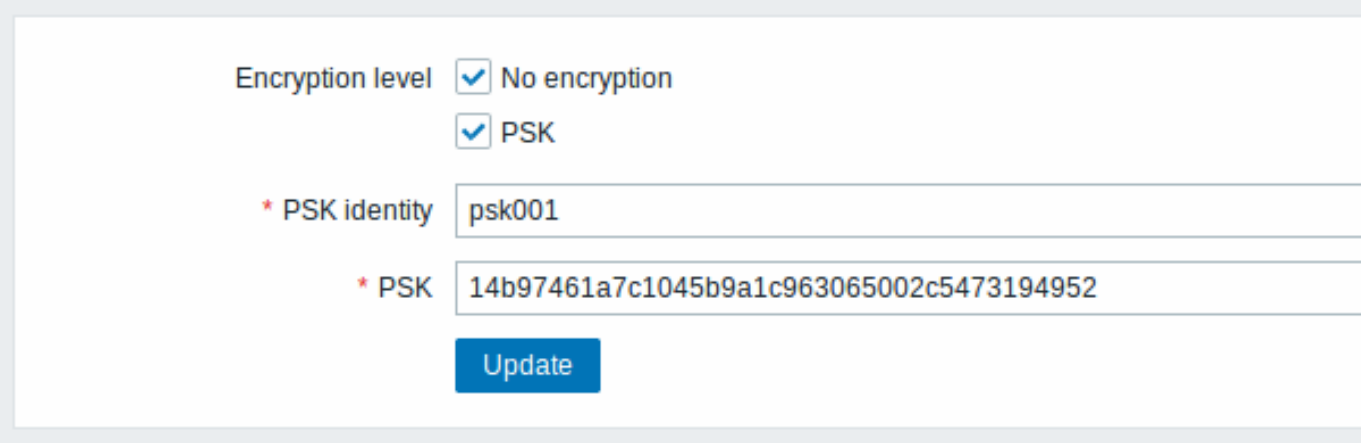
Konfigurationsparameter:

Parameter	Beschreibung
Standardsprache	Standardsprache für Benutzer, die in ihren Profilen keine Sprache angegeben haben, sowie für Gastbenutzer. Weitere Informationen finden Sie unter Installation zusätzlicher Frontend-Sprachen.
Standardzeitzone	Standard-Zeitzone für Benutzer, die in ihren Profilen keine Zeitzone angegeben haben, sowie für Gastbenutzer.
Standarddesign	Standarddesign für Benutzer, die in ihren Profilen kein Design angegeben haben, sowie für Gastbenutzer.
Begrenzung für Such- und Filterergebnisse	Maximale Anzahl von Elementen (Zeilen), die in einer Liste der Weboberfläche angezeigt werden, zum Beispiel unter *Datenerfassung* > *Hosts*. *Hinweis*: Wenn hier beispielsweise '50' festgelegt ist, werden in allen betroffenen Frontend-Listen nur die ersten 50 Elemente angezeigt. Enthält eine Liste mehr als fünfzig Elemente, wird dies durch das '+'-Zeichen in *"Anzeige von 1 bis 50 von **50+** gefunden"* angezeigt. Wenn außerdem gefiltert wird und dennoch mehr als 50 Treffer vorhanden sind, werden nur die ersten 50 angezeigt. Beachten Sie, dass eine Erhöhung dieses Werts zu geringerer Leistung und höherem Speicherverbrauch auf der Webserver-Seite führen kann.
Maximale Anzahl von Spalten und Zeilen in Übersichtstabellen	Maximale Anzahl von Spalten und Zeilen, die im Dashboard-Widget *Auslöserübersicht* angezeigt werden. Dieselbe Begrenzung gilt sowohl für Spalten als auch für Zeilen. Wenn mehr Zeilen und/oder Spalten vorhanden sind als angezeigt werden, zeigt das System unten in der Tabelle eine Warnung an: "Nicht alle Ergebnisse werden angezeigt. Bitte geben Sie spezifischere Suchkriterien an."
Maximale Anzahl von Elementen zur Anzeige innerhalb einer Tabellenzelle	Für Einträge, die in einer einzelnen Tabellenzelle angezeigt werden, werden nicht mehr als die hier konfigurierte Anzahl angezeigt.

Parameter	Beschreibung
Warnung anzeigen, wenn der Zabbix-Server ausgefallen ist	Dieser Parameter aktiviert die Anzeige einer Warnmeldung in einem Browserfenster, wenn der Zabbix-Server nicht erreicht werden kann (möglicherweise ausgefallen). Die Meldung bleibt sichtbar, auch wenn der Benutzer auf der Seite nach unten scrollt. Beim Darüberfahren mit der Maus wird die Meldung vorübergehend ausgeblendet, um den darunterliegenden Inhalt sichtbar zu machen.
Arbeitszeit	Dieser systemweite Parameter definiert die Arbeitszeiten. In Diagrammen wird die Arbeitszeit als weißer Hintergrund und die Nichtarbeitszeit als grau dargestellt. Eine Beschreibung des Zeitformats finden Sie auf der Seite Zeitperiodenspezifikation. Benutzermakros werden unterstützt.
Technische Fehler anzeigen	Wenn diese Option aktiviert ist, können alle registrierten Benutzer technische Fehler (PHP/SQL) sehen. Wenn sie deaktiviert ist, sind die Informationen nur für Zabbix Super Admins und Benutzer in Benutzergruppen mit aktiviertem Debug-Modus verfügbar.
Maximale Anzeigedauer des Verlaufs	Maximaler Zeitraum, für den historische Daten in *Überwachung* > *Neueste Daten*, Informationen zu Webszenarien von Hosts in *Überwachung* > *Hosts* sowie im Dashboard-Widget *Top-Datenpunkte* angezeigt werden. Zulässiger Bereich: 24 Stunden (Standard) - 1 Woche. Zeit-Suffixe, z. B. 1w (eine Woche), 36h (36 Stunden), werden unterstützt.
Standardzeitraum des Zeitfilters	Standardmäßig in Diagrammen und Dashboards zu verwendender Zeitraum. Zulässiger Bereich: 1 Minute - 10 Jahre (Standard: 1 Stunde). Zeit-Suffixe, z. B. 10m (zehn Minuten), 5w (fünf Wochen), werden unterstützt. Hinweis: Wenn ein Benutzer den Zeitraum beim Anzeigen eines Diagramms ändert, wird dieser Zeitraum als Benutzereinstellung gespeichert und ersetzt den globalen Standardwert oder eine frühere Benutzerauswahl.
Maximaler Zeitraum für den Zeitauswähler	Maximal verfügbarer Zeitraum für Diagramme und Dashboards. Benutzer können keine älteren Daten visualisieren. Zulässiger Bereich: 1 Jahr - 10 Jahre (Standard: 2 Jahre). Zeit-Suffixe, z. B. 1y (ein Jahr), 365w (365 Wochen), werden unterstützt.

Autoregistrierung

In diesem Abschnitt können Sie die Verschlüsselungsstufe für die automatische Registrierung aktiver Agenten konfigurieren.



Mit einem Sternchen markierte Parameter sind Pflichtfelder.

Konfigurationsparameter:

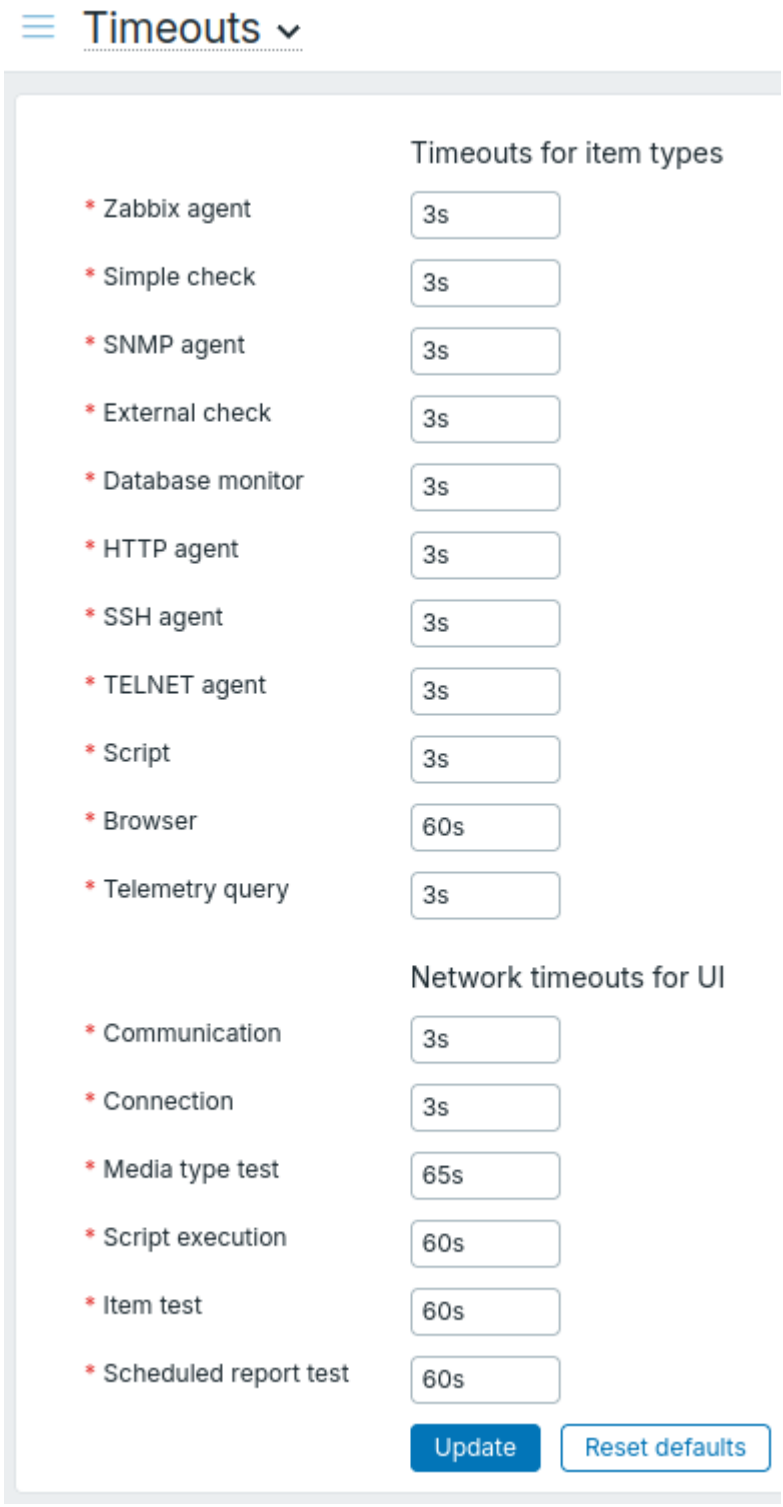
Parameter	Description
Verschlüsselungsstufe	Wählen Sie eine oder beide Optionen für die Verschlüsselungsstufe aus: **Keine Verschlüsselung** - unverschlüsselte Verbindungen sind zulässig **PSK** - TLS-verschlüsselte Verbindungen mit einem vorab gemeinsam genutzten Schlüssel sind zulässig
PSK identity	Geben Sie die Identitätszeichenfolge des vorab gemeinsam genutzten Schlüssels ein. Dieses Feld ist nur verfügbar, wenn 'PSK' als *Verschlüsselungsstufe* ausgewählt ist. Geben Sie keine sensiblen Informationen in die PSK-Identität ein, da sie unverschlüsselt über das Netzwerk übertragen wird, um einem Empfänger mitzuteilen, welchen PSK er verwenden soll.

Parameter	Description
PSK	Geben Sie den vorab gemeinsam genutzten Schlüssel ein (eine gerade Anzahl hexadezimaler Zeichen). Maximale Länge: 512 Hex-Zeichen (256-Byte-PSK), wenn Zabbix die GnuTLS- oder OpenSSL-Bibliothek verwendet, 64 Hex-Zeichen (32-Byte-PSK), wenn Zabbix die mbed TLS (PolarSSL)-Bibliothek verwendet. Beispiel: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952 Dieses Feld ist nur verfügbar, wenn 'PSK' als *Verschlüsselungsstufe* ausgewählt ist.

Siehe auch: Sichere Autoregistrierung

Timeouts

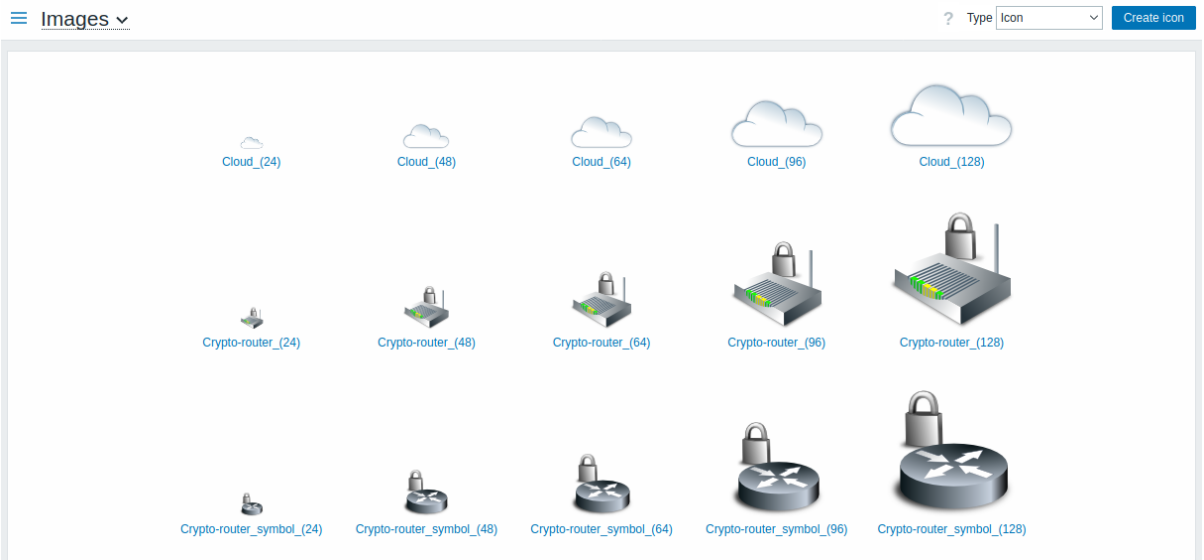
In diesem Abschnitt können globale Timeouts für Datentypen und Netzwerk-Timeouts festgelegt werden. Alle Felder in diesem Formular sind Pflichtfelder.



Parameter		Beschreibung
Timeouts for item types		Wie viele Sekunden Zabbix warten soll, bevor die Prüfung abgebrochen wird (abhängig vom Typ). Weitere Details und Einschränkungen finden Sie unter item timeout. Zulässiger Bereich: 1 - 600 s (Standard: 3 s; Standard für browser-Datenpunkte: 60 s). Zeitsuffixe, z. B. 30s, 1m, sowie Benutzermakros werden unterstützt. Beachten Sie, dass auf Proxy-Ebene konfigurierte timeouts diese globalen Einstellungen außer Kraft setzen. Wenn außerdem Timeouts für bestimmte items konfiguriert sind, überschreiben diese sowohl die globalen als auch die Proxy-Einstellungen.
Network timeouts for UI		
	Communication	Wie viele Sekunden gewartet werden soll, bevor ein inaktiver Socket geschlossen wird (wenn zuvor eine Verbindung zum Zabbix Server hergestellt wurde, das Frontend jedoch innerhalb dieser Zeit den Lese-/Sendevorgang der Daten nicht abschließen kann, wird die Verbindung getrennt). Zulässiger Bereich: 1 - 300 s (Standard: 3 s).
	Connection	Wie viele Sekunden gewartet werden soll, bevor ein Verbindungsversuch zum Zabbix Server abgebrochen wird. Zulässiger Bereich: 1 - 30 s (Standard: 3 s).
	Media type test	Wie viele Sekunden auf eine Antwort beim Testen eines Medientyps gewartet werden soll. Zulässiger Bereich: 1 - 300 s (Standard: 65 s).
	Script execution	Wie viele Sekunden auf eine Antwort beim Ausführen eines Skripts gewartet werden soll. Zulässiger Bereich: 1 - 300 s (Standard: 60 s). Dieses Timeout gilt für die gesamte Skriptausführungskette, die unterschiedlich lang sein kann. Wenn das Skript beispielsweise auf dem Agent ausgeführt wird, handelt es sich um einen Roundtrip über Server (möglicherweise auch Proxy) zum Agent und zurück. Für Skripte, die auf dem Agent ausgeführt werden, gilt das Timeout des server oder proxy. Für Skripte, die nur auf einem aktiven Agent ausgeführt werden, muss das Standard-Timeout des Servers/Proxys wahrscheinlich erhöht werden. Das Timeout des Servers/Proxys muss höher sein als die Aktualisierungsfrequenz der aktiven Prüfung, andernfalls kann das Timeout des Servers/Proxys überschritten werden, bevor der aktive Agent das Skript überhaupt empfängt. Siehe auch: Script timeout.
	Item test	Wie viele Sekunden auf zurückgegebene Daten beim Testen eines Datenpunkts gewartet werden soll. Zulässiger Bereich: 1 - 600 s (Standard: 60 s).
	Scheduled report test	Wie viele Sekunden auf zurückgegebene Daten beim Testen eines geplanten Berichts gewartet werden soll. Zulässiger Bereich: 1 - 300 s (Standard: 60 s).

Bilder

Der Abschnitt „Bilder“ zeigt alle in Zabbix verfügbaren Bilder an.
Bilder werden in der Datenbank gespeichert.

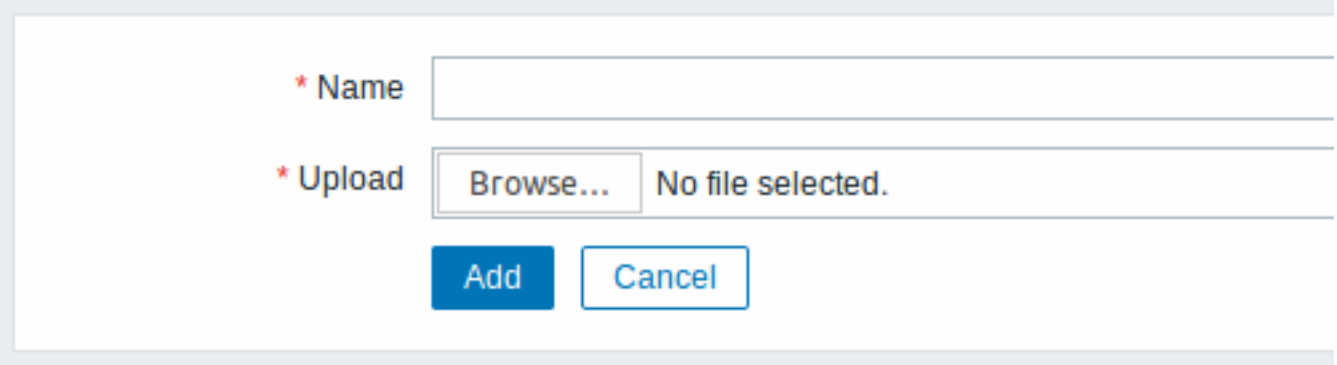


Über die Dropdown-Liste *Typ* können Sie zwischen Symbol- und Hintergrundbildern wechseln:

- Symbole werden zur Anzeige von Elementen in Netzwerkkarten verwendet.
- Hintergründe werden als Hintergrundbilder von Netzwerkkarten verwendet.

Bild hinzufügen

Sie können ein eigenes Bild hinzufügen, indem Sie oben rechts auf die Schaltfläche *Symbol erstellen* oder *Hintergrund erstellen* klicken.



Bildattribute:

Parameter	Beschreibung
Name	Eindeutiger Name eines Bildes.
Upload	Wählen Sie die Datei (PNG, JPEG, GIF oder WebP) aus einem lokalen System aus, die in Zabbix hochgeladen werden soll. *Hinweis*: Es ist möglicherweise möglich, andere Formate hochzuladen, die während des Uploads in PNG konvertiert werden. Für die Bildverarbeitung wird die GD-Bibliothek verwendet. Daher hängen die unterstützten Formate von der verwendeten Bibliotheksversion ab (Zabbix erfordert Version 2.0.28 oder höher).

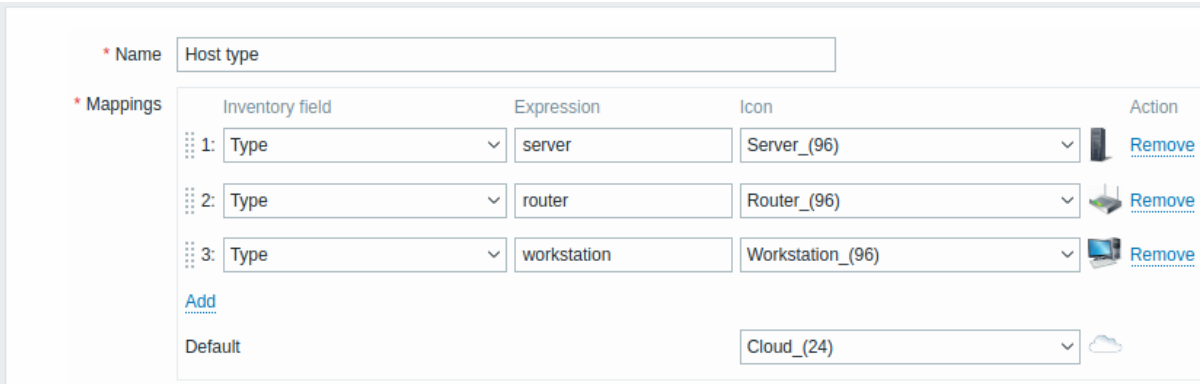
Note:
Die maximale Größe der Upload-Datei wird durch den Wert von `ZBX_MAX_IMAGE_SIZE` begrenzt, der 1024x1024 Bytes bzw. 1 MB beträgt.
Der Upload eines Bildes kann fehlschlagen, wenn die Bildgröße nahe bei 1 MB liegt und der MySQL-Konfigurationsparameter `max_allowed_packet` auf dem Standardwert von 1 MB steht.
Erhöhen Sie in diesem Fall den Parameter `max_allowed_packet`.

Icon-Zuordnung

In diesem Abschnitt können Zuordnungen bestimmter Hosts mit bestimmten Symbolen erstellt werden. Dabei werden Informationen aus den Inventarfeldern des Hosts verwendet, um die Zuordnung zu erstellen.

Die Zuordnungen können anschließend in der Netzwerkkarten-Konfiguration verwendet werden, um passenden Hosts automatisch geeignete Symbole zuzuweisen.

Um eine neue Icon-Zuordnung zu erstellen, klicken Sie oben rechts auf *Icon-Zuordnung erstellen*.



Konfigurationsparameter:

Parameter	Beschreibung
Name	Eindeutiger Name der Icon-Zuordnung.
Mappings	Eine Liste von Zuordnungen. Die Reihenfolge der Zuordnungen bestimmt, welche Priorität hat. Sie können Zuordnungen per Drag-and-drop in der Liste nach oben und unten verschieben.
Inventory field	Inventarfeld des Hosts, in dem nach einer Übereinstimmung gesucht wird.
Expression	Regulärer Ausdruck, der die Übereinstimmung beschreibt.
Icon	Symbol, das verwendet wird, wenn eine Übereinstimmung für den Ausdruck gefunden wird.
Default	Zu verwendendes Standardsymbol.

Reguläre Ausdrücke

In diesem Abschnitt können benutzerdefinierte reguläre Ausdrücke erstellt werden, die an mehreren Stellen im Frontend verwendet werden können. Weitere Informationen finden Sie im Abschnitt Reguläre Ausdrücke.

Optionen zur Anzeige von Auslösern

In diesem Abschnitt können Sie anpassen, wie der Status von Auslösern im Frontend angezeigt wird, sowie die Namen und Farben der Auslöser-Schweregrade.

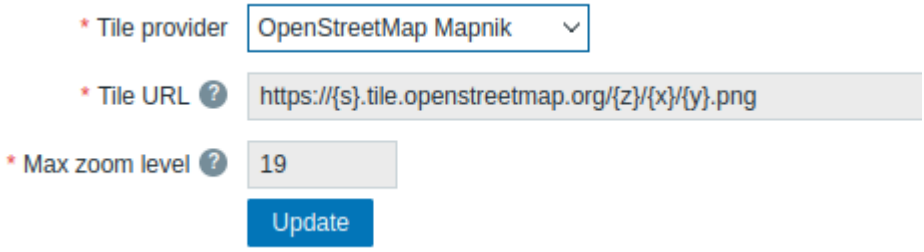


Parameter	Beschreibung
Benutzerdefinierte Farben für Ereignisstatus verwenden	Wenn Sie diesen Parameter aktivieren, werden die Farben für bestätigte/nicht bestätigte Probleme angepasst.
Nicht bestätigte PROBLEM-Ereignisse, *Bestätigte PROBLEM-Ereignisse*, *Nicht bestätigte RESOLVED-Ereignisse*, *Bestätigte RESOLVED-Ereignisse*	Geben Sie einen neuen Farbcode ein oder klicken Sie auf die Farbe, um eine neue aus der bereitgestellten Palette auszuwählen. Wenn das Kontrollkästchen *Blinken* aktiviert ist, blinken Auslöser nach einer Statusänderung für eine gewisse Zeit, um besser sichtbar zu sein.
OK-Auslöser anzeigen für	Zeitraum für die Anzeige von OK-Auslösern. Zulässiger Bereich: 0 - 24 Stunden. Zeit-Suffixe, z. B. 5m, 2h, 1d, werden unterstützt.
Bei Statusänderung blinken Auslöser für	Dauer des Blinkens von Auslösern. Zulässiger Bereich: 0 - 24 Stunden. Zeit-Suffixe, z. B. 5m, 2h, 1d, werden unterstützt.

Parameter	Beschreibung
Nicht klassifiziert, *Information*, *Warnung*, *Durchschnitt*, *Hoch*, *Katastrophe*	Benutzerdefinierte Schweregradnamen und/oder -farben, die anstelle der systemseitigen Standardwerte angezeigt werden. Geben Sie einen neuen Farbcode ein oder klicken Sie auf die Farbe, um eine neue aus der bereitgestellten Palette auszuwählen. Beachten Sie, dass die hier eingegebenen benutzerdefinierten Schweregradnamen in allen Sprachumgebungen verwendet werden. Wenn Sie sie für bestimmte Benutzer in andere Sprachen übersetzen müssen, siehe die Seite Benutzerdefinierte Auslöser-Schweregrade.

Geografische Karten

Dieser Abschnitt ermöglicht die Auswahl des Anbieters für geografische Kartenkacheln und die Konfiguration der Einstellungen des Anbieters für das Geomap-Dashboard-Widget. Zur Visualisierung mit geografischen Karten verwendet Zabbix die Open-Source-JavaScript-Bibliothek für interaktive Karten Leaflet. Bitte beachten Sie, dass Zabbix keinen Einfluss auf die Qualität der von Drittanbietern für Kacheln bereitgestellten Bilder hat, einschließlich der vordefinierten Kachelanbieter.



Parameter	Beschreibung
Kachelanbieter	Wählen Sie einen der verfügbaren Kachel-Dienstanbieter aus oder wählen Sie *Other*, um einen weiteren Kachelanbieter oder selbst gehostete Kacheln hinzuzufügen (siehe Verwendung eines benutzerdefinierten Kachel-Dienstanbieters).
Kachel-URL	Die URL-Vorlage (bis zu 2048 Zeichen) zum Laden und Anzeigen der Kachelebene auf geografischen Karten. Dieses Feld ist nur bearbeitbar, wenn *Tile provider* auf *Other* gesetzt ist. Die folgenden Platzhalter werden unterstützt: *{s}* steht für eine der verfügbaren Subdomains; *{z}* steht für den Zoomstufenparameter in der URL; *{x}* und *{y}* stehen für Kachelkoordinaten; *{r}* kann verwendet werden, um "@2x" zur URL hinzuzufügen und Retina-Kacheln zu laden. Beispiel: `https://{s}.example.com/{z}/{x}/{y}{r}.png`
Attributionstext	Attributionstext des Kachelanbieters, der in einem kleinen Textfeld auf der Karte angezeigt wird. Dieses Feld ist nur sichtbar, wenn *Tile provider* auf *Other* gesetzt ist.
Maximale Zoomstufe	Maximale Zoomstufe der Karte. Dieses Feld ist nur bearbeitbar, wenn *Tile provider* auf *Other* gesetzt ist.

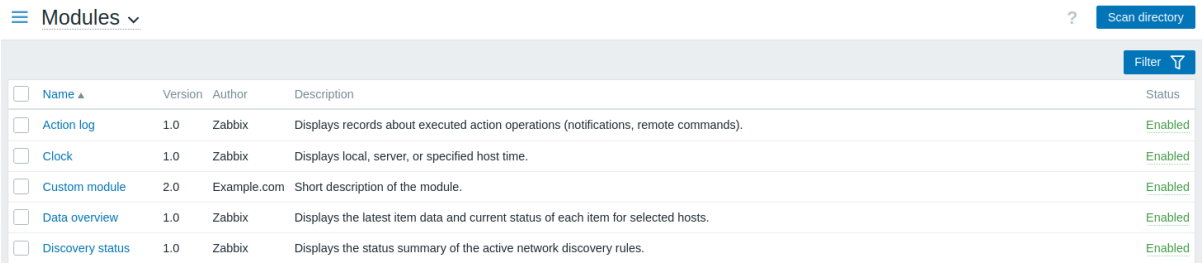
Verwendung eines benutzerdefinierten Tile-Service-Providers

Das Geomap-Widget kann Raster-Kachelbilder von einem benutzerdefinierten, selbst gehosteten oder von einem Drittanbieter bereitgestellten Tile-Provider-Dienst laden. Um einen benutzerdefinierten Drittanbieter-Tile-Provider-Dienst oder einen selbst gehosteten Tile-Ordner bzw. Server zu verwenden, wählen Sie *Other* im Feld *Tile provider* aus und geben Sie die benutzerdefinierte URL im Feld *Tile URL* unter Verwendung geeigneter Platzhalter an.

Module

In diesem Abschnitt können Sie benutzerdefinierte sowie integrierte Frontend-Module verwalten.

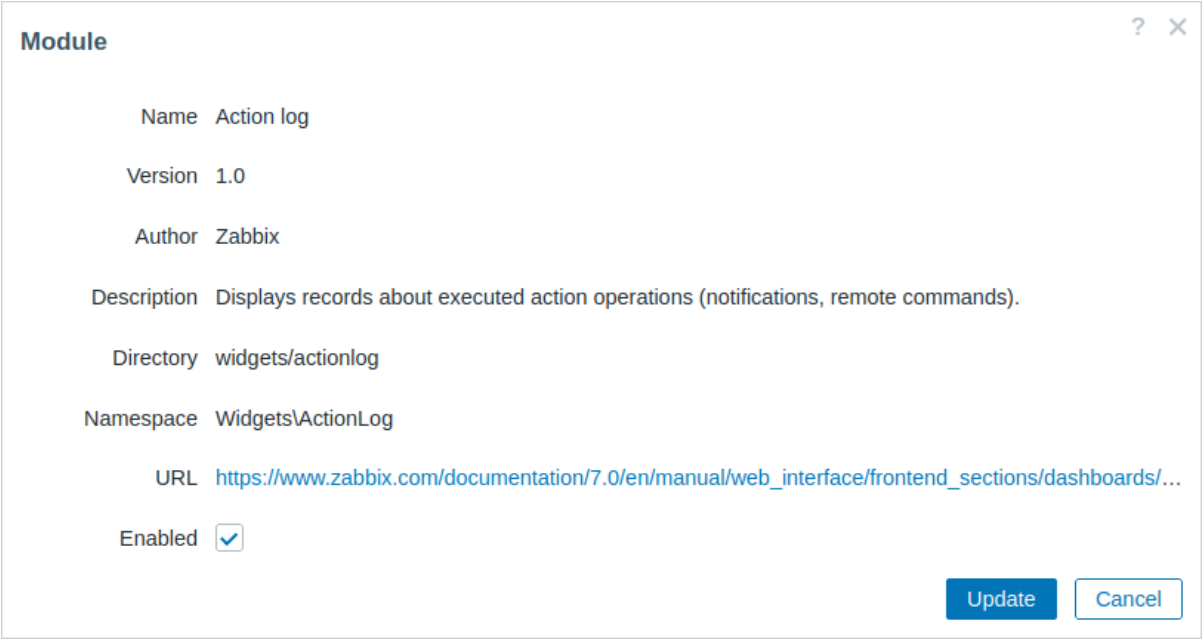
> **Note:**
> Wenn das Feature-Flag `$ZBX_FEATURE_FLAGS['modules_config_enabled']` in der Frontend-Konfigurationsdatei auf `false` gesetzt ist, ist der Zugriff auf diesen Abschnitt deaktiviert.



Klicken Sie auf *Verzeichnis scannen*, um benutzerdefinierte Module zu registrieren bzw. deren Registrierung aufzuheben. Registrierte Module werden in der Liste angezeigt; die Registrierung aufgehobener Module wird aus der Liste entfernt.

Klicken Sie in der Liste auf den Modulstatus, um ein Modul zu aktivieren bzw. zu deaktivieren. Sie können Module auch gesammelt aktivieren bzw. deaktivieren, indem Sie sie in der Liste auswählen und anschließend auf die Schaltflächen *Aktivieren/Deaktivieren* unterhalb der Liste klicken.

Klicken Sie in der Liste auf den Modulnamen, um dessen Details in einem Pop-up-Fenster anzuzeigen.

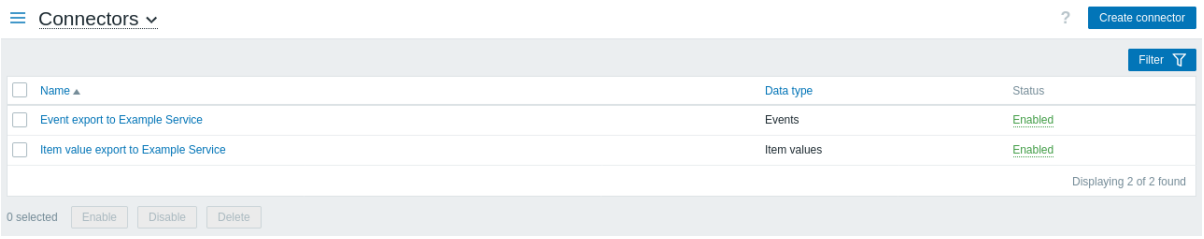


Der Modulstatus kann auch im Pop-up-Fenster mit den Moduldetails aktualisiert werden. Markieren bzw. deaktivieren Sie dazu das Kontrollkästchen *Aktiviert* und klicken Sie anschließend auf *Aktualisieren*.

Sie können Module nach Name oder Status (aktiviert/deaktiviert) filtern.

Konnektoren

In diesem Abschnitt können Sie Konnektoren für das Streaming von Zabbix-Daten über HTTP an externe Systeme konfigurieren.



Klicken Sie auf *Konnektor erstellen*, um einen neuen Konnektor zu konfigurieren.

Sie können Konnektoren nach Name oder Status (aktiviert/deaktiviert) filtern.
Klicken Sie in der Liste auf den Status eines Konnektors, um ihn zu aktivieren oder zu deaktivieren.
Sie können Konnektoren auch gesammelt aktivieren oder deaktivieren, indem Sie sie in der Liste auswählen und anschließend auf die Schaltflächen *Aktivieren/Deaktivieren* unterhalb der Liste klicken.

Sonstiges

In diesem Abschnitt können verschiedene sonstige Frontend-Parameter konfiguriert werden.



Parameter	Beschreibung
Frontend-URL	URL (bis zu 2048 Zeichen) zur Zabbix-Weboberfläche. Dieser Parameter wird vom Zabbix-Webservice für die Kommunikation mit dem Frontend verwendet und muss angegeben werden, um geplante Berichte zu aktivieren.
Gruppe für entdeckte Hosts	Hosts, die durch Netzwerkerkennung und Agent-Autoregistrierung entdeckt wurden, werden automatisch in die hier ausgewählte Host-Gruppe aufgenommen.
Standardmodus für Host-Inventar	Standard-Modus für das Host-Inventar. Er wird immer verwendet, wenn ein neuer Host oder Host-Prototyp durch den Server oder das Frontend erstellt wird, sofern er nicht während der Host-Erkennung/-Autoregistrierung durch die Operation *Host-Inventarmodus setzen* überschrieben wird.

Parameter	Beschreibung
Benutzergruppe für Datenbank-nicht-verfügbar-Meldung	Benutzergruppe zum Senden der Alarmmeldung bei nicht verfügbarer Datenbank. Der Zabbix Server ist von der Verfügbarkeit der Backend-Datenbank abhängig. Ohne Datenbank kann er nicht arbeiten. Wenn die Datenbank nicht verfügbar ist, können ausgewählte Benutzer von Zabbix benachrichtigt werden. Benachrichtigungen werden an die hier festgelegte Benutzergruppe unter Verwendung der aktivierten Benutzermedieneinträge gesendet. Benachrichtigungen werden über die folgenden Medientypen übertragen (wenn aktiviert): E-Mail; SMS; benutzerdefinierte Alarmskripte. Auch wenn ein webhook-Medieneintrag konfiguriert und aktiviert ist, löst er keine Benachrichtigungen aus. Der Zabbix Server wird nicht gestoppt; er wartet, bis die Datenbank wieder verfügbar ist, um die Verarbeitung fortzusetzen. Die Benachrichtigung besteht aus folgendem Inhalt: `[MySQL|PostgreSQL] database <DB Name> [on <DB Host>:<DB Port>] is not available: <error message depending on the type of DBMS (database)>` <DB Host> wird der Meldung nicht hinzugefügt, wenn er als leerer Wert definiert ist, und <DB Port> wird nicht hinzugefügt, wenn er den Standardwert („0“) hat. Der Alert-Manager (ein spezieller Zabbix-Serverprozess) versucht alle 10 Sekunden, eine neue Verbindung zur Datenbank herzustellen. Wenn die Datenbank weiterhin nicht verfügbar ist, wiederholt der Alert-Manager das Senden von Alarmmeldungen, jedoch nicht häufiger als alle 15 Minuten.
Nicht zugeordnete SNMP-Traps protokollieren	SNMP-Trap protokollieren, wenn keine entsprechenden SNMP-Schnittstellen gefunden wurden.

Autorisierung

Parameter	Beschreibung
Anmeldeversuche	Anzahl der erfolglosen Anmeldeversuche, bevor die Möglichkeit zur Anmeldung gesperrt wird.
Sperrintervall für die Anmeldung	Zeitraum, für den die Anmeldung untersagt wird, wenn das Limit für *Anmeldeversuche* überschritten wird. Zulässiger Bereich: 30 - 3600 Sekunden. Zeitsuffixe, z. B. 90s, 5m, 1h, werden unterstützt.

Speicherung von Geheimnissen

Parameter	Beschreibung
Vault provider	Wählen Sie die Software zur Geheimnisverwaltung für die Speicherung von Werten von Benutzermakros aus - *HashiCorp Vault* (Standard) oder *CyberArk Vault*.
Resolve secret vault macros by	Geheimnis-Vault-Makros auflösen durch: **Zabbix Server** - Geheimnisse werden vom Zabbix Server abgerufen und bei Bedarf an Proxys weitergeleitet (Standard); **Zabbix Server und Proxy** - Geheimnisse werden sowohl vom Zabbix Server als auch von Proxys abgerufen, sodass sie Makros unabhängig voneinander auflösen können.

Siehe auch: Speicherung von Geheimnissen.

Sicherheit

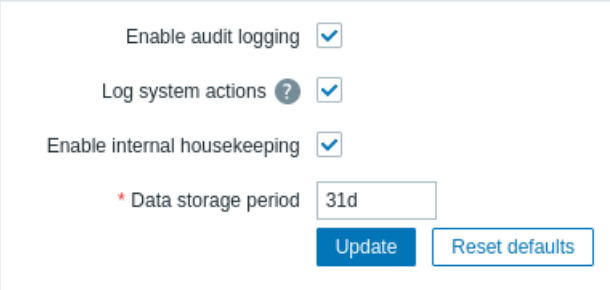
Parameter	Beschreibung
URI-Schemata validieren	Deaktivieren Sie dieses Kontrollkästchen, um die Validierung von URI-Schemata zu deaktivieren (standardmäßig aktiviert). Wenn es aktiviert ist, können Sie eine kommagetrennte Liste zulässiger URI-Schemata angeben (Standard: http,https,ftp,file,mailto,tel,ssh). Dies gilt für alle Felder im Frontend, in denen URIs verwendet werden (zum Beispiel URLs von Kartenelementen).

Parameter	Beschreibung
HTTP-Header X-Frame-Options verwenden	Deaktivieren Sie dieses Kontrollkästchen, um den HTTP-Header X-Frame-Options zu deaktivieren (nicht empfohlen). Wenn es aktiviert ist, können Sie den Wert des HTTP-Headers X-Frame-Options angeben. Unterstützte Werte: **SAMEORIGIN** (Standard) oder **'self'** (muss in einfache Anführungszeichen gesetzt werden) - die Seite kann nur in einem Frame mit derselben Herkunft wie die Seite selbst angezeigt werden; **DENY** oder **'none'** (muss in einfache Anführungszeichen gesetzt werden) - die Seite kann unabhängig von der Website, die dies versucht, nicht in einem Frame angezeigt werden; **eine Zeichenfolge aus durch Leerzeichen getrennten Hostnamen**; durch Hinzufügen von **'self'** (muss in einfache Anführungszeichen gesetzt werden) zur Liste kann die Seite in einem Frame mit derselben Herkunft wie die Seite selbst angezeigt werden. Beachten Sie, dass **'self'** oder **'none'** ohne einfache Anführungszeichen als Hostnamen behandelt werden.
iframe-Sandboxing verwenden	Deaktivieren Sie dieses Kontrollkästchen, um das Einfügen des Inhalts der abgerufenen URL in eine Sandbox zu deaktivieren (nicht empfohlen). Wenn es aktiviert ist, können Sie die Ausnahmen für das iframe-Sandboxing angeben; nicht angegebene Einschränkungen werden weiterhin angewendet. Wenn dieses Feld leer ist, gelten alle Einschränkungen des sandbox-Attributs. Weitere Informationen finden Sie in der Beschreibung des Attributs `sandbox`.

Auditprotokoll

Übersicht

In diesem Abschnitt können die Einstellungen für das Audit-Log konfiguriert werden.



Die folgenden Parameter sind verfügbar:

Parameter	Beschreibung
Audit-Protokollierung aktivieren	Aktiviert (Standard) oder deaktiviert die Audit-Protokollierung.
Systemaktionen protokollieren	Aktiviert (Standard) oder deaktiviert die Audit-Protokollierung von Low-Level-Discovery-, Netzwerk-Discovery- und Autoregistrierungsaktivitäten, die vom Server (Systembenutzer) durchgeführt werden.
Interne Bereinigung aktivieren	Aktiviert (Standard) oder deaktiviert die interne Bereinigung für Audit-Log-Einträge.
Speicherzeitraum der Daten	Anzahl der Tage, für die Audit-Log-Einträge aufbewahrt werden, bevor sie vom Housekeeper entfernt werden. Erforderlich, wenn die Bereinigung aktiviert ist. Standard: 31 Tage.

Die Parameter für die Audit-Log-Komprimierung sind im Abschnitt *Administration* > *Housekeeping* im Block *History, trends and audit log compression* verfügbar, der sichtbar wird, wenn TimescaleDB verwendet wird.

Housekeeping

Übersicht

Der Abschnitt *Administration > Housekeeping* ermöglicht es Ihnen, das interne Zabbix-Housekeeping zu konfigurieren, das veraltete oder vom Benutzer gelöschte Daten automatisch aus der Datenbank entfernt. Dadurch wird verhindert, dass die Datenbank unbegrenzt wächst, was unnötigen Speicherplatz belegen und die Datenbankleistung beeinträchtigen würde.

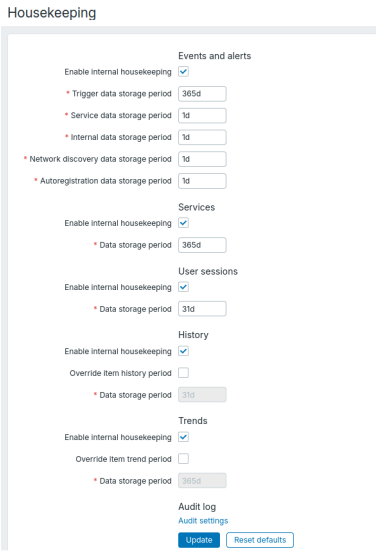
Housekeeping kann für die folgenden Datentypen aktiviert und konfiguriert werden:

- Ereignisse und Alarme aus Auslösern, Services, internen Daten, Netzwerkerkennung und automatischer Registrierung
- Services
- Benutzersitzungen
- History und Trends für Datenpunkte

Note:
Das Housekeeping für das Audit-Log wird separat konfiguriert.

Konfiguration

Die folgenden Parameter sind verfügbar:



Parameter	Beschreibung
Interne Housekeeping-Funktion aktivieren	Interne Housekeeping-Funktion aktivieren oder deaktivieren (standardmäßig aktiviert). Wenn aktiviert, entfernt das Housekeeping-Verfahren automatisch Daten aus der Datenbank, die den *Datenspeicherzeitraum* überschreiten.
Datenspeicherzeitraum	Geben Sie an, wie lange Daten aufbewahrt werden sollen, bevor sie vom Housekeeper entfernt werden. Erforderlich, wenn die interne Housekeeping-Funktion aktiviert ist. Bereich: 1 Tag (1 Stunde für History) - 25 Jahre; oder ”0”. Zeitsuffixe (z. B. 1d, 1w) werden unterstützt. Für *Events und Alarme* wird der Datenspeicherzeitraum separat für Auslöser, Services, interne Daten, Netzwerkerkennung und automatische Registrierung festgelegt. Der Housekeeper entfernt nur solche Events, die nicht mit Problemen verknüpft sind. Beispielsweise wird ein Problem-/Wiederherstellungs-Event, das älter als der *Datenspeicherzeitraum* ist, nicht entfernt, wenn es mit einem ungelösten Problem verknüpft ist. Wenn der Housekeeper veraltete Entitäten entfernt, entfernt er zuerst Probleme und dann Events. Für *History* und *Trends* bestimmen die Datenspeicherzeiträume außerdem, wie lange Daten in *Monitoring > Neueste Daten* sichtbar bleiben, selbst wenn die interne Housekeeping-Funktion deaktiviert ist. Beachten Sie, dass beim Löschen eines Datenpunkts oder Auslösers auch dessen Probleme und zugehörige Events gelöscht werden.

Parameter	Beschreibung
History-Zeitraum des Datenpunkts überschreiben	Wenn aktiviert, wird der in der Datenpunktkonfiguration angegebene Speicherzeitraum für History/Trends durch die Einstellung *Datenspeicherzeitraum* überschrieben (außer für Datenpunkte, für die die Option *Nicht speichern* aktiviert ist). Diese Option kann auch verwendet werden, wenn die interne Housekeeping-Funktion deaktiviert ist und ein externer Housekeeper verwendet wird.
Trend-Zeitraum des Datenpunkts überschreiben	

Wenn **ClickHouse** verwendet wird, wird der Abschnitt *ClickHouse-Datenspeicherzeitraum* verfügbar. Er zeigt den Datenspeicherzeitraum für jeden Datenpunkt-Werttyp an. Um den Zeitraum zu ändern, siehe ClickHouse-Datenspeicherzeitraum konfigurieren. Beachten Sie, dass der Housekeeper keine Daten aus ClickHouse löscht.

Wenn **Elasticsearch** verwendet wird, wird der Abschnitt *Elasticsearch-Datenspeicherzeitraum* verfügbar. Er zeigt die in Elasticsearch gespeicherten Datenpunkt-Werttypen an, wobei der Datenspeicherzeitraum als *Unbekannt* angezeigt wird. Um zu steuern, wie lange Daten aufbewahrt werden, siehe Index Lifecycle Management. Beachten Sie, dass der Housekeeper keine Daten aus Elasticsearch löscht.

Wenn **TimescaleDB** verwendet wird, wird der Abschnitt *Komprimierung von History, Trends und Audit-Log* verfügbar.

Note:
Für TimescaleDB aktivieren Sie *History-Zeitraum des Datenpunkts überschreiben*, *Trend-Zeitraum des Datenpunkts überschreiben* und *Interne Housekeeping-Funktion aktivieren* für History und Trends, um vollständig von der automatischen Partitionierung zu profitieren. Wenn diese Optionen deaktiviert sind, werden die in den History- und Trends-Tabellen gespeicherten Daten weiterhin partitioniert, aber der Housekeeper entfernt keine veralteten Partitionen, und es werden Konfigurationswarnungen angezeigt. Wenn das Entfernen veralteter Partitionen aktiviert ist, verfolgen Zabbix Server und Frontend gelöschte Datenpunkte nicht mehr, und die History für diese Datenpunkte wird gelöscht, wenn eine veraltete Partition entfernt wird.

Der Housekeeper wird standardmäßig automatisch jede Stunde ausgeführt. Sie können die Ausführungsfrequenz ändern, indem Sie den Zabbix-Server-Konfigurationsparameter `HousekeepingFrequency` festlegen.

Die Schaltfläche *Standardwerte zurücksetzen* ermöglicht es Ihnen, alle vorgenommenen Änderungen rückgängig zu machen.

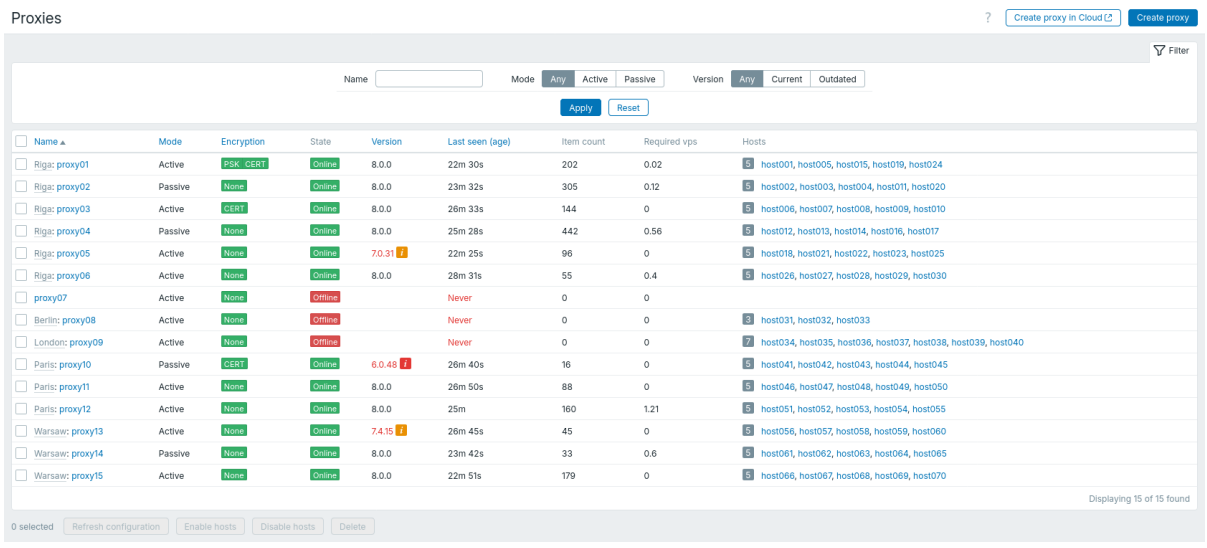
Proxies

Übersicht

Im Bereich *Administration* > *Proxies* können Proxies für die verteilte Überwachung im Zabbix Frontend konfiguriert werden.

Proxys

Eine Liste der vorhandenen Proxys mit ihren Details wird angezeigt.



Angezeigte Daten:

Spalte	Beschreibung
Name	Name des Proxys. Durch Klicken auf den Namen des Proxys wird das Proxy-Konfigurationsformular geöffnet. Wenn der Proxy zu einer Proxygruppe gehört, wird der Gruppenname vor dem Namen des Proxys als grauer Link angezeigt. Durch Klicken auf den Gruppennamen wird das Konfigurationsformular der Proxygruppe geöffnet.
Modus	Proxy-Modus - *Aktiv* oder *Passiv*.
Verschlüsselung	Verschlüsselungsstatus für Verbindungen vom Proxy: **Keine** - keine Verschlüsselung; **PSK** - Verwendung eines vorab gemeinsam vereinbarten Schlüssels; **Zertifikat** - Verwendung eines Zertifikats.
Status	Status des Proxys: **Unbekannt** - der Proxy wurde erstellt, während der Zabbix Server nicht verfügbar war, oder der Server hat den Status noch nicht aktualisiert; **Online** - der Proxy hat innerhalb des Failover-Zeitraums mit dem Zabbix Server kommuniziert (der passive Proxy hat auf eine Serveranfrage geantwortet; der aktive Proxy hat eine Anfrage gesendet); **Offline** - der Proxy hat innerhalb des Failover-Zeitraums nicht mit dem Zabbix Server kommuniziert.
Version	Proxy-Version (dreistellige Versionsnummer). Wenn der Proxy veraltet oder nicht unterstützt wird, wird die Versionsnummer hervorgehoben (rot) und ein Info-Statussymbol (gelb oder rot) angezeigt. Bewegen Sie den Mauszeiger über das Symbol, um Details anzuzeigen.
Zuletzt gesehen (Alter)	Der Zeitpunkt, zu dem der Proxy zuletzt vom Server erkannt wurde.
Datenpunktanzahl	Die Anzahl der aktivierten Datenpunkte auf aktivierten Hosts, die dem Proxy zugewiesen sind.
Erforderliche vps	Erforderliche Proxy-Leistung (die Anzahl der Werte, die pro Sekunde erfasst werden müssen).
Hosts	Anzahl der aktivierten Hosts, die dem Proxy zugewiesen sind, sowie eine Liste der vom Proxy überwachten Hosts. Durch Klicken auf den Hostnamen wird das Host-Konfigurationsformular geöffnet.

Um einen neuen Proxy zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Proxy erstellen*. Um einen neuen Proxy in Zabbix Cloud bereitzustellen, klicken Sie daneben auf die Schaltfläche *Proxy in der Cloud erstellen*. Dadurch wird Zabbix Cloud in einer neuen Browserregisterkarte geöffnet, in der Sie einen Proxy-Knoten erstellen können. Nachdem der Proxy-Knoten erstellt wurde, konfigurieren Sie den Proxy über die Schaltfläche *Proxy erstellen*. Weitere Informationen finden Sie unter Einen Zabbix Cloud-Proxy einrichten.

Optionen für die Massenbearbeitung

Die Schaltflächen unterhalb der Liste bieten einige Optionen für die Massenbearbeitung:

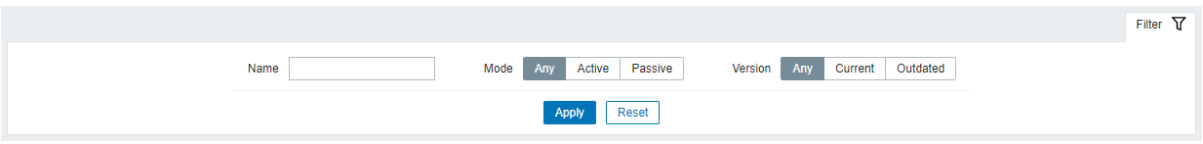
- *Konfiguration aktualisieren* - die Konfiguration der Proxies aktualisieren;
- *Hosts aktivieren* - den Status der vom Proxy überwachten Hosts auf *Überwacht* ändern;
- *Hosts deaktivieren* - den Status der vom Proxy überwachten Hosts auf *Nicht überwacht* ändern;
- *Löschen* - die Proxies löschen.

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Proxies und klicken Sie dann auf die gewünschte Schaltfläche.

Filter verwenden

Sie können den Filter verwenden, um nur die Proxys anzuzeigen, die Sie interessieren. Für eine bessere Suchleistung werden Daten mit nicht aufgelösten Makros durchsucht.

Der Link *Filter* ist oberhalb der Liste der Proxys verfügbar. Wenn Sie darauf klicken, wird ein Filter angezeigt, mit dem Sie Proxys nach Name, Modus und Version filtern können. Beachten Sie, dass die Filteroption *Veraltet* sowohl veraltete (teilweise unterstützte) als auch nicht unterstützte Proxys anzeigt.



Proxy-Gruppen

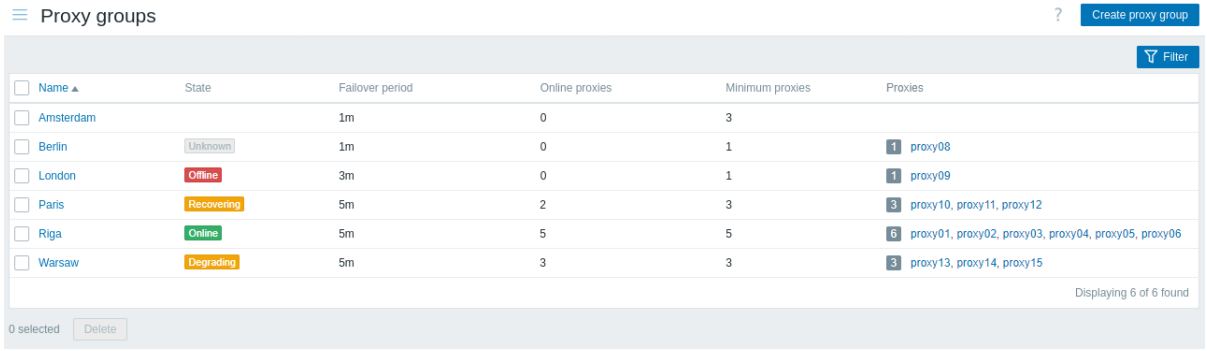
Übersicht

Unter *Administration* > *Proxy groups* ist es möglich, Proxy-Gruppen zu konfigurieren.

Proxy-Gruppen werden beim Proxy-Load-Balancing verwendet, um Hosts automatisch zwischen Proxys zu verteilen und eine hohe Verfügbarkeit zwischen Proxys sicherzustellen.

Proxy-Gruppen

Eine Liste der vorhandenen Proxy-Gruppen mit ihren Details wird angezeigt.



Angezeigte Daten:

Spalte	Beschreibung
Name	Name der Proxy-Gruppe. Durch Klicken auf den Namen der Proxy-Gruppe wird das Konfigurationsformular der Proxy-Gruppe geöffnet.
Status	Status der Proxy-Gruppe (wird angezeigt, wenn die Gruppe mindestens einen Proxy enthält): **Unbekannt** – Die Gruppe wurde erstellt, während der Zabbix-Server nicht verfügbar war, oder der Server hat den Status noch nicht aktualisiert; **Online** – Die Mindestanzahl an Proxys hat innerhalb des Failover-Zeitraums mit dem Zabbix-Server kommuniziert; **Wird schlechter** – Die Gruppe wird in Kürze offline gehen, da die Anzahl der Online-Proxys unter den Schwellenwert *Minimale Anzahl an Proxys* gefallen ist; **Offline** – Weniger als die Mindestanzahl an Proxys hat innerhalb des Failover-Zeitraums mit dem Zabbix-Server kommuniziert; **Wird wiederhergestellt** – Die Gruppe wird in Kürze wieder online sein, da die Anzahl der Online-Proxys den Schwellenwert *Minimale Anzahl an Proxys* erreicht hat. Der Status der Proxy-Gruppe kann mit den Datenpunkten zabbix[proxy group,<name>,state] und zabbix[proxy group,discovery] überwacht werden.
Failover-Zeitraum	Zeitraum, innerhalb dessen ein Proxy in der Proxy-Gruppe mit dem Zabbix-Server kommunizieren muss, um als online zu gelten.
Online-Proxys	Anzahl der Online-Proxys (wird rot angezeigt, wenn sie unter dem Gruppenminimum liegt).
Minimale Anzahl an Proxys	Mindestanzahl der Online-Proxys, die erforderlich ist, damit die Proxy-Gruppe online bleibt.
Proxys	Anzahl der Proxys in der Gruppe sowie eine Liste der Proxys in der Gruppe mit Links zum Proxy-Konfigurationsformular. Die maximale Anzahl der aufgelisteten Proxys wird durch den Wert *Maximale Anzahl der Elemente, die innerhalb einer Tabellenzelle angezeigt werden* begrenzt.

Um eine neue Proxy-Gruppe zu konfigurieren, klicken Sie oben rechts auf die Schaltfläche *Proxy-Gruppen erstellen*.

Optionen für die Massenbearbeitung

Die Schaltflächen unter der Liste bieten einige Optionen für die Massenbearbeitung:

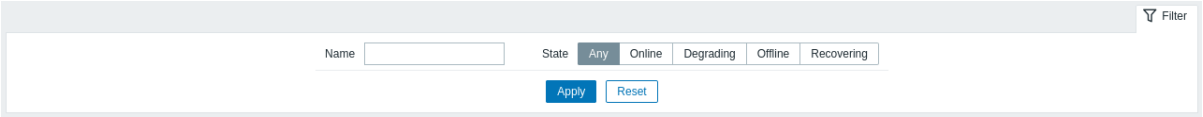
- *Löschen* - die Proxy-Gruppen löschen.

Um diese Optionen zu verwenden, markieren Sie die Kontrollkästchen vor den jeweiligen Proxy-Gruppen und klicken Sie dann auf die gewünschte Schaltfläche.

Filter verwenden

Sie können den Filter verwenden, um nur die Proxy-Gruppen anzuzeigen, die Sie interessieren. Für eine bessere Suchleistung werden die Daten mit nicht aufgelösten Makros durchsucht.

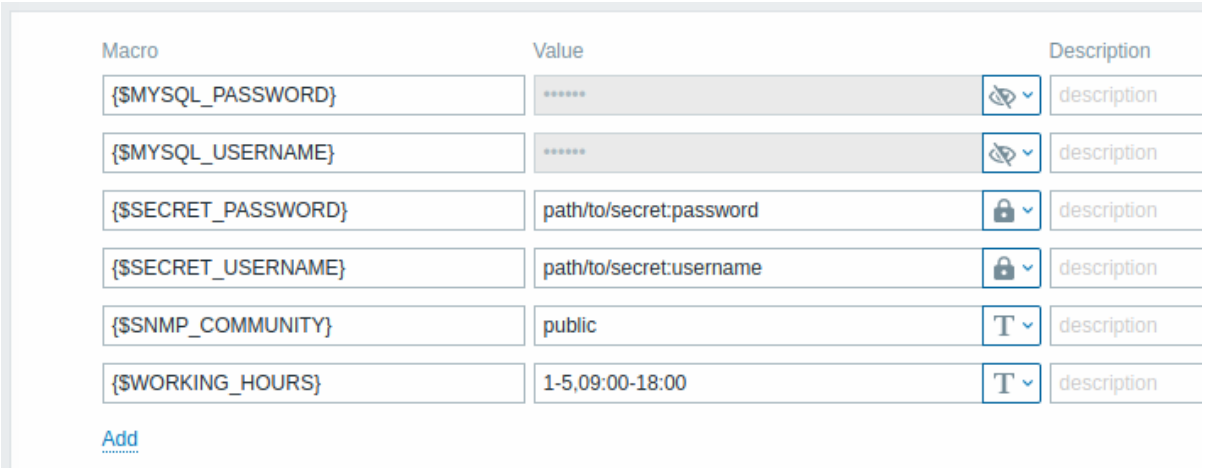
Der Link *Filter* ist oberhalb der Liste der Proxy-Gruppen verfügbar. Wenn Sie darauf klicken, wird ein Filter angezeigt, mit dem Sie Proxy-Gruppen nach Name und Status filtern können.



Makros

Übersicht

In diesem Abschnitt können Sie systemweite Benutzermakros als Name-Wert-Paare definieren. Beachten Sie, dass Makrowerte als Klartext, geheimer Text oder Vault-Geheimnis gespeichert werden können. Das Hinzufügen einer Beschreibung wird ebenfalls unterstützt.



Warteschlange

Übersicht

Im Bereich *Administration* > *Warteschlange* werden Datenpunkte angezeigt, deren Aktualisierung noch aussteht.

Idealerweise sollte beim Öffnen dieses Bereichs alles „grün“ sein, das heißt, es sollten sich keine Datenpunkte in der Warteschlange befinden. Wenn alle Datenpunkte ohne Verzögerung aktualisiert werden, warten keine Datenpunkte. Aufgrund unzureichender Serverleistung kann es jedoch zu Verzögerungen bei einigen Datenpunkten kommen. Die entsprechenden Informationen werden in diesem Bereich angezeigt. Weitere Informationen finden Sie im Abschnitt Warteschlange.

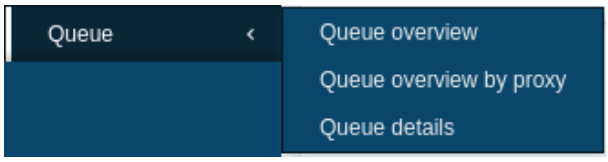
Note:
Der Zabbix Proxy kennt keine Wartungszeiträume. Weitere Informationen finden Sie unter Berechnung der Warteschlangen während der Wartung.

Note:
Die Warteschlange ist nur verfügbar, wenn der Zabbix Server ausgeführt wird. Datenpunkte werden nicht zur Warteschlange gezählt, wenn die Schnittstelle des Datenpunkts aufgrund von Verbindungsproblemen oder eines nicht ordnungsgemäß funktionierenden Agents nicht verfügbar ist.

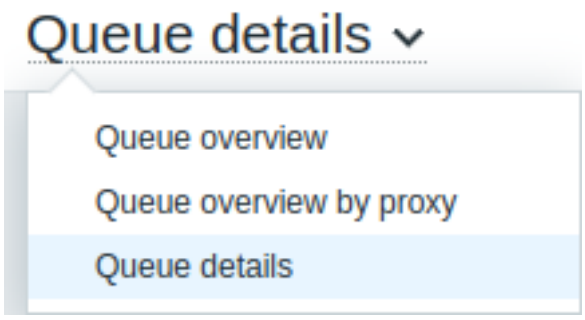
Der Bereich *Administration* > *Warteschlange* enthält die folgenden Seiten:

- Warteschlangenübersicht – zeigt die Warteschlange nach Datentyp an;
- Warteschlangenübersicht nach Proxy – zeigt die Warteschlange nach Proxy an;
- Warteschlangendetails – zeigt eine Liste der verzögerten Datenpunkte an.

Die Liste der verfügbaren Seiten wird angezeigt, wenn Sie im Menübereich *Administration* auf *Warteschlange* klicken. Sie können auch über das Titelauswahlmenü oben links zwischen den Seiten wechseln.



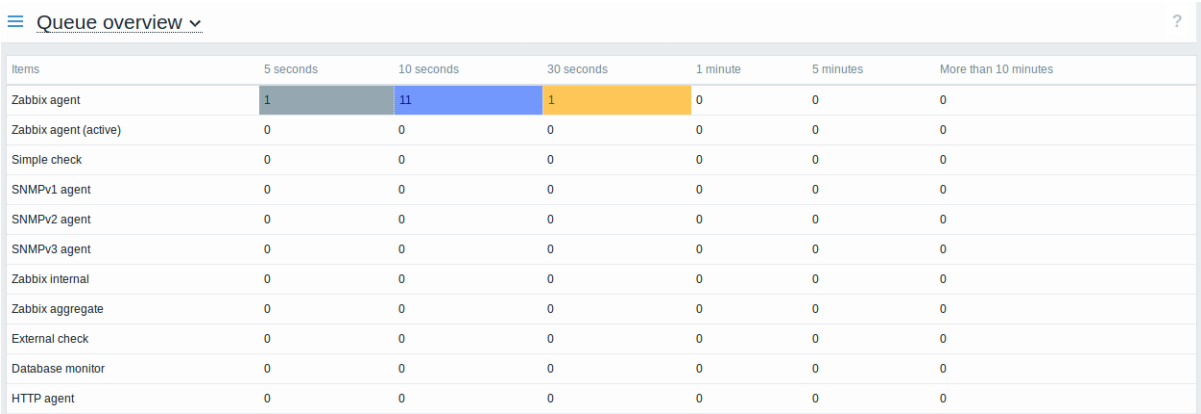
Menü der dritten Ebene.



Titelauswahlmenü.

Übersicht nach Datenpunkttyp

In diesem Bildschirm lässt sich leicht erkennen, ob das Problem mit einem oder mehreren Datenpunkttypen zusammenhängt.



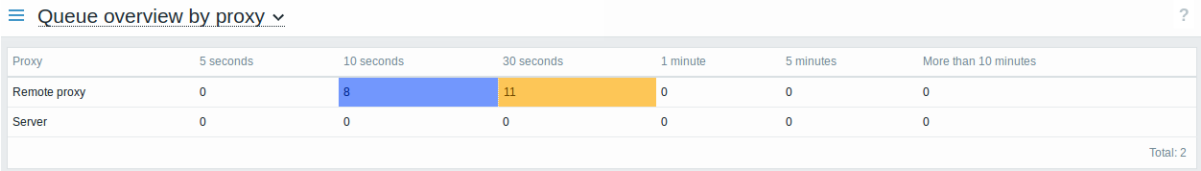
Queue overview

Items	5 seconds	10 seconds	30 seconds	1 minute	5 minutes	More than 10 minutes
Zabbix agent	1	11	1	0	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

Jede Zeile enthält einen Datenpunkttyp. Jede Spalte zeigt die Anzahl der wartenden Datenpunkte – jeweils wartend seit 5–10 Sekunden/10–30 Sekunden/30–60 Sekunden/1–5 Minuten/5–10 Minuten oder über 10 Minuten.

Übersicht nach Proxy

Auf diesem Bildschirm lässt sich leicht erkennen, ob das Problem mit einem der Proxys oder dem Server zusammenhängt.



Queue overview by proxy

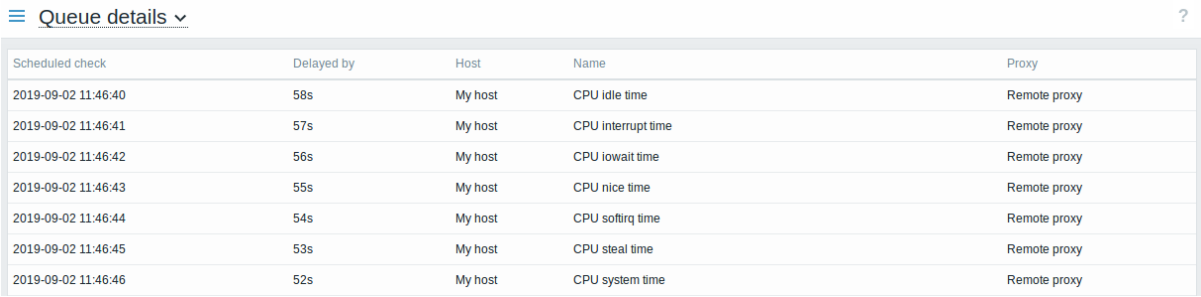
Proxy	5 seconds	10 seconds	30 seconds	1 minute	5 minutes	More than 10 minutes
Remote proxy	0	8	11	0	0	0
Server	0	0	0	0	0	0

Total: 2

Jede Zeile enthält einen Proxy, wobei der Server als Letzter in der Liste steht. Jede Spalte zeigt die Anzahl der wartenden Datenpunkte an – jeweils mit einer Wartezeit von 5–10 Sekunden/10–30 Sekunden/30–60 Sekunden/1–5 Minuten/5–10 Minuten oder über 10 Minuten.

Liste der wartenden Datenpunkte

Auf diesem Bildschirm wird jeder wartende Datenpunkt aufgelistet.



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

Angezeigte Daten:

Spalte	Beschreibung
Geplanter Check	Die Zeit, zu der der Check fällig war, wird angezeigt.
Verzögert um	Die Länge der Verzögerung wird angezeigt.
Host	Der Host des Datenpunkts wird angezeigt.
Name	Der Name des wartenden Datenpunkts wird angezeigt.
Proxy	Der Name des Proxy wird angezeigt, wenn der Host durch einen Proxy überwacht wird.

Mögliche Fehlermeldungen

Es kann vorkommen, dass keine Daten angezeigt werden und die folgende Fehlermeldung erscheint:



Die Fehlermeldung lautet in diesem Fall:

`Cannot display item queue. Permission denied`

Dies geschieht, wenn die PHP-Konfigurationsparameter in der Frontend-Konfigurationsdatei - `$ZBX_SERVER` oder sowohl `$ZBX_SERVER` als auch `$ZBX_SERVER_PORT` - auf einen vorhandenen Zabbix Server verweisen, der eine andere Datenbank verwendet.

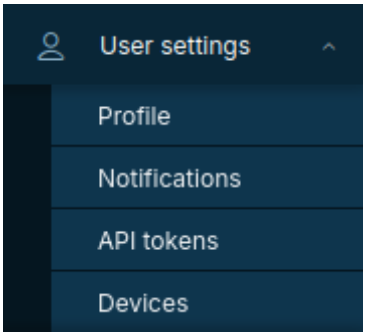
Benutzereinstellungen

Übersicht

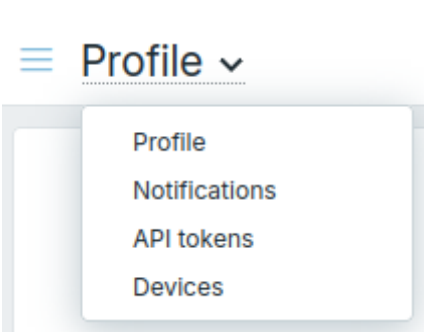
Abhängig von Ihrer Benutzerrolle kann der Abschnitt *Benutzereinstellungen* die folgenden Seiten enthalten:

- *Profil* oder *Benutzerprofil* – zum Anpassen bestimmter Zabbix-Frontend-Funktionen;
- *Benachrichtigungen* – zum Anpassen der Benachrichtigungen des aktuellen Benutzers;
- *API-Tokens* – zum Verwalten der dem aktuellen Benutzer zugewiesenen API-Tokens;
- *Geräte* – zum Verwalten der mobilen Geräte des aktuellen Benutzers.

Die Liste der verfügbaren Seiten wird angezeigt, wenn Sie auf das Benutzersymbol unten im Zabbix-Menü klicken (für den Benutzer *guest* nicht verfügbar). Sie können auch mithilfe der Titelauswahlliste in der oberen linken Ecke zwischen den Seiten wechseln.



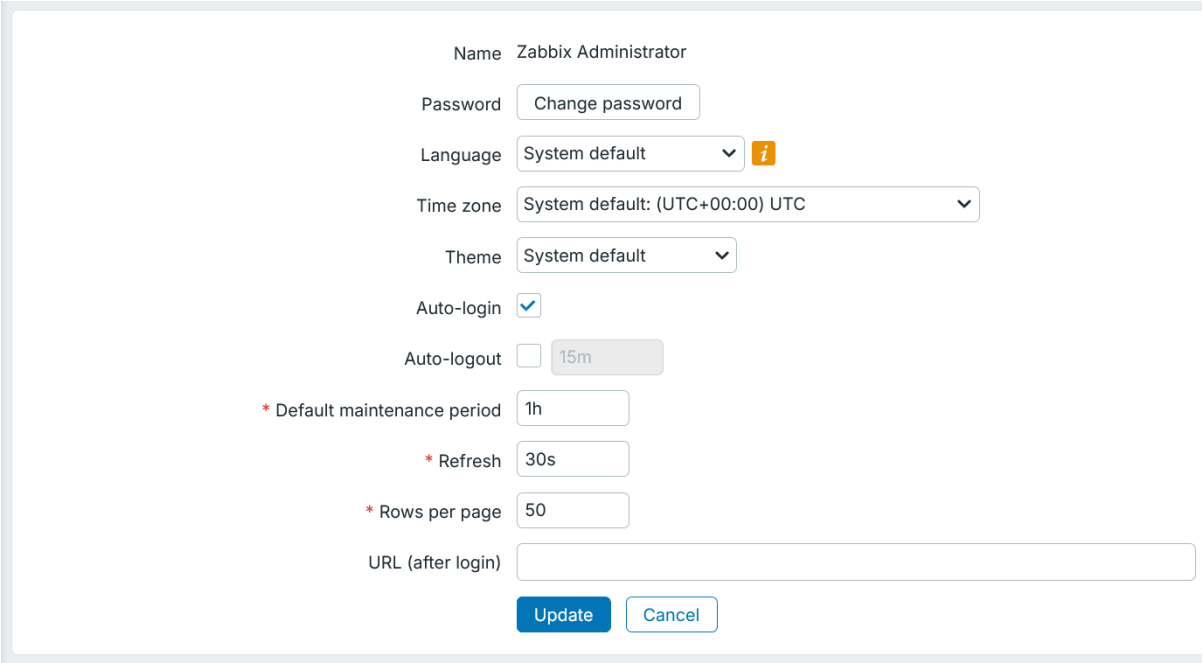
Menü der dritten Ebene.



Titelauswahlliste.

Benutzerprofil

Der Abschnitt *Benutzerprofil* bietet Optionen zum Festlegen einer benutzerdefinierten Sprache der Benutzeroberfläche, eines Farbdesigns, der Anzahl der in den Listen angezeigten Zeilen usw.
Die hier vorgenommenen Änderungen werden nur auf den aktuellen Benutzer angewendet.



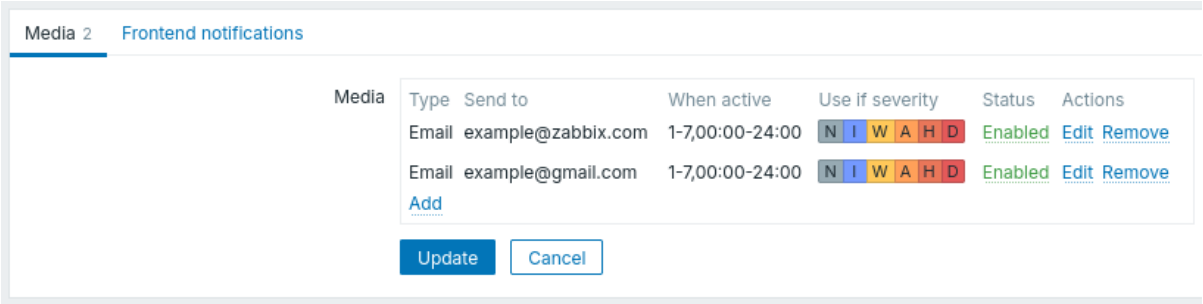
Parameter	Beschreibung
Name	Der Vor- und Nachname des Benutzers werden angezeigt. Wenn kein Vor- und Nachname vorhanden ist, wird der Benutzername angezeigt.
Passwort	Klicken Sie auf die Schaltfläche *Passwort ändern*, um drei Felder zu öffnen: *Altes Passwort*, *Neues Passwort*, *Neues Passwort (erneut)*. Nach einer erfolgreichen Passwortänderung wird der Benutzer von allen aktiven Sitzungen abgemeldet. Beachten Sie, dass das Passwort nur für Benutzer geändert werden kann, die die Zabbix-interne Authentifizierung verwenden.
Sprache	Wählen Sie die gewünschte Sprache der Benutzeroberfläche aus oder wählen Sie **Systemstandard**, um die Standardsystemeinstellungen zu verwenden. Durch die Auswahl von *Englisch (en_US)* wird im Frontend außerdem das US-amerikanische Zeit- und Datumsformat aktiviert. Weitere Informationen finden Sie unter Installation zusätzlicher Frontend-Sprachen.
Zeitzone	Wählen Sie die Zeitzone aus, um die globale Zeitzone auf Benutzerebene zu überschreiben, oder wählen Sie **Systemstandard**, um die globalen Zeitzoneneinstellungen zu verwenden.
Design	Wählen Sie ein Farbdesign speziell für Ihr Profil aus: **Systemstandard** - Standardsystemeinstellungen verwenden; **Blau** - standardmäßiges blaues Design; **Blau (klassisch)** - standardmäßiges blaues Design mit älteren Schriftarten; **Dunkel** - alternatives dunkles Design; **Dunkelblau** - dunkelblaues Design; **Dunkel (klassisch)** - alternatives dunkles Design mit älteren Schriftarten; **Hoher Kontrast, hell** - helles Design mit hohem Kontrast; **Hoher Kontrast, dunkel** - dunkles Design mit hohem Kontrast.
Automatische Anmeldung	Aktivieren Sie dieses Kontrollkästchen, damit Zabbix Sie 30 Tage lang speichert und automatisch anmeldet. Bei der Anmeldung mit aktivierter Option *30 Tage speichern*: - Der Timer für die automatische Abmeldung wird zurückgesetzt (die Sitzung bleibt bestehen, bis der Zeitraum von 30 Tagen endet). - Bei nachfolgenden Besuchen innerhalb von 30 Tagen wird die automatische Anmeldung aktiviert. Bei der Anmeldung ohne *30 Tage speichern*: - Der Parameter für die automatische Anmeldung wird gelöscht. - Die automatische Abmeldung folgt wie gewohnt dem konfigurierten Timeout. Hierfür werden Browser-Cookies verwendet.

Parameter	Beschreibung
Automatische Abmeldung	Wenn dieses Kontrollkästchen aktiviert ist, werden Sie nach der festgelegten Anzahl von Sekunden automatisch abgemeldet (mindestens 90 Sekunden, höchstens 1 Tag). Zeitsuffixe werden unterstützt, zum Beispiel: 90s, 5m, 2h, 1d. Beachten Sie, dass diese Option in den folgenden Fällen nicht funktioniert: * Wenn die Seiten des Menüs *Überwachung* Informationen im Hintergrund aktualisieren. Wenn Seiten, die Daten in einem bestimmten Zeitintervall aktualisieren (Dashboards, Graphen, neueste Daten usw.), geöffnet bleiben, wird die Sitzungsdauer entsprechend verlängert, wodurch die automatische Abmeldung deaktiviert wird. * Wenn die Anmeldung mit aktivierter Option *30 Tage angemeldet bleiben* erfolgt; in diesem Fall wird der Timer für die automatische Abmeldung überschrieben und Ihre Sitzung bleibt während des gesamten Zeitraums aktiv. * Wenn die Authentifizierung über einen SSO-Identitätsanbieter (IdP) erfolgt, beendet die automatische Abmeldung nur die Zabbix-Sitzung. Die SSO-Sitzung beim IdP bleibt aktiv, sodass sich der Benutzer möglicherweise erneut anmelden kann, ohne ein Passwort einzugeben. *Automatische Abmeldung* kann auch den Wert „0“ akzeptieren. In diesem Fall wird die Funktion zur automatischen Abmeldung nach der Aktualisierung der Profileinstellungen deaktiviert.
Standardwartungszeitraum	Legen Sie den Standardwartungszeitraum für Ad-hoc-Wartungen fest.
Aktualisierung	Legen Sie fest, wie oft die Informationen auf den Seiten des Menüs *Überwachung* aktualisiert werden (mindestens 0 Sekunden, höchstens 1 Stunde). Zeitsuffixe werden unterstützt, zum Beispiel: 30s, 90s, 1m, 1h.
Zeilen pro Seite	Legen Sie fest, wie viele Zeilen pro Seite in den Listen angezeigt werden. Weniger Zeilen (und weniger anzuzeigende Datensätze) führen zu kürzeren Ladezeiten.
URL (nach der Anmeldung)	Legen Sie eine bestimmte URL fest, die nach der Anmeldung angezeigt werden soll. Anstelle der standardmäßigen *Dashboards* kann dies beispielsweise die URL von *Überwachung* > *Auslöser* sein. Dieses Feld ist deaktiviert, wenn die URL nach der Anmeldung von einem *Superadministrator* für Ihre Benutzerrolle auf eine bestimmte Seite beschränkt wurde. Wenn das Feld deaktiviert ist oder der Benutzer keinen Wert angibt, wird der Standardwert unterhalb des Feldes angezeigt, z. B. `zabbix.php?action=dashboard.view`

Benachrichtigungen

Der Abschnitt *Benachrichtigungen* bietet Optionen zum Anpassen der Benachrichtigungen des aktuellen Benutzers.

Auf der Registerkarte *Medien* können Sie Mediendetails für den Benutzer festlegen, z. B. welche Medientypen und Adressen verwendet werden sollen und wann sie für die Zustellung von Benachrichtigungen verwendet werden sollen.



Wenn der Medientyp deaktiviert wurde:

- wird nach dem Namen ein gelbes Info-Symbol angezeigt;
- wird in der Spalte *Status* „Deaktiviert“ angezeigt.

Note:
Berechtigungen für Benutzer, ihre eigenen Mediendetails zu ändern, können basierend auf ihrer Benutzerrolle gewährt/entzogen werden (siehe Option *Eigene Medien erstellen und bearbeiten*). Berechtigungen für Super-Admin-Benutzer, Mediendetails anderer Benutzer zu ändern, können ebenfalls basierend auf ihrer Benutzerrolle gewährt/entzogen werden (siehe Option *Benutzermedien erstellen und bearbeiten*)

Beachten Sie, dass für bereitgestellte Benutzer gilt:

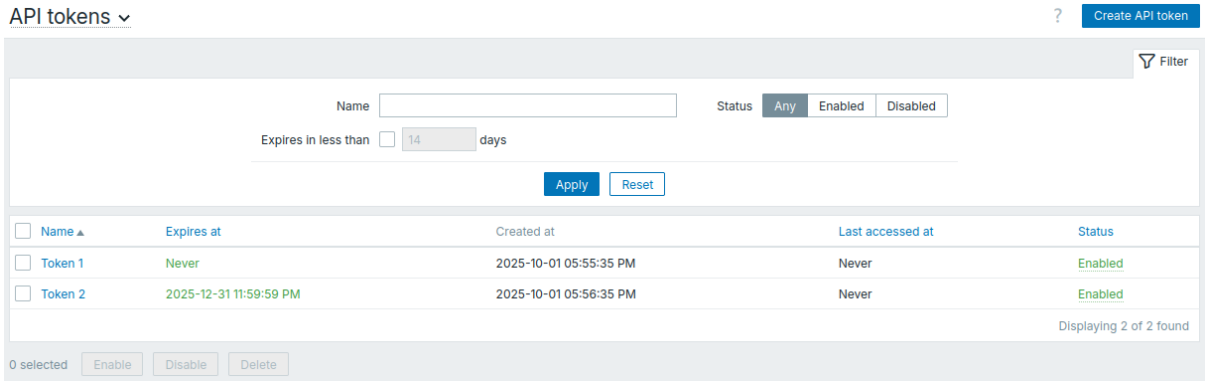
- bereitgestellte Benutzermedien können nicht gelöscht werden;
- bereitgestellte Benutzermedien können deaktiviert/aktiviert werden;

- Felder bereitgestellter Benutzermedien wie *Aktiv wenn*, *Verwenden bei Schweregrad* und *Aktiviert* können manuell bearbeitet werden;
- zusätzliche Benutzermedien können für bereitgestellte Benutzer manuell hinzugefügt werden (zum Beispiel zusätzliche E-Mail-Adressen);
- manuell hinzugefügte Benutzermedien können gelöscht werden.

Auf der Registerkarte *Frontend-Benachrichtigungen* können Sie globale Benachrichtigungen festlegen.

API-Tokens

Der Abschnitt *API-Tokens* ermöglicht es Ihnen, die dem Benutzer zugewiesenen Tokens anzuzeigen, die Tokendetails zu bearbeiten und neue Tokens zu erstellen. Dieser Abschnitt ist für einen Benutzer nur verfügbar, wenn die Aktion *API-Tokens verwalten* in den Einstellungen der Benutzerrolle erlaubt ist.



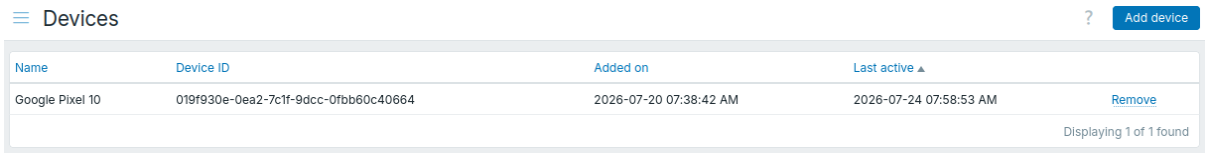
Sie können API-Tokens nach Name, Ablaufdatum oder Status (*Aktiviert/Deaktiviert*) filtern. Klicken Sie in der Liste auf den Token-Status, um einen Token schnell zu aktivieren/deaktivieren. Sie können außerdem mehrere Tokens gleichzeitig aktivieren/deaktivieren, indem Sie sie in der Liste auswählen und anschließend auf die Schaltflächen *Aktivieren/Deaktivieren* unterhalb der Liste klicken.

Attention:
Benutzer können den Wert des *Auth token* der ihnen zugewiesenen Tokens in Zabbix nicht anzeigen. Der Wert des *Auth token* wird nur einmal angezeigt - unmittelbar nach dem Erstellen eines Tokens. Wenn der Token verloren gegangen ist, muss er neu generiert werden.

Geräte

Im Abschnitt *Geräte* können Sie Ihre Mobilgeräte hinzufügen oder entfernen, die mit der Zabbix mobile app verwendet werden sollen.

Dieser Abschnitt wird angezeigt, wenn `EnableMobileDevices` aktiviert ist.



So fügen Sie ein Gerät hinzu:

1. Klicken Sie auf *Gerät hinzufügen*.
2. Scannen Sie den QR-Code in Ihrer Zabbix mobile app.

Note:
Wenn Sie ein Gerät hinzufügen, fügt Zabbix den Medientyp Push-Benachrichtigung auch zu Ihren Benutzermedien hinzu. Wenn Sie das Gerät entfernen, wird dieses Benutzermedium ebenfalls entfernt. Wenn Sie mehrere Medientypen für Push-Benachrichtigungen konfiguriert haben, fügt Zabbix den aktivierten Medientyp hinzu, der alphabetisch an erster Stelle steht.

Hinzugefügte Geräte werden in einer Liste angezeigt. Die Liste enthält ihren Namen und ihre ID sowie Datum und Uhrzeit des Hinzufügens und der letzten Aktivität.

Sie können Geräte auch entfernen. Wenn Sie ein Gerät entfernen, werden seine Zugangsdaten widerrufen und aus Zabbix entfernt. Danach kann das Gerät nicht mehr auf Zabbix zugreifen.

Super-Admins können auch die Geräte anderer Benutzer unter *Benutzer* > *Geräte* verwalten.

Globale Benachrichtigungen

Übersicht

Globale Benachrichtigungen bieten eine Möglichkeit, aktuelle Probleme in Echtzeit direkt auf Ihrem aktuellen Bildschirm im Zabbix Frontend anzuzeigen.

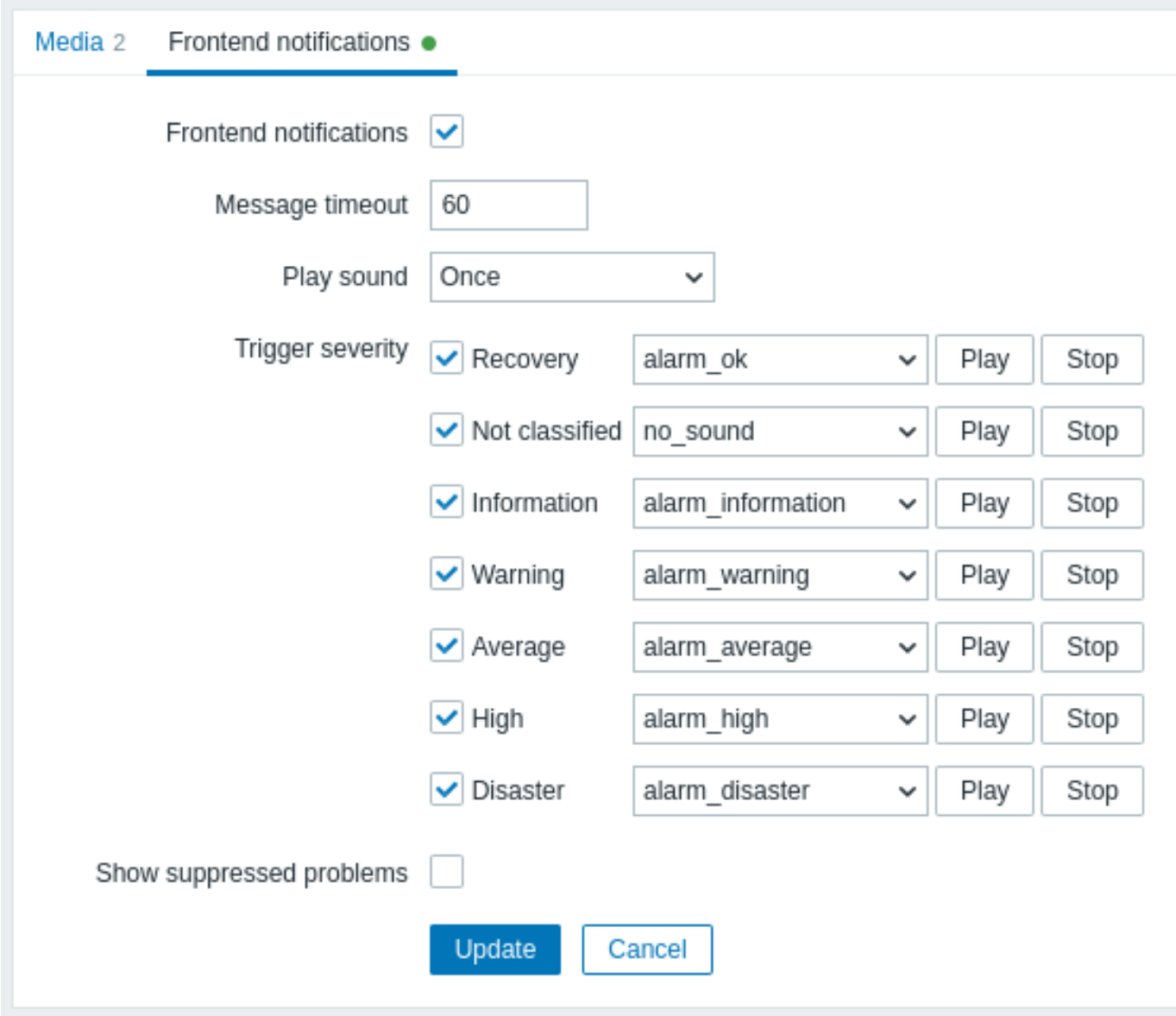
Ohne globale Benachrichtigungen würden Sie bei der Arbeit außerhalb der Bereiche *Probleme* oder *Dashboard* keine Informationen über aktuelle Probleme erhalten. Globale Benachrichtigungen stellen sicher, dass diese Informationen unabhängig von Ihrer aktuellen Position im Zabbix Frontend angezeigt werden.

Globale Benachrichtigungen umfassen sowohl das Anzeigen einer Nachricht als auch das Abspielen eines Tons.

Attention:
Die automatische Wiedergabe von Tönen könnte in aktuellen Browserversionen deaktiviert sein (standardmäßig). In solchen Fällen müssen Sie diese Einstellung manuell aktivieren.

Konfiguration

Globale Benachrichtigungen können pro Benutzer im Reiter **Frontend-Benachrichtigungen** im Abschnitt Benachrichtigungen aktiviert werden.

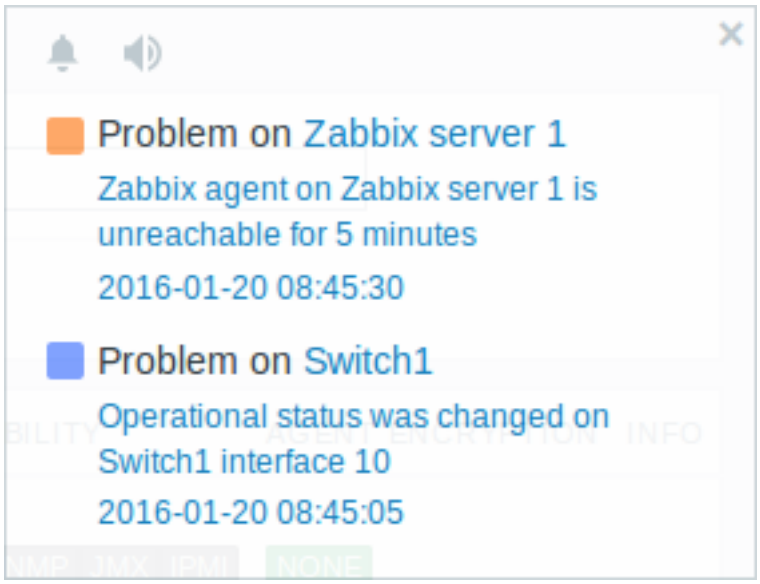


Parameter	Beschreibung
Frontend-Benachrichtigungen	Aktivieren Sie das Kontrollkästchen, um globale Benachrichtigungen zu aktivieren.

Parameter	Beschreibung
Zeitüberschreitung der Nachricht	Legen Sie die Dauer fest, für die die Nachricht angezeigt wird. Standardmäßig bleiben Nachrichten 60 Sekunden lang auf dem Bildschirm. Zeitsuffixe werden unterstützt, zum Beispiel: 30s, 5m, 2h, 1d.
Ton abspielen	Legen Sie die Dauer fest, für die der Ton abgespielt wird. **Einmal** - der Ton wird einmal vollständig abgespielt; **10 Sekunden** - der Ton wird 10 Sekunden lang wiederholt; **Zeitüberschreitung der Nachricht** - der Ton wird wiederholt, solange die Nachricht sichtbar ist.
Auslöser-Schweregrad	Legen Sie die Auslöser-Schweregrade fest, für die globale Benachrichtigungen und Töne aktiviert werden. Sie können auch passende Töne für verschiedene Schweregrade auswählen. Wenn kein Schweregrad markiert ist, werden keine Nachrichten angezeigt. Zusätzlich werden Wiederherstellungsmeldungen nur für markierte Schweregrade angezeigt. Wenn zum Beispiel *Wiederherstellung* und *Katastrophe* markiert sind, werden globale Benachrichtigungen für Probleme und Wiederherstellungen von Auslösern mit dem Schweregrad *Katastrophe* angezeigt.
Unterdrückte Probleme anzeigen	Aktivieren Sie das Kontrollkästchen, um Benachrichtigungen für Probleme anzuzeigen, die andernfalls aufgrund von Host-Wartung unterdrückt (nicht angezeigt) würden.

Global angezeigte Meldungen

Wenn Meldungen eingehen, werden sie in einem schwebenden Bereich auf der rechten Seite angezeigt. Sie können diesen Bereich frei neu positionieren, indem Sie die Bereichsüberschrift ziehen.



Für diesen Bereich stehen mehrere Steuerelemente zur Verfügung:

- Die Schaltfläche **Snooze** schaltet den aktuell aktiven Alarmton stumm;
- Die Schaltfläche **Mute/Unmute** schaltet zwischen dem Abspielen und dem vollständigen Nichtabspielen von Alarmtönen um.

Sound in Browsern

Übersicht

Ton wird in globalen Benachrichtigungen verwendet.

Damit Töne im Zabbix Frontend abgespielt werden, müssen *Frontend-Benachrichtigungen* im Benutzerprofil auf der Registerkarte *Frontend-Benachrichtigungen* aktiviert sein, wobei alle Auslöser-Schweregrade ausgewählt sein müssen. Zusätzlich sollten Töne im globalen Benachrichtigungs-Popup-Fenster aktiviert sein.

Falls Audio aus irgendeinem Grund auf dem Gerät nicht abgespielt werden kann, bleibt die Schaltfläche im globalen Benachrichtigungs-Popup-Fenster dauerhaft im Zustand „Stumm“, begleitet von der Meldung „Cannot support notification audio

for this device“, wenn Sie den Mauszeiger über die Schaltfläche  bewegen.

Töne, einschließlich der Standard-Audioclips, werden nur im MP3-Format unterstützt.

Die Töne des Zabbix Frontend wurden erfolgreich in aktuellen Firefox- und Opera-Browsern unter Linux sowie in Chrome-, Firefox-, Microsoft-Edge- und Opera-Browsern unter Windows getestet.

Attention:
Die automatische Wiedergabe von Tönen ist in aktuellen Browserversionen möglicherweise (standardmäßig) deaktiviert. In solchen Fällen müssen Sie diese Einstellung manuell aktivieren.

Globale Suche

Es ist möglich, im Zabbix-Frontend nach Hosts, Hostgruppen, Vorlagen und Vorlagengruppen zu suchen.

Das Eingabefeld für die Suche befindet sich unter dem Zabbix-Logo im Menü. Die Suche kann durch Drücken von *Enter* oder durch Klicken auf das Suchsymbol  gestartet werden.



Wenn es einen Host gibt, dessen Name an irgendeiner Stelle die eingegebene Zeichenfolge enthält, wird ein Dropdown angezeigt, das alle solchen Hosts auflistet (wobei der passende Teil orange hervorgehoben ist). Das Dropdown listet einen Host auch dann auf, wenn der sichtbare Name dieses Hosts mit dem als Suchzeichenfolge eingegebenen technischen Namen übereinstimmt; der passende Host wird dann aufgelistet, jedoch ohne Hervorhebung.

Durchsuchbare Attribute

Hosts können nach den folgenden Eigenschaften durchsucht werden:

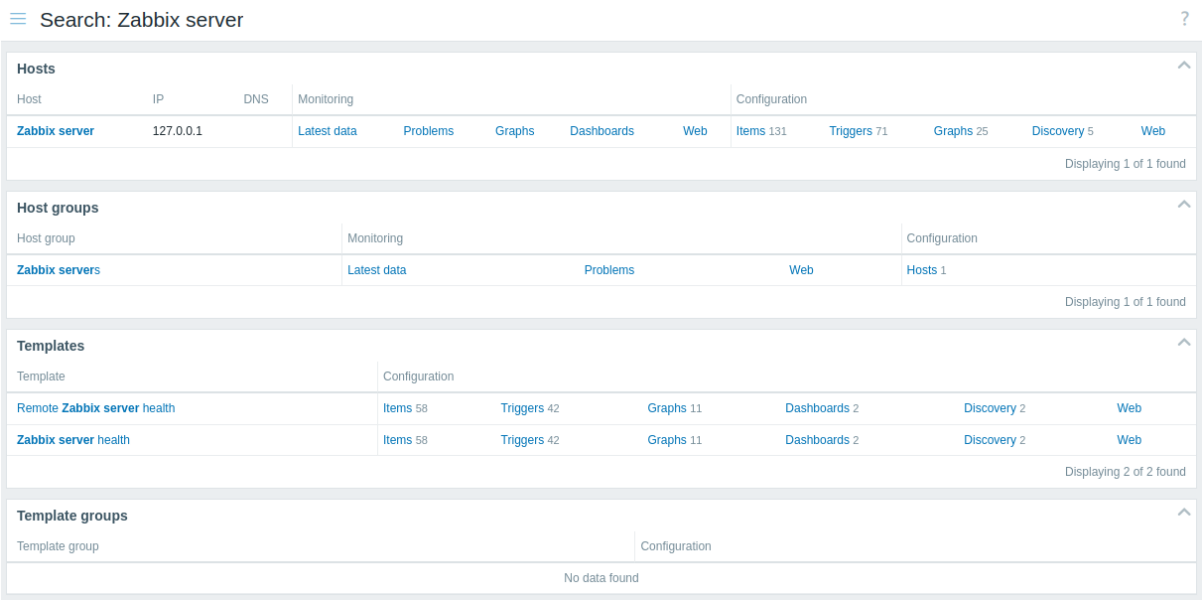
- Host-Name
- Sichtbarer Name
- IP-Adresse
- DNS-Name

Vorlagen können nach Name oder sichtbarem Namen durchsucht werden. Wenn Sie nach einem Namen suchen, der sich vom sichtbaren Namen (einer Vorlage/eines Hosts) unterscheidet, wird er in den Suchergebnissen unter dem sichtbaren Namen in Klammern angezeigt.

Host- und Vorlagengruppen können nach Namen durchsucht werden. Durch die Angabe einer übergeordneten Gruppe werden implizit alle verschachtelten Gruppen ausgewählt.

Suchergebnisse

Die Suchergebnisse bestehen aus vier separaten Blöcken für Hosts, Host-Gruppen, Vorlagen und Vorlagen-Gruppen.



Es ist möglich, jeden einzelnen Block ein- oder auszuklappen. Die Anzahl der Einträge wird am unteren Rand jedes Blocks angezeigt, zum Beispiel *13 von 13 gefundenen Einträgen werden angezeigt*. Wenn keine Einträge vorhanden sind, wird die Anzahl der Einträge nicht angezeigt. Die Gesamtzahl der innerhalb eines Blocks angezeigten Einträge ist auf 100 begrenzt.

Jeder Eintrag enthält Links zu Monitoring- und Konfigurationsdaten. Siehe die vollständige Liste der Links.

Für alle Konfigurationsdaten (wie Datenpunkte, Auslöser, Diagramme) wird die Anzahl der gefundenen Entitäten durch eine Zahl neben dem Entitätsnamen in Grau angezeigt. **Beachten Sie**, dass bei null Entitäten keine Zahl angezeigt wird.

Aktivierte Hosts werden blau angezeigt, deaktivierte Hosts rot.

Verfügbare Links

Für jeden Eintrag sind die folgenden Links verfügbar:

- Hosts
  - Überwachung
    - Neueste Daten
    - Probleme
    - Diagramme
    - Host-Dashboards
    - Webszenarien
  - Konfiguration
    - Datenpunkte
    - Auslöser
    - Diagramme
    - Discovery-Regeln
    - Webszenarien
- Host-Gruppen
  - Überwachung
    - Neueste Daten
    - Probleme
    - Webszenarien
  - Konfiguration
    - Hosts
- Vorlagen
  - Konfiguration
    - Datenpunkte
    - Auslöser
    - Diagramme
    - Vorlagen-Dashboards
    - Discovery-Regeln
    - Webszenarien
- Vorlagengruppen
  - Konfiguration
    - Vorlagen

Frontend-Wartungsmodus

Übersicht

Es ist möglich, das Zabbix Frontend vorübergehend zu deaktivieren, um den Zugriff einzuschränken. Dies ist nützlich, um die Zabbix-Datenbank vor von Benutzern initiierten Änderungen zu schützen und ihre Integrität zu bewahren.

Während sich das Zabbix Frontend im Wartungsmodus befindet, können Sie die Datenbank sicher anhalten und Wartungsaufgaben durchführen.

Benutzer von definierten IP-Adressen können während des Wartungsmodus normal mit dem Frontend interagieren.

Konfiguration

Um den Wartungsmodus zu aktivieren, öffnen Sie die Datei `maintenance.inc.php` (sie befindet sich im Verzeichnis `/conf` des HTML-Dokumentverzeichnisses von Zabbix auf dem Webserver) und entfernen Sie die Kommentarzeichen in den folgenden Zeilen:

```
// Wartungsmodus.
define('ZBX_DENY_GUI_ACCESS', 1);

// Array von IP-Adressen, die eine Verbindung zum Frontend herstellen dürfen (optional).
$ZBX_GUI_ACCESS_IP_RANGE = array('127.0.0.1');

// Auf dem Warnbildschirm angezeigte Meldung (optional).
$ZBX_GUI_ACCESS_MESSAGE = 'Wir aktualisieren die MySQL-Datenbank bis 15:00 Uhr. Bleiben Sie dran...';
```

Note:
In den meisten Fällen befindet sich die Datei `maintenance.inc.php` im Verzeichnis `/conf` des HTML-Dokumentverzeichnisses von Zabbix auf dem Webserver. Einige Betriebssysteme und Webserver verwenden jedoch möglicherweise einen anderen Speicherort.
Beispiele für den Speicherort:

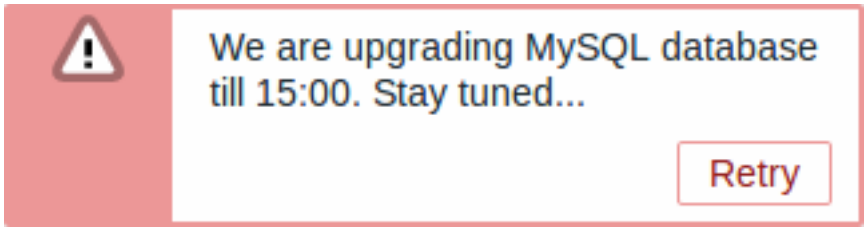
- SUSE und RedHat: `/etc/zabbix/web/maintenance.inc.php`.
- Debian-basierte Systeme: `/usr/share/zabbix/conf/`.

Siehe auch PHP-Dateien kopieren.

Parameter	Details
ZBX_DENY_GUI_ACCESS	Wenn der Parameter mit einem beliebigen Wert definiert ist, wird der Wartungsmodus aktiviert. Um den Wartungsmodus zu deaktivieren, kommentieren Sie den Parameter aus oder löschen Sie ihn.
ZBX_GUI_ACCESS_IP_RANGE	Array von IP-Adressen, die eine Verbindung zum Frontend herstellen dürfen (optional). Beispiel: `array('192.168.1.1', '192.168.1.2')`
ZBX_GUI_ACCESS_MESSAGE	Eine Nachricht, um Benutzer über die Wartungsarbeiten zu informieren (optional). Wenn der Parameter nicht definiert ist, wird die Standardnachricht *„Zabbix befindet sich im Wartungsmodus“* verwendet.

Anzeige

Benutzer sehen den folgenden Bildschirm, wenn sie versuchen, auf das Zabbix Frontend zuzugreifen, während der Wartungsmodus aktiv ist. Der Bildschirm wird alle 30 Sekunden aktualisiert, damit nach dem Ende der Wartung ohne Benutzereingriff zum normalen Zustand zurückgekehrt wird.



In *ZBX_GUI_ACCESS_IP_RANGE* definierte IP-Adressen können wie gewohnt auf das Frontend zugreifen.

Seitenparameter

Übersicht

Die meisten Seiten der Zabbix-Weboberfläche unterstützen verschiedene HTTP-GET-Parameter, die steuern, was angezeigt wird. Sie können übergeben werden, indem nach der URL Parameter=Wert-Paare angegeben werden, die durch ein Fragezeichen (?) von der URL und durch kaufmännische Und-Zeichen (&) voneinander getrennt sind.

Monitoring > Probleme

Die folgenden Parameter werden unterstützt:

Parameter	Beschreibung	Beispiel
show	Filteroption *Anzeigen*. Mögliche Werte: 1 - aktuelle Probleme; 2 - alle; 3 - im Problemzustand.	show=1
name	Filteroption *Problem*: freier Text.	name=Zabbix agent
severities	Filteroption *Schweregrad*: Array der ausgewählten Schweregrade im Format `severities[*]=*` (ersetzen Sie `*` durch den Schweregrad). Mögliche Werte: 0 - nicht klassifiziert; 1 - Information; 2 - Warnung; 3 - durchschnittlich; 4 - hoch; 5 - Katastrophe.	severities[3]=3
inventory	Filteroption *Host-Inventar*: Array von Inventarfeldern `[field],[value]`	inventory[0][field]=type&invento
evaltype	Filteroption *Tags*: Tag-Auswertungsmethode. Mögliche Werte: 0 - und/oder; 2 - oder.	evaltype=0
tags	Filteroption *Tags*: Array der definierten Tags `[tag], [operator], [value]` Mögliche Operator-Werte: 0 - enthält; 1 - gleich; 2 - enthält nicht; 3 - ist nicht gleich; 4 - existiert; 5 - existiert nicht.	tags[0][tag]=target&tags[0][ope
show_tags	Filteroption *Tags anzeigen*. Mögliche Werte: 0 - keine; 1 - eine; 2 - zwei; 3 - drei.	show_tags=3
tag_name_format	Filteroption *Tag-Name*. Mögliche Werte: 0 - vollständiger Name; 1 - gekürzter Name; 2 - keiner.	tag_name_format=1
tag_priority	Filteroption *Tag-Anzeigereihenfolge*: durch Kommas getrennte Zeichenfolge der Tag-Anzeigereihenfolge	tag_priority=customer, target

Parameter	Beschreibung	Beispiel
show_suppressed	Filteroption *Unterdrückte Probleme anzeigen*. Mögliche Werte: 1 - anzeigen; 2 - nicht anzeigen.	show_suppressed=1
acknowledgement_status	Filteroption *Bestätigungsstatus*. Mögliche Werte: 0 - alle; 1 - nicht bestätigt; 2 - bestätigt.	acknowledgement_status=0
acknowledged_by_me	Filteroption *Von mir*; nur mit `acknowledgement_status`=2 unterstützt. Mögliche Werte: 0 - deaktiviert; 1 - aktiviert.	acknowledged_by_me=1
compact_view	Filteroption *Kompakte Ansicht*. Mögliche Werte: 0 - deaktiviert; 1 - aktiviert.	compact_view=1
highlight_row	Filteroption *Ganze Zeile hervorheben*. Mögliche Werte: 0 - deaktiviert; 1 - aktiviert.	highlight_row=1
filter_name	Filtereigenschaft *Name*: freier Text	filter_name=Databases
filter_show_counter	Filtereigenschaft *Anzahl der Datensätze anzeigen*. Mögliche Werte: 0 - deaktiviert; 1 - aktiviert.	filter_show_counter=1
filter_custom_time	Filtereigenschaft *Benutzerdefinierten Zeitraum festlegen*. Mögliche Werte: 0 - deaktiviert; 1 - aktiviert.	filter_custom_time=1
sort	Zu sortierende Spalte. Mögliche Werte: clock - nach der Spalte *Zeit* sortieren; host - nach der Spalte *Host* sortieren; severity - nach der Spalte *Schweregrad* sortieren; name - nach der Spalte *Problem* sortieren.	sort=clock
sortorder	Sortierreihenfolge der Ergebnisse. Mögliche Werte: DESC - absteigend; ASC - aufsteigend.	sortorder=DESC
age_state	Filteroption *Älter als*; nur mit `show`=3 unterstützt. Mögliche Werte: 0 - Parameter age deaktivieren; 1 - Parameter age aktivieren.	age_state=1
age	Filteroption *Älter als*: Ganzzahl, Anzahl der Tage; nur mit `age_state`=1 und `show`=3 unterstützt.	age=7
groupids	Filteroption *Host-Gruppen*: Array von Host-Gruppen-IDs.	groupids[]=4
hostids	Filteroption *Hosts*: Array von Host-IDs	hostids[]=10084
triggerids	Filteroption *Auslöser*: Array von Auslöser-IDs	triggerids[]=22382

Parameter	Beschreibung	Beispiel
show_timeline	Filteroption *Zeitachse anzeigen*; nicht mit `compact_view=1` unterstützt. Mögliche Werte: 0 - nicht anzeigen; 1 - anzeigen.	show_timeline=1
details	Filteroption *Details anzeigen*. Mögliche Werte: 0 - nicht anzeigen; 1 - anzeigen.	details=1
from	Start des Datumsbereichs, kann relativ sein (z. B. now-1m); nur mit `filter_custom_time=1` unterstützt	from=now-2h
to	Ende des Datumsbereichs, kann relativ sein (z. B. now-1m); nur mit `filter_custom_time=1` unterstützt	to=now

Siehe auch: Seite *Probleme* Filteroptionen.

Kiosk-Modus

Der Kiosk-Modus auf unterstützten Frontend-Seiten kann mithilfe von URL-Parametern aktiviert werden. Beispielsweise in Dashboards:

- `/zabbix.php?action=dashboard.view&kiosk=1` - Kiosk-Modus aktivieren
- `/zabbix.php?action=dashboard.view&kiosk=0` - Normalen Modus aktivieren

Slideshow

Es ist möglich, eine Slideshow im Dashboard zu aktivieren:

- `/zabbix.php?action=dashboard.view&slideshow=on` - Slideshow aktivieren

Definitionen

Übersicht

Während viele Dinge im Frontend über das Frontend selbst konfiguriert werden können, sind einige Anpassungen derzeit nur durch Bearbeiten einer Definitionsdatei möglich.

Diese Datei ist `defines.inc.php` und befindet sich in /include des Zabbix-HTML-Dokumentverzeichnisses.

Parameter

Parameter in dieser Datei, die für Benutzer von Interesse sein könnten:

- ZBX_MIN_PERIOD

Minimale Diagrammperiode in Sekunden. Standardmäßig eine Minute.

- GRAPH_YAXIS_SIDE_DEFAULT

Standardposition der Y-Achse in einfachen Diagrammen und Standardwert für das Dropdown-Feld beim Hinzufügen von Datenpunkten zu benutzerdefinierten Diagrammen. Mögliche Werte: 0 - links, 1 - rechts.

Standard: 0

- ZBX_SESSION_NAME

Zeichenfolge, die als Name des Zabbix-Frontend-Sitzungscookies verwendet wird.

Standard: zbx_sessionid

- ZBX_DATA_CACHE_TTL

TTL-Timeout in Sekunden, das zum Ungültigmachen des Daten-Cache der Vault response verwendet wird. Setzen Sie 0, um das Caching der Vault response zu deaktivieren.

Standard: 60

- SUBFILTER_VALUES_PER_GROUP

Anzahl der Subfilter-Werte pro Gruppe (zum Beispiel im Subfilter der latest data).

Standard: 1000

- ZBX_MAX_WIDGET_LINES

Maximale Anzahl der anzuzeigenden Widget-Zeilen.

Standard: 1000

Zusätzliche Frontend-Sprachen

Übersicht Um im Zabbix-Webinterface eine andere Sprache als Englisch zu verwenden, muss das entsprechende Gebietsschema auf dem Webserver installiert sein. Zusätzlich ist die PHP-Erweiterung gettext erforderlich, damit die Übersetzungen funktionieren.

Installieren von Locales Um alle installierten Sprachen aufzulisten, führen Sie aus:

```
locale -a
```

Wenn einige benötigte Sprachen nicht aufgeführt sind, öffnen Sie die Datei `/etc/locale.gen` und kommentieren Sie die erforderlichen Locales aus. Da Zabbix die UTF-8-Kodierung verwendet, müssen Sie Locales mit UTF-8-Zeichensatz auswählen.

Führen Sie nun aus:

```
locale-gen
```

Starten Sie den Webserver neu. Beachten Sie, dass auch PHP-FPM neu gestartet werden muss, wenn es installiert ist und vom Webserver verwendet wird.

Die Locales sollten nun installiert sein. Möglicherweise müssen Sie die Zabbix Frontend-Seite im Browser mit `Ctrl` + `F5` neu laden, damit die neuen Sprachen angezeigt werden.

Zabbix installieren Wenn Sie Zabbix direkt aus dem Zabbix git repository installieren, müssen die Übersetzungsdateien manuell generiert werden. Um die Übersetzungsdateien zu generieren, führen Sie Folgendes aus:

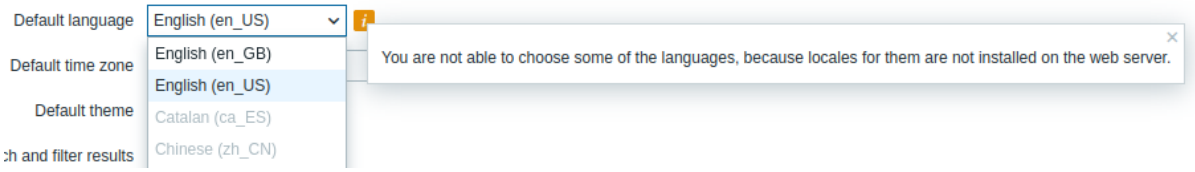
```
make gettext
locale/make_mo.sh
```

Dieser Schritt ist nicht erforderlich, wenn Sie Zabbix aus Paketen oder aus `tar.gz`-Quelldateien installieren.

Eine Sprache auswählen Es gibt mehrere Möglichkeiten, eine Sprache in der Zabbix-Weboberfläche auszuwählen:

- Bei der Installation der Weboberfläche - im Frontend Installationsassistenten. Die ausgewählte Sprache wird als Systemstandard festgelegt.
- Nach der Installation kann die systemweite Standardsprache in *Administration* > *Allgemein* > *GUI* geändert werden.
- Die Sprache für einen bestimmten Benutzer kann im Benutzerprofil geändert werden.

Wenn für eine Sprache auf dem System kein Locale installiert ist, wird diese Sprache im Zabbix-Sprachauswahlmenü ausgegraut angezeigt. Ein orangefarbenes Symbol wird neben dem Sprachauswahlmenü angezeigt, wenn mindestens ein Locale fehlt. Beim Klicken auf dieses Symbol wird die folgende Meldung angezeigt: "Sie können einige der Sprachen nicht auswählen, da die entsprechenden Locales auf dem Webserver nicht installiert sind."



Erstellen Ihres eigenen Themes

Übersicht

Standardmäßig stellt Zabbix eine Reihe vordefinierter Frontend-Themes bereit. Um ein benutzerdefiniertes Theme zu erstellen, folgen Sie der hier beschriebenen Schritt-für-Schritt-Anleitung.

Schritt 1: Erstellen Sie Ihr Design

Um Ihr eigenes Design zu erstellen:

1. Erstellen Sie eine CSS-Datei von Grund auf neu oder kopieren Sie ein vorhandenes Design (z. B. `blue-theme.css`, `dark-theme.css` oder ein anderes) aus dem Verzeichnis `assets/styles/` und passen Sie es an.
2. Speichern Sie Ihr Design (zum Beispiel `custom-theme.css`) im Verzeichnis `assets/styles/`.

Schritt 2: Aktivieren Sie Ihr Theme in der Themenliste

Damit Ihr benutzerdefiniertes Theme im Frontend verfügbar ist, müssen Sie es zur Liste der Themes hinzufügen, die von der Methode `APP::getThemes()` zurückgegeben werden. Dies erfolgt durch das Überschreiben der Methode `ZBase::getThemes()` innerhalb der Klasse `APP`:

1. Öffnen Sie die Datei `include/classes/core/APP.php` und suchen Sie die Klasse `APP`:

```
class APP extends ZBase {

}
```

2. Fügen Sie innerhalb der Klasse vor der abschließenden geschweiften Klammer } die folgende Methode hinzu:

```
public static function getThemes() {
          return array_merge(parent::getThemes(), [
              'custom-theme' => _('Custom theme')
          ]);
}
```

Dabei gilt:

- `custom-theme` - die interne Kennung Ihres Themes. Sie muss mit dem Namen Ihrer CSS-Datei ohne die Erweiterung `.css` übereinstimmen;
- `Custom theme` - der Anzeigename, der im Frontend angezeigt wird.

Um mehrere Themes hinzuzufügen, führen Sie sie im Array auf und trennen Sie die einzelnen Einträge durch ein Komma. Ein nachgestelltes Komma nach dem letzten Eintrag ist optional.

Beispiel:

```
      public static function getThemes() {
          return array_merge(parent::getThemes(), [
              'custom-theme' => _('Custom theme'),
              'anothertheme' => _('Another theme'),
              'onemoretheme' => _('One more theme')
          ]);
      }
```

Schritt 3: Diagrammfarben ändern

Um die Diagrammfarben zu ändern, muss ein Datensatz für das neue Theme zur Datenbanktabelle `graph_theme` hinzugefügt werden.

Beispiel für das Hinzufügen eines Datensatzes zu einer MySQL/MariaDB-Datenbank:

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

```
mysql> USE zabbix;
mysql> INSERT INTO graph_theme (
    graphthemeid,
    theme,
    backgroundcolor,
    graphcolor,
    gridcolor,
    maingridcolor,
    gridbordercolor,
    textcolor,
    highlightcolor,
    leftpercentilecolor,
    rightpercentilecolor,
    nonworktimecolor,
    colorpalette
```

```
)
SELECT
    MAX(graphthemeid) + 1,
    'custom-theme',
    'FFFFFF',
    'FFFFFF',
    'CCD5D9',
    'ACBBC2',
    'ACBBC2',
    '1F2C33',
    'CC745E',
    '5ECCAB',
    'CC745E',
    'EBEBEB',
    '329F7E,C2583D,346D91,B26E44,CC6C91,7A6DC2,C4AA56,7C2731,BF75B8,73A350,B04833,633A60,879CCC,7FAD6C,324
FROM graph_theme;
```

Schritt 4: Das neue Theme verwenden

Im Zabbix Frontend können Sie dieses Theme entweder als Standard-Theme festlegen oder es in einem Benutzerprofil auswählen.

Debug-Modus

Übersicht

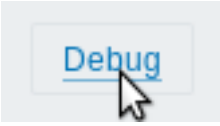
Der Debug-Modus kann verwendet werden, um Leistungsprobleme bei Frontend-Seiten zu diagnostizieren.

Konfiguration

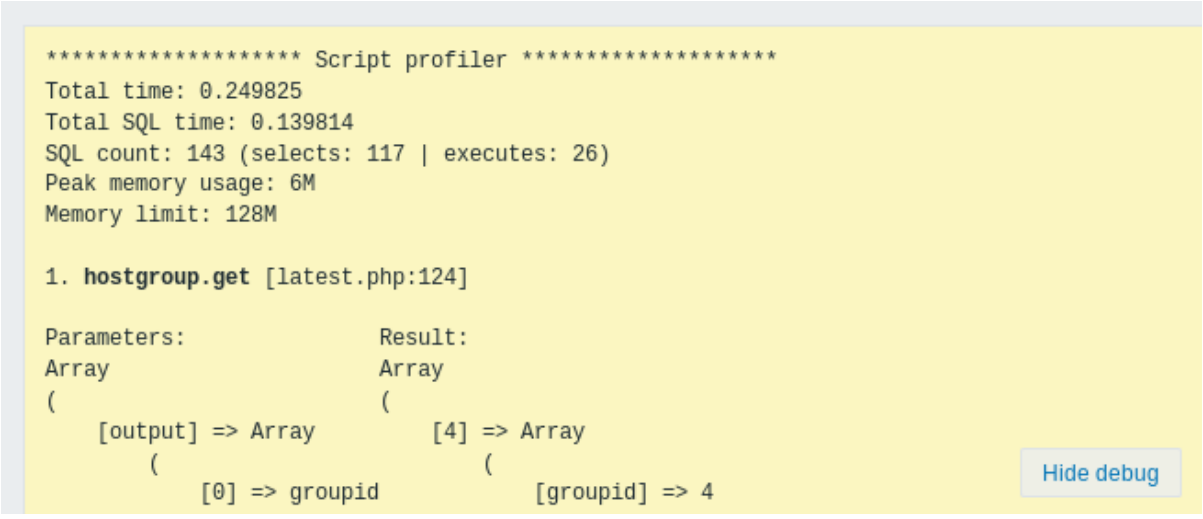
Der Debug-Modus kann für einzelne Benutzer aktiviert werden, die zu einer Benutzergruppe gehören:

- beim Konfigurieren einer Benutzergruppe;
- beim Anzeigen konfigurierter Benutzergruppen.

Wenn *Debug mode* für eine Benutzergruppe aktiviert ist, sehen deren Benutzer eine Schaltfläche *Debug* in der unteren rechten Ecke des Browserfensters:



Ein Klick auf die Schaltfläche *Debug* öffnet unterhalb des Seiteninhalts ein neues Fenster, das die SQL-Statistiken der Seite sowie eine Liste der API-Aufrufe und einzelnen SQL-Anweisungen enthält:



Bei Leistungsproblemen mit der Seite kann dieses Fenster verwendet werden, um die Ursache des Problems zu ermitteln.

Warning:
Der aktivierte *Debug mode* wirkt sich negativ auf die Leistung des Frontend aus.

Von Zabbix verwendete Cookies

Übersicht

Diese Seite enthält eine Liste der von Zabbix verwendeten Cookies.

Name	Beschreibung	Werte	Läuft ab/Max-Age	HttpOnly[1]	Secure[2]
ZBX_SESSION_NAME	Sitzungsdaten des Zabbix Frontend, gespeichert als per base64 kodiertes JSON		Sitzung (läuft ab, wenn die Browsing-Sitzung endet)	+	+ (nur wenn HTTPS auf einem Webserver aktiviert ist)
tab	Nummer des aktiven Tabs; dieses Cookie wird nur auf Seiten mit mehreren Tabs verwendet (z. B. auf der Konfigurationsseite für *Host*, *Auslöser* oder *Aktion*) und wird erstellt, wenn ein Benutzer von einem primären Tab zu einem anderen Tab navigiert (z. B. zum Tab *Tags* oder *Abhängigkeiten*). 0 wird für den primären Tab verwendet.	Beispiel: 1	Sitzung (läuft ab, wenn die Browsing-Sitzung endet)	-	-
browserwarning_ignore	Gibt an, ob eine Warnung zur Verwendung eines veralteten Browsers ignoriert werden soll.	yes	Sitzung (läuft ab, wenn die Browsing-Sitzung endet)	-	-
system-message-ok	Eine Meldung, die angezeigt wird, sobald die Seite neu geladen wird.	Klartextmeldung	Sitzung (läuft ab, wenn die Browsing-Sitzung endet) oder sobald die Seite neu geladen wird	+	-
system-message-error	Eine Fehlermeldung, die angezeigt wird, sobald die Seite neu geladen wird.	Klartextmeldung	Sitzung (läuft ab, wenn die Browsing-Sitzung endet) oder sobald die Seite neu geladen wird	+	-

Note:
Das Erzwingen des Flags 'HttpOnly' für Zabbix-Cookies durch eine Webserver- Direktive wird nicht unterstützt.

Fußnoten

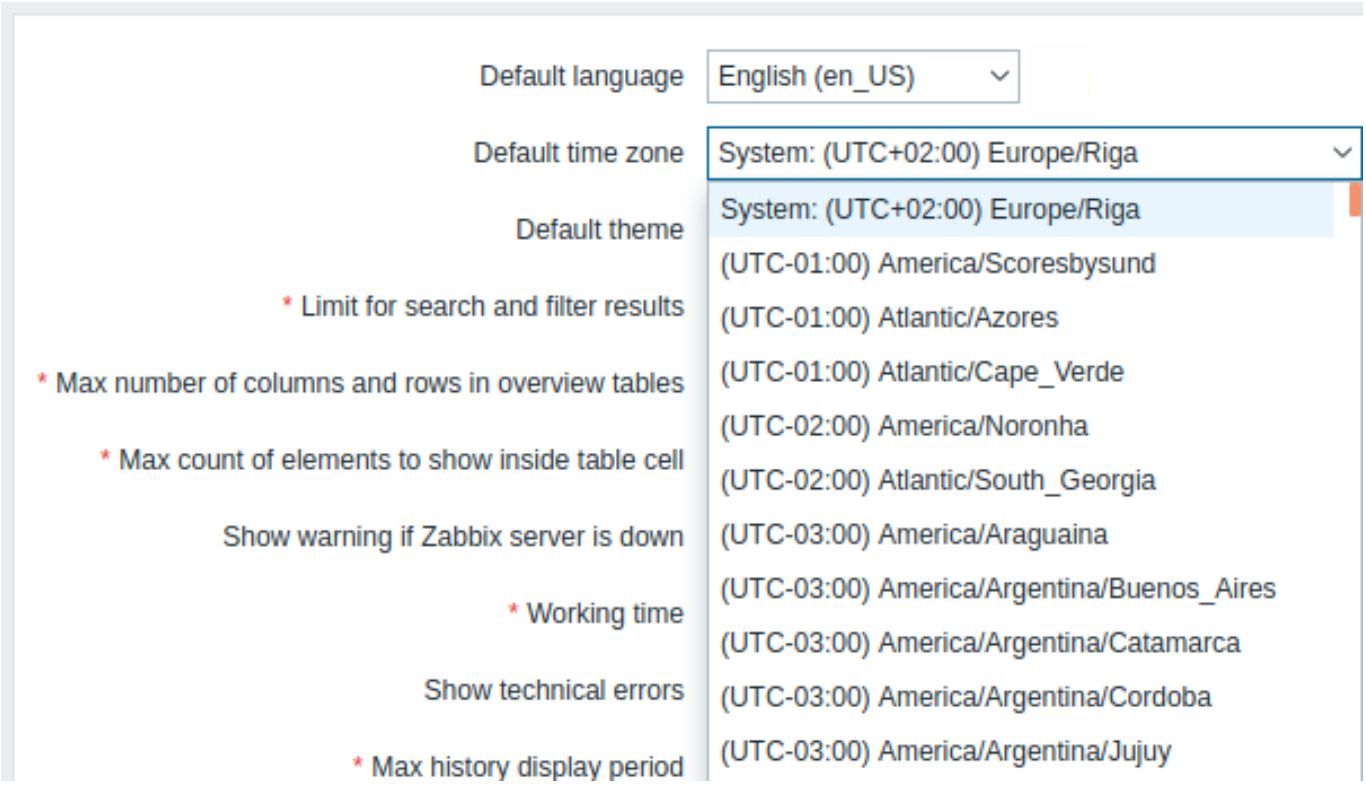
[1] Wenn `HttpOnly` auf 'true' gesetzt ist, ist das Cookie nur über das HTTP-Protokoll zugänglich. Das bedeutet, dass das Cookie nicht für Skriptsprachen wie JavaScript zugänglich ist. Diese Einstellung kann wirksam dazu beitragen, Identitätsdiebstahl durch XSS-Angriffe zu verringern (obwohl sie nicht von allen Browsern unterstützt wird).

[2] `Secure` gibt an, dass das Cookie nur über eine sichere HTTPS-Verbindung vom Client übertragen werden soll. Wenn es auf 'true' gesetzt ist, wird das Cookie nur gesetzt, wenn eine sichere Verbindung besteht.

Zeitzonen

Übersicht

Die Zeitzone des Frontends kann global im Frontend festgelegt und für einzelne Benutzer angepasst werden.



Wenn *System* ausgewählt ist, wird die Zeitzone des Webservers für das Frontend verwendet (einschließlich des Werts von `date.timezone` in `php.ini`, sofern festgelegt), während der Zabbix Server die Zeitzone des Computers verwendet, auf dem er ausgeführt wird.

Note:
Der Zabbix Server verwendet die angegebene globale Zeitzone bzw. Benutzerzeitzone nur beim Ersetzen von Makros in Benachrichtigungen (z. B. kann `{EVENT.TIME}` je nach Benutzer in eine andere Zeitzone aufgelöst werden) und für das Zeitlimit beim Senden von Benachrichtigungen (siehe Einstellung „Aktiv, wenn“ in der Benutzermedienkonfiguration).

Die Auswahl der Zeitzone wirkt sich nicht auf das Datums-/Zeitformat des Frontends aus. Stattdessen können Sie die Sprache der Benutzeroberfläche anpassen (entweder während der Installation oder unter Benutzereinstellungen. Wenn Sie *English (en_US)* auswählen, wird im Frontend auch das US-amerikanische Datums-/Zeitformat aktiviert.

Konfiguration

Die globale Zeitzone:

- Kann beim Installieren des Frontends manuell festgelegt werden.
- Kann unter *Administration* > *Allgemein* > *GUI* geändert werden.

Zeitzone auf Benutzerebene:

- Kann beim Konfigurieren/Aktualisieren eines Benutzers festgelegt werden.
- Kann von jedem Benutzer in seinem Benutzerprofil festgelegt werden.

Siehe auch: Zeitzonen angleichen bei der Verwendung von Planungsintervallen.

Passwort zurücksetzen

Übersicht Dieser Abschnitt beschreibt die Schritte zum Zurücksetzen von Benutzerpasswörtern in Zabbix.

Schritte Wenden Sie sich an Ihren Zabbix-Administrator, wenn Sie Ihr Zabbix-Passwort vergessen haben und sich nicht anmelden können.

Ein Super-Administrator kann die Passwörter aller Benutzer im Benutzerkonfigurationsformular ändern.

Wenn ein Super-Administrator sein Passwort vergessen hat und sich nicht anmelden kann, muss die folgende SQL-Abfrage ausgeführt werden, um das Standardpasswort für den Super-Administrator-Benutzer festzulegen (ersetzen Sie „Admin“ durch den entsprechenden Benutzernamen des Super-Administrators):

```
UPDATE users SET passwd = '$2a$10$ZXIvHAEP2ZM.dLXTm6uPHOMVlARXX7cqjbhM6Fn0cANzkCQBWpMrS' WHERE username =
```

Nach Ausführung dieser Abfrage wird das Benutzerpasswort auf *zabbix* gesetzt. Ändern Sie das Standardpasswort bei der ersten Anmeldung.

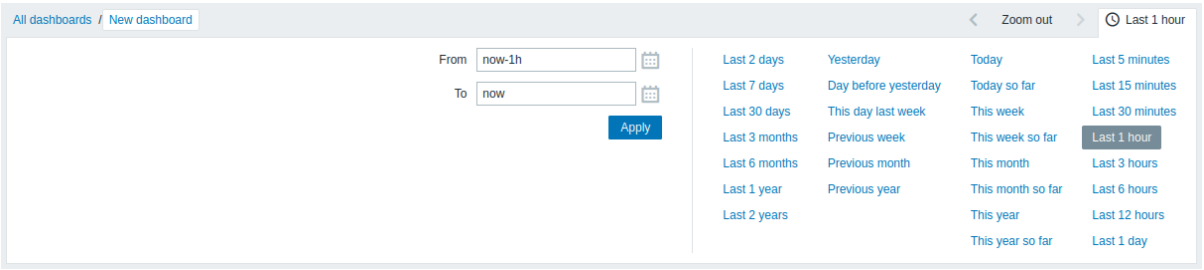
Zeitraum- und Host-Auswahlfelder

Übersicht

Diese Seite beschreibt die verfügbaren Zeitraums- und Host-Auswahlen in Dashboards. Mit diesen Auswahlen können die angezeigten Daten in allen kompatiblen Dashboard-Widgets dynamisch angepasst werden, ohne die Einstellungen einzelner Widgets zu bearbeiten.

Zeitraum-Auswahl

Mit der Zeitraum-Auswahl können Sie häufig benötigte Zeiträume mit einem Mausklick auswählen. Sie kann durch Klicken auf die Schaltfläche zur Zeitraum-Auswahl in der oberen rechten Ecke des Dashboards ein- oder ausgeblendet werden.



Optionen wie *Heute*, *Diese Woche* usw. zeigen den gesamten Zeitraum einschließlich der Stunden/Tage in der Zukunft an. Optionen wie *Heute bisher*, *Diese Woche bisher* usw. zeigen nur die bereits vergangenen Stunden an.

Nach der Auswahl eines Zeitraums kann dieser durch Klicken auf die Pfeilschaltflächen zeitlich vor- und zurückbewegt werden. Mit der Schaltfläche *Verkleinern* können Sie den Zeitraum in jede Richtung um 50 % verkleinern.

Note:
Bei Diagrammen (außer bei Diagrammen mit der *Benutzerdefiniert*-Option für den Zeitraum) können Sie den angezeigten Zeitraum auch auswählen, indem Sie mit der linken Maustaste einen Bereich im Diagramm markieren. Sobald Sie die linke Maustaste loslassen, wird das Diagramm auf den markierten Bereich vergrößert. Das Verkleinern ist auch durch einen Doppelklick im Diagramm möglich.

Die Felder *Von/Bis* zeigen den ausgewählten Zeitraum entweder in absoluter Zeitsyntax (im Format `Y-m-d H:i:s`) oder in relativer Zeitsyntax an. Ein relativer Zeitraum kann eine oder mehrere mathematische Operationen (- oder +) enthalten, zum Beispiel `now-1d` oder `now-1d-2h+5m`.

Die folgenden Abkürzungen für relative Zeitangaben werden unterstützt:

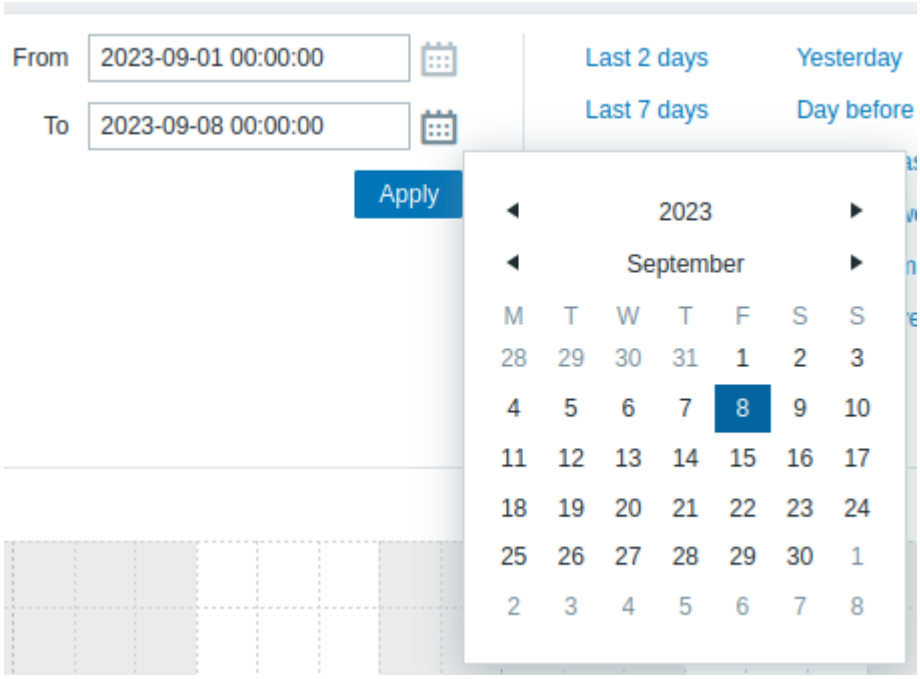
- `now`
- `s` (Sekunden)
- `m` (Minuten)
- `h` (Stunden)
- `d` (Tage)
- `w` (Wochen)
- `M` (Monate)
- `y` (Jahre)

Die Zeitraum-Auswahl unterstützt Genauigkeitsangaben (zum Beispiel `/M` in `now-1d/M`). Details zur Genauigkeit:

Genauigkeit	Von	Bis
`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

Genauigkeit	Von	Bis
`w`	Montag der Woche 00:00:00	Sonntag der Woche 23:59:59
`M`	Erster Tag des Monats 00:00:00	Letzter Tag des Monats 23:59:59
`y`	1. Januar des Jahres 00:00:00	31. Dezember des Jahres 23:59:59

Sie können einen Zeitraum auch mithilfe der Datumsauswahl auswählen. Klicken Sie zum Öffnen auf das Kalendersymbol neben den Feldern *Von/Bis*.



Note:
In der Datumsauswahl können Sie mit `Tab`, `Shift+Tab` und den Pfeiltasten der Tastatur zwischen Jahr, Monat und Datum navigieren. Drücken Sie `Enter`, um die Auswahl zu bestätigen.

Beispiele:

Von	Bis	Ausgewählter Zeitraum
`now/d`	`now/d`	00:00 - 23:59 heute
`now/d`	`now/d+1d`	00:00 heute - 23:59 morgen
`now/w`	`now/w`	Montag 00:00:00 - Sonntag 23:59:59 diese Woche
`now-1y/w`	`now-1y/w`	Die Woche von Montag 00:00:00 - Sonntag 23:59:59 vor einem Jahr

Attention:
Die Verwendung von "now/M+1M" für den Parameter *Bis* kann 31 Tage hinzufügen, was dazu führen kann, dass sich das Datum je nach Anzahl der Tage im Monat um 1-3 Tage verschiebt. Wenn dies beispielsweise im Januar verwendet wird, kann das Ergebnis der 02. März statt des erwarteten 28. Februar sein. Um dieses Problem zu vermeiden, verwenden Sie "now/M-3d+1M/M", das die Monatslänge korrekt berücksichtigt. Wenn *Von* entsprechend rückwärts konfiguriert werden soll, verwenden Sie analog "now/M+3d-1M/M".

Host-Auswahl

Mit der Host-Auswahl können Sie einen Host auswählen, dessen Daten in kompatiblen Dashboard-Widgets angezeigt werden.

Sie wird verfügbar, nachdem Sie ein Dashboard bearbeitet und gespeichert haben, wenn mindestens ein Widget mit dem Parameter *Host überschreiben* konfiguriert ist, der das Dashboard als Datenquelle verwendet.



Konfigurationsparameter: Zabbix Frontend

Übersicht Konfigurationsparameter ermöglichen die Anpassung des Verhaltens des Zabbix Frontends.

Geben Sie Parameterwerte in der Frontend-Konfigurationsdatei (`zabbix.conf.php`) an.

Diese Seite ist wie folgt strukturiert:

- Parameterübersicht (um vollständige Details zu einem Parameter anzuzeigen, klicken Sie auf seinen Namen)
- Parameterdetails

Parameter	Beschreibung
$DB['TYPE']	Datenbanktyp.
$DB['SERVER']	Hostname oder IP-Adresse des Datenbank-Hosts.
$DB['PORT']	Datenbankport.
$DB['DATABASE']	Name der Datenbank, die vom Zabbix Frontend für die Verbindung mit der konfigurierten Datenbank verwendet wird.
$DB['USER']	Benutzername, der vom Zabbix Frontend für die Verbindung mit der Datenbank verwendet wird.
$DB['PASSWORD']	Passwort, das vom Zabbix Frontend für die Verbindung mit der Datenbank verwendet wird.
$DB['SCHEMA']	Name des Datenbankschemas für PostgreSQL.
$DB['ENCRYPTION']	TLS-Verschlüsselung für Verbindungen zwischen dem Zabbix Frontend und der Datenbank aktivieren oder deaktivieren.
$DB['KEY_FILE']	Pfad zur privaten Client-Schlüsseldatei, die für die TLS-Verbindung zur Datenbank verwendet wird.
$DB['CERT_FILE']	Pfad zur Client-Zertifikatsdatei, die für die TLS-Verbindung zur Datenbank verwendet wird.
$DB['CA_FILE']	Pfad zur CA-Zertifikatsdatei, die für die TLS-Verbindung zur Datenbank verwendet wird.
$DB['VERIFY_HOST']	Überprüfung des Hostnamens des Datenbankservers im TLS-Zertifikat aktivieren oder deaktivieren.
$DB['CIPHER_LIST']	Liste der zulässigen TLS-Chiffren für die Datenbankverbindung.
$DB['VAULT']	Den Vault-Secrets-Manager angeben, der zum Speichern der Datenbankanmeldedaten verwendet wird.
$DB['VAULT_URL']	URL des Vault-Secrets-Managers, der zum Abrufen der Datenbankanmeldedaten verwendet wird.
$DB['VAULT_PREFIX']	Präfix für den Vault-Pfad oder die Abfrage, der bzw. die zum Abrufen der Datenbankanmeldedaten verwendet wird.
$DB['VAULT_DB_PATH']	Pfad zu den Datenbankanmeldedaten im Vault-Secrets-Manager.
$DB['VAULT_TOKEN']	Authentifizierungstoken für den Zugriff auf den Vault-Secrets-Manager.
$DB['VAULT_APP_ROLE_ID']	AppRole-Rollen-ID für die Authentifizierung bei HashiCorp Vault.
$DB['VAULT_APP_SECRET_ID']	AppRole-Secret-ID für die Authentifizierung bei HashiCorp Vault.
$DB['VAULT_CERT_FILE']	Pfad zur Client-Zertifikatsdatei für die Authentifizierung beim Vault-Secrets-Manager.
$DB['VAULT_KEY_FILE']	Pfad zur privaten Client-Schlüsseldatei für die Authentifizierung beim Vault-Secrets-Manager.
$DB['VAULT_CACHE']	Caching der aus dem Vault-Secrets-Manager abgerufenen Anmeldedaten aktivieren oder deaktivieren.
$ZBX_SERVER	Einen Wert angeben, um den Zabbix-Hostnamen bzw. die Zabbix-IP-Adresse zu überschreiben.
$ZBX_SERVER_PORT	Einen Wert angeben, um den Zabbix-Serverport zu überschreiben.
$ZBX_SERVER_NAME	Name des Zabbix-Servers.
$IMAGE_FORMAT_DEFAULT	Standardbildformat.
$HISTORY_PROVIDERS[]	Konfiguration der Anbieter für die Speicherung von Verlaufsdaten in Elasticsearch oder ClickHouse.
$SSO['SETTINGS']	Konfigurationseinstellungen für die SAML-Anmeldung.
$SSO['CERT_STORAGE']	Angeben, wo SAML-Zertifikate gespeichert werden.
$SSO['SP_KEY']	Pfad zur privaten Schlüsseldatei des SAML-Service-Providers (SP), wenn der private Schlüssel im Dateisystem gespeichert ist.
$SSO['SP_CERT']	Pfad zur Zertifikatsdatei des SAML-Service-Providers (SP), wenn das Zertifikat im Dateisystem gespeichert ist.
$SSO['IDP_CERT']	Pfad zur Zertifikatsdatei des SAML-Identity-Providers, wenn das Zertifikat im Dateisystem gespeichert ist.
$ZBX_FEATURE_FLAGS['banner_styling']	Unterstützung für Banner aktivieren oder deaktivieren.
$ZBX_FEATURE_FLAGS['http_auth_enabled']	HTTP-Authentifizierung für Benutzer aktivieren oder deaktivieren.
$ZBX_FEATURE_FLAGS['module_admin']	Zugriff auf Module aktivieren oder deaktivieren.
$ZBX_FEATURE_FLAGS['media_type_*']	Bestimmter vorkonfigurierter Medientypen deaktivieren.
$ZBX_SERVER_TLS['ACTIVE']	Gibt an, ob TLS für Verbindungen zum Zabbix-Server aktiviert ist.

Parameter	Beschreibung
$ZBX_SERVER_TLS['CA_FILE']	Pfad zur Zertifikatsdatei der Zertifizierungsstelle (CA), die zur Überprüfung des Serverzertifikats verwendet wird.
$ZBX_SERVER_TLS['KEY_FILE']	Pfad zur privaten Client-Schlüsseldatei, die dem Clientzertifikat entspricht.
$ZBX_SERVER_TLS['CERT_FILE']	Pfad zur Client-Zertifikatsdatei, wenn eine gegenseitige TLS-Authentifizierung erforderlich ist.
$ZBX_SERVER_TLS['CERTIFICATE_ISSUER']	Distinguished Name (DN) des Ausstellers, der mit dem Serverzertifikat abgeglichen werden soll.
$ZBX_SERVER_TLS['CERTIFICATE_SUBJECT']	Distinguished Name (DN) des Antragstellers, der mit dem Serverzertifikat abgeglichen werden soll.
$TELEMETRY_PROVIDERS[]	Konfiguration einer Telemetriedatenquelle.

Parameterdetails

$DB['TYPE']

Datenbanktyp.

Werte: `MYSQL` - MySQL, `POSTGRESQL` - PostgreSQL.

$DB['SERVER']

Hostname oder IP-Adresse der Datenbank.

Wenn `$DB['SERVER']` nicht definiert ist, wird der standardmäßige Unix-Domain-Socket verwendet. Bei MySQL führt die Einstellung von `$DB['SERVER']` auf `localhost` ebenfalls zu einer Verbindung über den standardmäßigen Unix-Socket.
 Bei PostgreSQL kann `$DB['SERVER']` mehrere durch Kommas getrennte `host:port`-Einträge enthalten. Wenn für einen Host in der durch Kommas getrennten Liste kein Port definiert ist, wird der in `$DB['PORT']` definierte Wert als Standardport verwendet. Wenn in `$DB['SERVER']` ein Socket-Pfad angegeben ist (zum Beispiel `/var/run/pgbouncer`), wird dieser Unix-Domain-Socket verwendet.

Beispiele:

```
$DB['SERVER'] = 'localhost';
$DB['SERVER'] = 127.0.0.1;
```

$DB['PORT']

Datenbankport.

Wenn nicht definiert, wird der Wert 0 verwendet.

$DB['DATABASE']

Datenbankname, den das Zabbix Frontend verwendet, um eine Verbindung zur konfigurierten Datenbank herzustellen.

Beispiel:

```
$DB['DATABASE'] = 'zabbix';
```

$DB['USER']

Benutzername, den das Zabbix-Frontend verwendet, um eine Verbindung zur Datenbank herzustellen.

$DB['PASSWORD']

Passwort, das vom Zabbix Frontend für die Verbindung zur Datenbank verwendet wird.

$DB['SCHEMA']

Name des Datenbankschemas für PostgreSQL.

$DB['ENCRYPTION']

Aktiviert oder deaktiviert die Verschlüsselung für Verbindungen zwischen dem Zabbix Frontend und der Datenbank.

Werte: `true` - aktiviert, `false` - deaktiviert.

$DB['KEY_FILE']

Pfad zur privaten Client-Schlüsseldatei, die für die TLS-Verbindung zur Datenbank verwendet wird.

Beispiel:

```
// Für MySQL:
$DB['KEY_FILE'] = '/etc/ssl/mysql/client-key.pem';
```

$DB['CERT_FILE']

Pfad zur Client-Zertifikatsdatei, die für die TLS-Verbindung zur Datenbank verwendet wird.

Beispiel:

```
// Für MySQL:
$DB['CERT_FILE'] = '/etc/ssl/mysql/client-cert.pem';
```

$DB['CA_FILE']

Pfad zur CA-Zertifikatsdatei, die für die TLS-Verbindung zur Datenbank verwendet wird.

Beispiele:

```
// Für MySQL:
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
// Für PostgreSQL:
$DB['CA_FILE'] = '/etc/ssl/pgsql/root.crt';
```

$DB['VERIFY_HOST']

Aktiviert oder deaktiviert die Überprüfung des Hostnamens des Datenbankservers im TLS-Zertifikat.

Werte: `true` - aktiviert, `false` - deaktiviert.

$DB['CIPHER_LIST']

Liste der zulässigen TLS-Chiffren für die Datenbankverbindung.

Beispiele:

```
// Für MySQL:
$DB['CIPHER_LIST'] = 'TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:TLS_AES_1
```

$DB['VAULT']

Geben Sie den Vault-Secrets-Manager an, der zum Speichern von Datenbankanmeldedaten verwendet wird.

Werte: `CyberArk` oder `HashiCorp`.

Beispiele:

```
// Für CyberArk:
$DB['VAULT'] = 'CyberArk';
// Für HashiCorp:
$DB['VAULT'] = 'HashiCorp';
```

$DB['VAULT_URL']

URL des Vault-Secrets-Managers, der zum Abrufen von Datenbankanmeldeinformationen verwendet wird.

Beispiele:

```
// Für CyberArk:
$DB['VAULT_URL'] = 'https://127.0.0.1:1858';
// Für HashiCorp:
$DB['VAULT_URL'] = 'https://localhost:8200';
```

$DB['VAULT_PREFIX']

Präfix für den Vault-Pfad oder die Abfrage zum Abrufen der Datenbankanmeldedaten.

Beispiel:

```
// Für HashiCorp:
$DB['VAULT_PREFIX'] = '/v1/secret/data/zabbix/';
```

$DB['VAULT_DB_PATH']

Pfad zu den Datenbankanmeldeinformationen im Vault-Secrets-Manager.

Beispiele:

```
// Für CyberArk:
$DB['VAULT_DB_PATH'] = 'AppID=foo&Query=Safe=bar;Object=buzz';
```

```
// Für HashiCorp:
$DB['VAULT_DB_PATH'] = 'database';
```

$DB['VAULT_TOKEN']

Authentifizierungstoken für den Zugriff auf den Vault-Secrets-Manager.

Beispiel:

```
// Für HashiCorp:
$DB['VAULT_TOKEN'] = 'hvs.CAESIIG_PILmULFYOsEyWHxkZ2mF2a8VPKNLE8eHqd4autYGGh4KHGh2cy5aeTYONFNSaUp3ZnpWbDF1
```

$DB['VAULT_APP_ROLE_ID']

Die Rollen-ID von VaultAppSecretID zur Authentifizierung bei HashiCorp Vault.

Beispiel:

```
$DB['VAULT_APP_ROLE_ID'] = '6f1567f1-538d-7d70-e8e4-5d3e59471b2e';
```

$DB['VAULT_APP_SECRET_ID']

VaultAppSecretID zur Authentifizierung bei HashiCorp Vault.

Beispiel:

```
$DB['VAULT_APP_SECRET_ID'] = '89dac8b1-f7d6-96a7-2ecf-f94d2de3d6f1';
```

$DB['VAULT_CERT_FILE']

Pfad zur Clientzertifikatsdatei, die zur Authentifizierung beim Vault-Secrets-Manager verwendet wird.

Beispiel:

```
// Für CyberArk:
$DB['VAULT_CERT_FILE'] = 'conf/certs/cyberark-cert.pem';
```

$DB['VAULT_KEY_FILE']

Pfad zur privaten Clientschlüsseldatei, die zur Authentifizierung beim Vault-Secrets-Manager verwendet wird.

Beispiel:

```
// Für CyberArk:
$DB['VAULT_KEY_FILE'] = 'conf/certs/cyberark-key.pem';
```

$DB['VAULT_CACHE']

Aktiviert oder deaktiviert das Caching von Anmeldeinformationen, die aus dem Vault-Secrets-Manager abgerufen werden.

Werte: `true` - aktiviert, `false` - deaktiviert.

$ZBX_SERVER

Geben Sie einen Wert an, um den Zabbix-Hostnamen bzw. die IP-Adresse zu überschreiben.
 Wenn `$ZBX_SERVER` ohne `$ZBX_SERVER_PORT` angegeben wird, wird der Standardport (10051) verwendet.

$ZBX_SERVER_PORT

Geben Sie einen Wert an, um den Port des Zabbix-Servers zu überschreiben.
 Dieser Parameter wird zusammen mit `$ZBX_SERVER` verwendet.

$ZBX_SERVER_NAME

Name des Zabbix-Servers.

$IMAGE_FORMAT_DEFAULT

Standardbildformat.

Standard: `IMAGE_FORMAT_PNG`
 Werte: - `IMAGE_FORMAT_PNG` - PNG-Format - `IMAGE_FORMAT_JPEG` - JPEG-Format - `IMAGE_FORMAT_GIF` - GIF-Format

$HISTORY_PROVIDERS[]

Konfiguration der Anbieter für die Speicherung von Verlaufsdaten in Elasticsearch oder ClickHouse.

Unterstützte Konfigurationsparameter: - `types` - Array der Datentypen, die im externen Speicher gespeichert werden sollen; - `provider` - Typ des Verlaufsdatenanbieters: `elasticsearch` oder `clickhouse`; - `url` - URL des Verlaufsdatenanbieters; - `db` -

Datenbankname (wird für ClickHouse verwendet); - `username` - Datenbankbenutzer (wird für ClickHouse verwendet); - `password` - Datenbankpasswort (wird für ClickHouse verwendet).

Beispiele:

```
// Für ClickHouse:
$HISTORY_PROVIDERS[] = [
    'types' => ['uint', 'dbl', 'str', 'log', 'text', 'json'],
    'provider' => 'clickhouse',
    'url' => 'http://localhost:8123',
    'db' => 'zabbix',
    'username' => 'zabbix',
    'password' => 'zabbix'
];
// Für Elasticsearch:
$HISTORY_PROVIDERS[] = [
    'types' => ['uint', 'dbl', 'str', 'log', 'text', 'json'],
    'provider' => 'elasticsearch',
    'url' => 'http://localhost:9200'
];
```

$SSO['SETTINGS']

Konfigurationseinstellungen für die SAML-Anmeldung. Das Array `$SSO['SETTINGS']` muss der von der Bibliothek *SAML PHP Toolkit* erwarteten Struktur entsprechen, die zusammen mit Zabbix (bereitgestellt) wird.

In `$SSO['SETTINGS']` können nur die folgenden Optionen angegeben werden:

- `strict`
- `baseurl`
- `compress`
- `contactPerson`
- `organization`
- `sp` (nur die in dieser Liste angegebenen Optionen)
  - `attributeConsumingService`
  - `x509certNew`
- `idp` (nur die in dieser Liste angegebenen Optionen)
  - `singleLogoutService` (nur eine Option)
    * `responseUrl`
  - `certFingerprint`
  - `certFingerprintAlgorithm`
  - `x509certMulti`
- `security` (nur die in dieser Liste angegebenen Optionen)
  - `signMetadata`
  - `wantNameId`
  - `requestedAuthnContext`
  - `requestedAuthnContextComparison`
  - `wantXMLValidation`
  - `relaxDestinationValidation`
  - `destinationStrictlyMatches`
  - `rejectUnsolicitedResponsesWithInResponseTo`
  - `signatureAlgorithm`
  - `digestAlgorithm`
  - `lowercaseUrlencoding`

Alle anderen Optionen werden aus der Datenbank übernommen und können nicht überschrieben werden. Die Option debug wird ignoriert.

Wenn sich die Zabbix-Benutzeroberfläche hinter einem Proxy oder Load Balancer befindet, kann die benutzerdefinierte Option `use_proxy_headers` verwendet werden:

- `false` (Standard) – die HTTP-Header `X-Forwarded-*` ignorieren;
- `true` – die HTTP-Header `X-Forwarded-*` verwenden, um die Basis-URL zu erstellen.

Eine vollständige Beschreibung der verfügbaren Konfigurationsoptionen finden Sie in der offiziellen Dokumentation der Bibliothek.

Beispiel:

```
$SSO['SETTINGS'] = [
    'security' => [
        'signatureAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#rsa-sha384',
        'digestAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#sha384',
        // ...
    ],
    // ...
];
```

$SSO['CERT_STORAGE']

Geben Sie an, wo SAML-Zertifikate gespeichert werden.

Standard: `file`
 Werte: - `database` – der private Schlüssel und die Zertifikate werden in der Datenbank gespeichert; - `file` – der private Schlüssel und die Zertifikate werden im Dateisystem gespeichert.

Beispiel:

```
$SSO['CERT_STORAGE'] = 'database';
```

$SSO['SP_KEY']

Pfad zur Datei mit dem privaten Schlüssel des SAML-Serviceproviders (SP), wenn der private Schlüssel im Dateisystem gespeichert ist.

Beispiel:

```
$SSO['SP_KEY'] = 'conf/certs/sp.key';
```

$SSO['SP_CERT']

Pfad zur Zertifikatsdatei des SAML-Service-Providers (SP), wenn das Zertifikat im Dateisystem gespeichert ist.

Beispiel:

```
$SSO['SP_CERT'] = 'conf/certs/sp.crt';
```

$SSO['IDP_CERT']

Pfad zur Zertifikatsdatei des SAML-Identitätsproviders, wenn das Zertifikat im Dateisystem gespeichert ist.

Beispiel:

```
$SSO['IDP_CERT'] = 'conf/certs/idp.crt';
```

$ZBX_FEATURE_FLAGS['banners_enabled']

Aktiviert oder deaktiviert die Unterstützung für Banner.

Standard: `true`
 Werte: `true` - aktiviert, `false` - deaktiviert.

$ZBX_FEATURE_FLAGS['http_auth_enabled']

Aktiviert oder deaktiviert die HTTP-Authentifizierung für Benutzer.

Standard: `true`
 Werte: `true` - aktiviert, `false` - deaktiviert.

$ZBX_FEATURE_FLAGS['modules_config_enabled']

Aktiviert oder deaktiviert den Zugriff auf Module.

Standard: `true`
 Werte: `true` - aktiviert, `false` - deaktiviert.

$ZBX_FEATURE_FLAGS['media_type_denylist']

Deaktiviert die Bearbeitung bestimmter vorkonfigurierter Medientypen.

Werte: `email`, `script`, `sms`, `webhook`, `push`.

Beispiel:

```
$ZBX_FEATURE_FLAGS['media_type_denylist'] = ['webhook','push'];
```

$ZBX_SERVER_TLS['ACTIVE']

Gibt an, ob eine TLS-Verbindung zwischen dem Zabbix Frontend und dem Zabbix Server aktiviert ist.

Standard: `false`
 Werte: `true` - aktiviert, `false` - deaktiviert.

$ZBX_SERVER_TLS['CA_FILE']

Pfad zur Zertifizierungsstellendatei (CA), die zur Überprüfung des Serverzertifikats verwendet wird.

Beispiel:

```
$ZBX_SERVER_TLS['CA_FILE'] = '/etc/zabbix/ssl/ca.crt';
```

$ZBX_SERVER_TLS['KEY_FILE']

Pfad zur privaten Client-Schlüsseldatei, die dem für die gegenseitige TLS-Authentifizierung verwendeten Clientzertifikat entspricht.

Beispiel:

```
$ZBX_SERVER_TLS['KEY_FILE'] = '/etc/zabbix/ssl/frontend.key';
```

$ZBX_SERVER_TLS['CERT_FILE']

Pfad zur Client-Zertifikatsdatei, wenn eine gegenseitige TLS-Authentifizierung erforderlich ist.

Beispiel:

```
$ZBX_SERVER_TLS['CERT_FILE'] = '/etc/zabbix/ssl/frontend.crt';
```

$ZBX_SERVER_TLS['CERTIFICATE_ISSUER']

Distinguished Name (DN) des Ausstellers, der mit dem Serverzertifikat abgeglichen werden soll.

Beispiel:

```
$ZBX_SERVER_TLS['CERTIFICATE_ISSUER']  = 'CN=MyZabbixCA';
```

$ZBX_SERVER_TLS['CERTIFICATE_SUBJECT']

Distinguished Name (DN) des Subjekts, der mit dem Serverzertifikat abgeglichen werden soll.

Beispiel:

```
$ZBX_SERVER_TLS['CERTIFICATE_SUBJECT'] = 'CN=zabbix-server.example.com';
```

$TELEMETRY_PROVIDERS[]

Konfiguration einer Telemetriedatenquelle.

Unterstützte Konfigurationsparameter für ClickHouse: - `url` – legt die HTTP- oder HTTPS-URL fest, die für die Verbindung zur Telemetriedatenquelle verwendet wird. - `db` – legt den Namen der Datenbank fest. - `username` – legt den Benutzernamen fest, der zur Anmeldung an der Datenbank verwendet wird. - `password` – legt das Passwort fest, das zur Anmeldung an der Datenbank verwendet wird. - `ssl_cert_file`, `ssl_key_file`, `ssl_key_password`, `ssl_verify_peer`, `ssl_verify_host`, `ssl_ca_location`, `ssl_ca_file` – legen diese Optionen fest, um eine sichere (TLS/SSL-)Verbindung zur Datenbank zu konfigurieren. - `vault_path` – legt den Pfad zu einem Vault-Geheimnis fest, das die Werte `username` und `password` enthält. Verwenden Sie diese Option nicht zusammen mit den Optionen `username` und `password`.

Beispiel:

```
$TELEMETRY_PROVIDERS[] = [
    'provider' => 'clickhouse',
    'url' => 'http://localhost:8123',
    'db' => 'zabbix',
    'username' => 'zabbix',
    'password' => 'zabbix'
];
```

API

Übersicht Die Zabbix API ermöglicht es Ihnen, die Konfiguration von Zabbix programmgesteuert abzurufen und zu ändern, und bietet Zugriff auf historische Daten. Sie wird häufig verwendet, um:

- Neue Anwendungen zu erstellen, die mit Zabbix arbeiten.
- Zabbix in Software von Drittanbietern zu integrieren.
- Routineaufgaben zu automatisieren.

Die Zabbix API ist eine HTTP-basierte API und wird als Teil des Frontend ausgeliefert. Sie verwendet das JSON-RPC-2.0-Protokoll, was zwei Dinge bedeutet:

- Die API besteht aus einer Reihe separater Methoden.
- Anfragen und Antworten zwischen den Clients und der API werden im JSON-Format codiert.

Weitere Informationen zum Protokoll und zu JSON finden Sie in der JSON-RPC-2.0-Spezifikation und auf der Startseite des JSON-Formats.

Weitere Informationen zur Integration von Zabbix-Funktionen in Ihre Python-Anwendungen finden Sie in der Python-Bibliothek für Zabbix.

> **Note:**
> Der Benutzerzugriff in Zabbix, einschließlich Konfiguration und historischer Daten, hängt vom Benutzertyp, der zugewiesenen Benutzerrolle und den Benutzergruppen ab.

Struktur Die API besteht aus einer Reihe von Methoden, die nominell in separate APIs gruppiert sind. Jede der Methoden führt eine bestimmte Aufgabe aus. Beispielsweise gehört die Methode `host.create` zur *host* API und wird verwendet, um neue Hosts zu erstellen. Historisch werden APIs manchmal auch als "Klassen" bezeichnet.

Die meisten APIs enthalten mindestens vier Methoden: `get`, `create`, `update` und `delete` zum Abrufen, Erstellen, Aktualisieren bzw. Löschen von Daten, aber einige APIs können einen völlig anderen Satz von Methoden bereitstellen.

Anfragen ausführen Nachdem Sie das Frontend eingerichtet haben, können Sie entfernte HTTP-Anfragen verwenden, um die API aufzurufen. Dazu müssen Sie HTTP-POST-Anfragen an die Datei `api_jsonrpc.php` senden, die sich im Frontend-Verzeichnis befindet.
 Wenn Ihr Zabbix Frontend beispielsweise unter `https://example.com/zabbix` installiert ist, kann eine HTTP-Anfrage zum Aufruf der Methode `apiinfo.version` wie folgt aussehen:

```
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}'
```

Die Anfrage muss den Header `Content-Type` mit einem der folgenden Werte enthalten: `application/json-rpc`, `application/json` oder `application/jsonrequest`.

Das Anfrageobjekt muss die folgenden Eigenschaften enthalten:

- `jsonrpc` - die vom API verwendete Version des JSON-RPC-Protokolls (die Zabbix API implementiert JSON-RPC Version 2.0);
- `method` - die aufgerufene API-Methode;
- `params` - die Parameter, die an die API-Methode übergeben werden;
- `id` - eine beliebige Kennung der Anfrage (wenn sie weggelassen wird, behandelt die API die Anfrage als eine Benachrichtigung).

Wenn die Anfrage korrekt ist, sollte die von der API zurückgegebene Antwort wie folgt aussehen:

```
{
    "jsonrpc": "2.0",
    "result": "8.0.0",
    "id": 1
}
```

Das Antwortobjekt enthält wiederum die folgenden Eigenschaften:

- `jsonrpc` - die Version des JSON-RPC-Protokolls;
- `result` - die von der Methode zurückgegebenen Daten;
- `id` - eine Kennung der entsprechenden Anfrage.

Authentifizierung Um auf Daten in Zabbix zuzugreifen, müssen Sie entweder:

- ein vorhandenes API-Token verwenden, das im Zabbix-Frontend im Abschnitt *Benutzer* > *API-Token* erstellt wurde, oder das über die Token-API erstellt wurde.
- ein Authentifizierungstoken verwenden, das mit der Methode `user.login` abgerufen wurde.

Wenn Sie beispielsweise durch die Anmeldung als normaler *Admin*-Benutzer ein neues Authentifizierungstoken erhalten möchten, würde eine JSON-Anfrage wie folgt aussehen:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"user.login","params":{"username":"Admin","password":"zabbix"},"id":1}
```

Wenn Sie die Anmeldedaten korrekt angegeben haben, sollte die von der API zurückgegebene Antwort das Benutzer-Authentifizierungstoken enthalten:

```
{
    "jsonrpc": "2.0",
    "result": "0424bd59b807674191e7d77572075f33",
    "id": 1
}
```

Autorisierungsmethoden Über den Header "Authorization"

Alle API-Anfragen erfordern eine Authentifizierung oder ein API-Token. Sie können die Zugangsdaten bereitstellen, indem Sie den Header Authorization in der Anfrage verwenden:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer 0424bd59b807674191e7d77572075f33'
```

> **Attention:**
> Wenn bei Ihnen Authentifizierungsprobleme auftreten, siehe Weiterleitung des Authorization-Headers.

Die Zabbix API akzeptiert Header ohne Beachtung der Groß-/Kleinschreibung (z. B. werden `authorization`, `Authorization` und `AUTHORIZATION` gleich behandelt).

Der Authorization-Header wird in Cross-Origin-Anfragen unterstützt (CORS).

Über das Zabbix-Cookie

Ein *„zbx_session"*-Cookie wird verwendet, um eine API-Anfrage aus der Zabbix UI zu autorisieren, die mit JavaScript ausgeführt wird (aus einem Modul oder einem benutzerdefinierten Widget).

Beispiel-Workflow Der folgende Abschnitt führt Sie ausführlicher durch mehrere Anwendungsbeispiele.

Hosts abrufen Jetzt haben Sie ein gültiges Benutzer-Authentifizierungstoken (in den folgenden Beispielen als Variable dargestellt), mit dem auf die Daten in Zabbix zugegriffen werden kann.
Beispielsweise können Sie die Methode `host.get` verwenden, um die IDs, Hostnamen und Schnittstellen aller konfigurierten Hosts abzurufen:

Anfrage:

```
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` ist eine Datei, die eine JSON-Abfrage enthält.
> Anstelle einer Datei können Sie die Abfrage im Argument `--data` übergeben.

`data.json`

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": [
            "hostid",
            "host"
        ],
        "selectInterfaces": [
```

```
            "interfaceid",
            "ip"
        ]
    },
    "id": 2
}
```

Das Antwortobjekt enthält die angeforderten Daten zu den Hosts:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10084",
            "host": "Zabbix server",
            "interfaces": [
                {
                    "interfaceid": "1",
                    "ip": "127.0.0.1"
                }
            ]
        }
    ],
    "id": 2
}
```

Note:
Aus Leistungsgründen wird immer empfohlen, die Objekt-Eigenschaften aufzulisten, die Sie abrufen möchten.
Auf diese Weise vermeiden Sie, dass alles abgerufen wird.

Erstellen eines neuen Datenpunkts Erstellen Sie nun einen neuen Datenpunkt auf dem Host "Zabbix server" unter Verwendung der Daten, die Sie aus der vorherigen `host.get`-Anfrage erhalten haben. Dies kann mit der Methode `item.create` erfolgen:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"item.create","params":{"name":"Free disk space on /home/joe/","key_":
```

Eine erfolgreiche Antwort enthält die ID des neu erstellten Datenpunkts, die verwendet werden kann, um in den folgenden Anfragen auf den Datenpunkt zu verweisen:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "24759"
        ]
    },
    "id": 3
}
```

Note:
Die Methode `item.create` sowie andere *create methods* können auch Arrays von Objekten akzeptieren und mehrere Datenpunkte mit einem einzigen API-Aufruf erstellen.

Erstellen mehrerer Auslöser Wenn *create methods* also Arrays akzeptieren, können Sie mehrere Auslöser hinzufügen, zum Beispiel diesen:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
```

```
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"trigger.create","params":[{"description":"Processor load is too high
```

Die erfolgreiche Antwort enthält die IDs der neu erstellten Auslöser:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17369",
            "17370"
        ]
    },
    "id": 4
}
```

Aktualisieren eines Datenpunkts Aktivieren Sie einen Datenpunkt, indem Sie seinen Status auf 0 setzen:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"item.update","params":{"itemid":"10092","status":0},"id":5}'
```

Die erfolgreiche Antwort enthält die ID des aktualisierten Datenpunkts:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "10092"
        ]
    },
    "id": 5
}
```

Note:
Die Methode `item.update` sowie andere *Aktualisierungsmethoden* können auch Arrays von Objekten akzeptieren und mehrere Datenpunkte mit einem einzigen API-Aufruf aktualisieren.

Aktualisieren mehrerer Auslöser Aktivieren Sie mehrere Auslöser, indem Sie ihren Status auf 0 setzen:

```
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
```

Die erfolgreiche Antwort enthält die IDs der aktualisierten Auslöser:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938",
            "13939"
        ]
    },
    "id": 6
}
```

Fehlerbehandlung Bis zum jetzigen Zeitpunkt hat alles, was Sie ausprobiert haben, einwandfrei funktioniert. Aber was würde passieren, wenn Sie einen fehlerhaften Aufruf an die API versuchen würden? Versuchen Sie, einen weiteren Host zu erstellen, indem Sie `host.create` aufrufen, dabei jedoch den obligatorischen Parameter `groups` weglassen:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer ${AUTHORIZATION_TOKEN}' \
  --header 'Content-Type: application/json-rpc' \
  --data '{"jsonrpc":"2.0","method":"host.create","params":{"host":"Linux server","interfaces":[{"type":1,
```

Die Antwort enthält dann eine Fehlermeldung:

```
{
    "jsonrpc": "2.0",
    "error": {
        "code": -32602,
        "message": "Invalid params.",
        "data": "No groups for host \"Linux server\"."
    },
    "id": 7
}
```

Wenn ein Fehler aufgetreten ist, enthält das Antwortobjekt anstelle der Eigenschaft `result` die Eigenschaft `error` mit den folgenden Daten:

- `code` - ein Fehlercode;
- `message` - eine kurze Fehlerzusammenfassung;
- `data` - eine ausführlichere Fehlermeldung.

Fehler können in verschiedenen Fällen auftreten, zum Beispiel bei der Verwendung falscher Eingabewerte, bei einem Sitzungs-Timeout oder beim Versuch, auf nicht vorhandene Objekte zuzugreifen. Ihre Anwendung sollte in der Lage sein, solche Fehler elegant zu behandeln.

API-Versionen Um die API-Versionierung zu vereinfachen, entspricht seit Zabbix 2.0.4 die Version der API der Version von Zabbix selbst. Sie können die Methode `apiinfo.version` verwenden, um die Version der API zu ermitteln, mit der Sie arbeiten. Dies kann nützlich sein, um Ihre Anwendung so anzupassen, dass versionsspezifische Funktionen verwendet werden.

Zabbix garantiert die Rückwärtskompatibilität von Funktionen innerhalb einer Hauptversion. Bei nicht rückwärtskompatiblen Änderungen zwischen Hauptversionen belässt Zabbix die alten Funktionen in der nächsten Version in der Regel als veraltet und entfernt sie erst in der darauffolgenden Version. Gelegentlich kann Zabbix Funktionen zwischen Hauptversionen auch ohne Rückwärtskompatibilität entfernen. Es ist wichtig, dass Sie sich niemals auf veraltete Funktionen verlassen und so schnell wie möglich auf neuere Alternativen migrieren.

> **Note:**
> Sie können alle an der API vorgenommenen Änderungen im API-Änderungsprotokoll nachverfolgen.

Weiterführende Literatur Nun verfügen Sie über genügend Wissen, um mit der Zabbix API zu arbeiten. Bleiben Sie jedoch nicht hier stehen. Für weiterführende Informationen empfehlen wir Ihnen, sich die Liste der verfügbaren APIs anzusehen.

Methodenreferenz

Dieser Abschnitt bietet einen Überblick über die Funktionen, die von der Zabbix-API bereitgestellt werden, und hilft Ihnen, sich in den verfügbaren Klassen und Methoden zurechtzufinden.

Überwachung Die Zabbix-API ermöglicht Ihnen den Zugriff auf den Verlauf und andere Daten, die während der Überwachung gesammelt wurden.

Dashboards

Verwalten Sie Dashboards und erstellen Sie geplante Berichte basierend auf ihnen.

Dashboard-API | Vorlagen-Dashboard API | Berichts API

Hochverfügbarkeitscluster

Rufen Sie eine Liste der Serverknoten und deren Status ab.

Hochverfügbarkeitscluster-API

Verlauf

Rufen Sie historische Werte ab, die von Zabbix-Überwachungsprozessen gesammelt wurden, zur Präsentation oder weiteren Verarbeitung.

Verlaufs-API

Trends

Rufen Sie Trendwerte ab, die vom Zabbix-Server berechnet wurden, zur Präsentation oder weiteren Verarbeitung.

Trend-API

Ereignisse

Rufen Sie Ereignisse ab, die von Triggern, der Netzwerkerkennung und anderen Zabbix-Systemen generiert wurden, für eine flexiblere Situationsverwaltung oder die Integration von Drittanbieter-Tools.

Ereignis-API

Probleme

Rufen Sie Probleme basierend auf den angegebenen Parametern ab.

Problem-API

Karten

Konfigurieren Sie Karten, um detaillierte dynamische Darstellungen Ihrer IT-Infrastruktur zu erstellen.

Karten-API

Aufgaben

Interagieren Sie mit dem Aufgabenmanager des Zabbix-Servers, erstellen Sie Aufgaben und rufen Sie Antworten ab.

Aufgaben-API

APM Die Zabbix API ermöglicht den Zugriff auf Telemetriedaten, die für das Application Performance Monitoring (APM) erfasst wurden.

Traces

Rufen Sie Traces und die darin enthaltenen Spans ab.

APM trace API | APM span API

Metriken

Rufen Sie Metrik-Datenpunkte ab.

APM metric API

Protokolle

Protokolleinträge abrufen.

APM log API

Dienste Die Zabbix-API ermöglicht Ihnen den Zugriff auf Daten, die während der Dienstüberwachung gesammelt wurden.

Service Level Agreement

Definieren Sie Service-Level-Ziele (SLO) und rufen Sie detaillierte Informationen zu Service-Level-Indikatoren (SLI) über die Service-Performance ab.

SLA API

Services

Verwalten Sie Services für die Überwachung von Service-Leveln und rufen Sie detaillierte SLA-Informationen zu jedem Service ab.

Service API

Datensammlung Mit der Zabbix-API können Sie die Konfiguration Ihres Überwachungssystems verwalten.

Hosts und Hostgruppen

Verwalten Sie Hostgruppen, Hosts und alles, was damit zusammenhängt, einschließlich Host-Schnittstellen, Host-Makros und Wartungszeiträumen.

Host API | Host group API | User macro API | Value map API | Maintenance API

Datenpunkte

Definieren Sie Datenpunkte zur Überwachung.

Datenpunkt-API

Auslöser

Konfigurieren Sie Auslöser, um sich über Probleme in Ihrem System benachrichtigen zu lassen.
Verwalten Sie Auslöserabhängigkeiten.

Trigger API

Graphen

Bearbeiten Sie Graphen oder einzelne Graphenelemente für eine bessere Darstellung der gesammelten Daten.

Graphen-API | Graphenelement-API API

Vorlagen und Vorlagengruppen

Verwalten Sie Vorlagen und verknüpfen Sie diese mit Hosts oder anderen Vorlagen.

Vorlagen-API | Vorlagengruppen-API API | Wertzuordnung-API API

Low-Level-Discovery

Konfigurieren Sie Low-Level-Discovery-Regeln sowie Datenpunkt-, Auslöser- und Graphenprototypen, um dynamische Entitäten zu überwachen.

LLD-Regel-API | Datenpunktprototyp- API | Auslöserprototyp- API | Graphenprototyp- API | Hostprototyp- API

Complex event processing

Create complex event processing rules.

Complex event processing API

Ereigniskorrelation

Erstellen Sie benutzerdefinierte Regeln zur Ereigniskorrelation.

Korrelation-API

Netzwerkerkennung

Verwalten Sie Regeln zur Netzwerkerkennung, um neue Hosts automatisch zu finden und zu überwachen. Erhalten Sie vollständigen Zugriff auf Informationen über erkannte Dienste und Hosts.

Erkennungsregel-API | Erkennungsprüfung- API | Erkannter-Host-API API | Erkannter-Dienst-API API

Export und Import

Exportieren und importieren Sie Zabbix-Konfigurationsdaten für Konfigurationssicherungen, Migrationen oder umfangreiche Konfigurationsaktualisierungen.

Konfigurations-API

Web-Überwachung

Konfigurieren Sie Webszenarien, um Ihre Webanwendungen und Dienste zu überwachen.

Webszenario-API

Alarme Die Zabbix-API ermöglicht es Ihnen, die Aktionen und Alarme Ihres Überwachungssystems zu verwalten.

Aktionen und Alarme

Definieren Sie Aktionen und Vorgänge, um Benutzer über bestimmte Ereignisse zu benachrichtigen oder automatisch Remote-Befehle auszuführen. Erhalten Sie Zugriff auf Informationen über generierte Alarme und deren Empfänger.

Aktionen-API | Alarm-API API

Medientypen

Konfigurieren Sie Medientypen und verschiedene Wege, wie Benutzer Alarme erhalten.

Medientyp-API

Skripte

Konfigurieren und führen Sie Skripte aus, um Ihnen bei Ihren täglichen Aufgaben zu helfen.

Skript-API

Benutzer Die Zabbix-API ermöglicht es Ihnen, die Benutzer Ihres Überwachungssystems zu verwalten.

Benutzer und Benutzergruppen

Fügen Sie Benutzer hinzu, die Zugriff auf Zabbix haben werden, ordnen Sie die Benutzergruppen zu und gewähren Sie Berechtigungen. Erstellen Sie Rollen für die granulare Verwaltung von Benutzerrechten.

Benutzer-API | Benutzergruppen-API API | Benutzerverzeichnis-API API | Benutzerrollen-API API

API-Token

Verwalten Sie Autorisierungs-Token.

Token-API

Authentifizierung

Ändern Sie die Konfigurationsoptionen für die Authentifizierung.

Authentifizierungs-API

Geräte

Verwalten Sie mobile Geräte zur Verwendung mit der Zabbix mobile app.

Geräte-API

Administration Mit der Zabbix-API können Sie die Administrations-Einstellungen Ihres Überwachungssystems ändern.

Allgemein

Ändern Sie bestimmte globale Konfigurationsoptionen.

Autoregistrierungs-API | Icon-Map-API | Bild-API | Einstellungen-API | Reguläre-Ausdruck-API | Modul-API | Connector-API

Audit-Log

Verfolgen Sie Konfigurationsänderungen, die von jedem Benutzer vorgenommen wurden.

Audit-Log-API

Housekeeping

Konfigurieren Sie das Housekeeping.

Housekeeping-API

Proxies und Proxy-Gruppen

Verwalten Sie die Proxies, die in Ihrer verteilten Überwachungsumgebung verwendet werden.

Proxy-API | Proxy-Gruppen-API

Makros

Verwalten Sie Makros.

Benutzer-Makro-API

Zabbix-Informationen Rufen Sie die Zabbix-API-Version oder den Zeitpunkt der letzten Aktualisierung vom Zabbix Server ab.

API info API | Server info API

Aktion

Diese Klasse ist für die Arbeit mit Aktionen vorgesehen.

Objektreferenzen:

- Aktion
- Aktionsoperation
  - Nachricht der Aktionsoperation
  - Bedingung der Aktionsoperation
- Wiederherstellungsoperation der Aktion
- Aktualisierungsoperation der Aktion
- Aktionsfilter
  - Bedingung des Aktionsfilters

Verfügbare Methoden:

- action.create - neue Aktionen erstellen
- action.delete - Aktionen löschen
- action.get - Aktionen abrufen
- action.update - Aktionen aktualisieren

Aktionsobjekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `action` API.

Aktion

Das Aktionsobjekt verfügt über die folgenden Eigenschaften.

Property	Type	Description
actionid	ID	ID der Aktion. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
esc_period	string	Standarddauer eines Operationsschritts. Muss mindestens 60 Sekunden betragen. Akzeptiert eine Anzahl von Sekunden (z. B. 60, 300), einen Wert mit einem Zeitsuffix (z. B. `30s`, `1m`, `2h`) oder ein Benutzermakro. Verhalten der Eigenschaft: - *unterstützt*, wenn `eventsource` auf „event created by a trigger“, „internal event“ oder „event created on service status update“ gesetzt ist
eventsource	integer	Typ der Ereignisse, die die Aktion verarbeitet. Eine Liste der unterstützten Ereignistypen finden Sie in der Eigenschaft `source` des Ereignisses. Verhalten der Eigenschaft: - *konstant* - *erforderlich* für Erstellungsvorgänge
name	string	Name der Aktion. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
status	integer	Gibt an, ob die Aktion aktiviert oder deaktiviert ist. Mögliche Werte: 0 - *(Standard)* aktiviert; 1 - deaktiviert.

Property	Type	Description
pause_symptoms	integer	Gibt an, ob die Eskalation pausiert werden soll, wenn es sich bei dem Ereignis um ein Symptomereignis handelt. Mögliche Werte: 0 - Eskalation für Symptomprobleme nicht pausieren; 1 - *(Standard)* Eskalation für Symptomprobleme pausieren. Verhalten der Eigenschaft: - *unterstützt*, wenn `eventsource` auf „event created by a trigger“ gesetzt ist
pause_suppressed	integer	Gibt an, ob die Eskalation während Wartungszeiträumen pausiert werden soll. Mögliche Werte: 0 - Eskalation nicht pausieren; 1 - *(Standard)* Eskalation pausieren. Verhalten der Eigenschaft: - *unterstützt*, wenn `eventsource` auf „event created by a trigger“ gesetzt ist
notify_if_canceled	integer	Gibt an, ob beim Abbrechen der Eskalation eine Benachrichtigung gesendet werden soll. Mögliche Werte: 0 - Beim Abbrechen der Eskalation keine Benachrichtigung senden; 1 - *(Standard)* Beim Abbrechen der Eskalation eine Benachrichtigung senden. Verhalten der Eigenschaft: - *unterstützt*, wenn `eventsource` auf „event created by a trigger“ gesetzt ist

Aktionsoperation

Das Aktionsoperationsobjekt definiert eine Operation, die ausgeführt wird, wenn eine Aktion ausgeführt wird. Es verfügt über die folgenden Eigenschaften.

Property	Type	Description
operationtype	integer	Typ der Operation. Mögliche Werte: 0 - Nachricht senden; 1 - globales Skript; 2 - Host hinzufügen; 3 - Host entfernen; 4 - zu einer Hostgruppe hinzufügen; 5 - aus einer Hostgruppe entfernen; 6 - Vorlage verknüpfen; 7 - Verknüpfung mit Vorlage aufheben; 8 - Host aktivieren; 9 - Host deaktivieren; 10 - Inventarmodus des Hosts festlegen; 13 - Host-Tags hinzufügen; 14 - Host-Tags entfernen. Mögliche Werte, wenn `eventsource` des Action object auf „Ereignis, das von einem Auslöser erstellt wurde“ oder „Ereignis, das bei einer Aktualisierung des Servicestatus erstellt wurde“ gesetzt ist: 0 - „Nachricht senden“; 1 - „Globales Skript“. Mögliche Werte, wenn `eventsource` des Action object auf „internes Ereignis“ gesetzt ist: 0 - „Nachricht senden“. Property behavior: - *required*
esc_period	string	Dauer eines Eskalationsschritts in Sekunden. Muss größer als 60 Sekunden sein. Akzeptiert eine Anzahl von Sekunden (z. B. 60, 300), einen Wert mit einem Zeit-Suffix (z. B. `30s`, `1m`, `2h`) oder ein Benutzermakro. Wenn der Wert auf 0 oder 0s gesetzt ist, wird der standardmäßige Eskalationszeitraum der Aktion verwendet. Standard: 0s. Property behavior: - *supported*, wenn `eventsource` des Action object auf „Ereignis, das von einem Auslöser erstellt wurde“, „internes Ereignis“ oder „Ereignis, das bei einer Aktualisierung des Servicestatus erstellt wurde“ gesetzt ist
esc_step_from	integer	Schritt, ab dem die Eskalation beginnen soll. Standard: 1. Property behavior: - *supported*, wenn `eventsource` des Action object auf „Ereignis, das von einem Auslöser erstellt wurde“, „internes Ereignis“ oder „Ereignis, das bei einer Aktualisierung des Servicestatus erstellt wurde“ gesetzt ist
esc_step_to	integer	Schritt, bei dem die Eskalation enden soll. Standard: 1. Property behavior: - *supported*, wenn `eventsource` des Action object auf „Ereignis, das von einem Auslöser erstellt wurde“, „internes Ereignis“ oder „Ereignis, das bei einer Aktualisierung des Servicestatus erstellt wurde“ gesetzt ist

Property	Type	Description
evaltype	integer	Auswertungsmethode der Operationsbedingung. Mögliche Werte: 0 - *(Standard)* Und/Oder; 1 - Und; 2 - Oder.
opcommand	object	Globales Skript, das ausgeführt werden soll. Das globale Skript muss die Eigenschaft `scriptid` definiert haben. Property behavior: - *required*, wenn `operationtype` auf „globales Skript“ gesetzt ist
opcommand_grp	array	Hostgruppen, auf denen globale Skripte ausgeführt werden sollen. Die Hostgruppen müssen die Eigenschaft `groupid` definiert haben. Property behavior: - *required*, wenn `operationtype` auf „globales Skript“ gesetzt ist und `opcommand_hst` nicht gesetzt ist
opcommand_hst	array	Host, auf denen globale Skripte ausgeführt werden sollen. Die Hosts müssen die Eigenschaft `hostid` definiert haben. Property behavior: - *required*, wenn `operationtype` auf „globales Skript“ gesetzt ist und `opcommand_grp` nicht gesetzt ist
opconditions	array	Operationsbedingungen, die für Auslöseraktionen verwendet werden. Das Operationsbedingungsobjekt wird weiter unten ausführlich beschrieben.
opgroup	array	Hostgruppen, zu denen Hosts hinzugefügt werden sollen. Die Hostgruppen müssen die Eigenschaft `groupid` definiert haben. Property behavior: - *required*, wenn `operationtype` auf „zu einer Hostgruppe hinzufügen“ oder „aus einer Hostgruppe entfernen“ gesetzt ist
opmessage	object	Objekt mit den Daten der von der Operation gesendeten Nachricht. Das Operationsnachrichtenobjekt wird weiter unten ausführlich beschrieben. Property behavior: - *required*, wenn `operationtype` auf „Nachricht senden“ gesetzt ist
opmessage_grp	array	Benutzergruppen, an die Nachrichten gesendet werden sollen. Die Benutzergruppen müssen die Eigenschaft `usrgrpid` definiert haben. Property behavior: - *required*, wenn `operationtype` auf „Nachricht senden“ gesetzt ist und `opmessage_usr` nicht gesetzt ist
opmessage_usr	array	Benutzer, an die Nachrichten gesendet werden sollen. Die Benutzer müssen die Eigenschaft `userid` definiert haben. Property behavior: - *required*, wenn `operationtype` auf „Nachricht senden“ gesetzt ist und `opmessage_grp` nicht gesetzt ist

Property	Type	Description
optemplate	array	Vorlagen, die mit den Hosts verknüpft werden sollen. Die Vorlagen müssen die Eigenschaft `templateid` definiert haben. Property behavior: - *required*, wenn `operationtype` auf „Vorlage verknüpfen“ oder „Verknüpfung mit Vorlage aufheben“ gesetzt ist
opinventory	object	Inventarmodus, auf den der Host gesetzt wird. Das Inventar muss die Eigenschaft `inventory_mode` definiert haben. Property behavior: - *required*, wenn `operationtype` auf „Inventarmodus des Hosts festlegen“ gesetzt ist
optag	array	Host-Tags, die hinzugefügt oder entfernt werden sollen. Tags müssen die Eigenschaft `tag` definiert haben. Die Eigenschaft `value` ist optional. Property behavior: - *supported*, wenn `operationtype` auf „Host-Tags hinzufügen“ oder „Host-Tags entfernen“ gesetzt ist.

Aktionsoperationsnachricht

Das Objekt für die Operationsnachricht enthält Daten über die Nachricht, die von der Operation gesendet wird. Es hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
default_msg	integer	Gibt an, ob der Standardtext und -betreff der Aktionsnachricht verwendet werden sollen. Mögliche Werte: 0 - die Daten aus der Operation verwenden; 1 - *(Standard)* die Daten aus dem Medientyp verwenden.
mediatypeid	ID	ID des Medientyps, der zum Senden der Nachricht verwendet wird. Verhalten der Eigenschaft: - *unterstützt*, wenn `operationtype` des Aktionsoperationsobjekts, Aktionswiederherstellungsoperationsobjekts oder Aktionsaktualisierungsoperationsobjekts auf „Nachricht senden“ gesetzt ist oder wenn `operationtype` des Aktionsaktualisierungsoperationsobjekts auf „alle Beteiligten benachrichtigen“ gesetzt ist
message	string	Text der Operationsnachricht. Verhalten der Eigenschaft: - *unterstützt*, wenn `default_msg` auf „die Daten aus der Operation verwenden“ gesetzt ist
subject	string	Betreff der Operationsnachricht. Verhalten der Eigenschaft: - *unterstützt*, wenn `default_msg` auf „die Daten aus der Operation verwenden“ gesetzt ist

Bedingung für Aktionsoperationen

Das Objekt für die Bedingung einer Aktionsoperation definiert eine Bedingung, die erfüllt sein muss, um die aktuelle Operation auszuführen. Es hat die folgenden Eigenschaften.

Eigenschaft	Type	Beschreibung
conditiontype	integer	Typ der Bedingung. Mögliche Werte: 14 - Ereignis bestätigt. Verhalten der Eigenschaft: - *erforderlich*
value	string	Wert, mit dem verglichen wird. Verhalten der Eigenschaft: - *erforderlich*
operator	integer	Bedingungsoperator. Mögliche Werte: 0 - *(Standard)* =

Die folgenden Operatoren und Werte werden für jeden Typ der Operationsbedingung unterstützt.

Bedingung	Name der Bedingung	Unterstützte Operatoren	Erwarteter Wert
14	Ereignis bestätigt	=	Ob das Ereignis bestätigt ist. Mögliche Werte: 0 - nicht bestätigt; 1 - bestätigt.

Wiederherstellungsoperation einer Aktion

Das Objekt für die Wiederherstellungsoperation einer Aktion definiert eine Operation, die ausgeführt wird, wenn ein Problem behoben wird. Wiederherstellungsoperationen sind **nur** für Auslöser-, interne und Service-Aktionen möglich. Es hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
operationtype	integer	Typ der Operation. Mögliche Werte, wenn `eventsource` des Aktionsobjekts auf „durch einen Auslöser erzeugtes Ereignis“ oder „bei Service-Statusaktualisierung erzeugtes Ereignis“ gesetzt ist: 0 - Nachricht senden; 1 - globales Skript; 11 - alle Beteiligten benachrichtigen. Mögliche Werte, wenn `eventsource` des Aktionsobjekts auf „internes Ereignis“ gesetzt ist: 0 - Nachricht senden; 11 - alle Beteiligten benachrichtigen. Verhalten der Eigenschaft: - *erforderlich*
opcommand	object	Auszuführendes globales Skript. Für das globale Skript muss die Eigenschaft `scriptid` definiert sein. Verhalten der Eigenschaft: - *erforderlich*, wenn `operationtype` auf „globales Skript“ gesetzt ist

Eigenschaft	Typ	Beschreibung
opcommand_grp	array	Host-Gruppen, auf denen globale Skripte ausgeführt werden sollen. Für die Host-Gruppen muss die Eigenschaft `groupid` definiert sein. Verhalten der Eigenschaft: - *erforderlich*, wenn `eventsource` des Aktionsobjekts auf „durch einen Auslöser erzeugtes Ereignis“ gesetzt ist, `operationtype` auf „globales Skript“ gesetzt ist und `opcommand_hst` nicht gesetzt ist
opcommand_hst	array	Host, auf dem globale Skripte ausgeführt werden sollen. Für die Hosts muss die Eigenschaft `hostid` definiert sein. Verhalten der Eigenschaft: - *erforderlich*, wenn `eventsource` des Aktionsobjekts auf „durch einen Auslöser erzeugtes Ereignis“ gesetzt ist, `operationtype` auf „globales Skript“ gesetzt ist und `opcommand_grp` nicht gesetzt ist
opmessage	object	Objekt mit den Daten zur von der Wiederherstellungsoperation gesendeten Nachricht. Das Operationsnachrichtenobjekt wird oben ausführlich beschrieben. Verhalten der Eigenschaft: - *erforderlich*, wenn `operationtype` auf „Nachricht senden“ gesetzt ist
opmessage_grp	array	Benutzergruppen, an die Nachrichten gesendet werden sollen. Für die Benutzergruppen muss die Eigenschaft `usrgrpid` definiert sein. Verhalten der Eigenschaft: - *erforderlich*, wenn `operationtype` auf „Nachricht senden“ gesetzt ist und `opmessage_usr` nicht gesetzt ist
opmessage_usr	array	Benutzer, an die Nachrichten gesendet werden sollen. Für die Benutzer muss die Eigenschaft `userid` definiert sein. Verhalten der Eigenschaft: - *erforderlich*, wenn `operationtype` auf „Nachricht senden“ gesetzt ist und `opmessage_grp` nicht gesetzt ist

Aktualisierungsoperation einer Aktion

Das Objekt für die Aktualisierungsoperation einer Aktion definiert eine Operation, die ausgeführt wird, wenn ein Problem aktualisiert wird (durch Kommentierung, Bestätigung, Änderung des Schweregrads oder manuelles Schließen). Aktualisierungsoperationen sind **nur** für Auslöser- und Service-Aktionen möglich. Es hat die folgenden Eigenschaften.

Eigenschaft	Type	Beschreibung
operationtype	integer	Typ der Operation. Mögliche Werte: 0 - Nachricht senden; 1 - globales Skript; 12 - alle Beteiligten benachrichtigen. Verhalten der Eigenschaft: - *erforderlich*

Eigenschaft	Type	Beschreibung
opcommand	object	Auszuführendes globales Skript. Für das globale Skript muss die Eigenschaft `scriptid` definiert sein. Verhalten der Eigenschaft: - *erforderlich*, wenn `operationtype` auf ”global script” gesetzt ist
opcommand_grp	array	Host-Gruppen, auf denen globale Skripte ausgeführt werden sollen. Für die Host-Gruppen muss die Eigenschaft `groupid` definiert sein. Verhalten der Eigenschaft: - *erforderlich*, wenn `eventsource` des Action object auf ”event created by a trigger” gesetzt ist, `operationtype` auf ”global script” gesetzt ist und `opcommand_hst` nicht gesetzt ist
opcommand_hst	array	Host, auf dem globale Skripte ausgeführt werden sollen. Für die Hosts muss die Eigenschaft `hostid` definiert sein. Verhalten der Eigenschaft: - *erforderlich*, wenn `eventsource` des Action object auf ”event created by a trigger” gesetzt ist, `operationtype` auf ”global script” gesetzt ist und `opcommand_grp` nicht gesetzt ist
opmessage	object	Objekt mit den Daten zur von der Aktualisierungsoperation gesendeten Nachricht. Das Objekt der Operationsnachricht wird oben ausführlich beschrieben.
opmessage_grp	array	Benutzergruppen, an die Nachrichten gesendet werden sollen. Für die Benutzergruppen muss die Eigenschaft `usrgrpid` definiert sein. Verhalten der Eigenschaft: - *erforderlich*, wenn `operationtype` auf ”send message” gesetzt ist und `opmessage_usr` nicht gesetzt ist
opmessage_usr	array	Benutzer, an die Nachrichten gesendet werden sollen. Für die Benutzer muss die Eigenschaft `userid` definiert sein. Verhalten der Eigenschaft: - *erforderlich*, wenn `operationtype` auf ”send message” gesetzt ist und `opmessage_grp` nicht gesetzt ist

Aktionsfilter

Das Aktionsfilter-Objekt definiert eine Menge von Bedingungen, die erfüllt sein müssen, um die konfigurierten Aktionsoperationen auszuführen. Es hat die folgenden Eigenschaften.

Property	Type	Description
conditions	array	Menge von Filterbedingungen, die zum Filtern der Ergebnisse verwendet werden. Die Bedingungen werden in der Reihenfolge ihrer Platzierung in der Formel sortiert. Property behavior: - *erforderlich*

Property	Type	Description
evaltype	integer	Auswertungsmethode der Filterbedingungen. Mögliche Werte: 0 - Und/Oder; 1 - Und; 2 - Oder; 3 - Benutzerdefinierter Ausdruck. Property behavior: - *erforderlich*
eval_formula	string	Generierter Ausdruck, der zur Auswertung der Filterbedingungen verwendet wird. Der Ausdruck enthält IDs, die über `formulaid` auf bestimmte Filterbedingungen verweisen. Der Wert von `eval_formula` entspricht dem Wert von `formula` bei Filtern mit einem benutzerdefinierten Ausdruck. Property behavior: - *schreibgeschützt*
formula	string	Benutzerdefinierter Ausdruck, der zur Auswertung der Bedingungen von Filtern mit einem benutzerdefinierten Ausdruck verwendet wird. Der Ausdruck muss IDs enthalten, die über `formulaid` auf bestimmte Filterbedingungen verweisen. Die im Ausdruck verwendeten IDs müssen exakt mit den in den Filterbedingungen definierten IDs übereinstimmen: Keine Bedingung darf ungenutzt bleiben oder ausgelassen werden. Property behavior: - *erforderlich*, wenn `evaltype` auf „custom expression“ gesetzt ist

Bedingung des Aktionsfilters

Das Objekt für die Bedingung des Aktionsfilters definiert eine bestimmte Bedingung, die vor der Ausführung der Aktionsvorgänge geprüft werden muss.

Eigenschaft	Typ	Beschreibung
conditiontype	integer	Typ der Bedingung. Mögliche Werte, wenn `eventsource` des Action-Objekts auf „event created by a trigger“ gesetzt ist: 0 - Hostgruppe; 1 - Host; 2 - Auslöser; 3 - Ereignisname; 4 - Schweregrad des Auslösers; 6 - Zeitraum; 13 - Host-Vorlage; 16 - Problem ist unterdrückt; 25 - Ereignis-Tag; 26 - Wert des Ereignis-Tags. Mögliche Werte, wenn `eventsource` des Action-Objekts auf „event created by a discovery rule“ gesetzt ist: 7 - Host-IP; 8 - Typ des erkannten Dienstes; 9 - Port des erkannten Dienstes; 10 - Erkennungsstatus; 11 - Dauer der Betriebs- oder Ausfallzeit; 12 - empfangener Wert; 18 - Erkennungsregel; 19 - Erkennungsprüfung; 20 - Proxy; 21 - Erkennungsobjekt; 29 - Proxygruppe. Mögliche Werte, wenn `eventsource` des Action-Objekts auf „event created by active agent autoregistration“ gesetzt ist: 20 - Proxy; 22 - Hostname; 24 - Host-Metadaten; 29 - Proxygruppe. Mögliche Werte, wenn `eventsource` des Action-Objekts auf „internal event“ gesetzt ist: 0 - Hostgruppe; 1 - Host; 13 - Host-Vorlage; 23 - Ereignistyp; 25 - Ereignis-Tag; 26 - Wert des Ereignis-Tags. Mögliche Werte, wenn `eventsource` des Action-Objekts auf „event created on service status update“ gesetzt ist: 25 - Ereignis-Tag; 26 - Wert des Ereignis-Tags; 27 - Dienst; 28 - Dienstname. Verhalten der Eigenschaft: - *erforderlich*
value	string	Wert, mit dem verglichen werden soll. Verhalten der Eigenschaft: - *erforderlich*

Eigenschaft	Typ	Beschreibung
value2	string	Sekundärer Wert, mit dem verglichen werden soll. Verhalten der Eigenschaft: - *erforderlich*, wenn `eventsource` des Action-Objekts auf „event created by a trigger“ gesetzt ist, `conditiontype` auf einen möglichen Wert für Auslöseraktionen gesetzt ist und der Bedingungstyp (siehe unten) „26“ ist - *erforderlich*, wenn `eventsource` des Action-Objekts auf „internal event“ gesetzt ist, `conditiontype` auf einen möglichen Wert für interne Aktionen gesetzt ist und der Bedingungstyp (siehe unten) „26“ ist - *erforderlich*, wenn `eventsource` des Action-Objekts auf „event created on service status update“ gesetzt ist, `conditiontype` auf einen möglichen Wert für Dienstaktionen gesetzt ist und der Bedingungstyp (siehe unten) „26“ ist
formulaid	string	Beliebige eindeutige ID, die verwendet wird, um in einem benutzerdefinierten Ausdruck auf die Bedingung zu verweisen. Darf nur Großbuchstaben enthalten. Die ID muss vom Benutzer beim Ändern von Filterbedingungen definiert werden, wird jedoch bei einer späteren Abfrage neu generiert.
operator	integer	Bedingungs-operator. Mögliche Werte: 0 - *(Standard)* ist gleich; 1 - ist nicht gleich; 2 - enthält; 3 - enthält nicht; 4 - ist enthalten in; 5 - ist größer oder gleich; 6 - ist kleiner oder gleich; 7 - ist nicht enthalten in; 8 - entspricht; 9 - entspricht nicht; 10 - Ja; 11 - Nein.

Note:
Weitere Informationen zur Verwendung von Filtern mit verschiedenen Ausdruckstypen finden Sie in den Beispielen auf den Seiten der Methoden action.get und action.create.

Die folgenden Operatoren und Werte werden für jeden Bedingungstyp unterstützt.

Bedingung	Bedingungsname	Unterstützte Operatoren	Erwarteter Wert
0	Hostgruppe	ist gleich, ist nicht gleich	ID der Hostgruppe.
1	Host	ist gleich, ist nicht gleich	Host-ID.
2	Auslöser	ist gleich, ist nicht gleich	ID des Auslösers.
3	Ereignisname	enthält, enthält nicht	Ereignisname.
4	Schweregrad des Auslösers	ist gleich, ist nicht gleich, ist größer oder gleich, ist kleiner oder gleich	Schweregrad des Auslösers. Eine Liste der unterstützten Schweregrade finden Sie in der Eigenschaft `severity` des Auslösers.
5	Wert des Auslösers	ist gleich	Wert des Auslösers. Eine Liste der unterstützten Werte finden Sie in der Eigenschaft `value` des Auslösers.

Bedingung	Bedingungsname	Unterstützte Operatoren	Erwarteter Wert
6	Zeitraum	ist enthalten in, ist nicht enthalten in	Zeitpunkt, zu dem das Ereignis ausgelöst wurde, als Zeitraum.
7	Host-IP	ist gleich, ist nicht gleich	Ein oder mehrere zu prüfende IP-Bereiche, durch Kommas getrennt. Weitere Informationen zu unterstützten Formaten von IP-Bereichen finden Sie im Abschnitt Konfiguration der Netzwerkerkennung.
8	Typ des erkannten Dienstes	ist gleich, ist nicht gleich	Typ des erkannten Dienstes. Der Diensttyp entspricht dem Typ der Erkennungsprüfung, mit der der Dienst erkannt wurde. Eine Liste der unterstützten Typen finden Sie in der Eigenschaft `type` der Erkennungsprüfung.
9	Port des erkannten Dienstes	ist gleich, ist nicht gleich	Ein oder mehrere Portbereiche, durch Kommas getrennt.
10	Erkennungsstatus	ist gleich	Status eines erkannten Objekts. Mögliche Werte: 0 - Host oder Dienst verfügbar; 1 - Host oder Dienst nicht verfügbar; 2 - Host oder Dienst erkannt; 3 - Host oder Dienst verloren.
11	Dauer der Betriebs- oder Ausfallzeit	ist größer oder gleich, ist kleiner oder gleich	Zeit in Sekunden, die angibt, wie lange sich das erkannte Objekt im aktuellen Status befindet.
12	Empfangene Werte	ist gleich, ist nicht gleich, ist größer oder gleich, ist kleiner oder gleich, enthält, enthält nicht	Wert, der bei der Durchführung einer Erkennungsprüfung mit einem Zabbix Agent, SNMPv1, SNMPv2 oder SNMPv3 zurückgegeben wird.
13	Host-Vorlage	ist gleich, ist nicht gleich	ID der verknüpften Vorlage.
16	Problem ist unterdrückt	Ja, Nein	Kein Wert erforderlich: Der Operator „Ja“ bedeutet, dass das Problem unterdrückt sein muss, „Nein“ bedeutet, dass es nicht unterdrückt sein darf.
18	Erkennungsregel	ist gleich, ist nicht gleich	ID der Erkennungsregel.
19	Erkennungsprüfung	ist gleich, ist nicht gleich	ID der Erkennungsprüfung.
20	Proxy	ist gleich, ist nicht gleich	ID des Proxy.
21	Erkennungsobjekt	ist gleich	Typ des Objekts, das das Erkennungsereignis ausgelöst hat. Mögliche Werte: 1 - erkannter Host; 2 - erkannter Dienst.
22	Hostname	enthält, enthält nicht, entspricht, entspricht nicht	Hostname. Die Verwendung eines regulären Ausdrucks wird für die Operatoren *entspricht* und *entspricht nicht* in Bedingungen der automatischen Registrierung unterstützt.
23	Ereignistyp	ist gleich	Bestimmtes internes Ereignis. Mögliche Werte: 0 - Datenpunkt im Status „nicht unterstützt“; 1 - Datenpunkt im Status „normal“; 2 - LLD-Regel im Status „nicht unterstützt“; 3 - LLD-Regel im Status „normal“; 4 - Auslöser im Status „unbekannt“; 5 - Auslöser im Status „normal“.

Bedingung	Bedingungsname	Unterstützte Operatoren	Erwarteter Wert
24	Host-Metadaten	enthält, enthält nicht, entspricht, entspricht nicht	Metadaten des automatisch registrierten Hosts. Die Verwendung eines regulären Ausdrucks wird für die Operatoren *entspricht* und *entspricht nicht* unterstützt.
25	Tag	ist gleich, ist nicht gleich, enthält, enthält nicht	Ereignis-Tag.
26	Tag-Wert	ist gleich, ist nicht gleich, enthält, enthält nicht	Wert des Ereignis-Tags.
27	Dienst	ist gleich, ist nicht gleich	Dienst-ID.
28	Dienstname	ist gleich, ist nicht gleich	Dienstname.
29	Proxygruppe	ist gleich, ist nicht gleich	ID der Proxygruppe.

action.create

Beschreibung

`object action.create(object/array actions)`

Diese Methode ermöglicht es Ihnen, neue Aktionen zu erstellen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellende Aktionen.

Zusätzlich zu den Standard-Aktionseigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
filter	object	Objekt für den Aktionsfilter der Aktion.
operations	array	Zu erstellende Aktionsoperationen für die Aktion.
recovery_operations	array	Zu erstellende Aktions-Wiederherstellungsoperationen für die Aktion.
update_operations	array	Zu erstellende Aktions-Aktualisierungsoperationen für die Aktion.

Rückgabe-Werte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Aktionen unter der Eigenschaft `actionids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Aktionen.

Beispiele

Eine Auslöser-Aktion erstellen

Erstellen Sie eine Auslöser-Aktion, die beginnt, sobald ein Auslöser (mit dem Wort „memory“ in seinem Namen) von Host „10084“ in den Status PROBLEM wechselt. Die Aktion wird 4 konfigurierte Operationen haben. Die erste und sofortige Operation sendet eine Nachricht an alle Benutzer der Benutzergruppe „7“ über den Medientyp „1“. Wenn das Ereignis nicht innerhalb von 30 Minuten gelöst wird, wird die zweite Operation das Skript „5“ (Skript mit dem Geltungsbereich „Action operation“) auf allen Hosts in Gruppe „2“ ausführen. Wenn das Ereignis gelöst wird, benachrichtigt eine Wiederherstellungsoperation alle Benutzer, die Nachrichten bezüglich des Problems erhalten haben. Wenn das Ereignis aktualisiert wird, benachrichtigt eine Bestätigungs-/Aktualisierungsoperation (mit einem benutzerdefinierten Betreff und einer benutzerdefinierten Nachricht) alle Benutzer, die Nachrichten bezüglich des Problems erhalten haben.

Anfrage:

```
{
    "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"
                }
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "17"
        ]
    },
    "id": 1
}
```

Eine Discovery-Aktion erstellen

Erstellen Sie eine Discovery-Aktion, die die Vorlage „10001“ mit erkannten Hosts verknüpft.

Anfrage:

```
{
    "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"
                    }
                ]
```

```
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "18"
        ]
    },
    "id": 1
}
```

Verwenden eines benutzerdefinierten Ausdrucksfilters

Erstellen Sie eine Auslöser-Aktion, die einen benutzerdefinierten Ausdruck – „A and (B or C)“ – zur Auswertung von Aktionsbedingungen verwendet. Sobald ein Auslöser mit einem Schweregrad höher oder gleich „Warning“ von Host „10084“ oder Host „10106“ in den Status PROBLEM wechselt, sendet die Aktion eine Nachricht über den Medientyp „1“ an alle Benutzer der Benutzergruppe „7“. Die Formel-IDs „A“, „B“ und „C“ wurden willkürlich gewählt.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "18"
        ]
    },
    "id": 1
}
```

Agent-Autoregistrierungsregel erstellen

Erstellen Sie eine Autoregistrierungsaktion, die einen Host zur Host-Gruppe „2“ hinzufügt, wenn der Host-Name „SRV“ enthält oder die Metadaten „AlmaLinux“ enthalten.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "action.create",
    "params": {
        "name": "Register Linux servers",
        "eventsource": "2",
        "filter": {
            "evaltype": "2",
            "conditions": [
                {
                    "conditiontype": "22",
                    "operator": "2",
                    "value": "SRV"
                },
                {
                    "conditiontype": "24",
                    "operator": "2",
                    "value": "AlmaLinux"
                }
            ]
        },
        "operations": [
            {
                "operationtype": "4",
                "opgroup": [
                    {
                        "groupid": "2"
                    }
                ]
            }
        ]
    },
    "id": 1
```

```
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            19
        ]
    },
    "id": 1
}
```

Agent-Autoregistrierungsregel mit Host-Tags erstellen

Erstellen Sie eine Autoregistrierungsaktion, die einen Host zur Host-Gruppe „2“ hinzufügt und zwei Host-Tags hinzufügt.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            20
        ]
    },
    "id": 1
}
```

Siehe auch

- Aktionsfilter
- Aktionsoperation
- Skript

Quelle

CAction::create() in *ui/include/classes/api/services/CAction.php*.

action.delete

Beschreibung

`object action.delete(array actionIds)`

Mit dieser Methode können Sie Auslöser löschen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.
 Der Benutzer kann keinen Auslöser löschen, der in seinen Filterbedingungen einen nicht zugänglichen Proxy oder eine nicht zugängliche Proxy-Gruppe enthält. Weitere Informationen finden Sie unter Konfiguration von Benutzergruppen (*Proxy-Zugriffsliste*).

Parameter

`(array)` IDs der zu löschenden Aktionen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Aktionen unter der Eigenschaft `actionids` enthält.

Beispiele

Mehrere Aktionen löschen

Zwei Aktionen löschen.

Request:

```
{
"jsonrpc": "2.0",
"method": "action.delete",
"params": [
"17",
"18"
],
"id": 1
}
```

Antwort:

```
{
"jsonrpc": "2.0",
"result": {
"actionids": [
"17",
"18"
]
},
"id": 1
}
```

Quelle

CAction::delete() in *ui/include/classes/api/services/CAction.php*.

action.get

Beschreibung

`integer/array action.get(object parameters)`

Die Methode ermöglicht es Ihnen, Aktionen anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigungen zum Aufruf der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
actionids	ID/array	Gibt nur Aktionen mit den angegebenen IDs zurück.
groupids	ID/array	Gibt nur Aktionen zurück, die die angegebenen Host-Gruppen in Aktionsbedingungen verwenden.
hostids	ID/array	Gibt nur Aktionen zurück, die die angegebenen Hosts in Aktionsbedingungen verwenden.
triggerids	ID/array	Gibt nur Aktionen zurück, die die angegebenen Auslöser in Aktionsbedingungen verwenden.
mediatypeids	ID/array	Gibt nur Aktionen zurück, die die angegebenen Medientypen zum Senden von Nachrichten verwenden.
usrgrpids	ID/array	Gibt nur Aktionen zurück, die so konfiguriert sind, dass Nachrichten an die angegebenen Benutzergruppen gesendet werden.
userids	ID/array	Gibt nur Aktionen zurück, die so konfiguriert sind, dass Nachrichten an die angegebenen Benutzer gesendet werden.
scriptids	ID/array	Gibt nur Aktionen zurück, die so konfiguriert sind, dass die angegebenen Skripte ausgeführt werden.
selectFilter	query	Gibt eine Eigenschaft `filter` mit dem Filter für Aktionsbedingungen zurück.
selectOperations	query	Gibt eine Eigenschaft `operations` mit Aktionsoperationen zurück.
selectRecoveryOperations	query	Gibt eine Eigenschaft `recovery_operations` mit Wiederherstellungsoperationen der Aktion zurück.
selectUpdateOperations	query	Gibt eine Eigenschaft `update_operations` mit Aktualisierungsoperationen der Aktion zurück.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `actionid`, `name`, `status`.
countOutput	boolean	Diese Parameter werden im Referenzkommentar beschrieben.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Kann die folgenden Dinge zurück geben:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Auslöser-Aktionen abrufen

Rufen Sie alle konfigurierten Auslöser-Aktionen zusammen mit Aktionsbedingungen und Operationen ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "action.get",
    "params": {
        "output": "extend",
        "selectOperations": "extend",
        "selectRecoveryOperations": "extend",
        "selectUpdateOperations": "extend",
        "selectFilter": "extend",
        "filter": {
            "eventsource": 0
        }
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "actionid": "3",
            "name": "Probleme an Zabbix-Administratoren melden",
            "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": "Auslöser: {TRIGGER.NAME}\r\nAuslöserstatus: {TRIGGER.STATUS}\r\nAuslös
                        "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": "Aktualisiert: {TRIGGER.NAME}",
                        "message": "{USER.FULLNAME} hat das Problem am {EVENT.UPDATE.DATE} um {EVENT.UPDAT
                        "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
}
```


Discovery-Aktionen abrufen

Rufen Sie alle konfigurierten Discovery-Aktionen zusammen mit Aktionsbedingungen und Operationen ab. Der Filter verwendet

den Auswertungstyp „and“, daher ist die Eigenschaft `formula` leer und `eval_formula` wird automatisch generiert.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "action.get",
    "params": {
        "output": "extend",
        "selectOperations": "extend",
        "selectFilter": "extend",
        "filter": {
            "eventsource": 1
        }
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```


Siehe auch

- Aktionsfilter
- Aktionsoperation

Quelle

CAction::get() in *ui/include/classes/api/services/CAction.php*.

Updateaktion

Beschreibung

`object action.update(object/array actions)`

Diese Methode ermöglicht es Ihnen, vorhandene Aktionen zu aktualisieren.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu aktualisierende Aktionseigenschaften.

Die Eigenschaft `actionid` muss für jede Aktion definiert werden, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Zusätzlich zu den Standard-Aktionseigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
filter	object	Objekt Aktionsfilter, das den aktuellen Filter ersetzt.
operations	array	Aktionsoperationen, die bestehende Operationen ersetzen.

Parameter	Type	Beschreibung
recovery_operations	array	Aktions-Wiederherstellungsoperationen, die bestehende Wiederherstellungsoperationen ersetzen. Parameterverhalten: - *unterstützt*, wenn `eventsource` des Aktionsobjekts auf „durch einen Auslöser erstelltes Ereignis“, „internes Ereignis“ oder „bei Aktualisierung des Servicestatus erstelltes Ereignis“ gesetzt ist
update_operations	array	Aktions-Aktualisierungsoperationen, die bestehende Aktualisierungsoperationen ersetzen. Parameterverhalten: - *unterstützt*, wenn `eventsource` des Aktionsobjekts auf „durch einen Auslöser erstelltes Ereignis“ oder „bei Aktualisierung des Servicestatus erstelltes Ereignis“ gesetzt ist

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Aktionen unter der Eigenschaft `actionids` enthält.

Beispiele

Aktion deaktivieren

Eine Aktion deaktivieren, d. h. ihren Status auf „1“ setzen.

Anfrage:

```
{
"jsonrpc": "2.0",
"method": "action.update",
"params": {
"actionid": "2",
"status": "1"
},
"id": 1
}
```

Antwort:

```
{
"jsonrpc": "2.0",
"result": {
"actionids": [
"2"
]
},
"id": 1
}
```

Siehe auch

- Aktionsfilter
- Aktionsoperation

Quelle

CAction::update() in *ui/include/classes/api/services/CAction.php*.

Alarm

Diese Klasse ist für die Arbeit mit Alarmen konzipiert.

Objektreferenzen:

- Alert

Verfügbare Methoden:

- alert.get - Alarme abrufen

Alarmobjekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `alert` API.

Warnung

> **Note:**
> Warnungen werden vom Zabbix Server erstellt und können nicht über die API geändert werden.

Das Warnungsobjekt enthält Informationen darüber, ob bestimmte Aktionsoperationen erfolgreich ausgeführt wurden. Es hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
alertid	ID	ID der Warnung.
actionid	ID	ID der Aktion, die die Warnung erzeugt hat.
alerttype	integer	Warnungstyp. Mögliche Werte: 0 - Nachricht; 1 - Remote-Befehl.
clock	timestamp	Zeitpunkt, zu dem die Warnung erzeugt wurde.
error	string	Fehlertext, falls beim Senden einer Nachricht oder beim Ausführen eines Befehls Probleme auftreten.
esc_step	integer	Eskalationsschritt der Aktion, während dessen die Warnung erzeugt wurde.
eventid	ID	ID des Ereignisses, das die Aktion ausgelöst hat.
mediatypeid	ID	ID des Medientyps, der zum Senden der Nachricht verwendet wurde.
message	text	Nachrichtentext. Verhalten der Eigenschaft: - *unterstützt*, wenn `alerttype` auf ”message” gesetzt ist
retries	integer	Anzahl der Versuche von Zabbix, die Nachricht zu senden.
sendto	string	Adresse, Benutzername oder andere Kennung des Empfängers. Verhalten der Eigenschaft: - *unterstützt*, wenn `alerttype` auf ”message” gesetzt ist
status	integer	Status, der angibt, ob die Aktionsoperation erfolgreich ausgeführt wurde. Mögliche Werte, wenn `alerttype` auf ”message” gesetzt ist: 0 - Nachricht nicht gesendet; 1 - Nachricht gesendet; 2 - nach mehreren Wiederholungsversuchen fehlgeschlagen; 3 - neue Warnung wurde noch nicht vom Warnungsmanager verarbeitet. Mögliche Werte, wenn `alerttype` auf ”remote command” gesetzt ist: 0 - Befehl nicht ausgeführt; 1 - Befehl ausgeführt; 2 - es wurde versucht, den Befehl auf dem Zabbix Agent auszuführen, aber dieser war nicht verfügbar.
subject	string	Betreff der Nachricht. Verhalten der Eigenschaft: - *unterstützt*, wenn `alerttype` auf ”message” gesetzt ist
userid	ID	ID des Benutzers, an den die Nachricht gesendet wurde.
p_eventid	ID	ID des Problemereignisses, das die Warnung erzeugt hat.
acknowledgeid	ID	ID der Bestätigung, die die Warnung erzeugt hat.

alert.get

Beschreibung

`integer/array alert.get(object parameters)`

Die Methode ermöglicht es Ihnen, Alarme anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
alertids	ID/array	Gibt nur Warnungen mit den angegebenen IDs zurück.
actionids	ID/array	Gibt nur Warnungen zurück, die durch die angegebenen Aktionen erzeugt wurden.
eventids	ID/array	Gibt nur Warnungen zurück, die durch die angegebenen Ereignisse erzeugt wurden.
groupids	ID/array	Gibt nur Warnungen zurück, die durch Objekte aus den angegebenen Host-Gruppen erzeugt wurden.
hostids	ID/array	Gibt nur Warnungen zurück, die durch Objekte aus den angegebenen Hosts erzeugt wurden.
mediatypeids	ID/array	Gibt nur Nachrichtenwarnungen zurück, die die angegebenen Medientypen verwendet haben.
objectids	ID/array	Gibt nur Warnungen zurück, die durch die angegebenen Objekte erzeugt wurden
userids	ID/array	Gibt nur Nachrichtenwarnungen zurück, die an die angegebenen Benutzer gesendet wurden.
eventobject	integer	Gibt nur Warnungen zurück, die durch Ereignisse erzeugt wurden, die sich auf Objekte des angegebenen Typs beziehen. Eine Liste der unterstützten Objekttypen finden Sie unter Ereignis-`object`. Standard: 0 - Auslöser.
eventsource	integer	Gibt nur Warnungen zurück, die durch Ereignisse des angegebenen Typs erzeugt wurden. Eine Liste der unterstützten Ereignistypen finden Sie unter Ereignis-`source`. Standard: 0 - Auslöser-Ereignisse.
time_from	timestamp	Gibt nur Warnungen zurück, die nach dem angegebenen Zeitpunkt erzeugt wurden.
time_till	timestamp	Gibt nur Warnungen zurück, die vor dem angegebenen Zeitpunkt erzeugt wurden.
selectHosts	query	Gibt eine Eigenschaft `hosts` mit Daten von Hosts zurück, die die Aktionsoperation ausgelöst haben.
selectMediatypes	query	Gibt eine Eigenschaft `mediatypes` mit einem Array der Medientypen zurück, die für die Nachrichtenwarnung verwendet wurden. Siehe `mediatype.get` für Einschränkungen basierend auf dem Benutzertyp.
selectUsers	query	Gibt eine Eigenschaft `users` mit einem Array der Benutzer zurück, an die die Nachricht adressiert war. Siehe `user.get` für Einschränkungen basierend auf dem Benutzertyp.

Parameter	Type	Beschreibung
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `alertid`, `clock`, `eventid`, `mediatypeid`, `sendto`, `status`.
countOutput	boolean	Diese Parameter sind in der Referenzkommentierung beschrieben.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` kann die folgenden Dinge zurück geben:

- einen Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Warnungen nach Aktions-ID abrufen

Rufen Sie alle Warnungen ab, die durch die Aktion „3“ erzeugt wurden.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "alert.get",
    "params": {
        "output": "extend",
        "actionids": "3"
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```

Siehe auch

- Host
- Medientyp
- Benutzer

Quelle

CAlert::get() in *ui/include/classes/api/services/CAlert.php*.

API-Info

Diese Klasse dient zum Abrufen von Metadaten zur API.

Verfügbare Methoden:

- apiinfo.version - Abrufen der Version der Zabbix-API

apiinfo.version

Beschreibung

`string apiinfo.version(array)`

Diese Methode ermöglicht es Ihnen, die Version der Zabbix API abzurufen.

> **Attention:**
> Diese Methode ist nur für nicht authentifizierte Benutzer verfügbar.

Parameter

`(array)` Die Methode akzeptiert ein leeres Array.

Rückgabewerte

`(string)` Gibt die Version der Zabbix-API zurück.

> **Note:**
> Ab Zabbix 2.0.4 entspricht die Version der API der Version von Zabbix.

Beispiele

Abrufen der Version der API

Rufen Sie die Version der Zabbix-API ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "apiinfo.version",
    "params": [],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": "8.0.0",
    "id": 1
}
```

Quelle

CAPIInfo::version() in *ui/include/classes/api/services/CAPIInfo.php*.

APM-Metrik

Diese Klasse ist für die Arbeit mit Metriken konzipiert, die für das Application Performance Monitoring (APM) erfasst werden.

Objektreferenzen:

- APM-Metrik
- Exemplar

Verfügbare Methoden:

- apmmetric.get – Metrikdatenpunkte abrufen

APM-Metrikobjekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `apmmetric`-API.

Note:
Jede Eigenschaft mit Ausnahme von `type` entspricht einer Spalte in den Metriktabellen (`otel_metrics_gauge`, `otel_metrics_sum`, `otel_metrics_histogram` und `otel_metrics_exponential_histogram`) der Telemetriedatenquelle. Die meisten Eigenschaften entsprechen der Spalte mit demselben Namen (z. B. entspricht die Eigenschaft `metric_name` der Spalte `MetricName`). Wenn sich die Namen unterscheiden, werden die Spalten der Datenquelle in den folgenden Tabellen angegeben. Eine Liste der Spalten finden Sie unter Unterstützte Telemetriedatenspalten.

APM-Metrik

Das APM-Metrikobjekt stellt einen Datenpunkt einer Metrik dar (den Wert oder aggregierte Werte einer Messung zu einem bestimmten Zeitpunkt).

Das APM-Metrikobjekt verfügt über die folgenden Eigenschaften:

Property	Type	Description
type	integer	Typ des Datenpunkts. Mögliche Werte: 0 - Gauge; 1 - Summe; 2 - Histogramm; 3 - Exponentielles Histogramm. Der Typ wird durch die Tabelle bestimmt, in der der Datenpunkt gespeichert ist: `otel_metrics_gauge`, `otel_metrics_sum`, `otel_metrics_histogram` oder `otel_metrics_exponential_histogram`. Verhalten der Eigenschaft: - *read-only*
resource_attributes	object	Schlüssel-Wert-Ressourcenattribute des Datenpunkts (z. B. `host.name: eabf8c2a5ddc`). Verhalten der Eigenschaft: - *read-only*
resource_schema_url	string	URL, sofern vorhanden, des OpenTelemetry-Schemas der Ressourcenattribute. Verhalten der Eigenschaft: - *read-only*
scope_name	string	Name des Instrumentierungsbereichs, der den Datenpunkt erzeugt hat. Verhalten der Eigenschaft: - *read-only*

Property	Type	Description
scope_version	string	Version des Instrumentierungsbereichs, sofern vorhanden. Verhalten der Eigenschaft: - *read-only*
scope_attributes	object	Schlüssel-Wert-Attribute des Instrumentierungsbereichs. Verhalten der Eigenschaft: - *read-only*
scope_schema_url	string	URL, sofern vorhanden, des OpenTelemetry-Schemas des Instrumentierungsbereichs und seiner Attribute. Verhalten der Eigenschaft: - *read-only*
service_name	string	Name des Dienstes, der den Datenpunkt erzeugt hat (z. B. `zabbix-frontend`), übernommen aus dem OpenTelemetry-Ressourcenattribut `service.name`. Verhalten der Eigenschaft: - *read-only*
metric_name	string	Name der Metrik, zu der der Datenpunkt gehört (z. B. `http.server.request.duration`). Verhalten der Eigenschaft: - *read-only*
metric_description	string	Beschreibung der Metrik, sofern vorhanden. Verhalten der Eigenschaft: - *read-only*
metric_unit	string	Einheit der Metrikwerte, sofern vorhanden (z. B. `s` oder `By`; siehe UCUM). Verhalten der Eigenschaft: - *read-only*
attributes	object	Schlüssel-Wert-Attribute des Datenpunkts (z. B. `http.request.method: POST`). Verhalten der Eigenschaft: - *read-only*
start_time_unix	integer	Unix-Zeit (in Nanosekunden), zu der der Messzeitraum des Datenpunkts begonnen hat. Verhalten der Eigenschaft: - *read-only*
time_unix	integer	Unix-Zeit (in Nanosekunden) des Datenpunkts. Bei Datenpunkten vom Typ *Summe*, *Histogramm* und *Exponentielles Histogramm* ist dies das Ende des Messzeitraums. Verhalten der Eigenschaft: - *read-only*
value	float	Wert des Datenpunkts. Gilt für Datenpunkte vom Typ *Gauge* und *Summe*. Standardwert: 0. Verhalten der Eigenschaft: - *read-only*
flags	integer	Bitmaske der Datenpunkt-Flags. Wenn das niederwertigste Bit gesetzt ist (Wert 1), wurde für den Datenpunkt kein Wert aufgezeichnet. Verhalten der Eigenschaft: - *read-only*

Property	Type	Description
exemplars	array	Array von Exemplar-Objekten, also einzelnen Messungen, die als Beispiele zusammen mit dem Trace und Span, während derer sie aufgezeichnet wurden, gespeichert werden. Verhalten der Eigenschaft: - *read-only*
aggregation_temporality	integer	Gibt an, wie die Werte des Datenpunkts über die Zeit aggregiert werden. Gilt für Datenpunkte vom Typ *Summe*, *Histogramm* und *Exponentielles Histogramm*. Mögliche Werte: 0 - *(Standardwert)* Nicht angegeben; 1 - Delta (aggregiert über den Zeitraum seit dem vorherigen Datenpunkt); 2 - Kumulativ (aggregiert über den Zeitraum seit `start_time_unix`). Verhalten der Eigenschaft: - *read-only*
is_monotonic	boolean	Gibt an, ob die Summe im Laufe der Zeit nur steigen oder gleich bleiben kann. Gilt für Datenpunkte vom Typ *Summe*. Standardwert: `false`. Verhalten der Eigenschaft: - *read-only*
count	integer	Anzahl der Messungen im Datenpunkt. Gilt für Datenpunkte vom Typ *Histogramm* und *Exponentielles Histogramm*. Standardwert: 0. Verhalten der Eigenschaft: - *read-only*
sum	float	Summe aller Messungen im Datenpunkt. Gilt für Datenpunkte vom Typ *Histogramm* und *Exponentielles Histogramm*. Standardwert: 0. Verhalten der Eigenschaft: - *read-only*
bucket_counts	array	Array von Ganzzahlen mit der Anzahl der Messungen in jedem Histogramm-Bucket. Die Buckets werden durch `explicit_bounds` definiert. Gilt für Datenpunkte vom Typ *Histogramm*. Standardwert: `[]`. Verhalten der Eigenschaft: - *read-only*
explicit_bounds	array	Array von Gleitkommazahlen mit den Grenzen zwischen Histogramm-Buckets. Gilt für Datenpunkte vom Typ *Histogramm*. Standardwert: `[]`. Verhalten der Eigenschaft: - *read-only*

Property	Type	Description
min	float	Kleinster Messwert im Datenpunkt. Gilt für Datenpunkte vom Typ *Histogramm* und *Exponentielles Histogramm*. Standardwert: 0. Verhalten der Eigenschaft: - *read-only*
max	float	Größter Messwert im Datenpunkt. Gilt für Datenpunkte vom Typ *Histogramm* und *Exponentielles Histogramm*. Standardwert: 0. Verhalten der Eigenschaft: - *read-only*
scale	integer	Skalierung des exponentiellen Histogramms, die die Bucket-Grenzen und die Auflösung bestimmt. Eine höhere Skalierung ergibt schmalere Buckets und eine feinere Auflösung; eine niedrigere Skalierung ergibt breitere Buckets und eine gröbere Auflösung. Gilt für Datenpunkte vom Typ *Exponentielles Histogramm*. Standardwert: 0. Verhalten der Eigenschaft: - *read-only*
zero_count	integer	Anzahl der Messungen im Null-Bucket des exponentiellen Histogramms. Gilt für Datenpunkte vom Typ *Exponentielles Histogramm*. Standardwert: 0. Verhalten der Eigenschaft: - *read-only*
positive_bucket_counts	array	Array von Ganzzahlen mit der Anzahl der positiven Messungen in jedem Bucket des exponentiellen Histogramms. Die Bucket-Grenzen werden durch `scale` bestimmt; der Index des ersten Buckets wird in `positive_offset` festgelegt. Gilt für Datenpunkte vom Typ *Exponentielles Histogramm*. Standardwert: `[]`. Verhalten der Eigenschaft: - *read-only*
positive_offset	integer	Index des ersten Buckets in `positive_bucket_counts`. Gilt für Datenpunkte vom Typ *Exponentielles Histogramm*. Standardwert: 0. Verhalten der Eigenschaft: - *read-only*
negative_bucket_counts	array	Array von Ganzzahlen mit der Anzahl der negativen Messungen in jedem Bucket des exponentiellen Histogramms. Die Bucket-Grenzen werden durch `scale` bestimmt; der Index des ersten Buckets wird in `negative_offset` festgelegt. Gilt für Datenpunkte vom Typ *Exponentielles Histogramm*. Standardwert: `[]`. Verhalten der Eigenschaft: - *read-only*

Property	Type	Description
negative_offset	integer	Index des ersten Buckets in `negative_bucket_counts`. Gilt für Datenpunkte vom Typ *Exponentielles Histogramm*. Standardwert: 0. Verhalten der Eigenschaft: - *read-only*

Exemplar

Das Exemplar-Objekt verfügt über die folgenden Eigenschaften:

Eigenschaft	Typ	Beschreibung
traceid	string	ID, sofern vorhanden, der Trace, während dessen die Messung aufgezeichnet wurde. Datenspalte: `Exemplars.TraceId`.
spanid	string	ID, sofern vorhanden, des Spans, während dessen die Messung aufgezeichnet wurde. Datenspalte: `Exemplars.SpanId`.
value	float	Wert der Messung. Datenspalte: `Exemplars.Value`.
time_unix	integer	Unix-Zeit (in Nanosekunden), zu der die Messung aufgezeichnet wurde. Datenspalte: `Exemplars.TimeUnix`.
filtered_attributes	object	Schlüssel-Wert-Attribute der Messung, die nicht in den `attributes` des Datenpunkts enthalten sind. Datenspalte: `Exemplars.FilteredAttributes`.

apmmetric.get

Beschreibung

`integer/array apmmetric.get(object parameters)`

Mit dieser Methode können Sie Datenpunkte für Metriken entsprechend den angegebenen Parametern abrufen.

> **Note:**
> Diese Methode steht Benutzern jeden Typs zur Verfügung. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Typ	Beschreibung
time_from	timestamp	Gibt nur Datenpunkte zurück, deren Zeit bei der angegebenen Zeit oder danach liegt. Parameterverhalten: - *erforderlich*
time_till	timestamp	Gibt nur Datenpunkte zurück, deren Zeit vor der angegebenen Zeit liegt. Parameterverhalten: - *erforderlich*
types	integer/array	Gibt nur Datenpunkte der angegebenen Typen zurück. Eine Liste der unterstützten Metriktypen finden Sie auf der Seite zum APM-Metrikobjekt.

Parameter	Typ	Beschreibung
with_flags_on	integer	Gibt nur Datenpunkte zurück, bei denen alle angegebenen Flags gesetzt sind. Um mehrere Flags anzugeben, verwenden Sie die Summe ihrer Werte.
with_flags_off	integer	Gibt nur Datenpunkte zurück, bei denen keines der angegebenen Flags gesetzt ist. Um mehrere Flags anzugeben, verwenden Sie die Summe ihrer Werte.
resource_attributes	object/array	Gibt nur Datenpunkte mit den angegebenen Ressourcenattributen zurück. Format: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Ein leeres Array gibt alle Datenpunkte zurück. Mögliche Werte für operator: 0 - *(Standard)* Enthält; 1 - Ist gleich; 2 - Enthält nicht; 3 - Ist nicht gleich; 4 - Ist vorhanden; 5 - Ist nicht vorhanden.
resource_attributes_evaltype	integer	Auswertungsmethode für die Bedingungen in `resource_attributes`. Mögliche Werte: 0 - *(Standard)* Und/Oder; 2 - Oder.
scope_attributes	object/array	Gibt nur Datenpunkte mit den angegebenen Scope-Attributen zurück. Format: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Ein leeres Array gibt alle Datenpunkte zurück. Mögliche Werte für operator: 0 - *(Standard)* Enthält; 1 - Ist gleich; 2 - Enthält nicht; 3 - Ist nicht gleich; 4 - Ist vorhanden; 5 - Ist nicht vorhanden.
scope_attributes_evaltype	integer	Auswertungsmethode für die Bedingungen in `scope_attributes`. Mögliche Werte: 0 - *(Standard)* Und/Oder; 2 - Oder.
attributes	object/array	Gibt nur Datenpunkte mit den angegebenen Attributen zurück. Format: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Ein leeres Array gibt alle Datenpunkte zurück. Mögliche Werte für operator: 0 - *(Standard)* Enthält; 1 - Ist gleich; 2 - Enthält nicht; 3 - Ist nicht gleich; 4 - Ist vorhanden; 5 - Ist nicht vorhanden.
attributes_evaltype	integer	Auswertungsmethode für die Bedingungen in `attributes`. Mögliche Werte: 0 - *(Standard)* Und/Oder; 2 - Oder.

Parameter	Typ	Beschreibung
filter	object	Gibt nur Ergebnisse zurück, die genau mit dem angegebenen Filter übereinstimmen. Akzeptiert ein Objekt, dessen Schlüssel Eigenschaftsnamen und dessen Werte entweder ein einzelner Wert oder ein Array von Werten sind, mit denen verglichen werden soll. Unterstützte Eigenschaften: `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `service_name`, `metric_name`, `metric_unit`, `flags`.
search	object	Gibt Ergebnisse zurück, die dem angegebenen Muster entsprechen (ohne Beachtung der Groß-/Kleinschreibung). Akzeptiert ein Objekt, dessen Schlüssel Eigenschaftsnamen und dessen Werte Zeichenfolgen sind, nach denen gesucht werden soll. Wenn keine zusätzlichen Optionen angegeben werden, wird eine `LIKE "%…%"`-Suche durchgeführt. Unterstützte Eigenschaften: `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `service_name`, `metric_name`, `metric_description`, `metric_unit`.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `type`, `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `service_name`, `metric_name`, `metric_unit`, `start_time_unix`, `time_unix`.
offset	integer	Anzahl der Datensätze, die vor der Rückgabe des Ergebnisses übersprungen werden sollen.
countOutput	boolean	Diese Parameter werden im Referenzkommentar beschrieben.
excludeSearch	boolean	
limit	integer	
output	query	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Gibt entweder Folgendes zurück:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Abrufen von Histogramm-Datenpunkten

Rufen Sie den neuesten Datenpunkt der Histogrammmetrik `http.server.request.duration` des Dienstes `zabbix-frontend` ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "apmmetric.get",
    "params": {
        "output": ["metric_name", "metric_unit", "time_unix", "count", "sum", "min", "max", "explicit_boun
        "time_from": 1790848800,
        "time_till": 1790852400,
        "types": 2,
        "filter": {
```

```
            "service_name": "zabbix-frontend",
            "metric_name": "http.server.request.duration"
        },
        "sortfield": "time_unix",
        "sortorder": "DESC",
        "limit": 1
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "metric_name": "http.server.request.duration",
            "metric_unit": "s",
            "time_unix": "1790852390000000000",
            "count": "1842",
            "sum": "412.773",
            "min": "0.0061",
            "max": "1.098",
            "explicit_bounds": ["0.005", "0.05", "0.1", "0.25", "0.5", "1"],
            "bucket_counts": ["0", "211", "904", "512", "167", "41", "7"],
            "exemplars": [
                {
                    "traceid": "dee625f33cb0994e86c5cd4640e3b09c",
                    "spanid": "3f8b2c6d1e9a7054",
                    "value": "1.098",
                    "time_unix": "1790850513439147623",
                    "filtered_attributes": {}
                }
            ]
        }
    ],
    "id": 1
}
```

Siehe auch

- APM-Trace
- APM-Span
- APM-Log

Quelle

CApmMetric::get() in *ui/include/classes/api/services/CApmMetric.php*.

APM-Protokoll

Diese Klasse ist für die Arbeit mit Protokollen konzipiert, die für das Application Performance Monitoring (APM) erfasst wurden.

Objektreferenzen:

- APM-Protokoll

Verfügbare Methoden:

- apmlog.get – Protokolldatensätze abrufen

APM-Log-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `apmlog`-API.

Note:
Jede Eigenschaft entspricht der Spalte mit demselben Namen in der Tabelle `otel_logs` der Telemetriedatenquelle (z. B. entspricht die Eigenschaft `severity_text` der Spalte `SeverityText`). Eine Liste der Spalten finden Sie unter Unterstützte Telemetriedatenspalten.

APM-Protokoll

Das APM-Protokollobjekt stellt einen Protokolldatensatz dar (einen Textdatensatz eines Ereignisses, z. B. einer fehlgeschlagenen Zahlung oder eines Datenbank-Timeouts). Ein Protokolldatensatz enthält Informationen zu einem Ereignis, das zu einem bestimmten Zeitpunkt aufgetreten ist, einschließlich seines Schweregrads, seiner Meldung und seiner Attribute. Wenn ein Protokolldatensatz während eines Vorgangs erzeugt wurde, der Teil eines Trace ist, enthält er außerdem die IDs dieses Trace und Spans.

Das APM-Protokollobjekt verfügt über folgende Eigenschaften:

Property	Type	Description
timestamp	integer	Unix-Zeit (in Nanosekunden), zu der das Ereignis aufgetreten ist. Property behavior: - *read-only*
traceid	string	ID, sofern vorhanden, des Trace, während dessen der Protokolldatensatz erzeugt wurde. Property behavior: - *read-only*
spanid	string	ID, sofern vorhanden, des Spans, während dessen der Protokolldatensatz erzeugt wurde. Property behavior: - *read-only*
trace_flags	integer	Trace-Flags des Trace-Kontexts des Protokolldatensatzes als Bitmaske. Wenn das niederwertigste Bit gesetzt ist (Wert 1), wird der Trace erfasst. Property behavior: - *read-only*
severity_text	string	Vom Anwendungsprogramm bereitgestellte Schweregradstufe des Protokolldatensatzes (z. B. `INFO`, `WARN` oder `ERROR`). Property behavior: - *read-only*
severity_number	integer	Schweregradstufe des Protokolldatensatzes als Zahl. Mögliche Werte: 0 - UNSPECIFIED; 1-4 - TRACE; 5-8 - DEBUG; 9-12 - INFO; 13-16 - WARN; 17-20 - ERROR; 21-24 - FATAL. Property behavior: - *read-only*
service_name	string	Name des Dienstes, der den Protokolldatensatz erzeugt hat (z. B. `zabbix-frontend`), übernommen aus dem OpenTelemetry-Ressourcenattribut `service.name`. Property behavior: - *read-only*
body	string	Meldung des Protokolldatensatzes. Property behavior: - *read-only*

Property	Type	Description
resource_schema_url	string	URL, sofern vorhanden, des OpenTelemetry-Schemas der Ressourcenattribute. Property behavior: - *read-only*
resource_attributes	object	Schlüssel-Wert-Ressourcenattribute des Protokolldatensatzes (z. B. `host.name: eabf8c2a5ddc`). Property behavior: - *read-only*
scope_schema_url	string	URL, sofern vorhanden, des OpenTelemetry-Schemas des Instrumentierungsbereichs und seiner Attribute. Property behavior: - *read-only*
scope_name	string	Name des Instrumentierungsbereichs, der den Protokolldatensatz erzeugt hat. Property behavior: - *read-only*
scope_version	string	Version, sofern vorhanden, des Instrumentierungsbereichs. Property behavior: - *read-only*
scope_attributes	object	Schlüssel-Wert-Attribute des Instrumentierungsbereichs. Property behavior: - *read-only*
log_attributes	object	Schlüssel-Wert-Protokollattribute des Protokolldatensatzes (z. B. `log.file.path: /var/log/zabbix/zabbix_frontend.log`). Property behavior: - *read-only*
event_name	string	Name, sofern vorhanden, des Ereignistyps, den der Protokolldatensatz darstellt. Property behavior: - *read-only*

apmlog.get

Beschreibung

`integer/array apmlog.get(object parameters)`

Mit dieser Methode können Sie anhand der angegebenen Parameter Protokolleinträge abrufen.

> **Note:**
> Diese Methode steht Benutzern jeden Typs zur Verfügung. Die Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
time_from	timestamp	Gibt nur Protokolldatensätze von Ereignissen zurück, die zum angegebenen Zeitpunkt oder danach aufgetreten sind. Parameterverhalten: - *erforderlich*
time_till	timestamp	Gibt nur Protokolldatensätze von Ereignissen zurück, die vor dem angegebenen Zeitpunkt aufgetreten sind. Parameterverhalten: - *erforderlich*
traceids	string/array	Gibt nur Protokolldatensätze zurück, die während der angegebenen Traces erzeugt wurden.
spanids	string/array	Gibt nur Protokolldatensätze zurück, die während der angegebenen Spans erzeugt wurden.
with_flags_on	integer	Gibt nur Protokolldatensätze zurück, bei denen alle angegebenen Trace-Flags gesetzt sind. Um mehrere Flags anzugeben, verwenden Sie die Summe ihrer Werte.
with_flags_off	integer	Gibt nur Protokolldatensätze zurück, bei denen keines der angegebenen Trace-Flags gesetzt ist. Um mehrere Flags anzugeben, verwenden Sie die Summe ihrer Werte.
resource_attributes	object/array	Gibt nur Protokolldatensätze mit den angegebenen Ressourcenattributen zurück. Format: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Ein leeres Array gibt alle Protokolldatensätze zurück. Mögliche Werte für operator: 0 - *(Standard)* Enthält; 1 - Ist gleich; 2 - Enthält nicht; 3 - Ist nicht gleich; 4 - Ist vorhanden; 5 - Ist nicht vorhanden.
resource_attributes_evaltype	integer	Auswertungsmethode für die Bedingungen in `resource_attributes`. Mögliche Werte: 0 - *(Standard)* Und/Oder; 2 - Oder.
scope_attributes	object/array	Gibt nur Protokolldatensätze mit den angegebenen Scope-Attributen zurück. Format: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Ein leeres Array gibt alle Protokolldatensätze zurück. Mögliche Werte für operator: 0 - *(Standard)* Enthält; 1 - Ist gleich; 2 - Enthält nicht; 3 - Ist nicht gleich; 4 - Ist vorhanden; 5 - Ist nicht vorhanden.
scope_attributes_evaltype	integer	Auswertungsmethode für die Bedingungen in `scope_attributes`. Mögliche Werte: 0 - *(Standard)* Und/Oder; 2 - Oder.

Parameter	Type	Beschreibung
log_attributes	object/array	Gibt nur Protokolldatensätze mit den angegebenen Protokollattributen zurück. Format: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Ein leeres Array gibt alle Protokolldatensätze zurück. Mögliche Werte für operator: 0 - *(Standard)* Enthält; 1 - Ist gleich; 2 - Enthält nicht; 3 - Ist nicht gleich; 4 - Ist vorhanden; 5 - Ist nicht vorhanden.
log_attributes_evaltype	integer	Auswertungsmethode für die Bedingungen in `log_attributes`. Mögliche Werte: 0 - *(Standard)* Und/Oder; 2 - Oder.
filter	object	Gibt nur Ergebnisse zurück, die exakt mit dem angegebenen Filter übereinstimmen. Akzeptiert ein Objekt, dessen Schlüssel Eigenschaftsnamen und dessen Werte entweder ein einzelner Wert oder ein Array von Werten sind, mit denen abgeglichen werden soll. Unterstützte Eigenschaften: `traceid`, `spanid`, `trace_flags`, `severity_text`, `severity_number`, `service_name`, `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `event_name`.
search	object	Gibt Ergebnisse zurück, die dem angegebenen Muster entsprechen (ohne Beachtung der Groß-/Kleinschreibung). Akzeptiert ein Objekt, dessen Schlüssel Eigenschaftsnamen und dessen Werte Zeichenfolgen sind, nach denen gesucht werden soll. Wenn keine zusätzlichen Optionen angegeben werden, wird eine `LIKE "%…%"`-Suche durchgeführt. Unterstützte Eigenschaften: `severity_text`, `service_name`, `body`, `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `event_name`.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `timestamp`, `traceid`, `spanid`, `trace_flags`, `severity_text`, `severity_number`, `service_name`, `resource_schema_url`, `scope_schema_url`, `scope_name`, `scope_version`, `event_name`.
offset	integer	Anzahl der Datensätze, die vor der Rückgabe des Ergebnisses übersprungen werden sollen.
countOutput	boolean	Diese Parameter werden im Referenzkommentar beschrieben.
excludeSearch	boolean	
limit	integer	
output	query	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Gibt entweder Folgendes zurück:

- ein Array von Objekten;

- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Abrufen von Fehlerprotokollen eines Traces

Rufen Sie Protokolleinträge mit dem Schweregrad ERROR (Schweregradnummern 17–20) ab, die dem Trace „8a4f1c7e2b9d4e6fa0c3b5d7e9f1a
zugeordnet sind.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "apmlog.get",
    "params": {
        "output": "extend",
        "time_from": 1790848800,
        "time_till": 1790852400,
        "traceids": "8a4f1c7e2b9d4e6fa0c3b5d7e9f1a2c4",
        "filter": {
            "severity_number": [17, 18, 19, 20]
        },
        "sortfield": "timestamp",
        "sortorder": "ASC"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "timestamp": "1790849305091452376",
            "traceid": "8a4f1c7e2b9d4e6fa0c3b5d7e9f1a2c4",
            "spanid": "5d2e8f1a9c3b7046",
            "trace_flags": "1",
            "severity_text": "ERROR",
            "severity_number": "17",
            "service_name": "zabbix-frontend",
            "body": "mysqli_query(): (HY000/1205): Lock wait timeout exceeded; try restarting transaction"
            "resource_schema_url": "https://opentelemetry.io/schemas/1.26.0",
            "resource_attributes": {
                "service.name": "zabbix-frontend",
                "host.name": "6583ceeb5f94",
                "host.arch": "x86_64",
                "os.type": "linux"
            },
            "scope_schema_url": "",
            "scope_name": "zabbix-frontend",
            "scope_version": "",
            "scope_attributes": {},
            "log_attributes": {
                "log.file.path": "/var/log/zabbix/zabbix_frontend.log",
                "code.filepath": "/usr/share/zabbix/include/db.inc.php"
            },
            "event_name": ""
        }
    ],
    "id": 1
}
```

Suchen nach Logmeldungen

Rufen Sie die Uhrzeit, den Dienst und die Meldung der 10 neuesten Logeinträge ab, deren Meldung „timeout“ enthält.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "apmlog.get",
    "params": {
        "output": ["timestamp", "service_name", "severity_text", "body"],
        "time_from": 1790848800,
        "time_till": 1790852400,
        "search": {
            "body": "timeout"
        },
        "sortfield": "timestamp",
        "sortorder": "DESC",
        "limit": 10
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "timestamp": "1790849305091452376",
            "service_name": "zabbix-frontend",
            "severity_text": "ERROR",
            "body": "mysqli_query(): (HY000/1205): Lock wait timeout exceeded; try restarting transaction"
        },
        {
            "timestamp": "1790848961730284115",
            "service_name": "zabbix-frontend",
            "severity_text": "ERROR",
            "body": "mysqli_query(): (HY000/1205): Lock wait timeout exceeded; try restarting transaction"
        }
    ],
    "id": 1
}
```

Siehe auch

- APM-Trace
- APM-Span
- APM-Metrik

Quelle

CApmLog::get() in *ui/include/classes/api/services/CApmLog.php*.

APM-Span

Diese Klasse ist für die Arbeit mit für das Application Performance Monitoring (APM) erfassten Spans vorgesehen.

Objektreferenzen:

- APM-Span

Verfügbare Methoden:

- apmspan.get - Spans abrufen

APM-Span-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `apmspan`-API.

Note:
Jede Eigenschaft entspricht einer Spalte in der Tabelle `otel_traces` der Telemetriedatenquelle. Eine Liste der Spalten finden Sie unter Unterstützte Telemetriedatenspalten.

APM-Span

Das APM-Span-Objekt stellt einen Span dar (eine einzelne Operation innerhalb eines Trace, z. B. eine HTTP-Anfrage, eine Datenbankabfrage oder ein Funktionsaufruf).

Spans bilden eine Hierarchie, die beim Root-Span beginnt, der die erste Operation der Anfrage darstellt. Jeder andere Span ist ein untergeordneter Span des Spans, während dessen Operation er gestartet wurde.

Das APM-Span-Objekt verfügt über die folgenden Eigenschaften:

Property	Type	Description
timestamp	integer	Unix-Zeit (in Nanosekunden), zu der der Span gestartet wurde. Property behavior: - *read-only*
traceid	string	ID des Trace, zu dem der Span gehört. Property behavior: - *read-only*
spanid	string	ID des Spans. Property behavior: - *read-only*
parent_spanid	string	ID des übergeordneten Spans. Beim Root-Span leer. Property behavior: - *read-only*
trace_state	string	Zusätzliche Daten, sofern vorhanden, die Tracing-Tools an den Trace anhängen (siehe W3C Trace Context). Property behavior: - *read-only*
span_name	string	Name der Operation, die der Span darstellt (z. B. `mysqli_query`). Property behavior: - *read-only*
span_kind	string	Rolle des Spans. Mögliche Werte: `Unspecified` - Span-Typ ist nicht angegeben; `Internal` - innerhalb einer Anwendung ausgeführte Operation; `Server` - Verarbeitung einer eingehenden Anfrage; `Client` - ausgehende Anfrage; `Producer` - Erstellung einer asynchronen Operation (z. B. das Senden einer Nachricht); `Consumer` - Verarbeitung einer asynchronen Operation. Property behavior: - *read-only*
service_name	string	Name des Dienstes, der den Span erzeugt hat (z. B. `zabbix-frontend`), übernommen aus dem OpenTelemetry-Ressourcenattribut `service.name`. Property behavior: - *read-only*

Property	Type	Description
resource_attributes	object	Schlüssel-Wert-Ressourcenattribute des Spans (z. B. `host.name: eabf8c2a5ddc`). Property behavior: - *read-only*
scope_name	string	Name des Instrumentierungsbereichs, der den Span erzeugt hat. Property behavior: - *read-only*
scope_version	string	Version des Instrumentierungsbereichs. Property behavior: - *read-only*
span_attributes	object	Schlüssel-Wert-Span-Attribute des Spans (z. B. `db.system: mysql`). Property behavior: - *read-only*
duration	integer	Dauer des Spans in Nanosekunden. Property behavior: - *read-only*
status_code	string	Status der Operation, die der Span darstellt. Mögliche Werte: `Unset` - dem Span wurde kein expliziter Status zugewiesen; `Ok` - die Operation wurde erfolgreich abgeschlossen; `Error` - die Operation wurde mit einem Fehler abgeschlossen. Property behavior: - *read-only*
status_message	string	Zusätzliche Informationen, sofern vorhanden, zum Status des Spans. Property behavior: - *read-only*
events	array	Array von Ereignis-Objekten des Spans, also bemerkenswerten Zeitpunkten während der Operation (z. B. Ausnahmen). Property behavior: - *read-only*
links	array	Array von Link-Objekten des Spans, die auf verwandte Spans im selben oder in einem anderen Trace verweisen. Property behavior: - *read-only*

apmspan.get

Beschreibung

`integer/array apmspan.get(object parameters)`

Mit dieser Methode können Sie Spans entsprechend den angegebenen Parametern abrufen.

> **Note:**
> Diese Methode steht Benutzern jeden Typs zur Verfügung. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Description
time_from	timestamp	Gibt nur Spans zurück, die zum angegebenen Zeitpunkt oder danach gestartet wurden. Parameterverhalten: - *erforderlich*
time_till	timestamp	Gibt nur Spans zurück, die vor dem angegebenen Zeitpunkt gestartet wurden. Parameterverhalten: - *erforderlich*
traceids	string/array	Gibt nur Spans zurück, die zu den angegebenen Traces gehören.
spanids	string/array	Gibt nur Spans mit den angegebenen IDs zurück.
parent_spanids	string/array	Gibt nur Spans mit den angegebenen übergeordneten Spans zurück.
resource_attributes	object/array	Gibt nur Spans mit den angegebenen Ressourcenattributen zurück. Format: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Ein leeres Array gibt alle Spans zurück. Mögliche Werte für operator: 0 - *(Standard)* Enthält; 1 - Ist gleich; 2 - Enthält nicht; 3 - Ist nicht gleich; 4 - Ist vorhanden; 5 - Ist nicht vorhanden.
resource_attributes_evaltype	integer	Auswertungsmethode für die Bedingungen in `resource_attributes`. Mögliche Werte: 0 - *(Standard)* Und/Oder; 2 - Oder.
span_attributes	object/array	Gibt nur Spans mit den angegebenen Spanattributen zurück. Format: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Ein leeres Array gibt alle Spans zurück. Mögliche Werte für operator: 0 - *(Standard)* Enthält; 1 - Ist gleich; 2 - Enthält nicht; 3 - Ist nicht gleich; 4 - Ist vorhanden; 5 - Ist nicht vorhanden.
span_attributes_evaltype	integer	Auswertungsmethode für die Bedingungen in `span_attributes`. Mögliche Werte: 0 - *(Standard)* Und/Oder; 2 - Oder.
min_duration	integer	Gibt nur Spans zurück, deren Dauer größer oder gleich dem angegebenen Wert in Nanosekunden ist.
max_duration	integer	Gibt nur Spans zurück, deren Dauer kleiner als der angegebene Wert in Nanosekunden ist.
filter	object	Gibt nur Ergebnisse zurück, die exakt mit dem angegebenen Filter übereinstimmen. Akzeptiert ein Objekt, dessen Schlüssel Eigenschaftsnamen und dessen Werte entweder ein einzelner Wert oder ein Array von Werten sind, mit denen abgeglichen werden soll. Unterstützte Eigenschaften: `traceid`, `spanid`, `parent_spanid`, `trace_state`, `span_name`, `span_kind`, `service_name`, `scope_name`, `scope_version`, `status_code`, `status_message`.

Parameter	Type	Description
search	object	Gibt Ergebnisse zurück, die dem angegebenen Muster entsprechen (ohne Beachtung der Groß-/Kleinschreibung). Akzeptiert ein Objekt, dessen Schlüssel Eigenschaftsnamen und dessen Werte Zeichenfolgen sind, nach denen gesucht werden soll. Wenn keine zusätzlichen Optionen angegeben werden, wird eine `LIKE "%…%"`-Suche durchgeführt. Unterstützte Eigenschaften: `trace_state`, `span_name`, `service_name`, `scope_name`, `scope_version`, `status_message`.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `timestamp`, `traceid`, `spanid`, `parent_spanid`, `trace_state`, `span_name`, `span_kind`, `service_name`, `scope_name`, `scope_version`, `duration`, `status_code`, `status_message`.
offset	integer	Anzahl der Datensätze, die vor der Rückgabe des Ergebnisses übersprungen werden.
countOutput	boolean	Diese Parameter werden in der Referenzdokumentation beschrieben.
excludeSearch	boolean	
limit	integer	
output	query	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Gibt entweder Folgendes zurück:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Abrufen der Span-Hierarchie eines Trace

Rufen Sie die ersten vier Spans des Traces „dee625f33cb0994e86c5cd4640e3b09c“ ab, sortiert nach Startzeit. Die Eigenschaft `parent_spanid` zeigt, wie die Spans miteinander verbunden sind: Der Root-Span hat kein übergeordnetes Element, und jeder andere Span verweist auf den Span, durch den er gestartet wurde.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "apmspan.get",
    "params": {
        "output": ["spanid", "parent_spanid", "service_name", "span_name", "duration", "status_code"],
        "time_from": 1790848800,
        "time_till": 1790852400,
        "traceids": "dee625f33cb0994e86c5cd4640e3b09c",
        "sortfield": "timestamp",
        "sortorder": "ASC",
        "limit": 4
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
```

```
            "spanid": "7c1a9e2f4b6d8a03",
            "parent_spanid": "",
            "service_name": "zabbix-nginx",
            "span_name": "POST /zabbix.php",
            "duration": "1104826531",
            "status_code": "Unset"
        },
        {
            "spanid": "3f8b2c6d1e9a7054",
            "parent_spanid": "7c1a9e2f4b6d8a03",
            "service_name": "zabbix-frontend",
            "span_name": "POST /zabbix.php",
            "duration": "1098230418",
            "status_code": "Unset"
        },
        {
            "spanid": "b3390ad41931ed08",
            "parent_spanid": "3f8b2c6d1e9a7054",
            "service_name": "zabbix-frontend",
            "span_name": "mysqli::real_connect",
            "duration": "36671360",
            "status_code": "Ok"
        },
        {
            "spanid": "e41f7a2c9d5b3086",
            "parent_spanid": "3f8b2c6d1e9a7054",
            "service_name": "zabbix-frontend",
            "span_name": "mysqli_query",
            "duration": "19704118",
            "status_code": "Ok"
        }
    ],
    "id": 1
}
```


Siehe auch

- APM-Trace
- APM-Log
- APM-Metrik

Quelle

CApmSpan::get() in *ui/include/classes/api/services/CApmSpan.php*.

APM-Trace

Diese Klasse ist für die Arbeit mit Traces konzipiert, die für das Application Performance Monitoring (APM) erfasst wurden.

Objektreferenzen:

- APM-Trace
- Ereignis
- Verknüpfung

Verfügbare Methoden:

- apmtrace.get – Traces abrufen

APM-Trace-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `apmtrace`-API.

Note:
Jede Eigenschaft mit Ausnahme von `span_count` und `error_count` entspricht einer Spalte in der Tabelle `otel_traces` der Telemetriedatenquelle. Die meisten Eigenschaften entsprechen der Spalte mit demselben Namen (z. B. entspricht die Eigenschaft `span_name` der Spalte `SpanName`). Wenn die Namen voneinander abweichen, werden die Spalten der Datenquelle in den folgenden Tabellen angegeben. Eine Liste der Spalten finden Sie unter Unterstützte Spalten für Telemetriedaten.

APM-Trace

Das APM-Trace-Objekt stellt einen Trace dar (den Pfad, den eine Anfrage durch Ihre Anwendungen und Dienste nimmt). Ein Trace besteht aus einem oder mehreren Spans; jeder Span stellt eine einzelne Operation dar.

Das APM-Trace-Objekt basiert auf dem Root-Span des Traces (dem Span, der den Trace startet und keinen übergeordneten Span hat). Alle Eigenschaften enthalten Daten des Root-Spans, mit Ausnahme von `span_count` und `error_count`, die aus allen Spans des Traces berechnet werden.

Das APM-Trace-Objekt hat die folgenden Eigenschaften:

Property	Type	Description
timestamp	integer	Unix-Zeit (in Nanosekunden), zu der der Root-Span gestartet wurde. Verhalten der Eigenschaft: - *nur lesbar*
traceid	string	ID des Traces, die von allen Spans dieses Traces gemeinsam verwendet wird. Verhalten der Eigenschaft: - *nur lesbar*
spanid	string	ID des Root-Spans. Verhalten der Eigenschaft: - *nur lesbar*
trace_state	string	Zusätzliche Daten, sofern vorhanden, die Tracing-Tools an den Trace anhängen (siehe W3C Trace Context). Verhalten der Eigenschaft: - *nur lesbar*
span_name	string	Name der Operation, die der Root-Span darstellt (z. B. `POST /zabbix.php`). Verhalten der Eigenschaft: - *nur lesbar*
span_kind	string	Rolle des Root-Spans. Mögliche Werte: `Unspecified` – Span-Art ist nicht angegeben; `Internal` – innerhalb einer Anwendung ausgeführte Operation; `Server` – Verarbeitung einer eingehenden Anfrage; `Client` – ausgehende Anfrage; `Producer` – Erstellung einer asynchronen Operation (z. B. Senden einer Nachricht); `Consumer` – Verarbeitung einer asynchronen Operation. Verhalten der Eigenschaft: - *nur lesbar*
service_name	string	Name des Dienstes, der den Root-Span erzeugt hat (z. B. `zabbix-nginx`), übernommen aus dem OpenTelemetry-Ressourcenattribut `service.name`. Verhalten der Eigenschaft: - *nur lesbar*

Property	Type	Description
resource_attributes	object	Schlüssel-Wert-Ressourcenattribute des Root-Spans (z. B. `host.name: eabf8c2a5ddc`). Verhalten der Eigenschaft: - *nur lesbar*
scope_name	string	Name des Instrumentierungsbereichs, der den Root-Span erzeugt hat. Verhalten der Eigenschaft: - *nur lesbar*
scope_version	string	Version des Instrumentierungsbereichs. Verhalten der Eigenschaft: - *nur lesbar*
span_attributes	object	Schlüssel-Wert-Span-Attribute des Root-Spans (z. B. `http.method: POST`). Verhalten der Eigenschaft: - *nur lesbar*
duration	integer	Dauer des Traces in Nanosekunden. Verhalten der Eigenschaft: - *nur lesbar*
status_code	string	Status der Operation, die der Root-Span darstellt. Mögliche Werte: `Unset` – dem Root-Span wurde kein expliziter Status zugewiesen; `Ok` – die Operation wurde erfolgreich abgeschlossen; `Error` – die Operation wurde mit einem Fehler abgeschlossen. Verhalten der Eigenschaft: - *nur lesbar*
status_message	string	Zusätzliche Informationen, sofern vorhanden, zum Status des Root-Spans. Verhalten der Eigenschaft: - *nur lesbar*
events	array	Array von Ereignis-Objekten des Root-Spans, also bemerkenswerten Zeitpunkten während der Operation (z. B. Ausnahmen). Verhalten der Eigenschaft: - *nur lesbar*
links	array	Array von Link-Objekten des Root-Spans, die auf zugehörige Spans im selben oder in einem anderen Trace verweisen. Verhalten der Eigenschaft: - *nur lesbar*
span_count	integer	Gesamtanzahl der Spans im Trace. Verhalten der Eigenschaft: - *nur lesbar*
error_count	integer	Anzahl der Spans im Trace, bei denen `status_code` auf `Error` gesetzt ist. Verhalten der Eigenschaft: - *nur lesbar*

Ereignis

Das Ereignisobjekt verfügt über die folgenden Eigenschaften.

Property	Type	Description
timestamp	integer	Unix-Zeit (in Nanosekunden), zu der das Ereignis aufgetreten ist. Spalte der Datenquelle: `Events.Timestamp`
name	string	Name des Ereignisses. Spalte der Datenquelle: `Events.Name`
attributes	object	Schlüssel-Wert-Attribute mit zusätzlichen Informationen zum Ereignis, sofern vorhanden. Spalte der Datenquelle: `Events.Attributes`

Link

Das Link-Objekt verfügt über die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
traceid	string	ID des Traces, der den verknüpften Span enthält. Datenqualtenspalte: `Links.TraceId`
spanid	string	ID des verknüpften Spans. Zusammen mit `traceid` identifiziert diese ID den verknüpften Span eindeutig. Datenqualtenspalte: `Links.SpanId`
trace_state	string	Trace-Status des verknüpften Spans, sofern vorhanden (siehe W3C Trace Context). Datenqualtenspalte: `Links.TraceState`
attributes	object	Schlüssel-Wert-Attribute mit zusätzlichen Informationen zum Link. Datenqualtenspalte: `Links.Attributes`

apmtrace.get

Beschreibung

`integer/array apmtrace.get(object parameters)`

Mit dieser Methode können Sie anhand der angegebenen Parameter Traces abrufen.

> **Note:**
> Diese Methode steht Benutzern jeden Typs zur Verfügung. Die Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Typ	Beschreibung
time_from	timestamp	Gibt nur Traces zurück, deren Root-Span zum angegebenen Zeitpunkt oder danach gestartet wurde. Parameterverhalten: - *erforderlich*
time_till	timestamp	Gibt nur Traces zurück, deren Root-Span vor dem angegebenen Zeitpunkt gestartet wurde. Parameterverhalten: - *erforderlich*
traceids	string/array	Gibt nur Traces mit den angegebenen IDs zurück.

Parameter	Typ	Beschreibung
resource_attributes	object/array	Gibt nur Traces zurück, deren Root-Span die angegebenen Ressourcenattribute besitzt. Format: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Ein leeres Array gibt alle Traces zurück. Mögliche Werte für operator: 0 - *(Standard)* Enthält; 1 - Ist gleich; 2 - Enthält nicht; 3 - Ist nicht gleich; 4 - Ist vorhanden; 5 - Ist nicht vorhanden.
resource_attributes_evaltype	integer	Auswertungsmethode für die Bedingungen in `resource_attributes`. Mögliche Werte: 0 - *(Standard)* Und/Oder; 2 - Oder.
span_attributes	object/array	Gibt nur Traces zurück, deren Root-Span die angegebenen Span-Attribute besitzt. Format: `[{"key": "<key>", "value": "<value>", "operator": "<operator>"}, ...]`. Ein leeres Array gibt alle Traces zurück. Mögliche Werte für operator: 0 - *(Standard)* Enthält; 1 - Ist gleich; 2 - Enthält nicht; 3 - Ist nicht gleich; 4 - Ist vorhanden; 5 - Ist nicht vorhanden.
span_attributes_evaltype	integer	Auswertungsmethode für die Bedingungen in `span_attributes`. Mögliche Werte: 0 - *(Standard)* Und/Oder; 2 - Oder.
min_duration	integer	Gibt nur Traces zurück, deren `duration` des Root-Spans größer oder gleich dem angegebenen Wert in Nanosekunden ist.
max_duration	integer	Gibt nur Traces zurück, deren `duration` des Root-Spans kleiner als der angegebene Wert in Nanosekunden ist.
filter	object	Gibt nur Ergebnisse zurück, die exakt mit dem angegebenen Filter übereinstimmen. Akzeptiert ein Objekt, dessen Schlüssel Eigenschaftsnamen und dessen Werte entweder ein einzelner Wert oder ein Array von Werten sind, mit denen abgeglichen wird. Unterstützte Eigenschaften: `traceid`, `spanid`, `trace_state`, `span_name`, `span_kind`, `service_name`, `scope_name`, `scope_version`, `status_code`, `status_message`.
search	object	Gibt Ergebnisse zurück, die dem angegebenen Muster entsprechen (ohne Beachtung der Groß-/Kleinschreibung). Akzeptiert ein Objekt, dessen Schlüssel Eigenschaftsnamen und dessen Werte Zeichenfolgen sind, nach denen gesucht wird. Wenn keine zusätzlichen Optionen angegeben werden, wird eine `LIKE "%…%"`-Suche durchgeführt. Unterstützte Eigenschaften: `trace_state`, `span_name`, `service_name`, `scope_name`, `scope_version`, `status_message`.

Parameter	Typ	Beschreibung
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `timestamp`, `traceid`, `spanid`, `trace_state`, `span_name`, `span_kind`, `service_name`, `scope_name`, `scope_version`, `duration`, `status_code`, `status_message`.
offset	integer	Anzahl der Datensätze, die vor der Rückgabe des Ergebnisses übersprungen werden.
countOutput	boolean	Diese Parameter werden in der Referenzdokumentation beschrieben.
excludeSearch	boolean	
limit	integer	
output	query	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Gibt entweder Folgendes zurück:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Langsame Traces abrufen

Rufen Sie die beiden langsamsten Traces des Dienstes `zabbix-nginx` ab.
Suchen Sie nach Traces vom 1. Oktober 2026 (UTC) zwischen 10:00 und 11:00 Uhr und berücksichtigen Sie nur diejenigen, deren Dauer mindestens 1 Sekunde betrug.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "apmtrace.get",
    "params": {
        "output": ["timestamp", "traceid", "span_name", "service_name", "duration", "status_code", "span_c
        "time_from": 1790848800,
        "time_till": 1790852400,
        "filter": {
            "service_name": "zabbix-nginx"
        },
        "min_duration": 1000000000,
        "sortfield": "duration",
        "sortorder": "DESC",
        "limit": 2
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "timestamp": "1790849274910226417",
            "traceid": "8a4f1c7e2b9d4e6fa0c3b5d7e9f1a2c4",
            "span_name": "POST /zabbix.php",
            "service_name": "zabbix-nginx",
            "duration": "30412907815",
            "status_code": "Error",
            "span_count": "49",
            "error_count": "2"
```

```
        },
        {
            "timestamp": "1790850512338104952",
            "traceid": "dee625f33cb0994e86c5cd4640e3b09c",
            "span_name": "POST /zabbix.php",
            "service_name": "zabbix-nginx",
            "duration": "1104826531",
            "status_code": "Unset",
            "span_count": "27",
            "error_count": "0"
        }
    ],
    "id": 1
}
```


Siehe auch

- APM span
- APM metric
- APM log

Quelle

CApmTrace::get() in *ui/include/classes/api/services/CApmTrace.php*.

Audit-Protokoll

Diese Klasse ist für die Arbeit mit dem Audit-Protokoll vorgesehen.

Objektreferenzen:

- Audit-Protokoll

Verfügbare Methoden:

- auditlog.get - Audit-Protokolleinträge abrufen

Audit-Log-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `auditlog` API.

Audit-Protokoll

Das Audit-Protokollobjekt enthält Informationen über Benutzeraktionen. Es hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
auditid	ID	ID des Audit-Protokolleintrags. Generiert mit dem CUID-Algorithmus.
userid	ID	userid des Autors des Audit-Protokolleintrags.
username	string	Benutzername des Autors des Audit-Protokolleintrags.
clock	timestamp	Zeitstempel der Erstellung des Audit-Protokolleintrags.
ip	string	IP-Adresse des Autors des Audit-Protokolleintrags.
action	integer	Aktion des Audit-Protokolleintrags. Mögliche Werte: 0 - Hinzufügen; 1 - Aktualisieren; 2 - Löschen; 4 - Abmelden; 7 - Ausführen; 8 - Anmelden; 9 - Fehlgeschlagene Anmeldung; 10 - Verlauf löschen; 11 - Konfiguration aktualisieren; 12 - Push.

Eigenschaft	Typ	Beschreibung
resourcetype	integer	Ressourcentyp des Audit-Protokolleintrags. Mögliche Werte: 0 - Benutzer; 3 - Medientyp; 4 - Host; 5 - Aktion; 6 - Diagramm; 11 - Benutzergruppe; 13 - Auslöser; 14 - Host-Gruppe; 15 - Datenpunkt; 16 - Bild; 17 - Wertezuordnung; 18 - Service; 19 - Karte; 22 - Webszenario; 23 - Discovery-Regel; 25 - Skript; 26 - Proxy; 27 - Wartung; 28 - Regulärer Ausdruck; 29 - Makro; 30 - Vorlage; 31 - Auslöserprototyp; 32 - Symbolzuordnung; 33 - Dashboard; 34 - Ereigniskorrelation; 35 - Diagrammprototyp; 36 - Datenpunktprototyp; 37 - Host-Prototyp; 38 - Autoregistrierung; 39 - Modul; 40 - Einstellungen; 41 - Housekeeping; 42 - Authentifizierung; 43 - Vorlagen-Dashboard; 44 - Benutzerrolle; 45 - API-Token; 46 - Geplanter Bericht; 47 - Hochverfügbarkeitsknoten; 48 - SLA; 49 - Benutzerverzeichnis; 50 - Vorlagengruppe; 51 - Konnektor; 52 - LLD-Regel; 53 - Verlauf; 54 - Multi-Faktor-Authentifizierung; 55 - Proxy-Gruppe; 56 - LLD-Regelprototyp.
resourceid	ID	Ressourcenkennung des Audit-Protokolleintrags.
resource_cuid	ID	Eindeutige Ressourcenkennung des Audit-Protokolleintrags, generiert mit dem CUID-Algorithmus.
resourcename	string	Für Menschen lesbarer Name der Ressource des Audit-Protokolleintrags.
recordsetid	ID	Recordset-ID des Audit-Protokolleintrags. Die während derselben Operation erstellten Audit-Protokolleinträge haben dieselbe Recordset-ID. Generiert mit dem CUID-Algorithmus.

Eigenschaft	Typ	Beschreibung
details	text	Details des Audit-Protokolleintrags. Die Details werden als JSON-Objekt gespeichert, wobei jeder Eigenschaftsname ein Pfad zur Eigenschaft oder zum verschachtelten Objekt ist, in dem die Änderung aufgetreten ist, und jeder Wert die Daten (im Array-Format) über die Änderung an dieser Eigenschaft oder diesem verschachtelten Objekt enthält. Mögliche Werteformate: ["add"] - Verschachteltes Objekt wurde hinzugefügt; ["add", "<value>"] - Die Eigenschaft des hinzugefügten Objekts ist gleich <value>; ["update"] - Verschachteltes Objekt wurde aktualisiert; ["update", "<new value>", "<old value>"] - Die Eigenschaft des aktualisierten Objekts wurde von <old value> auf <new value> geändert; ["delete"] - Verschachteltes Objekt wurde gelöscht.

auditlog.get

Beschreibung

`integer/array auditlog.get(object parameters)`

Die Methode ermöglicht es Ihnen, Audit-Log-Einträge anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Typ	Beschreibung
auditids	ID/array	Gibt nur Audit-Logs mit den angegebenen IDs zurück.
userids	ID/array	Gibt nur Audit-Logs zurück, die von den angegebenen Benutzern erstellt wurden.
time_from	timestamp	Gibt nur Audit-Log-Einträge zurück, die nach oder zum angegebenen Zeitpunkt erstellt wurden.
time_till	timestamp	Gibt nur Audit-Log-Einträge zurück, die vor oder zum angegebenen Zeitpunkt erstellt wurden.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `auditid`, `userid`, `clock`.
countOutput	boolean	Diese Parameter werden im Referenzkommentar beschrieben.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Gibt entweder Folgendes zurück:

- ein Array von Objekten;

- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Audit-Log abrufen

Die zwei neuesten Audit-Log-Einträge abrufen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "auditlog.get",
    "params": {
        "output": "extend",
        "sortfield": "clock",
        "sortorder": "DESC",
        "limit": 2
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```

Siehe auch

- Audit-Log-Objekt

Quelle

CAuditLog::get() in *ui/include/classes/api/services/CAuditLog.php*.

Aufgabe

Diese Klasse ist für die Arbeit mit Aufgaben vorgesehen (z. B. zum Prüfen von Datenpunkten oder Low-Level-Discovery-Regeln ohne Neuladen der Konfiguration).

Objektreferenzen:

- Aufgabe
  - Anfrageobjekt „Jetzt ausführen“
  - Anfrageobjekt „Proxy-Konfiguration aktualisieren“
  - Anfrageobjekt „Diagnoseinformationen“
    * Anfrageobjekt „Statistik“
    * Ergebnisobjekt „Statistik“

Verfügbare Methoden:

- task.create - neue Aufgaben erstellen
- task.get - Aufgaben abrufen

Task-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `task`-API.

Task

Das Task-Objekt hat die folgenden Eigenschaften:

Eigenschaft	Typ	Beschreibung
taskid	ID	ID der Task. Eigenschaftsverhalten: - *schreibgeschützt*
type	integer	Typ der Task. Mögliche Werte: 1 - Diagnoseinformationen; 2 - Proxy-Konfiguration aktualisieren; 6 - Jetzt ausführen. Benutzer der Typen *Admin* und *User* können nur Tasks vom Typ „Jetzt ausführen“ erstellen. Beachten Sie, dass `task.get` immer „7“ (Zusammenfassung der Task-Ausführung) zurückgibt. Eigenschaftsverhalten: - *erforderlich*
status	integer	Status der Task. Mögliche Werte: 1 - neue Task; 2 - Task in Bearbeitung; 3 - Task ist abgeschlossen; 4 - Task ist abgelaufen. Eigenschaftsverhalten: - *schreibgeschützt*
clock	timestamp	Zeitpunkt, zu dem die Task erstellt wurde. Eigenschaftsverhalten: - *schreibgeschützt*

Eigenschaft	Typ	Beschreibung
ttl	integer	Die Zeit in Sekunden, nach der die Task abläuft. Eigenschaftsverhalten: - *schreibgeschützt*
proxyid	ID	ID des Proxy, über den Statistiken zu Diagnoseinformationen erfasst werden. Eigenschaftsverhalten: - *unterstützt*, wenn `type` auf „Diagnoseinformationen“ oder „Proxy-Konfiguration aktualisieren“ gesetzt ist
request	object	Task-Anfrageobjekt entsprechend dem Task-Typ: Das Objekt der Task „Jetzt ausführen“ wird weiter unten ausführlich beschrieben; das Objekt der Task „Proxy-Konfiguration aktualisieren“ wird weiter unten ausführlich beschrieben; das Objekt der Task „Diagnoseinformationen“ wird weiter unten ausführlich beschrieben. Eigenschaftsverhalten: - *erforderlich*
result	object	Ergebnisobjekt der Task für Diagnoseinformationen. Kann NULL enthalten, wenn das Ergebnis noch nicht bereit ist. Das Ergebnisobjekt wird weiter unten ausführlich beschrieben. Eigenschaftsverhalten: - *schreibgeschützt*

Anfrageobjekt für „Jetzt ausführen“

Das Aufgaben-Anfrageobjekt „Jetzt ausführen“ hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
itemid	ID	ID von Datenpunkten und Low-Level-Discovery-Regeln. Benutzer des Typs *Admin* und *User* können Datenpunkte auf Hosts „Jetzt ausführen“, für die sie über *Lese-/Schreib-* Berechtigung verfügen oder über *Lese*-Berechtigung und die Aktion `invoke_execute_now` für ihre Rolle aktiviert ist. Dasselbe gilt für Benutzer des Typs *Admin* bei Low-Level-Discovery-(LLD-)Regeln.

Anforderungsobjekt „Refresh proxy configuration“

Das Anforderungsobjekt der Aufgabe „Refresh proxy configuration“ hat die folgenden Eigenschaften.

Eigenschaft	Type	Beschreibung
proxyids	array	Proxy-IDs.

Anfrageobjekt „Diagnoseinformationen“

Das Anfrageobjekt für die Aufgabe „Diagnoseinformationen“ hat die folgenden Eigenschaften. Das Statistik-Anfrageobjekt für alle Eigenschaftstypen wird weiter unten ausführlich beschrieben.

Eigenschaft	Typ	Beschreibung
historycache	object	Statistik-Anfrage für den Verlaufscache. Verfügbar auf Server und Proxy.
valuecache	object	Statistik-Anfrage für den Datenpunkt-Cache. Verfügbar auf Server.
preprocessing	object	Statistik-Anfrage für den Präprozessierungsmanager. Verfügbar auf Server und Proxy.

Eigenschaft	Typ	Beschreibung
alerting	object	Statistik-Anfrage für den Alarmmanager. Verfügbar auf Server.
lld	object	Statistik-Anfrage für den LLD-Manager. Verfügbar auf Server.

Objekt für Statistikabfragen

Das Objekt für Statistikabfragen wird verwendet, um festzulegen, welche Art von Informationen über interne Prozesse von Server/Proxy erfasst werden soll. Es hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
stats	query	Eigenschaften des Statistikobjekts, die zurückgegeben werden sollen. Die Liste der verfügbaren Felder für jeden Typ von Statistiken zu Diagnoseinformationen wird weiter unten im Detail beschrieben. Standard: `extend` gibt alle verfügbaren Statistikfelder zurück.
top	object	Objekt zum Sortieren und Begrenzen der zurückgegebenen Statistikwerte. Die Liste der verfügbaren Felder für jeden Typ von Statistiken zu Diagnoseinformationen wird weiter unten im Detail beschrieben. Beispiel: { “source.alerts”: 10 }

Liste der Statistikfelder, die für jeden Typ von Diagnoseinformationsanfrage verfügbar sind

Die folgenden Statistikfelder können für jede Eigenschaft eines Typs von Diagnoseinformationsanfrage angefordert werden.

Diagnosetyp	Verfügbare Felder	Beschreibung
historycache	items	Anzahl der zwischengespeicherten Datenpunkte.
	values	Anzahl der zwischengespeicherten Werte.
	memory	Statistiken zum Shared Memory (freier Speicherplatz, Anzahl verwendeter Blöcke, Anzahl freier Blöcke, maximale Größe eines freien Blocks).
	memory.data	Shared-Memory-Statistiken des Verlaufsdatencaches.
	memory.index	Shared-Memory-Statistiken des Verlaufsindexcaches.
valuecache	items	Anzahl der zwischengespeicherten Datenpunkte.
	values	Anzahl der zwischengespeicherten Werte.
	memory	Statistiken zum Shared Memory (freier Speicherplatz, Anzahl verwendeter Blöcke, Anzahl freier Blöcke, maximale Größe eines freien Blocks).
	mode	Modus des Wertecaches.
preprocessing	values	Anzahl der Werte in der Warteschlange.
	preproc.values	Anzahl der Werte in der Warteschlange mit Vorverarbeitungsschritten.
alerting	alerts	Anzahl der Benachrichtigungen in der Warteschlange.
lld	rules	Anzahl der Regeln in der Warteschlange.
	values	Anzahl der Werte in der Warteschlange.

Liste der Sortierfelder, die für jeden Typ von Diagnoseinformationsanfrage verfügbar sind

Die folgenden Statistikfelder können verwendet werden, um angeforderte Informationen zu sortieren und zu begrenzen.

Diagnosetyp	Verfügbare Felder	Typ
historycache	values	integer
valuecache	values	integer
	request.values	integer
preprocessing	values	integer
alerting	media.alerts	integer
	source.alerts	integer
lld	values	integer

Statistik-Ergebnisobjekt

Das Statistik-Ergebnisobjekt wird im Feld `result` des task-Objekts abgerufen.

Eigenschaft	Type	Beschreibung
status	integer	Status des task-Ergebnisses. Mögliche Werte: -1 - während der Ausführung des task ist ein Fehler aufgetreten; 0 - das task-Ergebnis wurde erstellt. Property behavior: - *read-only*
data	string/object	Ergebnisse entsprechend dem Statistik-Anfrageobjekt der jeweiligen Diagnoseinformationstask. Enthält eine Fehlerzeichenfolge, wenn während der Ausführung des task ein Fehler aufgetreten ist.

task.create

Beschreibung

`object task.create(object/array tasks)`

Diese Methode ermöglicht es Ihnen, Aufgaben zu erstellen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Eine zu erstellende Aufgabe.

Die Methode akzeptiert Aufgaben mit den Standard-Aufgabeneigenschaften.

Beachten Sie, dass Aufgaben vom Typ „Jetzt ausführen“ nur für die folgenden Typen von Datenpunkten/Discovery-Regeln erstellt werden können:

- Zabbix-Agent (passiv)
- Einfache Prüfung
- SNMP-Agent (v1/v2/v3)
- Zabbix-intern
- Externe Prüfung
- Datenbankmonitor
- HTTP-Agent
- IPMI-Agent
- SSH-Agent
- TELNET-Agent
- JMX-Agent
- Berechnet
- Abhängiger Datenpunkt
- Skript
- Browser

Wenn der Datenpunkt/die Discovery-Regel vom Typ „Abhängiger Datenpunkt“ ist, muss auch sein Master-Datenpunkt einer der oben genannten Typen sein.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Aufgaben unter der Eigenschaft `taskids` enthält. Für jeden Datenpunkt und jede Low-Level-Discovery-Regel wird eine Aufgabe erstellt. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen `itemids`.

Beispiele

Erstellen einer Aufgabe

Erstellen Sie eine Aufgabe vom Typ „Jetzt ausführen“ für einen Datenpunkt und eine Low-Level-Discovery-Regel.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "task.create",
    "params": [
        {
            "type": 6,
            "request": {
                "itemid": "10092"
            }
        },
        {
            "type": 6,
            "request": {
                "itemid": "10093"
            }
        }
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "taskids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

Erstellen Sie eine Aufgabe vom Typ „Proxy-Konfiguration aktualisieren“ für zwei Proxys.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "task.create",
    "params": [
        {
            "type": 2,
            "request": {
                "proxyids": ["10459", "10460"]
            }
        }
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "taskids": [
            "1"
        ]
    },
    "id": 1
}
```

Erstellen Sie eine Aufgabe vom Typ „Diagnoseinformationen“.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "taskids": [
            "3"
        ]
    },
    "id": 1
}
```

Siehe auch

- Aufgabe
- ’Jetzt ausführen’-Anfrageobjekt
- ’Proxy-Konfiguration aktualisieren’-Anfrageobjekt
- ’Diagnoseinformationen’-Anfrageobjekt

Quelle

CTask::create() in *ui/include/classes/api/services/CTask.php*.

task.get

Beschreibung

`integer/array task.get(object parameters)`

Die Methode ermöglicht es Ihnen, Aufgaben anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Typ	Beschreibung
taskids	ID/array	Gibt nur Aufgaben mit den angegebenen IDs zurück.
output	query	Diese Parameter sind im Referenzkommentar beschrieben.
preservekeys	boolean	

Rückgabewerte

`(integer/array)` Gibt ein Array von Objekten zurück.

Beispiele

Aufgabe nach ID abrufen

Rufen Sie alle Daten zur Aufgabe mit der ID „1“ ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "task.get",
    "params": {
        "output": "extend",
        "taskids": "1"
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```


Siehe auch

- Aufgabe
- Anfrageobjekt „Jetzt ausführen“
- Anfrageobjekt „Proxy-Konfiguration aktualisieren“
- Anfrageobjekt „Diagnoseinformationen“

Quelle

CTask::get() in *ui/include/classes/api/services/CTask.php*.

Auslöser

Diese Klasse ist für die Arbeit mit Auslösern vorgesehen.

Objektreferenzen:

- Auslöser
- Auslöser-Tag

Verfügbare Methoden:

- trigger.create - neue Auslöser erstellen
- trigger.delete - Auslöser löschen
- trigger.get - Auslöser abrufen
- trigger.update - Auslöser aktualisieren

Trigger-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `trigger` API.

Auslöser

Das Auslöser-Objekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
triggerid	ID	ID des Auslösers. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
description	string	Name des Auslösers. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge

Eigenschaft	Typ	Beschreibung
expression	string	Auslöser-Ausdruck. Datenpunkte mit auf 5 (binär) oder 6 (JSON) gesetztem `value_type` können nicht in Auslöser-Ausdrücken verwendet werden. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
event_name	string	Vom Auslöser erzeugter Ereignisname.
opdata	string	Betriebsdaten.
comments	string	Zusätzliche Beschreibung des Auslösers.
error	string	Fehlertext, falls beim Aktualisieren des Zustands des Auslösers Probleme aufgetreten sind. Verhalten der Eigenschaft: - *schreibgeschützt*
flags	integer	Herkunft des Auslösers. Mögliche Werte: 0 - *(Standard)* ein einfacher Auslöser; 4 - ein aus einem Prototyp konvertierter Auslöser. Verhalten der Eigenschaft: - *schreibgeschützt*
lastchange	timestamp	Zeitpunkt, zu dem der Auslöser zuletzt seinen Zustand geändert hat. Verhalten der Eigenschaft: - *schreibgeschützt*
priority	integer	Schweregrad des Auslösers. Mögliche Werte: 0 - *(Standard)* nicht klassifiziert; 1 - Information; 2 - Warnung; 3 - durchschnittlich; 4 - hoch; 5 - Katastrophe.
state	integer	Status des Auslöser-Ausdrucks. Mögliche Werte: 0 - *(Standard)* der Auslöserstatus ist aktuell; 1 - der aktuelle Auslöserstatus ist unbekannt. Verhalten der Eigenschaft: - *schreibgeschützt*
status	integer	Gibt an, ob der Auslöser aktiviert oder deaktiviert ist. Mögliche Werte: 0 - *(Standard)* aktiviert; 1 - deaktiviert.
templateid	ID	ID des Auslösers der übergeordneten Vorlage. Verhalten der Eigenschaft: - *schreibgeschützt*
type	integer	Gibt an, ob der Auslöser mehrere Problemereignisse erzeugen kann. Mögliche Werte: 0 - *(Standard)* keine mehrfachen Ereignisse erzeugen; 1 - mehrere Ereignisse erzeugen.
url	string	Dem Auslöser zugeordnete URL.
url_name	string	Bezeichnung für die dem Auslöser zugeordnete URL.

Eigenschaft	Typ	Beschreibung
value	integer	Gibt an, ob sich der Auslöser im OK- oder Problemstatus befindet. Mögliche Werte: 0 - *(Standard)* OK; 1 - Problem. Verhalten der Eigenschaft: - *schreibgeschützt*
recovery_mode	integer	Modus für die Erzeugung von OK-Ereignissen. Mögliche Werte: 0 - *(Standard)* Ausdruck; 1 - Wiederherstellungsausdruck; 2 - Keine.
recovery_expression	string	Wiederherstellungsausdruck des Auslösers. Datenpunkte mit auf 5 (binär) oder 6 (JSON) gesetztem `value_type` können nicht in Wiederherstellungsausdrücken von Auslösern verwendet werden.
correlation_mode	integer	OK-Ereignis schließt. Mögliche Werte: 0 - *(Standard)* Alle Probleme; 1 - Alle Probleme, wenn Tag-Werte übereinstimmen.
correlation_tag	string	Tag für den Abgleich.
manual_close	integer	Manuelles Schließen erlauben. Mögliche Werte: 0 - *(Standard)* Nein; 1 - Ja.
uuid	string	Universell eindeutige Kennung, die verwendet wird, um importierte Auslöser mit bereits vorhandenen zu verknüpfen. Wird automatisch erzeugt, wenn sie nicht angegeben wird. Verhalten der Eigenschaft: - *unterstützt*, wenn der Auslöser zu einer Vorlage gehört

Auslöser-Tag

Das Auslöser-Tag-Objekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
tag	string	Name des Auslöser-Tags. Verhalten der Eigenschaft: - *erforderlich*
value	string	Wert des Auslöser-Tags. Verhalten der Eigenschaft: - *schreibgeschützt*.
object	integer	Typ des Objekts, von dem das Tag geerbt wurde. Mögliche Werte: 0 - Vorlage; 1 - Host; 3 - Datenpunkt. Verhalten der Eigenschaft: - *schreibgeschützt*.

Eigenschaft	Typ	Beschreibung
objectid	ID	ID des Objekts, von dem das Tag geerbt wurde. Verhalten der Eigenschaft: - *schreibgeschützt*.
automatic	integer	Typ eines Auslöser-Tags: Mögliche Werte: 0 - *(Standard)* vom Benutzer erstelltes Tag; 1 - von LLD erstelltes Tag;Verhalten der Eigenschaft: - *schreibgeschützt*.

trigger.create

Beschreibung

`object trigger.create(object/array triggers)`

Diese Methode ermöglicht es Ihnen, neue Auslöser zu erstellen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellende Auslöser.

Zusätzlich zu den Standard-Auslöser-Eigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
dependencies	array	Auslöser, von denen der Auslöser abhängig ist. Für die Auslöser darf nur die Eigenschaft `triggerid` definiert sein.
tags	array	Auslöser-Tags.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Auslöser unter der Eigenschaft `triggerids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Auslöser.

Beispiele

Erstellen eines Auslösers

Erstellen Sie zwei Auslöser, die jeweils von einem anderen Auslöser abhängen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.create",
    "params": [
        {
            "description": "Die Prozessorlast ist auf {HOST.NAME} zu hoch",
            "expression": "last(/Linux server/system.cpu.load[percpu,avg1])>5",
            "dependencies": [
                {
                    "triggerid": "17367"
                }
            ]
        },
        {
            "description": "Dienst wurde gestoppt",
```

```
            "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17369",
            "17370"
        ]
    },
    "id": 1
}
```

Quelle

CTrigger::create() in *ui/include/classes/api/services/CTrigger.php*.

trigger.delete

Beschreibung

`object trigger.delete(array triggerIds)`

Diese Methode ermöglicht es Ihnen, Auslöser zu löschen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Berechtigungen zum Aufruf der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der zu löschenden Auslöser.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Auslöser unter der Eigenschaft `triggerids` enthält.

Beispiele

Mehrere Auslöser löschen

Löschen Sie zwei Auslöser.

Anfrage:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "trigger.delete",
    "params": [
        "12002",
        "12003"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "12002",
            "12003"
        ]
    },
    "id": 1
}
```

Quelle

CTrigger::delete() in *ui/include/classes/api/services/CTrigger.php*.

trigger.get

Beschreibung

`integer/array trigger.get(object parameters)`

Die Methode ermöglicht es Ihnen, Auslöser anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Description
triggerids	ID/array	Gibt nur Auslöser mit den angegebenen IDs zurück.
groupids	ID/array	Gibt nur Auslöser zurück, die zu Hosts oder Vorlagen aus den angegebenen Hostgruppen oder Vorlagengruppen gehören.
templateids	ID/array	Gibt nur Auslöser zurück, die zu den angegebenen Vorlagen gehören.
hostids	ID/array	Gibt nur Auslöser zurück, die zu den angegebenen Hosts gehören.
itemids	ID/array	Gibt nur Auslöser zurück, die die angegebenen Datenpunkte enthalten.
maintenanceids	ID/array	Gibt nur Auslöser zurück, die von den angegebenen Wartungen betroffen sind.
functions	string/array	Gibt nur Auslöser zurück, die die angegebenen Funktionen verwenden. Eine Liste der unterstützten Funktionen finden Sie auf der Seite Unterstützte Funktionen.
group	string	Gibt nur Auslöser zurück, die zu Hosts oder Vorlagen aus der Hostgruppe oder Vorlagengruppe mit dem angegebenen Namen gehören.
host	string	Gibt nur Auslöser zurück, die zum Host mit dem angegebenen technischen Namen gehören.
inherited	boolean	Wenn auf `true` gesetzt, werden nur von einer Vorlage geerbte Auslöser zurückgegeben.
templated	boolean	Wenn auf `true` gesetzt, werden nur Auslöser zurückgegeben, die zu Vorlagen gehören.

Parameter	Type	Description
dependent	boolean	Wenn auf `true` gesetzt, werden nur Auslöser zurückgegeben, die Abhängigkeiten haben. Wenn auf `false` gesetzt, werden nur Auslöser zurückgegeben, die keine Abhängigkeiten haben.
monitored	flag	Gibt nur aktivierte Auslöser zurück, die zu überwachten Hosts gehören und ausschließlich aktivierte Datenpunkte enthalten.
active	flag	Gibt nur aktivierte Auslöser zurück, die zu überwachten Hosts gehören.
maintenance	boolean	Wenn auf `true` gesetzt, werden nur aktivierte Auslöser zurückgegeben, die zu Hosts im Wartungsmodus gehören.
withUnacknowledgedEvents	flag	Gibt nur Auslöser zurück, für die nicht bestätigte Ereignisse vorhanden sind.
withAcknowledgedEvents	flag	Gibt nur Auslöser zurück, für die alle Ereignisse bestätigt wurden.
withLastEventUnacknowledged	flag	Gibt nur Auslöser zurück, deren letztes Ereignis nicht bestätigt wurde.
skipDependent	flag	Überspringt Auslöser im Problemzustand, die von anderen Auslösern abhängig sind. Beachten Sie, dass die anderen Auslöser ignoriert werden, wenn sie deaktiviert sind, deaktivierte Datenpunkte haben oder zu Hosts mit deaktivierten Datenpunkten gehören.
lastChangeSince	timestamp	Gibt nur Auslöser zurück, deren Zustand nach dem angegebenen Zeitpunkt geändert wurde.
lastChangeTill	timestamp	Gibt nur Auslöser zurück, deren Zustand vor dem angegebenen Zeitpunkt geändert wurde.
only_true	flag	Gibt nur Auslöser zurück, die sich kürzlich in einem Problemzustand befunden haben (innerhalb von `ok_period`).
min_severity	integer	Gibt nur Auslöser zurück, deren Schweregrad größer oder gleich dem angegebenen Schweregrad ist.
evaltype	integer	Auswertungsmethode für Tags. Mögliche Werte: 0 - *(Standard)* Und/Oder; 2 - Oder.
tags	array	Gibt nur Auslöser mit den angegebenen Tags zurück. Format: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Ein leeres Array gibt alle Auslöser zurück. Mögliche Werte für Operatoren: 0 - *(Standard)* Enthält; 1 - Ist gleich; 2 - Enthält nicht; 3 - Ist nicht gleich; 4 - Ist vorhanden; 5 - Ist nicht vorhanden.
expandComment	flag	Erweitert Makros in der Beschreibung des Auslösers.
expandDescription	flag	Erweitert Makros im Namen des Auslösers.
expandExpression	flag	Erweitert Funktionen und Makros im Ausdruck des Auslösers.
selectHostGroups	query	Gibt die Hostgruppen, zu denen der Auslöser gehört, in der Eigenschaft `hostgroups` zurück.
selectHosts	query	Gibt die Hosts, zu denen der Auslöser gehört, in der Eigenschaft `hosts` zurück.
selectItems	query	Gibt die im Auslöser enthaltenen Datenpunkte in der Eigenschaft `items` zurück.
selectFunctions	query	Gibt die im Auslöser verwendeten Funktionen in der Eigenschaft `functions` zurück. Die Funktionsobjekte stellen die im Ausdruck des Auslösers verwendeten Funktionen dar und haben die folgenden Eigenschaften: `functionid` - `(ID)` ID der Funktion; `itemid` - `(ID)` ID des in der Funktion verwendeten Datenpunkts; `function` - `(string)` Name der Funktion; `parameter` - `(string)` An die Funktion übergebener Parameter. Der Abfrageparameter wird in der zurückgegebenen Zeichenfolge durch das Symbol $ ersetzt.

Parameter	Type	Description
selectDependencies	query	Gibt die Auslöser, von denen der Auslöser abhängig ist, in der Eigenschaft `dependencies` zurück.
selectDiscoveryData	query	Gibt eine Eigenschaft `discoveryData` mit den Daten des Auslöser-Discovery-Objekts zurück. Das Auslöser-Discovery-Objekt verknüpft einen erkannten Auslöser mit einem Auslöser-Prototyp, aus dem er erkannt wurde. Es hat die folgenden Eigenschaften: `parent_triggerid` - `(ID)` ID des Auslöser-Prototyps, aus dem der Auslöser erstellt wurde; `status` - `(int)` Discovery-Status des Auslösers: 0 - *(Standard)* Auslöser wurde erkannt, 1 - Auslöser wird nicht mehr erkannt; `ts_delete` - `(timestamp)` Zeitpunkt, zu dem ein nicht mehr erkannter Auslöser gelöscht wird; `ts_disable` - `(timestamp)` Zeitpunkt, zu dem ein nicht mehr erkannter Auslöser deaktiviert wird; `disable_source` - `(int)` Gibt an, ob der Auslöser durch eine LLD-Regel oder manuell deaktiviert wurde: 0 - *(Standard)* automatisch deaktiviert, 1 - durch eine LLD-Regel deaktiviert.
selectDiscoveryRule	query	Gibt die Low-Level-Discovery-Regel, durch die der Auslöser erstellt wurde, in der Eigenschaft `discoveryRule` zurück.
selectLastEvent	query	Gibt das letzte relevante Auslöserereignis in der Eigenschaft `lastEvent` zurück.
selectTags	query	Gibt die Tags des Auslösers in der Eigenschaft `tags` zurück.
inheritedTags	boolean	Gibt Auslöser zurück, die die angegebenen `tags` auch in der Vorlage, dem Host oder verknüpften Vorlagen haben. Mögliche Werte: `true` - Vorlage, Host oder verknüpfte Vorlagen müssen die angegebenen Tags ebenfalls haben; `false` - *(Standard)* Tags aus Vorlage, Host oder verknüpften Vorlagen werden ignoriert.
selectInheritedTags	query	Gibt eine Eigenschaft `inheritedTags` mit von Vorlagen, verknüpften Vorlagen und Hosts sowie von in Auslöser- oder Wiederherstellungsausdrücken referenzierten Datenpunkten geerbten Tags zurück.
selectTemplateGroups	query	Gibt die Vorlagengruppen, zu denen der Auslöser gehört, in der Eigenschaft `templategroups` zurück.
filter	object	Gibt nur Ergebnisse zurück, die exakt mit dem angegebenen Filter übereinstimmen. Akzeptiert ein Objekt, dessen Schlüssel Eigenschaftsnamen und dessen Werte entweder ein einzelner Wert oder ein Array von abzugleichenden Werten sind. Unterstützt keine Eigenschaften des Datentyps `text`. Unterstützt zusätzliche Eigenschaften: `host` - technischer Name des Hosts, zu dem der Auslöser gehört; `hostid` - ID des Hosts, zu dem der Auslöser gehört.
limitSelects	integer	Begrenzt die Anzahl der von Unterabfragen zurückgegebenen Datensätze. Gilt für die folgenden Unterabfragen: `selectHosts` - Ergebnisse werden nach `host` sortiert.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften (Sortierung). Mögliche Werte: `triggerid`, `description`, `status`, `priority`, `lastchange`, `hostname`.

Parameter	Type	Description
countOutput	boolean	Diese Parameter werden in der Referenzdokumentation beschrieben.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectTriggerDiscovery	query	Gibt das Discovery-Objekt des Auslösers in der Eigenschaft `triggerDiscovery` zurück. Die Auslöser-Discovery-Objekte verknüpfen den Auslöser mit einem Auslöser-Prototyp, aus dem er erstellt wurde. Diese Abfrage ist **veraltet**. Verwenden Sie stattdessen `selectDiscoveryData`.

Rückgabewerte

`(integer/array)` Gibt entweder Folgendes zurück:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Abrufen von Daten nach Auslöser-ID

Rufen Sie alle Daten und die im Auslöser „14062“ verwendeten Funktionen ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.get",
    "params": {
        "triggerids": "14062",
        "output": "extend",
        "selectFunctions": "extend"
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```

Auslöser im Problemzustand abrufen

Rufen Sie die ID, den Namen und den Schweregrad aller Auslöser im Problemzustand ab und sortieren Sie sie nach Schweregrad in absteigender Reihenfolge.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.get",
    "params": {
        "output": [
            "triggerid",
            "description",
            "priority"
        ],
        "filter": {
            "value": 1
        },
        "sortfield": "priority",
        "sortorder": "DESC"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "13907",
            "description": "Zabbix-Selbstüberwachungsprozesse < 100 % ausgelastet",
            "priority": "4"
        },
        {
            "triggerid": "13824",
            "description": "Zabbix-Discoverer-Prozesse mehr als 75 % ausgelastet",
            "priority": "3"
        }
    ],
    "id": 1
```

```
}
```

Abrufen eines bestimmten Auslösers mit Tags

Rufen Sie einen bestimmten Auslöser mit Tags ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.get",
    "params": {
        "output": [
            "triggerid",
            "description"
        ],
        "selectTags": "extend",
        "triggerids": [
            "17578"
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```

Siehe auch

- Discovery-Regel
- Datenpunkt
- Host
- Host-Gruppe
- Vorlagengruppe

Quelle

CTrigger::get() in *ui/include/classes/api/services/CTrigger.php*.

trigger.update

Beschreibung

`object trigger.update(object/array triggers)`

Diese Methode ermöglicht es Ihnen, vorhandene Auslöser zu aktualisieren.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu aktualisierende Auslöser-Eigenschaften.

Die Eigenschaft `triggerid` muss für jeden Auslöser definiert sein, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Zusätzlich zu den standardmäßigen Auslöser-Eigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
dependencies	array	Auslöser, von denen der Auslöser abhängig ist. Für die Auslöser darf nur die Eigenschaft `triggerid` definiert sein.
tags	array	Auslöser-Tags.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Auslöser unter der Eigenschaft `triggerids` enthält.

Beispiele

Einen Auslöser aktivieren

Aktivieren Sie einen Auslöser, d. h. setzen Sie seinen Status auf „0“.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.update",
    "params": {
        "triggerid": "13938",
        "status": 0
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938"
        ]
    },
    "id": 1
}
```

Ersetzen von Auslöser-Tags

Ersetzt Tags für einen Auslöser.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938"
        ]
    },
    "id": 1
}
```

Ersetzen von Abhängigkeiten

Ersetzen Sie Abhängigkeiten für den Auslöser.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.update",
    "params": {
        "triggerid": "22713",
        "dependencies": [
            {
                "triggerid": "22712"
            },
            {
                "triggerid": "22772"
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "22713"
        ]
    },
    "id": 1
}
```

Quelle

CTrigger::update() in *ui/include/classes/api/services/CTrigger.php*.

Auslöserprototyp

Diese Klasse ist für die Arbeit mit Auslöserprototypen vorgesehen.

Objektreferenzen:

- Auslöserprototyp
- Tag des Auslöserprototyps

Verfügbare Methoden:

- triggerprototype.create - neue Auslöserprototypen erstellen
- triggerprototype.delete - Auslöserprototypen löschen
- triggerprototype.get - Auslöserprototypen abrufen
- triggerprototype.update - Auslöserprototypen aktualisieren

Auslöser-Prototyp-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `triggerprototype`-API.

Auslöser-Prototyp

Das Auslöser-Prototyp-Objekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
triggerid	ID	ID des Auslöser-Prototyps. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
description	string	Name des Auslöser-Prototyps. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
expression	string	Auslöser-Ausdruck. Muss mindestens einen Datenpunkt-Prototyp enthalten. Datenpunkt-Prototypen mit auf 5 (binär) oder 6 (JSON) gesetztem `value_type` können nicht in Auslöser-Ausdrücken verwendet werden. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
event_name	string	Vom Auslöser erzeugter Ereignisname.
opdata	string	Betriebsdaten.
comments	string	Zusätzliche Kommentare zum Auslöser-Prototyp.
priority	integer	Schweregrad des Auslöser-Prototyps. Mögliche Werte: 0 - *(Standard)* nicht klassifiziert; 1 - Information; 2 - Warnung; 3 - durchschnittlich; 4 - hoch; 5 - Katastrophe.
status	integer	Gibt an, ob der Auslöser-Prototyp aktiviert oder deaktiviert ist. Mögliche Werte: 0 - *(Standard)* aktiviert; 1 - deaktiviert.
flags	integer	Herkunft des Auslöser-Prototyps. Mögliche Werte: 2 - ein Auslöser-Prototyp; 6 - ein entdeckter Auslöser-Prototyp. Verhalten der Eigenschaft: - *schreibgeschützt*

Eigenschaft	Typ	Beschreibung
templateid	ID	ID des übergeordneten Vorlagen-Auslöser-Prototyps. Verhalten der Eigenschaft: - *schreibgeschützt*
type	integer	Gibt an, ob der Auslöser-Prototyp mehrere Problemereignisse erzeugen kann. Mögliche Werte: 0 - *(Standard)* keine mehreren Ereignisse erzeugen; 1 - mehrere Ereignisse erzeugen.
url	string	Dem Auslöser-Prototyp zugeordnete URL.
url_name	string	Bezeichnung für die dem Auslöser-Prototyp zugeordnete URL.
recovery_mode	integer	Modus zur Erzeugung von OK-Ereignissen. Mögliche Werte: 0 - *(Standard)* Ausdruck; 1 - Wiederherstellungsausdruck; 2 - Keine.
recovery_expression	string	Wiederherstellungsausdruck des Auslösers. Muss mindestens einen Datenpunkt-Prototyp enthalten. Datenpunkt-Prototypen mit auf 5 (binär) oder 6 (JSON) gesetztem `value_type` können nicht in Wiederherstellungsausdrücken des Auslösers verwendet werden.
correlation_mode	integer	Schließt OK-Ereignis. Mögliche Werte: 0 - *(Standard)* Alle Probleme; 1 - Alle Probleme, wenn Tag-Werte übereinstimmen.
correlation_tag	string	Tag für den Abgleich.
manual_close	integer	Manuelles Schließen erlauben. Mögliche Werte: 0 - *(Standard)* Nein; 1 - Ja.
discover	integer	Erkennungsstatus des Auslöser-Prototyps. Mögliche Werte: 0 - *(Standard)* neue Auslöser werden erkannt; 1 - neue Auslöser werden nicht erkannt und vorhandene Auslöser werden als verloren markiert.
uuid	string	Universell eindeutige Kennung, die verwendet wird, um importierte Auslöser-Prototypen mit bereits vorhandenen zu verknüpfen. Wird automatisch erzeugt, wenn sie nicht angegeben wird. Verhalten der Eigenschaft: - *unterstützt*, wenn der Auslöser-Prototyp zu einer Vorlage gehört

Tag des Auslöser-Prototyps

Das Tag-Objekt des Auslöser-Prototyps hat die folgenden Eigenschaften.

Property	Type	Description
tag	string	Name des Tags des Auslöser-Prototyps. Property behavior: - *erforderlich*

Property	Type	Description
value	string	Wert des Tags des Auslöser-Prototyps. Property behavior: - *schreibgeschützt*.
object	integer	Typ des Objekts, von dem das Tag geerbt wurde. Mögliche Werte: 0 - Vorlage; 1 - Host; 3 - Datenpunkt; 4 - Datenpunkt-Prototyp. Property behavior: - *schreibgeschützt*.
objectid	ID	ID des Objekts, von dem das Tag geerbt wurde. Property behavior: - *schreibgeschützt*.

auslöserprototyp.delete

Beschreibung

`object triggerprototype.delete(array triggerPrototypeIds)`

Diese Methode ermöglicht es Ihnen, Auslöservorlagen zu löschen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der zu löschenden Auslöser-Prototypen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Auslöser-Prototypen in der Eigenschaft `triggerids` enthält.

Beispiele

Mehrere Auslöser-Prototypen löschen

Löschen Sie zwei Auslöser-Prototypen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.delete",
    "params": [
        "12002",
        "12003"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "12002",
            "12003"
```

```
        ]
    },
    "id": 1
}
```

Quelle

CTriggerPrototype::delete() in *ui/include/classes/api/services/CTriggerPrototype.php*.

triggerprototype.create

Beschreibung

`object triggerprototype.create(object/array triggerPrototypes)`

Diese Methode ermöglicht es Ihnen, neue Auslöservorlagen zu erstellen.

Note:
Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellende Auslöser-Prototypen.

Zusätzlich zu den Standard-Eigenschaften von Auslöser-Prototypen akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
dependencies	array	Auslöser und Auslöser-Prototypen, von denen der Auslöser-Prototyp abhängig ist. Für die Auslöser darf nur die Eigenschaft `triggerid` definiert sein.
tags	array	Tags von Auslöser-Prototypen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Auslöser-Prototypen in der Eigenschaft `triggerids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Auslöser-Prototypen.

Beispiele

Erstellen eines Auslöser-Prototyps

Erstellen Sie einen Auslöser-Prototyp, um zu erkennen, wenn ein Dateisystem über weniger als 20 % freien Festplattenspeicher verfügt.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.create",
    "params": {
        "description": "Freier Festplattenspeicher ist auf Volume {#FSNAME} kleiner als 20%",
        "expression": "last(/Zabbix server/vfs.fs.size[{#FSNAME},pfree])<20",
        "tags": [
            {
                "tag": "volume",
                "value": "{#FSNAME}"
            },
            {
                "tag": "type",
                "value": "{#FSTYPE}"
            }
        ]
    },
```

```
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17372"
        ]
    },
    "id": 1
}
```

Quelle

CTriggerPrototype::create() in *ui/include/classes/api/services/CTriggerPrototype.php*.

triggerprototype.get

Beschreibung

`integer/array triggerprototype.get(object parameters)`

Die Methode ermöglicht es Ihnen, Auslöser-Prototypen anhand der angegebenen Parameter abzurufen.

Note:
Diese Methode ist für Benutzer aller Typen verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
active	flag	Gibt nur aktivierte Auslöser-Prototypen zurück, die zu überwachten Hosts gehören.
discoveryids	ID/array	Gibt nur Auslöser-Prototypen zurück, die zu den angegebenen LLD-Regeln gehören.
functions	string/array	Gibt nur Auslöser zurück, die die angegebenen Funktionen verwenden. Eine Liste der unterstützten Funktionen finden Sie auf der Seite Supported functions.
group	string	Gibt nur Auslöser-Prototypen zurück, die zu Hosts oder Vorlagen aus den Host-Gruppen oder Vorlagen-Gruppen mit dem angegebenen Namen gehören.
groupids	ID/array	Gibt nur Auslöser-Prototypen zurück, die zu Hosts oder Vorlagen aus den angegebenen Host-Gruppen oder Vorlagen-Gruppen gehören.
host	string	Gibt nur Auslöser-Prototypen zurück, die zu Hosts mit dem angegebenen Namen gehören.
hostids	ID/array	Gibt nur Auslöser-Prototypen zurück, die zu den angegebenen Hosts gehören.
inherited	boolean	Wenn auf `true` gesetzt, werden nur Auslöser-Prototypen zurückgegeben, die von einer Vorlage geerbt wurden.
maintenance	boolean	Wenn auf `true` gesetzt, werden nur aktivierte Auslöser-Prototypen zurückgegeben, die zu Hosts in Wartung gehören.
min_severity	integer	Gibt nur Auslöser-Prototypen mit einem Schweregrad zurück, der größer oder gleich dem angegebenen Schweregrad ist.
monitored	flag	Gibt nur aktivierte Auslöser-Prototypen zurück, die zu überwachten Hosts gehören und nur aktivierte Datenpunkte enthalten.
templated	boolean	Wenn auf `true` gesetzt, werden nur Auslöser-Prototypen zurückgegeben, die zu Vorlagen gehören.

Parameter	Type	Beschreibung
templateids	ID/array	Gibt nur Auslöser-Prototypen zurück, die zu den angegebenen Vorlagen gehören.
triggerids	ID/array	Gibt nur Auslöser-Prototypen mit den angegebenen IDs zurück.
expandExpression	flag	Erweitert Funktionen und Makros im Auslöser-Ausdruck.
selectDependencies	query	Gibt Auslöser-Prototypen und Auslöser, von denen der Auslöser-Prototyp abhängt, in der Eigenschaft `dependencies` zurück.
selectDiscoveryData	query	Gibt eine Eigenschaft `discoveryData` mit den Objektdaten der Auslöser-Prototyp-Erkennung zurück. Das Erkennungsobjekt des Auslöser-Prototyps verknüpft einen erkannten Auslöser-Prototypen mit einem Auslöser-Prototypen, von dem er erkannt wurde. Es hat die folgenden Eigenschaften: `parent_triggerid` - `(ID)` ID des Auslöser-Prototyps, aus dem der Auslöser-Prototyp erstellt wurde; `status` - `(int)` Erkennungsstatus des Auslöser-Prototyps: 0 - *(default)* Auslöser-Prototyp ist erkannt, 1 - Auslöser-Prototyp wird nicht mehr erkannt; `ts_delete` - `(timestamp)` Zeitpunkt, zu dem ein Auslöser-Prototyp, der nicht mehr erkannt wird, gelöscht wird; `ts_disable` - `(timestamp)` Zeitpunkt, zu dem ein Auslöser-Prototyp, der nicht mehr erkannt wird, deaktiviert wird; `disable_source` - `(int)` Kennzeichen dafür, ob der Auslöser-Prototyp durch eine LLD-Regel oder manuell deaktiviert wurde: 0 - *(default)* automatisch deaktiviert, 1 - durch eine LLD-Regel deaktiviert.
selectDiscoveryRule	query	Gibt die LLD-Regel, zu der der Auslöser-Prototyp gehört, in der Eigenschaft `discoveryRule` zurück.
selectDiscoveryRulePrototype	query	Gibt eine Eigenschaft `discoveryRulePrototype` mit dem übergeordneten LLD-Regelprototyp zurück, zu dem der Auslöser-Prototyp gehört.
selectFunctions	query	Gibt die im Auslöser-Prototyp verwendeten Funktionen in der Eigenschaft `functions` zurück. Die Funktionsobjekte stellen die im Auslöser-Ausdruck verwendeten Funktionen dar und haben die folgenden Eigenschaften: `functionid` - `(ID)` ID der Funktion; `itemid` - `(ID)` ID des in der Funktion verwendeten Datenpunkts; `function` - `(string)` Name der Funktion; `parameter` - `(string)` an die Funktion übergebener Parameter. Der Abfrageparameter wird in der zurückgegebenen Zeichenfolge durch das Symbol $ ersetzt.
selectHostGroups	query	Gibt die Host-Gruppen, zu denen der Auslöser-Prototyp gehört, in der Eigenschaft `hostgroups` zurück.
selectHosts	query	Gibt die Hosts, zu denen der Auslöser-Prototyp gehört, in der Eigenschaft `hosts` zurück.
selectInheritedTags	query	Gibt eine Eigenschaft `inheritedTags` mit Tags zurück, die von Vorlagen, verknüpften Vorlagen und Hosts sowie von Datenpunkt-Prototypen geerbt wurden, auf die in Auslöser- oder Wiederherstellungsausdrücken verwiesen wird.
selectItems	query	Gibt Datenpunkte und Datenpunkt-Prototypen, die vom Auslöser-Prototyp verwendet werden, in der Eigenschaft `items` zurück.
selectTags	query	Gibt die Tags des Auslöser-Prototyps in der Eigenschaft `tags` zurück.
selectTemplateGroups	query	Gibt die Vorlagen-Gruppen, zu denen der Auslöser-Prototyp gehört, in der Eigenschaft `templategroups` zurück.

Parameter	Type	Beschreibung
filter	object	Gibt nur die Ergebnisse zurück, die exakt dem angegebenen Filter entsprechen. Akzeptiert ein Objekt, bei dem die Schlüssel Eigenschaftsnamen sind und die Werte entweder ein einzelner Wert oder ein Array von Werten sind, mit denen abgeglichen werden soll. Unterstützt keine Eigenschaften des Datentyps `text`. Unterstützt zusätzliche Eigenschaften: `host` - technischer Name des Hosts, zu dem der Auslöser-Prototyp gehört; `hostid` - ID des Hosts, zu dem der Auslöser-Prototyp gehört.
limitSelects	integer	Begrenzt die Anzahl der von Unterabfragen zurückgegebenen Datensätze. Gilt für die folgenden Unterabfragen: `selectHosts` - Ergebnisse werden nach `host` sortiert.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `triggerid`, `description`, `status`, `priority`, `discovered`.
countOutput	boolean	Diese Parameter sind in der Referenzkommentierung beschrieben.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Gibt entweder Folgendes zurück:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Auslöser-Prototypen aus einer LLD-Regel abrufen

Rufen Sie alle Auslöser-Prototypen und ihre Funktionen aus einer LLD-Regel ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.get",
    "params": {
        "output": "extend",
        "selectFunctions": "extend",
        "discoveryids": "22450"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
```

```
        {
            "triggerid": "13272",
            "expression": "{12598}<20",
            "description": "Freie Inodes sind auf Volume {#FSNAME} kleiner als 20%",
            "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": "Freier Festplattenspeicher ist auf Volume {#FSNAME} kleiner als 20%",
            "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
}
```

Abrufen eines bestimmten Auslöser-Prototyps mit Tags

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.get",
    "params": {
        "output": [
            "triggerid",
            "description"
        ],
        "selectTags": "extend",
        "triggerids": [
            "17373"
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "17373",
            "description": "Freier Festplattenspeicher ist auf Volume {#FSNAME} kleiner als 20%",
            "tags": [
                {
                    "tag": "volume",
                    "value": "{#FSNAME}"
                },
                {
                    "tag": "type",
                    "value": "{#FSTYPE}"
                }
            ]
        }
    ],
    "id": 1
}
```

Siehe auch

- Discovery-Regel
- Datenpunkt
- Host
- Host-Gruppe
- Vorlagen-Gruppe

Quelle

CTriggerPrototype::get() in *ui/include/classes/api/services/CTriggerPrototype.php*.

triggerprototype.update

Beschreibung

`object triggerprototype.update(object/array triggerPrototypes)`

Diese Methode ermöglicht es Ihnen, vorhandene Auslöservorlagen zu aktualisieren.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu aktualisierende Eigenschaften des Auslöser-Prototyps.

Die Eigenschaft `triggerid` muss für jeden Auslöser-Prototyp definiert sein, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Zusätzlich zu den Standard-Eigenschaften des Auslöser-Prototyps akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
dependencies	array	Auslöser und Auslöser-Prototypen, von denen der Auslöser-Prototyp abhängig ist. Für die Auslöser darf nur die Eigenschaft `triggerid` definiert sein.
tags	array	Tags des Auslöser-Prototyps.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Auslöser-Prototypen in der Eigenschaft `triggerids` enthält.

Beispiele

Aktivieren eines Auslöser-Prototyps

Aktivieren Sie einen Auslöser-Prototypen, d. h. setzen Sie seinen Status auf „0“.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.update",
    "params": {
        "triggerid": "13938",
        "status": 0
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938"
        ]
    },
    "id": 1
}
```

Ersetzen von Tags eines Auslöser-Prototyps

Ersetzen Sie die Tags für einen Auslöser-Prototyp.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.update",
    "params": {
        "triggerid": "17373",
        "tags": [
            {
                "tag": "volume",
                "value": "{#FSNAME}"
            },
            {
                "tag": "type",
                "value": "{#FSTYPE}"
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17373"
        ]
    },
    "id": 1
}
```

Quelle

CTriggerPrototype::update() in *ui/include/classes/api/services/CTriggerPrototype.php*.

Authentifizierung

Diese Klasse wurde für die Arbeit mit Authentifizierungseinstellungen konzipiert.

Objektreferenzen:

- Authentifizierung

Verfügbare Methoden:

- authentication.get - Authentifizierungseinstellungen abrufen
- authentication.update - Authentifizierungseinstellungen aktualisieren

Authentifizierungs-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `authentication` API.

Authentifizierung

Das Authentifizierungsobjekt verfügt über die folgenden Eigenschaften.

Property	Type	Description
authentication_type	integer	Standardauthentifizierung. Mögliche Werte: 0 - *(default)* Intern; 1 - LDAP.

Property	Type	Description
http_auth_enabled	integer	HTTP-Authentifizierung. Mögliche Werte: 0 - *(default)* Deaktiviert; 1 - Aktiviert. Verhalten der Eigenschaft: - *unterstützt*, wenn `$ZBX_FEATURE_FLAGS['http_auth_enabled']` in der Frontend-Konfigurationsdatei auf `true` gesetzt ist.
http_login_form	integer	Standardanmeldeformular. Mögliche Werte: 0 - *(default)* Zabbix-Anmeldeformular; 1 - HTTP-Anmeldeformular. Verhalten der Eigenschaft: - *unterstützt*, wenn `$ZBX_FEATURE_FLAGS['http_auth_enabled']` in der Frontend-Konfigurationsdatei auf `true` gesetzt ist.
http_strip_domains	string	Zu entfernender Domänenname. Verhalten der Eigenschaft: - *unterstützt*, wenn `$ZBX_FEATURE_FLAGS['http_auth_enabled']` in der Frontend-Konfigurationsdatei auf `true` gesetzt ist.
http_case_sensitive	integer	Bei der HTTP-Anmeldung Groß-/Kleinschreibung beachten. Mögliche Werte: 0 - Deaktiviert; 1 - *(default)* Aktiviert. Verhalten der Eigenschaft: - *unterstützt*, wenn `$ZBX_FEATURE_FLAGS['http_auth_enabled']` in der Frontend-Konfigurationsdatei auf `true` gesetzt ist.
ldap_auth_enabled	integer	LDAP-Authentifizierung. Mögliche Werte: 0 - *(default)* Deaktiviert; 1 - Aktiviert.
ldap_case_sensitive	integer	Bei der LDAP-Anmeldung Groß-/Kleinschreibung beachten. Mögliche Werte: 0 - Deaktiviert; 1 - *(default)* Aktiviert.
ldap_userdirectoryid	ID	ID des Standardbenutzerverzeichnisses für die LDAP-Authentifizierung. Wird für Benutzergruppen verwendet, bei denen `gui_access` auf LDAP oder Systemstandard gesetzt ist. Verhalten der Eigenschaft: - *erforderlich*, wenn `ldap_auth_enabled` auf „Enabled“ gesetzt ist
saml_auth_enabled	integer	SAML-Authentifizierung. Mögliche Werte: 0 - *(default)* Deaktiviert; 1 - Aktiviert.
saml_case_sensitive	integer	Bei der SAML-Anmeldung Groß-/Kleinschreibung beachten. Mögliche Werte: 0 - Deaktiviert; 1 - *(default)* Aktiviert.

Property	Type	Description
passwd_min_length	integer	Mindestlänge des Passworts. Mögliche Werte liegen zwischen 1 und 70. Standard: 8.
passwd_check_rules	integer	Regeln für die Passwortprüfung. Mögliche Bitmaskenwerte: 0 - Passwortlänge prüfen; 1 - Prüfen, ob das Passwort Groß- und Kleinbuchstaben des lateinischen Alphabets verwendet; 2 - Prüfen, ob das Passwort Ziffern verwendet; 4 - Prüfen, ob das Passwort Sonderzeichen verwendet; 8 - *(default)* Prüfen, ob das Passwort nicht in der Liste häufig verwendeter Passwörter enthalten ist und keine Ableitungen des Wortes „Zabbix“ oder des Vor- oder Nachnamens bzw. Benutzernamens des Benutzers enthält. Dies ist ein Bitmaskenfeld; jede Summe möglicher Bitmaskenwerte ist zulässig (beispielsweise 15 zur Prüfung aller Regeln).
ldap_jit_status	integer	Status der LDAP-Bereitstellung. Mögliche Werte: 0 - Für konfigurierte LDAP-IdPs deaktiviert; 1 - Für konfigurierte LDAP-IdPs aktiviert.
saml_jit_status	integer	Status der SAML-Bereitstellung. Mögliche Werte: 0 - Für konfigurierte SAML-IdPs deaktiviert; 1 - Für konfigurierte SAML-IdPs aktiviert.
jit_provision_interval	string	Zeitintervall zwischen JIT-Bereitstellungsanforderungen für den angemeldeten Benutzer. Akzeptiert eine Anzahl von Sekunden (z. B. 3600) oder einen Wert mit einem Zeitsuffix (z. B. `3600s`, `60m`, `1h`, `1d`, `1M`, `1y`), einschließlich der Unterstützung für Monate und Jahre. Mindestwert: `1h`. Standard: `1h`. Nur für die LDAP-Bereitstellung verfügbar.
disabled_usrgrpid	ID	ID der Benutzergruppe, der der aus der Bereitstellung entfernte Benutzer zugewiesen wird. Die Benutzergruppe muss deaktiviert sein und kann nach der Konfiguration weder aktiviert noch gelöscht werden. Verhalten der Eigenschaft: - *erforderlich*, wenn `ldap_jit_status` auf „Enabled for configured LDAP IdPs“ oder `saml_jit_status` auf „Enabled for configured SAML IdPs“ gesetzt ist
mfa_status	integer	Multi-Faktor-Authentifizierung. Mögliche Werte: 0 - Deaktiviert (für alle konfigurierten MFA-Methoden); 1 - Aktiviert (für alle konfigurierten MFA-Methoden).
mfaid	ID	Standard-MFA-Methode für Benutzergruppen mit aktivierter MFA. Verhalten der Eigenschaft: - *erforderlich*, wenn `mfa_status` auf „Enabled“ gesetzt ist

authentication.get

Beschreibung

`object authentication.get(object parameters)`

Die Methode ermöglicht es Ihnen, ein Authentifizierungsobjekt anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt nur einen Parameter.

Parameter	Typ	Beschreibung
output	query	Dieser Parameter wird im Referenzkommentar beschrieben.

Rückgabewerte

`(object)` Gibt ein Authentifizierungsobjekt zurück.

Beispiele

Request:

```
{
    "jsonrpc": "2.0",
    "method": "authentication.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```

Quelle

CAuthentication::get() in *ui/include/classes/api/services/CAuthentication.php*.

authentication.update

Beschreibung

`object authentication.update(object authentication)`

Diese Methode ermöglicht es Ihnen, vorhandene Authentifizierungseinstellungen zu aktualisieren.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Siehe User roles für weitere Informationen.

Parameter

`(object)` Authentifizierungseigenschaften wird aktualisiert.

Rückgabewerte

`(array)` Gibt ein Array mit den Namen der aktualisierten Parameter zurück.

Beispiele

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "authentication.update",
    "params": {
        "http_auth_enabled": 1,
        "http_case_sensitive": 0,
        "http_login_form": 1
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        "http_auth_enabled",
        "http_case_sensitive",
        "http_login_form"
    ],
    "id": 1
}
```

Quelle

CAuthentication::update() in *ui/include/classes/api/services/CAuthentication.php*.

Autoregistrierung

Diese Klasse ist für die Arbeit mit der Autoregistrierung konzipiert.

Objektreferenzen:

- Autoregistrierung

Verfügbare Methoden:

- autoregistration.get - Autoregistrierung abrufen
- autoregistration.update - Autoregistrierung aktualisieren

Autoregistrierungsobjekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `autoregistration` API.

Autoregistrierung

Das Autoregistrierungsobjekt hat die folgenden Eigenschaften.

Eigenschaft	Type	Beschreibung
tls_accept	integer	Typ der zulässigen eingehenden Verbindungen für die Autoregistrierung. Mögliche Werte: 1 - unverschlüsselte Verbindungen zulassen; 2 - TLS mit PSK zulassen; 3 - sowohl unverschlüsselte Verbindungen als auch TLS-Verbindungen mit PSK zulassen.
tls_psk_identity	string	PSK-Identität; darf nur mit genau einem PSK verknüpft sein (über autoregistration, hosts und proxies hinweg). Nehmen Sie keine sensiblen Informationen in die PSK-Identität auf, da sie unverschlüsselt über das Netzwerk gesendet wird, um dem Empfänger mitzuteilen, welcher PSK verwendet werden soll. Property behavior: - *write-only*
tls_psk	string	Vorab geteilter Schlüssel (PSK); muss aus mindestens 32 hexadezimalen Ziffern bestehen. Property behavior: - *write-only*

autoregistration.get

Beschreibung

`object autoregistration.get(object parameters)`

Die Methode ermöglicht es Ihnen, das Autoregistrierungsobjekt anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt nur einen Parameter.

Parameter	Typ	Beschreibung
output	query	Dieser Parameter wird im Referenzkommentar beschrieben.

Rückgabewerte

`(object)` Gibt das Autoregistrierungsobjekt zurück.

Beispiele

Anfrage:

```
{
   "jsonrpc": "2.0",
    "method": "autoregistration.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tls_accept": "3"
    },
    "id": 1
}
```

Quelle

CAutoregistration::get() in *ui/include/classes/api/services/CAutoregistration.php*.

autoregistration.update

Beschreibung

`object autoregistration.update(object autoregistration)`

Diese Methode ermöglicht es Ihnen, eine vorhandene automatische Registrierung zu aktualisieren.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Autoregistrierungseigenschaften die aktualisiert werden sollen.

Rückgabewerte

`(boolean )` Gibt bei erfolgreicher Aktualisierung den Boolean-Wert true als Ergebnis zurück.

Beispiele

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "autoregistration.update",
    "params": {
        "tls_accept": "3",
        "tls_psk_identity": "PSK 001",
        "tls_psk": "11111595725ac58dd977beef14b97461a7c1045b9a1c923453302c5473193478"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```

Quelle

CAutoregistration::update() in *ui/include/classes/api/services/CAutoregistration.php*.

Benutzer

Diese Klasse ist für die Arbeit mit Benutzern vorgesehen.

Objektreferenzen:

- Benutzer
- Medien

Verfügbare Methoden:

- user.checkauthentication - Benutzersitzungen prüfen und verlängern
- user.create - neue Benutzer erstellen
- user.delete - Benutzer löschen
- user.get - Benutzer abrufen
- user.login - bei der API anmelden
- user.logout - von der API abmelden
- user.provision - LDAP-Benutzer bereitstellen
- user.resettotp - TOTP-Geheimnisse von Benutzern zurücksetzen
- user.unblock - Benutzer entsperren
- user.update - Benutzer aktualisieren

Benutzerobjekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `user` API.

Benutzer

Das Benutzerobjekt verfügt über die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
userid	ID	ID des Benutzers. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
username	string	Name des Benutzers. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge - *schreibgeschützt* für bereitgestellte Benutzer, wenn der Benutzer mit einem Benutzerverzeichnis verknüpft ist (`userdirectoryid` ist auf einen gültigen Wert ungleich ”0” gesetzt), der Bereitstellungsstatus des Benutzerverzeichnisses aktiviert ist (`provision_status` des Benutzerverzeichnisobjekts ist auf ”1” gesetzt) und der Authentifizierungsstatus für die LDAP- oder SAML-Bereitstellung aktiviert ist (`ldap_jit_status` des Authentifizierungsobjekts ist auf ”Enabled for configured LDAP IdPs” gesetzt oder `saml_jit_status` des Authentifizierungsobjekts ist auf ”Enabled for configured SAML IdPs” gesetzt)
passwd	string	Passwort des Benutzers. Der Wert dieses Parameters kann eine leere Zeichenfolge sein, wenn der Benutzer mit einem Benutzerverzeichnis verknüpft ist. Verhalten der Eigenschaft: - *nur schreiben*
roleid	ID	ID der Rolle des Benutzers. Beachten Sie, dass sich Benutzer ohne Rolle nur über die LDAP-- oder SAML-Authentifizierung bei Zabbix anmelden können, sofern ihre LDAP-/SAML-Informationen mit den in Zabbix konfigurierten Zuordnungen von Benutzergruppen übereinstimmen.
attempt_clock	timestamp	Zeitpunkt des letzten fehlgeschlagenen Anmeldeversuchs. Verhalten der Eigenschaft: - *schreibgeschützt*
attempt_failed	integer	Anzahl der kürzlich fehlgeschlagenen Anmeldeversuche. Verhalten der Eigenschaft: - *schreibgeschützt*

Eigenschaft	Typ	Beschreibung
attempt_ip	string	IP-Adresse, von der der letzte fehlgeschlagene Anmeldeversuch ausging. Verhalten der Eigenschaft: - *schreibgeschützt*
autologin	integer	Gibt an, ob die automatische Anmeldung aktiviert werden soll. Mögliche Werte: 0 - *(Standard)* automatische Anmeldung deaktiviert; 1 - automatische Anmeldung aktiviert.
autologout	string	Lebensdauer der Benutzersitzung. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300) oder einen Wert mit einem Zeitsuffix (z. B. 30s, 1m, 2h). Bei Einstellung auf 0s läuft die Sitzung nie ab. Standard: 15m.
default_maintenance_period	string	Standardwartungszeitraum für eine über eine Verknüpfung erstellte Wartung. Akzeptiert Sekunden oder eine Zeiteinheit mit Suffix (300s, 5m, 1h, 1d, 1w). Der angegebene Wert wird auf ganze Minuten abgerundet. Standard: 1h.
lang	string	Sprachcode der Sprache des Benutzers, z. B. en_US. Standard: default - Systemstandard.
name	string	Name des Benutzers.
provisioned	integer	Gibt an, ob der Benutzer bereitgestellt wurde. Mögliche Werte: 0 - nicht bereitgestellt; 1 - bereitgestellt. Verhalten der Eigenschaft: - *schreibgeschützt*
refresh	string	Automatisches Aktualisierungsintervall. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300) oder einen Wert mit einem Zeitsuffix (z. B. 30s, 1m, 1h). Standard: 30s.
rows_per_page	integer	Anzahl der Objektzeilen, die pro Seite angezeigt werden. Standard: 50.
surname	string	Nachname des Benutzers.
theme	string	Theme des Benutzers. Mögliche Werte: default - *(Standard)* Systemstandard; blue-theme - Blau; blue-classic-theme - Blau (klassisch); dark-theme - Dunkel; dark-blue-theme - Dunkelblau; dark-classic-theme - Dunkel (klassisch); hc-light - Helles Design mit hohem Kontrast; hc-dark - Dunkles Design mit hohem Kontrast.
ts_provisioned	timestamp	Zeitpunkt, zu dem der letzte Bereitstellungsvorgang durchgeführt wurde. Verhalten der Eigenschaft: - *schreibgeschützt* - *unterstützt* für Benutzer vom Typ *Super admin*
url	string	URL der Seite, auf die der Benutzer nach der Anmeldung weitergeleitet wird.

Eigenschaft	Typ	Beschreibung
userdirectoryid	ID	ID des Benutzerverzeichnisses, mit dem der Benutzer verknüpft ist. Wird für die Bereitstellung (Erstellung oder Aktualisierung) sowie für die Anmeldung eines Benutzers verwendet, der mit einem Benutzerverzeichnis verknüpft ist. Bei Anmeldevorgängen hat der Wert dieser Eigenschaft Vorrang vor der Eigenschaft `userdirectoryid` der Benutzergruppen, denen der Benutzer angehört. Standard: 0. Verhalten der Eigenschaft: - *schreibgeschützt* - *unterstützt* für Benutzer vom Typ *Super admin*
timezone	string	Zeitzone des Benutzers, z. B. `Europe/London`, `UTC`. Standard: `default` - Systemstandard. Eine vollständige Liste der unterstützten Zeitzonen finden Sie in der PHP-Dokumentation.

Medien

Das Medienobjekt verfügt über die folgenden Eigenschaften.

Property	Type	Description
mediaid	ID	ID des Mediums des Benutzers. Verhalten der Eigenschaft: - *schreibgeschützt*
mediatypeid	ID	ID des Medientyps, der vom Medium des Benutzers verwendet wird. Verhalten der Eigenschaft: - *erforderlich*
sendto	string/array	Adresse, Benutzername oder sonstige Kennung des Empfängers. Wenn `type` des Medientyps auf „Email“ oder „Push“ gesetzt ist, werden die Werte als Array dargestellt. Für „Push“ kann das Array entweder einen Platzhalter (`*`) oder eine oder mehrere IDs mobiler Geräte enthalten. Wenn `type` des Medientyps auf „Script“, „SMS“ oder „Webhook“ gesetzt ist, wird der Wert als Zeichenfolge dargestellt. Verhalten der Eigenschaft: - *erforderlich*
active	integer	Gibt an, ob das Medium aktiviert ist. Mögliche Werte: 0 - *(Standard)* aktiviert; 1 - deaktiviert.

Property	Type	Description
severity	integer	Schweregrade von Auslösern, über die Benachrichtigungen gesendet werden sollen. Mögliche Bitmap-Werte: 1 - Nicht klassifiziert; 2 - Information; 4 - Warnung; 8 - Durchschnittlich; 16 - Hoch; 32 - Katastrophe. Dies ist ein Bitmaskenfeld; jede Summe möglicher Bitmap-Werte ist zulässig (zum Beispiel 48 für „Durchschnittlich“, „Hoch“ und „Katastrophe“). Standard: 63.
period	string	Zeit, zu der die Benachrichtigungen als Zeitperiode oder durch Semikolons getrennte Benutzermakros gesendet werden können. Standard: 1-7,00:00-24:00.
provisioned	integer	Gibt an, ob der Benutzer bereitgestellt wurde. Mögliche Werte: 0 - nicht bereitgestellt; 1 - bereitgestellt.
userdirectory_mediaid	ID	ID der Medienzuordnung im Benutzerverzeichnis für bereitgestellte Medien. Verhalten der Eigenschaft: - *schreibgeschützt* - *unterstützt* für Benutzer vom Typ *Super admin*

user.checkAuthentication

Beschreibung

`object user.checkAuthentication`

Diese Methode prüft und verlängert die Benutzersitzung.

Attention:
Der Aufruf der Methode `user.checkAuthentication` mit dem Parameter `sessionid` verlängert standardmäßig die Benutzersitzung.

Parameter

Die Methode akzeptiert die folgenden Parameter.

Parameter	Type	Beschreibung
extend	boolean	Ob die Benutzersitzung verlängert werden soll. Standardwert: ”true”. Wenn der Wert auf ”false” gesetzt wird, können Sie die Benutzersitzung prüfen, ohne sie zu verlängern.
sessionid	string	Parameterverhalten: - *supported* wenn `sessionid` gesetzt ist Benutzer-Authentifizierungstoken. Parameterverhalten: - *required* wenn `token` nicht gesetzt ist

Parameter	Type	Beschreibung
token	string	Benutzer-API-Token. Parameterverhalten: - *required* wenn `sessionid` nicht gesetzt ist

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das Informationen über den Benutzer enthält.

Zusätzlich zu den Standard-Benutzereigenschaften werden die folgenden Informationen zurückgegeben.

Eigenschaft	Type	Beschreibung
auth_type	integer	Standardauthentifizierung für den Benutzer. Eine Liste der möglichen Werte finden Sie in der Eigenschaft `authentication_type` des Authentifizierungsobjekts.
debug_mode	integer	Gibt an, ob der Debug-Modus für den Benutzer aktiviert oder deaktiviert ist. Eine Liste der möglichen Werte finden Sie in der Eigenschaft `debug_mode` des Benutzergruppenobjekts.
deprovisioned	boolean	Gibt an, ob der Benutzer zu einer Gruppe deprovisionierter Benutzer gehört.
gui_access	string	Authentifizierungsmethode des Benutzers für das Frontend. Eine Liste der möglichen Werte finden Sie in der Eigenschaft `gui_access` des Benutzergruppenobjekts.
secret	string	Zufällige Zeichenfolge mit 32 Zeichen. Wird bei der Benutzeranmeldung generiert. Die Eigenschaft `secret` wird nicht zurückgegeben, wenn die Benutzersitzung mit einem API-Token geprüft wird.
sessionid	string	Authentifizierungstoken, das in den folgenden API-Anfragen verwendet werden muss. Die Eigenschaft `sessionid` wird nicht zurückgegeben, wenn die Benutzersitzung mit einem API-Token geprüft wird.
type	integer	Benutzertyp. Eine Liste der möglichen Werte finden Sie in der Eigenschaft `type` des Rollenobjekts.
userip	string	IP-Adresse des Benutzers.

Beispiele

Authentifizierung mit Authentifizierungs-Token prüfen

Prüfen und verlängern Sie eine Benutzersitzung mithilfe des Benutzerauthentifizierungs-Tokens und geben Sie zusätzliche Informationen über den Benutzer zurück.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "user.checkAuthentication",
    "params": {
        "sessionid": "673b8ba11562a35da902c66cf5c23fa2"
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```


Authentifizierung mit API-Token prüfen

Prüfen Sie eine Benutzersitzung mithilfe des Benutzer-API-Tokens und geben Sie zusätzliche Informationen über den Benutzer zurück.

Anfrage:


```
{
    "jsonrpc": "2.0",
    "method": "user.checkAuthentication",
    "params": {
        "token": "00aff470e07c12d707e50d98cfe39edef9e6ec349c14728dbdfbc8ddc5ea3eae"
    },
    "id": 1
}
```


Antwort:


```
{
    "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
}
```


Quelle

CUser::checkAuthentication() in *ui/include/classes/api/services/CUser.php*.

user.create

Beschreibung

`object user.create(object/array users)`

Mit dieser Methode können Sie neue Benutzer erstellen.

Note:
Diese Methode ist nur für den Benutzertyp *Super-Admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Note:
Die Stärke des Benutzerpassworts wird anhand der von der Authentication API definierten Regeln für Passwortrichtlinien überprüft. Weitere Informationen finden Sie unter Authentication-Objekt.

Parameter

`(object/array)` Zu erstellende Benutzer.

Zusätzlich zu den Standard-Benutzereigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Typ	Beschreibung
usrgrps	array	Benutzergruppen, zu denen der Benutzer hinzugefügt werden soll. Für die Benutzergruppen darf nur die Eigenschaft `usrgrpid` definiert sein.
medias	array	Benutzermedien, die erstellt werden sollen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Benutzer unter der Eigenschaft `userids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Benutzer.

Beispiele

Erstellen eines Benutzers

Erstellen Sie einen neuen Benutzer, fügen Sie ihn einer Benutzergruppe hinzu und erstellen Sie ein neues Medium für ihn.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "user.create",
    "params": {
        "username": "John",
        "passwd": "Doe12345",
        "roleid": "5",
        "usrgrps": [
            {
                "usrgrpid": "7"
            }
        ],
        "medias": [
            {
                "mediatypeid": "1",
                "sendto": [
                    "support@company.com"
                ],
                "active": 0,
                "severity": 63,
                "period": "1-7,00:00-24:00"
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "12"
        ]
    },
    "id": 1
}
```

Siehe auch

- Authentifizierung
- Medien
- Benutzergruppe
- Rolle

Quelle

CUser::create() in *ui/include/classes/api/services/CUser.php*.

user.delete

Beschreibung

`object user.delete(array users)`

Diese Methode ermöglicht es Ihnen, Benutzer zu löschen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der zu löschenden Benutzer.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Benutzer unter der Eigenschaft `userids` enthält.

Beispiele

Mehrere Benutzer löschen

Löschen Sie zwei Benutzer.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "user.delete",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

Quelle

CUser::delete() in *ui/include/classes/api/services/CUser.php*.

user.get

Beschreibung

`integer/array user.get(object parameters)`

Die Methode ermöglicht es Ihnen, Benutzer anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

> **Note:**
> Beim Anfordern von Benutzermedien, Berechtigungen oder Rolleninformationen können Benutzer des Typs *Admin* und *User* Daten nur über ihren eigenen Benutzer abrufen. Ein Beispiel finden Sie unter Abrufen von Benutzern als *Admin*.

Parameter	Typ	Beschreibung
mediaids	ID/array	Gibt nur Benutzer zurück, die die angegebenen Medien verwenden.
mediatypeids	ID/array	Gibt nur Benutzer zurück, die die angegebenen Medientypen verwenden.
userids	ID/array	Gibt nur Benutzer mit den angegebenen IDs zurück.
usrgrpids	ID/array	Gibt nur Benutzer zurück, die zu den angegebenen Benutzergruppen gehören.

Parameter	Typ	Beschreibung
getAccess	flag	Fügt zusätzliche Informationen über Benutzerberechtigungen hinzu. Fügt für jeden Benutzer die folgenden Eigenschaften hinzu: `gui_access` - `(integer)` Authentifizierungsmethode des Benutzers im Frontend. Eine Liste möglicher Werte finden Sie in der Eigenschaft `gui_access` des Benutzergruppenobjekts. `debug_mode` - `(integer)` gibt an, ob Debugging für den Benutzer aktiviert ist. Mögliche Werte: 0 - Debugging deaktiviert, 1 - Debugging aktiviert. `users_status` - `(integer)` gibt an, ob der Benutzer deaktiviert ist. Mögliche Werte: 0 - Benutzer aktiviert, 1 - Benutzer deaktiviert.
selectMedias	query	Gibt die vom Benutzer verwendeten Medien in der Eigenschaft `medias` zurück.
selectMediatypes	query	Gibt die vom Benutzer verwendeten Medientypen in der Eigenschaft `mediatypes` zurück. Siehe `mediatype.get` für Einschränkungen basierend auf dem Benutzertyp.
selectUsrgrps	query	Gibt die Benutzergruppen, zu denen der Benutzer gehört, in der Eigenschaft `usrgrps` zurück. Siehe `usergroup.get` für Einschränkungen basierend auf dem Benutzertyp.
selectRole	query	Gibt die Benutzerrolle in der Eigenschaft `role` zurück.
filter	object	Gibt nur Ergebnisse zurück, die exakt mit dem angegebenen Filter übereinstimmen. Akzeptiert ein Objekt, bei dem die Schlüssel Eigenschaftsnamen sind und die Werte entweder ein einzelner Wert oder ein Array von Werten sind, mit denen abgeglichen werden soll. Unterstützt keine Eigenschaften des Datentyps `text`. Mögliche Eigenschaften des Benutzerobjekts für Benutzer des Typs *Admin* und *User* beim Anfordern von Daten über Benutzer in ihrer Benutzergruppe: `userid`, `name`, `surname`, `username`.
output	query	Eigenschaften des Benutzerobjekts, die zurückgegeben werden sollen. Benutzer des Typs *Admin* und *User* können nur die folgenden Eigenschaften abrufen: - Für ihren eigenen Benutzer: `userid`, `attempt_clock`, `attempt_failed`, `attempt_ip`, `autologin`, `autologout`, `lang`, `name`, `provisioned`, `refresh`, `roleid`, `rows_per_page`, `surname`, `theme`, `timezone`, `url`, `username`. - Für Benutzer in ihrer Benutzergruppe: `userid`, `name`, `surname`, `username`. Standard: `extend`.
search	object	Gibt Ergebnisse zurück, die dem angegebenen Muster entsprechen (Groß-/Kleinschreibung wird nicht beachtet). Akzeptiert ein Objekt, bei dem die Schlüssel Eigenschaftsnamen sind und die Werte Zeichenfolgen sind, nach denen gesucht werden soll. Wenn keine zusätzlichen Optionen angegeben sind, wird eine Suche vom Typ `LIKE "%…%"` durchgeführt. Unterstützt nur Eigenschaften des Datentyps `string` und `text`. Mögliche Eigenschaften des Benutzerobjekts für Benutzer des Typs *Admin* und *User* beim Anfordern von Daten über Benutzer in ihrer Benutzergruppe: `name`, `surname`, `username`.

Parameter	Typ	Beschreibung
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `userid`, `username`.
countOutput	boolean	Diese Parameter werden in der Referenzkommentierung beschrieben.
editable	boolean	
excludeSearch	boolean	
limit	integer	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Gibt entweder Folgendes zurück:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Benutzer abrufen

Rufen Sie alle konfigurierten Benutzer ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "userid": "1",
            "username": "Admin",
            "name": "Zabbix",
            "surname": "Administrator",
            "url": "",
            "autologin": "1",
            "autologout": "0",
            "lang": "en_US",
            "default_maintenance_period": "1h",
            "refresh": "30s",
            "theme": "default",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "50",
            "timezone": "default",
            "roleid": "3",
            "userdirectoryid": "0",
            "ts_provisioned": "0",
            "provisioned": "0"
        },
        {
```

```
            "userid": "2",
            "username": "guest",
            "name": "",
            "surname": "",
            "url": "",
            "autologin": "0",
            "autologout": "15m",
            "lang": "default",
            "default_maintenance_period": "1h",
            "refresh": "30s",
            "theme": "default",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "50",
            "timezone": "default",
            "roleid": "4",
            "userdirectoryid": "0",
            "ts_provisioned": "0",
            "provisioned": "0"
        }
    ],
    "id": 1
}
```

Abrufen von Benutzern als *Admin*

Als Benutzer vom Typ *Admin* können Sie detaillierte Daten zu Ihrem eigenen Benutzer und eingeschränkte Daten zu Benutzern in Ihrer Benutzergruppe abrufen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": "extend",
        "getAccess": true,
        "selectMedias": "extend",
        "selectMediatypes": "extend",
        "selectUsrgrps": "extend",
        "selectRole": "extend"
    },
    "id": 1
}
```

Antwort:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "userid": "1",
      "username": "Admin",
      "name": "Zabbix",
      "surname": "Administrator",
      "url": "",
      "autologin": "1",
      "autologout": "0",
      "lang": "default",
      "default_maintenance_period": "1h",
      "refresh": "30s",
      "theme": "default",
      "attempt_failed": "0",
      "attempt_ip": "",
```

```
    "attempt_clock": "0",
    "rows_per_page": "50",
    "timezone": "default",
    "roleid": "3",
    "userdirectoryid": "0",
    "ts_provisioned": "0",
    "provisioned": "0",
    "gui_access": "1",
    "debug_mode": "0",
    "users_status": "0",
    "usrgrps": [
      {
        "usrgrpid": "7",
        "name": "Zabbix administrators",
        "gui_access": "0",
        "users_status": "0",
        "debug_mode": "0",
        "userdirectoryid": "0",
        "mfa_status": "0",
        "mfaid": "0",
        "proxy_mode": "0",
        "proxy_group_mode": "0"
      },
      {
        "usrgrpid": "13",
        "name": "Internal",
        "gui_access": "1",
        "users_status": "0",
        "debug_mode": "0",
        "userdirectoryid": "0",
        "mfa_status": "0",
        "mfaid": "0",
        "proxy_mode": "0",
        "proxy_group_mode": "0"
      }
    ],
    "medias": [],
    "mediatypes": [],
    "role": {
      "roleid": "3",
      "name": "Super admin role",
      "type": "3",
      "readonly": "1"
    }
  },
  {
    "userid": "2",
    "username": "guest",
    "name": "",
    "surname": "",
    "url": "",
    "autologin": "0",
    "autologout": "15m",
    "lang": "default",
    "default_maintenance_period": "1h",
    "refresh": "30s",
    "theme": "default",
    "attempt_failed": "0",
    "attempt_ip": "",
    "attempt_clock": "0",
    "rows_per_page": "50",
    "timezone": "default",
```

```
      "roleid": "4",
      "userdirectoryid": "0",
      "ts_provisioned": "0",
      "provisioned": "0",
      "gui_access": "1",
      "debug_mode": "0",
      "users_status": "1",
      "usrgrps": [
        {
          "usrgrpid": "8",
          "name": "Guests",
          "gui_access": "0",
          "users_status": "0",
          "debug_mode": "0",
          "userdirectoryid": "0",
          "mfa_status": "0",
          "mfaid": "0",
          "proxy_mode": "0",
          "proxy_group_mode": "0"
        },
        {
          "usrgrpid": "9",
          "name": "Disabled",
          "gui_access": "0",
          "users_status": "1",
          "debug_mode": "0",
          "userdirectoryid": "0",
          "mfa_status": "0",
          "mfaid": "0",
          "proxy_mode": "0",
          "proxy_group_mode": "0"
        },
        {
          "usrgrpid": "13",
          "name": "Internal",
          "gui_access": "1",
          "users_status": "0",
          "debug_mode": "0",
          "userdirectoryid": "0",
          "mfa_status": "0",
          "mfaid": "0",
          "proxy_mode": "0",
          "proxy_group_mode": "0"
        }
      ],
      "medias": [],
      "mediatypes": [],
      "role": {
        "roleid": "4",
        "name": "Guest role",
        "type": "1",
        "readonly": "0"
      }
    }
  ],
  "id": 1
}
```


Benutzerdaten abrufen

Rufen Sie die Daten des Benutzers mit der ID „2“ ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": ["userid", "username"],
        "selectRole": "extend",
        "userids": "2"
    },
    "id": 1
}
```

Antwort:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "userid": "2",
      "username": "guest",
      "role": {
        "roleid": "4",
        "name": "Gastrolle",
        "type": "1",
        "readonly": "0"
      }
    }
  ],
  "id": 1
}
```

Siehe auch

- Medien
- Medientyp
- Benutzergruppe
- Rolle

Quelle

CUser::get() in *ui/include/classes/api/services/CUser.php*.

user.login

Beschreibung

`string/object user.login(object parameters)`

Diese Methode ermöglicht es Ihnen, sich bei der API anzumelden und ein Authentifizierungstoken zu generieren.

Warning:
Wenn Sie diese Methode verwenden, müssen Sie auch user.logout ausführen, um die Erzeugung einer großen Anzahl offener Sitzungsdatensätze zu verhindern.

Attention:
Diese Methode ist nur für nicht authentifizierte Benutzer verfügbar, die keiner Benutzergruppe mit aktivierter Multi-Faktor-Authentifizierung angehören.

Parameter

`(object)` Parameter, die den Benutzernamen und das Passwort enthalten.

Die Methode akzeptiert die folgenden Parameter.

Parameter	Type	Beschreibung
password	string	Benutzerpasswort. Parameter behavior: - *erforderlich*
username	string	Benutzername. Parameter behavior: - *erforderlich*
userData	flag	Gibt Informationen über den authentifizierten Benutzer zurück.

Rückgabewerte

`(string/object)` Wenn der Parameter `userData` verwendet wird, wird ein Objekt zurückgegeben, das Informationen über den authentifizierten Benutzer enthält.

Zusätzlich zu den Standard-Benutzereigenschaften werden die folgenden Informationen zurückgegeben:

Eigenschaft	Typ	Beschreibung
auth_type	integer	Standardauthentifizierung für den Benutzer. Eine Liste der möglichen Werte finden Sie in der Eigenschaft `authentication_type` des Authentifizierungsobjekts.
debug_mode	integer	Gibt an, ob der Debug-Modus für den Benutzer aktiviert oder deaktiviert ist. Eine Liste der möglichen Werte finden Sie in der Eigenschaft `debug_mode` des Benutzergruppenobjekts.
deprovisioned	boolean	Gibt an, ob der Benutzer zu einer Gruppe deprovisionierter Benutzer gehört.
gui_access	string	Authentifizierungsmethode des Benutzers für das Frontend. Eine Liste der möglichen Werte finden Sie in der Eigenschaft `gui_access` des Benutzergruppenobjekts.
mfaid	integer	ID der MFA-Methode, die beim Anmelden für den Benutzer verwendet wird. Gibt ”0” zurück, wenn MFA global oder für alle Benutzergruppen, denen der Benutzer angehört, deaktiviert ist.
secret	string	Zufällige Zeichenfolge mit 32 Zeichen. Wird bei der Benutzeranmeldung generiert.
sessionid	string	Authentifizierungstoken, das in den folgenden API-Anfragen verwendet werden muss.
type	integer	Benutzertyp. Eine Liste der möglichen Werte finden Sie in der Eigenschaft `type` des Rollenobjekts.
userip	string	IP-Adresse des Benutzers.

Note:
Wenn ein Benutzer nach einem oder mehreren fehlgeschlagenen Versuchen erfolgreich authentifiziert wurde, gibt die Methode die aktuellen Werte für die Eigenschaften `attempt_clock`, `attempt_failed` und `attempt_ip` zurück und setzt sie anschließend zurück.

Wenn der Parameter `userData` nicht verwendet wird, gibt die Methode ein Authentifizierungstoken zurück, das für die Authentifizierung erforderlich ist.

Beispiele

Einen Benutzer authentifizieren

Authentifizieren Sie einen Benutzer.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "user.login",
    "params": {
        "username": "Admin",
        "password": "zabbix"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": "0424bd59b807674191e7d77572075f33",
    "id": 1
}
```

Informationen des authentifizierten Benutzers anfordern

Authentifizieren und zusätzliche Informationen über den Benutzer zurückgeben.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "user.login",
    "params": {
        "username": "Admin",
        "password": "zabbix",
        "userData": true
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```

Siehe auch

- user.logout

Quelle

CUser::login() in *ui/include/classes/api/services/CUser.php*.

user.logout

Beschreibung

`string/object user.logout(array)`

Diese Methode ermöglicht es Ihnen, sich von der API abzumelden, und macht das aktuelle Authentifizierungstoken ungültig.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` Die Methode akzeptiert ein leeres Array.

Rückgabewerte

`(boolean)` Gibt `true` zurück, wenn der Benutzer erfolgreich abgemeldet wurde.

Beispiele

Abmelden

Von der API abmelden.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "user.logout",
    "params": [],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```

Siehe auch

- user.login

Quelle

CUser::login() in *ui/include/classes/api/services/CUser.php*.

user.provision

Beschreibung

`object user.provision(object/array users)`

Diese Methode ermöglicht es Ihnen, LDAP-Benutzer bereitzustellen.

Note:
Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der bereitzustellenden Benutzer.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der bereitgestellten Benutzer in der Eigenschaft `userids` enthält.

Beispiele

Bereitstellung mehrerer Benutzer

Stellen Sie zwei Benutzer bereit.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "user.provision",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

Quelle

CUser::provision() in *ui/include/classes/api/services/CUser.php*.

user.resettotp

Beschreibung

`object user.resettotp(object/array users)`

Diese Methode ermöglicht es Ihnen, die TOTP-Geheimnisse von Benutzern zurückzusetzen.

Note:
Diese Methode ist nur für den Benutzertyp *Superadmin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der Benutzer, für die TOTP-Geheimnisse zurückgesetzt werden sollen.

Note:
Benutzersitzungen für die angegebenen Benutzer werden ebenfalls gelöscht (mit Ausnahme des Benutzers, der die Anfrage sendet).

Rückgabewerte

(`object`) Gibt ein Objekt zurück, das unter der Eigenschaft `userids` die IDs der Benutzer enthält, für die TOTP-Geheimnisse zurückgesetzt wurden.

Beispiele

TOTP-Geheimnisse für mehrere Benutzer zurücksetzen

Setzen Sie die TOTP-Geheimnisse für zwei Benutzer zurück.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "user.resettotp",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

Siehe auch

- MFA Objekt

Quelle

CUser::resettotp() in *ui/include/classes/api/services/CUser.php*.

user.unblock

Beschreibung

`object user.unblock(array userids)`

Diese Methode ermöglicht es Ihnen, Benutzer zu entsperren.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

(`array`) IDs der Benutzer, deren Sperre aufgehoben werden soll.

Rückgabewerte

(`object`) Gibt ein Objekt zurück, das die IDs der entsperrten Benutzer unter der Eigenschaft `userids` enthält.

Beispiele

Mehrere Benutzer entsperren

Entsperren Sie zwei Benutzer.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "user.unblock",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

Quelle

CUser::unblock() in *ui/include/classes/api/services/CUser.php*.

user.update

Beschreibung

`object user.update(object/array users)`

Diese Methode ermöglicht es Ihnen, vorhandene Benutzer zu aktualisieren.

Note:
Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Note:
Die Stärke des Benutzerkennworts wird gemäß den Kennwortrichtlinien überprüft, die von der Authentication API definiert sind. Weitere Informationen finden Sie unter Authentication API.

Parameter

`(object/array)` Zu aktualisierende Benutzereigenschaften.

Die Eigenschaft `userid` muss für jeden Benutzer definiert sein, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Zusätzlich zu den Standard-Benutzereigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
current_passwd	string	Aktuelles Passwort des Benutzers. Der Wert dieses Parameters kann eine leere Zeichenfolge sein, wenn der Benutzer mit einem Benutzerverzeichnis verknüpft ist. Parameter behavior: - *write-only* - *required* if `passwd` of User object is set and user changes own user password

Parameter	Type	Beschreibung
usrgrps	array	Benutzergruppen zum Ersetzen bestehender Benutzergruppen. Für die Benutzergruppen darf nur die Eigenschaft `usrgrpid` definiert sein.
medias	array	Benutzermedien zum Ersetzen vorhandener, nicht bereitgestellter Medien. Bereitgestellte Medien können beim Aktualisieren von Medien weggelassen werden.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Benutzer unter der Eigenschaft `userids` enthält.

Beispiele

Umbenennen eines Benutzers

Benennen Sie einen Benutzer in John Doe um.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "user.update",
    "params": {
        "userid": "1",
        "name": "John",
        "surname": "Doe"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1"
        ]
    },
    "id": 1
}
```

Ändern der Benutzerrolle

Ändern Sie die Rolle eines Benutzers.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "user.update",
    "params": {
        "userid": "12",
        "roleid": "6"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "12"
        ]
```

```
    },
    "id": 1
}
```

Siehe auch

- Authentifizierung

Quelle

CUser::update() in *ui/include/classes/api/services/CUser.php*.

Benutzergruppe

Diese Klasse ist für die Arbeit mit Benutzergruppen vorgesehen.

Objektreferenzen:

- Benutzergruppe
- Berechtigung
- Tag-basierte Berechtigung

Verfügbare Methoden:

- usergroup.create - neue Benutzergruppen erstellen
- usergroup.delete - Benutzergruppen löschen
- usergroup.get - Benutzergruppen abrufen
- usergroup.update - Benutzergruppen aktualisieren

Benutzergruppen-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `usergroup` API.

Benutzergruppe

Das Benutzergruppenobjekt verfügt über die folgenden Eigenschaften.

Property	Type	Description
usrgrpid	ID	ID der Benutzergruppe. Eigenschaftsverhalten: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
name	string	Name der Benutzergruppe. Eigenschaftsverhalten: - *erforderlich* für Erstellungsvorgänge
debug_mode	integer	Gibt an, ob der Debug-Modus aktiviert oder deaktiviert ist. Mögliche Werte: 0 - *(Standard)* deaktiviert; 1 - aktiviert.
gui_access	integer	Authentifizierungsmethode des Frontends für die Benutzer in der Gruppe. Mögliche Werte: 0 - *(Standard)* die systemweite Standardauthentifizierungsmethode verwenden; 1 - die interne Authentifizierung verwenden; 2 - die LDAP-Authentifizierung verwenden; 3 - den Zugriff auf das Frontend deaktivieren.

Property	Type	Description
mfa_status	integer	Gibt an, ob MFA für die Benutzer in der Gruppe aktiviert oder deaktiviert ist. Mögliche Werte: 0 - deaktiviert (für alle konfigurierten MFA-Methoden); 1 - aktiviert (für alle konfigurierten MFA-Methoden).
mfaid	ID	MFA-Methode, die für die Benutzer in der Gruppe verwendet wird. Eigenschaftsverhalten: - *unterstützt* für Benutzer des Typs *Super admin* - *unterstützt*, wenn `mfa_status` des Authentifizierungsobjekts auf „Enabled“ gesetzt ist
proxy_mode	integer	Modus für die Handhabung des Zugriffs auf Proxys, die nicht in Proxygruppen enthalten sind. Mögliche Werte: 0 - Verweigerungsliste; 1 - *(Standard)* Zulassungsliste. Eigenschaftsverhalten: - *unterstützt* für Benutzer des Typs *Super admin*.
proxy_group_mode	integer	Modus für die Handhabung des Zugriffs auf Proxygruppen (und deren Proxys). Mögliche Werte: 0 - Verweigerungsliste; 1 - *(Standard)* Zulassungsliste. Eigenschaftsverhalten: - *unterstützt* für Benutzer des Typs *Super admin*.
users_status	integer	Gibt an, ob die Benutzergruppe aktiviert oder deaktiviert ist. Für deprovisionierte Benutzer kann die Benutzergruppe nicht aktiviert werden. Mögliche Werte: 0 - *(Standard)* aktiviert; 1 - deaktiviert.
userdirectoryid	ID	ID des Benutzerverzeichnisses, das für die Authentifizierung verwendet wird. Eigenschaftsverhalten: - *unterstützt* für Benutzer des Typs *Super admin* - *unterstützt*, wenn `gui_access` auf „use the system default authentication method“ oder „use LDAP authentication“ gesetzt ist

Berechtigung

Das Berechtigungsobjekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
id	ID	ID der Host-Gruppe oder Vorlagengruppe, der eine Berechtigung hinzugefügt werden soll. Verhalten der Eigenschaft: - *erforderlich*

Eigenschaft	Typ	Beschreibung
permission	integer	Zugriffsebene auf die Host-Gruppe oder Vorlagengruppe. Mögliche Werte: 0 - Zugriff verweigert; 2 - schreibgeschützter Zugriff; 3 - Lese-/Schreibzugriff. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsoperationen

Tag-basierte Berechtigung

Das Objekt für tag-basierte Berechtigungen hat die folgenden Eigenschaften.

Eigenschaft	Type	Beschreibung
groupid	ID	ID der Host-Gruppe, der die Berechtigung hinzugefügt werden soll. Property behavior: - *required*
tag	string	Tag-Name.
value	string	Tag-Wert.

APM-based permission

The APM-based permission object has the following properties.

Property	Type	Description
key	string	APM resource attribute key. Property behavior: - *required*
value	string	APM resource attribute value.
signal_type	integer	APM resource attribute category. Possible values: -1 - *(default)* All; 0 - Traces; 1 - Metrics; 2 - Logs.

usergroup.create

Beschreibung

`object usergroup.create(object/array userGroups)`

Diese Methode ermöglicht es Ihnen, neue Benutzergruppen zu erstellen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Benutzergruppen, die erstellt werden sollen.

Zusätzlich zu den Standardeigenschaften von Benutzergruppen akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
hostgroup_rights	object/array	Dem Benutzergruppe zuzuweisende Berechtigungen für Hostgruppen.
proxies	object/array	Proxies, die mit der Benutzergruppe verknüpft werden sollen. Die Proxies dürfen nur die Eigenschaft `proxyid` definiert haben.
proxy_groups	object/array	Proxygruppen, die mit der Benutzergruppe verknüpft werden sollen. Die Proxygruppen dürfen nur die Eigenschaft `proxy_groupid` definiert haben.
templategroup_rights	object/array	Der Benutzergruppe zuzuweisende Berechtigungen für Vorlagengruppen.
tag_filters	array	Der Benutzergruppe zuzuweisende tagbasierte Berechtigungen.
users	object/array	Benutzer, die der Benutzergruppe hinzugefügt werden sollen. Der Benutzer darf nur die Eigenschaft `userid` definiert haben.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Benutzergruppen in der Eigenschaft `usrgrpids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Benutzergruppen.

Beispiele

Erstellen einer Benutzergruppe

Erstellen Sie eine Benutzergruppe *Operation managers* mit verweigertem Zugriff auf die Host-Gruppe „2“ und fügen Sie ihr einen Benutzer hinzu.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.create",
    "params": {
        "name": "Operation managers",
        "hostgroup_rights": {
            "id": "2",
            "permission": 0
        },
        "users": [
            {
                "userid": "12"
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "usrgrpids": [
            "20"
        ]
    },
    "id": 1
}
```

Siehe auch

- Berechtigung

Quelle

CUserGroup::create() in *ui/include/classes/api/services/CUserGroup.php*.

usergroup.delete

Beschreibung

`object usergroup.delete(array userGroupIds)`

Mit dieser Methode können Sie Benutzergruppen löschen.

Attention:
Die Gruppe Deprovisioned users (die in Authentication für `disabled_usrgrpid` angegebene Benutzergruppe) kann nicht gelöscht werden.

Note:
Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter User roles.

Parameter

`(array)` IDs der zu löschenden Benutzergruppen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Benutzergruppen in der Eigenschaft `usrgrpids` enthält.

Beispiele

Mehrere Benutzergruppen löschen

Löschen Sie zwei Benutzergruppen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.delete",
    "params": [
        "20",
        "21"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "usrgrpids": [
            "20",
            "21"
        ]
    },
    "id": 1
}
```

Quelle

CUserGroup::delete() in *ui/include/classes/api/services/CUserGroup.php*.

usergroup.get

Beschreibung

`integer/array usergroup.get(object parameters)`

Die Methode ermöglicht es Ihnen, Benutzergruppen anhand der angegebenen Parameter abzurufen.

Note:
Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Typ	Beschreibung
mfaids	ID/array	Gibt nur Benutzergruppen mit den angegebenen MFA-Methoden zurück. Parameterverhalten: - *unterstützt* für Benutzer des Typs *Super admin*
mfa_status	integer	Gibt nur Benutzergruppen mit dem angegebenen MFA-Status zurück. Eine Liste der unterstützten Status finden Sie auf der Seite zu Benutzergruppen.
proxyids	ID/array	Gibt nur Benutzergruppen zurück, die die angegebenen Proxys enthalten. Parameterverhalten: - *unterstützt* für Benutzer des Typs *Super admin*.
proxy_groupids	ID/array	Gibt nur Benutzergruppen zurück, die die angegebenen Proxy-Gruppen enthalten. Parameterverhalten: - *unterstützt* für Benutzer des Typs *Super admin*.
selectProxies	ID/array	Gibt eine Eigenschaft `proxies` mit den Proxys der Benutzergruppe zurück. Parameterverhalten: - *unterstützt* für Benutzer des Typs *Super admin*.
selectProxyGroups	ID/array	Gibt eine Eigenschaft `proxy_groups` mit den Proxy-Gruppen der Benutzergruppe zurück. Parameterverhalten: - *unterstützt* für Benutzer des Typs *Super admin*.
status	integer	Gibt nur Benutzergruppen mit dem angegebenen Status zurück. Eine Liste der unterstützten Status finden Sie auf der Seite zu Benutzergruppen.
userids	ID/array	Gibt nur Benutzergruppen zurück, die die angegebenen Benutzer enthalten.
usrgrpids	ID/array	Gibt nur Benutzergruppen mit den angegebenen IDs zurück.
selectTagFilters	query	Gibt die tagbasierten Berechtigungen der Benutzergruppe in der Eigenschaft `tag_filters` zurück.
selectUsers	query	Gibt die Benutzer der Benutzergruppe in der Eigenschaft `users` zurück. Informationen zu Einschränkungen basierend auf dem Benutzertyp finden Sie unter `user.get`.
selectHostGroupRights	query	Gibt die Berechtigungen der Benutzergruppe für Hostgruppen in der Eigenschaft `hostgroup_rights` zurück. Eine Liste der Zugriffsebenen für Hostgruppen finden Sie auf der Seite zu Benutzergruppen.

Parameter	Typ	Beschreibung
selectTemplateGroupRights	query	Gibt die Berechtigungen der Benutzergruppe für Vorlagengruppen in der Eigenschaft `templategroup_rights` zurück. Eine Liste der Zugriffsebenen für Vorlagengruppen finden Sie auf der Seite zu Benutzergruppen.
limitSelects	integer	Begrenzt die Anzahl der von Unterabfragen zurückgegebenen Datensätze.
output	query	Zurückzugebende Eigenschaften des Benutzergruppenobjekts. Benutzer des Typs *Admin* und *User* können nur die folgenden Eigenschaften abrufen: `usrgrpid`, `name`, `gui_access`, `users_status`, `debug_mode`, `mfa_status`. Standard: `extend`.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `usrgrpid`, `name`.
countOutput	boolean	Diese Parameter werden im Referenzkommentar beschrieben.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Gibt entweder Folgendes zurück:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Aktivierte Benutzergruppen abrufen

Rufen Sie alle aktivierten Benutzergruppen ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.get",
    "params": {
        "output": "extend",
        "status": 0
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "usrgrpid": "7",
            "name": "Zabbix-Administratoren",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "1",
```

```
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "8",
            "name": "Gäste",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "0",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "11",
            "name": "Debug-Modus aktiviert",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "1",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "12",
            "name": "Kein Zugriff auf das Frontend",
            "gui_access": "2",
            "users_status": "0",
            "debug_mode": "0",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "14",
            "name": "Schreibgeschützt",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "0",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        },
        {
            "usrgrpid": "18",
            "name": "Verweigern",
            "gui_access": "0",
            "users_status": "0",
            "debug_mode": "0",
            "userdirectoryid": "0",
            "mfa_status": "0",
            "mfaid": "0"
        }
    ],
    "id": 1
}
```


Siehe auch

- User

Quelle

CUserGroup::get() in *ui/include/classes/api/services/CUserGroup.php*.

usergroup.update

Beschreibung

`object usergroup.update(object/array userGroups)`

Diese Methode ermöglicht es Ihnen, vorhandene Benutzergruppen zu aktualisieren.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Eigenschaften der Benutzergruppen, die aktualisiert werden sollen.

Die Eigenschaft `usrgrpid` muss für jede Benutzergruppe definiert sein. Alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Zusätzlich zu den Standard-Eigenschaften von Benutzergruppen akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Description
hostgroup_rights	object/array	Berechtigungen für Hostgruppen (permissions), die die aktuellen, der Benutzergruppe zugewiesenen Berechtigungen ersetzen sollen.
proxies	object/array	Proxies, die mit der Benutzergruppe verknüpft werden sollen. Für die Proxies darf nur die Eigenschaft `proxyid` definiert sein.
proxy_groups	object/array	Proxygruppen, die mit der Benutzergruppe verknüpft werden sollen. Für die Proxygruppen darf nur die Eigenschaft `proxy_groupid` definiert sein.
templategroup_rights	object/array	Berechtigungen für Vorlagengruppen (permissions), die die aktuellen, der Benutzergruppe zugewiesenen Berechtigungen ersetzen sollen.
tag_filters	array	Tag-basierte Berechtigungen, die die aktuellen, der Benutzergruppe zugewiesenen Berechtigungen ersetzen sollen.
users	object/array	Benutzer, die die aktuellen, der Benutzergruppe zugewiesenen Benutzer ersetzen sollen. Für den Benutzer darf nur die Eigenschaft `userid` definiert sein.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Benutzergruppen in der Eigenschaft `usrgrpids` enthält.

Beispiele

Aktivieren einer Benutzergruppe und Aktualisieren von Berechtigungen

Aktivieren Sie eine Benutzergruppe und gewähren Sie ihr Lese-/Schreibzugriff auf die Host-Gruppen „2“ und „4“.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.update",
    "params": {
        "usrgrpid": "17",
        "users_status": "0",
        "hostgroup_rights": [
            {
                "id": "2",
                "permission": 3
            },
            {
                "id": "4",
                "permission": 3
```

```
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "usrgrpids": [
            "17"
        ]
    },
    "id": 1
}
```

Siehe auch

- Berechtigung

Quelle

CUserGroup::update() in *ui/include/classes/api/services/CUserGroup.php*.

Benutzermakro

Diese Klasse ist für die Arbeit mit Benutzermakros auf Host-Ebene und globalen Benutzermakros vorgesehen.

Objektreferenzen:

- Globales Makro
- Host-Makro
    - Makrokonfiguration

Verfügbare Methoden:

- usermacro.create - neue Host-Makros erstellen
- usermacro.createglobal - neue globale Makros erstellen
- usermacro.delete - Host-Makros löschen
- usermacro.deleteglobal - globale Makros löschen
- usermacro.get - Host- und globale Makros abrufen
- usermacro.update - Host-Makros aktualisieren
- usermacro.updateglobal - globale Makros aktualisieren

Benutzer-Makro-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `usermacro` API.

Globales Makro

Das globale Makro-Objekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
globalmacroid	ID	ID des globalen Makros. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
description	string	Beschreibung des Makros.
macro	string	Makro-Zeichenfolge. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge

Eigenschaft	Typ	Beschreibung
type	integer	Typ des Makros. Mögliche Werte: 0 - *(Standard)* Textmakro; 1 - Geheimes Makro; 2 - Vault-Geheimnis.
value	string	Wert des Makros. Verhalten der Eigenschaft: - *nur schreibbar*, wenn `type` auf ”Geheimes Makro” gesetzt ist - *erforderlich* für Erstellungsvorgänge

Host-Makro

Das Host-Makro-Objekt definiert ein Makro, das auf einem Host, Host-Prototyp oder in einer Vorlage verfügbar ist. Es hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
hostmacroid	ID	ID des Makros. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
automatic	integer	Definiert, ob das Makro durch eine Discovery-Regel gesteuert wird. Mögliche Werte: 0 - *(Standard)* Das Makro wird vom Benutzer verwaltet; 1 - Das Makro wird von der Discovery-Regel verwaltet. Der Benutzer darf kein automatisches Makro erstellen. Um ein automatisches Makro zu aktualisieren, muss es in ein manuelles Makro umgewandelt werden.
config	object/array	Makrokonfiguration, die dafür verantwortlich ist, wie das Makro im Host Wizard angezeigt wird.
description	string	Beschreibung des Makros.
hostid	ID	ID des Hosts, Host-Prototyps oder der Vorlage, zu dem bzw. zu der das Makro gehört. Verhalten der Eigenschaft: - *konstant* - *erforderlich* für Erstellungsvorgänge
macro	string	Makro-String. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
type	integer	Typ des Makros. Mögliche Werte: 0 - *(Standard)* Textmakro; 1 - Geheimes Makro; 2 - Vault-Geheimnis.
value	string	Wert des Makros. Verhalten der Eigenschaft: - *nur schreibbar*, wenn `type` auf ”Secret macro” gesetzt ist - *erforderlich* für Erstellungsvorgänge

Makrokonfiguration

Das Makrokonfigurationsobjekt definiert, wie ein Makro im Host Wizard angezeigt wird.

Eigenschaft	Typ	Beschreibung
type	integer	Typ des Makro-Eingabefelds. Mögliche Werte: 0 - Makro wird im Host Wizard nicht verwendet; 1 - Textfeld; 2 - Liste; 3 - Kontrollkästchen. Eigenschaftsverhalten: - *erforderlich*
label	string	Beschriftung für das Makro-Eingabefeld. Eigenschaftsverhalten: - *erforderlich*, wenn `type` auf ”Textbox”, ”List” oder ”Checkbox” gesetzt ist
description	text	Hilfetext, der neben dem Makro-Eingabefeld angezeigt wird. Unterstützt Markdown-Formatierung.
priority	integer	Position des Makro-Eingabefelds in der Makroliste. Wenn `priority` nicht gesetzt ist, wird das Makro am Ende der nicht gruppierten Makroliste hinzugefügt.
required	integer	Kennzeichnet das Makro als obligatorisch. Mögliche Werte: 0 - Nicht obligatorisch; 1 - Obligatorisch. Eigenschaftsverhalten: - *unterstützt*, wenn `type` auf ”Textbox” oder ”List” gesetzt ist
regex	string	Regulärer Ausdruck zur Validierung der Benutzereingabe in einem Textfeld. Eigenschaftsverhalten: - *unterstützt*, wenn `type` auf ”Textbox” gesetzt ist
section_name	string	Beschriftung des einklappbaren Abschnitts, in dem das Makro gruppiert wird. Wenn `section_name` nicht gesetzt ist, wird das Makro nicht gruppiert. Nicht gruppierte Makros werden zuerst angezeigt; gruppierte Makros werden darunter angezeigt und innerhalb jedes einklappbaren Abschnitts nach ihrer `priority` sortiert.
options	text	JSON-String zur Definition von Listeneinträgen oder Kontrollkästchenwerten. Für Listen: ein Array von Objekten mit den Eigenschaften `value` und `text`. Beispiel: `[{"value": "http", "text": "HTTP"}, {"value": "https", "text": "HTTPS"}]` Für Kontrollkästchen: ein Objekt mit den Eigenschaften `checked` und `unchecked`. Beispiel: `{"checked": true, "unchecked": false}` Eigenschaftsverhalten: - *erforderlich*, wenn `type` auf ”List” oder ”Checkbox” gesetzt ist

usermacro.create

Beschreibung

`object usermacro.create(object/array hostMacros)`

Diese Methode ermöglicht es Ihnen, neue Host-Makros zu erstellen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellende Host-Makros.

Die Methode akzeptiert Host-Makros mit den standardmäßigen Eigenschaften von Host-Makros.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Host- Makros in der Eigenschaft `hostmacroids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Host- Makros.

Beispiele

Erstellen eines Host-Makros

Erstellen Sie auf dem Host "10198" ein Host-Makro `{$SNMP_COMMUNITY}` mit dem Wert "public".

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.create",
    "params": {
        "hostid": "10198",
        "macro": "{$SNMP_COMMUNITY}",
        "value": "public"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "11"
        ]
    },
    "id": 1
}
```

Quelle

CUserMacro::create() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.createglobal

Beschreibung

`object usermacro.createglobal(object/array globalMacros)`

Diese Methode ermöglicht es Ihnen, neue globale Makros zu erstellen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellende globale Makros.

Die Methode akzeptiert globale Makros mit den Standard-Eigenschaften globaler Makros.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten globalen Makros in der Eigenschaft `globalmacroids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen globalen Makros.

Beispiele

Erstellen eines globalen Makros

Erstellen Sie ein globales Makro `{$SNMP_COMMUNITY}` mit dem Wert "public".

Request:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.createglobal",
    "params":  {
        "macro": "{$SNMP_COMMUNITY}",
        "value": "public"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "globalmacroids": [
            "6"
        ]
    },
    "id": 1
}
```

Quelle

CUserMacro::createGlobal() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.delete

Beschreibung

`object usermacro.delete(array hostMacroIds)`

Diese Methode ermöglicht es Ihnen, Host-Makros zu löschen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der zu löschenden Host-Makros.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Host-Makros in der Eigenschaft `hostmacroids` enthält.

Beispiele

Mehrere Host-Makros löschen

Löschen Sie zwei Host-Makros.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.delete",
    "params": [
        "32",
```

```
        "11"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "32",
            "11"
        ]
    },
    "id": 1
}
```

Quelle

CUserMacro::delete() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.deleteglobal

Beschreibung

`object usermacro.deleteglobal(array globalMacroIds)`

Diese Methode ermöglicht es Ihnen, globale Makros zu löschen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Berechtigungen zum Aufruf der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der zu löschenden globalen Makros.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten globalen Makros in der Eigenschaft `globalmacroids` enthält.

Beispiele

Mehrere globale Makros löschen

Löschen Sie zwei globale Makros.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.deleteglobal",
    "params": [
        "32",
        "11"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "globalmacroids": [
            "32",
            "11"
        ]
```

```
    },
    "id": 1
}
```

Quelle

CUserMacro::deleteGlobal() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.get

Beschreibung

`integer/array usermacro.get(object parameters)`

Die Methode ermöglicht es Ihnen, Host- und globale Makros anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
globalmacro	flag	Globale Makros anstelle von Host-Makros zurückgeben.
globalmacroids	ID/array	Nur globale Makros mit den angegebenen IDs zurückgeben.
groupids	ID/array	Nur Host-Makros zurückgeben, die zu Hosts oder Vorlagen aus den angegebenen Host-Gruppen oder Vorlagengruppen gehören.
hostids	ID/array	Nur Makros zurückgeben, die zu den angegebenen Hosts oder Vorlagen gehören.
hostmacroids	ID/array	Nur Host-Makros mit den angegebenen IDs zurückgeben.
inherited	boolean	Wenn auf `true` gesetzt, nur von einer Vorlage geerbte Benutzermakros von Host-Prototypen zurückgeben.
selectHostGroups	query	Host-Gruppen, zu denen das Host-Makro gehört, in der Eigenschaft `hostgroups` zurückgeben. Wird nur beim Abrufen von Host-Makros verwendet.
selectHosts	query	Hosts, zu denen das Host-Makro gehört, in der Eigenschaft `hosts` zurückgeben. Wird nur beim Abrufen von Host-Makros verwendet.
selectTemplateGroups	query	Vorlagengruppen, zu denen das Vorlagenmakro gehört, in der Eigenschaft `templategroups` zurückgeben. Wird nur beim Abrufen von Vorlagenmakros verwendet.
selectTemplates	query	Vorlagen, zu denen das Host-Makro gehört, in der Eigenschaft `templates` zurückgeben. Wird nur beim Abrufen von Host-Makros verwendet.
sortfield	string/array	Das Ergebnis nach den angegebenen Eigenschaften sortieren. Mögliche Werte: `macro`.
countOutput	boolean	Diese Parameter sind in der Referenzkommentierung beschrieben.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	

Parameter	Type	Beschreibung
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Gibt entweder Folgendes zurück:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Abrufen von Host-Makros für einen Host

Rufen Sie alle für den Host „10198“ definierten Host-Makros ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.get",
    "params": {
        "output": "extend",
        "hostids": "10198"
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```

Abrufen von Host-Makros für eine Vorlage

Rufen Sie die für die Vorlage „10265“ definierten Host-Makros ab, die „STATUS“ im Makronamen enthalten.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.get",
    "params": {
        "output": "extend",
```

```
        "hostids": "10265",
        "search": {
            "macro": "STATUS"
        }
    },
    "id": 1
}
```

Antwort:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostmacroid": "6709",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.HOST}",
            "value": "",
            "description": "Der Hostname oder die IP-Adresse des Apache-Statusseiten-Hosts.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "1",
                "priority": "1",
                "section_name": "",
                "label": "Apache-Status-Host",
                "description": "Hostname oder IP-Adresse des Apache-Statusseiten-Hosts.",
                "required": "1",
                "regex": "",
                "options": []
            }
        },
        {
            "hostmacroid": "814",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.PATH}",
            "value": "server-status?auto",
            "description": "Der URL-Pfad der Apache-Statusseite.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "1",
                "priority": "3",
                "section_name": "",
                "label": "Apache-Statusseitenpfad",
                "description": "URL-Pfad der Apache-Statusseite.",
                "required": "1",
                "regex": "",
                "options": []
            }
        },
        {
            "hostmacroid": "815",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.PORT}",
            "value": "80",
            "description": "Der Port der Apache-Statusseite.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "1",
                "priority": "2",
                "section_name": "",
```

```
                "label": "Apache-Statusseiten-Port",
                "description": "Im Bereich von 1 bis 65535 einschließlich.",
                "required": "1",
                "regex": "^-?([0-9]+|(([0-9]+)\\.([0-9]+)))$",
                "options": []
            }
        },
        {
            "hostmacroid": "816",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.SCHEME}",
            "value": "http",
            "description": "Das Anfrage-Schema, das entweder HTTP oder HTTPS sein kann.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "2",
                "priority": "4",
                "section_name": "",
                "label": "Anfrage-Schema",
                "description": "Anfrage-Schema, das entweder HTTP oder HTTPS sein kann.",
                "required": "0",
                "regex": "",
                "options": [
                    {
                        "value": "http",
                        "text": "HTTP"
                    },
                    {
                        "value": "https",
                        "text": "HTTPS"
                    }
                ]
            }
        }
    ],
    "id": 1
}
```


Abrufen globaler Makros

Rufen Sie alle globalen Makros ab.

Anfrage:


```
{
    "jsonrpc": "2.0",
    "method": "usermacro.get",
    "params": {
        "output": "extend",
        "globalmacro": true
    },
    "id": 1
}
```


Antwort:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "globalmacroid": "6",
            "macro": "{$SNMP_COMMUNITY}",
            "value": "public",
            "description": "",
```

```
            "type": "0"
        }
    ],
    "id": 1
}
```

Quelle

CUserMacro::get() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.update

Beschreibung

`object usermacro.update(object/array hostMacros)`

Diese Methode ermöglicht es Ihnen, vorhandene Host-Makros zu aktualisieren.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` zu aktualisierende Eigenschaften von Host-Makros.

Die Eigenschaft `hostmacroid` muss für jedes Host-Makro definiert werden, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Host-Makros in der Eigenschaft `hostmacroids` enthält.

Beispiele

Ändern des Werts eines Host-Makros

Ändern Sie den Wert eines Host-Makros auf „public“.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.update",
    "params": {
        "hostmacroid": "1",
        "value": "public"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "1"
        ]
    },
    "id": 1
}
```

Makrowert ändern, der durch eine Discovery-Regel erstellt wurde

Ein von einer Discovery-Regel erstelltes ”automatisches” Makro in ”manuell” umwandeln und seinen Wert in ”new-value” ändern.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.update",
    "params": {
        "hostmacroid": "1",
        "value": "new-value",
        "automatic": "0"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "1"
        ]
    },
    "id": 1
}
```

Quelle

CUserMacro::update() in *ui/include/classes/api/services/CUserMacro.php*.

usermacro.updateglobal

Beschreibung

`object usermacro.updateglobal(object/array globalMacros)`

Diese Methode ermöglicht es Ihnen, vorhandene globale Makros zu aktualisieren.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` zu aktualisierende Eigenschaften globaler Makros.

Die Eigenschaft `globalmacroid` muss für jedes globale Makro definiert werden, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten globalen Makros unter der Eigenschaft `globalmacroids` enthält.

Beispiele

Ändern des Werts eines globalen Makros

Ändern Sie den Wert eines globalen Makros in "public".

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.updateglobal",
    "params": {
        "globalmacroid": "1",
        "value": "public"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "globalmacroids": [
            "1"
        ]
    },
    "id": 1
}
```

Quelle

CUserMacro::updateGlobal() in *ui/include/classes/api/services/CUserMacro.php*.

Benutzerverzeichnis

Diese Klasse ist für die Arbeit mit Benutzerverzeichnissen konzipiert.

Objektreferenzen:

- Benutzerverzeichnis
  - Medientyp-Zuordnungen
  - Zuordnungen von Bereitstellungsgruppen

Verfügbare Methoden:

- userdirectory.create - neues Benutzerverzeichnis erstellen
- userdirectory.delete - Benutzerverzeichnis löschen
- userdirectory.get - Benutzerverzeichnis abrufen
- userdirectory.update - Benutzerverzeichnis aktualisieren
- userdirectory.test - Verbindung zum Benutzerverzeichnis testen

Benutzerverzeichnis-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `userdirectory` API.

Benutzerverzeichnis

Das Objekt des Benutzerverzeichnisses verfügt über die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
userdirectoryid	ID	ID des Benutzerverzeichnisses. Wenn ein Benutzerverzeichnis gelöscht wird, wird der Wert der Eigenschaft `userdirectoryid` des Benutzerobjekts für alle Benutzer, die mit dem gelöschten Benutzerverzeichnis verknüpft sind, auf "0" gesetzt. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
idp_type	integer	Typ des vom Identitätsanbieter für das Benutzerverzeichnis verwendeten Authentifizierungsprotokolls. Beachten Sie, dass nur ein Benutzerverzeichnis vom Typ SAML vorhanden sein kann. Mögliche Werte: 1 - Benutzerverzeichnis vom Typ LDAP; 2 - Benutzerverzeichnis vom Typ SAML. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge

Eigenschaft	Typ	Beschreibung
group_name	string	Attribut des LDAP/SAML-Benutzerverzeichnisses, das den Gruppennamen enthält, der zur Zuordnung von Gruppen zwischen dem LDAP/SAML-Benutzerverzeichnis und Zabbix verwendet wird. Beispiel: *cn* Verhalten der Eigenschaft: - *erforderlich*, wenn `provision_status` auf ”Enabled” gesetzt ist und `saml_jit_status` des Authentifizierungsobjekts auf ”Enabled for configured SAML IdPs” gesetzt ist
user_username	string	Attribut des LDAP/SAML-Benutzerverzeichnisses (auch SCIM-Attribut, wenn `scim_status` auf ”SCIM provisioning is enabled” gesetzt ist), das den Namen des Benutzers enthält. Dieser wird beim Bereitstellen des Benutzers als Wert der Eigenschaft `name` des Benutzerobjekts verwendet. Beispiele: *cn*, *commonName*, *displayName*, *name*
user_lastname	string	Attribut des LDAP/SAML-Benutzerverzeichnisses (auch SCIM-Attribut, wenn `scim_status` auf ”SCIM provisioning is enabled” gesetzt ist), das den Nachnamen des Benutzers enthält. Dieser wird beim Bereitstellen des Benutzers als Wert der Eigenschaft `surname` des Benutzerobjekts verwendet. Beispiele: *sn*, *surname*, *lastName*
provision_status	integer	Bereitstellungsstatus des Benutzerverzeichnisses. Mögliche Werte: 0 - *(Standard)* Disabled (Bereitstellung der von diesem Benutzerverzeichnis erstellten Benutzer ist deaktiviert); 1 - Enabled (Bereitstellung der von diesem Benutzerverzeichnis erstellten Benutzer ist aktiviert; zusätzlich muss der Status der LDAP- oder SAML-Bereitstellung (`ldap_jit_status` oder `saml_jit_status` des Authentifizierungsobjekts) aktiviert sein).
provision_groups	array	Array von Objekten für Zuordnungen von Bereitstellungsgruppen, um LDAP/SAML-Benutzergruppenmuster einer Zabbix-Benutzergruppe und Benutzerrolle zuzuordnen. Verhalten der Eigenschaft: - *erforderlich*, wenn `provision_status` auf ”Enabled” gesetzt ist
provision_media	array	Array von Objekten für Zuordnungen von Medientypen, um LDAP/SAML-Medienattribute des Benutzers (z. B. E-Mail) den Zabbix-Benutzermedien zum Senden von Benachrichtigungen zuzuordnen.
LDAP-spezifische Eigenschaften:		
name	string	Eindeutiger Name des Benutzerverzeichnisses. Verhalten der Eigenschaft: - *erforderlich*, wenn `idp_type` auf ”User directory of type LDAP” gesetzt ist
host	string	Hostname, IP-Adresse oder URI des LDAP-Servers. Die URI muss Schema (`ldap://` oder `ldaps://`), Host und optional den Port enthalten. Beispiele: *host.example.com* *127.0.0.1* *ldap://ldap.example.com:389* Verhalten der Eigenschaft: - *erforderlich*, wenn `idp_type` auf ”User directory of type LDAP” gesetzt ist

Eigenschaft	Typ	Beschreibung
port	integer	Port des LDAP-Servers. Verhalten der Eigenschaft: - *erforderlich*, wenn `idp_type` auf "User directory of type LDAP" gesetzt ist
base_dn	string	Basispfad des LDAP-Benutzerverzeichnisses zu den Benutzerkonten. Beispiele: *ou=Users,dc=example,dc=org* *ou=Users,ou=system* (für OpenLDAP) *DC=company,DC=com* (für Microsoft Active Directory) *uid=%{user},dc=example,dc=com* (für direkte Benutzerbindung; der Platzhalter "*%{user}*" ist obligatorisch) Verhalten der Eigenschaft: - *erforderlich*, wenn `idp_type` auf "User directory of type LDAP" gesetzt ist
search_attribute	string	Attribut des LDAP-Benutzerverzeichnisses, anhand dessen das Benutzerkonto anhand der im Anmeldeantrag bereitgestellten Informationen identifiziert wird. Beispiele: *uid* (für OpenLDAP) *sAMAccountName* (für Microsoft Active Directory) Verhalten der Eigenschaft: - *erforderlich*, wenn `idp_type` auf "User directory of type LDAP" gesetzt ist
bind_dn	string	Konto des LDAP-Servers zum Binden und Durchsuchen des LDAP-Servers. Bei direkter Benutzerbindung und anonymer Bindung muss `bind_dn` leer sein. Beispiele: *uid=ldap_search,ou=system* (für OpenLDAP) *CN=ldap_search,OU=user_group,DC=company,DC=com* (für Microsoft Active Directory) *CN=Admin,OU=Users,OU=Zabbix,DC=zbx,DC=local* Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf "User directory of type LDAP" gesetzt ist
bind_password	string	LDAP-Passwort des Kontos zum Binden und Durchsuchen des LDAP-Servers. Bei direkter Benutzerbindung und anonymer Bindung muss `bind_password` leer sein. Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf "User directory of type LDAP" gesetzt ist
description	string	Beschreibung des Benutzerverzeichnisses. Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf "User directory of type LDAP" gesetzt ist

Eigenschaft	Typ	Beschreibung
group_basedn	string	Basispfad des LDAP-Benutzerverzeichnisses zu den Gruppen; wird zur Konfiguration einer Prüfung der Benutzerzugehörigkeit im LDAP-Benutzerverzeichnis verwendet. Bei der Bereitstellung eines Benutzers ignoriert, wenn `group_membership` gesetzt ist. Beispiel: *ou=Groups,dc=example,dc=com* Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf "User directory of type LDAP" gesetzt ist
group_filter	string	Filters Zeichenfolge zum Abrufen der LDAP-Benutzerverzeichnisgruppen, denen der Benutzer angehört; wird zur Konfiguration einer Prüfung der Benutzerzugehörigkeit im LDAP-Benutzerverzeichnis verwendet. Bei der Bereitstellung eines Benutzers ignoriert, wenn `group_membership` gesetzt ist. Unterstützte Platzhalter für `group_filter`: *%{attr}* - Suchattribut (ersetzt durch den Eigenschaftswert von `search_attribute`); *%{groupattr}* - Gruppenattribut (ersetzt durch den Eigenschaftswert von `group_member`); *%{host}* - Hostname, IP-Adresse oder URI des LDAP-Servers (ersetzt durch den Eigenschaftswert von `host`); *%{user}* - Benutzername des Zabbix-Benutzers. Standard: *(%{groupattr}=%{user})* Beispiele: - *(member=uid=%{ref},ou=Users,dc=example,dc=com)* stimmt mit "User1" überein, wenn ein LDAP-Gruppenobjekt das Attribut "*member*" mit dem Wert "*uid=User1,ou=Users,dc=example,dc=com*" enthält, und gibt die Gruppe zurück, der "User1" angehört; - *(%{groupattr}=cn=%{ref},ou=Users,ou=Zabbix,DC=example,DC=com)* stimmt mit "User1" überein, wenn ein LDAP-Gruppenobjekt das in der Eigenschaft `group_member` angegebene Attribut mit dem Wert "*cn=User1,ou=Users,ou=Zabbix,DC=example,DC=com*" enthält, und gibt die Gruppe zurück, der "User1" angehört. Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf "User directory of type LDAP" gesetzt ist
group_member	string	Attribut des LDAP-Benutzerverzeichnisses, das Informationen über die Gruppenmitglieder enthält; wird zur Konfiguration einer Prüfung der Benutzerzugehörigkeit im LDAP-Benutzerverzeichnis verwendet. Bei der Bereitstellung eines Benutzers ignoriert, wenn `group_membership` gesetzt ist. Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf "User directory of type LDAP" gesetzt ist

Eigenschaft	Typ	Beschreibung
group_membership	string	Attribut des LDAP-Benutzerverzeichnisses, das Informationen über die Gruppen enthält, denen ein Benutzer angehört. Beispiel: *memberOf* Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf "User directory of type LDAP" gesetzt ist
search_filter	string	Benutzerdefinierte Filters Zeichenfolge zum Auffinden und Authentifizieren eines Benutzers in einem LDAP-Benutzerverzeichnis anhand der im Anmeldeantrag bereitgestellten Informationen. Unterstützte Platzhalter für `search_filter`: *%{attr}* - Name des Suchattributs (z. B. *uid*, *sAMAccountName*); *%{user}* - Benutzername des Zabbix-Benutzers. Standard: *(%{attr}=%{user})* Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf "User directory of type LDAP" gesetzt ist
start_tls	integer	LDAP-Serverkonfigurationsoption, die es ermöglicht, die Kommunikation mit dem LDAP-Server mithilfe von Transport Layer Security (TLS) zu sichern. Beachten Sie, dass `start_tls` für Hosts, die das Protokoll `ldaps://` verwenden, auf "Disabled" gesetzt sein muss. Mögliche Werte: 0 - *(Standard)* Disabled; 1 - Enabled. Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf "User directory of type LDAP" gesetzt ist
user_ref_attr	string	Attribut des LDAP-Benutzerverzeichnisses, das zur Referenzierung eines Benutzerobjekts verwendet wird. Der Wert von `user_ref_attr` wird verwendet, um Werte aus dem angegebenen Attribut im Benutzerverzeichnis abzurufen und sie anstelle des Platzhalters *%{ref}* in der Zeichenfolge `group_filter` einzusetzen. Beispiele: *cn*, *uid*, *member*, *uniqueMember* Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf "User directory of type LDAP" gesetzt ist
SAML-spezifische Eigenschaften:		
idp_entityid	string	URI, die den Identitätsanbieter identifiziert und zur Kommunikation mit dem Identitätsanbieter in SAML-Nachrichten verwendet wird. Beispiel: *https://idp.example.com/idp* Verhalten der Eigenschaft: - *erforderlich*, wenn `idp_type` auf "User directory of type SAML" gesetzt ist

Eigenschaft	Typ	Beschreibung
sp_entityid	string	URL oder beliebige Zeichenfolge, die den Service Provider des Identitätsanbieters identifiziert. Beispiele: *https://idp.example.com/sp* *zabbix* Verhalten der Eigenschaft: - *erforderlich*, wenn `idp_type` auf "User directory of type SAML" gesetzt ist
username_attribute	string	Attribut des SAML-Benutzerverzeichnisses (auch SCIM-Attribut, wenn `scim_status` auf "SCIM provisioning is enabled" gesetzt ist), das den Benutzernamen enthält. Dieser wird bei der Authentifizierung mit dem Wert der Eigenschaft `username` des Benutzerobjekts verglichen. Beispiele: *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* Verhalten der Eigenschaft: - *erforderlich*, wenn `idp_type` auf "User directory of type SAML" gesetzt ist
sso_url	string	URL des SAML-Single-Sign-on-Dienstes des Identitätsanbieters, an den Zabbix die SAML-Authentifizierungsanfragen sendet. Beispiel: *http://idp.example.com/idp/sso/saml* Verhalten der Eigenschaft: - *erforderlich*, wenn `idp_type` auf "User directory of type SAML" gesetzt ist
slo_url	string	URL des SAML-Single-Log-out-Dienstes des Identitätsanbieters, an den Zabbix die SAML-Abmeldeanfragen sendet. Beispiel: *https://idp.example.com/idp/slo/saml* Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf "User directory of type SAML" gesetzt ist
encrypt_nameid	integer	Gibt an, ob die SAML-Namens-ID verschlüsselt werden soll. Mögliche Werte: 0 - *(Standard)* Namens-ID nicht verschlüsseln; 1 - Namens-ID verschlüsseln. Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf "User directory of type SAML" gesetzt ist
encrypt_assertions	integer	Gibt an, ob die SAML-Assertions verschlüsselt werden sollen. Mögliche Werte: 0 - *(Standard)* Assertions nicht verschlüsseln; 1 - Assertions verschlüsseln. Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf "User directory of type SAML" gesetzt ist

Eigenschaft	Typ	Beschreibung
nameid_format	string	Namens-ID-Format des Service Providers des SAML-Identitätsanbieters. Beispiele: *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* Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf ”User directory of type SAML” gesetzt ist
scim_status	integer	Gibt an, ob die SCIM-Bereitstellung für SAML aktiviert oder deaktiviert ist. Mögliche Werte: 0 - *(Standard)* SCIM-Bereitstellung ist deaktiviert; 1 - SCIM-Bereitstellung ist aktiviert. Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf ”User directory of type SAML” gesetzt ist
sign_assertions	integer	Gibt an, ob die SAML-Assertions mit einer SAML-Signatur signiert werden sollen. Mögliche Werte: 0 - Assertions nicht signieren; 1 - *(Standard)* Assertions signieren. Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf ”User directory of type SAML” gesetzt ist
sign_authn_requests	integer	Gibt an, ob die SAML-AuthN-Anfragen mit einer SAML-Signatur signiert werden sollen. Mögliche Werte: 0 - AuthN-Anfragen nicht signieren; 1 - *(Standard)* AuthN-Anfragen signieren. Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf ”User directory of type SAML” gesetzt ist
sign_messages	integer	Gibt an, ob die SAML-Nachrichten mit einer SAML-Signatur signiert werden sollen. Mögliche Werte: 0 - Nachrichten nicht signieren; 1 - *(Standard)* Nachrichten signieren. Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf ”User directory of type SAML” gesetzt ist
sign_logout_requests	integer	Gibt an, ob die SAML-Abmeldeanfragen mit einer SAML-Signatur signiert werden sollen. Mögliche Werte: 0 - Abmeldeanfragen nicht signieren; 1 - *(Standard)* Abmeldeanfragen signieren. Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf ”User directory of type SAML” gesetzt ist

Eigenschaft	Typ	Beschreibung
sign_logout_responses	integer	Gibt an, ob die SAML-Abmeldeantworten mit einer SAML-Signatur signiert werden sollen. Mögliche Werte: 0 - Abmeldeantworten nicht signieren; 1 - *(Standard)* Abmeldeantworten signieren. Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf "User directory of type SAML" gesetzt ist
idp_certificate	string	Inhalt des Service-Provider-Zertifikats (SP) für die Einrichtung des SAML-Single-Sign-on-Dienstes (SSO). Beispiele: *-----BEGIN CERTIFICATE----- ...more encoded data... -----END CERTIFICATE-----* Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf "User directory of type SAML" gesetzt ist und `$SSO['CERT_STORAGE']` in der Frontend-Konfigurationsdatei auf `database` gesetzt ist - *nur schreibbar*
sp_private_key	string	Inhalt des privaten Schlüssels des Service Providers (SP) für die Einrichtung des SAML-Single-Sign-on-Dienstes (SSO). Er ermöglicht eine sichere Authentifizierung und einen sicheren Datenaustausch mit dem Identitätsanbieter (IdP). Beispiele: *-----BEGIN CERTIFICATE----- ...more encoded data... -----END CERTIFICATE-----* Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf "User directory of type SAML" gesetzt ist und `$SSO['CERT_STORAGE']` in der Frontend-Konfigurationsdatei auf `database` gesetzt ist - *nur schreibbar*
sp_certificate	string	Inhalt des Service-Provider-Zertifikats (SP) für die Einrichtung des SAML-Single-Sign-on-Dienstes (SSO). Beispiele: *-----BEGIN CERTIFICATE----- ...more encoded data... -----END CERTIFICATE-----* Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf "User directory of type SAML" gesetzt ist und `$SSO['CERT_STORAGE']` in der Frontend-Konfigurationsdatei auf `database` gesetzt ist - *nur schreibbar*
idp_certificate_hash	string	Der MD5-Hash des Werts von idp_certificate. Für ein leeres idp_certificate wird eine leere Zeichenfolge zurückgegeben. Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf "User directory of type SAML" gesetzt ist und `$SSO['CERT_STORAGE']` in der Frontend-Konfigurationsdatei auf `database` gesetzt ist - *schreibgeschützt*

Eigenschaft	Typ	Beschreibung
sp_private_key_hash	string	Der MD5-Hash des Werts von sp_private_key. Für einen leeren sp_private_key wird eine leere Zeichenfolge zurückgegeben. Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf "User directory of type SAML" gesetzt ist und `$SSO['CERT_STORAGE']` in der Frontend-Konfigurationsdatei auf `database` gesetzt ist - *schreibgeschützt*
sp_certificate_hash	string	Der MD5-Hash des Werts von sp_certificate. Für ein leeres sp_certificate wird eine leere Zeichenfolge zurückgegeben. Verhalten der Eigenschaft: - *unterstützt*, wenn `idp_type` auf "User directory of type SAML" gesetzt ist und `$SSO['CERT_STORAGE']` in der Frontend-Konfigurationsdatei auf `database` gesetzt ist - *schreibgeschützt*

Zuordnungen von Medientypen

Das Objekt für Medientyp-Zuordnungen hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
userdirectory_mediaid	ID	ID der Medientyp-Zuordnung. Verhalten der Eigenschaft: - *schreibgeschützt*
name	string	Sichtbarer Name in der Liste der Medientyp-Zuordnungen. Verhalten der Eigenschaft: - *erforderlich*
mediatypeid	ID	ID des zu erstellenden Medientyps; wird als Wert für die Eigenschaft `mediatypeid` des Media-Objekts verwendet. Verhalten der Eigenschaft: - *erforderlich*
attribute	string	LDAP-/SAML-Benutzerverzeichnisattribut (auch SCIM-Attribut, wenn `scim_status` auf "SCIM provisioning is enabled" gesetzt ist), das das Medium des Benutzers enthält (z. B. *user@example.com*), das als Wert für die Eigenschaft `sendto` des Media-Objekts verwendet wird. Falls es in den vom LDAP-/SAML-Identitätsanbieter empfangenen Daten vorhanden ist und der Wert nicht leer ist, löst dies die Erstellung eines Mediums für den bereitgestellten Benutzer aus. Verhalten der Eigenschaft: - *erforderlich*
active	integer	Wert der Benutzermedien-Eigenschaft `active`, wenn für den bereitgestellten Benutzer ein Medium erstellt wird. Mögliche Werte: 0 - *(Standard)* aktiviert; 1 - deaktiviert.
severity	integer	Wert der Benutzermedien-Eigenschaft `severity`, wenn für den bereitgestellten Benutzer ein Medium erstellt wird. Standard: 63.
period	string	Wert der Benutzermedien-Eigenschaft `period`, wenn für den bereitgestellten Benutzer ein Medium erstellt wird. Standard: 1-7,00:00-24:00.

Zuordnungen von Provisioning-Gruppen

Die Zuordnungen von Provisioning-Gruppen haben die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
name	string	Vollständiger Name einer Gruppe (z. B. *Zabbix administrators*) im LDAP-/SAML-Benutzerverzeichnis (auch SCIM, wenn `scim_status` auf „SCIM-Provisionierung ist aktiviert“ gesetzt ist). Unterstützt das Platzhalterzeichen „*“. Eindeutig über alle Zuordnungen von Provisioning-Gruppen hinweg. Verhalten der Eigenschaft: - *erforderlich*
roleid	ID	ID der Benutzerrolle, die dem Benutzer zugewiesen werden soll. Wenn mehrere Zuordnungen von Provisioning-Gruppen übereinstimmen, wird dem Benutzer die Rolle mit dem höchsten Benutzertyp (*User*, *Admin* oder *Super admin*) zugewiesen. Wenn es mehrere Rollen mit demselben Benutzertyp gibt, wird dem Benutzer die erste Rolle (alphabetisch sortiert) zugewiesen. Verhalten der Eigenschaft: - *erforderlich*
user_groups	array	Array von Zabbix-Benutzergruppen-ID-Objekten. Jedes Objekt hat die folgenden Eigenschaften: `usrgrpid` - `(ID)` ID der Zabbix-Benutzergruppe, die dem Benutzer zugewiesen werden soll. Wenn mehrere Zuordnungen von Provisioning-Gruppen übereinstimmen, werden dem Benutzer die Zabbix-Benutzergruppen aller übereinstimmenden Zuordnungen zugewiesen. Verhalten der Eigenschaft: - *erforderlich*

userdirectory.create

Beschreibung

`object userdirectory.create(object/array userDirectory)`

Diese Methode ermöglicht es Ihnen, neue Benutzerverzeichnisse zu erstellen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar.

Parameter

`(object/array)` Zu erstellende Benutzerverzeichnisse.

Die Methode akzeptiert Benutzerverzeichnisse mit den Standard-Benutzerverzeichniseigenschaften.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Benutzerverzeichnisse in der Eigenschaft `userdirectoryids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Benutzerverzeichnisse.

Beispiele

Erstellen eines Benutzerverzeichnisses

Erstellen Sie ein Benutzerverzeichnis, um Benutzer mit StartTLS über LDAP zu authentifizieren. Beachten Sie, dass zur Authentifizierung von Benutzern über LDAP die LDAP-Authentifizierung aktiviert sein muss.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "3"
        ]
    },
    "id": 1
}
```

Erstellen eines Benutzerverzeichnisses (JIT-Provisionierung aktiviert)

Erstellen Sie ein Benutzerverzeichnis, um Benutzer über LDAP zu authentifizieren (mit aktivierter JIT-Provisionierung). Beachten Sie, dass zur Authentifizierung von Benutzern über LDAP die LDAP-Authentifizierung aktiviert sein muss.

Anfrage:

```
{
    "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
}
```


Antwort:


```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "2"
        ]
    },
    "id": 1
}
```


Quelle

CUserDirectory::create() in *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.delete

Beschreibung

`object userdirectory.delete(array userDirectoryIds)`

Diese Methode ermöglicht es Ihnen, Benutzerverzeichnisse zu löschen. Ein Benutzerverzeichnis kann nicht gelöscht werden, wenn es direkt für mindestens eine Benutzergruppe verwendet wird.
 Das standardmäßige LDAP-Benutzerverzeichnis kann nicht gelöscht werden, wenn `authentication.ldap_configured` auf 1 gesetzt ist oder wenn keine weiteren Benutzerverzeichnisse mehr vorhanden sind.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar.

Parameter

`(array)` IDs der zu löschenden Benutzerverzeichnisse.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Benutzerverzeichnisse in der Eigenschaft `userdirectoryids` enthält.

Beispiele

Mehrere Benutzerverzeichnisse löschen

Löschen Sie zwei Benutzerverzeichnisse.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.delete",
    "params": [
        "2",
        "12"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "2",
            "12"
        ]
    },
    "id": 1
}
```

Quelle

CUserDirectory::delete() in *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.get

Beschreibung

`integer/array userdirectory.get(object parameters)`

Die Methode ermöglicht es Ihnen, Benutzerverzeichnisse anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist nur für Benutzertypen vom Typ *Super admin* verfügbar.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
userdirectoryids	ID/array	Gibt nur Benutzerverzeichnisse mit den angegebenen IDs zurück.
selectUsrgrps	query	Gibt eine Eigenschaft `usrgrps` mit Benutzergruppen zurück, die einem Benutzerverzeichnis zugeordnet sind. Unterstützt `count`.
selectProvisionMedia	query	Gibt eine Eigenschaft `provision_media` mit Medientyp-Zuordnungen zurück, die einem Benutzerverzeichnis zugeordnet sind.
selectProvisionGroups	query	Gibt eine Eigenschaft `provision_groups` mit Zuordnungen von Provisioning-Gruppen zurück, die einem Benutzerverzeichnis zugeordnet sind.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `name`.

Parameter	Type	Beschreibung
filter	object	Gibt nur Ergebnisse zurück, die exakt mit dem angegebenen Filter übereinstimmen. Akzeptiert ein Objekt, bei dem die Schlüssel Eigenschaftsnamen sind und die Werte entweder ein einzelner Wert oder ein Array von Werten sind. Unterstützte Eigenschaften: `userdirectoryid`, `idp_type`, `provision_status`.
search	object	Gibt Ergebnisse zurück, die dem angegebenen Muster entsprechen (Groß-/Kleinschreibung wird nicht beachtet). Akzeptiert ein Objekt, bei dem die Schlüssel Eigenschaftsnamen sind und die Werte Zeichenfolgen sind, nach denen gesucht werden soll. Wenn keine zusätzlichen Optionen angegeben sind, wird eine Suche vom Typ `LIKE "%…%"` durchgeführt. Unterstützte Eigenschaften: `name`, `description`. Ein Benutzerverzeichnis vom Typ SAML hat sowohl für die Eigenschaften `name` als auch `description` einen leeren Wert. Beide Eigenschaften können mit der Operation `userdirectory.update` geändert werden.
countOutput	boolean	Diese Parameter werden in der Referenzkommentierung beschrieben.
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Gibt entweder Folgendes zurück:

- ein Array von Objekten
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde

Beispiele

Abrufen von Benutzerverzeichnissen

Rufen Sie alle Benutzerverzeichnisse mit zusätzlichen Eigenschaften ab, die Zuordnungen von Medientypen und Provisionierungsgruppen anzeigen, die jedem Benutzerverzeichnis zugeordnet sind.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.get",
    "params": {
        "output": "extend",
        "selectProvisionMedia": "extend",
        "selectProvisionGroups": "extend"
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```

Siehe auch

- Benutzergruppe

Quelle

CUserDirectory::get() in *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.test

Beschreibung

`object userdirectory.test(array userDirectory)`

Mit dieser Methode können Sie die Verbindungseinstellungen des Benutzerverzeichnisses testen.

Note:
Mit dieser Methode können Sie auch testen, welche konfigurierten Daten mit den Einstellungen des Benutzerverzeichnisses für die Benutzerbereitstellung übereinstimmen (z. B. welche Benutzerrolle, welche Benutzergruppen und welche Benutzermedien dem Benutzer zugewiesen werden). Für diese Art von Test sollte die API-Anfrage für ein Benutzerverzeichnis ausgeführt werden, bei dem `provision_status` auf aktiviert gesetzt ist.

Note:
Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar.

Parameter

`(object)` Eigenschaften des Benutzerverzeichnisses.

Da die API `userdirectory.get` das Feld `bind_password` nicht zurückgibt, sollten `userdirectoryid` und/oder `bind_password` angegeben werden.
 Zusätzlich zu den Standard-Eigenschaften des Benutzerverzeichnisses akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
test_username	string	Benutzername, der im Benutzerverzeichnis getestet werden soll.
test_password	string	Dem Benutzernamen zugeordnetes Passwort, das im Benutzerverzeichnis getestet werden soll.

Rückgabewerte

`(bool)` Gibt bei Erfolg true zurück.

Beispiele

Benutzerverzeichnis für vorhandenen Benutzer testen

Benutzerverzeichnis „3“ für „user1“ testen.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```

Benutzerverzeichnis für nicht vorhandenen Benutzer testen

Testen Sie das Benutzerverzeichnis „3“ für den nicht vorhandenen Benutzer „user2“.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "error": {
        "code": -32500,
        "message": "Application error.",
        "data": "Incorrect user name or password or account is temporarily blocked."
    },
    "id": 1
}
```

Benutzerverzeichnis für die Benutzerbereitstellung testen

Testen Sie das Benutzerverzeichnis „3“, um festzustellen, welche konfigurierten Daten mit den Einstellungen des Benutzerverzeichnisses für die Bereitstellung von „user3“ übereinstimmen (z. B. welche Benutzerrolle, Benutzergruppen und Benutzermedien dem Benutzer zugewiesen werden).

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "username": "user3",
        "name": "John",
        "surname": "Doe",
        "medias": [],
        "usrgrps": [
            {
                "usrgrpid": "8"
            },
            {
                "usrgrpid": "7"
            }
        ],
        "roleid": "2",
        "userdirectoryid": "2"
    },
    "id": 1
}
```

Quelle

CUserDirectory::test() in *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.update

Beschreibung

`object userdirectory.update(object/array userDirectory)`

Diese Methode ermöglicht es Ihnen, vorhandene Benutzerverzeichnisse zu aktualisieren.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar.

Parameter

`(object/array)` zu aktualisierende Eigenschaften des Benutzerverzeichnisses.

Die Eigenschaft `userdirectoryid` muss für jedes Benutzerverzeichnis definiert werden, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Benutzerverzeichnisse unter der Eigenschaft `userdirectoryids` enthält.

Beispiele

Bind-Passwort für Benutzerverzeichnis aktualisieren

Legen Sie ein neues Bind-Passwort für ein Benutzerverzeichnis fest.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.update",
    "params": {
        "userdirectoryid": "3",
        "bind_password": "newldappassword"
```

```
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "3"
        ]
    },
    "id": 1
}
```

Zuordnungen für Benutzerverzeichnis aktualisieren

Aktualisieren Sie die Bereitstellungsgruppen-Zuordnungen und Medientyp-Zuordnungen für das Benutzerverzeichnis „2“.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "2"
        ]
    },
    "id": 1
}
```

Quelle

CUserDirectory::update() in *ui/include/classes/api/services/CUserDirectory.php*.

Bereinigung

Diese Klasse ist für die Arbeit mit der Bereinigung vorgesehen.

Objektreferenzen:

- Bereinigung

Verfügbare Methoden:

- housekeeping.get - Bereinigung abrufen
- housekeeping.update - Bereinigung aktualisieren

Hauswirtschaftliches Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `housekeeping` API.

Housekeeping

Das Einstellungsobjekt verfügt über die folgenden Eigenschaften.

Property	Type	Description
hk_events_mode	integer	Aktiviert das interne Housekeeping für Ereignisse und Alarme. Mögliche Werte: 0 - Deaktiviert; 1 - *(Standard)* Aktiviert.
hk_events_trigger	string	Speicherzeitraum für Auslöserdaten. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300) oder einen Wert mit einem Zeitsuffix (z. B. `30s`, `1m`, `2h`, `1d`). Standard: 365d.
hk_events_service	string	Speicherzeitraum für Servicedaten. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300) oder einen Wert mit einem Zeitsuffix (z. B. `30s`, `1m`, `2h`, `1d`). Standard: 1d.
hk_events_internal	string	Speicherzeitraum für interne Daten. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300) oder einen Wert mit einem Zeitsuffix (z. B. `30s`, `1m`, `2h`, `1d`). Standard: 1d.
hk_events_discovery	string	Speicherzeitraum für Netzwerkerkennungsdaten. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300) oder einen Wert mit einem Zeitsuffix (z. B. `30s`, `1m`, `2h`, `1d`). Standard: 1d.
hk_events_autoreg	string	Speicherzeitraum für Autoregistrierungsdaten. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300) oder einen Wert mit einem Zeitsuffix (z. B. `30s`, `1m`, `2h`, `1d`). Standard: 1d.
hk_services_mode	integer	Aktiviert das interne Housekeeping für Services. Mögliche Werte: 0 - Deaktiviert; 1 - *(Standard)* Aktiviert.
hk_services	string	Speicherzeitraum für Servicedaten. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300) oder einen Wert mit einem Zeitsuffix (z. B. `30s`, `1m`, `2h`, `1d`). Standard: 365d.

Property	Type	Description
hk_audit_mode	integer	Aktiviert das interne Housekeeping für Auditdaten. Mögliche Werte: 0 - Deaktiviert; 1 - *(Standard)* Aktiviert.
hk_audit	string	Speicherzeitraum für Auditdaten. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300) oder einen Wert mit einem Zeitsuffix (z. B. 30s, 1m, 2h, 1d). Standard: 31d.
hk_sessions_mode	integer	Aktiviert das interne Housekeeping für Sitzungen. Mögliche Werte: 0 - Deaktiviert; 1 - *(Standard)* Aktiviert.
hk_sessions	string	Speicherzeitraum für Sitzungsdaten. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300) oder einen Wert mit einem Zeitsuffix (z. B. 30s, 1m, 2h, 1d). Standard: 31d.
hk_history_mode	integer	Aktiviert das interne Housekeeping für den Verlauf. Mögliche Werte: 0 - Deaktiviert; 1 - *(Standard)* Aktiviert.
hk_history_global	integer	Überschreibt den Verlaufszeitraum des Datenpunkts. Mögliche Werte: 0 - Nicht überschreiben; 1 - *(Standard)* Überschreiben.
hk_history	string	Speicherzeitraum für Verlaufsdaten. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300) oder einen Wert mit einem Zeitsuffix (z. B. 30s, 1m, 2h, 1d). Standard: 31d.
hk_trends_mode	integer	Aktiviert das interne Housekeeping für Trends. Mögliche Werte: 0 - Deaktiviert; 1 - *(Standard)* Aktiviert.
hk_trends_global	integer	Überschreibt den Trendzeitraum des Datenpunkts. Mögliche Werte: 0 - Nicht überschreiben; 1 - *(Standard)* Überschreiben.
hk_trends	string	Speicherzeitraum für Trenddaten. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300) oder einen Wert mit einem Zeitsuffix (z. B. 30s, 1m, 2h, 1d). Standard: 365d.
db_extension	string	Konfigurationskennzeichen für die DB-Erweiterung. Wenn dieses Kennzeichen auf ”timescaledb” gesetzt ist, ändert der Server sein Verhalten beim Housekeeping und beim Löschen von Datenpunkten. Eigenschaftsverhalten: - *schreibgeschützt*

Property	Type	Description
compression_availability	integer	Gibt an, ob die Datenbank (oder ihre Erweiterung) Datenkomprimierung unterstützt. Mögliche Werte: 0 - Nicht verfügbar; 1 - Verfügbar. Eigenschaftsverhalten: - *schreibgeschützt*
compression_status	integer	Aktiviert die TimescaleDB-Komprimierung für Verlauf und Trends. Mögliche Werte: 0 - *(Standard)* Aus; 1 - Ein.
compress_older	string	Komprimiert Verlaufs- und Trenddatensätze, die älter als der angegebene Zeitraum sind. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300) oder einen Wert mit einem Zeitsuffix (z. B. 30s, 1m, 2h, 1d). Standard: 7d.

housekeeping.get

Beschreibung

`object housekeeping.get(object parameters)`

Die Methode ermöglicht es Ihnen, ein Housekeeping-Objekt anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt nur einen Parameter.

Parameter	Type	Beschreibung
output	query	Dieser Parameter wird im Referenzkommentar beschrieben.

Rückgabewerte

`(object)` Liefert ein Housekeeping-Objekt zurück.

Beispiele

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "housekeeping.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```


Quelle

CHousekeeping ::get() in *ui/include/classes/api/services/CHousekeeping.php*.

housekeeping.update

Beschreibung

`object housekeeping.update(object housekeeping)`

Diese Methode ermöglicht es Ihnen, vorhandene Housekeeping-Einstellungen zu aktualisieren.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Siehe Benutzerrollen für weitere Informationen.

Parameter

`(object)` zu aktualisierende Housekeeping-Eigenschaften.

Rückgabewerte

`(array)` Gibt ein Array mit den Namen der aktualisierten Parameter zurück.

Beispiele

Anfrage:


```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        "hk_events_mode",
        "hk_events_trigger",
        "hk_events_internal",
        "hk_events_discovery"
    ],
    "id": 1
}
```

Quelle

CHousekeeping::update() in *ui/include/classes/api/services/CHousekeeping.php*.

Bericht

Diese Klasse ist für die Arbeit mit geplanten Berichten vorgesehen.

Objektreferenzen:

- Report
- Benutzer
- Benutzergruppen

Verfügbare Methoden:

- report.create - neue geplante Berichte erstellen
- report.delete - geplante Berichte löschen
- report.get - geplante Berichte abrufen
- report.update - geplante Berichte aktualisieren

Berichts-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `report` API.

Bericht

Das Berichtsobjekt verfügt über die folgenden Eigenschaften:

Eigenschaft	Typ	Beschreibung
reportid	ID	ID des Berichts. Eigenschaftsverhalten: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
userid	ID	ID des Benutzers, der den Bericht erstellt hat. Eigenschaftsverhalten: - *erforderlich* für Erstellungsvorgänge
name	string	Eindeutiger Name des Berichts. Eigenschaftsverhalten: - *erforderlich* für Erstellungsvorgänge
dashboardid	ID	ID des Dashboards, auf dem der Bericht basiert. Eigenschaftsverhalten: - *erforderlich* für Erstellungsvorgänge

Eigenschaft	Typ	Beschreibung
period	integer	Zeitraum, für den der Bericht erstellt wird. Mögliche Werte: 0 - *(Standard)* vorheriger Tag; 1 - vorherige Woche; 2 - vorheriger Monat; 3 - vorheriges Jahr.
cycle	integer	Wiederholungsintervall des Zeitplans. Mögliche Werte: 0 - *(Standard)* täglich; 1 - wöchentlich; 2 - monatlich; 3 - jährlich.
start_time	integer	Tageszeit in Sekunden, zu der der Bericht für den Versand erstellt wird. Standard: 0.
weekdays	integer	Wochentage, an denen der Bericht versendet wird. Mögliche Bitmap-Werte: 1 - Montag; 2 - Dienstag; 4 - Mittwoch; 8 - Donnerstag; 16 - Freitag; 32 - Samstag; 64 - Sonntag. Dies ist ein Bitmaskenfeld; jede Summe möglicher Bitmap-Werte ist zulässig (beispielsweise 21 für Montag, Mittwoch und Freitag). Standard: 0. Eigenschaftsverhalten: - *erforderlich*, wenn `cycle` auf "weekly" gesetzt ist.
active_since	string	Datum, an dem begonnen werden soll. Mögliche Werte: leere Zeichenfolge - *(Standard)* nicht angegeben (gespeichert als 0); bestimmtes Datum im Format JJJJ-MM-TT (gespeichert als Zeitstempel des Tagesbeginns (00:00:00)).
active_till	string	Datum, an dem beendet werden soll. Mögliche Werte: leere Zeichenfolge - *(Standard)* nicht angegeben (gespeichert als 0); bestimmtes Datum im Format JJJJ-MM-TT (gespeichert als Zeitstempel des Tagesendes (23:59:59)).
subject	string	Betreff der Berichtsnachricht.
message	string	Text der Berichtsnachricht.
status	integer	Gibt an, ob der Bericht aktiviert oder deaktiviert ist. Mögliche Werte: 0 - *(Standard)* aktiviert; 1 - deaktiviert.
description	text	Beschreibung des Berichts.

Eigenschaft	Typ	Beschreibung
state	integer	Status des Berichts. Mögliche Werte: 0 - *(Standard)* Bericht wurde noch nicht verarbeitet; 1 - Bericht wurde erstellt und erfolgreich an alle Empfänger gesendet; 2 - Erstellung des Berichts ist fehlgeschlagen; ”info” enthält Fehlerinformationen; 3 - Bericht wurde erstellt, aber der Versand an einige oder alle Empfänger ist fehlgeschlagen; ”info” enthält Fehlerinformationen. Eigenschaftsverhalten: - *schreibgeschützt*
lastsent	timestamp	Unix-Zeitstempel des zuletzt erfolgreich gesendeten Berichts. Eigenschaftsverhalten: - *schreibgeschützt*
info	string	Fehlerbeschreibung oder zusätzliche Informationen. Eigenschaftsverhalten: - *schreibgeschützt*

Benutzer

Das Benutzerobjekt hat die folgenden Eigenschaften:

Eigenschaft	Typ	Beschreibung
userid	ID	ID des Benutzers, an den der Bericht gesendet werden soll. Verhalten der Eigenschaft: - *erforderlich*
access_userid	ID	ID des Benutzers, in dessen Namen der Bericht erstellt wird. 0 - *(Standard)* Bericht durch den Empfänger erstellen.
exclude	integer	Gibt an, ob der Benutzer von der Mailingliste ausgeschlossen werden soll. Mögliche Werte: 0 - *(Standard)* Einschließen; 1 - Ausschließen.

Benutzergruppen

Das Benutzergruppen-Objekt hat die folgenden Eigenschaften:

Eigenschaft	Typ	Beschreibung
usrgrpid	ID	ID der Benutzergruppe, an die der Bericht gesendet wird. Verhalten der Eigenschaft: - *erforderlich*
access_userid	ID	ID des Benutzers, in dessen Namen der Bericht erstellt wird. 0 - *(Standard)* Bericht nach Empfänger erstellen.

report.create

Beschreibung

`object report.create(object/array reports)`

Diese Methode ermöglicht es Ihnen, neue geplante Berichte zu erstellen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellende geplante Berichte.

Zusätzlich zu den standardmäßigen Eigenschaften geplanter Berichte akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
users	object/array	Benutzer, an die der Bericht gesendet werden soll. Parameter behavior: - *erforderlich*, wenn `user_groups` nicht gesetzt ist
user_groups	object/array	Benutzergruppen, an die der Bericht gesendet werden soll. Parameter behavior: - *erforderlich*, wenn `users` nicht gesetzt ist

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten geplanten Berichte in der Eigenschaft `reportids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen geplanten Berichte.

Beispiele

Erstellen eines geplanten Berichts

Erstellen Sie einen wöchentlichen Bericht, der für die vorherige Woche von Montag bis Freitag jeweils um 12:00 Uhr im Zeitraum vom 2021-04-01 bis 2021-08-31 erstellt wird.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "reportids": [
            "1"
        ]
    },
    "id": 1
}
```

Siehe auch

- Benutzer
- Benutzergruppen

Quelle

CReport::create() in *ui/include/classes/api/services/CReport.php*.

report.delete

Beschreibung

`object report.delete(array reportids)`

Diese Methode ermöglicht es Ihnen, geplante Berichte zu löschen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der geplanten Berichte, die gelöscht werden sollen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten geplanten Berichte in der Eigenschaft `reportids` enthält.

Beispiele

Mehrere geplante Berichte löschen

Löschen Sie zwei geplante Berichte.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "report.delete",
    "params": [
        "1",
        "2"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "reportids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

Quelle

CReport::delete() in *ui/include/classes/api/services/CReport.php*.

report.get

Beschreibung

`integer/array report.get(object parameters)`

Mit dieser Methode können Sie geplante Berichte anhand der angegebenen Parameter abrufen.

Note:
Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
reportids	ID/array	Gibt nur geplante Berichte mit den angegebenen Berichts-IDs zurück.
expired	boolean	Wenn auf `true` gesetzt, werden nur abgelaufene geplante Berichte zurückgegeben.
selectUsers	query	Gibt eine Eigenschaft `users` mit Benutzern zurück, an die der Bericht gesendet werden soll.
selectUserGroups	query	Gibt eine Eigenschaft `user_groups` mit Benutzergruppen zurück, an die der Bericht gesendet werden soll.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `reportid`, `name`, `status`.
countOutput	boolean	Diese Parameter sind in der Referenzkommentierung beschrieben.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Gibt entweder Folgendes zurück:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Abrufen von Berichtsdaten

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "report.get",
    "params": {
        "output": "extend",
        "selectUsers": "extend",
        "selectUserGroups": "extend",
        "reportids": ["1", "2"]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "reportid": "1",
            "userid": "1",
            "name": "Wöchentlicher Bericht",
            "dashboardid": "1",
            "period": "1",
            "cycle": "1",
            "start_time": "43200",
            "weekdays": "31",
            "active_since": "2021-04-01",
            "active_till": "2021-08-31",
            "subject": "Wöchentlicher Bericht",
            "message": "Begleittext des Berichts",
            "status": "1",
            "description": "Beschreibung des Berichts",
            "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": "Monatlicher Bericht",
            "dashboardid": "2",
```

```
            "period": "2",
            "cycle": "2",
            "start_time": "0",
            "weekdays": "0",
            "active_since": "2021-05-01",
            "active_till": "",
            "subject": "Monatlicher Bericht",
            "message": "Begleittext des Berichts",
            "status": "1",
            "description": "",
            "state": "0",
            "lastsent": "0",
            "info": "",
            "users": [
                {
                    "userid": "1",
                    "access_userid": "1",
                    "exclude": "0"
                }
            ],
            "user_groups": []
        }
    ],
    "id": 1
}
```


Siehe auch

- Benutzer
- Benutzergruppen

Quelle

CReport::get() in *ui/include/classes/api/services/CReport.php*.

report.update

Beschreibung

`object report.update(object/array reports)`

Diese Methode ermöglicht es Ihnen, vorhandene geplante Berichte zu aktualisieren.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Eigenschaften des geplanten Berichts, die aktualisiert werden sollen.

Die Eigenschaft `reportid` muss für jeden geplanten Bericht definiert sein, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Zusätzlich zu den Standard-Eigenschaften geplanter Berichte akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
users	object/array	Benutzer, die die aktuell dem geplanten Bericht zugewiesenen Benutzer ersetzen. Parameterverhalten: - *erforderlich*, wenn `user_groups` nicht gesetzt ist

Parameter	Type	Beschreibung
user_groups	object/array	Benutzergruppen, die die aktuell dem geplanten Bericht zugewiesenen Benutzergruppen ersetzen. Parameterverhalten: - *erforderlich*, wenn `users` nicht gesetzt ist

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten geplanten Berichte in der Eigenschaft `reportids` enthält.

Beispiele

Geplanten Bericht deaktivieren

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "report.update",
    "params": {
        "reportid": "1",
        "status": "0"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "reportids": [
            "1"
        ]
    },
    "id": 1
}
```

Siehe auch

- Benutzer
- Benutzergruppen

Quelle

CReport::update() in *ui/include/classes/api/services/CReport.php*.

Bild

Diese Klasse ist für die Arbeit mit Bildern vorgesehen.

Objektreferenzen:

- Image

Verfügbare Methoden:

- image.create - neue Bilder erstellen
- image.delete - Bilder löschen
- image.get - Bilder abrufen
- image.update - Bilder aktualisieren

Bild-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `image` API.

Bild

Das Bildobjekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
imageid	ID	ID des Bildes. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
name	string	Name des Bildes. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
imagetype	integer	Typ des Bildes. Mögliche Werte: 1 - *(Standard)* Symbol; 2 - Hintergrundbild. Verhalten der Eigenschaft: - *konstant* - *erforderlich* für Erstellungsvorgänge
image	string	Base64-kodiertes Bild. Die maximale Größe des kodierten Bildes beträgt 1 MB. Die maximale Größe kann durch Ändern des Konstantenwerts `ZBX_MAX_IMAGE_SIZE` angepasst werden. Unterstützte Bildformate: PNG, JPEG, GIF und WebP. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge

image.create

Beschreibung

`object image.create(object/array images)`

Diese Methode ermöglicht es Ihnen, neue Bilder zu erstellen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellende Bilder.

Die Methode akzeptiert Bilder mit den standardmäßigen Bildeigenschaften.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Bilder unter der Eigenschaft `imageids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Bilder.

Beispiele

Ein Symbolbild erstellen

Erstellen Sie ein Cloud-Symbol.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "image.create",
    "params": {
        "imagetype": 1,
```

```
        "name": "Cloud_(24)",
        "image": "iVBORw0KGgoAAAANSUhEUgAAABgAAAANCAYAAACzbK7QAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzAAACmAAAApgB
    },
    "id": 1
}
```


Antwort:


```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "188"
        ]
    },
    "id": 1
}
```


Ein Hintergrundbild erstellen

Erstellen Sie ein kleines Hintergrundbild, das als Muster verwendet werden soll.

Anfrage:


```
{
    "jsonrpc": "2.0",
    "method": "image.create",
    "params": {
        "imagetype": 2,
        "name": "background_image",
        "image": "iVBORw0KGgoAAAANSUhEUgAAABgAAAAYCAIAAABvFaqvAAABhWlDQ1BJQ0MgcHJvZmlsZQAAKJF9kb1Lw0AYxp+m
    },
    "id": 1
}
```


Antwort:


```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "189"
        ]
    },
    "id": 1
}
```


Quelle

CImage::create() in *ui/include/classes/api/services/CImage.php*.

image.delete

Beschreibung

`object image.delete(array imageIds)`

Diese Methode ermöglicht es Ihnen, Bilder zu löschen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der zu löschenden Bilder.

Rückgabewerte

(object) Gibt ein Objekt zurück, das die IDs der gelöschten Bilder unter der Eigenschaft imageids enthält.

Beispiele

Mehrere Bilder löschen

Löschen Sie zwei Bilder.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "image.delete",
    "params": [
        "188",
        "192"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "188",
            "192"
        ]
    },
    "id": 1
}
```

Quelle

CImage::delete() in *ui/include/classes/api/services/CImage.php*.

image.get

Beschreibung

`integer/array image.get(object parameters)`

Die Methode ermöglicht es Ihnen, Bilder anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

(object) Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
imageids	ID/array	Gibt nur Bilder mit den angegebenen IDs zurück.
sysmapids	ID/array	Gibt Bilder zurück, die auf den angegebenen Karten verwendet werden.
select_image	flag	Gibt eine image-Eigenschaft mit dem Base64-kodierten Bild zurück.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: imageid, name.
countOutput	boolean	Diese Parameter werden im Referenzkommentar beschrieben.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	

Parameter	Type	Beschreibung
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Kann die folgenden Dinge zurück geben:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Ein Bild abrufen

Rufen Sie alle Daten für das Bild mit der ID „2“ ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "image.get",
    "params": {
        "output": "extend",
        "select_image": true,
        "imageids": "2"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "imageid": "2",
            "imagetype": "1",
            "name": "Cloud_(24)",
            "image": "iVBORw0KGgoAAAANSUhEUgAAABgAAAANCAYAAACzbK7QAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzAAACmAAA
        }
    ],
    "id": 1
}
```

Quelle

CImage::get() in *ui/include/classes/api/services/CImage.php*.

image.update

Beschreibung

`object image.update(object/array images)`

Diese Methode ermöglicht es Ihnen, vorhandene Bilder zu aktualisieren.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

(object/array) Zu aktualisierende Bildeigenschaften.

Die Eigenschaft imageid muss für jedes Bild definiert sein, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Die Methode akzeptiert Bilder mit den Standard-Bildeigenschaften.

Rückgabewerte

(object) Gibt ein Objekt zurück, das die IDs der aktualisierten Bilder unter der Eigenschaft imageids enthält.

Beispiele

Bild umbenennen

Benennen Sie das Bild in „Cloud icon“ um.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "image.update",
    "params": {
        "imageid": "2",
        "name": "Cloud icon"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "2"
        ]
    },
    "id": 1
}
```

Quelle

CImage::update() in *ui/include/classes/api/services/CImage.php*.

Complex event processing

This class is designed to work with complex event processing.

Object references:

- CEP rule
- CEP rule filter
  - CEP rule filter condition
- CEP rule time window
- CEP rule operation
- CEP rule operation filter
- CEP rule operation filter condition

Available methods:

- ceprule.create - create new CEP rules
- ceprule.delete - delete CEP rules
- ceprule.get - retrieve CEP rules
- ceprule.update - update CEP rules
- ceprule.resettimewindows - reset time windows for CEP rule

CEP rule object

The following objects are directly related to the ceprule API.

CEP rule

The CEP rule object has the following properties.

Property	Type	Description
cep_ruleid	ID	ID of the CEP rule. Property behavior: - *read-only* - *required* for update operations
name	string	Name of the CEP rule. Property behavior: - *required* for create operations
window_type	integer	Time window type: 0 - (default) None 1 - Simple (multiple); 2 - Cause and symptom grouping; 3 - Simple (once); 4 - Event pattern match.
stop	integer	Possible values: 0 - (default) Continue with the next CEP rule according to sort order; 1 - Stop after executing this rule.
sortorder	integer	Execution order of the CEP rule.
description	string	Description of the CEP rule.
status	integer	Whether the CEP rule is enabled or disabled. Possible values: 0 - *(default)* enabled; 1 - disabled.
error	string	Error text if the CEP rule processing failed on rule level. Property behavior: - *read-only*

CEP rule filter

The CEP rule filter object defines a set of conditions that must be met for events to be processed by CEP rule. It has the following properties.

Property	Type	Description
conditions	array	Set of filter conditions to use for filtering results. The conditions will be sorted in the order of their placement in the formula. Property behavior: - *required*
evaltype	integer	Filter condition evaluation method. Possible values: 0 - And/Or; 1 - And; 2 - Or; 3 - Custom expression. Property behavior: - *required*

Property	Type	Description
eval_formula	string	Generated expression that will be used for evaluating filter conditions. The expression contains IDs that reference specific filter conditions by its `formulaid`. The value of `eval_formula` is equal to the value of `formula` for filters with a custom expression. Property behavior: - *read-only*
formula	string	User-defined expression to be used for evaluating conditions of filters with a custom expression. The expression must contain IDs that reference specific filter conditions by its `formulaid`. The IDs used in the expression must exactly match the ones defined in the filter conditions: no condition can remain unused or omitted. Property behavior: - *required* if `evaltype` is set to ”custom expression”

CEP rule filter condition

The CEP rule filter condition object defines a specific condition that must be checked before processing events by the CEP rule.

Property	Type	Description
type	integer	Type of condition. Possible values: 3 - event name; 4 - severity; 6 - time period; 25 - tag name; 26 - tag value; 36 - host visible name; 37 - host group name. Property behavior: - *required*
operator	integer	Condition operator. Possible values: 0 - equals (default); 1 - does not equal; 2 - contains; 3 - does not contain; 4 - in; 5 - is greater than or equals; 6 - is less than or equals; 7 - not in. Property behavior: - *required*.
event_name	string	Event name for a specific operation type. Property behavior: - *required* if `type` is set to ”event name”
severity	string	Event severity for a specific operation type. Property behavior: - *required* if `type` is set to ”severity”
time_period	string	Time period for a specific operation type. Property behavior: - *required* if `type` is set to ”time period”

Property	Type	Description
tag	string	Tag name for a specific operation type. Property behavior: - *required* if `type` is set to ”tag name” or ”tag value”
tag_value	string	Tag value for a specific operation type. Property behavior: - *required* if `type` is set to ”tag value”
event_severity	string	Event severity for a specific operation type. Property behavior: - *required* if `type` is set to ”severity”
host_name	string	Host name for a specific operation type. Property behavior: - *required* if `type` is set to ”host visible name”
host_group_name	string	Host group name for a specific operation type. Property behavior: - *required* if `type` is set to ”host group name”
formulaid	string	Arbitrary unique ID that is used to reference the condition from a custom expression. Can only contain capital-case letters. The ID must be defined by the user when modifying filter conditions, but will be generated anew when requesting them afterward. Property behavior: - *required* if CEP rule filter `evaltype` is set to ”custom expression”

CEP rule time window

The CEP rule time window object defines the time window of the CEP rule. It has the following properties.

Property	Type	Description
duration	string	Duration of the CEP rule window (365 days maximum). Accepts seconds and time units with suffix. Default: `10m`
capacity	integer	Capacity of the CEP rule window. Supports user macros. Default: 0 (unlimited)
group_by_host_group	integer	Group by flag for host group. Possible values: 0 - no (default); 1 - yes.
group_by_host	integer	Group by flag for host. Possible values: 0 - no (default); 1 - yes.
group_by_tags	integer	Group by flag for tag. Possible values: 0 - no (default); 1 - yes.
tags	string	Tag names to use for grouping. Property behavior: - *required* if `group_by_tag` is set to ”1”

Property	Type	Description
event_count_tag	string	Tag name for displaying the count of affected events. Property behavior: - *supported* if CEP rule `window_type` is set to ”2”; not allowed otherwise
script	string	Script to execute. Property behavior: - *supported* if CEP rule `window_type` is set to ”4”; not allowed otherwise

CEP rule operation

The CEP rule operation object defines an operation that will be performed when a CEP rule is executed. It has the following properties.

Property	Type	Description
sortorder	integer	Execution order (ascending) of the operation. Property behavior: - *required*
execute_when	integer	Type of condition for operation. Possible values: 0 - event occurred; 1 - event added; 2 - event evicted; 3 - window closed; 4 - event pattern matched; 5 - event closed. Property behavior: - *required*
scope	integer	Scope of the operation. Possible values: 0 - current event (default); 1 - other events; 2 - all events. Possible values when `execute_when` is ”Window closed” or ”Pattern matched”: 2 - all events. Possible values when `execute_when` is ”Event ocurred”: 0 - current event.
filter	object	Filter object.
scope_filter	object	Scope filter object. scope_filter is editable only if the `scope` field contains ”Other events” or ”All events” value. Otherwise scope_filter accepts only object with default values.

Property	Type	Description
type	integer	CEP operation type. Possible values: 1 - set event name; 2 - close event; 3 - discard event; 4 - set event severity; 5 - increase event severity; 6 - decrease event severity; 7 - suppress event; 8 - unsuppress event; 9 - copy first event as new; 10 - copy last event as new; 11 - add tag; 12 - set tag; 13 - set tag value; 14 - increase tag value; 15 - decrease tag value; 16 - rename tag; 17 - remove tag; 18 - close window. Property behavior: - *required* - *supported*: 1, 2, 3, 4, 5, 6, 7, 8, 11, 12, 13, 14, 15, 16, 17 - supported if ”Event occurred” (see `execute_when`) 1, 2, 4, 5, 6, 7, 8, 11, 12, 13, 14, 15, 16, 17, 18 - supported if ”Event added” (see `execute_when`) 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 - supported if ”Event evicted” (see `execute_when`) 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 - supported if ”Event closed” (see `execute_when`) 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17 - supported if ”Window closed” (see `execute_when`) 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 - supported if ”Event pattern matched” (see `execute_when`)
event_name	string	Event name for a specific operation type. Property behavior: - *required* if `type` is set to ”Set event name”
tag	string	Tag name for specific operation types. Property behavior: - *required* if `type` is set to ”Add tag”, ”Set tag”, ”Set tag value”, ”Increase tag value”, ”Decrease tag value”, ”Rename tag”, ”Remove tag”
new_tag	string	New tag name for a specific operation type. Property behavior: - *required* if `type` is set to ”Rename tag”
tag_value	string	Tag value for specific operation types. Property behavior: - *required* if `type` is set to ”Add tag”, ”Set tag”, ”Set tag value”, ”Increase tag value”, ”Decrease tag value”, ”Rename tag”, ”Remove tag”
severity	integer	Event severity for a specific operation type. Property behavior: - *required* if `type` is set to ”Set event severity”

Property	Type	Description
suppress_duration	string	Duration of the event suppression. Accepts seconds and time unit with suffix. Property behavior: - *required* if `type` is set to ”Suppress event”

CEP rule operation filter

The CEP rule operation filter object defines a set of conditions that must be met to perform the configured CEP operations. It has the following properties.

Property	Type	Description
conditions	array	Set of operation filter conditions to use for filtering results. The conditions will be sorted in the order of their placement in the formula. Property behavior: - *required*
evaltype	integer	Filter condition evaluation method. Possible values: 0 - And/Or; 1 - And; 2 - Or; 3 - Custom expression. Property behavior: - *required*
eval_formula	string	Generated expression that will be used for evaluating filter conditions. The expression contains IDs that reference specific filter conditions by its `formulaid`. The value of `eval_formula` is equal to the value of `formula` for filters with a custom expression. Property behavior: - *read-only*
formula	string	Arbitrary unique ID that is used to reference the condition from a custom expression. Can only contain capital-case letters. The ID must be defined by the user when modifying filter conditions, but will be generated anew when requesting them afterward. Property behavior: - *required*

CEP rule operation filter condition

The CEP rule operation filter condition object defines a specific condition that must be checked before running the CEP rule operations.

Property	Type	Description
type	integer	Type of operation condition. Possible values: 25 - tag name; 26 - tag value; 30 - problem opened; 31 - first; 32 - last; 33 - symptom; 34 - cloned; 35 - suppressed. Property behavior: - *required* - *supported*: 25, 26, 30, 33, 34, 35 - supported if ”Event occurred” (see `execute_when`)
operator	integer	Condition operator. Possible values: 0 - equals (default); 1 - does not equal; 2 - contains; 3 - does not contain; 5 - is greater than or equals; 6 - is less than or equals; 10 - true; 11 - false. Property behavior: - *required* - *supported*: 0, 1, 2, 3 - supported values if `type` set to ”Tag name” 0, 1, 2, 3, 5, 6 - supported values if `type` set to ”Tag value” 10, 11 - supported values if `type` set to ”Problem opened”, ”First”, ”Last”, ”Symptom”, ”Cloned”, ”Suppressed”.
tag	string	Tag name of operation condition. Property behavior: - *required* if `type` is set to ”Tag name” or ”Tag value”
tag_value	string	Value to compare with. Property behavior: - *required* if `type` is set to ”Tag value”

CEP rule operation scope filter

The CEP rule operation scope filter object defines a set of conditions that must be met to apply the configured CEP operations to events. It has the following properties.

Property	Type	Description
conditions	array	Set of operation scope filter conditions to use for filtering results. The conditions will be sorted in the order of their placement in the formula. Property behavior: - *required*

Property	Type	Description
evaltype	integer	Filter condition evaluation method. Possible values: 0 - And/Or; 1 - And; 2 - Or; 3 - Custom expression. Property behavior: - *required*
eval_formula	string	Generated expression that will be used for evaluating filter conditions. The expression contains IDs that reference specific filter conditions by its `formulaid`. The value of `eval_formula` is equal to the value of `formula` for filters with a custom expression. Property behavior: - *read-only*
formula	string	Arbitrary unique ID that is used to reference the condition from a custom expression. Can only contain capital-case letters. The ID must be defined by the user when modifying filter conditions, but will be generated anew when requesting them afterward. Property behavior: - *required*

CEP rule operation scope filter condition

The CEP rule operation scope filter condition object defines a specific condition that must be checked before applying the CEP rule operations to events.

Property	Type	Description
type	integer	Type of operation condition. Possible values: 25 - tag name; 26 - tag value; 30 - problem opened; 31 - first; 32 - last; 33 - symptom; 34 - cloned; 35 - suppressed. Property behavior: - *required* - *supported*: 25, 26, 30, 33, 34, 35 - supported if ”Event occurred” (see `execute_when`)

Property	Type	Description
operator	integer	Condition operator. Possible values: 0 - equals (default); 1 - does not equal; 2 - contains; 3 - does not contain; 5 - is greater than or equals; 6 - is less than or equals; 10 - true; 11 - false. Property behavior: - *required* - *supported*: 0, 1, 2, 3 - supported values if `type` set to ”Tag name” 0, 1, 2, 3, 5, 6 - supported values if `type` set to ”Tag value” 10, 11 - supported values if `type` set to ”Problem opened”, ”First”, ”Last”, ”Symptom”, ”Cloned”, ”Suppressed”.
tag	string	Tag name of operation condition. Property behavior: - *required* if `type` is set to ”Tag name” or ”Tag value”
tag_value	string	Value to compare with. Property behavior: - *required* if `type` is set to ”Tag value”

ceprule.create

Description

`object ceprule.create(object/array CEP rules)`

This method allows you to create new CEP rules.

Note:
This method is only available to *Super admin* user type. Permissions to call the method can be revoked in user role settings. See User roles for more information.

Parameters

`(object/array)` CEP rules to create.

Additionally to the standard CEP properties, the method accepts the following parameters.

Parameter	Type	Description
operations	array	CEP rule operations to create for the CEP rule. Parameter behavior: - *required*
filter	object	CEP rule filter object for the CEP rule. Parameter behavior: - *required*

Return values

`(object)` Returns an object containing the IDs of the created CEP rules under the `cep_ruleids` property. The order of the returned IDs matches the order of the passed CEP rules.

Examples

Create a new CEP rule

Create a CEP rule using evaluation method `AND/OR` with one condition and one window. By default, the rule will be enabled.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "ceprule.create",
    "params": {
      "name": "Group events by cause",
      "stop": "1",
      "sortorder": "0",
      "description": "",
      "status": "0",
      "window_type": "2",
      "filter": {
        "evaltype": "0",
        "formula": "",
        "conditions": [
          {
            "type": "3",
            "operator": "2",
            "event_name": "Printer",
            "tag": "",
            "tag_value": "",
            "severity": "0",
            "host_name": "",
            "host_group_name": "",
            "time_period": "",
            "formulaid": ""
          }
        ]
      },
      "window": {
        "duration": "1d",
        "capacity": "0",
        "group_by_host_group": "0",
        "group_by_host": "1",
        "group_by_tags": "1",
        "tags": [
          "cause"
        ],
        "event_count_tag": "",
        "script": ""
      },
      "operations": []
    },
    "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": {
    "cep_ruleids": [
      "1"
    ]
  },
  "id": 1
}
```

See also

- CEP rule

Source

CCepRule::create() in *ui/include/classes/api/services/CCepRule.php*.

ceprule.delete

Description

`object ceprule.delete(array cep_ruleids)`

This method allows you to delete CEP rules.

> **Note:**
> This method is only available to *Super admin* user type. Permissions to call the method can be revoked in user role settings. See User roles for more information.

Parameters

`(array)` IDs of the CEP rules to delete.

Return values

`(object)` Returns an object containing the IDs of the deleted CEP rules under the `cep_ruleids` property.

Example

Delete multiple CEP rules

Delete two CEP rules.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "ceprule.delete",
    "params": [
        "3",
        "4"
    ],
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "cep_ruleids": [
            "3",
            "4"
        ]
    },
    "id": 1
}
```

Source

CCepRule::delete() in *ui/include/classes/api/services/CCepRule.php*.

ceprule.get

Description

`integer/array ceprule.get(object parameters)`

The method allows you to retrieve CEP rules according to the given parameters.

Note:
This method is available to users of any type. Permissions to call the method can be revoked in user role settings. See User roles for more information.

Parameters

`(object)` Parameters defining the desired output.

The method supports the following parameters.

Parameter	Type	Description
cep_ruleids	ID/array	Return only CEP rules with the given IDs.
selectFilter	query	Return a `filter` property with the CEP rule conditions.
selectOperations	query	Return an `operations` property with the CEP rule operations.
selectWindow	query	Return a `[window]`(/manual/api/reference/ceprule/object#cep-rule-time-window) property with the CEP rule time window.
sortfield	string/array	Sort the result by the given properties. Possible values: `cep_ruleid`, `name`, `stop`, `sortorder`, `status`.
countOutput	boolean	These parameters are described in the reference commentary.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Return values

`(integer/array)` Returns either:

- an array of objects;
- the count of retrieved objects, if the `countOutput` parameter has been used.

Examples

Retrieve CEP rules

Retrieve enabled CEP rules, including conditions, windows, and operations.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "ceprule.get",
    "params": {
        "output": "extend",
        "filter": {"status": 0},
        "selectFilter": "extend",
        "selectWindow": "extend",
        "selectOperations": "extend"
    },
    "id": 1
}
```

Response:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "cep_ruleid": "2",
```

```
      "name": "Group events by cause",
      "stop": "0",
      "sortorder": "1",
      "description": "",
      "status": "0",
      "window_type": "2",
      "error": "",
      "filter": {
        "evaltype": "0",
        "eval_formula": "A",
        "formula": "",
        "conditions": [
          {
            "type": "3",
            "operator": "2",
            "event_name": "Printer",
            "tag": "",
            "tag_value": "",
            "severity": "0",
            "host_name": "",
            "host_group_name": "",
            "time_period": "",
            "formulaid": "A"
          }
        ]
      },
      "window": {
        "duration": "1d",
        "capacity": "0",
        "group_by_host_group": "0",
        "group_by_host": "1",
        "group_by_tags": "1",
        "tags": [
          "cause"
        ],
        "event_count_tag": "symptoms",
        "script": ""
      },
      "operations": []
    },
    {
      "cep_ruleid": "10",
      "name": "Suppress consecutive events",
      "stop": "1",
      "sortorder": "2",
      "description": "",
      "status": "0",
      "window_type": "3",
      "error": "",
      "filter": {
        "evaltype": "0",
        "eval_formula": "A",
        "formula": "",
        "conditions": [
          {
            "type": "3",
            "operator": "2",
            "event_name": "Printer",
            "tag": "",
            "tag_value": "",
            "severity": "0",
            "host_name": "",
```

```
          "host_group_name": "",
          "time_period": "",
          "formulaid": "A"
        }
      ]
    },
    "window": {
      "duration": "3m",
      "capacity": "0",
      "group_by_host_group": "0",
      "group_by_host": "1",
      "group_by_tags": "1",
      "tags": [
        "cause"
      ],
      "event_count_tag": "",
      "script": ""
    },
    "operations": [
      {
        "sortorder": "1",
        "execute_when": "1",
        "type": "7",
        "event_name": "",
        "severity": "0",
        "suppress_duration": "6m",
        "tag": "",
        "new_tag": "",
        "tag_value": "",
        "scope": "0",
        "filter": {
          "evaltype": "0",
          "eval_formula": "A",
          "formula": "",
          "conditions": [
            {
              "type": "33",
              "operator": "10",
              "tag": "",
              "tag_value": "",
              "formulaid": "A"
            }
          ]
        },
        "scope_filter": {
          "evaltype": "0",
          "eval_formula": "",
          "formula": "",
          "conditions": []
        }
      },
      {
        "sortorder": "2",
        "execute_when": "2",
        "type": "8",
        "event_name": "",
        "severity": "0",
        "suppress_duration": "0",
        "tag": "",
        "new_tag": "",
        "tag_value": "",
        "scope": "0",
```

```
          "filter": {
            "evaltype": "0",
            "eval_formula": "A",
            "formula": "",
            "conditions": [
              {
                "type": "35",
                "operator": "10",
                "tag": "",
                "tag_value": "",
                "formulaid": "A"
              }
            ]
          },
          "scope_filter": {
            "evaltype": "0",
            "eval_formula": "",
            "formula": "",
            "conditions": []
          }
        }
      ]
    }
  ],
  "id": 1
}
```

See also

- CEP rule

Source

CCepRule::get() in *ui/include/classes/api/services/CCepRule.php*.

ceprule.resettimewindows

Description

`object ceprule.resettimewindows(object/array cep_ruleid)`

This method allows you to reset time windows for an existing CEP rule.

Note:
This method is only available to *Super admin* user type. Permissions to call the method can be revoked in user role settings. See User roles for more information.

Parameters

`(object)` CEP rule object must have a `cep_ruleid` property defined.

Return values

`(object)` Returns an object containing the ID of the CEP rule whose time windows have been reset under the `cep_ruleid` property.

Examples

Reset time windows for CEP rule

Request:

```
{
    "jsonrpc": "2.0",
    "method": "ceprule.resettimewindows",
    "params": {
        "cep_ruleid": "1"
    },
```

```
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
      "cep_ruleid": "1"
    },
    "id": 1
}
```

See also

- CEP rule

Source

CCepRule::resettimewindows() in *ui/include/classes/api/services/CCepRule.php*.

ceprule.update

Description

`object ceprule.update(object/array CEP rules)`

This method allows you to update existing CEP rules.

Note:
This method is only available to *Super admin* user type. Permissions to call the method can be revoked in user role settings. See User roles for more information.

Parameters

`(object/array)` CEP rule properties to be updated.

The `cep_ruleid` property must be defined for each CEP rule, all other properties are optional. Only the passed properties will be updated, all others will remain unchanged.

Additionally to the standard CEP rule properties, the method accepts the following parameters.

Parameter	Type	Description
filter	object	CEP rule filter object to replace the current filter.
window	array	CEP rule window to be updated. Property behavior: - *required* if `window_type` is changed to ”Cause and symptom grouping” and ”group_by_host_group”, ”group_by_host”, ”group_by_tags” are all set to NO; - *required* if `window_type` is changed to ”Event pattern match”; - *supported* if `window_type` is ”Simple (multiple)” or ”Simple (once)”
operations	array	CEP rule operations to replace existing operations. Property behavior: - *required* if `window_type` is changed to ”None” (0), ”Simple (multiple)” (1), ”Simple (once)” (3) or ”Event pattern match” (4).

Return values

`(object)` Returns an object containing the IDs of the updated CEP rules under the `cep_ruleids` property.

Examples

Disable CEP rule

Request:

```
{
    "jsonrpc": "2.0",
    "method": "ceprule.update",
    "params": {
        "cep_ruleid": "1",
        "status": "1"
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "cep_ruleids": [
            "1"
        ]
    },
    "id": 1
}
```

Replace condition

Request:

```
{
    "jsonrpc": "2.0",
    "method": "ceprule.update",
    "params": {
        "cep_ruleid": "3",
        "filter": {
        "evaltype": "0",
        "formula": "",
            "conditions": [
                {
                    "type": "3",
                    "operator": "2",
                    "event_name": "Inaccessible port",
                    "severity": "0"
                }
            ]
        }
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "cep_ruleids": [
            "1"
        ]
    },
    "id": 1
}
```

See also

- CEP rule

Source

CCepRule::update() in *ui/include/classes/api/services/CCepRule.php*.

Connector

Diese Klasse ist für die Arbeit mit Connectors konzipiert.

Objektreferenzen:

- Connector
- Tag-Filter

Verfügbare Methoden:

- connector.create - neue Connectors erstellen
- connector.delete - Connectors löschen
- connector.get - Connectors abrufen
- connector.update - Connectors aktualisieren

Konnektor Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der`connector` API.

Connector

Das Connector-Objekt verfügt über die folgenden Eigenschaften.

Property	Type	Description
connectorid	ID	ID des Connectors. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
name	string	Name des Connectors. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
url	string	Endpunkt-URL, also die URL des Empfängers. Benutzermakros werden unterstützt. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
protocol	integer	Kommunikationsprotokoll. Mögliche Werte: 0 - *(Standard)* Zabbix Streaming Protocol v1.0.
data_type	integer	Datentyp. Mögliche Werte: 0 - *(Standard)* Datenpunktwerte; 1 - Ereignisse.
item_value_type	integer	Summe der zu sendenden Datenpunktwerttypen. Mögliche Werte: 1 - Numerisch (Gleitkomma); 2 - Zeichen; 4 - Log; 8 - Numerisch (vorzeichenlos); 16 - Text; 32 - Binär; 64 - JSON. Standard: 31 - Alle Datenpunkttypen (außer binär und JSON). Verhalten der Eigenschaft: - *unterstützt*, wenn `data_type` auf „Datenpunktwerte“ gesetzt ist.

Property	Type	Description
max_records	integer	Maximale Anzahl von Ereignissen oder Datenpunkten, die innerhalb einer Nachricht gesendet werden können. Mögliche Werte: 0-2147483647 (Maximalwert einer vorzeichenbehafteten 32-Bit-Ganzzahl). Standard: 0 - Unbegrenzt.
max_senders	integer	Anzahl der Senderprozesse, die für diesen Connector ausgeführt werden sollen. Mögliche Werte: 1-100. Standard: 1.
max_attempts	integer	Anzahl der Versuche. Mögliche Werte: 1-5. Standard: 1.
attempt_interval	string	Intervall zwischen Wiederholungsversuchen. Akzeptiert Sekunden. Mögliche Werte: 0s-10s. Standard: 5s. Verhalten der Eigenschaft: - *unterstützt*, wenn `max_attempts` größer als 1 ist.
timeout	string	Zeitüberschreitung. Zeitsuffixe werden unterstützt (z. B. `30s`, `1m`). Benutzermakros werden unterstützt. Mögliche Werte: 1s-60s. Standard: 5s.
http_proxy	string	HTTP(S)-Proxy-Verbindungszeichenfolge im Format *[protocol]://[username[:password]@]proxy.example.com[:port]*. Benutzermakros werden unterstützt.
authtype	integer	HTTP-Authentifizierungsmethode. Mögliche Werte: 0 - *(Standard)* Keine; 1 - Basic; 2 - NTLM; 3 - Kerberos; 4 - Digest; 5 - Bearer.
username	string	Benutzername. Benutzermakros werden unterstützt. Verhalten der Eigenschaft: - *unterstützt*, wenn `authtype` auf „Basic“, „NTLM“, „Kerberos“ oder „Digest“ gesetzt ist
password	string	Passwort. Benutzermakros werden unterstützt. Verhalten der Eigenschaft: - *unterstützt*, wenn `authtype` auf „Basic“, „NTLM“, „Kerberos“ oder „Digest“ gesetzt ist

Property	Type	Description
token	string	Bearer-Token. Benutzermakros werden unterstützt. Verhalten der Eigenschaft: - *erforderlich*, wenn `authtype` auf „Bearer“ gesetzt ist
verify_peer	integer	Gibt an, ob überprüft werden soll, dass das Zertifikat des Hosts authentisch ist. Mögliche Werte: 0 - Nicht überprüfen; 1 - *(Standard)* Überprüfen.
verify_host	integer	Gibt an, ob überprüft werden soll, dass der Hostname der Verbindung mit dem im Zertifikat des Hosts übereinstimmt. Mögliche Werte: 0 - Nicht überprüfen; 1 - *(Standard)* Überprüfen.
ssl_cert_file	string	Dateipfad zum öffentlichen SSL-Schlüssel. Benutzermakros werden unterstützt.
ssl_key_file	string	Dateipfad zum privaten SSL-Schlüssel. Benutzermakros werden unterstützt.
ssl_key_password	string	Passwort für die SSL-Schlüsseldatei. Benutzermakros werden unterstützt.
description	text	Beschreibung des Connectors.
status	integer	Gibt an, ob der Connector aktiviert ist. Mögliche Werte: 0 - Deaktiviert; 1 - *(Standard)* Aktiviert.
tags_evaltype	integer	Auswertungsmethode der Tags. Mögliche Werte: 0 - *(Standard)* Und/Oder; 2 - Oder.

Tag-Filter

Der Tag-Filter ermöglicht es Ihnen, nur übereinstimmende Datenpunktwerte oder Ereignisse zu exportieren. Wenn er nicht gesetzt ist, wird alles exportiert. Das Tag-Filter-Objekt hat die folgenden Eigenschaften.

Property	Type	Description
tag	string	Tag-Name. Property behavior: - *required*
operator	integer	Bedingungs-operator. Mögliche Werte: 0 - *(default)* Gleich; 1 - Ungleich; 2 - Enthält; 3 - Enthält nicht; 12 - Existiert; 13 - Existiert nicht.
value	string	Tag-Wert.

connector.create

Beschreibung

`object connector.create(object/array connectors)`

Diese Methode ermöglicht es Ihnen, neue Connector-Objekte zu erstellen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellende Connector-Objekte.

Zusätzlich zu den Standard-Connector-Eigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
tags	array	Connector-Tag-Filter.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Verbindungen unter der Eigenschaft `connectorids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Konnektoren.

Beispiele

Erstellen eines Verbinders

Erstellen Sie einen Connector, um Auslöser (trigger)-Ereignisse mit einem Tag-Filter zu exportieren. Die HTTP-Authentifizierung wird mit einem Bearer-Token durchgeführt.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorid": [
            "3"
        ]
    },
    "id": 1
}
```

Quelle

CConnector::create() in *ui/include/classes/api/services/CConnector.php*.

connector.delete

Beschreibung

`object connector.delete(array connectorids)`

Diese Methode ermöglicht es Ihnen, Connector-Einträge zu löschen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der zu löschenden Konnektoren.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Verbindungen unter der Eigenschaft `connectorids` enthält.

Beispiele

Löschen mehrerer Konnektoren

Löschen Sie zwei Konnektoreinträge.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "connector.delete",
    "params": [
        3,
        5
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorids": [
            "3",
            "5"
        ]
    },
    "id": 1
}
```

Quelle

CConnector::delete() in *ui/include/classes/api/services/CConnector.php*.

connector.get

Beschreibung

`integer/array connector.get(object parameters)`

Die Methode ermöglicht es Ihnen, Connector-Objekte anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
connectorids	ID/array	Gibt nur Connectoren mit den angegebenen IDs zurück.
selectTags	query	Gibt eine Eigenschaft `tags` mit dem Tag-Filter des Connectors zurück. Unterstützt `count`.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `connectorid`, `name`, `data_type`, `status`.
countOutput	boolean	Diese Parameter sind in der Referenzkommentierung beschrieben.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Liefert entweder:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Abrufen aller Konnektoren

Abrufen aller Daten über alle Konnektoren und deren Eigenschaften.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "connector.get",
    "params": {
        "output": "extend",
        "selectTags": ["tag", "operator", "value"],
        "preservekeys": true
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```


Quelle

CConnector:get() in *ui/include/classes/api/services/CConnector.php*.

connector.update

Beschreibung

`object connector.update(object/array connectors)`

Diese Methode ermöglicht es Ihnen, vorhandene Connectoren zu aktualisieren.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu aktualisierende Connector-Eigenschaften.

Die Eigenschaft `connectorid` muss für jeden Connector definiert werden, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Zusätzlich zu den Standard-Connector-Eigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
tags	array	Connector-Tag-Filter, der den aktuellen Tag-Filter ersetzt.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Anschlüsse unter der Eigenschaft `connectorids` enthält.

Beispiele

Ändern des HTTP-Authentifizierungstyps

Ändern Sie den HTTP-Authentifizierungstyp in Bearer für den Connector mit der ID ”3”.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "connector.update",
    "params": {
        "connectorid": 3,
        "authtype": 5,
        "token": "{$DATA_EXPORT_BEARER_TOKEN}"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorids": [
            "3"
        ]
    },
    "id": 1
}
```

Tag-Filter aktualisieren

Tag-Filter für Konnektor mit ID ”5” ändern.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorids": [
            "5"
        ]
    },
    "id": 1
}
```

Quelle

CConnector::update() in *ui/include/classes/api/services/CConnector.php*.

Dashboard

Diese Klasse ist für die Arbeit mit Dashboards konzipiert.

Objektreferenzen:

- Dashboard
- Dashboard-Seite
  - Dashboard-Widget
    * Dashboard-Widget-Feld
- Dashboard-Benutzergruppe
- Dashboard-Benutzer

Verfügbare Methoden:

- dashboard.create - neue Dashboards erstellen
- dashboard.delete - Dashboards löschen
- dashboard.get - Dashboards abrufen
- dashboard.update - Dashboards aktualisieren

Dashboard-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `dashboard` API.

Dashboard

Das Dashboard-Objekt hat die folgenden Eigenschaften.

Property	Type	Description
dashboardid	ID	ID des Dashboards. Property behavior: - *read-only* - *required* for update operations
name	string	Name des Dashboards. Property behavior: - *required* for create operations
userid	ID	ID des Benutzers, der der Eigentümer des Dashboards ist.
private	integer	Typ der Dashboard-Freigabe. Mögliche Werte: 0 - öffentliches Dashboard; 1 - *(default)* privates Dashboard.
display_period	integer	Standardintervall der Diashow (in Sekunden). Mögliche Werte: 10, 30, 60, 120, 600, 1800, 3600. Standard: 30.
auto_start	integer	Diashow automatisch starten. Mögliche Werte: 0 - Diashow nicht automatisch starten; 1 - *(default)* Diashow automatisch starten.

Dashboard-Seite

Das Objekt der Dashboard-Seite hat die folgenden Eigenschaften.

Property	Type	Description
dashboard_pageid	ID	ID der Dashboard-Seite. Property behavior: - *read-only*
name	string	Name der Dashboard-Seite. Standard: leerer String.
display_period	integer	Intervall der Dashboard-Diashow (in Sekunden). Mögliche Werte: 0, 10, 30, 60, 120, 600, 1800, 3600. Standard: 0 (verwendet das Standard-Diashow-Intervall).
widgets	array	Array der Objekte vom Typ dashboard widget.

Dashboard-Widget

Das Dashboard-Widget-Objekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
widgetid	ID	ID des Dashboard-Widgets. Eigenschaftsverhalten: - *schreibgeschützt*

Eigenschaft	Typ	Beschreibung
type	string	Typ des Dashboard-Widgets. Mögliche Werte: actionlog - Aktionsprotokoll; clock - Uhr; discovery - Discovery-Status; favgraphs - Bevorzugte Graphen; favmaps - Bevorzugte Karten; gauge - Messanzeige; geomap - Geomap; graph - Graph (klassisch); graphprototype - Graphprototyp; honeycomb - Honeycomb; hostavail - Host-Verfügbarkeit; hostcard - Host-Karte; hostnavigator - Host-Navigator; itemcard - Datenpunkt-Karte; itemhistory - Datenpunkt-Verlauf; itemnavigator - Datenpunkt-Navigator; item - Datenpunkt-Wert; map - Karte; navtree - Karten-Navigationsbaum; piechart - Kreisdiagramm; problemhosts - Problem-Hosts; problems - Probleme; problemsbysv - Probleme nach Schweregrad; scatterplot - Streudiagramm; slareport - SLA-Bericht; svggraph - Graph; systeminfo - Systeminformationen; tophosts - Top-Hosts; topitems - Top-Datenpunkte; toptriggers - Top-Auslöser; trigover - Auslöserübersicht; url - URL; web - Web-Überwachung. Eigenschaftsverhalten: - *erforderlich*
name	string	Benutzerdefinierter Widget-Name.
x	integer	Eine horizontale Position von der linken Seite des Dashboards aus. Mögliche Werte reichen von 0 bis 71.
y	integer	Eine vertikale Position vom oberen Rand des Dashboards aus. Mögliche Werte reichen von 0 bis 63.
width	integer	Die Widget-Breite. Mögliche Werte reichen von 1 bis 72.
height	integer	Die Widget-Höhe. Mögliche Werte reichen von 1 bis 64.
view_mode	integer	Der Widget-Anzeigemodus. Mögliche Werte: 0 - *(Standard)* Standard-Widget-Ansicht; 1 - mit ausgeblendetem Header;
fields	array	Array von Objekten des Typs Dashboard-Widget-Feld. Eigenschaftsverhalten: - siehe einzelne Widgets unter Dashboard-Widget-Felder

Dashboard-Widget-Feld

Das Dashboard-Widget-Feldobjekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
type	integer	Typ des Widget-Feldes. Mögliche Werte: 0 - Ganzzahl; 1 - Zeichenfolge; 2 - Host-Gruppe; 3 - Host; 4 - Datenpunkt; 5 - Datenpunkt-Prototyp; 6 - Graph; 7 - Graph-Prototyp; 8 - Karte; 9 - Service; 10 - SLA; 11 - Benutzer; 12 - Aktion; 13 - Medientyp. Verhalten der Eigenschaft: - *erforderlich*
name	string	Name des Widget-Feldes. Mögliche Werte: siehe Dashboard-Widget-Felder. Verhalten der Eigenschaft: - *erforderlich*
value	mixed	Wert des Widget-Feldes abhängig vom Typ. Mögliche Werte: siehe Dashboard-Widget-Felder. Verhalten der Eigenschaft: - *erforderlich*

Dashboard-Benutzergruppe

Liste der Dashboard-Berechtigungen basierend auf Benutzergruppen. Sie hat die folgenden Eigenschaften.

Eigenschaft	Type	Beschreibung
usrgrpid	ID	ID der Benutzergruppe. Property behavior: - *required*
permission	integer	Typ der Berechtigungsstufe. Mögliche Werte: 2 - schreibgeschützt; 3 - Lesen und Schreiben. Property behavior: - *required*

Dashboard-Benutzer

Liste der Dashboard-Berechtigungen basierend auf Benutzern. Sie hat die folgenden Eigenschaften.

Eigenschaft	Type	Beschreibung
userid	ID	ID des Benutzers. Verhalten der Eigenschaft: - *erforderlich*
permission	integer	Typ der Berechtigungsstufe. Mögliche Werte: 2 - schreibgeschützt; 3 - Lesen und Schreiben. Verhalten der Eigenschaft: - *erforderlich*

dashboard.create

Beschreibung

`object dashboard.create(object/array dashboards)`

Diese Methode ermöglicht es Ihnen, neue Dashboards zu erstellen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellende Dashboards.

Zusätzlich zu den Standard-Dashboard-Eigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
pages	array	Für das Dashboard zu erstellende Dashboard-Seiten. Dashboard-Seiten werden in derselben Reihenfolge angeordnet, wie sie angegeben sind. Parameterverhalten: - *erforderlich*
users	array	Auf dem Dashboard zu erstellende Freigaben für Dashboard-Benutzer.
userGroups	array	Auf dem Dashboard zu erstellende Freigaben für Dashboard-Benutzergruppen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Dashboards unter der Eigenschaft `dashboardids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Dashboards.

Beispiele

Erstellen eines Dashboards

Erstellen Sie ein Dashboard mit dem Namen „My dashboard“ mit einem Problems-Widget mit Tags und unter Verwendung von zwei Freigabetypen (Benutzergruppe und Benutzer) auf einer einzelnen Dashboard-Seite.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

Siehe auch

- Dashboard-Seite
- Dashboard-Widget
- Dashboard-Widget-Feld
- Dashboard-Benutzer

- Dashboard-Benutzergruppe

Quelle

CDashboard::create() in *ui/include/classes/api/services/CDashboard.php*.

dashboard.delete

Beschreibung

`object dashboard.delete(array dashboardids)`

Diese Methode ermöglicht es Ihnen, Dashboards zu löschen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der zu löschenden Dashboards.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Dashboards unter der Eigenschaft `dashboardids` enthält.

Beispiele

Mehrere Dashboards löschen

Löschen Sie zwei Dashboards.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.delete",
    "params": [
        "2",
        "3"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2",
            "3"
        ]
    },
    "id": 1
}
```

Quelle

CDashboard::delete() in *ui/include/classes/api/services/CDashboard.php*.

dashboard.get

Beschreibung

`integer/array dashboard.get(object parameters)`

Die Methode ermöglicht es Ihnen, Dashboards anhand der angegebenen Parameter abzurufen.

Note:
Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
dashboardids	ID/array	Gibt nur Dashboards mit den angegebenen IDs zurück.
selectPages	query	Gibt eine Eigenschaft `pages` mit Dashboard-Seiten in der korrekten Reihenfolge zurück.
selectUsers	query	Gibt eine Eigenschaft `users` mit Benutzern zurück, für die das Dashboard freigegeben ist.
selectUserGroups	query	Gibt eine Eigenschaft `userGroups` mit Benutzergruppen zurück, für die das Dashboard freigegeben ist.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `dashboardid`.
countOutput	boolean	Diese Parameter werden in den Referenzkommentaren beschrieben.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Kann die folgenden Dinge zurück geben:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Abrufen eines Dashboards anhand der ID

Rufen Sie alle Daten zu den Dashboards „1“ und „2“ ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.get",
    "params": {
        "output": "extend",
        "selectPages": "extend",
        "selectUsers": "extend",
        "selectUserGroups": "extend",
        "dashboardids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```


Siehe auch

- Dashboard-Seite
- Dashboard-Widget
- Dashboard-Widget-Feld
- Dashboard-Benutzer
- Dashboard-Benutzergruppe

Quelle

CDashboard::get() in *ui/include/classes/api/services/CDashboard.php*.

dashboard.update

Beschreibung

`object dashboard.update(object/array dashboards)`

Diese Methode ermöglicht es Ihnen, vorhandene Dashboards zu aktualisieren.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu aktualisierende Dashboard-Eigenschaften.

Die Eigenschaft `dashboardid` muss für jedes Dashboard definiert sein, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Zusätzlich zu den Standard-Dashboard-Eigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
pages	array	Dashboard-Seiten zum Ersetzen der vorhandenen Dashboard-Seiten. Dashboard-Seiten werden über die Eigenschaft `dashboard_pageid` aktualisiert. Neue Dashboard-Seiten werden für Objekte ohne die Eigenschaft `dashboard_pageid` erstellt, und vorhandene Dashboard-Seiten werden gelöscht, wenn sie nicht wiederverwendet werden. Dashboard-Seiten werden in derselben Reihenfolge angeordnet wie angegeben. Nur die angegebenen Eigenschaften der Dashboard-Seiten werden aktualisiert.
users	array	Dashboard-Benutzer Freigaben zum Ersetzen der vorhandenen Elemente.
userGroups	array	Dashboard-Benutzergruppen Freigaben zum Ersetzen der vorhandenen Elemente.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Dashboards unter der Eigenschaft `dashboardids` enthält.

Beispiele

Umbenennen eines Dashboards

Benennen Sie ein Dashboard in „SQL server status“ um.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.update",
    "params": {
        "dashboardid": "2",
        "name": "SQL server status"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

Dashboard-Seiten aktualisieren

Benennen Sie die erste Dashboard-Seite um, ersetzen Sie die Widgets auf der zweiten Dashboard-Seite und fügen Sie als dritte eine neue Seite hinzu. Löschen Sie alle anderen Dashboard-Seiten.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.update",
    "params": {
        "dashboardid": "2",
        "pages": [
            {
                "dashboard_pageid": 1,
                "name": "Umbenannte Seite"
            },
            {
                "dashboard_pageid": 2,
                "widgets": [
                    {
                        "type": "clock",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3
                    }
                ]
            },
            {
                "display_period": 60
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

Dashboard-Eigentümer ändern

Nur für Admins und Super-Admins verfügbar.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.update",
    "params": {
        "dashboardid": "2",
        "userid": "1"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
```

```
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

Siehe auch

- Dashboard-Seite
- Dashboard-Widget
- Dashboard-Widget-Feld
- Dashboard-Benutzer
- Dashboard-Benutzergruppe

Quelle

CDashboard::update() in *ui/include/classes/api/services/CDashboard.php*.

Dashboard-Widget-Felder

Diese Seite enthält Navigationslinks für Dashboard-Widget-Parameter und mögliche Eigenschaftswerte für die jeweiligen Objekte vom Typ dashboard widget field.

Um die Parameter und Eigenschaftswerte für jedes Widget anzuzeigen, verwenden Sie bitte die Seitenleiste, um zur entsprechenden Widget-Seite zu navigieren.

1 Aktionsprotokoll

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das Widget *Aktionsprotokoll* in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

> **Attention:**
> Die Eigenschaften von `fields` des Widgets werden während der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dies ermöglicht es Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets falsch zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Aktionsprotokoll* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Parameterverhalten.

Parameter

Die folgenden Parameter werden für das Widget *Aktionsprotokoll* unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	0	rf_rate	0 - Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - *(Standard)* 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - 15 Minuten.
Empfänger	11	userids.0	ID des Benutzers. Hinweis: Um mehrere Benutzer zu konfigurieren, erstellen Sie für jeden Benutzer ein Dashboard-Widget-Feldobjekt mit einer fortlaufenden Nummer im Eigenschaftsnamen.

Parameter	type	name	value
Aktionen	12	actionids.0	ID der Aktion. Hinweis: Um mehrere Aktionen zu konfigurieren, erstellen Sie für jede Aktion ein Dashboard-Widget-Feldobjekt mit einer fortlaufenden Nummer im Eigenschaftsnamen.
Medientypen	13	mediatypeids.0	ID des Medientyps. Hinweis: Um mehrere Medientypen zu konfigurieren, erstellen Sie für jeden Medientyp ein Dashboard-Widget-Feldobjekt mit einer fortlaufenden Nummer im Eigenschaftsnamen.
Status	0	statuses.0	0 - In Bearbeitung; 1 - Gesendet/Ausgeführt; 2 - Fehlgeschlagen. Hinweis: Um mehrere Werte zu konfigurieren, erstellen Sie für jeden Wert ein Dashboard-Widget-Feldobjekt mit einer fortlaufenden Nummer im Eigenschaftsnamen.
Suchzeichenfolge	1	message	Beliebiger Zeichenfolgenwert.
Zeitraum	1	time_period._referenc	`DASHBOARD._timeperiod` - den Zeitraumauswahlschalter des Dashboards als Datenquelle festlegen; `ABCDE._timeperiod` - ein kompatibles Widget (dessen Parameter *Referenz* auf "`ABCDE`" gesetzt ist) als Datenquelle festlegen. Standard: `DASHBOARD._timeperiod` Alternativ können Sie den Zeitraum nur in den Parametern *Von* und *Bis* festlegen.
Von	1	time_period.from	Gültige Zeitangabe im absoluten (`YYYY-MM-DD hh:mm:ss`) oder relativen Zeitformat (`now`, `now/d`, `now/w-1w` usw.). Parameterverhalten: - *unterstützt*, wenn *Zeitraum* nicht festgelegt ist - *erforderlich*, wenn `time_period.to` festgelegt ist
Bis	1	time_period.to	Gültige Zeitangabe im absoluten (`YYYY-MM-DD hh:mm:ss`) oder relativen Zeitformat (`now`, `now/d`, `now/w-1w` usw.). Parameterverhalten: - *unterstützt*, wenn *Zeitraum* nicht festgelegt ist - *erforderlich*, wenn `time_period.from` festgelegt ist
Einträge sortieren nach	0	sort_triggers	3 - Zeit (aufsteigend); 4 - *(Standard)* Zeit (absteigend); 5 - Typ (aufsteigend); 6 - Typ (absteigend); 7 - Status (aufsteigend); 8 - Status (absteigend); 11 - Empfänger (aufsteigend); 12 - Empfänger (absteigend).
Zeilen anzeigen	0	show_lines	Mögliche Werte liegen zwischen 1 und 100. Standard: 25.

Beispiele

Die folgenden Beispiele sollen lediglich die Konfiguration der Dashboard-Widget-Feldobjekte für das *Aktionsprotokoll*-Widget beschreiben. Weitere Informationen zur Konfiguration eines Dashboards finden Sie unter `dashboard.create`.

Konfiguration eines Widgets *Aktionsprotokoll*

Konfigurieren Sie ein Widget *Aktionsprotokoll*, das 10 Einträge mit Details zu Aktionsoperationen anzeigt, nach Zeit sortiert (in aufsteigender Reihenfolge). Zeigen Sie außerdem nur Details für diejenigen Aktionsoperationen an, die versucht haben, eine E-Mail an Benutzer „1“ zu senden, jedoch nicht erfolgreich waren.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Siehe auch

- Dashboard Widget Feld
- `dashboard.create`
- `dashboard.update`

2 Uhr

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das Widget *Uhr* in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

Attention:
Die Eigenschaften von `fields` des Widgets werden während der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dies ermöglicht es Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets falsch zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Uhr* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Parameterverhalten.

Parameter

Die folgenden Parameter werden für das *Uhr*-Widget unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	0	rf_rate	0 - Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - *(Standard)* 15 Minuten.
Zeittyp	0	time_type	0 - *(Standard)* Lokale Zeit; 1 - Serverzeit; 2 - Host-Zeit.
Uhrtyp	0	clock_type	0 - *(Standard)* Analog; 1 - Digital.

Die folgenden Parameter werden unterstützt, wenn *Zeittyp* auf „Host-Zeit“ gesetzt ist.

Parameter	type	name	value
Datenpunkt	4	itemid.0	ID des Datenpunkts. Parameterverhalten: - *erforderlich*

Die folgenden Parameter werden unterstützt, wenn *Uhrtyp* auf „Digital“ gesetzt ist.

Parameter	type	name	value
Anzeigen	0	show.0	1 - Datum; 2 - *(Standard)* Uhrzeit; 3 - Zeitzone. Hinweis: Um mehrere Werte zu konfigurieren, erstellen Sie für jeden Wert ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Nummer im Eigenschaftsnamen.

Erweiterte Konfiguration

Die folgenden erweiterten Konfigurationsparameter werden unterstützt, wenn *Clock type* auf „Digital“ gesetzt ist.

Parameter	type	name	value
Hintergrundfarbe	1	bg_color	Hexadezimaler Farbcode (z. B. FF0000). Standard: "" (leer).

Datum

Die folgenden erweiterten Konfigurationsparameter werden unterstützt, wenn *Clock type* auf „Digital“ und *Show* auf „Date“ gesetzt ist.

Parameter	type	name	value
Fett	0	date_bold	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
Farbe	1	date_color	Hexadezimaler Farbcode (z. B. FF0000). Standard: "" (leer).

Zeit

Die folgenden erweiterten Konfigurationsparameter werden unterstützt, wenn *Clock type* auf „Digital“ und *Show* auf „Time“ gesetzt ist.

Parameter	type	name	value
Fett	0	time_bold	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
Farbe	1	time_color	Hexadezimaler Farbcode (z. B. FF0000). Standard: "" (leer).
Sekunden	0	time_sec	0 - Deaktiviert; 1 - *(Standard)* Aktiviert.
Format	0	time_format	0 - *(Standard)* 24-Stunden-Format; 1 - 12-Stunden-Format.

Zeitzone

Die folgenden erweiterten Konfigurationsparameter werden unterstützt, wenn *Clock type* auf „Digital“ und *Show* auf „Zeitzone“ gesetzt ist.

Parameter	type	name	value
Fett	0	tzone_bold	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
Farbe	1	tzone_color	Hexadezimaler Farbcode (z. B. FF0000). Standard: "" (leer).

Parameter	type	name	value
Zeitzone	1	tzone_timezone	Gültige Zeitzonen-Zeichenfolge (z. B. `Europe/Riga`, `system`, `UTC` usw.). Eine vollständige Liste der unterstützten Zeitzonen finden Sie in der PHP-Dokumentation. Standard: `local`. Parameterverhalten: - *unterstützt*, wenn *Time type* auf „Local time“ oder „Server time“ gesetzt ist
Format	0	tzone_format	0 - *(Standard)* Kurz; 1 - Vollständig. Parameterverhalten: - *unterstützt*, wenn *Time type* auf „Local time“ oder „Server time“ gesetzt ist

Beispiele

Die folgenden Beispiele beschreiben nur die Konfiguration der Dashboard-Widget-Feldobjekte für das *Clock*-Widget. Weitere Informationen zur Konfiguration eines Dashboards finden Sie unter `dashboard.create`.

Konfigurieren eines *Uhr*-Widgets

Konfigurieren Sie ein *Uhr*-Widget, das lokales Datum, Uhrzeit und Zeitzone in einer angepassten digitalen Uhr anzeigt.

Anfrage:

```
{
    "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
}
```


Antwort:


```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```


Siehe auch

- Dashboard Widget Feld

- dashboard.create
- dashboard.update

3 Status der Entdeckung

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das Widget *Discovery status* in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

Parameter

Die folgenden Parameter werden für das Widget *Discovery status* unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	0	rf_rate	0 - Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - *(Standard)* 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - 15 Minuten.

Beispiele

Die folgenden Beispiele beschreiben nur die Konfiguration der Dashboard-Widget-Feldobjekte für das Widget *Discovery status*. Weitere Informationen zur Konfiguration eines Dashboards finden Sie unter `dashboard.create`.

Konfiguration des Widgets *Discovery status*

Konfigurieren Sie ein Widget *Discovery status* mit einem Aktualisierungsintervall von 15 Minuten.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Siehe auch

- Dashboard Widget Feld
- `dashboard.create`
- `dashboard.update`

4 Bevorzugte Diagramme

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Dashboard-Widget-Feldobjekte können Sie das Widget *Favorite graphs* in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

Parameter

Die folgenden Parameter werden für das Widget *Favorite graphs* unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	integer	rf_rate	0 - Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - *(Standard)* 15 Minuten.

Beispiele

Die folgenden Beispiele beschreiben nur die Konfiguration der Dashboard-Widget-Feldobjekte für das Widget *Favoritengraphen*. Weitere Informationen zur Konfiguration eines Dashboards finden Sie unter `dashboard.create`.

Konfigurieren eines Widgets *Favorite graphs*

Konfigurieren Sie ein Widget *Favorite graphs* mit dem Aktualisierungsintervall auf 10 Minuten gesetzt.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Siehe auch

- Dashboard Widget Feld
- `dashboard.create`
- `dashboard.update`

5 Bevorzugte Karten

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das Widget *Favorite maps* in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

Parameter

Die folgenden Parameter werden für das Widget *Bevorzugte Karten* unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	0	rf_rate	0 - Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - *(Standard)* 15 Minuten.

Beispiele

Die folgenden Beispiele beschreiben nur die Konfiguration der Dashboard-Widget-Feldobjekte für das Widget *Favorite Maps*. Weitere Informationen zur Konfiguration eines Dashboards finden Sie unter `dashboard.create`.

Konfigurieren eines Widgets *Bevorzugte Karten*

Konfigurieren Sie ein Widget *Bevorzugte Karten* mit dem Aktualisierungsintervall auf 10 Minuten gesetzt.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Siehe auch

- Dashboard Widget Feld
- `dashboard.create`
- `dashboard.update`

6 Pegel

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das Widget *Gauge* in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

Attention:
Die Eigenschaften von `fields` des Widgets werden beim Erstellen oder Aktualisieren eines Dashboards nicht validiert. Dies ermöglicht es Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets falsch zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des *Gauge*-Widgets sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Parameterverhalten.

Parameter

Die folgenden Parameter werden für das *Gauge*-Widget unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	0	rf_rate	0 - Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - *(Standard)* 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - 15 Minuten.
Datenpunkt	4	itemid.0	ID des Datenpunkts. Parameterverhalten: - *erforderlich*, wenn *Datenpunkt (Widget)* nicht gesetzt ist

Parameter	type	name	value
Datenpunkt (Widget)	1	itemid._reference	Anstelle der ID des Datenpunkts: `ABCDE._itemid` - legt ein kompatibles Widget (mit dem auf "ABCDE" gesetzten Parameter *Referenz*) als Datenquelle für Datenpunkte fest. Parameterverhalten: - *erforderlich*, wenn *Datenpunkt* nicht gesetzt ist
Min	1	min	Beliebiger numerischer Wert. Suffixe (z. B. "1d", "2w", "4K", "8G") werden unterstützt. Standard: "0".
Max	1	max	Beliebiger numerischer Wert. Suffixe (z. B. "1d", "2w", "4K", "8G") werden unterstützt. Standard: "100".
Wertbogen	1	value_arc_color	Hexadezimaler Farbcode (z. B. `FF0000`). Standard: "" (leer).
Bogenhintergrund	1	empty_color	Hexadezimaler Farbcode (z. B. `FF0000`). Standard: "" (leer).
Hintergrund	1	bg_color	Hexadezimaler Farbcode (z. B. `FF0000`). Standard: "" (leer).
Anzeigen	0	show.0	1 - Beschreibung; 2 - Wert; 3 - Zeiger; 4 - Skala; 5 - Wertbogen. Hinweis: Um mehrere Werte zu konfigurieren, erstellen Sie für jeden Wert ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Nummer im Eigenschaftsnamen. Standard: 1, 2, 4, 5. Die Werte "Zeiger" und "Skala" werden nicht unterstützt, wenn beides zutrifft: - kein Dashboard-Widget-Feldobjekt für *Anzeigen* mit dem Wert "Wertbogen" gesetzt ist; - der Parameter *Bogen anzeigen* der erweiterten Konfiguration auf "Deaktiviert" gesetzt ist. Parameter der erweiterten Konfiguration für *Anzeigen*-Optionen werden nicht unterstützt, wenn keine Dashboard-Widget-Feldobjekte mit den jeweiligen Werten gesetzt sind.
Host überschreiben	1	override_hostid._reference	`ABCDE._hostid` - legt ein kompatibles Widget (mit dem auf "ABCDE" gesetzten Parameter *Referenz*) als Datenquelle für Hosts fest; `DASHBOARD._hostid` - legt die Host-Auswahl des Dashboards als Datenquelle für Hosts fest. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.

Erweiterte Konfiguration

Die folgenden erweiterten Konfigurationsparameter werden für das Widget *Gauge* unterstützt.

Note:
Die Zahl im Namen der Eigenschaft *Thresholds* (z. B. thresholds.0.color) verweist auf die Position des Schwellenwerts in einer aufsteigend sortierten Liste. Wenn die Schwellenwerte jedoch in einer anderen Reihenfolge konfiguriert werden, werden die Werte nach der Aktualisierung der Widget-Konfiguration im Zabbix Frontend aufsteigend sortiert (z. B. `"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"`; `"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`).

Parameter	type	name	value
Winkel	0	angle	Mögliche Werte: 180 *(Standard)* oder 270.
Beschreibung			
Beschreibung	1	description	Beliebiger Zeichenfolgenwert, einschließlich Makros. Unterstützte Makros: {HOST.*}, {ITEM.*}, {INVENTORY.*}, Benutzermakros. Standard: {ITEM.NAME}.
Größe	0	desc_size	Mögliche Werte liegen zwischen 1 und 100. Standard: 15.
Vertikale Position	0	desc_v_pos	0 - Oben; 1 - *(Standard)* Unten.
Fett	0	desc_bold	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
Farbe	1	desc_color	Hexadezimaler Farbcode (z. B. FF0000). Standard: "" (leer).
Wert			
Dezimalstellen	0	decimal_places	Mögliche Werte liegen zwischen 1 und 10. Standard: 2.
Größe	0	value_size	Mögliche Werte liegen zwischen 1 und 100. Standard: 25.
Fett	0	value_bold	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
Farbe	1	value_color	Hexadezimaler Farbcode (z. B. FF0000). Standard: "" (leer).
Einheiten			
Einheiten (Kontrollkästchen)	0	units_show	0 - Deaktiviert; 1 - *(Standard)* Aktiviert.
Einheiten (Wert)	1	units	Beliebiger Zeichenfolgenwert. Parameterverhalten: - *unterstützt*, wenn *Einheiten* (Kontrollkästchen) auf „Aktiviert“ gesetzt ist
Größe	0	units_size	Mögliche Werte liegen zwischen 1 und 100. Standard: 25. Parameterverhalten: - *unterstützt*, wenn *Einheiten* (Kontrollkästchen) auf „Aktiviert“ gesetzt ist
Fett	0	units_bold	0 - *(Standard)* Deaktiviert; 1 - Aktiviert. Parameterverhalten: - *unterstützt*, wenn *Einheiten* (Kontrollkästchen) auf „Aktiviert“ gesetzt ist

Parameter	type	name	value
Position	0	units_pos	0 - Vor dem Wert; 1 - Über dem Wert; 2 - *(Standard)* Nach dem Wert; 3 - Unter dem Wert. Parameterverhalten: - *unterstützt*, wenn *Einheiten* (Kontrollkästchen) auf „Aktiviert“ gesetzt ist Dieser Parameter wird ignoriert, wenn eine der folgenden zeitbezogenen Einheiten festgelegt ist: unixtime, uptime, s.
Farbe	1	units_color	Hexadezimaler Farbcode (z. B. FF0000). Standard: "" (leer).
Wertbogen			
Bogengröße	0	value_arc_size	Mögliche Werte liegen zwischen 1 und 100. Standard: 20.
Zeiger			
Farbe	1	needle_color	Hexadezimaler Farbcode (z. B. FF0000). Standard: "" (leer). Parameterverhalten: - *unterstützt*, wenn ein Dashboard-Widget-Feldobjekt für *Anzeigen* mit dem Wert „Wertbogen“ festgelegt ist oder *Bogen anzeigen* auf „Aktiviert“ gesetzt ist
Skala			
Einheiten anzeigen	0	scale_show_units	0 - Deaktiviert; 1 - *(Standard)* Aktiviert. Parameterverhalten: - *unterstützt*, wenn *Einheiten* (Kontrollkästchen) auf „Aktiviert“ gesetzt ist und entweder ein Dashboard-Widget-Feldobjekt für *Anzeigen* mit dem Wert „Wertbogen“ festgelegt ist oder *Bogen anzeigen* auf „Aktiviert“ gesetzt ist
Größe	0	scale_size	Mögliche Werte liegen zwischen 1 und 100. Standard: 15. Parameterverhalten: - *unterstützt*, wenn ein Dashboard-Widget-Feldobjekt für *Anzeigen* mit dem Wert „Wertbogen“ festgelegt ist oder *Bogen anzeigen* auf „Aktiviert“ gesetzt ist
Dezimalstellen	0	scale_decimal_places	Mögliche Werte liegen zwischen 1 und 10. Standard: 0. Parameterverhalten: - *unterstützt*, wenn ein Dashboard-Widget-Feldobjekt für *Anzeigen* mit dem Wert „Wertbogen“ festgelegt ist oder *Bogen anzeigen* auf „Aktiviert“ gesetzt ist
Schwellenwerte			
Farbe	1	thresholds.0.color	Hexadezimaler Farbcode (z. B. FF0000).
Schwellenwert	1	thresholds.0.threshold	Beliebiger numerischer Wert. Suffixe (z. B. „1d“, „2w“, „4K“, „8G“) werden unterstützt.

Parameter	type	name	value
Beschriftungen anzeigen	0	th_show_labels	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
Bogen anzeigen	0	th_show_arc	Parameterverhalten: - *unterstützt*, wenn *Schwellenwerte* festgelegt sind und entweder ein Dashboard-Widget-Feldobjekt für *Anzeigen* mit dem Wert „Wertbogen“ festgelegt ist oder *Bogen anzeigen* auf „Aktiviert“ gesetzt ist 0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
Bogengröße	0	th_arc_size	Parameterverhalten: - *unterstützt*, wenn *Schwellenwerte* festgelegt sind Mögliche Werte liegen zwischen 1 und 100. Standard: 5. Parameterverhalten: - *unterstützt*, wenn *Bogen anzeigen* auf „Aktiviert“ gesetzt ist

Beispiele

Die folgenden Beispiele beschreiben nur die Konfiguration der Dashboard-Widget-Feldobjekte für das *Gauge*-Widget. Weitere Informationen zur Konfiguration eines Dashboards finden Sie unter `dashboard.create`.

Konfigurieren eines *Gauge*-Widgets

Konfigurieren Sie ein *Gauge*-Widget, das den Datenpunktwert für den Datenpunkt „44474“ (Schnittstelle enp0s3: Gesendete Bits) anzeigt. Zusätzlich können Sie das Widget mit mehreren erweiterten Optionen, einschließlich Schwellenwerten, visuell feinabstimmen.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Siehe auch

- Dashboard Widget Feld
- `dashboard.create`
- `dashboard.update`

7 Landkarte

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das *Geomap*-Widget in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

Attention:
Die Eigenschaften von `fields` des Widgets werden während der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dies ermöglicht es Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets falsch zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des *Geomap*-Widgets sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Parameterverhalten.

Parameter

Die folgenden Parameter werden für das Widget *Geomap* unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	0	rf_rate	0 - Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - *(Standard)* 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - 15 Minuten.
Hostgruppen	2	groupids.0	ID der Hostgruppe. Hinweis: Um mehrere Hostgruppen zu konfigurieren, erstellen Sie für jede Hostgruppe ein Dashboard-Widget-Feldobjekt mit einer fortlaufend erhöhten Zahl im Eigenschaftsnamen. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.

Parameter	type	name	value
Hostgruppen (Widget)	1	groupids._reference	Anstelle der ID der Hostgruppe: `ABCDE._hostgroupids` - Legen Sie ein kompatibles Widget (mit dem Parameter *Referenz*, der auf "`ABCDE`" gesetzt ist) als Datenquelle für Hostgruppen fest. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Hosts	3	hostids.0	ID des Hosts. Hinweis: Um mehrere Hosts zu konfigurieren, erstellen Sie für jeden Host ein Dashboard-Widget-Feldobjekt mit einer fortlaufend erhöhten Zahl im Eigenschaftsnamen. Bei mehreren Hosts muss der Parameter *Hostgruppen* entweder überhaupt nicht konfiguriert oder mit mindestens einer Hostgruppe konfiguriert sein, der die konfigurierten Hosts angehören. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Hosts (Widget/Dashboard)	1	hostids._reference	Anstelle der ID des Hosts: `DASHBOARD.hostids` - Legen Sie den Hostauswahlschalter des Dashboards als Datenquelle für Hosts fest; `ABCDE._hostids` - Legen Sie ein kompatibles Widget (mit dem Parameter *Referenz*, der auf "`ABCDE`" gesetzt ist) als Datenquelle für Hosts fest. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Tags			
Auswertungstyp	0	evaltype	0 - *(Standard)* Und/Oder; 2 - Oder. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Tagname	1	tags.0.tag	Beliebiger Zeichenfolgenwert. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Reihenfolge des Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei der Konfiguration von *Tags* Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Operator	0	tags.0.operator	0 - Enthält; 1 - Ist gleich; 2 - Enthält nicht; 3 - Ist nicht gleich; 4 - Existiert; 5 - Existiert nicht. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Reihenfolge des Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei der Konfiguration von *Tags* Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.

Parameter	type	name	value
Tagwert	1	tags.0.value	Beliebiger Zeichenfolgenwert. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Reihenfolge des Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei der Konfiguration von *Tags* Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Anfangsansicht	1	default_view	Durch Kommas getrennte Werte für *Breitengrad*, *Längengrad* und *Zoomstufe* (*optional*, mögliche Werte liegen zwischen 0 und 30). Beispiel: `40.6892494,-74.0466891,10`.
Clustering			
Clustering-Modus	0	clustering_mode	0 - *(Standard)* Automatisch; 1 - Zoomstufe.
Clustering-Zoomstufe	0	clustering_zoom_level	Mögliche Werte liegen zwischen 0 und 30. Standard: 0. Parameterverhalten: - *unterstützt*, wenn *Clustering-Modus* auf „Zoomstufe“ gesetzt ist
Referenz	1	reference	Beliebiger Zeichenfolgenwert aus 5 Zeichen (z. B. `ABCDE` oder `JBPNL`). Dieser Wert muss innerhalb des Dashboards, zu dem das Widget gehört, eindeutig sein. Parameterverhalten: - *erforderlich*

Beispiele

Die folgenden Beispiele beschreiben nur die Konfiguration der Dashboard-Widget-Feldobjekte für das *Geomap*-Widget. Weitere Informationen zur Konfiguration eines Dashboards finden Sie unter `dashboard.create`.

Konfigurieren eines *Geomap*-Widgets

Konfigurieren Sie ein *Geomap*-Widget, das Hosts aus den Host-Gruppen „2“ und „22“ basierend auf der folgenden Tag-Konfiguration anzeigt: Ein Tag mit dem Namen „component“ enthält den Wert „node“, und ein Tag mit dem Namen „location“ entspricht dem Wert „New York“. Legen Sie außerdem die anfängliche Kartenansicht auf die Koordinaten „40.6892494“ (Breitengrad), „-74.0466891“ (Längengrad) mit der Zoomstufe „10“ fest.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Siehe auch

- Dashboard Widget Feld
- `dashboard.create`
- `dashboard.update`

8 Diagramm

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das Widget *Graph* in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

Attention:
Die Eigenschaften von `fields` des Widgets werden beim Erstellen oder Aktualisieren eines Dashboards nicht validiert. Dies ermöglicht es Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets falsch zu erstellen oder zu aktualisieren. Um sicherzustellen, dass das Widget *Graph* erfolgreich erstellt oder aktualisiert wird, beachten Sie bitte das in den folgenden Tabellen beschriebene Parameterverhalten.

Parameter

Die folgenden Parameter werden für das Widget *Graph* unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	0	rf_rate	0 - Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - *(Standard)* 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - 15 Minuten.
Referenz	1	reference	Beliebiger Zeichenfolgenwert, der aus 5 Zeichen besteht (z. B. `ABCDE` oder `JBPNL`). Dieser Wert muss innerhalb des Dashboards, zu dem das Widget gehört, eindeutig sein. Parameter behavior: - *erforderlich*

Datensatz

Die folgenden Parameter werden für die Konfiguration eines *Datensatzes* unterstützt.

Note:
Die erste Zahl im Eigenschaftsnamen (z. B. ds.0.hosts.0, ds.0.items.0) stellt den jeweiligen Datensatz dar, während die zweite Zahl, sofern vorhanden, den konfigurierten Host oder Datenpunkt darstellt.

Parameter	type	name	value
Datensatztyp	0	ds.0.dataset_type	0 - Datenpunktliste; 1 - *(Standard)* Datenpunktmuster.
Datenpunkte	4	ds.0.itemids.0	ID des Datenpunkts. Wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird, dürfen nur auf der Vorlage konfigurierte Datenpunkte festgelegt werden. Hinweis: Um mehrere Datenpunkte zu konfigurieren, erstellen Sie für jeden Datenpunkt ein Dashboard-Widget-Feldobjekt mit einer erhöhten Zahl im Eigenschaftsnamen. Parameterverhalten: - *erforderlich*, wenn *Datensatztyp* auf „Datenpunktliste“ gesetzt ist und *Datenpunkte (Widget)* nicht festgelegt ist
Datenpunkte (Widget)	1	ds.0.itemids.0._reference	Anstelle der ID des Datenpunkts: `ABCDE._itemid` - legt ein kompatibles Widget (mit dem Parameter *Referenz* auf „ABCDE“ gesetzt) als Datenquelle für Datenpunkte fest. Hinweis: Um mehrere Widgets zu konfigurieren, erstellen Sie für jedes Widget ein Dashboard-Widget-Feldobjekt mit einer erhöhten Zahl im Eigenschaftsnamen. Parameterverhalten: - *erforderlich*, wenn *Datensatztyp* auf „Datenpunktliste“ gesetzt ist und *Datenpunkte* nicht festgelegt ist
Farbe	1	ds.0.color.0	Hexadezimaler Farbcode (z. B. FF0000). Parameterverhalten: - *erforderlich*, wenn *Datensatztyp* auf „Datenpunktliste“ gesetzt ist
Hostmuster	1	ds.0.hosts.0	Name oder Muster des Hosts (z. B. „Zabbix*“). Parameterverhalten: - *erforderlich*, wenn *Datensatztyp* auf „Datenpunktmuster“ gesetzt ist Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Datenpunktmuster	1	ds.0.items.0	Name oder Muster des Datenpunkts (z. B. „*: Anzahl der verarbeiteten *Werte pro Sekunde“). Wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird, dürfen nur die Muster für auf der Vorlage konfigurierte Datenpunkte festgelegt werden. Parameterverhalten: - *erforderlich*, wenn *Datensatztyp* auf „Datenpunktmuster“ gesetzt ist
Farbe	1	ds.0.color	Hexadezimaler Farbcode (z. B. FF0000). Parameterverhalten: - *unterstützt*, wenn *Datensatztyp* auf „Datenpunktmuster“ gesetzt ist und *Farbpalette* nicht festgelegt ist

Parameter	type	name	value
Farbpalette	0	ds.0.color_palette	Index der Farbpalette. Mögliche Werte liegen zwischen 0 und 11. Standard: 0. Parameterverhalten: - *unterstützt*, wenn *Datensatztyp* auf „Datenpunktmuster“ gesetzt ist und *Farbe* nicht festgelegt ist
Darstellung	0	ds.0.type	0 - *(Standard)* Linie; 1 - Punkte; 2 - Treppen; 3 - Balken.
Gestapelt	0	ds.0.stacked	0 - *(Standard)* Deaktiviert; 1 - Aktiviert. Parameterverhalten: - *unterstützt*, wenn *Darstellung* auf „Linie“, „Treppen“ oder „Balken“ gesetzt ist
Breite	0	ds.0.width	Mögliche Werte liegen zwischen 1 und 10. Standard: 1. Parameterverhalten: - *unterstützt*, wenn *Darstellung* auf „Linie“ oder „Treppen“ gesetzt ist
Punktgröße	0	ds.0.pointsize	Mögliche Werte liegen zwischen 1 und 10. Standard: 3. Parameterverhalten: - *unterstützt*, wenn *Darstellung* auf „Punkte“ gesetzt ist
Transparenz	0	ds.0.transparency	Mögliche Werte liegen zwischen 1 und 10. Standard: 5.
Füllung	0	ds.0.fill	Mögliche Werte liegen zwischen 1 und 10. Standard: 3. Parameterverhalten: - *unterstützt*, wenn *Darstellung* auf „Linie“ oder „Treppen“ gesetzt ist
Fehlende Daten	0	ds.0.missingdatafunc	0 - *(Standard)* Keine; 1 - Verbunden; 2 - Als 0 behandeln; 3 - Letzter bekannter Wert. Parameterverhalten: - *unterstützt*, wenn *Darstellung* auf „Linie“ oder „Treppen“ gesetzt ist
Host überschreiben	1	ds.0.override_hostid	`ABCDE._hostid` - legt ein kompatibles Widget (mit dem Parameter „Referenz“ auf „`ABCDE`“ gesetzt) als Datenquelle für Hosts fest; `DASHBOARD._hostid` - legt die Hostauswahl des Dashboards als Datenquelle für Hosts fest. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Y-Achse	0	ds.0.axisy	0 - *(Standard)* Links; 1 - Rechts.
Werte invertieren	0	ds.0.invert_values	0 - (Standard) Deaktiviert; 1 - Aktiviert.

Parameter	type	name	value
Zeitverschiebung	1	ds.0.timeshift	Gültige Zeitzeichenfolge (z. B. 3600, 1h usw.). Sie können Zeitsuffixe verwenden. Negative Werte sind ebenfalls zulässig. Standard: "" (leer).
Aggregationsfunktion	0	ds.0.aggregate_function	0 - *(Standard)* Nicht verwendet; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last.
Aggregationsintervall	1	ds.0.aggregate_interval	Gültige Zeitzeichenfolge (z. B. 3600, 1h usw.). Sie können Zeitsuffixe verwenden. Standard: 1h.
Aggregieren	0	ds.0.aggregate_grouping	0 - *(Standard)* Jeder Datenpunkt; 1 - Datensatz. Parameterverhalten: - *unterstützt*, wenn *Aggregationsfunktion* auf „min“, „max“, „avg“, „count“, „sum“, „first“ oder „last“ gesetzt ist
Approximation	0	ds.0.approximation	1 - min; 2 - *(Standard)* avg; 4 - max; 7 - alle.
Datensatzbezeichnung	1	ds.0.data_set_label	Beliebiger Zeichenfolgenwert. Standard: "" (leer).
Datenpunkt-Tags			
Auswertungstyp	0	ds.0.item_tags_evaltype	0 - *(Standard)* Und/Oder; 2 - Oder.
Tagname	1	ds.0.item_tags.0.tag	Beliebiger Zeichenfolgenwert. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Reihenfolge des Tags in der Tagauswertungsliste. Parameterverhalten: - *erforderlich*, wenn *Datenpunkt-Tags* konfiguriert werden
Operator	0	ds.0.item_tags.0.operator	0 - Enthält; 1 - Ist gleich; 2 - Enthält nicht; 3 - Ist nicht gleich; 4 - Existiert; 5 - Existiert nicht. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Reihenfolge des Tags in der Tagauswertungsliste. Parameterverhalten: - *erforderlich*, wenn *Datenpunkt-Tags* konfiguriert werden
Tagwert	1	ds.0.item_tags.0.value	Beliebiger Zeichenfolgenwert. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Reihenfolge des Tags in der Tagauswertungsliste. Parameterverhalten: - *erforderlich*, wenn *Datenpunkt-Tags* konfiguriert werden

Anzeigeoptionen

Die folgenden Parameter werden zur Konfiguration der *Anzeigeoptionen* unterstützt.

Parameter	type	name	value
Auswahl der Verlaufsdaten	0	source	0 - *(Standard)* Automatisch; 1 - Verlauf; 2 - Trends.
Einfache Auslöser	0	simple_triggers	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
Arbeitszeit	0	working_time	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
Host-Namen in Beschriftungen	0	show_hostnames	0 - *(Standard)* Automatisch; 1 - Anzeigen; 2 - Ausblenden. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Perzentillinie (links)			
Status	0	percentile_left	0 - *(Standard)* Deaktiviert; 1 - Aktiviert. Parameterverhalten: - *unterstützt*, wenn *Y-Achse* (in der Konfiguration von *Datensatz*) auf „Links“ gesetzt ist
Wert	0	percentile_left_value	Mögliche Werte liegen zwischen 1 und 100. Parameterverhalten: - *unterstützt*, wenn *Y-Achse* (in der Konfiguration von *Datensatz*) auf „Links“ gesetzt ist
Perzentillinie (rechts)			
Status	0	percentile_right	0 - *(Standard)* Deaktiviert; 1 - Aktiviert. Parameterverhalten: - *unterstützt*, wenn *Y-Achse* (in der Konfiguration von *Datensatz*) auf „Rechts“ gesetzt ist
Wert	0	percentile_right_value	Mögliche Werte liegen zwischen 1 und 100. Parameterverhalten: - *unterstützt*, wenn *Y-Achse* (in der Konfiguration von *Datensatz*) auf „Rechts“ gesetzt ist

Zeitperiode

Die folgenden Parameter werden für die Konfiguration von *Zeitperiode* unterstützt.

Parameter	type	name	value
Zeitperiode	1	time_period._referenz	`DASHBOARD._timeperiod` - legt den Zeitperiodenwähler des Dashboards als Datenquelle fest; `ABCDE._timeperiod` - legt ein kompatibles Widget (mit dem Parameter *Referenz* auf "`ABCDE`" gesetzt) als Datenquelle fest. Standard: `DASHBOARD._timeperiod` Alternativ können Sie die Zeitperiode nur in den Parametern *Von* und *Bis* festlegen.

Parameter	type	name	value
Von	1	time_period.from	Gültige Zeitangabe in absoluter (`YYYY-MM-DD hh:mm:ss`) oder relativer Zeitsyntax (`now`, `now/d`, `now/w-1w` usw.). Parameter behavior: - *unterstützt*, wenn *Zeitperiode* nicht gesetzt ist - *erforderlich*, wenn `time_period.to` gesetzt ist
Bis	1	time_period.to	Gültige Zeitangabe in absoluter (`YYYY-MM-DD hh:mm:ss`) oder relativer Zeitsyntax (`now`, `now/d`, `now/w-1w` usw.). Parameter behavior: - *unterstützt*, wenn *Zeitperiode* nicht gesetzt ist - *erforderlich*, wenn `time_period.from` gesetzt ist

Achsen

Die folgenden Parameter werden für die Konfiguration von *Achsen* unterstützt.

Parameter	type	name	value
Linke Y-Achse	0	lefty	0 - Deaktiviert; 1 - *(Standard)* Aktiviert. Parameterverhalten: - *unterstützt*, wenn *Y-Achse* (in der Konfiguration von *Datensatz*) auf „Links“ gesetzt ist
Rechte Y-Achse	0	righty	0 - *(Standard)* Deaktiviert; 1 - Aktiviert. Parameterverhalten: - *unterstützt*, wenn *Y-Achse* (in der Konfiguration von *Datensatz*) auf „Rechts“ gesetzt ist
Skalierung	0	lefty_scale righty_scale	0 - *(Standard)* Linear; 1 - Logarithmisch.
Min	1	lefty_min righty_min	Beliebiger numerischer Wert. Standard: "" (leer).
Max	1	lefty_max righty_max	Beliebiger numerischer Wert. Standard: "" (leer).
Einheiten (Typ)	0	lefty_units righty_units	0 - *(Standard)* Automatisch; 1 - Statisch.
Einheiten (Wert)	1	lefty_static_units righty_static_units	Beliebiger Zeichenfolgenwert. Standard: "" (leer).
X-Achse	0	xaxis	0 - Deaktiviert; 1 - *(Standard)* Aktiviert.

Legende

Die folgenden Parameter werden für die Konfiguration der *Legende* unterstützt.

Parameter	type	name	value
Legende anzeigen	0	legend	0 - Deaktiviert; 1 - *(Standard)* Aktiviert.

Parameter	type	name	value
Min./Durchschn./Max. anzeigen	0	legend_statistic	0 - *(Standard)* Deaktiviert; 1 - Aktiviert. Parameterverhalten: - *unterstützt*, wenn *Legende anzeigen* auf „Aktiviert“ gesetzt ist
Aggregationsfunktion anzeigen	0	legend_aggregation	0 - *(Standard)* Deaktiviert; 1 - Aktiviert. Parameterverhalten: - *unterstützt*, wenn *Legende anzeigen* auf „Aktiviert“ gesetzt ist
Zeilen	0	legend_lines_mode	0 - *(Standard)* Fest; 1 - Variabel. Parameterverhalten: - *unterstützt*, wenn *Legende anzeigen* auf „Aktiviert“ gesetzt ist
Anzahl der Zeilen/ Maximale Anzahl der Zeilen	0	legend_lines	Mögliche Werte liegen zwischen 1 und 10. Standard: 1. Parameterverhalten: - *unterstützt*, wenn *Legende anzeigen* auf „Aktiviert“ gesetzt ist
Anzahl der Spalten	0	legend_columns	Mögliche Werte liegen zwischen 1 und 4. Standard: 4. Parameterverhalten: - *unterstützt*, wenn *Legende anzeigen* auf „Aktiviert“ und *Min./Durchschn./Max. anzeigen* auf „Deaktiviert“ gesetzt ist

Probleme

Die folgenden Parameter werden für die Konfiguration von *Problemen* unterstützt.

Parameter	type	name	value
Probleme anzeigen	0	show_problems	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
Nur ausgewählte Datenpunkte	0	graph_item_problems	0 - Deaktiviert; 1 - *(Standard)* Aktiviert.
Problem-Hosts	1	problemhosts.0	Host-Name. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf den konfigurierten Host. Um mehrere Hosts zu konfigurieren, erstellen Sie für jeden Host ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Zahl im Eigenschaftsnamen. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.

Parameter	type	name	value
Schweregrad	0	severities.0	0 - Nicht klassifiziert; 1 - Information; 2 - Warnung; 3 - Durchschnittlich; 4 - Hoch; 5 - Katastrophe. Standard: leer (alle aktiviert). Hinweis: Um mehrere Werte zu konfigurieren, erstellen Sie für jeden Wert ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Zahl im Eigenschaftsnamen.
Problem	1	problem_name	Problem-Ereignisname (Groß-/Kleinschreibung wird nicht beachtet, vollständiger Name oder ein Teil davon).
Problem-Tags			
Auswertungstyp	0	evaltype	0 - *(Standard)* Und/Oder; 2 - Oder.
Tag-Name	1	tags.0.tag	Beliebiger Zeichenfolgenwert. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Reihenfolge der Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei der Konfiguration von *Problem-Tags*
Operator	0	tags.0.operator	0 - Enthält; 1 - Entspricht; 2 - Enthält nicht; 3 - Entspricht nicht; 4 - Existiert; 5 - Existiert nicht. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Reihenfolge der Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei der Konfiguration von *Problem-Tags*
Tag-Wert	1	tags.0.value	Beliebiger Zeichenfolgenwert. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Reihenfolge der Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei der Konfiguration von *Problem-Tags*

Überschreibungen

Die folgenden Parameter werden für die Konfiguration von *Überschreibungen* unterstützt.

Note:
Die erste Zahl im Eigenschaftsnamen (z. B. or.0.hosts.0, or.0.items.0) stellt den jeweiligen Datensatz dar, während die zweite Zahl, sofern vorhanden, den konfigurierten Host oder Datenpunkt darstellt.

Parameter	type	name	value
Host-Muster	1	or.0.hosts.0	Host-Name oder -Muster (z. B. `Zabbix*`). Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird. Parameterverhalten: - *erforderlich* bei der Konfiguration von *Überschreibungen*
Datenpunkt-Muster	1	or.0.items.0	Datenpunkt-Name oder -Muster (z. B. `*: Number of processed *values per second`). Bei der Konfiguration des Widgets auf einem Vorlagen-Dashboard sollten nur die Muster für Datenpunkte festgelegt werden, die auf der Vorlage konfiguriert sind. Parameterverhalten: - *erforderlich* bei der Konfiguration von *Überschreibungen*
Basisfarbe	1	or.0.color	Hexadezimaler Farbcode (z. B. `FF0000`).
Breite	0	or.0.width	Mögliche Werte liegen zwischen 1 und 10.
Darstellung	0	or.0.type	0 - Linie; 1 - Punkte; 2 - Treppen; 3 - Balken.
Transparenz	0	or.0.transparency	Mögliche Werte liegen zwischen 1 und 10.
Füllung	0	or.0.fill	Mögliche Werte liegen zwischen 1 und 10.
Punktgröße	0	or.0.pointsize	Mögliche Werte liegen zwischen 1 und 10.
Fehlende Daten	0	or.0.missingdatafunc	0 - Keine; 1 - Verbunden; 2 - Als 0 behandeln; 3 - Letzter bekannter Wert.
Y-Achse	0	or.0.axisy	0 - Links; 1 - Rechts.
Werte invertieren	0	or.0.invert_values	0 - (Standard) Deaktiviert; 1 - Aktiviert.
Zeitverschiebung	1	or.0.timeshift	Gültige Zeitangabe (z. B. `3600`, `1h` usw.). Sie können Zeitsuffixe verwenden. Negative Werte sind zulässig.

Beispiele

Die folgenden Beispiele beschreiben nur die Konfiguration der Dashboard-Widget-Feldobjekte für das *Graph* -Widget. Für weitere Informationen zur Konfiguration eines Dashboards siehe `dashboard.create`.

Konfigurieren eines *Graph*-Widgets

Konfigurieren Sie ein *Graph*-Widget wie folgt:

- 2 Datensätze mit insgesamt 9 Datenpunkten auf 1 Host.
- Der erste Datensatz ist vom Typ „Datenpunktliste“ und besteht aus 3 Datenpunkten, die als Linien mit unterschiedlicher Farbe, aber gleicher Breite, Transparenz und Füllung dargestellt werden.
- Der zweite Datensatz ist vom Typ „Datenpunktmuster“, besteht aus 6 Datenpunkten, hat eine konfigurierte Aggregation und wird als Linie mit benutzerdefinierter Farbe, Breite, Transparenz und Füllung dargestellt.
- Der zweite Datensatz hat außerdem eine benutzerdefinierte Datensatzbeschriftung.
- Daten im Diagramm werden für einen Zeitraum der letzten 3 Stunden angezeigt.
- Probleme im Diagramm werden nur für die konfigurierten Datenpunkte angezeigt.
- Das Diagramm hat zwei Y-Achsen, wobei die rechte Y-Achse Werte nur für den zweiten Datensatz anzeigt.
- Die Diagrammlegende zeigt konfigurierte Datenpunkte in 4 Zeilen sowie Minimal-, Maximal- und Durchschnittswerte der Datensätze an.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Siehe auch

- Dashboard Widget Feld
- `dashboard.create`
- `dashboard.update`

9 Diagramm (klassisch)

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das Widget *Graph (klassisch)* in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

Attention:
Die Eigenschaften von `fields` des Widgets werden während der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dies ermöglicht es Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets falsch zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Graph (klassisch)* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Parameterverhalten.

Parameter

Die folgenden Parameter werden für das Widget *Graph (classic)* unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	0	rf_rate	0 - Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - *(Standard)* 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - 15 Minuten.
Quelle	0	source_type	0 - *(Standard)* Graph; 1 - Einfacher Graph.
Graph	6	graphid.0	Graph-ID. Parameterverhalten: - *erforderlich*, wenn *Quelle* auf „Graph“ gesetzt ist
Graph (Widget)	1	graphid._reference	Anstelle der Graph-ID: `ABCDE._graphid` - legt ein kompatibles Widget (mit dem auf "ABCDE" gesetzten Parameter *Referenz*) als Datenquelle für Graphen fest. Parameterverhalten: - *erforderlich*, wenn *Quelle* auf „Simple graph“ gesetzt ist und *Graph* nicht gesetzt ist
Datenpunkt	4	itemid.0	Item-ID. Parameterverhalten: - *erforderlich*, wenn *Quelle* auf „Simple graph“ gesetzt ist und *Datenpunkt (Widget)* nicht gesetzt ist
Datenpunkt (Widget)	1	itemid._reference	Anstelle der Item-ID: `ABCDE._itemid` - legt ein kompatibles Widget (mit dem auf "ABCDE" gesetzten Parameter *Referenz*) als Datenquelle für Datenpunkte fest. Parameterverhalten: - *erforderlich*, wenn *Quelle* auf „Simple graph“ gesetzt ist und *Datenpunkt* nicht gesetzt ist
Zeitperiode	1	time_period._reference	`DASHBOARD._timeperiod` - legt die Zeitperiodenauswahl des Dashboards als Datenquelle fest; `ABCDE._timeperiod` - legt ein kompatibles Widget (mit dem auf "ABCDE" gesetzten Parameter *Referenz*) als Datenquelle fest. Standard: `DASHBOARD._timeperiod` Alternativ können Sie die Zeitperiode nur in den Parametern *Von* und *Bis* festlegen.

Parameter	type	name	value
Von	1	time_period.from	Gültige Zeitzeichenfolge in absoluter (`YYYY-MM-DD hh:mm:ss`) oder relativer Zeitsyntax (`now`, `now/d`, `now/w-1w` usw.). Parameterverhalten: - *unterstützt*, wenn *Zeitperiode* nicht gesetzt ist - *erforderlich*, wenn `time_period.to` gesetzt ist
Bis	1	time_period.to	Gültige Zeitzeichenfolge in absoluter (`YYYY-MM-DD hh:mm:ss`) oder relativer Zeitsyntax (`now`, `now/d`, `now/w-1w` usw.). Parameterverhalten: - *unterstützt*, wenn *Zeitperiode* nicht gesetzt ist - *erforderlich*, wenn `time_period.from` gesetzt ist
Legende anzeigen	0	show_legend	0 - Deaktiviert; 1 - *(Standard)* Aktiviert.
Host überschreiben	1	override_hostid._refe	`ABCDE._hostid` - legt ein kompatibles Widget (mit dem auf "`ABCDE`" gesetzten Parameter *Referenz*) als Datenquelle für Hosts fest; `DASHBOARD._hostid` - legt die Host-Auswahl des Dashboards als Datenquelle für Hosts fest. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Referenz	1	reference	Beliebiger Zeichenfolgenwert, der aus 5 Zeichen besteht (z. B. `ABCDE` oder `JBPNL`). Dieser Wert muss innerhalb des Dashboards, zu dem das Widget gehört, eindeutig sein. Parameterverhalten: - *erforderlich*

Beispiele

Die folgenden Beispiele beschreiben nur die Konfiguration der Dashboard-Widget-Feldobjekte für das *Graph (classic)* Widget. Für weitere Informationen zur Konfiguration eines Dashboards siehe `dashboard.create`.

Konfigurieren eines Widgets *Graph (classic)*

Konfigurieren Sie ein Widget *Graph (classic)*, das ein einfaches Diagramm für den Datenpunkt „42269“ anzeigt.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Siehe auch

- Dashboard Widget Feld
- `dashboard.create`
- `dashboard.update`

10 Diagramm Prototyp

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das Widget *Graph-Prototyp* in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

Attention:
Die Eigenschaften der `fields` des Widgets werden beim Erstellen oder Aktualisieren eines Dashboards nicht validiert. Dies ermöglicht es Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets falsch zu erstellen oder zu aktualisieren. Um sicherzustellen, dass das Widget *Graph-Prototyp* erfolgreich erstellt oder aktualisiert wird, beachten Sie bitte das in den folgenden Tabellen beschriebene Parameterverhalten.

Parameter

Die folgenden Parameter werden für das Widget *Graph-Prototyp* unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	0	rf_rate	0 - Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - *(Standard)* 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - 15 Minuten.
Quelle	0	source_type	2 - *(Standard)* Graph-Prototyp; 3 - Einfacher Graph-Prototyp.
Graph-Prototyp	7	graphid.0	ID des Graph-Prototyps. Parameterverhalten: - *erforderlich*, wenn *Quelle* auf „Graph-Prototyp“ gesetzt ist
Datenpunkt-Prototyp	5	itemid.0	ID des Datenpunkt-Prototyps. Parameterverhalten: - *erforderlich*, wenn *Quelle* auf „Einfacher Graph-Prototyp“ gesetzt ist
Zeitraum	1	time_period._reference	`DASHBOARD._timeperiod` - den Zeitraumauswahlschalter des Dashboards als Datenquelle festlegen; `ABCDE._timeperiod` - ein kompatibles Widget (mit dem Parameter *Referenz*, der auf „`ABCDE`“ gesetzt ist) als Datenquelle festlegen. Standard: `DASHBOARD._timeperiod` Alternativ können Sie den Zeitraum nur in den Parametern *Von* und *Bis* festlegen.
Von	1	time_period.from	Gültige Zeitangabe im absoluten Format (`YYYY-MM-DD hh:mm:ss`) oder in der relativen Zeitsyntax (`now`, `now/d`, `now/w-1w` usw.). Parameterverhalten: - *unterstützt*, wenn *Zeitraum* nicht festgelegt ist - *erforderlich*, wenn `time_period.to` festgelegt ist
Bis	1	time_period.to	Gültige Zeitangabe im absoluten Format (`YYYY-MM-DD hh:mm:ss`) oder in der relativen Zeitsyntax (`now`, `now/d`, `now/w-1w` usw.). Parameterverhalten: - *unterstützt*, wenn *Zeitraum* nicht festgelegt ist - *erforderlich*, wenn `time_period.from` festgelegt ist
Legende anzeigen	0	show_legend	0 - Deaktiviert; 1 - *(Standard)* Aktiviert.
Host überschreiben	1	override_hostid._reference	`ABCDE._hostid` - ein kompatibles Widget (mit dem Parameter *Referenz*, der auf „`ABCDE`“ gesetzt ist) als Datenquelle für Hosts festlegen; `DASHBOARD._hostid` - den Host-Auswahlschalter des Dashboards als Datenquelle für Hosts festlegen. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Spalten	0	columns	Mögliche Werte liegen zwischen 1 und 24. Standard: 2.
Zeilen	0	rows	Mögliche Werte liegen zwischen 1 und 16. Standard: 1.

Parameter	type	name	value
Referenz	1	reference	Beliebiger Zeichenfolgenwert aus 5 Zeichen (z. B. `ABCDE` oder `JBPNL`). Dieser Wert muss innerhalb des Dashboards, zu dem das Widget gehört, eindeutig sein. Parameterverhalten: - *erforderlich*

Beispiele

Die folgenden Beispiele beschreiben nur die Konfiguration der Dashboard-Widget-Feldobjekte für das *Graph prototype* Widget. Für weitere Informationen zur Konfiguration eines Dashboards siehe `dashboard.create`.

Konfiguration eines Widgets *Graph prototype*

Konfigurieren Sie ein Widget *Graph prototype*, das ein Raster aus 3 Diagrammen (3 Spalten, 1 Zeile) anzeigt, die durch Low-Level-Discovery aus einem Datenpunkt-Prototypen (ID: ”42316”) erstellt wurden.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Siehe auch

- Dashboard Widget Feld
- `dashboard.create`
- `dashboard.update`

11 Honigwabe

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das *Honeycomb*-Widget in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

> **Attention:**
> Die Eigenschaften von `fields` des Widgets werden während der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dies ermöglicht es Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets falsch zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des *Honeycomb*-Widgets sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Parameterverhalten.

Parameter

Die folgenden Parameter werden für das *Honeycomb*-Widget unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	0	rf_rate	0 - Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - *(Standard)* 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - 15 Minuten.

Parameter	type	name	value
Host-Gruppen	2	groupids.0	ID der Host-Gruppe. Hinweis: Um mehrere Host-Gruppen zu konfigurieren, erstellen Sie für jede Host-Gruppe ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Nummer im Eigenschaftsnamen. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Host-Gruppen (Widget)	1	groupids._reference	Anstelle der ID der Host-Gruppe: `ABCDE._hostgroupids` - ein kompatibles Widget (mit dem Parameter *Referenz* auf "`ABCDE`" gesetzt) als Datenquelle für Host-Gruppen festlegen. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Hosts	3	hostids.0	ID des Hosts. Hinweis: Um mehrere Hosts zu konfigurieren, erstellen Sie für jeden Host ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Nummer im Eigenschaftsnamen. Bei mehreren Hosts darf der Parameter *Host-Gruppen* entweder gar nicht konfiguriert sein oder muss mit mindestens einer Host-Gruppe konfiguriert sein, zu der die konfigurierten Hosts gehören. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Hosts (Widget/Dashboard)	1	hostids._reference	Anstelle der ID des Hosts: `DASHBOARD.hostids` - die Host-Auswahl des Dashboards als Datenquelle für Hosts festlegen; `ABCDE._hostids` - ein kompatibles Widget (mit dem Parameter *Referenz* auf "`ABCDE`" gesetzt) als Datenquelle für Hosts festlegen. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Host-Tags			
Auswertungstyp	0	evaltype_host	0 - *(Standard)* Und/Oder; 2 - Oder. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Tag-Name	1	host_tags.0.tag	Beliebiger Zeichenfolgenwert. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf die Tag-Reihenfolge in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei Konfiguration von *Host-Tags* Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.

Parameter	type	name	value
Operator	0	host_tags.0.operator	0 - Enthält; 1 - Entspricht; 2 - Enthält nicht; 3 - Entspricht nicht; 4 - Existiert; 5 - Existiert nicht. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf die Tag-Reihenfolge in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei Konfiguration von *Host-Tags* Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Tag-Wert	1	host_tags.0.value	Beliebiger Zeichenfolgenwert. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf die Tag-Reihenfolge in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei Konfiguration von *Host-Tags* Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Datenpunkt-Muster	1	items.0	Name oder Muster des Datenpunkts. Hinweis: Um mehrere Datenpunkt-Muster zu konfigurieren, erstellen Sie für jedes Datenpunkt-Muster ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Nummer im Eigenschaftsnamen. Parameterverhalten: - *erforderlich*
Datenpunkt-Tags			
Auswertungstyp	0	evaltype_item	0 - *(Standard)* Und/Oder; 2 - Oder.
Tag-Name	1	item_tags.0.tag	Beliebiger Zeichenfolgenwert. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf die Tag-Reihenfolge in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei Konfiguration von *Datenpunkt-Tags*
Operator	0	item_tags.0.operator	0 - Enthält; 1 - Entspricht; 2 - Enthält nicht; 3 - Entspricht nicht; 4 - Existiert; 5 - Existiert nicht. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf die Tag-Reihenfolge in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei Konfiguration von *Datenpunkt-Tags*

Parameter	type	name	value
Tag-Wert	1	item_tags.0.value	Beliebiger Zeichenfolgenwert. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf die Tag-Reihenfolge in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei Konfiguration von *Datenpunkt-Tags*
Hosts in Wartung anzeigen	0	maintenance	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
Anzeigen	0	show.0	1 - Primäre Beschriftung; 2 - Sekundäre Beschriftung. Hinweis: Um mehrere Werte zu konfigurieren, erstellen Sie für jeden Wert ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Nummer im Eigenschaftsnamen. Standard: 1, 2.
Referenz	1	reference	Beliebiger Zeichenfolgenwert mit 5 Zeichen (z. B. `ABCDE` oder `JBPNL`). Dieser Wert muss innerhalb des Dashboards, zu dem das Widget gehört, eindeutig sein. Parameterverhalten: - *erforderlich*

Erweiterte Konfiguration

Die folgenden erweiterten Konfigurationsparameter werden für das Widget *Honeycomb* unterstützt.

Note:
Die Zahl im Namen der Eigenschaft *Schwellenwerte* (z. B. thresholds.0.color) verweist auf die Position des Schwellenwerts in einer aufsteigend sortierten Liste. Wenn die Schwellenwerte jedoch in einer anderen Reihenfolge konfiguriert werden, werden die Werte nach der Aktualisierung der Widget-Konfiguration im Zabbix Frontend aufsteigend sortiert (z. B. `"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"`; `"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`).

Parameter	type	name	value
Primäre Beschriftung			
Typ	0	primary_label_type	0 - *(Standard)* Text; 1 - Wert.
Text	1	primary_label	Beliebiger Zeichenfolgenwert, einschließlich Makros. Unterstützte Makros: `{HOST.*}`, `{ITEM.*}`, `{INVENTORY.*}`, Benutzermakros. Standard: `{HOST.NAME}` Parameterverhalten: - *unterstützt*, wenn *Typ* auf „Text“ gesetzt ist
Dezimalstellen	0	primary_label_decimal_places	Mögliche Werte reichen von 0 bis 6. Standard: 2. Parameterverhalten: - *unterstützt*, wenn *Typ* auf „Wert“ gesetzt ist
Größe (Typ)	0	primary_label_size_type	0 - *(Standard)* Automatisch; 1 - Benutzerdefiniert.

Parameter	type	name	value
Größe	0	primary_label_size	Mögliche Werte reichen von 1 bis 100. Standard: 20. Parameterverhalten: - *unterstützt*, wenn *Größe* (Typ) auf „Benutzerdefiniert“ gesetzt ist
Fett	0	primary_label_bold	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
Farbe	1	primary_label_color	Hexadezimaler Farbcode (z. B. `FF0000`). Standard: basierend auf `theme` des Settings-Objekts und des Benutzerobjekts: `1F2C33` für „blue-theme“, „blue-classic-theme“ oder „hc-light“; `EEEEEE` für „dark-theme“, „dark-classic-theme“ oder „hc-dark“; `E5F1FF` für „dark-blue-theme“.
Einheiten (Kontrollkästchen)	0	primary_label_units_show	0 - Deaktiviert; 1 - *(Standard)* Aktiviert. Parameterverhalten: - *unterstützt*, wenn *Typ* auf „Wert“ gesetzt ist
Einheiten (Wert)	1	primary_label_units	Beliebiger Zeichenfolgenwert. "" (leer) Parameterverhalten: - *unterstützt*, wenn *Typ* auf „Wert“ und *Einheiten* (Kontrollkästchen) auf „Aktiviert“ gesetzt ist
Position	0	primary_label_units_pos	0 - Vor dem Wert; 1 - *(Standard)* Nach dem Wert. Parameterverhalten: - *unterstützt*, wenn *Typ* auf „Wert“ und *Einheiten* (Kontrollkästchen) auf „Aktiviert“ gesetzt ist Dieser Parameter wird ignoriert, wenn er auf eine der folgenden zeitbezogenen Einheiten gesetzt ist: unixtime, uptime, s.
Sekundäre Beschriftung			
Typ	0	secondary_label_type	0 - Text; 1 - *(Standard)* Wert.
Text	1	secondary_label	Beliebiger Zeichenfolgenwert, einschließlich Makros. Unterstützte Makros: `{HOST.*}`, `{ITEM.*}`, `{INVENTORY.*}`, Benutzermakros. Standard: `{{ITEM.LASTVALUE}.fmtnum(2)}` Parameterverhalten: - *unterstützt*, wenn *Typ* auf „Text“ gesetzt ist
Dezimalstellen	0	secondary_label_decimal_places	Mögliche Werte reichen von 0 bis 6. Standard: 2. Parameterverhalten: - *unterstützt*, wenn *Typ* auf „Wert“ gesetzt ist
Größe (Typ)	0	secondary_label_size_type	0 - *(Standard)* Automatisch; 1 - Benutzerdefiniert.
Größe	0	secondary_label_size	Mögliche Werte reichen von 1 bis 100. Standard: 30. Parameterverhalten: - *unterstützt*, wenn *Größe* (Typ) auf „Benutzerdefiniert“ gesetzt ist

Parameter	type	name	value
Fett	0	secondary_label_bold	0 - Deaktiviert; 1 - *(Standard)* Aktiviert.
Farbe	1	secondary_label_color	Hexadezimaler Farbcode (z. B. FF0000). Standard: basierend auf theme des Settings-Objekts und des Benutzerobjekts: 1F2C33 für „blue-theme“, „blue-classic-theme“ oder „hc-light“; EEEEEE für „dark-theme“, „dark-classic-theme“ oder „hc-dark“; E5F1FF für „dark-blue-theme“.
Einheiten (Kontrollkästchen)	0	secondary_label_units_show	0 - Deaktiviert; 1 - *(Standard)* Aktiviert. Parameterverhalten: - *unterstützt*, wenn *Typ* auf „Wert“ gesetzt ist
Einheiten (Wert)	1	secondary_label_units	Beliebiger Zeichenfolgenwert. "" (leer) Parameterverhalten: - *unterstützt*, wenn *Typ* auf „Wert“ und *Einheiten* (Kontrollkästchen) auf „Aktiviert“ gesetzt ist
Position	0	secondary_label_position	0 - Vor dem Wert; 1 - *(Standard)* Nach dem Wert. Parameterverhalten: - *unterstützt*, wenn *Typ* auf „Wert“ und *Einheiten* (Kontrollkästchen) auf „Aktiviert“ gesetzt ist Dieser Parameter wird ignoriert, wenn er auf eine der folgenden zeitbezogenen Einheiten gesetzt ist: unixtime, uptime, s.
Hintergrundfarbe			
Hintergrundfarbe	1	bg_color	Hexadezimaler Farbcode (z. B. FF0000). Standard: basierend auf theme des Settings-Objekts und des Benutzerobjekts: D9E7ED für „blue-theme“ und „blue-classic-theme“; 3D5059 für „dark-theme“ und „dark-classic-theme“; 004FA8 für „dark-blue-theme“; AAD7E9 für „hc-light“; 335463 für „hc-dark“.
Schwellenwerte			
Farbinterpolation	0	interpolation	0 - Deaktiviert; 1 - *(Standard)* Aktiviert.
Farbe	1	thresholds.0.color	Hexadezimaler Farbcode (z. B. FF0000).
Schwellenwert	1	thresholds.0.threshold	Beliebiger numerischer Wert. Suffixe (z. B. „1d“, „2w“, „4K“, „8G“) werden unterstützt.

Beispiele

Die folgenden Beispiele beschreiben nur die Konfiguration der Dashboard-Widget-Feldobjekte für das *Honeycomb* -Widget. Für weitere Informationen zur Konfiguration eines Dashboards siehe `dashboard.create`.

Konfigurieren eines *Honeycomb*-Widgets

Konfigurieren Sie ein *Honeycomb*-Widget, das die Auslastung von Zabbix-Serverprozessen anzeigt. Ändern Sie außerdem die primäre Beschriftung der Honeycomb-Zellen und passen Sie das Widget visuell mit Schwellenwerten an.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Siehe auch

- Dashboard Widget Feld
- `dashboard.create`
- `dashboard.update`

12 Verfügbarkeit des Hosts

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das Widget *Host-Verfügbarkeit* in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

Parameter

Die folgenden Parameter werden für das Widget *Host-Verfügbarkeit* unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	0	rf_rate	0 - Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - *(Standard)* 15 Minuten.
Host-Gruppen	2	groupids.0	ID der Host-Gruppe. Hinweis: Um mehrere Host-Gruppen zu konfigurieren, erstellen Sie für jede Host-Gruppe ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Nummer im Eigenschaftsnamen. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Host-Gruppen (Widget)	1	groupids._reference	Anstelle der ID der Host-Gruppe: `ABCDE._hostgroupids` - legen Sie ein kompatibles Widget (mit dem Parameter *Referenz* auf "ABCDE" gesetzt) als Datenquelle für Host-Gruppen fest. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Schnittstellentyp	0	interface_type.0	0 - Keine; 1 - Zabbix-Agent (passive Prüfungen); 2 - SNMP; 3 - IPMI; 4 - JMX; 5 - Zabbix-Agent (aktive Prüfungen). Standard: 1, 2, 3, 4, 5 (alle aktiviert). Hinweis: Um mehrere Werte zu konfigurieren, erstellen Sie für jeden Wert ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Nummer im Eigenschaftsnamen.
Layout	0	layout	0 - *(Standard)* Horizontal; 1 - Vertikal.
Hosts in Wartung anzeigen	0	maintenance	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
Nur Summen anzeigen	0	only_totals	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.

Beispiele

Die folgenden Beispiele beschreiben nur die Konfiguration der Feldobjekte des Dashboard-Widgets für das *Host availability* Widget. Für weitere Informationen zur Konfiguration eines Dashboards siehe `dashboard.create`.

Konfiguration eines Widgets *Host availability*

Konfigurieren Sie ein Widget *Host availability*, das Verfügbarkeitsinformationen (in einem vertikalen Layout) für Hosts in der Host-Gruppe „4“ mit konfigurierten „Zabbix agent“- und „SNMP“-Schnittstellen anzeigt.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
```

```
    "id": 1
}
```

Siehe auch

- Dashboard Widget Feld
- `dashboard.create`
- `dashboard.update`

13 Host-Karte

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das Widget *Host card* in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

Attention:
Die Eigenschaften von `fields` des Widgets werden während der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dies ermöglicht es Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets falsch zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Host card* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Parameterverhalten.

Parameter

Die folgenden Parameter werden für das Widget *Host card* unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	0	rf_rate	0 - Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - *(Standard)* 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - 15 Minuten.
Host	3	hostid.0	ID des Hosts. Parameterverhalten: - *erforderlich*, wenn *Host (Widget/Dashboard)* nicht gesetzt ist Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Host (Widget/Dashboard)	1	hostid._reference	Anstelle der ID des Hosts: `DASHBOARD.hostid` - den Host-Selektor des Dashboards als Datenquelle für den Host festlegen; `ABCDE._hostid` - ein kompatibles Widget (mit dem Parameter *Reference* auf "`ABCDE`" gesetzt) als Datenquelle für den Host festlegen. Parameterverhalten: - *erforderlich*, wenn *Host* nicht gesetzt ist Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Unterdrückte Probleme anzeigen	0	show_suppressed	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.

Parameter	type	name	value
Anzeigen	0	sections.0	0 - Host-Gruppen; 1 - Beschreibung; 2 - Überwachung; 3 - Verfügbarkeit; 4 - Überwacht von; 5 - Vorlagen; 6 - Inventar; 7 - Tags. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Reihenfolge des Abschnitts in der Abschnittsliste. Um mehrere Abschnitte zu konfigurieren, erstellen Sie für jeden Abschnitt ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Zahl im Eigenschaftsnamen.

Die folgenden Parameter werden unterstützt, wenn *Anzeigen* auf ”Inventar” gesetzt ist.

Parameter	type	name	value
Inventarfelder	0	inventory.0	ID des Inventars. Hinweis: Um mehrere Inventarfelder zu konfigurieren, erstellen Sie für jedes Inventarfeld ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Zahl im Eigenschaftsnamen.

Beispiele

Die folgenden Beispiele beschreiben nur die Konfiguration der Feldobjekte des Dashboard-Widgets für das *Host card* Widget. Für weitere Informationen zur Konfiguration eines Dashboards siehe `dashboard.create`.

Konfigurieren eines *Host card*-Widgets

Konfigurieren Sie ein *Host card*-Widget, das die folgenden Abschnitte anzeigt: „Monitoring“, „Availability“, „Monitored by“, „Inventory“ und „Tags“.

Anfrage:

```
{
    "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
}
```


Antwort:


```
{
    "jsonrpc": "2.0",
```

```
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Siehe auch

- Dashboard Widget Feld
- `dashboard.create`
- `dashboard.update`

14 Host-Navigator

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das Widget *Host navigator* in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

Attention:
Die Eigenschaften von `fields` des Widgets werden während der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dies ermöglicht es Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets falsch zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Host navigator* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Parameterverhalten.

Parameter

Die folgenden Parameter werden für das Widget *Host-Navigator* unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	0	rf_rate	0 - Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - *(Standard)* 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - 15 Minuten.
Host-Gruppen	2	groupids.0	ID der Host-Gruppe. Hinweis: Um mehrere Host-Gruppen zu konfigurieren, erstellen Sie für jede Host-Gruppe ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Zahl im Eigenschaftsnamen. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Host-Gruppen (Widget)	1	groupids._reference	Anstelle der ID der Host-Gruppe: `ABCDE._hostgroupids` - Legen Sie ein kompatibles Widget (mit dem Parameter *Referenz* auf "ABCDE" gesetzt) als Datenquelle für Host-Gruppen fest. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.

Parameter	type	name	value
Host-Muster	1	hosts.0	Name oder Muster des Hosts. Hinweis: Um mehrere Host-Muster zu konfigurieren, erstellen Sie für jedes Host-Muster ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Zahl im Eigenschaftsnamen. Bei mehreren Host-Mustern muss der Parameter *Host-Gruppen* entweder überhaupt nicht konfiguriert oder mit mindestens einer Host-Gruppe konfiguriert werden, zu der die Hosts gehören, die den konfigurierten Host-Mustern entsprechen. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Host-Status	0	status	-1 - *(Standard)* Beliebig; 0 - Aktiviert; 1 - Deaktiviert. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Host-Tags			
Auswertungstyp	0	host_tags_evaltype	0 - *(Standard)* Und/Oder; 2 - Oder. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Tag-Name	1	host_tags.0.tag	Beliebiger Zeichenfolgenwert. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Tag-Reihenfolge in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich*, wenn *Host-Tags* konfiguriert werden Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Operator	0	host_tags.0.operator	0 - Enthält; 1 - Ist gleich; 2 - Enthält nicht; 3 - Ist nicht gleich; 4 - Existiert; 5 - Existiert nicht. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Tag-Reihenfolge in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich*, wenn *Host-Tags* konfiguriert werden Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Tag-Wert	1	host_tags.0.value	Beliebiger Zeichenfolgenwert. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Tag-Reihenfolge in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich*, wenn *Host-Tags* konfiguriert werden Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.

Parameter	type	name	value
Schweregrad	0	severities.0	0 - Nicht klassifiziert; 1 - Information; 2 - Warnung; 3 - Durchschnittlich; 4 - Hoch; 5 - Katastrophe. Standard: leer (alle aktiviert). Hinweis: Um mehrere Werte zu konfigurieren, erstellen Sie für jeden Wert ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Zahl im Eigenschaftsnamen.
Hosts in Wartung anzeigen	0	maintenance	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
Probleme anzeigen	0	show_problems	0 - Alle; 1 - *(Standard)* Nicht unterdrückte; 2 - Keine.
Gruppieren nach			
Attribut	0	group_by.0.attribute	0 - Host-Gruppe; 1 - Tag-Wert; 2 - Schweregrad. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Attributreihenfolge in der Liste der Gruppierungsattribute. Parameterverhalten: - *erforderlich*, wenn *Gruppieren nach* konfiguriert wird
Wert	1	group_by.0.tag_name	Beliebiger Zeichenfolgenwert. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf das im Parameter *Attribut* festgelegte Gruppierungsattribut. Parameterverhalten: - *erforderlich*, wenn *Gruppieren nach* konfiguriert wird und *Attribut* auf ”Tag-Wert” gesetzt ist
Host-Limit	0	show_lines	Mögliche Werte liegen zwischen 1 und 9999. Standard: 100. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Referenz	1	reference	Beliebiger Zeichenfolgenwert aus 5 Zeichen (z. B. `ABCDE` oder `JBPNL`). Dieser Wert muss innerhalb des Dashboards, zu dem das Widget gehört, eindeutig sein. Parameterverhalten: - *erforderlich*

Beispiele

Die folgenden Beispiele beschreiben nur die Konfiguration der Dashboard-Widget-Feldobjekte für das *Host navigator* Widget. Für weitere Informationen zur Konfiguration eines Dashboards siehe `dashboard.create`.

Konfiguration eines *Host navigator*-Widgets

Konfigurieren Sie ein *Host navigator*-Widget, das Hosts anzeigt, die nach ihrer Host-Gruppe und anschließend nach dem Wert des Tags „city“ gruppiert sind.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Siehe auch

- Dashboard Widget Feld
- `dashboard.create`
- `dashboard.update`

15 Datenpunkt-Karte

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das Widget *Datenpunktkarte* in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

Attention:
Die Eigenschaften von `fields` des Widgets werden während der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dies ermöglicht es Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets falsch zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Datenpunktkarte* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Parameterverhalten.

Parameter

Die folgenden Parameter werden für das Widget *Datenpunkt-Karte* unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	0	rf_rate	0 - Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - *(Standard)* 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - 15 Minuten.
Datenpunkt	4	itemid.0	Datenpunkt-ID.
Datenpunkt (Widget)	1	itemid._reference	Parameterverhalten: - *erforderlich*, wenn *Datenpunkt (Widget)* nicht gesetzt ist Anstelle der Datenpunkt-ID: `ABCDE._itemid` - legt ein kompatibles Widget (mit dem auf "ABCDE" gesetzten Parameter *Referenz*) als Datenquelle für den Datenpunkt fest. Parameterverhalten: - *erforderlich*, wenn *Datenpunkt* nicht gesetzt ist

Parameter	type	name	value
Anzeigen	0	sections.0	0 - Beschreibung; 1 - Fehlertext; 2 - Metriken; 3 - Letzte Daten; 4 - Informationstyp; 5 - Auslöser; 6 - Host-Schnittstelle; 7 - Typ; 8 - Host-Inventar; 9 - Tags. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Reihenfolge des Abschnitts in der Abschnittsliste. Um mehrere Abschnitte zu konfigurieren, erstellen Sie für jeden Abschnitt ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Zahl im Eigenschaftsnamen.
Host überschreiben	1	override_hostid._reference	`ABCDE._hostid` - legt ein kompatibles Widget (mit dem auf "`ABCDE`" gesetzten Parameter *Referenz*) als Datenquelle für Hosts fest; `DASHBOARD._hostid` - legt die Host-Auswahl des Dashboards als Datenquelle für Hosts fest. Standard: "" (leer) Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.

Sparkline

Die folgenden Parameter werden unterstützt, wenn *Anzeigen* auf „Letzte Daten“ gesetzt ist.

Parameter	type	name	value
Breite	0	sparkline.width	Mögliche Werte liegen zwischen 0 und 10. Standard: 1.
Füllung	0	sparkline.fill	Mögliche Werte liegen zwischen 0 und 10. Standard: 3.
Farbe	1	sparkline.color	Hexadezimaler Farbcode (z. B. `FF0000`). Standard: `42A5F5`.
Zeitraum	1	sparkline.time_period._reference	`DASHBOARD._timeperiod` – den Zeitraumauswahlschalter des Dashboards als Datenquelle festlegen; `ABCDE._timeperiod` – ein kompatibles Widget (mit dem Parameter `reference`, der auf `ABCDE` gesetzt ist) als Datenquelle festlegen. Standard: "" (leer) Alternativ können Sie den Zeitraum nur in den Parametern *Von* und *Bis* festlegen.
Von	1	sparkline.time_period.from	Gültige Zeitangabe im absoluten (`YYYY-MM-DD hh:mm:ss`) oder relativen Zeitformat (`now`, `now/d`, `now/w-1w` usw.). Standard: now-1h. Parameterverhalten: - *unterstützt*, wenn *Zeitraum* nicht festgelegt ist - *erforderlich*, wenn `sparkline.time_period.to` festgelegt ist

Parameter	type	name	value
Bis	1	sparkline.time_period.to	Gültige Zeitangabe im absoluten (`YYYY-MM-DD hh:mm:ss`) oder relativen Zeitformat (`now`, `now/d`, `now/w-1w` usw.). Standard: now. Parameterverhalten: - *unterstützt*, wenn *Zeitraum* nicht festgelegt ist - *erforderlich*, wenn `time_period.to` festgelegt ist
Verlaufsdaten	0	sparkline.history	0 - *(Standard)* Automatisch; 1 - Verlauf; 2 - Trends.

Beispiele

Die folgenden Beispiele sollen ausschließlich die Konfiguration der Dashboard-Widget-Feldobjekte für das Widget *Datenpunktkarte* beschreiben. Weitere Informationen zur Konfiguration eines Dashboards finden Sie unter `dashboard.create`.

Konfiguration eines *Datenpunkt-Karten*-Widgets

Konfigurieren Sie ein *Datenpunkt-Karten*-Widget, das diese Abschnitte anzeigt: „Beschreibung“, „Neueste Daten“, „Auslöser“ und „Tags“.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Siehe auch

- Dashboard-Widget-Feld
- `dashboard.create`
- `dashboard.update`

16 Datenpunkt-Verlauf

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das Widget *Datenpunktverlauf* in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

Attention:
Die Eigenschaften von `fields` des Widgets werden während der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dies ermöglicht es Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets falsch zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Datenpunktverlauf* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Parameterverhalten.

Parameter

Die folgenden Parameter werden für das Widget *Datenpunktverlauf* unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	0	rf_rate	0 - Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - *(Standard)* 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - 15 Minuten.
Layout	0	layout	0 - *(Standard)* Horizontal; 1 - Vertikal.
Spalten (siehe unten)			
Zeilen anzeigen	0	show_lines	Mögliche Werte liegen zwischen 1 und 100. Standard: 25.
Host überschreiben	1	override_hostid._reference	`ABCDE._hostid` - ein kompatibles Widget (mit dem Parameter *Referenz*, der auf "`ABCDE`" gesetzt ist) als Datenquelle für Hosts festlegen; `DASHBOARD._hostid` - die Hostauswahl des Dashboards als Datenquelle für Hosts festlegen. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Erweiterte Konfiguration (siehe unten)			
Referenz	1	reference	Ein beliebiger Zeichenfolgenwert aus 5 Zeichen (z. B. `ABCDE` oder `JBPNL`). Dieser Wert muss innerhalb des Dashboards, zu dem das Widget gehört, eindeutig sein. Parameterverhalten: - *erforderlich*

Spalten

Spalten haben gemeinsame Parameter und zusätzliche Parameter, abhängig von der Konfiguration des Parameters *Datenpunkt*.

Note:
Bei allen Parametern, die sich auf Spalten beziehen, verweist die Zahl im Eigenschaftsnamen (z. B. columns.0.name) auf eine Spalte, für die der Parameter konfiguriert ist.

Die folgenden Parameter werden für alle Spalten unterstützt.

Parameter	type	name	value
Name	1	columns.0.name	Beliebiger Zeichenfolgenwert. Parameterverhalten: - *erforderlich*
Datenpunkt	4	columns.0.itemid	ID des Datenpunkts. Beim Konfigurieren des Widgets in einem Vorlagen-Dashboard sollten nur Datenpunkte festgelegt werden, die in der Vorlage konfiguriert sind. Parameterverhalten: - *erforderlich*
Basisfarbe	1	columns.0.base_color	Hexadezimaler Farbcode (z. B. `FF0000`). Standard: "" (leer).

Die folgenden Spaltenparameter werden unterstützt, wenn der konfigurierte *Datenpunkt* ein Datenpunkt vom numerischen Typ ist.

Parameter	type	name	value
Anzeige	0	columns.0.display	1 - *(Standard)* Wie vorhanden; 2 - Balken; 3 - Indikatoren.
Min	1	columns.0.min	Beliebiger numerischer Wert. Parameterverhalten: - *unterstützt*, wenn *Anzeige* auf „Balken“ oder „Indikatoren“ gesetzt ist
Max	1	columns.0.max	Beliebiger numerischer Wert. Parameterverhalten: - *unterstützt*, wenn *Anzeige* auf „Balken“ oder „Indikatoren“ gesetzt ist
Schwellenwerte			
Farbe	1	columns.0.thresholds.color	Hexadezimaler Farbcode (z. B. FF0000).
Schwellenwert	1	columns.0.thresholds.threshold	Beliebiger numerischer Wert. Suffixe (z. B. „1d“, „2w“, „4K“, „8G“) werden unterstützt.
Verlaufsdaten	0	columns.0.history	0 - *(Standard)* Automatisch; 1 - Verlauf; 2 - Trends.

Die folgenden Spaltenparameter werden unterstützt, wenn der konfigurierte *Datenpunkt* vom Typ Zeichen, Text oder Log ist.

Parameter	type	name	value
Hervorhebungen			
Hervorhebung	1	columns.0.highlights.color	Hexadezimaler Farbcode (z. B. FF0000).
Schwellenwert	1	columns.0.highlights.pattern	Beliebiger regulärer Ausdruck.
Anzeige	0	columns.0.display	1 - *(Standard)* Wie vorhanden; 4 - HTML; 5 - Einzeilig.
Einzeilig	0	columns.0.max_length	Mögliche Werte liegen zwischen 1 und 500. Standard: 100. Parameterverhalten: - *unterstützt*, wenn *Anzeige* auf „Einzeilig“ gesetzt ist
Monospace-Schriftart verwenden	0	columns.0.monospace_font	0 - *(Standard)* Standardschriftart verwenden; 1 - Monospace-Schriftart verwenden.
Lokale Zeit anzeigen	0	columns.0.local_time	0 - *(Standard)* Zeitstempel anzeigen; 1 - Lokale Zeit anzeigen. Parameterverhalten: - *unterstützt*, wenn *Datenpunkt* auf einen Datenpunkt vom Typ Log gesetzt ist und *Zeitstempel anzeigen* auf „Aktiviert“ gesetzt ist

Die folgenden Spaltenparameter werden unterstützt, wenn der konfigurierte *Datenpunkt* ein Datenpunkt vom Binärtyp ist.

Parameter	type	name	value
Miniaturansicht anzeigen	1	columns.0.show_thumbnail	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.

Erweiterte Konfiguration

Die folgenden erweiterten Konfigurationsparameter werden für das Widget *Datenpunkthistorie* unterstützt.

Parameter	type	name	value
Neue Werte	0	sortorder	0 - *(Standard)* Oben; 1 - Unten.
Zeitstempel anzeigen	0	show_timestamp	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
Spaltenüberschrift anzeigen	0	show_column_header	0 - Aus; 1 - Horizontal; 2 - *(Standard)* Vertikal.
Zeitraum	1	time_period._referenc	`DASHBOARD._timeperiod` - den Zeitraumauswahl des Dashboards als Datenquelle festlegen; `ABCDE._timeperiod` - ein kompatibles Widget (mit dem auf "`ABCDE`" gesetzten Parameter *Referenz*) als Datenquelle festlegen. Standard: `DASHBOARD._timeperiod` Alternativ können Sie den Zeitraum nur in den Parametern *Von* und *Bis* festlegen.
Von	1	time_period.from	Gültige Zeitzeichenfolge in absoluter (`YYYY-MM-DD hh:mm:ss`) oder relativer Zeitsyntax (`now`, `now/d`, `now/w-1w` usw.). Parameterverhalten: - *unterstützt*, wenn *Zeitraum* nicht gesetzt ist - *erforderlich*, wenn `time_period.to` gesetzt ist
Bis	1	time_period.to	Gültige Zeitzeichenfolge in absoluter (`YYYY-MM-DD hh:mm:ss`) oder relativer Zeitsyntax (`now`, `now/d`, `now/w-1w` usw.). Parameterverhalten: - *unterstützt*, wenn *Zeitraum* nicht gesetzt ist - *erforderlich*, wenn `time_period.from` gesetzt ist

Beispiele

Die folgenden Beispiele beschreiben nur die Konfiguration der Dashboard-Widget-Feldobjekte für das *Elementverlauf* Widget. Für weitere Informationen zur Konfiguration eines Dashboards siehe `dashboard.create`.

Konfiguration eines Widgets *Datenpunkthistorie*

Konfigurieren Sie ein Widget *Datenpunkthistorie*, das die neuesten Daten für zwei numerische Datenpunkte „42269“ und „42270“ anzeigt. Konfigurieren Sie außerdem die Datenpunktspalten so, dass sie vertikal angezeigt werden, wobei die Spaltennamen horizontal dargestellt werden; begrenzen Sie die Anzeige auf 15 Datenzeilen und fügen Sie eine separate Zeitstempelspalte hinzu.

Anfrage:

```
{
    "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
```

```
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Siehe auch

- Dashboard Widget Feld
- `dashboard.create`
- `dashboard.update`

17 Datenpunkt-Navigator

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das Widget *Datenpunkt-Navigator* in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

Attention:
Die Eigenschaften von `fields` des Widgets werden während der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dies ermöglicht es Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets falsch zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Datenpunkt-Navigator* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Parameterverhalten.

Parameter

Die folgenden Parameter werden für das Widget *Datenpunkt-Navigator* unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	0	rf_rate	0 - Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - *(Standard)* 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - 15 Minuten.
Host-Gruppen	2	groupids.0	ID der Host-Gruppe. Hinweis: Um mehrere Host-Gruppen zu konfigurieren, erstellen Sie für jede Host-Gruppe ein Dashboard-Widget-Feldobjekt mit einer fortlaufenden Nummer im Eigenschaftsnamen. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Host-Gruppen (Widget)	1	groupids._reference	Anstelle der ID der Host-Gruppe: `ABCDE._hostgroupids` - legen Sie ein kompatibles Widget (mit dem Parameter *Referenz* auf "`ABCDE`" gesetzt) als Datenquelle für Host-Gruppen fest. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.

Parameter	type	name	value
Hosts	3	hostids.0	ID des Hosts. Hinweis: Um mehrere Hosts zu konfigurieren, erstellen Sie für jeden Host ein Dashboard-Widget-Feldobjekt mit einer fortlaufenden Nummer im Eigenschaftsnamen. Bei mehreren Hosts muss der Parameter *Host-Gruppen* entweder überhaupt nicht konfiguriert sein oder mindestens eine Host-Gruppe enthalten, der die konfigurierten Hosts angehören. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Hosts (Widget/Dashboard)	1	hostids._reference	Anstelle der ID des Hosts: `DASHBOARD.hostid` - legen Sie den Host-Auswahlschalter des Dashboards als Datenquelle für Hosts fest; `ABCDE._hostid` - legen Sie ein kompatibles Widget (mit dem Parameter *Referenz* auf "`ABCDE`" gesetzt) als Datenquelle für Hosts fest. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Host-Tags			
Auswertungstyp	0	host_tags_evaltype	0 - *(Standard)* Und/Oder; 2 - Oder. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Tag-Name	1	host_tags.0.tag	Beliebiger Zeichenfolgenwert. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Reihenfolge des Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich*, wenn *Host-Tags* konfiguriert werden Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Operator	0	host_tags.0.operator	0 - Enthält; 1 - Ist gleich; 2 - Enthält nicht; 3 - Ist nicht gleich; 4 - Ist vorhanden; 5 - Ist nicht vorhanden. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Reihenfolge des Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich*, wenn *Host-Tags* konfiguriert werden Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Tag-Wert	1	host_tags.0.value	Beliebiger Zeichenfolgenwert. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Reihenfolge des Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich*, wenn *Host-Tags* konfiguriert werden Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.

Parameter	type	name	value
Datenpunktmuster	1	items.0	Name oder Muster des Datenpunkts. Hinweis: Um mehrere Datenpunktmuster zu konfigurieren, erstellen Sie für jedes Datenpunktmuster ein Dashboard-Widget-Feldobjekt mit einer fortlaufenden Nummer im Eigenschaftsnamen.
Datenpunkt-Tags			
Auswertungstyp	0	item_tags_evaltype	0 - *(Standard)* Und/Oder; 2 - Oder.
Tag-Name	1	item_tags.0.tag	Beliebiger Zeichenfolgenwert. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Reihenfolge des Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich*, wenn *Datenpunkt-Tags* konfiguriert werden
Operator	0	item_tags.0.operator	0 - Enthält; 1 - Ist gleich; 2 - Enthält nicht; 3 - Ist nicht gleich; 4 - Ist vorhanden; 5 - Ist nicht vorhanden. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Reihenfolge des Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich*, wenn *Datenpunkt-Tags* konfiguriert werden
Tag-Wert	1	item_tags.0.value	Beliebiger Zeichenfolgenwert. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Reihenfolge des Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich*, wenn *Datenpunkt-Tags* konfiguriert werden
Status	0	state	-1 - *(Standard)* Alle; 0 - Normal; 1 - Nicht unterstützt.
Probleme anzeigen	0	show_problems	0 - Alle; 1 - *(Standard)* Nicht unterdrückte; 2 - Keine.
Gruppieren nach			
Attribut	0	group_by.0.attribute	0 - Host-Gruppe; 1 - Host-Name; 2 - Host-Tag-Wert; 3 - Datenpunkt-Tag-Wert. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Reihenfolge des Attributs in der Liste der Gruppierungsattribute. Parameterverhalten: - *erforderlich*, wenn *Gruppieren nach* konfiguriert wird
Wert	1	group_by.0.tag_name	Beliebiger Zeichenfolgenwert. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf das im Parameter *Attribut* festgelegte Gruppierungsattribut. Parameterverhalten: - *erforderlich*, wenn *Gruppieren nach* konfiguriert wird und *Attribut* auf „Host-Tag-Wert“ oder „Datenpunkt-Tag-Wert“ gesetzt ist

Parameter	type	name	value
Datenpunktlimit	0	show_lines	Mögliche Werte liegen zwischen 1 und 9999. Standard: 100.
Referenz	1	reference	Beliebiger Zeichenfolgenwert aus 5 Zeichen (z. B. `ABCDE` oder `JBPNL`). Dieser Wert muss innerhalb des Dashboards, zu dem das Widget gehört, eindeutig sein. Parameterverhalten: - *erforderlich*

Beispiele

Die folgenden Beispiele beschreiben nur die Konfiguration der Dashboard-Widget-Feldobjekte für das *Item navigator* Widget. Für weitere Informationen zur Konfiguration eines Dashboards siehe `dashboard.create`.

Konfiguration eines *Datenpunkt-Navigator*-Widgets

Konfigurieren Sie ein *Datenpunkt-Navigator*-Widget, das bis zu 1000 Datenpunkte anzeigt, gruppiert nach ihrem Host und anschließend nach dem Wert des Datenpunkt-Tags „component“.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Siehe auch

- Dashboard Widget Feld
- `dashboard.create`
- `dashboard.update`

18 Datenpunkt-Wert

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das Widget *Datenpunktwert* in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

Attention:
Die Eigenschaften von `fields` des Widgets werden beim Erstellen oder Aktualisieren eines Dashboards nicht validiert. Dies ermöglicht es Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets falsch zu erstellen oder zu aktualisieren. Um sicherzustellen, dass das Widget *Datenpunktwert* erfolgreich erstellt oder aktualisiert wird, beachten Sie bitte das in den folgenden Tabellen beschriebene Parameterverhalten.

Parameter

Die folgenden Parameter werden für das Widget *Datenpunktwert* unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	0	rf_rate	0 - Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - *(Standard)* 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - 15 Minuten.
Datenpunkt	4	itemid.0	Datenpunkt-ID. Parameterverhalten: - *erforderlich*, wenn *Datenpunkt (Widget)* nicht gesetzt ist
Datenpunkt (Widget)	1	itemid._reference	Anstelle der Datenpunkt-ID: `ABCDE._itemid` - legt ein kompatibles Widget (mit dem auf "`ABCDE`" gesetzten Parameter *Referenz*) als Datenquelle für Datenpunkte fest. Parameterverhalten: - *erforderlich*, wenn *Datenpunkt* nicht gesetzt ist
Anzeigen	0	show.0	1 - Beschreibung; 2 - Wert; 3 - Zeit; 4 - Änderungsindikator; 5 - Sparkline. Standard: 1, 2, 3, 4. Hinweis: Um mehrere Werte zu konfigurieren, erstellen Sie für jeden Wert ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Nummer im Eigenschaftsnamen.
Host überschreiben	1	override_hostid._reference	`ABCDE._hostid` - legt ein kompatibles Widget (mit dem auf "`ABCDE`" gesetzten Parameter *Referenz*) als Datenquelle für Hosts fest; `DASHBOARD._hostid` - legt die Host-Auswahl des Dashboards als Datenquelle für Hosts fest. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.

Erweiterte Konfiguration

Die folgenden erweiterten Konfigurationsparameter werden für das Widget *Datenpunktwert* unterstützt.

Note:
Die Zahl im Eigenschaftsnamen *Thresholds* (z. B. thresholds.0.color) verweist auf die Position des Schwellenwerts in einer Liste, die in aufsteigender Reihenfolge sortiert ist. Wenn Schwellenwerte jedoch in einer anderen Reihenfolge konfiguriert werden, werden die Werte nach der Aktualisierung der Widget-Konfiguration im Zabbix Frontend in aufsteigender Reihenfolge sortiert (z. B. `"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"`; `"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`).

Parameter	type	name	value
Hintergrundfarbe	1	bg_color	Hexadezimaler Farbcode (z. B. `FF0000`). Standard: `""` (leer).
Thresholds			
Farbe	1	thresholds.0.color	Hexadezimaler Farbcode (z. B. `FF0000`).
Schwellenwert	1	thresholds.0.threshold	Beliebiger Zeichenfolgenwert.

Parameter	type	name	value
Aggregationsfunktion	0	aggregate_function	0 - *(Standard)* nicht verwendet; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last.
Zeitraum	1	time_period._referenc	`DASHBOARD._timeperiod` - den Zeitraumauswähler des Dashboards als Datenquelle festlegen; `ABCDE._timeperiod` - ein kompatibles Widget (mit dem Parameter `reference` gleich `ABCDE`) als Datenquelle festlegen. Standard: `DASHBOARD._timeperiod` Alternativ können Sie den Zeitraum nur in den Parametern *Von* und *Bis* festlegen. Parameterverhalten: - *unterstützt*, wenn *Aggregationsfunktion* auf ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first”, ”last” gesetzt ist
Von	1	time_period.from	Gültige Zeitzeichenfolge in absoluter (`YYYY-MM-DD hh:mm:ss`) oder relativer Zeitsyntax (`now`, `now/d`, `now/w-1w` usw.). Parameterverhalten: - *unterstützt*, wenn *Zeitraum* nicht gesetzt ist und *Aggregationsfunktion* auf ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first”, ”last” gesetzt ist - *erforderlich*, wenn `time_period.to` gesetzt ist
Bis	1	time_period.to	Gültige Zeitzeichenfolge in absoluter (`YYYY-MM-DD hh:mm:ss`) oder relativer Zeitsyntax (`now`, `now/d`, `now/w-1w` usw.). Parameterverhalten: - *unterstützt*, wenn *Zeitraum* nicht gesetzt ist und *Aggregationsfunktion* auf ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first”, ”last” gesetzt ist - *erforderlich*, wenn `time_period.from` gesetzt ist
Verlaufsdaten	0	history	0 - *(Standard)* Auto; 1 - Verlauf; 2 - Trends.

Beschreibung

Die folgenden erweiterten Konfigurationsparameter werden unterstützt, wenn *Anzeigen* auf „Beschreibung“ gesetzt ist.

Parameter	type	name	value
Beschreibung	1	description	Beliebiger Zeichenfolgenwert, einschließlich Makros. Unterstützte Makros: `{HOST.*}`, `{ITEM.*}`, `{INVENTORY.*}`, Benutzermakros. Standard: `{ITEM.NAME}`.
Horizontale Position	0	desc_h_pos	0 - Links; 1 - *(Standard)* Zentriert; 2 - Rechts. Zwei oder mehr Elemente (Beschreibung, Wert, Zeit) dürfen nicht dieselbe *Horizontale Position* und *Vertikale Position* verwenden.

Parameter	type	name	value
Vertikale Position	0	desc_v_pos	0 - Oben; 1 - Mitte; 2 - *(Standard)* Unten. Zwei oder mehr Elemente (Beschreibung, Wert, Zeit) dürfen nicht dieselbe *Horizontale Position* und *Vertikale Position* verwenden.
Größe	0	desc_size	Mögliche Werte liegen zwischen 1 und 100. Standard: 15.
Fett	0	desc_bold	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
Farbe	1	desc_color	Hexadezimaler Farbcode (z. B. FF0000). Standard: "" (leer).

Wert

Die folgenden erweiterten Konfigurationsparameter werden unterstützt, wenn *Anzeigen* auf „Wert“ gesetzt ist.

Parameter	type	name	value
Dezimalstellen			
Dezimalstellen	0	decimal_places	Mögliche Werte liegen zwischen 1 und 10. Standard: 2.
Größe	0	decimal_size	Mögliche Werte liegen zwischen 1 und 100. Standard: 35.
Position			
Horizontale Position	0	value_h_pos	0 - Links; 1 - *(Standard)* Mitte; 2 - Rechts. Zwei oder mehr Elemente (Beschreibung, Wert, Zeit) dürfen nicht dieselbe *Horizontale Position* und *Vertikale Position* verwenden.
Vertikale Position	0	value_v_pos	0 - Oben; 1 - *(Standard)* Mitte; 2 - Unten. Zwei oder mehr Elemente (Beschreibung, Wert, Zeit) dürfen nicht dieselbe *Horizontale Position* und *Vertikale Position* verwenden.
Größe	0	value_size	Mögliche Werte liegen zwischen 1 und 100. Standard: 45.
Fett	0	value_bold	0 - Deaktiviert; 1 - *(Standard)* Aktiviert.
Farbe	1	value_color	Hexadezimaler Farbcode (z. B. FF0000). Standard: "" (leer).
Einheiten			
Einheiten (Kontrollkästchen)	0	units_show	0 - Deaktiviert; 1 - *(Standard)* Aktiviert.
Einheiten (Wert)	1	units	Beliebiger Zeichenfolgenwert.
Position	0	units_pos	0 - Vor dem Wert; 1 - Über dem Wert; 2 - *(Standard)* Nach dem Wert; 3 - Unter dem Wert.
Größe	0	units_size	Mögliche Werte liegen zwischen 1 und 100. Standard: 35.

Parameter	type	name	value
Fett	0	units_bold	0 - Deaktiviert; 1 - *(Standard)* Aktiviert.
Farbe	1	units_color	Hexadezimaler Farbcode (z. B. FF0000). Standard: "" (leer).

Zeit

Die folgenden erweiterten Konfigurationsparameter werden unterstützt, wenn *Anzeigen* auf „Zeit“ gesetzt ist.

Parameter	type	name	value
Horizontale Position	0	time_h_pos	0 - Links; 1 - *(Standard)* Zentriert; 2 - Rechts. Zwei oder mehr Elemente (Beschreibung, Wert, Zeit) dürfen nicht dieselbe *Horizontale Position* und *Vertikale Position* verwenden.
Vertikale Position	0	time_v_pos	0 - *(Standard)* Oben; 1 - Mitte; 2 - Unten. Zwei oder mehr Elemente (Beschreibung, Wert, Zeit) dürfen nicht dieselbe *Horizontale Position* und *Vertikale Position* verwenden.
Größe	0	time_size	Mögliche Werte liegen zwischen 1 und 100. Standard: 15.
Fett	0	time_bold	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
Farbe	1	time_color	Hexadezimaler Farbcode (z. B. FF0000). Standard: "" (leer).

Änderungsindikator

Die folgenden erweiterten Konfigurationsparameter werden unterstützt, wenn *Anzeigen* auf „Änderungsindikator“ gesetzt ist.

Parameter	type	name	value
Farbe des Änderungsindikators ↑	1	up_color	Hexadezimaler Farbcode (z. B. FF0000). Standard: "" (leer).
Farbe des Änderungsindikators ↓	1	down_color	Hexadezimaler Farbcode (z. B. FF0000). Standard: "" (leer).
Farbe des Änderungsindikators ↕	1	updown_color	Hexadezimaler Farbcode (z. B. FF0000). Standard: "" (leer).

Sparkline

Die folgenden erweiterten Konfigurationsparameter werden unterstützt, wenn *Anzeigen* auf „Sparkline“ gesetzt ist.

Parameter	type	name	value
Breite	0	sparkline.width	Mögliche Werte liegen zwischen 0 und 10. Standard: 1.
Füllung	0	sparkline.fill	Mögliche Werte liegen zwischen 0 und 10. Standard: 3.

Parameter	type	name	value
Farbe	1	sparkline.color	Hexadezimaler Farbcode (z. B. FF0000). Standard: 42A5F5.
Zeitraum	1	sparkline.time_period.data_source	DASHBOARD._timeperiod – den Zeitraumauswahlschalter des Dashboards als Datenquelle festlegen; ABCDE._timeperiod – ein kompatibles Widget (mit dem Parameter reference, der auf ABCDE gesetzt ist) als Datenquelle festlegen. Standard: DASHBOARD._timeperiod
Von	1	sparkline.time_period.from	Alternativ können Sie den Zeitraum nur in den Parametern *Von* und *Bis* festlegen. Gültige Zeitangabe im absoluten (YYYY-MM-DD hh:mm:ss) oder relativen Zeitformat (now, now/d, now/w-1w usw.). Parameterverhalten: - *erforderlich*, wenn sparkline.time_period.to festgelegt ist
Bis	1	sparkline.time_period.to	Gültige Zeitangabe im absoluten (YYYY-MM-DD hh:mm:ss) oder relativen Zeitformat (now, now/d, now/w-1w usw.). Parameterverhalten: - *erforderlich*, wenn sparkline.time_period.from festgelegt ist
Verlaufsdaten	0	sparkline.history	0 – *(Standard)* Automatisch; 1 – Verlauf; 2 – Trends.

Beispiele

Die folgenden Beispiele beschreiben nur die Konfiguration der Dashboard-Widget-Feldobjekte für das *Einzelwert* Widget. Für weitere Informationen zur Konfiguration eines Dashboards siehe dashboard.create.

Konfigurieren eines Widgets *Datenpunktwert*

Konfigurieren Sie ein Widget *Datenpunktwert*, das den Datenpunktwert für den Datenpunkt „42266“ (Verfügbarkeit des Zabbix-Agenten) anzeigt. Zusätzlich können Sie das Widget mit mehreren erweiterten Optionen visuell feinabstimmen, einschließlich einer dynamischen Hintergrundfarbe, die sich abhängig vom Verfügbarkeitsstatus des Zabbix-Agenten ändert.

Anfrage:

```
{
    "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
```

```
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Siehe auch

- Dashboard Widget Feld
- `dashboard.create`
- `dashboard.update`

19 Karte

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das Widget *Karte* in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

Attention:
Die Eigenschaften von `fields` des Widgets werden während der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dies ermöglicht es Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets falsch zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Karte* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Parameterverhalten.

Parameter

Die folgenden Parameter werden für das Widget *Karte* unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	0	rf_rate	0 - Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - *(Standard)* 15 Minuten.
Karte	8	sysmapid.0	ID der Karte. Parameterverhalten: - *erforderlich*, wenn *Karte (Widget)* nicht gesetzt ist
Karte (Widget)	1	sysmapid._reference	`ABCDE._mapid` - legt ein Widget *Kartennavigationsbaum* (mit dem Parameter *Referenz* auf "`ABCDE`" gesetzt) als Datenquelle für Karten fest. Parameterverhalten: - *erforderlich*, wenn *Karte* nicht gesetzt ist
Referenz	1	reference	Beliebiger Zeichenfolgenwert, der aus 5 Zeichen besteht (z. B. `ABCDE` oder `JBPNL`). Dieser Wert muss innerhalb des Dashboards, zu dem das Widget gehört, eindeutig sein. Parameterverhalten: - *erforderlich*

Beispiele

Die folgenden Beispiele beschreiben nur die Konfiguration der Dashboard-Widget-Feldobjekte für das *Map* -Widget. Für weitere Informationen zur Konfiguration eines Dashboards siehe `dashboard.create`.

Konfigurieren eines *Karten*-Widgets

Konfigurieren Sie ein *Karten*-Widget, das die Karte „1“ anzeigt.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
```

```
}
```

Konfigurieren eines verknüpften *Karten*-Widgets

Konfigurieren Sie ein *Karten*-Widget, das mit einem Widget vom Typ *Karten-Navigationsbaum* verknüpft ist.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

See also

- Dashboard Widget Feld
- `dashboard.create`
- `dashboard.update`
- Karten-Navigationsbaum

20 Navigationsbaum der Karte

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das Widget *Navigationsbaum der Karten* in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

Attention:
Die Eigenschaften von `fields` des Widgets werden beim Erstellen oder Aktualisieren eines Dashboards nicht validiert. Dies ermöglicht es Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets falsch zu erstellen oder zu aktualisieren. Um sicherzustellen, dass das Widget *Navigationsbaum der Karten* erfolgreich erstellt oder aktualisiert wird, beachten Sie bitte das in den folgenden Tabellen beschriebene Parameterverhalten.

Parameter

Die folgenden Parameter werden für das Widget *Map-Navigationsbaum* unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	0	rf_rate	0 - Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - *(Standard)* 15 Minuten.
Nicht verfügbare Karten anzeigen	1	show_unavailable	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
Referenz	1	reference	Beliebiger Zeichenfolgenwert, der aus 5 Zeichen besteht (z. B. `ABCDE` oder `JBPNL`). Dieser Wert muss innerhalb des Dashboards, zu dem das Widget gehört, eindeutig sein. Parameter behavior: - *erforderlich*

Die folgenden Parameter werden für die Konfiguration von Elementen des Map-Navigationsbaums unterstützt.

Parameter	type	name	value
Name	1	navtree.1.name	Beliebiger Zeichenfolgenwert. Hinweis: Die Zahl im Eigenschaftsnamen legt die Elementnummer fest.
Verknüpfte Karte	8	navtree.1.sysmapid	ID der Karte. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf das Element, mit dem die Karte verknüpft ist.
Parameter zum Erstellen der Elementhierarchie	0	navtree.1.parent	Nummer des übergeordneten Elements. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf das untergeordnete Element. Der Eigenschaftswert verweist auf das übergeordnete Element.
	0	navtree.1.order	Elementposition im Map-Navigationsbaum. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Elementnummer. Der Eigenschaftswert verweist auf die Elementposition im Map-Navigationsbaum. Die Position des übergeordneten Elements wird innerhalb des gesamten Map-Navigationsbaums bestimmt. Die Position des untergeordneten Elements wird innerhalb des übergeordneten Elements bestimmt.

Beispiele

Die folgenden Beispiele beschreiben nur die Konfiguration der Feldobjekte des Dashboard-Widgets für das *Kartennavigationsbaum* Widget. Für weitere Informationen zur Konfiguration eines Dashboards siehe `dashboard.create`.

Konfiguration eines Widgets *Map navigation tree*

Konfigurieren Sie ein Widget *Map navigation tree*, das den folgenden Karten-Navigationsbaum anzeigt:

- Element A

    - Element A1
    - Element A2
- Element B
    - Element B1 (enthält die verknüpfte Karte „1“, die in einem verknüpften Widget *Map widget* angezeigt werden kann)
    - Element B2
- Element C

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Siehe auch

- Dashboard Widget Feld
- `dashboard.create`
- `dashboard.update`
- Karte

21 Kreisdiagramm

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das Widget *Kreisdiagramm* in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

Attention:
Die Eigenschaften von `fields` des Widgets werden beim Erstellen oder Aktualisieren eines Dashboards nicht validiert. Dies ermöglicht es Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets falsch zu erstellen oder zu aktualisieren. Um sicherzustellen, dass das Widget *Kreisdiagramm* erfolgreich erstellt oder aktualisiert wird, beachten Sie bitte das in den folgenden Tabellen beschriebene Parameterverhalten.

Parameter

Die folgenden Parameter werden für das Widget *Kreisdiagramm* unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	0	rf_rate	0 - Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - *(Standard)* 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - 15 Minuten.

Datensatz

Die folgenden Parameter werden für die Konfiguration eines *Datensatzes* unterstützt.

Note:
Die erste Zahl im Eigenschaftsnamen (z. B. ds.0.hosts.0, ds.0.items.0) stellt den jeweiligen Datensatz dar, während die zweite Zahl, sofern vorhanden, den konfigurierten Host oder Datenpunkt darstellt.

Parameter	type	name	value
Datensatztyp	0	ds.0.dataset_type	0 - Datenpunktliste; 1 - *(Standard)* Datenpunktmuster.
Datenpunkte	4	ds.0.itemids.0	ID des Datenpunkts. Bei der Konfiguration des Widgets auf einem Vorlagen-Dashboard dürfen nur Datenpunkte festgelegt werden, die auf der Vorlage konfiguriert sind. Hinweis: Um mehrere Datenpunkte zu konfigurieren, erstellen Sie für jeden Datenpunkt ein Dashboard-Widget-Feldobjekt mit einer erhöhten Zahl im Eigenschaftsnamen. Parameterverhalten: - *erforderlich*, wenn *Datensatztyp* auf „Datenpunktliste“ gesetzt ist und *Datenpunkte (Widget)* nicht festgelegt ist
Datenpunkte (Widget)	1	ds.0.itemids.0._reference	Anstelle der ID des Datenpunkts: `ABCDE._itemid` - Legen Sie ein kompatibles Widget (mit dem Parameter *Referenz* auf „ABCDE“ gesetzt) als Datenquelle für Datenpunkte fest. Hinweis: Um mehrere Widgets zu konfigurieren, erstellen Sie für jedes Widget ein Dashboard-Widget-Feldobjekt mit einer erhöhten Zahl im Eigenschaftsnamen. Parameterverhalten: - *erforderlich*, wenn *Datensatztyp* auf „Datenpunktliste“ gesetzt ist und *Datenpunkte* nicht festgelegt ist
Farbe	1	ds.0.color.0	Hexadezimaler Farbcode (z. B. `FF0000`). Parameterverhalten: - *erforderlich*, wenn *Datensatztyp* auf „Datenpunktliste“ gesetzt ist
Datenpunkttyp	0	ds.0.type.0	0 - *(Standard)* Normal; 1 - Gesamt. Der Wert „Gesamt“ kann nur für einen Datenpunkt im gesamten Diagramm festgelegt werden. Parameterverhalten: - *unterstützt*, wenn *Datensatztyp* auf „Datenpunktliste“ gesetzt ist

Parameter	type	name	value
Hostmuster	1	ds.0.hosts.0	Name oder Muster des Hosts (z. B. „Zabbix*“). Parameterverhalten: - *erforderlich*, wenn *Datensatztyp* auf „Datenpunktmuster“ gesetzt ist Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Datenpunktmuster	1	ds.0.items.0	Name oder Muster des Datenpunkts (z. B. „*: Anzahl der verarbeiteten *Werte pro Sekunde“). Bei der Konfiguration des Widgets auf einem Vorlagen-Dashboard dürfen nur die Muster für Datenpunkte festgelegt werden, die auf der Vorlage konfiguriert sind. Parameterverhalten: - *erforderlich*, wenn *Datensatztyp* auf „Datenpunktmuster“ gesetzt ist
Farbe	1	ds.0.color	Hexadezimaler Farbcode (z. B. FF0000). Parameterverhalten: - *unterstützt*, wenn *Datensatztyp* auf „Datenpunktmuster“ gesetzt ist und *Farbpalette* nicht festgelegt ist
Farbpalette	0	ds.0.color_palette	Index der Farbpalette. Mögliche Werte liegen zwischen 0 und 11. Standard: 0. Parameterverhalten: - *unterstützt*, wenn *Datensatztyp* auf „Datenpunktmuster“ gesetzt ist und *Farbe* nicht festgelegt ist
Aggregationsfunktion	0	ds.0.aggregate_function	1 - Minimum; 2 - Maximum; 3 - Durchschnitt; 4 - Anzahl; 5 - Summe; 6 - erster Wert; 7 - *(Standard)* letzter Wert.
Datensatzaggregation	0	ds.0.dataset_aggregation	0 - *(Standard)* keine; 1 - Minimum; 2 - Maximum; 3 - Durchschnitt; 4 - Anzahl; 5 - Summe. Parameterverhalten: - *unterstützt*, wenn *Datenpunkttyp* auf „Gesamt“ gesetzt ist
Datensatzbezeichnung	1	ds.0.data_set_label	Beliebiger Zeichenfolgenwert. Standard: "" (leer).
Datenpunkt-Tags			
Auswertungstyp	0	ds.0.item_tags_evaltype	0 - *(Standard)* Und/Oder; 2 - Oder.
Tag-Name	1	ds.0.item_tags.0.tag	Beliebiger Zeichenfolgenwert. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Reihenfolge des Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei der Konfiguration von *Datenpunkt-Tags*

Parameter	type	name	value
Operator	0	ds.0.item_tags.0.operator	0 - Enthält; 1 - Ist gleich; 2 - Enthält nicht; 3 - Ist nicht gleich; 4 - Existiert; 5 - Existiert nicht. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Reihenfolge des Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei der Konfiguration von *Datenpunkt-Tags*
Tag-Wert	1	ds.0.item_tags.0.value	Beliebiger Zeichenfolgenwert. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Reihenfolge des Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei der Konfiguration von *Datenpunkt-Tags*

Anzeigeoptionen

Die folgenden Parameter werden für die Konfiguration der *Anzeigeoptionen* unterstützt.

Parameter	type	name	value
Auswahl der Verlaufsdaten	0	source	0 - *(Standard)* Automatisch; 1 - Verlauf; 2 - Trends.
Darstellung	0	draw_type	0 - *(Standard)* Kreisdiagramm; 1 - Ringdiagramm.
Breite	0	width	20 - 20 % des Radius; 30 - 30 % des Radius; 40 - 40 % des Radius; 50 - *(Standard)* 50 % des Radius. Parameterverhalten: - *unterstützt*, wenn *Darstellung* auf „Ringdiagramm“ gesetzt ist
Strichbreite	0	stroke	Mögliche Werte liegen zwischen 0 und 10. Standard: 0. Parameterverhalten: - *unterstützt*, wenn *Darstellung* auf „Ringdiagramm“ gesetzt ist
Gesamtwert anzeigen	0	total_show	0 - *(Standard)* Deaktiviert; 1 - Aktiviert. Parameterverhalten: - *unterstützt*, wenn *Darstellung* auf „Ringdiagramm“ gesetzt ist
Größe	0	value_size_type	0 - *(Standard)* Automatisch; 1 - Benutzerdefiniert. Parameterverhalten: - *unterstützt*, wenn *Gesamtwert anzeigen* auf „Aktiviert“ gesetzt ist

Parameter	type	name	value
Größe (Wert für benutzerdefinierte Größe)	0	value_size	Mögliche Werte liegen zwischen 1 und 100. Standard: 20. Parameterverhalten: - *unterstützt*, wenn *Gesamtwert anzeigen* auf „Aktiviert“ gesetzt ist
Dezimalstellen	0	decimal_places	Mögliche Werte liegen zwischen 0 und 6. Standard: 2. Parameterverhalten: - *unterstützt*, wenn *Gesamtwert anzeigen* auf „Aktiviert“ gesetzt ist
Einheiten (Kontrollkästchen)	0	units_show	0 - *(Standard)* Deaktiviert; 1 - Aktiviert. Parameterverhalten: - *unterstützt*, wenn *Gesamtwert anzeigen* auf „Aktiviert“ gesetzt ist
Einheiten (Wert)	1	units	Beliebiger Zeichenfolgenwert. Parameterverhalten: - *unterstützt*, wenn *Einheiten* (Kontrollkästchen) auf „Aktiviert“ gesetzt ist
Fett	0	value_bold	0 - *(Standard)* Deaktiviert; 1 - Aktiviert. Parameterverhalten: - *unterstützt*, wenn *Gesamtwert anzeigen* auf „Aktiviert“ gesetzt ist
Farbe	1	value_color	Hexadezimaler Farbcode (z. B. FF0000). Parameterverhalten: - *unterstützt*, wenn *Gesamtwert anzeigen* auf „Aktiviert“ gesetzt ist
Abstand zwischen Sektoren	0	space	Mögliche Werte liegen zwischen 0 und 10. Standard: 1.
Sektoren zusammenführen, die kleiner als N % sind (Kontrollkästchen)	0	merge	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
Sektoren zusammenführen, die kleiner als N % sind (Wert)	0	merge_percent	Mögliche Werte liegen zwischen 1 und 10. Standard: 1. Parameterverhalten: - *unterstützt*, wenn *Sektoren zusammenführen, die kleiner als N % sind* (Kontrollkästchen) auf „Aktiviert“ gesetzt ist

Parameter	type	name	value
Sektoren zusammenführen, die kleiner als N % sind (Farbe)	1	merge_color	Hexadezimaler Farbcode (z. B. FF0000). Parameterverhalten: - *unterstützt*, wenn *Sektoren zusammenführen, die kleiner als N % sind* (Kontrollkästchen) auf „Aktiviert“ gesetzt ist

Zeitperiode

Die folgenden Parameter werden für die Konfiguration von *Zeitperiode* unterstützt.

Parameter	type	name	value
Zeitperiode	1	time_period._referende	`DASHBOARD._timeperiod` - legt den Zeitperiodenwähler des Dashboards als Datenquelle fest; `ABCDE._timeperiod` - legt ein kompatibles Widget (mit dem Parameter *Reference* auf "`ABCDE`" gesetzt) als Datenquelle fest. Standard: `DASHBOARD._timeperiod` Alternativ können Sie die Zeitperiode nur in den Parametern *From* und *To* festlegen.
From	1	time_period.from	Gültige Zeitzeichenfolge in absoluter (`YYYY-MM-DD hh:mm:ss`) oder relativer Zeitsyntax (`now`, `now/d`, `now/w-1w` usw.). Parameter behavior: - *unterstützt*, wenn *Zeitperiode* nicht gesetzt ist - *erforderlich*, wenn `time_period.to` gesetzt ist
To	1	time_period.to	Gültige Zeitzeichenfolge in absoluter (`YYYY-MM-DD hh:mm:ss`) oder relativer Zeitsyntax (`now`, `now/d`, `now/w-1w` usw.). Parameter behavior: - *unterstützt*, wenn *Zeitperiode* nicht gesetzt ist - *erforderlich*, wenn `time_period.from` gesetzt ist

Legende

Die folgenden Parameter werden für die Konfiguration der *Legende* unterstützt.

Parameter	type	name	value
Legende anzeigen	0	legend	0 - Deaktiviert; 1 - *(Standard)* Aktiviert.
Wert anzeigen	0	legend_value	0 - *(Standard)* Deaktiviert; 1 - Aktiviert. Parameterverhalten: - *unterstützt*, wenn *Legende anzeigen* auf „Aktiviert“ gesetzt ist
Aggregationsfunktion anzeigen	0	legend_aggregation	0 - *(Standard)* Deaktiviert; 1 - Aktiviert. Parameterverhalten: - *unterstützt*, wenn *Legende anzeigen* auf „Aktiviert“ gesetzt ist

Parameter	type	name	value
Zeilen	0	legend_lines_mode	0 - *(Standard)* Fest; 1 - Variabel.
Anzahl der Zeilen/ Maximale Anzahl der Zeilen	0	legend_lines	Parameterverhalten: - *unterstützt*, wenn *Legende anzeigen* auf „Aktiviert“ gesetzt ist Mögliche Werte liegen zwischen 1 und 10. Standard: 1.
Anzahl der Spalten	0	legend_columns	Parameterverhalten: - *unterstützt*, wenn *Legende anzeigen* auf „Aktiviert“ gesetzt ist Mögliche Werte liegen zwischen 1 und 4. Standard: 4.
			Parameterverhalten: - *unterstützt*, wenn *Legende anzeigen* auf „Aktiviert“ und *Wert anzeigen* auf „Deaktiviert“ gesetzt ist

Beispiele

Die folgenden Beispiele beschreiben nur die Konfiguration der Dashboard-Widget-Feldobjekte für das *Kuchendiagramm* Widget. Für weitere Informationen zur Konfiguration eines Dashboards siehe `dashboard.create`.

Konfiguration eines *Kreisdiagramm*-Widgets

Konfigurieren Sie ein *Kreisdiagramm*-Widget wie folgt:

- 2 Datensätze mit insgesamt 9 Datenpunkten auf 1 Host.
- Der erste Datensatz ist vom Typ „Datenpunktliste“ und besteht aus 3 Datenpunkten, die alle vom Typ „Normal“ sind und jeweils durch eine andere Farbe dargestellt werden.
- Der zweite Datensatz ist vom Typ „Datenpunktmuster“, besteht aus 6 Datenpunkten, hat für jeden Datenpunkt eine konfigurierte Aggregation und wird durch eine benutzerdefinierte Farbe dargestellt.
- Der zweite Datensatz hat außerdem eine benutzerdefinierte Datensatzbeschriftung.
- Die Daten im Kreisdiagramm werden als Ringdiagramm mit einer benutzerdefinierten Breite und dem Gesamtwert mit Einheiten in der Mitte angezeigt.
- Die Daten im Kreisdiagramm werden für einen benutzerdefinierten Zeitraum der letzten 3 Stunden angezeigt und aggregiert.
- Die Legende des Kreisdiagramms zeigt konfigurierte Datenpunkte in 4 Zeilen an.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Siehe auch

- Dashboard Widget Feld
- `dashboard.create`
- `dashboard.update`

22 Problem-Hosts

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das Widget *Problem hosts* in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

Attention:
Die Eigenschaften von `fields` des Widgets werden während der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dies ermöglicht es Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets falsch zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Problem hosts* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Parameterverhalten.

Parameter

Die folgenden Parameter werden für das Widget *Problem-Hosts* unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	0	rf_rate	0 - Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - *(Standard)* 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - 15 Minuten.
Host-Gruppen	2	groupids.0	ID der Host-Gruppe. Hinweis: Um mehrere Host-Gruppen zu konfigurieren, erstellen Sie für jede Host-Gruppe ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Nummer im Eigenschaftsnamen. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.

Parameter	type	name	value
Host-Gruppen (Widget)	1	groupids._reference	Anstelle der ID der Host-Gruppe: `ABCDE._hostgroupids` - legen Sie ein kompatibles Widget (mit dem Parameter *Referenz* auf "`ABCDE`" gesetzt) als Datenquelle für Host-Gruppen fest. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Host groups auss-chließen	2	exclude_groupids.0	Host group ID. Hinweis: Um mehrere Hostgruppen auszuschließen, erstellen Sie für jede Hostgruppe ein Dashboard-Widget-Feldobjekt mit einer fortlaufenden Nummer im Eigenschaftsnamen. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem template dashboard konfiguriert wird.
Hosts	3	hostids.0	ID des Hosts. Hinweis: Um mehrere Hosts zu konfigurieren, erstellen Sie für jeden Host ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Nummer im Eigenschaftsnamen. Bei mehreren Hosts darf der Parameter *Host-Gruppen* entweder gar nicht konfiguriert sein oder muss mit mindestens einer Host-Gruppe konfiguriert sein, zu der die konfigurierten Hosts gehören. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Hosts (Widget/Dashboard)	1	hostids._reference	Anstelle der ID des Hosts: `DASHBOARD.hostids` - legen Sie die Host-Auswahl des Dashboards als Datenquelle für Hosts fest; `ABCDE._hostids` - legen Sie ein kompatibles Widget (mit dem Parameter *Referenz* auf "`ABCDE`" gesetzt) als Datenquelle für Hosts fest. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Problem	1	problem	Ereignisname des Problems (Groß-/Kleinschreibung wird nicht beachtet, vollständiger Name oder ein Teil davon).
Schweregrad	0	severities.0	0 - Nicht klassifiziert; 1 - Information; 2 - Warnung; 3 - Durchschnittlich; 4 - Hoch; 5 - Katastrophe. Standard: leer (alle aktiviert). Hinweis: Um mehrere Werte zu konfigurieren, erstellen Sie für jeden Wert ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Nummer im Eigenschaftsnamen.
Problem-Tags			
Auswertungstyp	0	evaltype	0 - *(Standard)* Und/Oder; 2 - Oder.
Tag-Name	1	tags.0.tag	Beliebiger Zeichenfolgenwert. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf die Reihenfolge des Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei Konfiguration von *Problem-Tags*

Parameter	type	name	value
Operator	0	tags.0.operator	0 - Enthält; 1 - Entspricht; 2 - Enthält nicht; 3 - Entspricht nicht; 4 - Existiert; 5 - Existiert nicht. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf die Reihenfolge des Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei Konfiguration von *Problem-Tags*
Tag-Wert	1	tags.0.value	Beliebiger Zeichenfolgenwert. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf die Reihenfolge des Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei Konfiguration von *Problem-Tags*
Unterdrückte Probleme anzeigen	0	show_suppressed	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
Gruppen ohne Probleme ausblenden	0	hide_empty_groups	0 - *(Standard)* Deaktiviert; 1 - Aktiviert. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Problemanzeige	0	ext_ack	0 - *(Standard)* Alle; 1 - Nur unbestätigte; 2 - Getrennt.
Referenz	1	reference	Beliebiger Zeichenfolgenwert, der aus 5 Zeichen besteht (z. B. `ABCDE` oder `JBPNL`). Dieser Wert muss innerhalb des Dashboards, zu dem das Widget gehört, eindeutig sein. Parameterverhalten: - *erforderlich*

Beispiele

Die folgenden Beispiele beschreiben nur die Konfiguration der Feldobjekte des Dashboard-Widgets für das *Problem Hosts* Widget. Für weitere Informationen zur Konfiguration eines Dashboards siehe `dashboard.create`.

Konfigurieren eines Widgets *Problem hosts*

Konfigurieren Sie ein Widget *Problem hosts*, das Hosts aus den Host-Gruppen „2“ und „4“ anzeigt, die Probleme mit einem Namen haben, der die Zeichenfolge „CPU“ enthält, und die die folgenden Schweregrade aufweisen: „Warning“, „Average“, „High“, „Disaster“.

Anfrage:

```
{
    "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
```

```
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Siehe auch

- Dashboard Widget Feld
- `dashboard.create`
- `dashboard.update`

23 Probleme

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das Widget *Problems* in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

Attention:
Die Eigenschaften von `fields` des Widgets werden während der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dies ermöglicht es Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets falsch zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Problems* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Parameterverhalten.

Parameter

Die folgenden Parameter werden für das Widget *Probleme* unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	0	rf_rate	0 - Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - *(Standard)* 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - 15 Minuten.
Anzeigen	0	show	1 - *(Standard)* Aktuelle Probleme; 2 - Verlauf; 3 - Probleme.
Host-Gruppen	2	groupids.0	ID der Host-Gruppe. Hinweis: Um mehrere Host-Gruppen zu konfigurieren, erstellen Sie für jede Host-Gruppe ein Dashboard-Widget-Feldobjekt mit einer fortlaufend erhöhten Nummer im Eigenschaftsnamen. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Host-Gruppen (Widget)	1	groupids._reference	Anstelle der ID der Host-Gruppe: `ABCDE._hostgroupids` - legen Sie ein kompatibles Widget (dessen Parameter *Referenz* auf "`ABCDE`" gesetzt ist) als Datenquelle für Host-Gruppen fest. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.

Parameter	type	name	value
Host-Gruppen auss-chließen	2	exclude_groupids.0	ID der Host-Gruppe. Hinweis: Um mehrere Host-Gruppen auszuschließen, erstellen Sie für jede Host-Gruppe ein Dashboard-Widget-Feldobjekt mit einer fortlaufend erhöhten Nummer im Eigenschaftsnamen. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Hosts	3	hostids.0	ID des Hosts. Hinweis: Um mehrere Hosts zu konfigurieren, erstellen Sie für jeden Host ein Dashboard-Widget-Feldobjekt mit einer fortlaufend erhöhten Nummer im Eigenschaftsnamen. Bei mehreren Hosts muss der Parameter *Host-Gruppen* entweder überhaupt nicht konfiguriert oder mit mindestens einer Host-Gruppe konfiguriert sein, der die konfigurierten Hosts angehören. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Hosts (Widget/Dashboard)	1	hostids._reference	Anstelle der ID des Hosts: `DASHBOARD.hostids` - legen Sie den Host-Auswahlschalter des Dashboards als Datenquelle für Hosts fest; `ABCDE._hostids` - legen Sie ein kompatibles Widget (dessen Parameter *Referenz* auf "`ABCDE`" gesetzt ist) als Datenquelle für Hosts fest. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Problem	1	problem	Problem-Ereignisname (Groß-/Kleinschreibung wird nicht berücksichtigt, vollständiger Name oder Teil davon).
Schweregrad	0	severities.0	0 - Nicht klassifiziert; 1 - Information; 2 - Warnung; 3 - Durchschnittlich; 4 - Hoch; 5 - Katastrophe. Standard: leer (alle aktiviert). Hinweis: Um mehrere Werte zu konfigurieren, erstellen Sie für jeden Wert ein Dashboard-Widget-Feldobjekt mit einer fortlaufend erhöhten Nummer im Eigenschaftsnamen.
Problem-Tags			
Auswertungstyp	0	evaltype	0 - *(Standard)* Und/Oder; 2 - Oder.
Tag-Name	1	tags.0.tag	Beliebiger Zeichenfolgenwert. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf die Reihenfolge des Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich*, wenn *Problem-Tags* konfiguriert werden

Parameter	type	name	value
Operator	0	tags.0.operator	0 - Enthält; 1 - Ist gleich; 2 - Enthält nicht; 3 - Ist nicht gleich; 4 - Existiert; 5 - Existiert nicht. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf die Reihenfolge des Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich*, wenn *Problem-Tags* konfiguriert werden
Tag-Wert	1	tags.0.value	Beliebiger Zeichenfolgenwert. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf die Reihenfolge des Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich*, wenn *Problem-Tags* konfiguriert werden
Tags anzeigen	0	show_tags	0 - *(Standard)* Keine; 1 - 1; 2 - 2; 3 - 3.
Tag-Name (Format)	0	tag_name_format	0 - *(Standard)* Vollständig; 1 - Gekürzt; 2 - Keine. Parameterverhalten: - *unterstützt*, wenn *Tags anzeigen* auf ”1”, ”2” oder ”3” gesetzt ist
Priorität der Tag-Anzeige	1	tag_priority	Kommagetrennte Liste von Tags. Parameterverhalten: - *unterstützt*, wenn *Tags anzeigen* auf ”1”, ”2” oder ”3” gesetzt ist
Betriebsdaten anzeigen	0	show_opdata	0 - *(Standard)* Keine; 1 - Separat; 2 - Mit Problemname.
Unterdrückte Probleme anzeigen	0	show_suppressed	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
Bestätigungsstatus	0	acknowledgement_status	0 - *(Standard)* Alle; 1 - Nicht bestätigt; 2 - Bestätigt.
Von mir	0	acknowledged_by_me	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
Einträge sortieren nach	0	sort_triggers	1 - Schweregrad (absteigend); 2 - Host (aufsteigend); 3 - Zeit (aufsteigend); 4 - *(Standard)* Zeit (absteigend); 13 - Schweregrad (aufsteigend); 14 - Host (absteigend); 15 - Problem (aufsteigend); 16 - Problem (absteigend). Für alle Werte außer ”Zeit (absteigend)” und ”Zeit (aufsteigend)” muss der Parameter *Zeitachse anzeigen* auf ”Deaktiviert” gesetzt sein. Die Werte ”Host (aufsteigend)” und ”Host (absteigend)” werden nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.

Parameter	type	name	value
Zeitachse anzeigen	0	show_timeline	0 - Deaktiviert; 1 - *(Standard)* Aktiviert. Parameterverhalten: - *unterstützt*, wenn *Einträge sortieren nach* auf ”Zeit (absteigend)” oder ”Zeit (aufsteigend)” gesetzt ist
Ganze Zeile hervorheben	0	highlight_row	0 - *(Standard)* Deaktiviert; 1 - Aktiviert. Dieser Parameter wird in kontrastreichen Designs nicht unterstützt.
Zeilen anzeigen	0	show_lines	Mögliche Werte reichen von 1 bis 100. Standard: 25.
Referenz	1	reference	Beliebiger Zeichenfolgenwert aus 5 Zeichen (z. B. `ABCDE` oder `JBPNL`). Dieser Wert muss innerhalb des Dashboards, zu dem das Widget gehört, eindeutig sein. Parameterverhalten: - *erforderlich*

Beispiele

Die folgenden Beispiele beschreiben nur die Konfiguration der Feldobjekte des Dashboard-Widgets für das *Probleme* Widget. Für weitere Informationen zur Konfiguration eines Dashboards siehe `dashboard.create`.

Konfigurieren eines *Probleme*-Widgets

Konfigurieren Sie ein *Probleme*-Widget, das Probleme für die Host-Gruppe „4“ anzeigt, die die folgenden Bedingungen erfüllen:

- Probleme, die ein Tag mit dem Namen „scope“ haben, das die Werte „performance“ oder „availability“ oder „capacity“ enthält.
- Probleme mit den folgenden Schweregraden: „Warning“, „Average“, „High“, „Disaster“.

Konfigurieren Sie das Widget außerdem so, dass Tags und Betriebsdaten angezeigt werden.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Siehe auch

- Dashboard Widget Feld
- `dashboard.create`
- `dashboard.update`

24 Probleme nach Schweregrad

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das Widget *Probleme nach Schweregrad* in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

Attention:
Die Eigenschaften von `fields` des Widgets werden beim Erstellen oder Aktualisieren eines Dashboards nicht validiert. Dies ermöglicht es Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets falsch zu erstellen oder zu aktualisieren. Um sicherzustellen, dass das Widget *Probleme nach Schweregrad* erfolgreich erstellt oder aktualisiert wird, beachten Sie bitte das in den folgenden Tabellen beschriebene Parameterverhalten.

Parameter

Die folgenden Parameter werden für das Widget *Probleme nach Schweregrad* unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	0	rf_rate	0 - Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - *(Standard)* 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - 15 Minuten.
Host-Gruppen	2	groupids.0	ID der Host-Gruppe. Hinweis: Um mehrere Host-Gruppen zu konfigurieren, erstellen Sie für jede Host-Gruppe ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Nummer im Eigenschaftsnamen. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Host-Gruppen (Widget)	1	groupids._reference	Anstelle der ID der Host-Gruppe: `ABCDE._hostgroupids` - legen Sie ein kompatibles Widget (mit dem Parameter *Referenz* auf "`ABCDE`" gesetzt) als Datenquelle für Host-Gruppen fest. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Host-Gruppen ausschließen	2	exclude_groupids.0	ID der Host-Gruppe. Hinweis: Um mehrere Host-Gruppen auszuschließen, erstellen Sie für jede Host-Gruppe ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Nummer im Eigenschaftsnamen. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Hosts	3	hostids.0	ID des Hosts. Hinweis: Um mehrere Hosts zu konfigurieren, erstellen Sie für jeden Host ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Nummer im Eigenschaftsnamen. Bei mehreren Hosts darf der Parameter *Host-Gruppen* entweder gar nicht konfiguriert sein oder muss mit mindestens einer Host-Gruppe konfiguriert sein, zu der die konfigurierten Hosts gehören. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Hosts (Widget/Dashboard)	1	hostids._reference	Anstelle der ID des Hosts: `DASHBOARD.hostids` - legen Sie den Host-Selektor des Dashboards als Datenquelle für Hosts fest; `ABCDE._hostids` - legen Sie ein kompatibles Widget (mit dem Parameter *Referenz* auf "`ABCDE`" gesetzt) als Datenquelle für Hosts fest. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Problem	1	problem	Problem-Ereignisname (Groß-/Kleinschreibung wird nicht beachtet, vollständiger Name oder Teil davon).

Parameter	type	name	value
Schweregrad	0	severities.0	0 - Nicht klassifiziert; 1 - Information; 2 - Warnung; 3 - Durchschnittlich; 4 - Hoch; 5 - Katastrophe. Standard: leer (alle aktiviert). Hinweis: Um mehrere Werte zu konfigurieren, erstellen Sie für jeden Wert ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Nummer im Eigenschaftsnamen.
Problem-Tags			
Auswertungstyp	0	evaltype	0 - *(Standard)* Und/Oder; 2 - Oder.
Tag-Name	1	tags.0.tag	Beliebiger Zeichenfolgenwert. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf die Reihenfolge der Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei Konfiguration von *Problem-Tags*
Operator	0	tags.0.operator	0 - Enthält; 1 - Entspricht; 2 - Enthält nicht; 3 - Entspricht nicht; 4 - Existiert; 5 - Existiert nicht. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf die Reihenfolge der Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei Konfiguration von *Problem-Tags*
Tag-Wert	1	tags.0.value	Beliebiger Zeichenfolgenwert. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf die Reihenfolge der Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei Konfiguration von *Problem-Tags*
Anzeigen	0	show_type	0 - *(Standard)* Host-Gruppen; 1 - Summen. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird, und ist standardmäßig auf ”Summen” gesetzt.
Layout	0	layout	0 - *(Standard)* Horizontal; 1 - Vertikal. Parameterverhalten: - *unterstützt*, wenn *Anzeigen* auf ”Summen” gesetzt ist
Betriebsdaten anzeigen	0	show_opdata	0 - *(Standard)* Keine; 1 - Getrennt; 2 - Mit Problemnamen.
Unterdrückte Probleme anzeigen	0	show_suppressed	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.

Parameter	type	name	value
Gruppen ohne Probleme ausblenden	0	hide_empty_groups	0 - *(Standard)* Deaktiviert; 1 - Aktiviert. Parameterverhalten: - *unterstützt*, wenn *Anzeigen* auf "Host-Gruppen" gesetzt ist Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Problemanzeige	0	ext_ack	0 - *(Standard)* Alle; 1 - Nur unbestätigte; 2 - Getrennt.
Zeitachse anzeigen	0	show_timeline	0 - Deaktiviert; 1 - *(Standard)* Aktiviert.
Referenz	1	reference	Beliebiger Zeichenfolgenwert, bestehend aus 5 Zeichen (z. B. `ABCDE` oder `JBPNL`). Dieser Wert muss innerhalb des Dashboards, zu dem das Widget gehört, eindeutig sein. Parameterverhalten: - *erforderlich*

Beispiele

Die folgenden Beispiele beschreiben nur die Konfiguration der Feldobjekte des Dashboard-Widgets für das *Probleme nach Schweregrad* Widget. Für weitere Informationen zur Konfiguration eines Dashboards siehe `dashboard.create`.

Konfiguration eines Widgets *Probleme nach Schweregrad*

Konfigurieren Sie ein Widget *Probleme nach Schweregrad*, das die Gesamtzahl der Probleme für alle Host-Gruppen anzeigt.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Siehe auch

- Dashboard Widget Feld
- `dashboard.create`
- `dashboard.update`

25 Streudiagramm

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das Widget *Streudiagramm* in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

Attention:
Die Eigenschaften der `fields` des Widgets werden während der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dies ermöglicht es Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets falsch zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Streudiagramm* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Parameterverhalten.

Parameter

Die folgenden Parameter werden für das Widget *Streudiagramm* unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	0	rf_rate	0 - Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - *(Standard)* 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - 15 Minuten.

Parameter	type	name	value
Referenz	1	reference	Beliebiger Zeichenfolgenwert, der aus 5 Zeichen besteht (z. B. `ABCDE` oder `JBPNL`). Dieser Wert muss innerhalb des Dashboards, zu dem das Widget gehört, eindeutig sein. Parameterverhalten: - *erforderlich*

Datensatz

Die folgenden Parameter werden für die Konfiguration eines *Datensatzes* unterstützt.

Note:
Die erste Zahl im Eigenschaftsnamen (z. B. ds.0.hosts.0, ds.0.items.0) steht für den jeweiligen Datensatz, während die zweite Zahl, falls vorhanden, für den konfigurierten Host oder Datenpunkt steht.

Parameter	type	name	value
Datensatztyp ***Datensatz: Datenpunktliste***	0	ds.0.dataset_type	0 - Datenpunktliste; 1 - *(Standard)* Datenpunktmuster.
Farbe	1	ds.0.color.0	Hexadezimaler Farbcode (z. B. `FF0000`). Parameterverhalten: - *erforderlich*, wenn *Datensatztyp* auf „Datenpunktliste“ gesetzt ist
X-Achse	4	ds.0.x_axis_itemids.0	Datenpunkt-ID. Wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird, sollten nur in der Vorlage konfigurierte Datenpunkte gesetzt werden. Hinweis: Um mehrere Datenpunkte zu konfigurieren, erstellen Sie für jeden Datenpunkt ein Dashboard-Widget-Feldobjekt mit einer erhöhten Zahl im Eigenschaftsnamen. Parameterverhalten: - *erforderlich*, wenn *Datensatztyp* auf „Datenpunktliste“ gesetzt ist und *X-Achse (Widget)* nicht gesetzt ist
X-Achse (Widget)	1	ds.0.x_axis_itemids.0._reference	Referenz einer Datenpunkt-ID: `ABCDE._itemid` - ein kompatibles Widget (mit dem auf „`ABCDE`“ gesetzten Parameter *Referenz*) als Datenquelle für Datenpunkte festlegen. Hinweis: Um mehrere Widgets zu konfigurieren, erstellen Sie für jedes Widget ein Dashboard-Widget-Feldobjekt mit einer erhöhten Zahl im Eigenschaftsnamen. Parameterverhalten: - *erforderlich*, wenn *Datensatztyp* auf „Datenpunktliste“ gesetzt ist und *X-Achse* nicht gesetzt ist

Parameter	type	name	value
Y-Achse	4	ds.0.y_axis_itemids.0	Datenpunkt-ID. Wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird, sollten nur in der Vorlage konfigurierte Datenpunkte gesetzt werden. Hinweis: Um mehrere Datenpunkte zu konfigurieren, erstellen Sie für jeden Datenpunkt ein Dashboard-Widget-Feldobjekt mit einer erhöhten Zahl im Eigenschaftsnamen. Parameterverhalten: - *erforderlich*, wenn *Datensatztyp* auf „Datenpunktliste“ gesetzt ist und *Y-Achse (Widget)* nicht gesetzt ist
Y-Achse (Widget)	1	ds.0.y_axis_itemids.0	Referenz einer Datenpunkt-ID: `ABCDE._itemid` - ein kompatibles Widget (mit dem auf „ABCDE“ gesetzten Parameter *Referenz*) als Datenquelle für Datenpunkte festlegen. Hinweis: Um mehrere Widgets zu konfigurieren, erstellen Sie für jedes Widget ein Dashboard-Widget-Feldobjekt mit einer erhöhten Zahl im Eigenschaftsnamen. Parameterverhalten: - *erforderlich*, wenn *Datensatztyp* auf „Datenpunktliste“ gesetzt ist und *Y-Achse* nicht gesetzt ist
Host überschreiben	1	ds.0.override_hostid.0	Referenz: `ABCDE._hostid` - ein kompatibles Widget (mit dem auf „ABCDE“ gesetzten Parameter „Reference“) als Datenquelle für Hosts festlegen; `DASHBOARD._hostid` - die Host-Auswahl des Dashboards als Datenquelle für Hosts festlegen. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Datensatz: Datenpunkt-muster			
Farbe	1	ds.0.color	Hexadezimaler Farbcode (z. B. FF0000). Parameterverhalten: - *unterstützt*, wenn *Datensatztyp* auf „Datenpunktmuster“ gesetzt ist und *Farbpalette* nicht gesetzt ist
Farbpalette	0	ds.0.color_palette	Index der Farbpalette. Möglicher Wertebereich: 0-11. Standard: 0. Parameterverhalten: - *unterstützt*, wenn *Datensatztyp* auf „Datenpunktmuster“ gesetzt ist und *Farbe* nicht gesetzt ist
Host-Muster	1	ds.0.hosts.0	Host-Name oder -Muster (z. B. ”Zabbix*”). Parameterverhalten: - *erforderlich*, wenn *Datensatztyp* auf „Datenpunktmuster“ gesetzt ist Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.

Parameter	type	name	value
X-Achsen-Datenpunktmuster	1	ds.0.x_axis_items.0	Datenpunkt-Name oder -Muster (z. B. "*: Number of processed *values per second"). Wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird, sollten nur die Muster für in der Vorlage konfigurierte Datenpunkte gesetzt werden. Parameterverhalten: - *erforderlich*, wenn *Datensatztyp* auf „Datenpunktmuster" gesetzt ist
Y-Achsen-Datenpunktmuster	1	ds.0.y_axis_items.0	Datenpunkt-Name oder -Muster (z. B. "*: Number of processed *values per second"). Wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird, sollten nur die Muster für in der Vorlage konfigurierte Datenpunkte gesetzt werden. Parameterverhalten: - *erforderlich*, wenn *Datensatztyp* auf „Datenpunktmuster" gesetzt ist
Host-Gruppen	2	groupids.0	Host-Gruppe-ID. Hinweis: Um mehrere Host-Gruppen zu konfigurieren, erstellen Sie für jede Host-Gruppe ein Dashboard-Widget-Feldobjekt mit einer erhöhten Zahl im Eigenschaftsnamen. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Host-Gruppen (Widget)	1	groupids._reference	Anstelle einer Host-Gruppe-ID: `ABCDE._hostgroupids` - ein kompatibles Widget (mit dem auf „`ABCDE`" gesetzten Parameter *Referenz*) als Datenquelle für Host-Gruppen festlegen. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Host überschreiben	1	ds.0.override_hostid._reference	Anstelle einer Host-ID: `ABCDE._hostid` - ein kompatibles Widget (mit dem auf „`ABCDE`" gesetzten Parameter „Reference") als Datenquelle für Hosts festlegen; `DASHBOARD._hostid` - die Host-Auswahl des Dashboards als Datenquelle für Hosts festlegen. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Host-Tags			
Auswertungstyp	0	evaltype_host	0 - *(Standard)* Und/Oder; 2 - Oder. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Tag-Name	1	host_tags.0.tag	Beliebiger Zeichenfolgenwert. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Reihenfolge des Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich*, wenn *Host-Tags* konfiguriert werden Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.

Parameter	type	name	value
Operator	0	host_tags.0.operator	0 - Enthält; 1 - Entspricht; 2 - Enthält nicht; 3 - Entspricht nicht; 4 - Existiert; 5 - Existiert nicht. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Reihenfolge des Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich*, wenn *Host-Tags* konfiguriert werden Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Tag-Wert	1	host_tags.0.value	Beliebiger Zeichenfolgenwert. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Reihenfolge des Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich*, wenn *Host-Tags* konfiguriert werden Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Markierung	0	ds.0.marker	0 - *(Standard)* Ellipse; 1 - Quadrat; 2 - Dreieck; 3 - Raute; 4 - Stern; 5 - Kreuz.
Markierungsgröße	0	ds.0.marker_size	0 - *(Standard)* Klein (6px); 1 - Mittel (9px); 2 - Groß (15px).
Zeitverschiebung	1	ds.0.timeshift	Gültige Zeitzeichenfolge (z. B. 3600, 1h usw.). Sie können Zeitsuffixe verwenden. Negative Werte sind ebenfalls zulässig. Standard: "" (leer).
Aggregationsintervall	1	ds.0.aggregate_interval	Gültige Zeitzeichenfolge (z. B. 3600, 1h usw.). Sie können Zeitsuffixe verwenden. Standard: 15m.
Aggregationsfunktion	0	ds.0.aggregate_function	0 - nicht verwendet; 1 - min; 2 - max; 3 - *(Standard)* avg; 4 - count; 5 - sum; 6 - first; 7 - last.

Anzeigeoptionen

Die folgenden Parameter werden für die Konfiguration von *Anzeigeoptionen* unterstützt.

Parameter	type	name	value
Auswahl der Verlaufsdaten	0	source	0 - *(Standard)* Auto; 1 - Verlauf; 2 - Trends.

Parameter	type	name	value
Host-Namen in Beschriftungen	0	show_hostnames	0 - *(Standard)* Auto; 1 - Anzeigen; 2 - Ausblenden. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.

Zeitperiode

Die folgenden Parameter werden für die Konfiguration der *Zeitperiode* unterstützt.

Parameter	type	name	value
Zeitperiode	1	time_period._reference	`DASHBOARD._timeperiod` - legt den Zeitperiodenwähler des Dashboards als Datenquelle fest; `ABCDE._timeperiod` - legt ein kompatibles Widget (dessen Parameter *Referenz* auf "`ABCDE`" gesetzt ist) als Datenquelle fest. Standard: `DASHBOARD._timeperiod` Alternativ können Sie die Zeitperiode nur in den Parametern *Von* und *Bis* festlegen.
Von	1	time_period.from	Gültige Zeitzeichenfolge in absoluter (`YYYY-MM-DD hh:mm:ss`) oder relativer Zeitsyntax (`now`, `now/d`, `now/w-1w` usw.). Parameterverhalten: - *unterstützt*, wenn *Zeitperiode* nicht gesetzt ist - *erforderlich*, wenn `time_period.to` gesetzt ist
Bis	1	time_period.to	Gültige Zeitzeichenfolge in absoluter (`YYYY-MM-DD hh:mm:ss`) oder relativer Zeitsyntax (`now`, `now/d`, `now/w-1w` usw.). Parameterverhalten: - *unterstützt*, wenn *Zeitperiode* nicht gesetzt ist - *erforderlich*, wenn `time_period.from` gesetzt ist

Achsen

Die folgenden Parameter werden für die Konfiguration von *Achsen* unterstützt.

Parameter	type	name	value
X-Achse	0	x_axis	0 - Deaktiviert; 1 - *(Standard)* Aktiviert.
Y-Achse	0	y_axis	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
Min	1	x_axis_min	Beliebiger numerischer Wert. Standard: "" (leer).
		y_axis_min	
Max	1	x_axis_max	Beliebiger numerischer Wert. Standard: "" (leer).
		y_axis_max	
Einheiten (Typ)	0	x_axis_units	0 - *(Standard)* Automatisch; 1 - Statisch.
		y_axis_units	
Einheiten (Wert)	1	x_axis_static_units	Beliebiger Zeichenfolgenwert. Standard: "" (leer).
		y_axis_static_units	

Legende

Die folgenden Parameter werden für die Konfiguration der *Legende* unterstützt.

Parameter	type	name	value
Legende anzeigen	0	legend	0 - Deaktiviert; 1 - *(Standard)* Aktiviert.
Aggregationsfunktion anzeigen	0	legend_aggregation	0 - *(Standard)* Deaktiviert; 1 - Aktiviert. Parameterverhalten: - *unterstützt*, wenn *Legende anzeigen* auf „Aktiviert“ gesetzt ist
Zeilen	0	legend_lines_mode	0 - *(Standard)* Fest; 1 - Variabel. Parameterverhalten: - *unterstützt*, wenn *Legende anzeigen* auf „Aktiviert“ gesetzt ist
Anzahl der Zeilen/ Maximale Anzahl der Zeilen	0	legend_lines	Mögliche Wertebereich: 1-10. Standard: 1. Parameterverhalten: - *unterstützt*, wenn *Legende anzeigen* auf „Aktiviert“ gesetzt ist
Anzahl der Spalten	0	legend_columns	Mögliche Wertebereich: 1-4. Standard: 4. Parameterverhalten: - *unterstützt*, wenn *Legende anzeigen* auf „Aktiviert“ gesetzt ist

Schwellenwerte

Die folgenden Parameter werden für die Konfiguration von *Schwellenwerten* unterstützt.

Note:
Die Zahl im Eigenschaftsnamen *Thresholds* (z. B. thresholds.0.color) verweist auf die Position des Schwellenwerts in einer Liste, die in aufsteigender Reihenfolge sortiert ist. Wenn Schwellenwerte jedoch in einer anderen Reihenfolge konfiguriert werden, werden die Werte nach der Aktualisierung der Widget-Konfiguration im Zabbix Frontend in aufsteigender Reihenfolge sortiert (z. B. `"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"`; `"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`).

Parameter	type	name	value
Farbinterpolation	0	interpolation	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
Farbe	1	thresholds.0.color	Hexadezimaler Farbcode (z. B. FF0000).
X-Achse	1	thresholds.0.x_axis_threshold	Beliebiger numerischer Wert. Suffixe (z. B. ”1d”, ”2w”, ”4K”, ”8G”) werden unterstützt.
Y-Achse	1	thresholds.0.y_axis_threshold	Beliebiger numerischer Wert. Suffixe (z. B. ”1d”, ”2w”, ”4K”, ”8G”) werden unterstützt.

Beispiele

Die folgenden Beispiele sollen ausschließlich die Konfiguration der Dashboard-Widget-Feldobjekte für das Widget *Streudiagramm* beschreiben. Weitere Informationen zur Konfiguration eines Dashboards finden Sie unter `dashboard.create`.

Konfiguration eines Widgets vom Typ *Scatter plot*

Konfigurieren Sie ein Widget vom Typ *Scatter plot* wie folgt:

- 5 Datensätze vom Typ „Datenpunkt-Muster“, jeweils mit Standardaggregation (avg, 15m) und unterschiedlichen Markierungen.
- Die Legende zeigt konfigurierte Datenpunkte in einer einzelnen Zeile und zwei Spalten an.

- Schwellenwerte sind so konfiguriert, dass eine Markierung rot eingefärbt wird, wenn ihr Wert auf einer beliebigen Achse 80 erreicht.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Siehe auch

- Dashboard-Widget-Feld
- `dashboard.create`
- `dashboard.update`

26 SLA-Bericht

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das Widget *SLA report* in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

Attention:
Die Eigenschaften von `fields` des Widgets werden beim Erstellen oder Aktualisieren eines Dashboards nicht validiert. Dies ermöglicht es Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets falsch zu erstellen oder zu aktualisieren. Um sicherzustellen, dass das Widget *SLA report* erfolgreich erstellt oder aktualisiert wird, beachten Sie bitte das in den folgenden Tabellen beschriebene Parameterverhalten.

Parameter

Die folgenden Parameter werden für das Widget *SLA report* unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	0	rf_rate	0 - *(Standard)* Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - 15 Minuten.
SLA	10	slaid.0	ID der SLA. Parameterverhalten: - *erforderlich*
Service	9	serviceid.0	ID des Service.
Zeiträume anzeigen	0	show_periods	Mögliche Werte liegen zwischen 1 und 100. Standard: 20.
Von	1	date_from	Gültige Datumszeichenfolge im Format `YYYY-MM-DD`. Relative Datumsangaben mit den Modifikatoren `d`, `w`, `M`, `y` (z. B. `now`, `now/d`, `now/w-1w` usw.) werden unterstützt.
Bis	1	date_to	Gültige Datumszeichenfolge im Format `YYYY-MM-DD`. Relative Datumsangaben mit den Modifikatoren `d`, `w`, `M`, `y` (z. B. `now`, `now/d`, `now/w-1w` usw.) werden unterstützt.

Beispiele

Die folgenden Beispiele beschreiben nur die Konfiguration der Dashboard-Widget-Feldobjekte für das *SLA-Bericht* Widget. Für weitere Informationen zur Konfiguration eines Dashboards siehe `dashboard.create`.

Konfigurieren eines *SLA-Bericht*-Widgets

Konfigurieren Sie ein *SLA-Bericht*-Widget, das den SLA-Bericht für den SLA „4“-Service „2“ für den Zeitraum der letzten 30 Tage anzeigt.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
```

```
    "id": 1
}
```

Siehe auch

- Dashboard Widget Feld
- `dashboard.create`
- `dashboard.update`

27 Systeminformationen

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das Widget *Systeminformationen* in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

Attention:
Die Eigenschaften von `fields` des Widgets werden beim Erstellen oder Aktualisieren eines Dashboards nicht validiert. Dies ermöglicht es Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets falsch zu erstellen oder zu aktualisieren. Um sicherzustellen, dass das Widget *Systeminformationen* erfolgreich erstellt oder aktualisiert wird, beachten Sie bitte das in den folgenden Tabellen beschriebene Parameterverhalten.

Parameter

Die folgenden Parameter werden für das Widget *System Information* unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	0	rf_rate	0 - Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - *(Standard)* 15 Minuten.
Anzeigen	0	info_type	0 - *(Standard)* Systemstatistiken; 1 - Knoten mit hoher Verfügbarkeit.
Details der Prüfung auf Softwareaktualisierungen anzeigen	0	show_software_update_check_details	0 - *(Standard)* Deaktiviert; 1 - Aktiviert. Parameterverhalten: - *unterstützt*, wenn `AllowSoftwareUpdateCheck` in der Konfiguration des Zabbix-Servers aktiviert ist und *Anzeigen* auf "System stats" gesetzt ist

Beispiele

Die folgenden Beispiele beschreiben nur die Konfiguration der Feldobjekte des Dashboard-Widgets für das *Systeminformationen* Widget. Für weitere Informationen zur Konfiguration eines Dashboards siehe `dashboard.create`.

Konfigurieren eines Widgets *Systeminformationen*

Konfigurieren Sie ein Widget *Systeminformationen*, das Systemstatistiken mit einem Aktualisierungsintervall von 10 Minuten anzeigt und bei dem die Prüfung auf Software-Updates aktiviert ist.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Siehe auch

- Dashboard Widget Feld
- `dashboard.create`
- `dashboard.update`

28 Top-Hosts

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das Widget *Top hosts* in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

Attention:
Die Eigenschaften von `fields` des Widgets werden beim Erstellen oder Aktualisieren eines Dashboards nicht validiert. Dies ermöglicht es Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets falsch zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Top hosts* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Parameterverhalten.

Parameter

Die folgenden Parameter werden für das Widget *Top-Hosts* unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	0	rf_rate	0 - Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - *(Standard)* 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - 15 Minuten.
Hostgruppen	2	groupids.0	ID der Hostgruppe. Hinweis: Um mehrere Hostgruppen zu konfigurieren, erstellen Sie für jede Hostgruppe ein Dashboard-Widget-Feldobjekt mit einer fortlaufenden Nummer im Eigenschaftsnamen. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Hostgruppen (Widget)	1	groupids._reference	Anstelle der ID der Hostgruppe: `ABCDE._hostgroupids` - Legt ein kompatibles Widget (mit dem Parameter *Referenz* auf "`ABCDE`" gesetzt) als Datenquelle für Hostgruppen fest. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Hosts	3	hostids.0	ID des Hosts. Hinweis: Um mehrere Hosts zu konfigurieren, erstellen Sie für jeden Host ein Dashboard-Widget-Feldobjekt mit einer fortlaufenden Nummer im Eigenschaftsnamen. Bei mehreren Hosts muss der Parameter *Hostgruppen* entweder überhaupt nicht konfiguriert sein oder mindestens eine Hostgruppe enthalten, der die konfigurierten Hosts angehören. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Hosts (Widget/Dashboard)	1	hostids._reference	Anstelle der ID des Hosts: `DASHBOARD.hostids` - Legt den Hostauswahlschalter des Dashboards als Datenquelle für Hosts fest; `ABCDE._hostids` - Legt ein kompatibles Widget (mit dem Parameter *Referenz* auf "`ABCDE`" gesetzt) als Datenquelle für Hosts fest. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Host-Tags			
Auswertungstyp	0	evaltype	0 - *(Standard)* Und/Oder; 2 - Oder. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.

Parameter	type	name	value
Tagname	1	tags.0.tag	Beliebiger Zeichenfolgenwert. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Tag-Reihenfolge in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei der Konfiguration von *Host-Tags* Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Operator	0	tags.0.operator	0 - Enthält; 1 - Ist gleich; 2 - Enthält nicht; 3 - Ist nicht gleich; 4 - Ist vorhanden; 5 - Ist nicht vorhanden. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Tag-Reihenfolge in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei der Konfiguration von *Host-Tags* Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Tagwert	1	tags.0.value	Beliebiger Zeichenfolgenwert. Hinweis: Die Zahl im Eigenschaftsnamen verweist auf die Tag-Reihenfolge in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei der Konfiguration von *Host-Tags* Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Hosts im Wartungsmodus anzeigen	0	maintenance	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
Spalten (siehe unten)			
Sortieren nach	0	column	Numerischer Spaltenwert aus den konfigurierten Spalten.
Reihenfolge	0	order	2 - *(Standard)* Obere N; 3 - Untere N.
Hostlimit	0	show_lines	Mögliche Werte liegen zwischen 1 und 1000. Standard: 10. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.

Spalten

Spalten haben allgemeine Parameter und zusätzliche Parameter, abhängig von der Konfiguration des Parameters *Data*.

Note:
Bei allen Parametern, die sich auf Spalten beziehen, verweist die Zahl im Eigenschaftsnamen (z. B. columns.0.name) auf eine Spalte, für die der Parameter konfiguriert ist.

Die folgenden Parameter werden für alle Spalten unterstützt.

Parameter	type	name	value
Name	1	columns.0.name	Beliebiger Zeichenfolgenwert. Parameterverhalten: - *erforderlich*
Daten	0	columns.0.data	1 - Datenpunkt-Wert; 2 - Host-Name; 3 - Text. Parameterverhalten: - *erforderlich*
Grundfarbe	1	columns.0.base_color	Hexadezimaler Farbcode (z. B. FF0000). Parameterverhalten: - *erforderlich*

Datenpunktwert

Die folgenden Parameter werden unterstützt, wenn *Daten* auf „Datenpunktwert“ gesetzt ist.

Note:
Die erste Zahl im Namen der Eigenschaft *Schwellenwerte* (z. B. columnsthresholds.0.color.0) verweist auf die Spalte, für die Schwellenwerte konfiguriert sind, während die zweite Zahl auf die Position des Schwellenwerts in einer aufsteigend sortierten Liste verweist. Wenn Schwellenwerte jedoch in einer anderen Reihenfolge konfiguriert werden, werden die Werte nach der Aktualisierung der Widget-Konfiguration im Zabbix Frontend aufsteigend sortiert (z. B. `"threshold.0.threshold":"5"` → `"threshold.0.threshold":"1"`; `"threshold.1.threshold":"1"` → `"threshold.1.threshold": "5"`).

Parameter	type	name	value
Datenpunkt	1	columns.0.item	Gültiger Datenpunktname. Bei der Konfiguration des Widgets in einem Vorlagen-Dashboard dürfen nur Datenpunkte festgelegt werden, die auf der Vorlage konfiguriert sind.
Datenpunktwert anzeigen als	0	columns.0.display_value_as	0 - *(Standard)* Numerisch; 1 - Text; 2 - Binär.
Anzeige	0	columns.0.display	1 - *(Standard)* Wie vorhanden; 2 - Balken; 3 - Indikatoren; 6 - Sparkline. Parameterverhalten: - *unterstützt*, wenn *Datenpunktwert anzeigen als* auf „Numerisch“ gesetzt ist
Min	1	columns.0.min	Beliebiger numerischer Wert. Parameterverhalten: - *unterstützt*, wenn *Datenpunktwert anzeigen als* auf „Numerisch“ und *Anzeige* auf „Balken“ oder „Indikatoren“ gesetzt ist

Parameter	type	name	value
Max	1	columns.0.max	Beliebiger numerischer Wert. Parameterverhalten: - *unterstützt*, wenn *Datenpunktwert anzeigen als* auf „Numerisch“ und *Anzeige* auf „Balken“ oder „Indikatoren“ gesetzt ist
Dezimalstellen	0	columns.0.decimal_places	Mögliche Werte liegen zwischen 0 und 10. Standard: 2. Parameterverhalten: - *unterstützt*, wenn *Datenpunktwert anzeigen als* auf „Numerisch“ gesetzt ist
Sparkline			
Breite	0	columns.0.sparkline.width	Mögliche Werte liegen zwischen 0 und 10. Standard: 1. Parameterverhalten: - *unterstützt*, wenn *Datenpunktwert anzeigen als* auf „Numerisch“ und *Anzeige* auf „Sparkline“ gesetzt ist
Füllung	0	columns.0.sparkline.fill	Mögliche Werte liegen zwischen 0 und 10. Standard: 3. Parameterverhalten: - *unterstützt*, wenn *Datenpunktwert anzeigen als* auf „Numerisch“ und *Anzeige* auf „Sparkline“ gesetzt ist
Farbe	1	columns.0.sparkline.color	Hexadezimaler Farbcode (z. B. `FF0000`). Standard: `42A5F5`. Parameterverhalten: - *unterstützt*, wenn *Datenpunktwert anzeigen als* auf „Numerisch“ und *Anzeige* auf „Sparkline“ gesetzt ist
Zeitraum	1	columns.0.sparkline.time_period	`DASHBOARD._timeperiod` - den Zeitraumauswahlschalter des Dashboards als Datenquelle festlegen; `ABCDE._timeperiod` - ein kompatibles Widget (mit dem Parameter `reference` gleich `ABCDE`) als Datenquelle festlegen. Standard: `DASHBOARD._timeperiod` Alternativ können Sie den Zeitraum nur in den Parametern *Von* und *Bis* festlegen. Parameterverhalten: - *unterstützt*, wenn *Datenpunktwert anzeigen als* auf „Numerisch“ und *Anzeige* auf „Sparkline“ gesetzt ist
Von	1	columns.0.sparkline.time_period.from	Gültige Zeitangabe im absoluten (`YYYY-MM-DD hh:mm:ss`) oder relativen Zeitformat (`now`, `now/d`, `now/w-1w` usw.). Parameterverhalten: - *unterstützt*, wenn *Datenpunktwert anzeigen als* auf „Numerisch“ und *Anzeige* auf „Sparkline“ gesetzt ist - *erforderlich*, wenn `columns.0.sparkline.time_period.to` festgelegt ist

Parameter	type	name	value
Bis	1	columns.0.sparkline.time_period.to	Gültige Zeitangabe im absoluten (`YYYY-MM-DD hh:mm:ss`) oder relativen Zeitformat (`now`, `now/d`, `now/w-1w` usw.). Parameterverhalten: - *unterstützt*, wenn *Datenpunktwert anzeigen als* auf „Numerisch“ und *Anzeige* auf „Sparkline“ gesetzt ist - *erforderlich*, wenn `columns.0.sparkline.time_period.from` festgelegt ist
Verlaufsdaten	0	columns.0.sparkline.history	0 - *(Standard)* Automatisch; 1 - Verlauf; 2 - Trends. Parameterverhalten: - *unterstützt*, wenn *Datenpunktwert anzeigen als* auf „Numerisch“ und *Anzeige* auf „Sparkline“ gesetzt ist
Schwellenwerte			
Farbe	1	columnsthresholds.0.color	Hexadezimaler Farbcode (z. B. `FF0000`). Parameterverhalten: - *unterstützt*, wenn *Datenpunktwert anzeigen als* auf „Numerisch“ gesetzt ist
Schwellenwert	1	columnsthresholds.0.threshold	Beliebiger Zeichenfolgenwert. Parameterverhalten: - *unterstützt*, wenn *Datenpunktwert anzeigen als* auf „Numerisch“ gesetzt ist
Hervorhebungen			
Farbe	1	columns.0.highlights.color	Hexadezimaler Farbcode (z. B. `FF0000`). Parameterverhalten: - *unterstützt*, wenn *Datenpunktwert anzeigen als* auf „Text“ gesetzt ist
Muster	1	columns.0.highlights.pattern	Beliebiger Zeichenfolgenwert. Parameterverhalten: - *unterstützt*, wenn *Datenpunktwert anzeigen als* auf „Text“ gesetzt ist
Miniaturansicht anzeigen	0	columns.0.show_thumbnail	0 - *(Standard)* Deaktiviert; 1 - Aktiviert. Parameterverhalten: - *unterstützt*, wenn *Datenpunktwert anzeigen als* auf „Binär“ gesetzt ist
Aggregationsfunktion	0	columns.0.aggregate_function	0 - *(Standard)* nicht verwendet; 1 - Minimum; 2 - Maximum; 3 - Durchschnitt; 4 - Anzahl; 5 - Summe; 6 - Erster; 7 - Letzter.

Parameter	type	name	value
Zeitraum	1	columns.0.time_period.filter	`DASHBOARD._timeperiod` - den Zeitraumauswahlschalter des Dashboards als Datenquelle festlegen; `ABCDE._timeperiod` - ein kompatibles Widget (mit dem Parameter `reference` gleich `ABCDE`) als Datenquelle festlegen. Standard: `DASHBOARD._timeperiod` Alternativ können Sie den Zeitraum nur in den Parametern *Von* und *Bis* festlegen. Parameterverhalten: - *unterstützt*, wenn *Aggregationsfunktion* auf „Minimum“, „Maximum“, „Durchschnitt“, „Anzahl“, „Summe“, „Erster“ oder „Letzter“ gesetzt ist
Von	1	columns.0.time_period.from	Gültige Zeitangabe im absoluten (`YYYY-MM-DD hh:mm:ss`) oder relativen Zeitformat (`now`, `now/d`, `now/w-1w` usw.). Parameterverhalten: - *unterstützt*, wenn *Zeitraum* nicht festgelegt und *Aggregationsfunktion* auf „Minimum“, „Maximum“, „Durchschnitt“, „Anzahl“, „Summe“, „Erster“ oder „Letzter“ gesetzt ist - *erforderlich*, wenn `columns.0.time_period.to` festgelegt ist
Bis	1	columns.0.time_period.to	Gültige Zeitangabe im absoluten (`YYYY-MM-DD hh:mm:ss`) oder relativen Zeitformat (`now`, `now/d`, `now/w-1w` usw.). Parameterverhalten: - *unterstützt*, wenn *Zeitraum* nicht festgelegt und *Aggregationsfunktion* auf „Minimum“, „Maximum“, „Durchschnitt“, „Anzahl“, „Summe“, „Erster“ oder „Letzter“ gesetzt ist - *erforderlich*, wenn `columns.0.time_period.from` festgelegt ist
Verlaufsdaten	0	columns.0.history	0 - *(Standard)* Automatisch; 1 - Verlauf; 2 - Trends. Parameterverhalten: - *unterstützt*, wenn *Datenpunktwert anzeigen als* auf „Numerisch“ gesetzt ist
Referenz	1	reference	Beliebiger Zeichenfolgenwert aus 5 Zeichen (z. B. `ABCDE` oder `JBPNL`). Dieser Wert muss innerhalb des Dashboards, zu dem das Widget gehört, eindeutig sein. Parameterverhalten: - *erforderlich*

Text

Die folgenden Parameter werden unterstützt, wenn *Data* auf ”Text” gesetzt ist.

Parameter	type	name	value
Text	1	columns.0.text	Beliebiger Zeichenfolgenwert, einschließlich Makros. Unterstützte Makros: `{HOST.*}`, `{INVENTORY.*}`. Parameterverhalten: - *erforderlich*, wenn *Data* auf ”Text” gesetzt ist

Beispiele

Die folgenden Beispiele beschreiben nur die Konfiguration der Feldobjekte des Dashboard-Widgets für das *Top hosts* Widget. Für weitere Informationen zur Konfiguration eines Dashboards siehe `dashboard.create`.

Konfigurieren eines Widgets *Top hosts*

Konfigurieren Sie ein Widget *Top hosts*, das die Top-Hosts nach CPU-Auslastung in der Host-Gruppe „4“ anzeigt. Konfigurieren Sie außerdem die folgenden benutzerdefinierten Spalten: „Host-Name“, „CPU-Auslastung in %“, „1m-Durchschnitt“, „5m-Durchschnitt“, „15m-Durchschnitt“, „Prozesse“.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Siehe auch

- Dashboard Widget Feld
- `dashboard.create`
- `dashboard.update`

29 Top-Datenpunkte

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das Widget *Top items* in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

Attention:
Die Eigenschaften von `fields` des Widgets werden während der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dies ermöglicht es Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets falsch zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Top items* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Parameterverhalten.

Parameter

Die folgenden Parameter werden für das Widget *Top items* unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	0	rf_rate	0 - Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - *(Standard)* 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - 15 Minuten.
Host-Gruppen	2	groupids.0	ID der Host-Gruppe. Hinweis: Um mehrere Host-Gruppen zu konfigurieren, erstellen Sie für jede Host-Gruppe ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Nummer im Eigenschaftsnamen. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Host-Gruppen (Widget)	1	groupids._reference	Anstelle der ID der Host-Gruppe: `ABCDE._hostgroupids` - ein kompatibles Widget (mit dem Parameter *Reference* auf "`ABCDE`" gesetzt) als Datenquelle für Host-Gruppen festlegen. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Hosts	3	hostids.0	ID des Hosts. Hinweis: Um mehrere Hosts zu konfigurieren, erstellen Sie für jeden Host ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Nummer im Eigenschaftsnamen. Bei mehreren Hosts darf der Parameter *Host-Gruppen* entweder gar nicht konfiguriert sein oder muss mit mindestens einer Host-Gruppe konfiguriert sein, zu der die konfigurierten Hosts gehören. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Hosts (Widget/Dashboard)	1	hostids._reference	Anstelle der ID des Hosts: `DASHBOARD.hostids` - den Dashboard-Host-Selektor als Datenquelle für Hosts festlegen; `ABCDE._hostids` - ein kompatibles Widget (mit dem Parameter *Reference* auf "`ABCDE`" gesetzt) als Datenquelle für Hosts festlegen. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Host-Tags			
Auswertungstyp	0	host_tags_evaltype	0 - *(Standard)* Und/Oder; 2 - Oder. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.

Parameter	type	name	value
Tag-Name	1	host_tags.0.tag	Beliebiger Zeichenfolgenwert. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf die Reihenfolge der Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei Konfiguration von *Host-Tags* Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Operator	0	host_tags.0.operator	0 - Enthält; 1 - Entspricht; 2 - Enthält nicht; 3 - Entspricht nicht; 4 - Existiert; 5 - Existiert nicht. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf die Reihenfolge der Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei Konfiguration von *Host-Tags* Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Tag-Wert	1	host_tags.0.value	Beliebiger Zeichenfolgenwert. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf die Reihenfolge der Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei Konfiguration von *Host-Tags* Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Layout	0	layout	0 - *(Standard)* Horizontal; 1 - Vertikal.
Probleme anzeigen	0	show_problems	0 - Alle; 1 - *(Standard)* Nicht unterdrückt; 2 - Keine.
Datenpunkte (siehe unten)			

Erweiterte Konfiguration

Die folgenden erweiterten Konfigurationsparameter werden für das Widget *Top-Datenpunkte* unterstützt.

Parameter	type	name	value
Host-Reihenfolge			
Sortieren nach	0	host_ordering_order_by	0 - *(Standard)* Host-Name; 3 - Datenpunktwert.

Parameter	type	name	value
Datenpunktmuster	1	host_ordering_item.0	Name oder Muster des Datenpunkts (z. B. "*: Anzahl der verarbeiteten *Werte pro Sekunde"). Hinweis: Um mehrere Datenpunktmuster zu konfigurieren, erstellen Sie für jedes Datenpunktmuster ein Dashboard-Widget-Feldobjekt mit einer fortlaufenden Nummer im Eigenschaftsnamen. Bei der Konfiguration des Widgets auf einem Vorlagen-Dashboard sollten nur die Muster für Datenpunkte festgelegt werden, die auf der Vorlage konfiguriert sind. Parameterverhalten: - *erforderlich*, wenn *Sortieren nach* auf „Datenpunktwert" gesetzt ist
Reihenfolge	0	host_ordering_order	2 - *(Standard)* Top N; 3 - Bottom N.
Limit	0	host_ordering_limit	Mögliche Werte liegen zwischen 1 und 100. Standard: 10.
Datenpunkt-Reihenfolge			
Sortieren nach	0	item_ordering_order_by	1 - Host; 2 - Datenpunktname; 3 - *(Standard)* Datenpunktwert.
Host-Muster	1	item_ordering_host.0	Name oder Muster des Hosts. Hinweis: Um mehrere Host-Muster zu konfigurieren, erstellen Sie für jedes Host-Muster ein Dashboard-Widget-Feldobjekt mit einer fortlaufenden Nummer im Eigenschaftsnamen. Parameterverhalten: - *erforderlich*, wenn *Sortieren nach* auf „Host" gesetzt ist
Reihenfolge	0	item_ordering_order	2 - *(Standard)* Top N; 3 - Bottom N.
Limit	0	item_ordering_limit	Mögliche Werte liegen zwischen 1 und 100. Standard: 10.

Spalten

Note:
Die erste Zahl im Eigenschaftsnamen (z. B. columns.0.items.0, columns.0.item_tags_evaltype) bezeichnet die jeweilige Spalte, während die zweite Zahl, sofern vorhanden, die konfigurierte Entität (z. B. Datenpunktmuster, Tag) bezeichnet.

Parameter	type	name	value
Datenpunktmuster	1	columns.0.items.0	Name oder Muster des Datenpunkts (z. B. "*: Anzahl der verarbeiteten *Werte pro Sekunde"). Hinweis: Um mehrere Datenpunktmuster zu konfigurieren, erstellen Sie für jedes Datenpunktmuster ein Dashboard-Widget-Feldobjekt mit einer um eins erhöhten zweiten Zahl im Eigenschaftsnamen. Beim Konfigurieren des Widgets auf einem Vorlagen-Dashboard sollten nur die Muster für Datenpunkte festgelegt werden, die auf der Vorlage konfiguriert sind. Parameterverhalten: - *erforderlich*
Datenpunkt-Tags			

Parameter	type	name	value
Auswertungstyp	0	columns.0.item_tags.0.evaltype	0 - *(Standard)* Und/Oder; 2 - Oder.
Tag-Name	1	columns.0.item_tags.0.tag	Beliebiger Zeichenfolgenwert. Hinweis: Die zweite Zahl im Eigenschaftsnamen verweist auf die Reihenfolge des Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich*, wenn *Datenpunkt-Tags* konfiguriert werden
Operator	0	columns.0.item_tags.0.operator	0 - Enthält; 1 - Ist gleich; 2 - Enthält nicht; 3 - Ist nicht gleich; 4 - Existiert; 5 - Existiert nicht. Hinweis: Die zweite Zahl im Eigenschaftsnamen verweist auf die Reihenfolge des Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich*, wenn *Datenpunkt-Tags* konfiguriert werden
Tag-Wert	1	columns.0.item_tags.0.value	Beliebiger Zeichenfolgenwert. Hinweis: Die zweite Zahl im Eigenschaftsnamen verweist auf die Reihenfolge des Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich*, wenn *Datenpunkt-Tags* konfiguriert werden
Grundfarbe	1	columns.0.base_color	Hexadezimaler Farbcode (z. B. FF0000).
Wert anzeigen als	0	columns.0.display_value_as	1 - *(Standard)* Numerisch; 2 - Text.
Anzeige	0	columns.0.display	1 - *(Standard)* Wie angegeben; 2 - Balken; 3 - Indikatoren; 6 - Sparkline.
Min.	1	columns.0.min	Beliebiger numerischer Wert. Suffixe (z. B. ”1d”, ”2w”, ”4K”, ”8G”) werden unterstützt. Parameterverhalten: - *unterstützt*, wenn *Wert anzeigen als* auf ”Numerisch” und *Anzeige* auf ”Balken” oder ”Indikatoren” gesetzt ist
Max.	1	columns.0.max	Beliebiger numerischer Wert. Suffixe (z. B. ”1d”, ”2w”, ”4K”, ”8G”) werden unterstützt. Parameterverhalten: - *unterstützt*, wenn *Wert anzeigen als* auf ”Numerisch” und *Anzeige* auf ”Balken” oder ”Indikatoren” gesetzt ist
Sparkline			
Breite	0	columns.0.sparkline.width	Mögliche Werte liegen zwischen 0 und 10. Standard: 1. Parameterverhalten: - *unterstützt*, wenn *Anzeige* auf ”Sparkline” gesetzt ist
Füllung	0	columns.0.sparkline.fill	Mögliche Werte liegen zwischen 0 und 10. Standard: 3. Parameterverhalten: - *unterstützt*, wenn *Anzeige* auf ”Sparkline” gesetzt ist

Parameter	type	name	value
Farbe	1	columns.0.sparkline.color	Hexadezimaler Farbcode (z. B. FF0000). Standard: 42A5F5. Parameterverhalten: - *unterstützt*, wenn *Anzeige* auf "Sparkline" gesetzt ist
Zeitraum	1	columns.0.sparkline.time_period	DASHBOARD._timeperiod - den Zeitraumauswahlschalter des Dashboards als Datenquelle festlegen; ABCDE._timeperiod - ein kompatibles Widget (dessen reference-Parameter auf ABCDE gesetzt ist) als Datenquelle festlegen. Standard: DASHBOARD._timeperiod Alternativ können Sie den Zeitraum nur in den Parametern *Von* und *Bis* festlegen. Parameterverhalten: - *unterstützt*, wenn *Anzeige* auf "Sparkline" gesetzt ist
Von	1	columns.0.sparkline.time_period.from	Gültige Zeitangabe in absoluter (YYYY-MM-DD hh:mm:ss) oder relativer Zeitsyntax (now, now/d, now/w-1w usw.). Parameterverhalten: - *unterstützt*, wenn *Anzeige* auf "Sparkline" gesetzt ist - *erforderlich*, wenn columns.0.sparkline.time_period.to festgelegt ist
Bis	1	columns.0.sparkline.time_period.to	Gültige Zeitangabe in absoluter (YYYY-MM-DD hh:mm:ss) oder relativer Zeitsyntax (now, now/d, now/w-1w usw.). Parameterverhalten: - *unterstützt*, wenn *Anzeige* auf "Sparkline" gesetzt ist - *erforderlich*, wenn columns.0.sparkline.time_period.from festgelegt ist
Verlaufsdaten	0	columns.0.sparkline.history	0 - *(Standard)* Automatisch; 1 - Verlauf; 2 - Trends. Parameterverhalten: - *unterstützt*, wenn *Anzeige* auf "Sparkline" gesetzt ist
Schwellenwerte			
Farbe	1	columns.0.thresholds.color	Hexadezimaler Farbcode (z. B. FF0000). Parameterverhalten: - *unterstützt*, wenn *Wert anzeigen als* auf "Numerisch" gesetzt ist
Schwellenwert	1	columns.0.thresholds.threshold	Beliebiger numerischer Wert. Suffixe (z. B. "1d", "2w", "4K", "8G") werden unterstützt. Parameterverhalten: - *unterstützt*, wenn *Wert anzeigen als* auf "Numerisch" gesetzt ist
Hervorhebungen			
Hervorhebung	1	columns.0.highlights.color	Hexadezimaler Farbcode (z. B. FF0000). Parameterverhalten: - *unterstützt*, wenn *Wert anzeigen als* auf "Text" gesetzt ist
Schwellenwert	1	columns.0.highlights.pattern	Beliebiger regulärer Ausdruck. Parameterverhalten: - *unterstützt*, wenn *Wert anzeigen als* auf "Text" gesetzt ist
Dezimalstellen	0	columns.0.decimal_places	Mögliche Werte liegen zwischen 0 und 10. Standard: 2.

Parameter	type	name	value
Erweiterte Konfiguration			
Aggregationsfunktion	0	columns.0.aggregate_function	Wenn *Wert anzeigen als* auf ”Numerisch” gesetzt ist: 0 - *(Standard)* nicht verwendet; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last. Wenn *Wert anzeigen als* auf ”Text” gesetzt ist: 0 - *(Standard)* nicht verwendet; 4 - count; 6 - first; 7 - last.
Zeitraum	1	columns.0.time_period.reference	`DASHBOARD._timeperiod` - den Zeitraumauswahlschalter des Dashboards als Datenquelle festlegen; `ABCDE._timeperiod` - ein kompatibles Widget (dessen `reference`-Parameter auf `ABCDE` gesetzt ist) als Datenquelle festlegen. Standard: `DASHBOARD._timeperiod` Alternativ können Sie den Zeitraum nur in den Parametern *Von* und *Bis* festlegen. Parameterverhalten: - *unterstützt*, wenn *Aggregationsfunktion* auf ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first” oder ”last” gesetzt ist
Von	1	columns.0.time_period.from	Gültige Zeitangabe in absoluter (`YYYY-MM-DD hh:mm:ss`) oder relativer Zeitsyntax (`now`, `now/d`, `now/w-1w` usw.). Parameterverhalten: - *unterstützt*, wenn *Zeitraum* nicht festgelegt ist und *Aggregationsfunktion* auf ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first” oder ”last” gesetzt ist - *erforderlich*, wenn `columns.0.time_period.to` festgelegt ist
Bis	1	columns.0.time_period.to	Gültige Zeitangabe in absoluter (`YYYY-MM-DD hh:mm:ss`) oder relativer Zeitsyntax (`now`, `now/d`, `now/w-1w` usw.). Parameterverhalten: - *unterstützt*, wenn *Zeitraum* nicht festgelegt ist und *Aggregationsfunktion* auf ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first” oder ”last” gesetzt ist - *erforderlich*, wenn `columns.0.time_period.from` festgelegt ist
Verlaufsdaten	0	columns.0.history	0 - *(Standard)* Automatisch; 1 - Verlauf; 2 - Trends. Parameterverhalten: - *unterstützt*, wenn *Wert anzeigen als* auf ”Numerisch” gesetzt ist

Parameter	type	name	value
Aggregieren	0	columns.0.aggregate_function	0 - (Standard) Jeder Datenpunkt 1 - Kombiniert. Parameterverhalten: - *unterstützt*, wenn *Wert anzeigen als* auf ”Numerisch” und *Anzeige* auf ”Wie angegeben”, ”Balken” oder ”Indikatoren” gesetzt ist.
Kombinierte Aggregationsfunktion	0	columns.0.column_aggregate_function	1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum *(Standard)*. Parameterverhalten: - *erforderlich*, wenn *Aggregieren* konfiguriert wird.
Name der kombinierten Spalte	1	columns.0.combined_column_name	Beliebige Zeichenfolgenwert. Parameterverhalten: - *erforderlich*, wenn *Aggregieren* konfiguriert wird.

Beispiele

Die folgenden Beispiele beschreiben nur die Konfiguration der Feldobjekte des Dashboard-Widgets für das *Top items* Widget. Für weitere Informationen zur Konfiguration eines Dashboards siehe `dashboard.create`.

Konfiguration eines Widgets *Top items*

Konfigurieren Sie ein Widget *Top items*, das Daten für den Host „10084“ anzeigt und nur für Datenpunkte, bei denen das Tag mit dem Namen „component“ den Wert „cpu“ enthält. Zeigen Sie außerdem die Daten so an, dass sich die Hosts oben befinden, und verwenden Sie für die Zellendarstellung einen farbigen Balken als Messanzeige.

Anfrage:

```
{
    "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
}
```


Antwort:


```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```


Siehe auch

- Dashboard Widget Feld
- `dashboard.create`
- `dashboard.update`

30 wichtigste Auslöser

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das Widget *Top triggers* in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

Attention:
Die Eigenschaften von `fields` des Widgets werden beim Erstellen oder Aktualisieren eines Dashboards nicht validiert. Dies ermöglicht es Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets falsch zu erstellen oder zu aktualisieren. Um sicherzustellen, dass das Widget *Top triggers* erfolgreich erstellt oder aktualisiert wird, beachten Sie bitte das in den folgenden Tabellen beschriebene Parameterverhalten.

Parameter

Die folgenden Parameter werden für das Widget *Top-Auslöser* unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	0	rf_rate	0 - *(Standard)* Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - 15 Minuten.
Host-Gruppen	2	groupids.0	ID der Host-Gruppe. Hinweis: Um mehrere Host-Gruppen zu konfigurieren, erstellen Sie für jede Host-Gruppe ein Dashboard-Widget-Feldobjekt mit einer fortlaufend erhöhten Nummer im Eigenschaftsnamen. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Hosts	3	hostids.0	ID des Hosts. Hinweis: Um mehrere Hosts zu konfigurieren, erstellen Sie für jeden Host ein Dashboard-Widget-Feldobjekt mit einer fortlaufend erhöhten Nummer im Eigenschaftsnamen. Bei mehreren Hosts muss der Parameter *Host-Gruppen* entweder überhaupt nicht konfiguriert sein oder mindestens eine Host-Gruppe enthalten, der die konfigurierten Hosts angehören. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Problem	1	problem	Problem-Ereignisname (Groß-/Kleinschreibung wird nicht berücksichtigt, vollständiger Name oder Teil davon).
Schweregrad	0	severities.0	Auslöser-Schweregrade. 0 - Nicht klassifiziert; 1 - Information; 2 - Warnung; 3 - Durchschnittlich; 4 - Hoch; 5 - Katastrophe. Standard: leer (alle aktiviert). Hinweis: Um mehrere Werte zu konfigurieren, erstellen Sie für jeden Wert ein Dashboard-Widget-Feldobjekt mit einer fortlaufend erhöhten Nummer im Eigenschaftsnamen.
Problem-Tags			

Parameter	type	name	value
Auswertungstyp	0	evaltype	0 - *(Standard)* Und/Oder; 2 - Oder.
Tag-Name	1	tags.0.tag	Beliebiger Zeichenfolgenwert. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf die Tag-Reihenfolge in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich*, wenn *Problem-Tags* konfiguriert werden
Operator	0	tags.0.operator	0 - Enthält; 1 - Ist gleich; 2 - Enthält nicht; 3 - Ist nicht gleich; 4 - Existiert; 5 - Existiert nicht. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf die Tag-Reihenfolge in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich*, wenn *Problem-Tags* konfiguriert werden
Tag-Wert	1	tags.0.value	Beliebiger Zeichenfolgenwert. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf die Tag-Reihenfolge in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich*, wenn *Problem-Tags* konfiguriert werden
Zeitraum	1	time_period._referenceDASHBOARD._timeperiod	`DASHBOARD._timeperiod` - den Zeitraumauswahlschalter des Dashboards als Datenquelle festlegen; `ABCDE._timeperiod` - ein kompatibles Widget (mit dem Parameter *Referenz*, der auf "`ABCDE`" gesetzt ist) als Datenquelle festlegen. Standard: `DASHBOARD._timeperiod` Alternativ können Sie den Zeitraum nur in den Parametern *Von* und *Bis* festlegen.
Von	1	time_period.from	Gültige Zeitangabe im absoluten (`YYYY-MM-DD hh:mm:ss`) oder relativen Zeitformat (`now`, `now/d`, `now/w-1w` usw.). Parameterverhalten: - *unterstützt*, wenn *Zeitraum* nicht festgelegt ist - *erforderlich*, wenn `time_period.to` festgelegt ist
Bis	1	time_period.to	Gültige Zeitangabe im absoluten (`YYYY-MM-DD hh:mm:ss`) oder relativen Zeitformat (`now`, `now/d`, `now/w-1w` usw.). Parameterverhalten: - *unterstützt*, wenn *Zeitraum* nicht festgelegt ist - *erforderlich*, wenn `time_period.from` festgelegt ist
Auslöserlimit	0	show_lines	Mögliche Werte liegen zwischen 1 und 100. Standard: 10.

Beispiele

Die folgenden Beispiele beschreiben nur die Konfiguration der Feldobjekte des Dashboard-Widgets für das *Top Triggers* Widget. Für weitere Informationen zur Konfiguration eines Dashboards siehe `dashboard.create`.

Konfigurieren eines Widgets *Top triggers*

Konfigurieren Sie ein Widget *Top triggers*, das die 5 wichtigsten Auslöser für die Host-Gruppe „4" mit der Anzahl aller Probleme für jeden Auslöser anzeigt. Das Widget zeigt nur Auslöser an, die die Schweregrade „Warning", „Average", „High" oder „Disas-

ter“ haben, sowie Probleme, die ein Tag mit dem Namen „scope“ besitzen, das die Werte „performance“ oder „availability“ oder „capacity“ enthält.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
```

```
        ]
    },
    "id": 1
}
```

Siehe auch

- Dashboard Widget Feld
- `dashboard.create`
- `dashboard.update`

31 Übersicht über Auslöser

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das Widget *Auslöserübersicht* in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

Attention:
Die Eigenschaften von `fields` des Widgets werden während der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dies ermöglicht es Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets falsch zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Auslöserübersicht* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Parameterverhalten.

Parameter

Die folgenden Parameter werden für das Widget *Auslöserübersicht* unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	0	rf_rate	0 - Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - *(Standard)* 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - 15 Minuten.
Anzeigen	0	show	1 - *(Standard)* Aktuelle Probleme; 2 - Beliebig; 3 - Probleme.
Host-Gruppen	2	groupids.0	ID der Host-Gruppe. Hinweis: Um mehrere Host-Gruppen zu konfigurieren, erstellen Sie für jede Host-Gruppe ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Nummer im Eigenschaftsnamen. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Host-Gruppen (Widget)	1	groupids._reference	Anstelle der ID der Host-Gruppe: `ABCDE._hostgroupids` - ein kompatibles Widget (mit dem Parameter *Referenz* auf "ABCDE" gesetzt) als Datenquelle für Host-Gruppen festlegen. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.

Parameter	type	name	value
Hosts	3	hostids.0	ID des Hosts. Hinweis: Um mehrere Hosts zu konfigurieren, erstellen Sie für jeden Host ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Nummer im Eigenschaftsnamen. Bei mehreren Hosts darf der Parameter *Host-Gruppen* entweder gar nicht konfiguriert sein oder muss mit mindestens einer Host-Gruppe konfiguriert sein, zu der die konfigurierten Hosts gehören. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Hosts (Widget/Dashboard)	1	hostids._reference	Anstelle der ID des Hosts: `DASHBOARD.hostids` - die Dashboard-Host-Auswahl als Datenquelle für Hosts festlegen; `ABCDE._hostids` - ein kompatibles Widget (mit dem Parameter *Referenz* auf "`ABCDE`" gesetzt) als Datenquelle für Hosts festlegen. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.
Problem-Tags			
Auswertungstyp	0	evaltype	0 - *(Standard)* Und/Oder; 2 - Oder.
Tag-Name	1	tags.0.tag	Beliebiger Zeichenfolgenwert. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf die Reihenfolge des Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei Konfiguration von *Problem-Tags*
Operator	0	tags.0.operator	0 - Enthält; 1 - Entspricht; 2 - Enthält nicht; 3 - Entspricht nicht; 4 - Existiert; 5 - Existiert nicht. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf die Reihenfolge des Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei Konfiguration von *Problem-Tags*
Tag-Wert	1	tags.0.value	Beliebiger Zeichenfolgenwert. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf die Reihenfolge des Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei Konfiguration von *Problem-Tags*
Unterdrückte Probleme anzeigen	0	show_suppressed	0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
Layout	0	layout	0 - *(Standard)* Horizontal; 1 - Vertikal.

Beispiele

Die folgenden Beispiele beschreiben nur die Konfiguration der Dashboard-Widget-Feldobjekte für das *Auslöserübersicht* Widget. Für weitere Informationen zur Konfiguration eines Dashboards siehe `dashboard.create`.

Konfiguration eines Widgets *Auslöserübersicht*

Konfigurieren Sie ein Widget *Auslöserübersicht*, das Auslöserzustände für alle Host-Gruppen anzeigt, die Auslöser mit einem Tag haben, dessen Name „scope“ ist und dessen Wert „availability“ enthält.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.create",
    "params": {
        "name": "Mein Dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "trigover",
                        "name": "Auslöserübersicht",
                        "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Siehe auch

- Dashboard Widget Feld
- `dashboard.create`
- `dashboard.update`

32 URL

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das Widget *URL* in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

Attention:
Die Eigenschaften von `fields` des Widgets werden während der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dies ermöglicht es Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets falsch zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des *URL*-Widgets sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Parameterverhalten.

Parameter

Die folgenden Parameter werden für das *URL*-Widget unterstützt.

Parameter	Typ	Name	Wert
Aktualisierungsintervall	0	rf_rate	0 - *(Standard)* Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - 15 Minuten.
URL	1	url	Gültige URL-Zeichenfolge. Parameterverhalten: - *erforderlich*
Host überschreiben	1	override_hostid._reference	`ABCDE._hostid` - ein kompatibles Widget (mit dem Parameter *Referenz* auf "`ABCDE`" gesetzt) als Datenquelle für Hosts festlegen; `DASHBOARD._hostid` - die Host-Auswahl des Dashboards als Datenquelle für Hosts festlegen. Dieser Parameter wird nicht unterstützt, wenn das Widget in einem Vorlagen-Dashboard konfiguriert wird.

Beispiele

Die folgenden Beispiele beschreiben nur die Konfiguration der Dashboard-Widget-Feldobjekte für das *URL* -Widget. Für weitere Informationen zur Konfiguration eines Dashboards siehe `dashboard.create`.

Konfigurieren eines *URL*-Widgets

Konfigurieren Sie ein *URL*-Widget, das die Startseite des Zabbix-Handbuchs anzeigt.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Siehe auch

- Dashboard Widget Feld
- dashboard.create
- dashboard.update

33 Web-Überwachung

Beschreibung

Mit diesen Parametern und den möglichen Eigenschaftswerten für die jeweiligen Feldobjekte des Dashboard-Widgets können Sie das Widget *Web monitoring* in den Methoden `dashboard.create` und `dashboard.update` konfigurieren.

Attention:
Die Eigenschaften von `fields` des Widgets werden während der Erstellung oder Aktualisierung eines Dashboards nicht validiert. Dies ermöglicht es Benutzern, integrierte Widgets zu ändern und benutzerdefinierte Widgets zu erstellen, birgt jedoch auch das Risiko, Widgets falsch zu erstellen oder zu aktualisieren. Um die erfolgreiche Erstellung oder Aktualisierung des Widgets *Web monitoring* sicherzustellen, beachten Sie bitte das in den folgenden Tabellen beschriebene Parameterverhalten.

Parameter

Die folgenden Parameter werden für das Widget *Webüberwachung* unterstützt.

Parameter	type	name	value
Aktualisierungsintervall	0	rf_rate	0 - Keine Aktualisierung; 10 - 10 Sekunden; 30 - 30 Sekunden; 60 - *(Standard)* 1 Minute; 120 - 2 Minuten; 600 - 10 Minuten; 900 - 15 Minuten.
Host-Gruppen	2	groupids.0	ID der Host-Gruppe. Hinweis: Um mehrere Host-Gruppen zu konfigurieren, erstellen Sie für jede Host-Gruppe ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Nummer im Eigenschaftsnamen. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Host-Gruppen (Widget)	1	groupids._reference	Anstelle der ID der Host-Gruppe: `ABCDE._hostgroupids` - ein kompatibles Widget (mit dem Parameter *Referenz* auf "`ABCDE`" gesetzt) als Datenquelle für Host-Gruppen festlegen. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Host groups ausschließen	2	exclude_groupids.0	Host group ID. Hinweis: Um mehrere Hostgruppen auszuschließen, erstellen Sie für jede Hostgruppe ein Dashboard-Widget-Feldobjekt mit einer fortlaufenden Nummer im Eigenschaftsnamen. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem template dashboard konfiguriert wird.
Hosts	3	hostids.0	ID des Hosts. Hinweis: Um mehrere Hosts zu konfigurieren, erstellen Sie für jeden Host ein Dashboard-Widget-Feldobjekt mit einer inkrementierten Nummer im Eigenschaftsnamen. Bei mehreren Hosts darf der Parameter *Host-Gruppen* entweder gar nicht konfiguriert sein oder muss mit mindestens einer Host-Gruppe konfiguriert sein, zu der die konfigurierten Hosts gehören. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.

Parameter	type	name	value
Hosts (Widget/Dashboard)	1	hostids._reference	Anstelle der ID des Hosts: `DASHBOARD.hostids` - den Dashboard-Host-Selektor als Datenquelle für Hosts festlegen; `ABCDE._hostids` - ein kompatibles Widget (mit dem Parameter *Referenz* auf "`ABCDE`" gesetzt) als Datenquelle für Hosts festlegen. Dieser Parameter wird nicht unterstützt, wenn das Widget auf einem Vorlagen-Dashboard konfiguriert wird.
Szenario-Tags			
Auswertungstyp	0	evaltype	0 - *(Standard)* Und/Oder; 2 - Oder.
Tag-Name	1	tags.0.tag	Beliebiger Zeichenfolgenwert. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf die Reihenfolge der Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei der Konfiguration von *Szenario-Tags*
Operator	0	tags.0.operator	0 - Enthält; 1 - Entspricht; 2 - Enthält nicht; 3 - Entspricht nicht; 4 - Existiert; 5 - Existiert nicht. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf die Reihenfolge der Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei der Konfiguration von *Szenario-Tags*
Tag-Wert	1	tags.0.value	Beliebiger Zeichenfolgenwert. Hinweis: Die Nummer im Eigenschaftsnamen verweist auf die Reihenfolge der Tags in der Tag-Auswertungsliste. Parameterverhalten: - *erforderlich* bei der Konfiguration von *Szenario-Tags*
Hosts in Wartung anzeigen	0	maintenance	0 - Deaktiviert; 1 - *(Standard)* Aktiviert.
Referenz	1	reference	Beliebiger Zeichenfolgenwert, der aus 5 Zeichen besteht (z. B. `ABCDE` oder `JBPNL`). Dieser Wert muss innerhalb des Dashboards, zu dem das Widget gehört, eindeutig sein. Parameterverhalten: - *erforderlich*

Beispiele

Die folgenden Beispiele beschreiben nur die Konfiguration der Dashboard-Widget-Feldobjekte für das *Webmonitoring* Widget. Für weitere Informationen zur Konfiguration eines Dashboards siehe `dashboard.create`.

Konfigurieren eines *Web monitoring*-Widgets

Konfigurieren Sie ein *Web monitoring*-Widget, das eine Statusübersicht der aktiven Web-Monitoring-Szenarien für die Host-Gruppe „4“ anzeigt.

Anfrage:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "dashboard.create",
    "params": {
        "name": "My dashboard",
        "display_period": 30,
        "auto_start": 1,
        "pages": [
            {
                "widgets": [
                    {
                        "type": "web",
                        "name": "Web monitoring",
                        "x": 0,
                        "y": 0,
                        "width": 18,
                        "height": 3,
                        "view_mode": 0,
                        "fields": [
                            {
                                "type": 2,
                                "name": "groupids.0",
                                "value": 4
                            }
                        ]
                    }
                ]
            }
        ],
        "userGroups": [
            {
                "usrgrpid": 7,
                "permission": 2
            }
        ],
        "users": [
            {
                "userid": 1,
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Siehe auch

- Dashboard Widget Feld
- `dashboard.create`
- `dashboard.update`

Datenpunkt

Diese Klasse ist für die Arbeit mit Datenpunkten vorgesehen.

Objektreferenzen:

- Datenpunkt
  - HTTP-Header
  - HTTP-Abfragefeld
- Datenpunkt-Tag
- Datenpunkt-Vorverarbeitung

Verfügbare Methoden:

- item.create - neue Datenpunkte erstellen
- item.delete - Datenpunkte löschen
- item.get - Datenpunkte abrufen
- item.update - Datenpunkte aktualisieren

Item Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `item` API.

Datenpunkt

> **Note:**
> Web-Datenpunkte können nicht direkt über die Zabbix-API erstellt, aktualisiert oder gelöscht werden.

Das Datenpunktobjekt verfügt über die folgenden Eigenschaften.

Property	Type	Description
itemid	ID	ID des Datenpunkts. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
delay	string	Aktualisierungsintervall des Datenpunkts. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300), einen Wert mit einem Zeit-Suffix (z. B. `30s`, `1m`, `2h`, `1d`) oder optional ein oder mehrere benutzerdefinierte Intervalle, jeweils durch Semikolons getrennt. Benutzerdefinierte Intervalle können eine Mischung aus flexiblen Intervallen und Zeitplanintervallen sein. Akzeptiert Benutzermakros. Bei Verwendung muss der Wert aus einem einzelnen Makro bestehen. Mehrere Makros oder Makros in Kombination mit Text werden nicht unterstützt. Flexible Intervalle können als zwei durch einen Schrägstrich getrennte Makros geschrieben werden (z. B. `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`). Beispiel: `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf ”Zabbix agent” (0), ”Simple check” (3), ”Zabbix internal” (5), ”External check” (10), ”Database monitor” (11), ”IPMI agent” (12), ”SSH agent” (13), ”TELNET agent” (14), ”Calculated” (15), ”JMX agent” (16), ”HTTP agent” (19), ”SNMP agent” (20), ”Script” (21), ”Browser” (22), ”Telemetry query” (24) gesetzt ist oder wenn `type` auf ”Zabbix agent (active)” (7) gesetzt ist und `key_` nicht ”mqtt.get” enthält
hostid	ID	ID des Hosts oder der Vorlage, zu dem bzw. der der Datenpunkt gehört. Verhalten der Eigenschaft: - *konstant* - *erforderlich* für Erstellungsvorgänge

Property	Type	Description
interfaceid	ID	ID der Host-Schnittstelle des Datenpunkts. Verhalten der Eigenschaft: - *erforderlich*, wenn der Datenpunkt zu einem Host gehört und `type` auf ”Zabbix agent”, ”IPMI agent”, ”JMX agent”, ”SNMP trap” oder ”SNMP agent” gesetzt ist - *unterstützt*, wenn der Datenpunkt zu einem Host gehört und `type` auf ”Simple check”, ”External check”, ”SSH agent”, ”TELNET agent” oder ”HTTP agent” gesetzt ist - *schreibgeschützt* für ermittelte Objekte
key_	string	Schlüssel des Datenpunkts. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge - *schreibgeschützt* für geerbte oder ermittelte Objekte
name	string	Name des Datenpunkts. Unterstützt Benutzermakros. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge - *schreibgeschützt* für geerbte oder ermittelte Objekte
name_resolved	string	Name des Datenpunkts mit aufgelösten Benutzermakros. Verhalten der Eigenschaft: - *schreibgeschützt*
type	integer	Typ des Datenpunkts. Mögliche Werte: 0 - Zabbix agent; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix agent (active); 9 - Web item; 10 - External check; 11 - Database monitor; 12 - IPMI agent; 13 - SSH agent; 14 - TELNET agent; 15 - Calculated; 16 - JMX agent; 17 - SNMP trap; 18 - Dependent item; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser; 24 - Telemetry query. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge - *schreibgeschützt* für geerbte oder ermittelte Objekte
url	string	URL-Zeichenfolge. Unterstützt Benutzermakros, {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für geerbte oder ermittelte Objekte

Property	Type	Description
value_type	integer	Informationstyp des Datenpunkts. Mögliche Werte: 0 - numerischer Gleitkommawert; 1 - Zeichen; 2 - Protokoll; 3 - numerischer vorzeichenloser Wert; 4 - Text; 5 - Binärdaten; 6 - JSON. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge - *schreibgeschützt* für geerbte oder ermittelte Objekte
allow_traps	integer	Erlaubt, den Wert ähnlich wie bei einem Trapper-Datenpunkt zu übernehmen. 0 - *(Standard)* Keine eingehenden Daten akzeptieren; 1 - Eingehende Daten akzeptieren. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für ermittelte Objekte
authtype	integer	Authentifizierungsmethode. Mögliche Werte, wenn `type` auf ”SSH agent” gesetzt ist: 0 - *(Standard)* Passwort; 1 - Öffentlicher Schlüssel. Mögliche Werte, wenn `type` auf ”HTTP agent” gesetzt ist: 0 - *(Standard)* Keine; 1 - Basic; 2 - NTLM; 3 - Kerberos; 4 - Digest. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”SSH agent” oder ”HTTP agent” gesetzt ist - *schreibgeschützt* für geerbte Objekte (wenn `type` auf ”HTTP agent” gesetzt ist) oder ermittelte Objekte
description	string	Beschreibung des Datenpunkts. Verhalten der Eigenschaft: - *schreibgeschützt* für ermittelte Objekte
error	string	Fehlertext, wenn beim Aktualisieren des Datenpunktwerts Probleme auftreten. Verhalten der Eigenschaft: - *schreibgeschützt*
flags	integer	Ursprung des Datenpunkts. Mögliche Werte: 0 - ein normaler Datenpunkt; 4 - ein aus einem Prototyp konvertierter Datenpunkt. Verhalten der Eigenschaft: - *schreibgeschützt*

Property	Type	Description
follow_redirects	integer	Antwortweiterleitungen beim Abrufen von Daten verfolgen. Mögliche Werte: 0 - Weiterleitungen nicht verfolgen; 1 - *(Standard)* Weiterleitungen verfolgen. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für geerbte oder ermittelte Objekte
headers	array	Array von Headern, die bei einer HTTP-Anfrage gesendet werden. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für geerbte oder ermittelte Objekte
history	string	Zeiteinheit, wie lange die Verlaufsdaten gespeichert werden sollen. Akzeptiert auch ein Benutzermakro. Standard: `31d`. Verhalten der Eigenschaft: - *schreibgeschützt* für ermittelte Objekte
http_proxy	string	HTTP(S)-Proxy-Verbindungszeichenfolge. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für geerbte oder ermittelte Objekte
inventory_link	integer	ID des Hostinventarfelds, das durch den Datenpunkt befüllt wird. Eine Liste der unterstützten Hostinventarfelder und ihrer IDs finden Sie auf der Hostinventarseite. Standard: `0`. Verhalten der Eigenschaft: - *unterstützt*, wenn `value_type` auf ”numeric float”, ”character”, ”numeric unsigned” oder ”text” gesetzt ist - *schreibgeschützt* für ermittelte Objekte
ipmi_sensor	string	IPMI-Sensor. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf ”IPMI agent” gesetzt ist und `key_` nicht auf ”ipmi.get” gesetzt ist - *unterstützt*, wenn `type` auf ”IPMI agent” gesetzt ist - *schreibgeschützt* für geerbte oder ermittelte Objekte
jmx_endpoint	string	Benutzerdefinierte Verbindungszeichenfolge des JMX agent. Standardwert: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”JMX agent” gesetzt ist - *schreibgeschützt* für ermittelte Objekte
lastclock	timestamp	Zeitpunkt der letzten Aktualisierung des Datenpunktwerts. Standardmäßig werden nur Werte angezeigt, die innerhalb der letzten 24 Stunden liegen. Sie können diesen Zeitraum verlängern, indem Sie den Wert des Parameters *Max history display period* im Menübereich *Administration > General* ändern. Verhalten der Eigenschaft: - *schreibgeschützt*

Property	Type	Description
lastns	integer	Nanosekunden zum Zeitpunkt der letzten Aktualisierung des Datenpunktwerts. Standardmäßig werden nur Werte angezeigt, die innerhalb der letzten 24 Stunden liegen. Sie können diesen Zeitraum verlängern, indem Sie den Wert des Parameters *Max history display period* im Menübereich *Administration > General* ändern. Verhalten der Eigenschaft: - *schreibgeschützt*
lastvalue	string	Letzter Wert des Datenpunkts. Standardmäßig werden nur Werte angezeigt, die innerhalb der letzten 24 Stunden liegen. Sie können diesen Zeitraum verlängern, indem Sie den Wert des Parameters *Max history display period* im Menübereich *Administration > General* ändern. Verhalten der Eigenschaft: - *schreibgeschützt*
logtimefmt	string	Zeitformat in Protokolleinträgen. Verhalten der Eigenschaft: - *unterstützt*, wenn `value_type` auf ”log” gesetzt ist - *schreibgeschützt* für geerbte oder ermittelte Objekte
master_itemid	ID	ID des Master-Datenpunkts. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf ”Dependent item” gesetzt ist - *schreibgeschützt* für geerbte oder ermittelte Objekte
output_format	integer	Gibt an, ob die Antwort in JSON konvertiert werden soll. 0 - *(Standard)* Rohdaten speichern; 1 - In JSON konvertieren. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für geerbte oder ermittelte Objekte
params	string	Zusätzliche Parameter abhängig vom Typ des Datenpunkts: - ausgeführtes Skript für Datenpunkte des Typs SSH agent und TELNET agent; - SQL-Abfrage für Datenpunkte des Typs Database monitor; - Formel für berechnete Datenpunkte; - Skript für Datenpunkte des Typs Script und Browser. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf ”Database monitor”, ”SSH agent”, ”TELNET agent”, ”Calculated”, ”Script” oder ”Browser” gesetzt ist - *schreibgeschützt* für geerbte Objekte (wenn `type` auf ”Script” oder ”Browser” gesetzt ist) oder ermittelte Objekte
parameters	object/array	Zusätzliche Parameter, wenn `type` auf ”Script” oder ”Browser” gesetzt ist. Array von Objekten mit den Eigenschaften `name` und `value`, wobei `name` eindeutig sein muss. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”Script” oder ”Browser” gesetzt ist - *schreibgeschützt* für geerbte oder ermittelte Objekte

Property	Type	Description
password	string	Passwort für die Authentifizierung. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf ”JMX agent” gesetzt ist und `username` festgelegt ist - *unterstützt*, wenn `type` auf ”Simple check”, ”SSH agent”, ”TELNET agent”, ”Database monitor” oder ”HTTP agent” gesetzt ist - *schreibgeschützt* für geerbte Objekte (wenn `type` auf ”HTTP agent” gesetzt ist) oder ermittelte Objekte
post_type	integer	Typ des im Attribut `posts` gespeicherten POST-Datenkörpers. Mögliche Werte: 0 - *(Standard)* Rohdaten; 2 - JSON-Daten; 3 - XML-Daten. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für geerbte oder ermittelte Objekte
posts	string	Daten des HTTP(S)-Anforderungskörpers. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf ”HTTP agent” und `post_type` auf ”JSON data” oder ”XML data” gesetzt ist - *unterstützt*, wenn `type` auf ”HTTP agent” und `post_type` auf ”Raw data” gesetzt ist - *schreibgeschützt* für geerbte oder ermittelte Objekte
prevvalue	string	Vorheriger Wert des Datenpunkts. Standardmäßig werden nur Werte angezeigt, die innerhalb der letzten 24 Stunden liegen. Sie können diesen Zeitraum verlängern, indem Sie den Wert des Parameters *Max history display period* im Menübereich *Administration > General* ändern. Verhalten der Eigenschaft: - *schreibgeschützt*
privatekey	string	Name der Datei mit dem privaten Schlüssel. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf ”SSH agent” und `authtype` auf ”public key” gesetzt ist - *schreibgeschützt* für ermittelte Objekte
publickey	string	Name der Datei mit dem öffentlichen Schlüssel. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf ”SSH agent” und `authtype` auf ”public key” gesetzt ist - *schreibgeschützt* für ermittelte Objekte
query_fields	array	Array von Abfragefeldern, die bei einer HTTP-Anfrage gesendet werden. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für geerbte oder ermittelte Objekte

Property	Type	Description
request_method	integer	Typ der Anforderungsmethode. Mögliche Werte: 0 - *(Standard)* GET; 1 - POST; 2 - PUT; 3 - HEAD. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für geerbte oder ermittelte Objekte
retrieve_mode	integer	Welcher Teil der Antwort gespeichert werden soll. Mögliche Werte, wenn `request_method` auf ”GET”, ”POST” oder ”PUT” gesetzt ist: 0 - *(Standard)* Body; 1 - Header; 2 - Body und Header werden gespeichert. Mögliche Werte, wenn `request_method` auf ”HEAD” gesetzt ist: 1 - Header. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für geerbte oder ermittelte Objekte
snmp_oid	string	SNMP-OID. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf ”SNMP agent” gesetzt ist - *schreibgeschützt* für geerbte oder ermittelte Objekte
ssl_cert_file	string	Dateipfad zum öffentlichen SSL-Schlüssel. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für geerbte oder ermittelte Objekte
ssl_key_file	string	Dateipfad zum privaten SSL-Schlüssel. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für geerbte oder ermittelte Objekte
ssl_key_password	string	Passwort für die SSL-Schlüsseldatei. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für geerbte oder ermittelte Objekte
state	integer	Status des Datenpunkts. Mögliche Werte: 0 - *(Standard)* normal; 1 - nicht unterstützt. Verhalten der Eigenschaft: - *schreibgeschützt*
status	integer	Status des Datenpunkts. Mögliche Werte: 0 - *(Standard)* aktivierter Datenpunkt; 1 - deaktivierter Datenpunkt.

Property	Type	Description
status_codes	string	Bereiche der erforderlichen HTTP-Statuscodes, durch Kommas getrennt. Unterstützt auch Benutzermakros als Teil einer durch Kommas getrennten Liste. Beispiel: 200,200-{$M},{$M},200-400 Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf "HTTP agent" gesetzt ist - *schreibgeschützt* für geerbte oder ermittelte Objekte
templateid	ID	ID des Datenpunkts der übergeordneten Vorlage. *Hinweis*: Verwenden Sie die Eigenschaft `hostid`, um die Vorlage anzugeben, zu der der Datenpunkt gehört. Verhalten der Eigenschaft: - *schreibgeschützt*
timeout	string	Zeitüberschreitung für die Abfrage von Datenpunkten. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300) oder einen Wert mit einem Zeit-Suffix (z. B. 300s, 1m). Akzeptiert auch Benutzermakros. Möglicher Wertebereich: 1-600s. Standard: "" - Proxy-/globale Einstellungen verwenden. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf "Zabbix agent" (0), "Simple check" (3) gesetzt ist und `key_` nicht mit "`vmware.`" oder "`icmpping`" beginnt, wenn `type` auf "Zabbix agent (active)" (7), "External check" (10), "Database monitor" (11), "SSH agent" (13), "TELNET agent" (14), "HTTP agent" (19), "SNMP agent" (20) gesetzt ist und `snmp_oid` mit "`walk[`" oder "`get[`" beginnt, sowie für "Script" (21) und "Browser" (22) - *schreibgeschützt* für geerbte und ermittelte Objekte, - wenn der Standardwert angewendet wird und der Datenpunkt zu einem Host gehört, der von einem nicht erreichbaren Proxy oder einer nicht erreichbaren Proxy-Gruppe überwacht wird, wird die globale Zeitüberschreitung verwendet.
trapper_hosts	string	Zulässige Hosts. Verhalten der Eigenschaft: - *schreibgeschützt* für ermittelte Objekte - *unterstützt*, wenn `type` auf "Zabbix trapper" gesetzt ist oder wenn `type` auf "HTTP agent" und `allow_traps` auf "Allow incoming data" gesetzt ist. Wenn die Eigenschaft nicht angegeben wird, wird sie automatisch auf das {$TRAPPER.ALLOWED_HOSTS}-Benutzermakro (Standardwert) gesetzt. Standardmäßig ist {$TRAPPER.ALLOWED_HOSTS} auf `127.0.0.1,::1` gesetzt. Wenn die Eigenschaft auf eine leere Zeichenfolge gesetzt ist, werden eingehende Verbindungen von allen Hosts abgewiesen.
trends	string	Zeiteinheit, wie lange die Trenddaten gespeichert werden sollen. Akzeptiert auch ein Benutzermakro. Standard: 365d. Verhalten der Eigenschaft: - *unterstützt*, wenn `value_type` auf "numeric float" oder "numeric unsigned" gesetzt ist - *schreibgeschützt* für ermittelte Objekte

Property	Type	Description
units	string	Werteinheiten. Verhalten der Eigenschaft: - *unterstützt*, wenn `value_type` auf ”numeric float” oder ”numeric unsigned” gesetzt ist - *schreibgeschützt* für geerbte oder ermittelte Objekte
username	string	Benutzername für die Authentifizierung. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf ”SSH agent” oder ”TELNET agent” gesetzt ist oder wenn `type` auf ”JMX agent” und `password` festgelegt ist - *unterstützt*, wenn `type` auf ”Simple check”, ”Database monitor” oder ”HTTP agent” gesetzt ist - *schreibgeschützt* für geerbte Objekte (wenn `type` auf ”HTTP agent” gesetzt ist) oder ermittelte Objekte
uuid	string	Universell eindeutiger Bezeichner, der zum Verknüpfen eines importierten Datenpunkts mit bereits vorhandenen Datenpunkten verwendet wird. Wird automatisch generiert, wenn er nicht angegeben ist. Verhalten der Eigenschaft: - *unterstützt*, wenn der Datenpunkt zu einer Vorlage gehört
valuemapid	ID	ID der zugehörigen Wertzuordnung. Verhalten der Eigenschaft: - *unterstützt*, wenn `value_type` auf ”numeric float”, ”character” oder ”numeric unsigned” gesetzt ist - *schreibgeschützt* für geerbte oder ermittelte Objekte
verify_host	integer	Gibt an, ob überprüft werden soll, dass der Hostname für die Verbindung mit dem Hostnamen im Zertifikat des Hosts übereinstimmt. Mögliche Werte: 0 - *(Standard)* Nicht überprüfen; 1 - Überprüfen. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für geerbte oder ermittelte Objekte
verify_peer	integer	Gibt an, ob überprüft werden soll, dass das Zertifikat des Hosts authentisch ist. Mögliche Werte: 0 - *(Standard)* Nicht überprüfen; 1 - Überprüfen. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”HTTP agent” gesetzt ist - *schreibgeschützt* für geerbte oder ermittelte Objekte
query	object	Konfiguration der Telemetry-Abfrage. Legt fest, welche Daten der Datenpunkt aus der Telemetriedatenquelle abfragt, wie die Daten gefiltert werden und wie das Ergebnis aggregiert wird. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf ”Telemetry query” gesetzt ist - *schreibgeschützt* für geerbte oder ermittelte Objekte

Property	Type	Description
time_shift	string	Gibt an, wie weit zurück (vor der aktuellen Zeit) der Zeitraum endet, den der Datenpunkt abfragt. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300) oder einen Wert mit einem Zeit-Suffix (z. B. 30s, 1m, 2h, 1d). Akzeptiert auch Benutzermakros. Möglicher Wertebereich: 0-86400s. Standard: 15s. Verhalten der Eigenschaft: - *unterstützt*, wenn type auf "Telemetry query" gesetzt ist
lookback_limit	string	Gibt an, wie weit zurück (vor der aktuellen Zeit) der Datenpunkt Daten abfragt, wenn keine vorherige Abfrage vorhanden ist, an die angeknüpft werden kann (z. B. beim ersten Lauf des Datenpunkts), oder wenn die vorherige Abfrage älter als delay ist (z. B. weil der Zabbix-Server ausgefallen war). Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300) oder einen Wert mit einem Zeit-Suffix (z. B. 30s, 1m, 2h, 1d). Akzeptiert auch Benutzermakros. Möglicher Wertebereich: 1-259200s. Standard: 10m. Verhalten der Eigenschaft: - *unterstützt*, wenn type auf "Telemetry query" gesetzt ist
granularity	string	Länge eines Abschnitts, in den der gesamte von der Abfrage verwendete Zeitraum aufgeteilt wird. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300) oder einen Wert mit einem Zeit-Suffix (z. B. 30s, 1m, 2h, 1d). Akzeptiert auch Benutzermakros. Möglicher Wertebereich: 1-86400s. Standard: 15s. Verhalten der Eigenschaft: - *unterstützt*, wenn type auf "Telemetry query" gesetzt ist

HTTP-Header

Das Header-Objekt hat die folgenden Eigenschaften:

Eigenschaft	Typ	Beschreibung
name	string	Name des HTTP-Headers. Eigenschaftsverhalten: - *erforderlich*
value	string	Wert des Headers. Eigenschaftsverhalten: - *erforderlich*

HTTP-Abfragefeld

Das Abfragefeldobjekt definiert einen Namen und einen Wert, die zur Angabe eines URL-Parameters verwendet werden. Es hat die folgenden Eigenschaften:

Eigenschaft	Typ	Beschreibung
name	string	Name des Parameters. Eigenschaftsverhalten: - *erforderlich*
value	string	Parameterwert. Eigenschaftsverhalten: - *erforderlich*

Telemetrieabfrage

Das Telemetrieabfrageobjekt verfügt über die folgenden Eigenschaften.

Property	Type	Description
signal_type	integer	Signal-Typ, den der Datenpunkt abfragt. Mögliche Werte: 0 - Traces; 1 - Metriken; 2 - Logs. Property behavior: - *erforderlich*
metric_point_type	integer	Metric point-Typ, den der Datenpunkt abfragt. Mögliche Werte: 0 - Summe; 1 - Gauge; 2 - Histogramm; 3 - Exponentielles Histogramm. Property behavior: - *erforderlich*, wenn `signal_type` auf „Metrics“ gesetzt ist
columns	array	Array von Telemetriedatenspalten, die der Datenpunkt abfragt. Kann leer sein. Property behavior: - *erforderlich*
aggregated_columns	array	Array von aggregierten Telemetriedatenspalten, die der Datenpunkt abfragt, sowie die Art und Weise, wie aus den Spalteneinträgen ein Wert berechnet wird. Property behavior: - *erforderlich*
filter	object	Filter für Telemetrieabfragen für Spalteneinträge in der Abfrage. Property behavior: - *erforderlich*

Spalten für Telemetriedaten

Das Objekt für Spalten mit Telemetriedaten verfügt über die folgenden Eigenschaften.

Property	Type	Description
column	string	Name der Spalte mit Telemetriedaten, die der Datenpunkt abfragt. Siehe unterstützte Spalten für Telemetriedaten. Bei mehreren Spalten mit jeweils einem Wert muss jeder `column`-Wert eindeutig sein. Bei mehreren Spalten, die Schlüssel-Wert-Paare enthalten, muss jedes Paar aus `column` und `attribute_key` eindeutig sein.
attribute_key	string	Name des Schlüssels, den der Datenpunkt innerhalb der Spalte mit Schlüssel-Wert-Paaren abfragt. Bei mehreren Spalten, die Schlüssel-Wert-Paare enthalten, muss jedes Paar aus `column` und `attribute_key` eindeutig sein. Verhalten der Eigenschaft: - *erforderlich*, wenn `column` auf eine Spalte gesetzt ist, die Schlüssel-Wert-Paare enthält

Aggregierte Spalten von Telemetriedaten

Das Objekt für aggregierte Spalten von Telemetriedaten verfügt über die folgenden Eigenschaften.

Property	Type	Description
column	string	Name der Telemetriedatenspalte, die der Datenpunkt abfragt. Siehe unterstützte Telemetriedatenspalten. Bei mehreren Spalten muss jeder `column`-Wert eindeutig sein. Diese Eigenschaft wird nicht unterstützt, wenn `function` auf „Count“ gesetzt ist. Verhalten der Eigenschaft: - *erforderlich*, wenn `function` auf „Min“, „Max“, „Average“, „Sum“ oder „Percentile“ gesetzt ist
function	integer	Funktion, die der Datenpunkt auf die Spalteneinträge anwendet. Mögliche Werte: 1 - Min; 2 - Max; 3 - Average; 4 - Count; 5 - Sum; 8 - Percentile. Verhalten der Eigenschaft: - *erforderlich*
parameters	array	Perzentil, das der Datenpunkt aus den Spalteneinträgen berechnet. Möglicher Wert: eine Zahl von 0 bis 100, als Zeichenfolge angegeben. Verhalten der Eigenschaft: - *erforderlich*, wenn `function` auf „Percentile“ gesetzt ist

Property	Type	Description
alias	string	Eindeutiger Name für den berechneten Wert, der zur Identifizierung in dem Datenpunktwert verwendet wird. Jeder `alias`-Wert muss eindeutig sein. Verhalten der Eigenschaft: - *erforderlich*

Telemetriedatenfilter

Das Objekt für den Telemetriedatenfilter verfügt über die folgenden Eigenschaften.

Property	Type	Beschreibung
conditions	array	Bedingungen des Telemetriedatenfilters, die ein Datensatz in einer Telemetriedatenspalte erfüllen muss. Verhalten der Eigenschaft: - *erforderlich*
evaltype	integer	Auswertungsmethode der Filterbedingung. Mögliche Werte: 0 - And/Or; 1 - And; 2 - Or; 3 - Benutzerdefinierter Ausdruck. Verhalten der Eigenschaft: - *erforderlich*
eval_formula	string	Generierter Ausdruck, der zur Auswertung der Filterbedingungen verwendet wird. Der Ausdruck enthält IDs, die über ihre `formulaid` auf bestimmte Filterbedingungen verweisen. Der Wert von `eval_formula` entspricht bei Filtern mit einem benutzerdefinierten Ausdruck dem Wert von `formula`. Verhalten der Eigenschaft: - *schreibgeschützt*
formula	string	Benutzerdefinierter Ausdruck zur Auswertung der Bedingungen von Filtern mit einem benutzerdefinierten Ausdruck. Der Ausdruck muss IDs enthalten, die über ihre `formulaid` auf bestimmte Filterbedingungen verweisen. Die im Ausdruck verwendeten IDs müssen exakt mit den in den Filterbedingungen definierten IDs übereinstimmen: Keine Bedingung darf ungenutzt bleiben oder ausgelassen werden. Verhalten der Eigenschaft: - *erforderlich*, wenn `evaltype` auf „Benutzerdefinierter Ausdruck“ gesetzt ist

Bedingungen für den Telemetriedatenfilter

Das Objekt für die Bedingungen des Telemetriedatenfilters verfügt über die folgenden Eigenschaften.

Property	Type	Description
column	string	Name der Spalte mit Telemetriedaten, auf die die Filterbedingung angewendet wird. Siehe unterstützte Spalten mit Telemetriedaten. Verhalten der Eigenschaft: - *erforderlich*
attribute_key	string	Name des Schlüssels, auf den die Bedingung innerhalb der Spalte angewendet wird, die Schlüssel-Wert-Paare enthält. Verhalten der Eigenschaft: - *erforderlich*, wenn `column` auf eine Spalte festgelegt ist, die Schlüssel-Wert-Paare enthält
operator	integer	Operator der Filterbedingung. Mögliche Werte für Spalten mit einzelnen Werten: 0 - Ist gleich; 1 - Ist nicht gleich; 2 - Enthält; 3 - Enthält nicht. Mögliche Werte für Spalten, die Schlüssel-Wert-Paare enthalten: 0 - Ist gleich; 1 - Ist nicht gleich; 12 - Ist vorhanden. Verhalten der Eigenschaft: - *erforderlich*
value	string	Wert, mit dem verglichen werden soll. Verhalten der Eigenschaft: - *erforderlich*, wenn `operator` auf „Ist gleich“, „Ist nicht gleich“, „Enthält“ oder „Enthält nicht“ festgelegt ist
formulaid	string	Beliebige eindeutige ID, die verwendet wird, um in einem benutzerdefinierten Ausdruck auf die Bedingung zu verweisen. Darf nur Großbuchstaben enthalten. Die ID muss vom Benutzer beim Ändern von Filterbedingungen definiert werden, wird jedoch bei einer späteren Abfrage neu generiert. Verhalten der Eigenschaft: - *erforderlich*, wenn `evaltype` des Objekts Telemetriedatenabfragefilter auf „Benutzerdefinierter Ausdruck“ festgelegt ist

Datenpunkt-Tag

Das Datenpunkt-Tag-Objekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
tag	string	Name des Datenpunkt-Tags. Verhalten der Eigenschaft: - *erforderlich*
value	string	Wert des Datenpunkt-Tags. Verhalten der Eigenschaft: - *schreibgeschützt*.

Eigenschaft	Typ	Beschreibung
object	integer	Typ des Objekts, von dem das Tag geerbt wurde. Mögliche Werte: 0 - Vorlage; 1 - Host. Verhalten der Eigenschaft: - *schreibgeschützt*.
objectid	ID	ID des Objekts, von dem das Tag geerbt wurde. Verhalten der Eigenschaft: - *schreibgeschützt*.

Datenpunkt-Vorverarbeitung

Das Objekt für die Datenpunkt-Vorverarbeitung verfügt über die folgenden Eigenschaften.

Property	Type	Description
type	integer	Der Typ der Vorverarbeitungsoption. Mögliche Werte: 1 - Benutzerdefinierter Multiplikator; 2 - Rechts trimmen; 3 - Links trimmen; 4 - Trimmen; 5 - Regulärer Ausdruck; 6 - Boolesch in Dezimalzahl; 7 - Oktal in Dezimalzahl; 8 - Hexadezimal in Dezimalzahl; 9 - Einfache Änderung; 10 - Änderung pro Sekunde; 11 - XML XPath; 12 - JSONPath; 13 - Im Bereich; 14 - Entspricht regulärem Ausdruck; 15 - Entspricht nicht regulärem Ausdruck; 16 - Auf Fehler in JSON prüfen; 17 - Auf Fehler in XML prüfen; 18 - Mit regulärem Ausdruck auf Fehler prüfen; 19 - Unveränderte Werte verwerfen; 20 - Unveränderte Werte mit Heartbeat verwerfen; 21 - JavaScript; 22 - Prometheus-Muster; 23 - Prometheus in JSON; 24 - CSV in JSON; 25 - Ersetzen; 26 - Nicht unterstützt prüfen; 27 - XML in JSON; 28 - SNMP-Walk-Wert; 29 - SNMP-Walk in JSON; 30 - SNMP-Get-Wert. Property behavior: - *required*

Property	Type	Description
params	string	Zusätzliche Parameter, die von der Vorverarbeitungsoption verwendet werden. Mehrere Parameter werden durch das Zeilenumbruchzeichen (\n) getrennt. Wenn `type` auf „Nicht unterstützt prüfen“ gesetzt ist, folgen die Parameter der Syntax `<scope>[\n<pattern>]`, wobei *pattern* ein regulärer Ausdruck ist und *scope* einem der folgenden Werte entspricht: -1 - auf jeden Fehler zutreffen; 0 - prüfen, ob die Fehlermeldung mit *pattern* übereinstimmt; 1 - prüfen, ob die Fehlermeldung nicht mit *pattern* übereinstimmt. Property behavior: - *required*, wenn `type` auf „Benutzerdefinierter Multiplikator“ (1), „Rechts trimmen“ (2), „Links trimmen“ (3), „Trimmen“ (4), „Regulärer Ausdruck“ (5), „XML XPath“ (11), „JSONPath“ (12), „Im Bereich“ (13), „Entspricht regulärem Ausdruck“ (14), „Entspricht nicht regulärem Ausdruck“ (15), „Auf Fehler in JSON prüfen“ (16), „Auf Fehler in XML prüfen“ (17), „Mit regulärem Ausdruck auf Fehler prüfen“ (18), „Unveränderte Werte mit Heartbeat verwerfen“ (20), „JavaScript“ (21), „Prometheus-Muster“ (22), „Prometheus in JSON“ (23), „CSV in JSON“ (24), „Ersetzen“ (25), Nicht unterstützt prüfen (26), „SNMP-Walk-Wert“ (28), „SNMP-Walk in JSON“ (29) oder „SNMP-Get-Wert“ (30) gesetzt ist
error_handler	integer	Aktionstyp, der bei einem Fehler in einem Vorverarbeitungsschritt verwendet wird. Mögliche Werte: 0 - Fehlermeldung wird vom Zabbix Server festgelegt; 1 - Wert verwerfen; 2 - Benutzerdefinierten Wert festlegen; 3 - Benutzerdefinierte Fehlermeldung festlegen. Mögliche Werte, wenn `type` auf „Nicht unterstützt prüfen“ gesetzt ist: 1 - Wert verwerfen; 2 - Benutzerdefinierten Wert festlegen; 3 - Benutzerdefinierte Fehlermeldung festlegen. Property behavior: - *required*, wenn `type` auf „Benutzerdefinierter Multiplikator“ (1), „Regulärer Ausdruck“ (5), „Boolesch in Dezimalzahl“ (6), „Oktal in Dezimalzahl“ (7), „Hexadezimal in Dezimalzahl“ (8), „Einfache Änderung“ (9), „Änderung pro Sekunde“ (10), „XML XPath“ (11), „JSONPath“ (12), „Im Bereich“ (13), „Entspricht regulärem Ausdruck“ (14), „Entspricht nicht regulärem Ausdruck“ (15), „Auf Fehler in JSON prüfen“ (16), „Auf Fehler in XML prüfen“ (17), „Mit regulärem Ausdruck auf Fehler prüfen“ (18), „Prometheus-Muster“ (22), „Prometheus in JSON“ (23), „CSV in JSON“ (24), „Nicht unterstützt prüfen“ (26), „XML in JSON“ (27), „SNMP-Walk-Wert“ (28), „SNMP-Walk in JSON“ (29) oder „SNMP-Get-Wert“ (30) gesetzt ist
error_handler_params	string	Parameter des Fehlerhandlers. Property behavior: - *required*, wenn `error_handler` auf „Benutzerdefinierten Wert festlegen“ oder „Benutzerdefinierte Fehlermeldung festlegen“ gesetzt ist

Die folgenden Parameter und Fehlerhandler werden für jeden Vorverarbeitungstyp unterstützt.

Preprocessing type	Name	Parameter 1	Parameter 2	Parameter 3	Supported error handlers
1	Benutzerdefinierter Multiplikator	number[1], 6^			0, 1, 2, 3
2	Rechts trimmen	Liste von Zeichen[2]			
3	Links trimmen	Liste von Zeichen[2]			
4	Trimmen	Liste von Zeichen[2]			
5	Regulärer Ausdruck	pattern[3]	output[2]		0, 1, 2, 3
6	Boolesch in Dezimalzahl				0, 1, 2, 3
7	Oktal in Dezimalzahl				0, 1, 2, 3
8	Hexadezimal in Dezimalzahl				0, 1, 2, 3
9	Einfache Änderung				0, 1, 2, 3
10	Änderung pro Sekunde				0, 1, 2, 3
11	XML XPath	path[4]			0, 1, 2, 3
12	JSONPath	path[4]			0, 1, 2, 3
13	Im Bereich	min^1, 6^	max^1, 6^		0, 1, 2, 3
14	Entspricht regulärem Ausdruck	pattern[3]			0, 1, 2, 3
15	Entspricht nicht regulärem Ausdruck	pattern[3]			0, 1, 2, 3
16	Auf Fehler in JSON prüfen	path[4]			0, 1, 2, 3
17	Auf Fehler in XML prüfen	path[4]			0, 1, 2, 3

Preprocessing type	Name	Parameter 1	Parameter 2	Parameter 3	Supported error handlers
18	Mit regulärem Ausdruck auf Fehler prüfen	pattern[3]	output[2]		0, 1, 2, 3
19	Unveränderte Werte verwerfen	seconds^5, 6^			
20	Unveränderte Werte mit Heartbeat verwerfen	seconds^5, 6^			
21	JavaScript	script[2]			
22	Prometheus-Muster	pattern^6, 7^	`value`, `label`, `function`	output^8, 9^	0, 1, 2, 3
23	Prometheus in JSON	pattern^6, 7^			0, 1, 2, 3
24	CSV in JSON	character[2]	character[2]	0,1	0, 1, 2, 3
25	Ersetzen	search string[2]	replacement[2]		
26	Nicht unterstützt prüfen	scope[1]	pattern^3, 6^		1, 2, 3
27	XML in JSON				0, 1, 2, 3
28	SNMP-Walk-Wert	OID[2]	Format: 0 - Unverändert 1 - UTF-8 aus Hex-STRING 2 - MAC aus Hex-STRING 3 - Ganzzahl aus BITS		0, 1, 2, 3
29	SNMP-Walk in JSON[10]	Field name[2]	OID prefix[2]	Format: 0 - Unverändert 1 - UTF-8 aus Hex-STRING 2 - MAC aus Hex-STRING 3 - Ganzzahl aus BITS	0, 1, 2, 3

Preprocessing type	Name	Parameter 1	Parameter 2	Parameter 3	Supported error handlers
30	SNMP-Get-Wert	Format: 1 - UTF-8 aus Hex-STRING 2 - MAC aus Hex-STRING 3 - Ganzzahl aus BITS			0, 1, 2, 3

[1] Gleitkommazahl (Ganzzahlen werden implizit in Gleitkommawerte konvertiert)
 [2] Zeichenfolge
 [3] regulärer Ausdruck
 [4] JSONPath oder XML XPath
 [5] positive Ganzzahl (mit Unterstützung von Zeitsuffixen, z. B. 30s, 1m, 2h, 1d)
 [6] Benutzermakro
 [7] Prometheus-Muster gemäß der Syntax: `<metric name>{<label name>="<label value>", ...} == <value>`. Jede Komponente des Prometheus-Musters (Metrik, Labelname, Labelwert und Metrikwert) kann ein Benutzermakro sein.
 [8] Prometheus-Ausgabe gemäß der Syntax: `<label name>` (kann ein Benutzermakro sein), wenn `label` als zweiter Parameter ausgewählt ist.
 [9] Eine der Aggregationsfunktionen: `sum`, `min`, `max`, `avg`, `count`, wenn `function` als zweiter Parameter ausgewählt ist.
 [10] Unterstützt mehrere durch ein Zeilenumbruchzeichen getrennte Datensätze im Format „Field name,OID prefix,Format“.

datenpunkt.create

Beschreibung

`object item.create(object/array items)`

Diese Methode ermöglicht es Ihnen, neue Datenpunkte zu erstellen.

Note:
Web-Datenpunkte können nicht über die Zabbix API erstellt werden.

Note:
Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellende Datenpunkte.

Zusätzlich zu den Standard-Datenpunkt-Eigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
preprocessing	array	Optionen für die Datenpunkt-Vorverarbeitung.
tags	array	Datenpunkt-Tags.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Datenpunkte in der Eigenschaft `itemids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Datenpunkte.

Beispiele

Erstellen eines Datenpunkts

Erstellen Sie einen numerischen Zabbix-Agent-Datenpunkt mit 2 Datenpunkt-Tags, um den freien Festplattenspeicher auf dem Host mit der ID „30074“ zu überwachen.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "24758"
        ]
    },
    "id": 1
}
```

Erstellen eines Host-Inventar-Datenpunkts

Erstellen Sie einen Zabbix-Agent-Datenpunkt, um das Inventarfeld „OS“ des Hosts zu befüllen.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "24759"
        ]
    },
    "id": 1
}
```

Erstellen eines Datenpunkts mit Vorverarbeitung

Erstellen Sie einen Datenpunkt mit benutzerdefiniertem Multiplikator.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44210"
        ]
    },
    "id": 1
}
```

Erstellen eines abhängigen Datenpunkts

Erstellen Sie einen abhängigen Datenpunkt für den Master-Datenpunkt mit der ID 24759. Es sind nur Abhängigkeiten auf demselben Host zulässig, daher sollten der Master-Datenpunkt und der abhängige Datenpunkt dieselbe hostid haben.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
```

```
            "44211"
        ]
    },
    "id": 1
}
```

HTTP-Agent-Datenpunkt erstellen

Erstellen Sie einen Datenpunkt mit der Anfragemethode POST und Vorverarbeitung der JSON-Antwort.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Skript-Datenpunkt erstellen

Erstellen Sie eine einfache Datenerfassung mit einem Skript-Datenpunkt.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "name": "Skriptbeispiel",
        "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Quelle

CItem::create() in *ui/include/classes/api/services/CItem.php*.

datenpunkt.delete

Beschreibung

`object item.delete(array itemIds)`

Diese Methode ermöglicht es Ihnen, Datenpunkte zu löschen.

> **Note:**
> Web-Datenpunkte können nicht über die Zabbix API gelöscht werden.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der zu löschenden Datenpunkte.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Datenpunkte unter der Eigenschaft `itemids` enthält.

Beispiele

Mehrere Datenpunkte löschen

Löschen Sie zwei Datenpunkte.
 Abhängige Datenpunkte und Datenpunkt-Prototypen werden automatisch entfernt, wenn der Master-Datenpunkt gelöscht wird.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "item.delete",
    "params": [
        "22982",
        "22986"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "22982",
            "22986"
        ]
    },
    "id": 1
}
```

Quelle

CItem::delete() in *ui/include/classes/api/services/CItem.php*.

datenpunkt.get

Beschreibung

`integer/array item.get(object parameters)`

Die Methode ermöglicht es Ihnen, Datenpunkte anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
itemids	ID/array	Gibt nur Datenpunkte mit den angegebenen IDs zurück.
groupids	ID/array	Gibt nur Datenpunkte zurück, die zu den Hosts aus den angegebenen Gruppen gehören.
templateids	ID/array	Gibt nur Datenpunkte zurück, die zu den angegebenen Vorlagen gehören.
hostids	ID/array	Gibt nur Datenpunkte zurück, die zu den angegebenen Hosts gehören.
proxyids	ID/array	Gibt nur Datenpunkte zurück, die von den angegebenen Proxys überwacht werden.
interfaceids	ID/array	Gibt nur Datenpunkte zurück, die die angegebenen Host-Schnittstellen verwenden.
graphids	ID/array	Gibt nur Datenpunkte zurück, die in den angegebenen Diagrammen verwendet werden.
triggerids	ID/array	Gibt nur Datenpunkte zurück, die in den angegebenen Auslösern verwendet werden.
webitems	flag	Schließt Web-Datenpunkte in das Ergebnis ein.

Parameter	Type	Beschreibung
inherited	boolean	Wenn auf `true` gesetzt, werden nur Datenpunkte zurückgegeben, die von einer Vorlage geerbt wurden.
inheritedTags	boolean	Gibt Datenpunkte zurück, die die angegebenen `tags` auch in Host/Vorlage/verknüpften Vorlagen haben. Mögliche Werte: `true` - Vorlage/Host/verknüpfte Vorlagen müssen die angegebenen Tags ebenfalls haben; `false` - *(Standard)* Tags aus Vorlage/Host/verknüpften Vorlagen werden ignoriert.
templated	boolean	Wenn auf `true` gesetzt, werden nur Datenpunkte zurückgegeben, die zu Vorlagen gehören.
monitored	boolean	Wenn auf `true` gesetzt, werden nur aktivierte Datenpunkte zurückgegeben, die zu überwachten Hosts gehören.
group	string	Gibt nur Datenpunkte zurück, die zu einer Gruppe mit dem angegebenen Namen gehören.
host	string	Gibt nur Datenpunkte zurück, die zu einem Host mit dem angegebenen Namen gehören.
evaltype	integer	Auswertungsmethode für Tags. Mögliche Werte: 0 - *(Standard)* Und/Oder; 2 - Oder.
tags	array	Gibt nur Datenpunkte mit den angegebenen Tags zurück. Format: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Ein leeres Array gibt alle Datenpunkte zurück. Mögliche Werte für operator: 0 - *(Standard)* Enthält; 1 - Gleich; 2 - Enthält nicht; 3 - Ungleich; 4 - Existiert; 5 - Existiert nicht.
with_triggers	boolean	Wenn auf `true` gesetzt, werden nur Datenpunkte zurückgegeben, die in Auslösern verwendet werden.
selectHosts	query	Gibt eine Eigenschaft `hosts` mit einem Array von Hosts zurück, zu denen der Datenpunkt gehört.
selectInterfaces	query	Gibt eine Eigenschaft `interfaces` mit einem Array von Host-Schnittstellen zurück, die vom Datenpunkt verwendet werden.
selectTriggers	query	Gibt eine Eigenschaft `triggers` mit den Auslösern zurück, in denen der Datenpunkt verwendet wird. Unterstützt `count`.
selectGraphs	query	Gibt eine Eigenschaft `graphs` mit den Diagrammen zurück, die den Datenpunkt enthalten. Unterstützt `count`.

Parameter	Type	Beschreibung
selectDiscoveryData	query	Gibt eine Eigenschaft `discoveryData` mit den Objektdaten der Datenpunkterkennung zurück. Das Datenpunkterkennungsobjekt verknüpft einen erkannten Datenpunkt mit einem Datenpunktprototyp, aus dem er erkannt wurde. Es hat die folgenden Eigenschaften: `parent_itemid` - `(string)` ID des Datenpunktprototyps, aus dem der Datenpunkt erstellt wurde; `key_` - `(string)` Schlüssel des Datenpunktprototyps; `status` - `(int)` Status der Datenpunkterkennung: 0 - *(Standard)* Datenpunkt ist erkannt, 1 - Datenpunkt wird nicht mehr erkannt; `ts_delete` - `(timestamp)` Zeitpunkt, zu dem ein nicht mehr erkannter Datenpunkt gelöscht wird; `ts_disable` - `(timestamp)` Zeitpunkt, zu dem ein nicht mehr erkannter Datenpunkt deaktiviert wird; `disable_source` - `(int)` Kennzeichen, ob der Datenpunkt durch eine LLD-Regel oder manuell deaktiviert wurde: 0 - *(Standard)* automatisch deaktiviert, 1 - durch eine LLD-Regel deaktiviert.
selectDiscoveryRule	query	Gibt eine Eigenschaft `discoveryRule` mit der LLD-Regel zurück, die den Datenpunkt erstellt hat.
selectInheritedTags	query	Gibt eine Eigenschaft `inheritedTags` mit Tags zurück, die sich auf Vorlage/Host/verknüpften Vorlagen befinden.
selectPreprocessing	query	Gibt eine Eigenschaft `preprocessing` mit Optionen für die Datenpunktvorverarbeitung zurück.
selectTags	query	Gibt die Datenpunkt-Tags in der Eigenschaft `tags` zurück.
selectValueMap	query	Gibt eine Eigenschaft `valuemap` mit der Wertezuordnung des Datenpunkts zurück.
filter	object	Gibt nur die Ergebnisse zurück, die exakt dem angegebenen Filter entsprechen. Akzeptiert ein Objekt, bei dem die Schlüssel Eigenschaftsnamen sind und die Werte entweder ein einzelner Wert oder ein Array von Werten sind, mit denen verglichen werden soll. Unterstützt keine Eigenschaften vom Datentyp `text`. Unterstützt zusätzliche Eigenschaften: `host` - technischer Name des Hosts, zu dem der Datenpunkt gehört.
limitSelects	integer	Begrenzt die Anzahl der von Unterabfragen zurückgegebenen Datensätze. Gilt für die folgenden Unterabfragen: `selectGraphs` - Ergebnisse werden nach `name` sortiert; `selectTriggers` - Ergebnisse werden nach `description` sortiert.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `itemid`, `name`, `key_`, `delay`, `history`, `trends`, `type`, `status`.
countOutput	boolean	Diese Parameter sind in der Referenzkommentierung beschrieben.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Parameter	Type	Beschreibung
selectItemDiscovery	query	Gibt eine Eigenschaft `itemDiscovery` mit dem Datenpunkterkennungsobjekt zurück. Das Datenpunkterkennungsobjekt verknüpft den Datenpunkt mit einem Datenpunktprototyp, aus dem er erstellt wurde. Diese Abfrage ist **veraltet**; bitte verwenden Sie stattdessen `selectDiscoveryData`.

Rückgabewerte

`(integer/array)` Gibt entweder Folgendes zurück:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Datenpunkte nach Schlüssel finden

Rufen Sie alle Datenpunkte ab, die in Auslösern für eine bestimmte Host-ID verwendet werden, deren Datenpunktschlüssel das Wort „system.cpu“ enthält, und sortieren Sie die Ergebnisse nach Namen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "output": "extend",
        "hostids": "10084",
        "with_triggers": true,
        "search": {
            "key_": "system.cpu"
        },
        "sortfield": "name"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "42269",
            "type": "18",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "CPU utilization",
            "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": "CPU utilization in %.",
        "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": "CPU utilization",
        "parameters": [],
        "lastclock": "0",
        "lastns": "0",
        "lastvalue": "0",
        "prevvalue": "0"
    },
    {
        "itemid": "42259",
        "type": "0",
        "snmp_oid": "",
        "hostid": "10084",
        "name": "Load average (15m avg)",
        "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": "Load average (15m avg)",
        "parameters": [],
        "lastclock": "0",
        "lastns": "0",
        "lastvalue": "0",
        "prevvalue": "0"
    },
    {
        "itemid": "42249",
        "type": "0",
        "snmp_oid": "",
        "hostid": "10084",
        "name": "Load average (1m avg)",
        "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": "Load average (1m avg)",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0"
        },
        {
            "itemid": "42257",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "Load average (5m avg)",
            "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": "Load average (5m avg)",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0"
        },
        {
            "itemid": "42260",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "Number of CPUs",
            "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": "Number of CPUs",
            "parameters": [],
            "lastclock": "0",
            "lastns": "0",
            "lastvalue": "0",
            "prevvalue": "0"
        }
    ],
    "id": 1
}
```


Abhängige Datenpunkte nach Schlüssel finden

Rufen Sie alle abhängigen Datenpunkte vom Host mit der ID „10116“ ab, die das Wort „apache“ im Schlüssel enthalten.

Anfrage:


```
{
    "jsonrpc": "2.0",
    "method": "item.get",
```

```
    "params": {
        "output": "extend",
        "hostids": "10116",
        "search": {
            "key_": "apache"
        },
        "filter": {
            "type": 18
        }
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```


HTTP-Agent-Datenpunkt finden

Finden Sie einen HTTP-Agent-Datenpunkt mit dem Post-Body-Typ XML für eine bestimmte Host-ID.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "hostids": "10255",
        "filter": {
            "type": 19,
            "post_type": 3
        }
    },
    "id": 1
}
```

Antwort:


```
{
    "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
}
```


Abrufen von Datenpunkten mit Vorverarbeitungsregeln

Rufen Sie alle Datenpunkte und ihre Vorverarbeitungsregeln für eine bestimmte Host-ID ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "output": ["itemid", "name", "key_"],
        "selectPreprocessing": "extend",
        "hostids": "10254"
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```

Siehe auch

- Discovery-Regel
- Graph
- Host
- Host-Schnittstelle
- Auslöser

Quelle

CItem::get() in *ui/include/classes/api/services/CItem.php*.

item.update

Beschreibung

`object item.update(object/array items)`

Diese Methode ermöglicht es Ihnen, vorhandene Datenpunkte zu aktualisieren.

> **Note:**
> Web-Datenpunkte können nicht über die Zabbix API aktualisiert werden.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu aktualisierende Datenpunkt-Eigenschaften.

Die Eigenschaft `itemid` muss für jeden Datenpunkt definiert sein, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Zusätzlich zu den Standard-Datenpunkt-Eigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
preprocessing	array	Optionen für die Datenpunkt-Vorverarbeitung, die die aktuellen Vorverarbeitungsoptionen ersetzen. Parameterverhalten: - *schreibgeschützt* für geerbte Objekte oder entdeckte Objekte
tags	array	Datenpunkt-Tags. Parameterverhalten: - *schreibgeschützt* für entdeckte Objekte

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Elemente unter der Eigenschaft `itemids` enthält.

Beispiele

Einen Datenpunkt aktivieren

Aktivieren Sie einen Datenpunkt, d. h. setzen Sie seinen Status auf „0“.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "10092",
        "status": 0
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "10092"
        ]
    },
    "id": 1
}
```

Abhängigen Datenpunkt aktualisieren

Aktualisieren Sie den Namen des abhängigen Datenpunkts und die ID des Master-Datenpunkts. Es sind nur Abhängigkeiten auf demselben Host zulässig, daher sollten Master- und abhängiger Datenpunkt dieselbe hostid haben.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "name": "Dependent item updated name",
        "master_itemid": "25562",
        "itemid": "189019"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "itemids": [
            "189019"
        ]
    },
    "id": 1
}
```

HTTP-Agent-Datenpunkt aktualisieren

Aktivieren Sie das Trapping von Datenpunktwerten.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "23856",
        "allow_traps": 1
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23856"
        ]
    },
    "id": 1
}
```

Aktualisieren eines Datenpunkts mit Vorverarbeitung

Aktualisieren Sie einen Datenpunkt mit der Datenpunkt-Vorverarbeitungsregel „Im Bereich“.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "23856",
        "preprocessing": [
            {
                "type": 13,
                "params": "\n100",
                "error_handler": 1,
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23856"
        ]
    },
```

```
    "id": 1
}
```

Aktualisieren eines Skript-Datenpunkts

Aktualisieren Sie einen Skript-Datenpunkt mit einem anderen Skript und entfernen Sie unnötige Parameter, die vom vorherigen Skript verwendet wurden.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "23865",
        "parameters": [],
        "params": "Zabbix.log(3, 'Log test');\nreturn 1;"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Quelle

CItem::update() in *ui/include/classes/api/services/CItem.php*.

Datenpunkt-Prototyp

Diese Klasse ist für die Arbeit mit Datenpunkt-Prototypen vorgesehen.

Objektreferenzen:

- Datenpunkt-Prototyp
  - HTTP-Header
  - HTTP-Abfragefeld
- Datenpunkt-Prototyp-Tag
- Datenpunkt-Prototyp-Vorverarbeitung

Verfügbare Methoden:

- itemprototype.create - neue Datenpunkt-Prototypen erstellen
- itemprototype.delete - Datenpunkt-Prototypen löschen
- itemprototype.get - Datenpunkt-Prototypen abrufen
- itemprototype.update - Datenpunkt-Prototypen aktualisieren

Item-Prototyp-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `itemprototype` API.

Datenpunkt-Prototyp

Das Datenpunkt-Prototypobjekt verfügt über die folgenden Eigenschaften.

Property	Type	Beschreibung
itemid	ID	ID des Datenpunkt-Prototyps. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
delay	string	Aktualisierungsintervall des Datenpunkt-Prototyps. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300), einen Wert mit einem Zeit-Suffix (z. B. 30s, 1m, 2h, 1d) oder optional ein oder mehrere benutzerdefinierte Intervalle, die durch Semikolons getrennt sind. Benutzerdefinierte Intervalle können eine Mischung aus flexiblen Intervallen und Zeitplanintervallen sein. Akzeptiert Benutzermakros und LLD-Makros. Bei Verwendung muss der Wert aus einem einzelnen Makro bestehen. Mehrere Makros oder Makros in Kombination mit Text werden nicht unterstützt. Flexible Intervalle können als zwei durch einen Schrägstrich getrennte Makros geschrieben werden (z. B. `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`). Beispiel: `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „Zabbix agent“ (0), „Simple check“ (3), „Zabbix internal“ (5), „External check“ (10), „Database monitor“ (11), „IPMI agent“ (12), „SSH agent“ (13), „TELNET agent“ (14), „Calculated“ (15), „JMX agent“ (16), „HTTP agent“ (19), „SNMP agent“ (20), „Script“ (21), „Browser“ (22), „Telemetry query“ (24) gesetzt ist oder wenn `type` auf „Zabbix agent (active)“ (7) gesetzt ist und `key_` nicht „mqtt.get“ enthält
hostid	ID	ID des Hosts, zu dem der Datenpunkt-Prototyp gehört. Verhalten der Eigenschaft: - *konstant* - *erforderlich* für Erstellungsvorgänge
interfaceid	ID	ID der Host-Schnittstelle des Datenpunkt-Prototyps. Verhalten der Eigenschaft: - *erforderlich*, wenn der Datenpunkt-Prototyp zu einem Host gehört und `type` auf „Zabbix agent“, „IPMI agent“, „JMX agent“, „SNMP trap“ oder „SNMP agent“ gesetzt ist - *unterstützt*, wenn der Datenpunkt-Prototyp zu einem Host gehört und `type` auf „Simple check“, „External check“, „SSH agent“, „TELNET agent“ oder „HTTP agent“ gesetzt ist
key_	string	Schlüssel des Datenpunkt-Prototyps. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge - *schreibgeschützt* für geerbte Objekte
name	string	Name des Datenpunkt-Prototyps. Unterstützt Benutzermakros. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge - *schreibgeschützt* für geerbte Objekte

Property	Type	Beschreibung
type	integer	Typ des Datenpunkt-Prototyps. Mögliche Werte: 0 - Zabbix agent; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix agent (active); 10 - External check; 11 - Database monitor; 12 - IPMI agent; 13 - SSH agent; 14 - TELNET agent; 15 - Calculated; 16 - JMX agent; 17 - SNMP trap; 18 - Dependent item; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser; 24 - Telemetry query. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge - *schreibgeschützt* für geerbte Objekte
url	string	URL-Zeichenfolge. Unterstützt LLD-Makros, Benutzermakros, {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
value_type	integer	Informationstyp des Datenpunkt-Prototyps. Mögliche Werte: 0 - numerischer Gleitkommawert; 1 - Zeichen; 2 - Protokoll; 3 - numerischer Ganzzahlwert ohne Vorzeichen; 4 - Text; 5 - Binärdaten; 6 - JSON. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge - *schreibgeschützt* für geerbte Objekte
allow_traps	integer	Ermöglicht das Befüllen des Werts ähnlich wie bei einem Trapper-Datenpunkt. 0 - *(Standard)* Keine eingehenden Daten akzeptieren; 1 - Eingehende Daten akzeptieren. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist

Property	Type	Beschreibung
authtype	integer	Authentifizierungsmethode. Mögliche Werte, wenn `type` auf „SSH agent“ gesetzt ist: 0 - *(Standard)* Passwort; 1 - öffentlicher Schlüssel. Mögliche Werte, wenn `type` auf „HTTP agent“ gesetzt ist: 0 - *(Standard)* keine; 1 - Basic; 2 - NTLM; 3 - Kerberos; 4 - Digest. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „SSH agent“ oder „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte (wenn `type` auf „HTTP agent“ gesetzt ist)
description	string	Beschreibung des Datenpunkt-Prototyps.
flags	integer	Herkunft des Datenpunkt-Prototyps. Mögliche Werte: 2 - ein Datenpunkt-Prototyp; 6 - ein ermittelter Datenpunkt-Prototyp Verhalten der Eigenschaft: - *schreibgeschützt*
follow_redirects	integer	Weiterleitungen der Antwort beim Abrufen von Daten verfolgen. Mögliche Werte: 0 - Weiterleitungen nicht verfolgen; 1 - *(Standard)* Weiterleitungen verfolgen. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
headers	array	Array von Headern, die bei einer HTTP-Anfrage gesendet werden. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
history	string	Zeiteinheit für die Dauer der Speicherung von Verlaufsdaten. Akzeptiert außerdem ein Benutzermakro und ein LLD-Makro. Standard: 31d.
http_proxy	string	HTTP(S)-Proxy-Verbindungszeichenfolge. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
ipmi_sensor	string	IPMI-Sensor. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „IPMI agent“ gesetzt und `key_` nicht auf „ipmi.get“ gesetzt ist - *unterstützt*, wenn `type` auf „IPMI agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte

Property	Type	Beschreibung
jmx_endpoint	string	Benutzerdefinierte Verbindungszeichenfolge des JMX agent. Standard: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „JMX agent“ gesetzt ist
logtimefmt	string	Zeitformat in Protokolleinträgen. Verhalten der Eigenschaft: - *unterstützt*, wenn `value_type` auf „log“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
master_itemid	ID	ID des Master-Datenpunkts. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „Dependent item“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
output_format	integer	Gibt an, ob die Antwort in JSON konvertiert werden soll. Mögliche Werte: 0 - *(Standard)* Rohdaten speichern; 1 - In JSON konvertieren. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
params	string	Zusätzliche Parameter abhängig vom Typ des Datenpunkt-Prototyps: - ausgeführtes Skript für Datenpunkt-Prototypen vom Typ SSH agent und TELNET agent; - SQL-Abfrage für Datenpunkt-Prototypen vom Typ Database monitor; - Formel für Datenpunkt-Prototypen vom Typ Calculated; - Skript für Datenpunkt-Prototypen vom Typ Script und Browser. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „Database monitor“, „SSH agent“, „TELNET agent“, „Calculated“, „Script“ oder „Browser“ gesetzt ist - *schreibgeschützt* für geerbte Objekte (wenn `type` auf „Script“ oder „Browser“ gesetzt ist)
parameters	object/array	Zusätzliche Parameter, wenn `type` auf „Script“ oder „Browser“ gesetzt ist. Array von Objekten mit den Eigenschaften `name` und `value`, wobei `name` eindeutig sein muss. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „Script“ oder „Browser“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
password	string	Passwort für die Authentifizierung. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „JMX agent“ gesetzt und `username` angegeben ist - *unterstützt*, wenn `type` auf „Simple check“, „SSH agent“, „TELNET agent“, „Database monitor“ oder „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte (wenn `type` auf „HTTP agent“ gesetzt ist)

Property	Type	Beschreibung
post_type	integer	Typ des im Feld `posts` gespeicherten Textkörpers. Mögliche Werte: 0 - *(Standard)* Rohdaten. 2 - JSON-Daten. 3 - XML-Daten. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
posts	string	Textkörper der HTTP(S)-Anfrage. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „HTTP agent“ und `post_type` auf „JSON data“ oder „XML data“ gesetzt ist - *unterstützt*, wenn `type` auf „HTTP agent“ und `post_type` auf „Raw data“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
privatekey	string	Name der Datei mit dem privaten Schlüssel. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „SSH agent“ und `authtype` auf „public key“ gesetzt ist
publickey	string	Name der Datei mit dem öffentlichen Schlüssel. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „SSH agent“ und `authtype` auf „public key“ gesetzt ist
query_fields	array	Array von Abfragefeldern, die bei einer HTTP-Anfrage gesendet werden. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
request_method	integer	Typ der Anfragemethode. Mögliche Werte: 0 - *(Standard)* GET; 1 - POST; 2 - PUT; 3 - HEAD. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
retrieve_mode	integer	Gibt an, welcher Teil der Antwort gespeichert werden soll. Mögliche Werte, wenn `request_method` auf „GET“, „POST“ oder „PUT“ gesetzt ist: 0 - *(Standard)* Textkörper; 1 - Header; 2 - Textkörper und Header werden gespeichert. Mögliche Werte, wenn `request_method` auf „HEAD“ gesetzt ist: 1 - Header. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte

Property	Type	Beschreibung
snmp_oid	string	SNMP-OID. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „SNMP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
ssl_cert_file	string	Dateipfad des öffentlichen SSL-Schlüssels. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
ssl_key_file	string	Dateipfad des privaten SSL-Schlüssels. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
ssl_key_password	string	Passwort für die SSL-Schlüsseldatei. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
status	integer	Status des Datenpunkt-Prototyps. Mögliche Werte: 0 - *(Standard)* aktivierter Datenpunkt-Prototyp; 1 - deaktivierter Datenpunkt-Prototyp; 3 - nicht unterstützter Datenpunkt-Prototyp.
status_codes	string	Durch Kommas getrennte Bereiche erforderlicher HTTP-Statuscodes. Unterstützt außerdem Benutzermakros oder LLD-Makros als Teil einer durch Kommas getrennten Liste. Beispiel: 200,200-{$M},{$M},200-400 Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
templateid	ID	ID des Datenpunkt-Prototyps der übergeordneten Vorlage. Verhalten der Eigenschaft: - *schreibgeschützt*
timeout	string	Zeitüberschreitung für die Abfrage von Datenpunkten. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300) oder einen Wert mit einem Zeit-Suffix (z. B. 30s, 1m). Akzeptiert außerdem Benutzermakros und LLD-Makros. Möglicher Wertebereich: 1-600s. Standard: ”” - Proxy-/globale Einstellungen verwenden. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „Zabbix agent“ (0), „Simple check“ (3) gesetzt ist und `key_` nicht mit „`vmware.`“ oder „`icmpping`“ beginnt, „Zabbix agent (active)“ (7), „External check“ (10), „Database monitor“ (11), „SSH agent“ (13), „TELNET agent“ (14), „HTTP agent“ (19), „SNMP agent“ (20) gesetzt ist und `snmp_oid` mit „`walk[`“ oder „`get[`“ beginnt, sowie „Script“ (21) oder „Browser“ (22) - *schreibgeschützt* für geerbte Objekte

Property	Type	Beschreibung
trapper_hosts	string	Zulässige Hosts. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „Zabbix trapper“ gesetzt ist oder wenn `type` auf „HTTP agent“ und `allow_traps` auf „Allow incoming data“ gesetzt ist. Wenn die Eigenschaft nicht angegeben wird, wird sie automatisch auf das Benutzermakro `{$TRAPPER.ALLOWED_HOSTS}` gesetzt (Standardwert). Standardmäßig ist `{$TRAPPER.ALLOWED_HOSTS}` `127.0.0.1,::1`. Wenn die Eigenschaft auf eine leere Zeichenfolge gesetzt ist, werden eingehende Verbindungen von allen Hosts abgelehnt.
trends	string	Zeiteinheit für die Dauer der Speicherung von Trenddaten. Akzeptiert außerdem ein Benutzermakro und ein LLD-Makro. Standard: 365d. Verhalten der Eigenschaft: - *unterstützt*, wenn `value_type` auf „numeric float“ oder „numeric unsigned“ gesetzt ist
units	string	Werteinheiten. Verhalten der Eigenschaft: - *unterstützt*, wenn `value_type` auf „numeric float“ oder „numeric unsigned“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
username	string	Benutzername für die Authentifizierung. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „SSH agent“ oder „TELNET agent“ gesetzt ist oder wenn `type` auf „JMX agent“ und `password` angegeben ist - *unterstützt*, wenn `type` auf „Simple check“, „Database monitor“ oder „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte (wenn `type` auf „HTTP agent“ gesetzt ist)
uuid	string	Universell eindeutiger Bezeichner, der zum Verknüpfen importierter Datenpunkt-Prototypen mit bereits vorhandenen Prototypen verwendet wird. Wird automatisch generiert, wenn er nicht angegeben ist. Verhalten der Eigenschaft: - *unterstützt*, wenn der Datenpunkt-Prototyp zu einer Vorlage gehört
valuemapid	ID	ID der zugehörigen Wertzuordnung. Verhalten der Eigenschaft: - *unterstützt*, wenn `value_type` auf „numeric float“, „character“ oder „numeric unsigned“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
verify_host	integer	Gibt an, ob überprüft werden soll, dass der Hostname der Verbindung mit dem Hostnamen im Zertifikat des Hosts übereinstimmt. Mögliche Werte: 0 - *(Standard)* Nicht überprüfen; 1 - Überprüfen. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte

Property	Type	Beschreibung
verify_peer	integer	Gibt an, ob überprüft werden soll, dass das Zertifikat des Hosts authentisch ist. Mögliche Werte: 0 - *(Standard)* Nicht überprüfen; 1 - Überprüfen. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
discover	integer	Ermittlungsstatus des Datenpunkt-Prototyps. Mögliche Werte: 0 - *(Standard)* Neue Datenpunkte werden ermittelt; 1 - Neue Datenpunkte werden nicht ermittelt und vorhandene Datenpunkte als verloren markiert.
query	object	Konfiguration der Telemetry-Abfrage. Definiert, welche Daten der Datenpunkt aus der Telemetriedatenquelle abfragt, wie die Daten gefiltert werden und wie das Ergebnis aggregiert wird. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „Telemetry query“ gesetzt ist - *schreibgeschützt* für geerbte oder ermittelte Objekte
time_shift	string	Gibt an, wie weit vor der aktuellen Zeit der Zeitraum endet, den der Datenpunkt abfragt. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300) oder einen Wert mit einem Zeit-Suffix (z. B. 30s, 1m, 2h, 1d). Akzeptiert außerdem Benutzermakros und LLD-Makros. Möglicher Wertebereich: 0-86400s. Standard: 15s. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „Telemetry query“ gesetzt ist
lookback_limit	string	Gibt an, wie weit vor der aktuellen Zeit der Datenpunkt Daten abfragt, wenn keine vorherige Abfrage vorhanden ist, von der aus fortgefahren werden kann (z. B. beim ersten Lauf des Datenpunkts), oder wenn die vorherige Abfrage älter als `delay` ist (z. B. weil der Zabbix Server nicht verfügbar war). Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300) oder einen Wert mit einem Zeit-Suffix (z. B. 30s, 1m, 2h, 1d). Akzeptiert außerdem Benutzermakros und LLD-Makros. Möglicher Wertebereich: 1-259200s. Standard: 10m. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „Telemetry query“ gesetzt ist

Property	Type	Beschreibung
granularity	string	Länge eines Abschnitts, in den der gesamte Zeitraum der Abfrage aufgeteilt wird. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300) oder einen Wert mit einem Zeit-Suffix (z. B. 30s, 1m, 2h, 1d). Akzeptiert außerdem Benutzermakros und LLD-Makros. Möglicher Wertebereich: 1-86400s. Standard: 15s. Verhalten der Eigenschaft: - *unterstützt*, wenn type auf „Telemetry query“ gesetzt ist

HTTP-Header

Das Header-Objekt hat die folgenden Eigenschaften:

Eigenschaft	Typ	Beschreibung
name	string	Name des HTTP-Headers. Eigenschaftsverhalten: - *erforderlich*
value	string	Wert des Headers. Eigenschaftsverhalten: - *erforderlich*

HTTP-Abfragefeld

Das HTTP-Abfragefeldobjekt definiert einen Namen und einen Wert, die zur Angabe eines URL-Parameters verwendet werden. Es hat die folgenden Eigenschaften:

Eigenschaft	Typ	Beschreibung
name	string	Name des Parameters. Eigenschaftsverhalten: - *erforderlich*
value	string	Wert des Parameters. Eigenschaftsverhalten: - *erforderlich*

Telemetrieabfrage

Das Telemetrieabfrageobjekt verfügt über die folgenden Eigenschaften.

Property	Type	Description
signal_type	integer	Typ des Signals, den der Datenpunkt abfragt. Mögliche Werte: 0 - Traces; 1 - Metriken; 2 - Logs. Verhalten der Eigenschaft: - *erforderlich*
metric_point_type	integer	Typ des Metrikpunkts, den der Datenpunkt abfragt. Mögliche Werte: 0 - Summe; 1 - Gauge; 2 - Histogramm; 3 - Exponentielles Histogramm. Verhalten der Eigenschaft: - *erforderlich*, wenn `signal_type` auf „Metrics“ gesetzt ist
columns	array	Array von Telemetriedatenspalten, die der Datenpunkt abfragt. Kann leer sein. Verhalten der Eigenschaft: - *erforderlich*
aggregated_columns	array	Array von aggregierten Telemetriedatenspalten, die der Datenpunkt abfragt, sowie die Art und Weise, wie aus den Spalteneinträgen ein Wert berechnet wird. Verhalten der Eigenschaft: - *erforderlich*
filter	object	Filter der Telemetrieabfrage für Spalteneinträge in der Abfrage. Verhalten der Eigenschaft: - *erforderlich*

Spalten für Telemetriedaten

Das Objekt für Spalten mit Telemetriedaten verfügt über die folgenden Eigenschaften.

Property	Type	Description
column	string	Name der Spalte mit Telemetriedaten, die der Datenpunkt abfragt. Siehe unterstützte Spalten für Telemetriedaten. Bei mehreren Spalten mit jeweils einem Wert muss jeder `column`-Wert eindeutig sein. Bei mehreren Spalten, die Schlüssel-Wert-Paare enthalten, muss jedes Paar aus `column` und `attribute_key` eindeutig sein.
attribute_key	string	Name des Schlüssels, den der Datenpunkt innerhalb der Spalte mit Schlüssel-Wert-Paaren abfragt. Bei mehreren Spalten, die Schlüssel-Wert-Paare enthalten, muss jedes Paar aus `column` und `attribute_key` eindeutig sein. Verhalten der Eigenschaft: - *required*, wenn `column` auf eine Spalte gesetzt ist, die Schlüssel-Wert-Paare enthält

Aggregierte Spalten von Telemetriedaten

Das Objekt für aggregierte Spalten von Telemetriedaten verfügt über die folgenden Eigenschaften.

Property	Type	Description
column	string	Name der Telemetriedatenspalte, die der Datenpunkt abfragt. Siehe unterstützte Telemetriedatenspalten. Bei mehreren Spalten muss jeder `column`-Wert eindeutig sein. Diese Eigenschaft wird nicht unterstützt, wenn `function` auf „Count“ gesetzt ist. Verhalten der Eigenschaft: - *erforderlich*, wenn `function` auf „Min“, „Max“, „Average“, „Sum“ oder „Percentile“ gesetzt ist
function	integer	Funktion, die der Datenpunkt auf die Spalteneinträge anwendet. Mögliche Werte: 1 - Min; 2 - Max; 3 - Average; 4 - Count; 5 - Sum; 8 - Percentile. Verhalten der Eigenschaft: - *erforderlich*
parameters	array	Perzentil, das der Datenpunkt aus den Spalteneinträgen berechnet. Möglicher Wert: eine Zahl von 0 bis 100, angegeben als Zeichenkette. Verhalten der Eigenschaft: - *erforderlich*, wenn `function` auf „Percentile“ gesetzt ist
alias	string	Eindeutiger Name für den berechneten Wert, der zur Identifizierung in dem Datenpunktwert verwendet wird. Jeder `alias`-Wert muss eindeutig sein. Verhalten der Eigenschaft: - *erforderlich*

Telemetriedatenfilter

Das Objekt für den Telemetriedatenfilter verfügt über die folgenden Eigenschaften.

Property	Type	Description
conditions	array	Bedingungen des Telemetriedatenfilters, die ein Datensatz in einer Telemetriedatenspalte erfüllen muss. Verhalten der Eigenschaft: - *erforderlich*

Property	Type	Description
evaltype	integer	Auswertungsmethode der Filterbedingung. Mögliche Werte: 0 - Und/Oder; 1 - Und; 2 - Oder; 3 - Benutzerdefinierter Ausdruck. Verhalten der Eigenschaft: - *erforderlich*
eval_formula	string	Generierter Ausdruck, der zur Auswertung der Filterbedingungen verwendet wird. Der Ausdruck enthält IDs, die über ihre `formulaid` auf bestimmte Filterbedingungen verweisen. Der Wert von `eval_formula` entspricht dem Wert von `formula` für Filter mit einem benutzerdefinierten Ausdruck. Verhalten der Eigenschaft: - *schreibgeschützt*
formula	string	Benutzerdefinierter Ausdruck zur Auswertung der Bedingungen von Filtern mit einem benutzerdefinierten Ausdruck. Der Ausdruck muss IDs enthalten, die auf bestimmte Filterbedingungen über ihre `formulaid` verweisen. Die im Ausdruck verwendeten IDs müssen genau mit den in den Filterbedingungen definierten IDs übereinstimmen: Keine Bedingung darf ungenutzt bleiben oder ausgelassen werden. Verhalten der Eigenschaft: - *erforderlich*, wenn `evaltype` auf „Benutzerdefinierter Ausdruck“ gesetzt ist

Bedingungen für den Telemetriedatenfilter

Das Objekt für die Bedingungen des Telemetriedatenfilters verfügt über die folgenden Eigenschaften.

Property	Type	Description
column	string	Name der Telemetriedatenspalte, auf die die Filterbedingung angewendet wird. Siehe unterstützte Telemetriedatenspalten. Verhalten der Eigenschaft: - *erforderlich*
attribute_key	string	Name des Schlüssels, auf den die Bedingung innerhalb der Spalte angewendet wird, die Schlüssel-Wert-Paare enthält. Verhalten der Eigenschaft: - *erforderlich*, wenn `column` auf eine Spalte festgelegt ist, die Schlüssel-Wert-Paare enthält

Property	Type	Description
operator	integer	Operator der Filterbedingung. Mögliche Werte für Spalten mit Einzelwerten: 0 - Ist gleich; 1 - Ist nicht gleich; 2 - Enthält; 3 - Enthält nicht. Mögliche Werte für Spalten, die Schlüssel-Wert-Paare enthalten: 0 - Ist gleich; 1 - Ist nicht gleich; 12 - Ist vorhanden. Verhalten der Eigenschaft: - *erforderlich*
value	string	Wert, mit dem verglichen werden soll. Verhalten der Eigenschaft: - *erforderlich*, wenn `operator` auf „Ist gleich“, „Ist nicht gleich“, „Enthält“ oder „Enthält nicht“ festgelegt ist
formulaid	string	Beliebige eindeutige ID, die verwendet wird, um in einem benutzerdefinierten Ausdruck auf die Bedingung zu verweisen. Darf nur Großbuchstaben enthalten. Die ID muss vom Benutzer beim Ändern von Filterbedingungen definiert werden, wird jedoch bei einer anschließenden Abfrage neu generiert. Verhalten der Eigenschaft: - *erforderlich*, wenn `evaltype` des Objekts Telemetriedatenabfragefilter auf „Benutzerdefinierter Ausdruck“ festgelegt ist

Tag des Datenpunktprototyps

Das Tag-Objekt des Datenpunktprototyps hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
tag	string	Name des Tags des Datenpunktprototyps. Verhalten der Eigenschaft: - *erforderlich*
value	string	Wert des Tags des Datenpunktprototyps. Verhalten der Eigenschaft: - *schreibgeschützt*.
object	integer	Typ des Objekts, von dem das Tag geerbt wurde. Mögliche Werte: 0 - Vorlage; 1 - Host. Verhalten der Eigenschaft: - *schreibgeschützt*.
objectid	ID	ID des Objekts, von dem das Tag geerbt wurde. Verhalten der Eigenschaft: - *schreibgeschützt*.

Vorverarbeitung des Datenpunkt-Prototyps

Das Vorverarbeitungsobjekt des Datenpunkt-Prototyps verfügt über die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
type	integer	Der Typ der Vorverarbeitungsoption. Mögliche Werte: 1 - Benutzerdefinierter Multiplikator; 2 - Rechts trimmen; 3 - Links trimmen; 4 - Trimmen; 5 - Regulärer Ausdruck; 6 - Boolesch in Dezimalzahl; 7 - Oktal in Dezimalzahl; 8 - Hexadezimal in Dezimalzahl; 9 - Einfache Änderung; 10 - Änderung pro Sekunde; 11 - XML XPath; 12 - JSONPath; 13 - Im Bereich; 14 - Entspricht regulärem Ausdruck; 15 - Entspricht nicht regulärem Ausdruck; 16 - Auf Fehler in JSON prüfen; 17 - Auf Fehler in XML prüfen; 18 - Mit regulärem Ausdruck auf Fehler prüfen; 19 - Unveränderte Werte verwerfen; 20 - Unveränderte Werte mit Heartbeat verwerfen; 21 - JavaScript; 22 - Prometheus-Muster; 23 - Prometheus in JSON; 24 - CSV in JSON; 25 - Ersetzen; 26 - Nicht unterstützten Wert prüfen; 27 - XML in JSON; 28 - SNMP-Walk-Wert; 29 - SNMP-Walk in JSON; 30 - SNMP-Get-Wert. Verhalten der Eigenschaft: - *erforderlich*
params	string	Zusätzliche Parameter, die von der Vorverarbeitungsoption verwendet werden. Mehrere Parameter werden durch das Zeilenumbruchzeichen (\n) getrennt. Wenn `type` auf „Nicht unterstützten Wert prüfen“ gesetzt ist, folgen die Parameter der Syntax `<scope>[\n<pattern>]`, wobei *pattern* ein regulärer Ausdruck ist und *scope* einem der folgenden Werte entspricht: -1 - jeden Fehler abgleichen; 0 - prüfen, ob die Fehlermeldung mit *pattern* übereinstimmt; 1 - prüfen, ob die Fehlermeldung nicht mit *pattern* übereinstimmt. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „Benutzerdefinierter Multiplikator“ (1), „Rechts trimmen“ (2), „Links trimmen“ (3), „Trimmen“ (4), „Regulärer Ausdruck“ (5), „XML XPath“ (11), „JSONPath“ (12), „Im Bereich“ (13), „Entspricht regulärem Ausdruck“ (14), „Entspricht nicht regulärem Ausdruck“ (15), „Auf Fehler in JSON prüfen“ (16), „Auf Fehler in XML prüfen“ (17), „Mit regulärem Ausdruck auf Fehler prüfen“ (18), „Unveränderte Werte mit Heartbeat verwerfen“ (20), „JavaScript“ (21), „Prometheus-Muster“ (22), „Prometheus in JSON“ (23), „CSV in JSON“ (24), „Ersetzen“ (25), „Nicht unterstützten Wert prüfen“ (26), „SNMP-Walk-Wert“ (28), „SNMP-Walk in JSON“ (29) oder „SNMP-Get-Wert“ (30) gesetzt ist

Eigenschaft	Typ	Beschreibung
error_handler	integer	Aktionstyp, der bei einem Fehler in einem Vorverarbeitungsschritt verwendet wird. Mögliche Werte: 0 - Fehlermeldung wird vom Zabbix Server festgelegt; 1 - Wert verwerfen; 2 - Benutzerdefinierten Wert festlegen; 3 - Benutzerdefinierte Fehlermeldung festlegen. Mögliche Werte, wenn `type` auf „Nicht unterstützten Wert prüfen“ gesetzt ist: 1 - Wert verwerfen; 2 - Benutzerdefinierten Wert festlegen; 3 - Benutzerdefinierte Fehlermeldung festlegen. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „Benutzerdefinierter Multiplikator“ (1), „Regulärer Ausdruck“ (5), „Boolesch in Dezimalzahl“ (6), „Oktal in Dezimalzahl“ (7), „Hexadezimal in Dezimalzahl“ (8), „Einfache Änderung“ (9), „Änderung pro Sekunde“ (10), „XML XPath“ (11), „JSONPath“ (12), „Im Bereich“ (13), „Entspricht regulärem Ausdruck“ (14), „Entspricht nicht regulärem Ausdruck“ (15), „Auf Fehler in JSON prüfen“ (16), „Auf Fehler in XML prüfen“ (17), „Mit regulärem Ausdruck auf Fehler prüfen“ (18), „Prometheus-Muster“ (22), „Prometheus in JSON“ (23), „CSV in JSON“ (24), „Nicht unterstützten Wert prüfen“ (26), „XML in JSON“ (27), „SNMP-Walk-Wert“ (28), „SNMP-Walk in JSON“ (29) oder „SNMP-Get-Wert“ (30) gesetzt ist
error_handler_params	string	Parameter des Fehlerhandlers. Verhalten der Eigenschaft: - *erforderlich*, wenn `error_handler` auf „Benutzerdefinierten Wert festlegen“ oder „Benutzerdefinierte Fehlermeldung festlegen“ gesetzt ist

Die folgenden Parameter und Fehlerhandler werden für jeden Vorverarbeitungstyp unterstützt.

Vorverarbeitungstyp	Name	Parameter 1	Parameter 2	Parameter 3	Unterstützte Fehlerhandler
1	Benutzerdefinierter Multiplikator	definierter Multiplikator[1], 6^			0, 1, 2, 3
2	Rechts trimmen	Liste von Zeichen[2]			
3	Links trimmen	Liste von Zeichen[2]			
4	Trimmen	Liste von Zeichen[2]			
5	Regulärer Ausdruck	pattern[3]	output[2]		0, 1, 2, 3
6	Boolesch in Dezimalzahl				0, 1, 2, 3

Vorverarbeitungstyp	Name	Parameter 1	Parameter 2	Parameter 3	Unterstützte Fehlerhandler
7	Oktal in Dezimalzahl				0, 1, 2, 3
8	Hexadezimal in Dezimalzahl				0, 1, 2, 3
9	Einfache Änderung				0, 1, 2, 3
10	Änderung pro Sekunde				0, 1, 2, 3
11	XML XPath	path[4]			0, 1, 2, 3
12	JSONPath	path[4]			0, 1, 2, 3
13	Im Bereich	min^1, 6^	max^1, 6^		0, 1, 2, 3
14	Entspricht regulärem Ausdruck	pattern[3]			0, 1, 2, 3
15	Entspricht nicht regulärem Ausdruck	pattern[3]			0, 1, 2, 3
16	Auf Fehler in JSON prüfen	path[4]			0, 1, 2, 3
17	Auf Fehler in XML prüfen	path[4]			0, 1, 2, 3
18	Mit regulärem Ausdruck auf Fehler prüfen	pattern[3]	output[2]		0, 1, 2, 3
19	Unveränderte Werte verwerfen				
20	Unveränderte Werte mit Heartbeat verwerfen	seconds^5, 6^			

Vorverarbeitungstyp	Name	Parameter 1	Parameter 2	Parameter 3	Unterstützte Fehlerhandler
21	JavaScript	script[2]			
22	Prometheus-Muster	pattern^6, 7^	`value`, `label`, `function`	output^8, 9^	0, 1, 2, 3
23	Prometheus in JSON	pattern^6, 7^			0, 1, 2, 3
24	CSV in JSON	character[2]	character[2]	0,1	0, 1, 2, 3
25	Ersetzen	search string[2]	replacement[2]		
26	Nicht unterstützten Wert prüfen	scope[1]	pattern^3, 6^		1, 2, 3
27	XML in JSON				0, 1, 2, 3
28	SNMP-Walk-Wert	OID[2]	Format: 0 - Unverändert 1 - UTF-8 aus Hex-STRING 2 - MAC aus Hex-STRING 3 - Ganzzahl aus BITS		0, 1, 2, 3
29	SNMP-Walk in JSON[10]	Field name[2]	OID prefix[2]	Format: 0 - Unverändert 1 - UTF-8 aus Hex-STRING 2 - MAC aus Hex-STRING 3 - Ganzzahl aus BITS	0, 1, 2, 3
30	SNMP-Get-Wert	Format: 1 - UTF-8 aus Hex-STRING 2 - MAC aus Hex-STRING 3 - Ganzzahl aus BITS			0, 1, 2, 3

[1] Ganzzahl oder Gleitkommazahl
 [2] Zeichenkette
 [3] Regulärer Ausdruck
 [4] JSONPath oder XML XPath
 [5] Positive Ganzzahl (mit Unterstützung von Zeitsuffixen, z. B. `30s`, `1m`, `2h`, `1d`)
 [6] Benutzermakro, LLD-Makro
 [7] Prometheus-Muster gemäß der Syntax: `<metric name>{<label name>="<label value>", ...} == <value>`. Jede Komponente des Prometheus-Musters (Metrik, Labelname, Labelwert und Metrikwert) kann ein Benutzermakro oder LLD-Makro sein.
 [8] Prometheus-Ausgabe gemäß der Syntax: `<label name>` (kann ein Benutzermakro oder LLD-Makro sein), wenn `label` als zweiter Parameter ausgewählt ist.
 [9] Eine der Aggregationsfunktionen: `sum`, `min`, `max`, `avg`, `count`, wenn `function` als zweiter Parameter ausgewählt ist.
 [10] Unterstützt mehrere Datensätze im Format „Field name,OID prefix,Format“, die durch ein Zeilenumbruchzeichen getrennt sind.

datenpunktprototyp.create

Beschreibung

`object itemprototype.create(object/array itemPrototypes)`

Diese Methode ermöglicht es Ihnen, neue Datenpunktprototypen zu erstellen.

Note:
Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellender Datenpunkt-Prototyp.

Zusätzlich zu den Standard-Eigenschaften von Datenpunkt-Prototypen akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
ruleid	ID	ID der LLD-Regel, zu der der Datenpunkt gehört. Parameterverhalten: - *erforderlich*
preprocessing	array	Optionen für die Vorverarbeitung von Datenpunkt-Prototypen.
tags	array	Tags von Datenpunkt-Prototypen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Item-Prototypen unter der Eigenschaft `itemids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Item-Prototypen.

Beispiele

Erstellen eines Datenpunktprototyps

Erstellen Sie einen Datenpunktprototyp, um den freien Festplattenspeicher auf einem erkannten Dateisystem zu überwachen. Erkannte Datenpunkte sollten numerische Zabbix-Agent-Datenpunkte sein, die alle 30 Sekunden aktualisiert werden.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
        "name": "Freier Festplattenspeicher auf {#FSNAME}",
        "key_": "vfs.fs.size[{#FSNAME},free]",
        "hostid": "10197",
        "ruleid": "27665",
        "type": 0,
        "value_type": 3,
        "interfaceid": "112",
        "delay": "30s"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27666"
        ]
    },
    "id": 1
}
```

Erstellen eines Datenpunktprototyps mit Vorverarbeitung

Erstellen Sie einen Datenpunkt, der als zweiten Schritt „Änderung pro Sekunde“ und einen benutzerdefinierten Multiplikator verwendet.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
        "name": "Eingehender Netzwerkverkehr auf {#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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

Erstellen eines abhängigen Datenpunktprototyps

Erstellen Sie einen abhängigen Datenpunktprototyp für den Master-Datenpunktprototyp mit der ID 44211. Es sind nur Abhängigkeiten auf demselben Host (Vorlage/Discovery-Regel) zulässig, daher sollten Master- und abhängiger Datenpunkt dieselbe hostid und ruleid haben.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
      "hostid": "10001",
      "ruleid": "27665",
      "name": "Abhängiger Test-Datenpunktprototyp",
      "key_": "dependent.prototype",
      "type": 18,
      "master_itemid": "44211",
      "value_type": 3
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44212"
        ]
    },
    "id": 1
}
```

HTTP-Agent-Datenpunktprototyp erstellen

Erstellen Sie einen Datenpunktprototyp mit URL unter Verwendung eines Benutzermakros, Abfragefeldern und benutzerdefinierten Headern.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
        "type": "19",
        "hostid": "10254",
        "ruleid": "28256",
        "interfaceid": "2",
        "name": "api item prototype example",
        "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28305"
        ]
    },
    "id": 1
}
```

Skript-Datenpunktprototyp erstellen

Erstellen Sie eine einfache Datenerfassung mit einem Skript-Datenpunktprototyp.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Quelle

CItemPrototype::create() in *ui/include/classes/api/services/CItemPrototype.php*.

datenpunktprototyp.delete

Beschreibung

`object itemprototype.delete(array itemPrototypeIds)`

Diese Methode ermöglicht es Ihnen, Datenpunkt-Prototypen zu löschen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der zu löschenden Datenpunkt-Prototypen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der Prototypen der gelöschten Elemente unter der Eigenschaft `prototypeids` enthält.

Beispiele

Löschen mehrerer Datenpunkt-Prototypen

Löschen Sie zwei Datenpunkt-Prototypen.
 Abhängige Datenpunkt-Prototypen werden automatisch entfernt, wenn der Master-Datenpunkt oder Datenpunkt-Prototyp gelöscht wird.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.delete",
    "params": [
        "27352",
        "27356"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "prototypeids": [
            "27352",
            "27356"
        ]
    },
    "id": 1
}
```

Quelle

CItemPrototype::delete() in *ui/include/classes/api/services/CItemPrototype.php*.

datenpunktprototyp.get

Beschreibung

`integer/array itemprototype.get(object parameters)`

Die Methode ermöglicht es Ihnen, Datenpunktprototypen anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
discoveryids	ID/array	Gibt nur Datenpunkt-Prototypen zurück, die zu den angegebenen LLD-Regeln gehören.
graphids	ID/array	Gibt nur Datenpunkt-Prototypen zurück, die in den angegebenen Graph-Prototypen verwendet werden.
hostids	ID/array	Gibt nur Datenpunkt-Prototypen zurück, die zu den angegebenen Hosts gehören.
inherited	boolean	Wenn auf `true` gesetzt, werden nur Datenpunkt-Prototypen zurückgegeben, die von einer Vorlage geerbt wurden.
itemids	ID/array	Gibt nur Datenpunkt-Prototypen mit den angegebenen IDs zurück.
monitored	boolean	Wenn auf `true` gesetzt, werden nur aktivierte Datenpunkt-Prototypen zurückgegeben, die zu überwachten Hosts gehören.
templated	boolean	Wenn auf `true` gesetzt, werden nur Datenpunkt-Prototypen zurückgegeben, die zu Vorlagen gehören.
templateids	ID/array	Gibt nur Datenpunkt-Prototypen zurück, die zu den angegebenen Vorlagen gehören.
triggerids	ID/array	Gibt nur Datenpunkt-Prototypen zurück, die in den angegebenen Auslöser-Prototypen verwendet werden.

Parameter	Type	Beschreibung
selectDiscoveryData	query	Gibt eine `discoveryData`-Eigenschaft mit den Objektdaten der Datenpunkt-Prototyp-Erkennung zurück. Das Erkennungsobjekt des Datenpunkt-Prototyps verknüpft einen erkannten Datenpunkt-Prototyp mit einem Datenpunkt-Prototyp, aus dem er erkannt wurde. Es hat die folgenden Eigenschaften: `parent_itemid` - `(string)` ID des Datenpunkt-Prototyps, aus dem der Datenpunkt-Prototyp erstellt wurde; `key_` - `(string)` Schlüssel des Datenpunkt-Prototyps; `status` - `(int)` Erkennungsstatus des Datenpunkt-Prototyps: 0 - *(Standard)* Datenpunkt-Prototyp wurde erkannt, 1 - Datenpunkt-Prototyp wird nicht mehr erkannt; `ts_delete` - `(timestamp)` Zeitpunkt, zu dem ein Datenpunkt-Prototyp, der nicht mehr erkannt wird, gelöscht wird; `ts_disable` - `(timestamp)` Zeitpunkt, zu dem ein Datenpunkt-Prototyp, der nicht mehr erkannt wird, deaktiviert wird; `disable_source` - `(int)` Kennzeichen dafür, ob der Datenpunkt-Prototyp durch eine LLD-Regel oder manuell deaktiviert wurde: 0 - *(Standard)* automatisch deaktiviert, 1 - durch eine LLD-Regel deaktiviert.
selectDiscoveryRule	query	Gibt eine `discoveryRule`-Eigenschaft mit der Low-Level-Discovery-Regel zurück, zu der der Datenpunkt-Prototyp gehört.
selectDiscoveryRulePrototype	query	Gibt eine `discoveryRulePrototype`-Eigenschaft mit dem übergeordneten LLD-Regelprototyp zurück, zu dem der Datenpunkt-Prototyp gehört.
selectGraphs	query	Gibt eine `graphs`-Eigenschaft mit Graph-Prototypen zurück, in denen der Datenpunkt-Prototyp verwendet wird. Unterstützt `count`.
selectHosts	query	Gibt eine `hosts`-Eigenschaft mit einem Array von Hosts zurück, zu denen der Datenpunkt-Prototyp gehört.
selectInheritedTags	query	Gibt eine `inheritedTags`-Eigenschaft mit Tags zurück, die auf Vorlage/Host/verknüpften Vorlagen vorhanden sind.
selectTags	query	Gibt die Tags des Datenpunkt-Prototyps in der Eigenschaft `tags` zurück.
selectTriggers	query	Gibt eine `triggers`-Eigenschaft mit Auslöser-Prototypen zurück, in denen der Datenpunkt-Prototyp verwendet wird. Unterstützt `count`.
selectPreprocessing	query	Gibt eine `preprocessing`-Eigenschaft mit Vorverarbeitungsoptionen des Datenpunkt-Prototyps zurück.
selectValueMap	query	Gibt eine `valuemap`-Eigenschaft mit der Wertezuordnung des Datenpunkt-Prototyps zurück.
filter	object	Gibt nur die Ergebnisse zurück, die exakt dem angegebenen Filter entsprechen. Akzeptiert ein Objekt, bei dem die Schlüssel Eigenschaftsnamen sind und die Werte entweder ein einzelner Wert oder ein Array von Werten sind, mit denen abgeglichen werden soll. Unterstützt keine Eigenschaften vom `text`-data type. Unterstützt zusätzliche Eigenschaften: `host` - technischer Name des Hosts, zu dem der Datenpunkt-Prototyp gehört.

Parameter	Type	Beschreibung
limitSelects	integer	Begrenzt die Anzahl der von Unterabfragen zurückgegebenen Datensätze. Gilt für die folgenden Unterabfragen: `selectGraphs` - Ergebnisse werden nach `name` sortiert; `selectTriggers` - Ergebnisse werden nach `description` sortiert.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `itemid`, `name`, `key_`, `delay`, `type`, `status`, `history`, `trends`, `discovered`.
countOutput	boolean	Diese Parameter werden in der Referenzkommentierung beschrieben.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Kann die folgenden Dinge zurück geben:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Abrufen von Datenpunkt-Prototypen aus einer LLD-Regel

Rufen Sie alle Datenpunkt-Prototypen für eine bestimmte LLD-Regel-ID ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.get",
    "params": {
        "output": "extend",
        "discoveryids": "27426"
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```


Abhängigen Datenpunkt finden

Einen abhängigen Datenpunkt für eine bestimmte Datenpunkt-ID finden.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.get",
    "params": {
        "output": "extend",
        "filter": {
            "type": 18,
            "master_itemid": "25545"
        },
        "limit": "1"
    },
    "id": 1
}
```

Antwort:


```
{
    "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
}
```

HTTP-Agent-Datenpunktprototyp finden

HTTP-Agent-Datenpunktprototyp mit der Anfragemethode HEAD für eine bestimmte Host-ID finden.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.get",
    "params": {
        "hostids": "10254",
        "filter": {
            "type": 19,
            "request_method": 3
        }
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```


Siehe auch

- Host
- Graph-Prototyp
- Auslöser-Prototyp

Quelle

CItemPrototype::get() in *ui/include/classes/api/services/CItemPrototype.php*.

datenpunktprototyp.update

Beschreibung

`object itemprototype.update(object/array itemPrototypes)`

Diese Methode ermöglicht es Ihnen, vorhandene Datenpunktprototypen zu aktualisieren.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Berechtigungen zum Aufruf der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu aktualisierende Eigenschaften des Datenpunktprototyps.

Die Eigenschaft `itemid` muss für jeden Datenpunktprototyp definiert sein, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Zusätzlich zu den Standard-Eigenschaften des Datenpunktprototyps akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
preprocessing	array	Optionen für die Vorverarbeitung des Datenpunktprototyps, um die aktuellen Vorverarbeitungsoptionen zu ersetzen. Parameterverhalten: - *schreibgeschützt* für vererbte Objekte
tags	array	Tags des Datenpunktprototyps.

Rückgabewerte

(`object`) Gibt ein Objekt zurück, das die IDs der aktualisierten Item-Prototypen unter der Eigenschaft `itemids` enthält.

Beispiele

Ändern der Schnittstelle eines Datenpunktprototyps

Ändern Sie die Host-Schnittstelle, die von erkannten Datenpunkten verwendet wird.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid": "27428",
        "interfaceid": "132"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27428"
        ]
    },
    "id": 1
}
```

Abhängigen Datenpunkt-Prototyp aktualisieren

Aktualisieren Sie den abhängigen Datenpunkt-Prototyp mit einer neuen ID des Master-Datenpunkt-Prototyps. Es sind nur Abhängigkeiten auf demselben Host (Vorlage/Discovery-Regel) zulässig, daher sollten Master- und abhängiger Datenpunkt dieselbe hostid und ruleid haben.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "master_itemid": "25570",
        "itemid": "189030"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "189030"
        ]
    },
    "id": 1
}
```

HTTP-Agent-Datenpunktprototyp aktualisieren

Abfragefelder ändern und alle benutzerdefinierten Header entfernen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
```

```
    "params": {
        "itemid":"28305",
        "query_fields": [
            {
                "name": "random",
                "value": "qwertyuiopasdfghjklzxcvbnm"
            }
        ],
        "headers": []
    }
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28305"
        ]
    },
    "id": 1
}
```

Aktualisieren der Vorverarbeitungsoptionen eines Datenpunkts

Aktualisieren Sie einen Datenpunkt-Prototyp mit der Vorverarbeitungsregel „Benutzerdefinierter Multiplikator“.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid": "44211",
        "preprocessing": [
            {
                "type": 1,
                "params": "4",
                "error_handler": 2,
                "error_handler_params": "5"
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

Aktualisieren eines Skript-Datenpunktprototyps

Aktualisieren Sie einen Skript-Datenpunktprototyp mit einem anderen Skript und entfernen Sie unnötige Parameter, die vom vorherigen Skript verwendet wurden.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid": "23865",
        "parameters": [],
        "script": "Zabbix.log(3, 'Log test');\nreturn 1;"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Quelle

CItemPrototype::update() in *ui/include/classes/api/services/CItemPrototype.php*.

Discovery-Prüfung

Diese Klasse ist für die Arbeit mit Discovery-Prüfungen vorgesehen.

Objektreferenzen:

- Discovery-Prüfung

Verfügbare Methoden:

- dcheck.get - Discovery-Prüfungen abrufen

Objekt der Entdeckungskontrolle

Die folgenden Objekte stehen in direktem Zusammenhang mit der `dcheck` API.

Discovery-Prüfung

Das Objekt der Discovery-Prüfung definiert eine bestimmte Prüfung, die von einer Netzwerk-Discovery-Regel durchgeführt wird. Es hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
dcheckid	ID	ID der Discovery-Prüfung.
druleid	ID	ID der Discovery-Regel, zu der die Prüfung gehört.
key_	string	Datenpunktschlüssel (wenn `type` auf ”Zabbix agent” gesetzt ist) oder SNMP-OID (wenn `type` auf ”SNMPv1 agent”, ”SNMPv2 agent” oder ”SNMPv3 agent” gesetzt ist). Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf ”Zabbix agent”, ”SNMPv1 agent”, ”SNMPv2 agent” oder ”SNMPv3 agent” gesetzt ist

Eigenschaft	Typ	Beschreibung
ports	string	Ein oder mehrere zu prüfende Portbereiche, durch Kommas getrennt. Standard: 0. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”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) oder ”Telnet” (15) gesetzt ist
snmp_community	string	SNMP-Community. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf ”SNMPv1 agent” oder ”SNMPv2 agent” gesetzt ist
snmpv3_authpassphrase	string	Authentifizierungs-Passphrase. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”SNMPv3 agent” gesetzt ist und `snmpv3_securitylevel` auf ”authNoPriv” oder ”authPriv” gesetzt ist
snmpv3_authprotocol	integer	Authentifizierungsprotokoll. Mögliche Werte: 0 - *(Standard)* MD5; 1 - SHA1; 2 - SHA224; 3 - SHA256; 4 - SHA384; 5 - SHA512. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”SNMPv3 agent” gesetzt ist und `snmpv3_securitylevel` auf ”authNoPriv” oder ”authPriv” gesetzt ist
snmpv3_contextname	string	SNMPv3-Kontextname. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”SNMPv3 agent” gesetzt ist
snmpv3_privpassphrase	string	Privacy-Passphrase. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”SNMPv3 agent” gesetzt ist und `snmpv3_securitylevel` auf ”authPriv” gesetzt ist
snmpv3_privprotocol	integer	Privacy-Protokoll. Mögliche Werte: 0 - *(Standard)* DES; 1 - AES128; 2 - AES192; 3 - AES256; 4 - AES192C; 5 - AES256C. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”SNMPv3 agent” gesetzt ist und `snmpv3_securitylevel` auf ”authPriv” gesetzt ist

Eigenschaft	Typ	Beschreibung
snmpv3_securitylevel	string	Sicherheitsstufe. Mögliche Werte: 0 - noAuthNoPriv; 1 - authNoPriv; 2 - authPriv. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”SNMPv3 agent” gesetzt ist
snmpv3_securityname	string	Sicherheitsname. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”SNMPv3 agent” gesetzt ist
type	integer	Typ der Prüfung. Mögliche Werte: 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. Verhalten der Eigenschaft: - *erforderlich*
uniq	integer	Gibt an, ob diese Prüfung als Kriterium für die Eindeutigkeit eines Geräts verwendet werden soll. Für eine Discovery-Regel kann nur eine einzige eindeutige Prüfung konfiguriert werden. Mögliche Werte: 0 - *(Standard)* diese Prüfung nicht als Eindeutigkeitskriterium verwenden; 1 - diese Prüfung als Eindeutigkeitskriterium verwenden. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf ”Zabbix agent”, ”SNMPv1 agent”, ”SNMPv2 agent” oder ”SNMPv3 agent” gesetzt ist
host_source	integer	Quelle für den Hostnamen. Mögliche Werte: 1 - *(Standard)* DNS; 2 - IP; 3 - Discovery-Wert dieser Prüfung.
name_source	integer	Quelle für den sichtbaren Namen. Mögliche Werte: 0 - *(Standard)* nicht angegeben; 1 - DNS; 2 - IP; 3 - Discovery-Wert dieser Prüfung.

Eigenschaft	Typ	Beschreibung
allow_redirect	integer	Eine Situation zulassen, in der das per ICMP-Ping angesprochene Ziel von einer anderen IP-Adresse antwortet. Mögliche Werte: 0 - *(Standard)* umgeleitete Antworten so behandeln, als wäre der Ziel-Host nicht erreichbar (Fehler); 1 - umgeleitete Antworten so behandeln, als wäre der Ziel-Host erreichbar (Erfolg). Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf "ICMP ping" gesetzt ist

dcheck.get

Beschreibung

`integer/array dcheck.get(object parameters)`

Die Methode ermöglicht es Ihnen, Discovery-Checks anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
dcheckids	ID/array	Gibt nur Discovery-Checks mit den angegebenen IDs zurück.
druleids	ID/array	Gibt nur Discovery-Checks zurück, die zu den angegebenen Discovery-Regeln gehören.
dserviceids	ID/array	Gibt nur Discovery-Checks zurück, die die angegebenen erkannten Services erkannt haben.
selectDRules	query	Gibt Discovery-Regeln zurück, die mit den Discovery-Checks verknüpft sind.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `dcheckid`, `druleid`.
countOutput	boolean	Diese Parameter werden im Referenzkommentar beschrieben.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Kann die folgenden Dinge zurück geben:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Discovery-Prüfungen für eine Discovery-Regel abrufen

Rufen Sie alle Discovery-Prüfungen ab, die von der Discovery-Regel „6“ verwendet werden.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "dcheck.get",
    "params": {
        "output": "extend",
        "dcheckids": "6"
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```

Quelle

CDCheck::get() in *ui/include/classes/api/services/CDCheck.php*.

Discovery-Regel

Diese Klasse ist für die Arbeit mit Regeln zur Netzwerkdiscovery konzipiert.

> **Note:**
> Diese API ist für die Arbeit mit Regeln zur Netzwerkdiscovery vorgesehen. Für Low-Level-Discovery-Regeln siehe LLD-Regel-API.

Objektreferenzen:

- Discovery-Regel

Verfügbare Methoden:

- drule.create - neue Regeln zur Netzwerkdiscovery erstellen
- drule.delete - Regeln zur Netzwerkdiscovery löschen
- drule.get - Regeln zur Netzwerkdiscovery abrufen

- drule.update - Regeln zur Netzwerkdiscovery aktualisieren

Objekt der Entdeckungsregel

Die folgenden Objekte stehen in direktem Zusammenhang mit der `drule` API.

Discovery-Regel

Das Discovery-Regel-Objekt definiert eine Netzwerk-Discovery-Regel. Es verfügt über die folgenden Eigenschaften.

Property	Type	Description
druleid	ID	ID der Discovery-Regel. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
iprange	string	Ein oder mehrere zu überprüfende IP-Bereiche, durch Kommas getrennt. Weitere Informationen zu den unterstützten Formaten von IP-Bereichen finden Sie im Abschnitt Konfiguration der Netzwerk-Discovery. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
name	string	Name der Discovery-Regel. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
delay	string	Ausführungsintervall der Discovery-Regel. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300), einen Wert mit einem Zeit-Suffix (z. B. 30s, 1m, 2h, 1d) oder ein Benutzermakro. Standard: 1h.
proxyid	ID	ID des für die Discovery verwendeten Proxy.
status	integer	Gibt an, ob die Discovery-Regel aktiviert ist. Mögliche Werte: 0 - *(Standard)* aktiviert; 1 - deaktiviert.
concurrency_max	integer	Maximale Anzahl gleichzeitiger Prüfungen pro Discovery-Regel. Mögliche Werte: 0 - *(Standard)* unbegrenzte Anzahl von Prüfungen; 1 - eine Prüfung; 2-999 - benutzerdefinierte Anzahl von Prüfungen.
error	string	Fehlertext, wenn bei der Ausführung der Discovery-Regel Probleme aufgetreten sind. Verhalten der Eigenschaft: - *schreibgeschützt*

drule.create

Beschreibung

`object drule.create(object/array discoveryRules)`

Diese Methode ermöglicht es Ihnen, neue Erkennungsregeln zu erstellen.

Note:
Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellende Discovery-Regeln.

Zusätzlich zu den Standard-Discovery-Regel- Eigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
dchecks	array	Discovery-Prüfungen, die für die Discovery-Regel erstellt werden sollen. Parameterverhalten: - *erforderlich*

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Entdeckungsregeln unter der Eigenschaft `druleids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Entdeckungsregeln.

Beispiele

Eine Discovery-Regel erstellen

Erstellen Sie eine Discovery-Regel, um Rechner zu finden, auf denen der Zabbix Agent im lokalen Netzwerk ausgeführt wird. Die Regel muss eine einzelne Zabbix-Agent-Prüfung auf Port 10050 verwenden.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "druleids": [
            "6"
        ]
    },
    "id": 1
}
```

Siehe auch

- Discovery-Prüfung

Quelle

CDRule::create() in *ui/include/classes/api/services/CDRule.php*.

drule.delete

Beschreibung

`object drule.delete(array discoveryRuleIds)`

Diese Methode ermöglicht es Ihnen, Erkennungsregeln zu löschen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der zu löschenden Discovery-Regeln.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Discovery-Regeln unter der Eigenschaft `druleids` enthält.

Beispiele

Mehrere Discovery-Regeln löschen

Löschen Sie zwei Discovery-Regeln.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "drule.delete",
    "params": [
        "4",
        "6"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "druleids": [
            "4",
            "6"
        ]
    },
    "id": 1
}
```

Quelle

CDRule::delete() in *ui/include/classes/api/services/CDRule.php*.

drule.get

Beschreibung

`integer/array drule.get(object parameters)`

Die Methode ermöglicht es Ihnen, Discovery-Regeln anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
dhostids	ID/array	Gibt nur Discovery-Regeln zurück, die die angegebenen entdeckten Hosts erstellt haben.
druleids	ID/array	Gibt nur Discovery-Regeln mit den angegebenen IDs zurück.
dserviceids	ID/array	Gibt nur Discovery-Regeln zurück, die die angegebenen entdeckten Services erstellt haben.
selectDChecks	query	Gibt eine Eigenschaft `dchecks` mit den von der Discovery-Regel verwendeten Discovery-Prüfungen zurück. Unterstützt `count`.
selectDHosts	query	Gibt eine Eigenschaft `dhosts` mit den von der Discovery-Regel erstellten entdeckten Hosts zurück. Unterstützt `count`.
limitSelects	integer	Begrenzt die Anzahl der von Unterabfragen zurückgegebenen Datensätze. Gilt für die folgenden Unterabfragen: `selectDChecks` - Ergebnisse werden nach `dcheckid` sortiert; `selectDHosts` - Ergebnisse werden nach `dhostsid` sortiert.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `druleid`, `name`.
countOutput	boolean	Diese Parameter sind im Referenzkommentar beschrieben.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Kann die folgenden Dinge zurück geben:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Alle Discovery-Regeln abrufen

Rufen Sie alle konfigurierten Discovery-Regeln und die von ihnen verwendeten Discovery-Checks ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "drule.get",
    "params": {
        "output": "extend",
        "selectDChecks": "extend"
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```


Siehe auch

- Erkannter Host
- Discovery-Prüfung

Quelle

CDRule::get() in *ui/include/classes/api/services/CDRule.php*.

drule.update

Beschreibung

`object drule.update(object/array discoveryRules)`

Diese Methode ermöglicht es Ihnen, vorhandene Discovery-Regeln zu aktualisieren.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Berechtigungen zum Aufruf der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu aktualisierende Eigenschaften von Discovery-Regeln.

Die Eigenschaft `druleid` muss für jede Discovery-Regel definiert werden, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Zusätzlich zu den Standard-Eigenschaften von Discovery-Regeln akzeptiert die Methode die folgenden Parameter.

Parameter	Typ	Beschreibung
dchecks	array	Discovery-Prüfungen zum Ersetzen vorhandener Prüfungen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Entdeckungsregeln unter der Eigenschaft `druleids` enthält.

Beispiele

Ändern Sie den IP-Bereich einer Discovery-Regel

Ändern Sie den IP-Bereich einer Discovery-Regel in „192.168.2.1-255“.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "drule.update",
    "params": {
        "druleid": "6",
        "iprange": "192.168.2.1-255"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "druleids": [
            "6"
        ]
    },
    "id": 1
}
```

Siehe auch

- Discovery-Prüfung

Quelle

CDRule::update() in *ui/include/classes/api/services/CDRule.php*.

Einstellungen

Diese Klasse ist für die Arbeit mit allgemeinen Administrationseinstellungen vorgesehen.

Objektreferenzen:

- Settings

Verfügbare Methoden:

- settings.get - Einstellungen abrufen
- settings.update - Einstellungen aktualisieren

Einstellungs-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `settings` API.

Einstellungen

Das Einstellungsobjekt verfügt über die folgenden Eigenschaften.

Property	Type	Description
default_lang	string	Standardsystemsprache. Standard: `en_US`.
default_timezone	string	Systemzeitzone als Standard. Standard: `system` - Systemstandard. Eine vollständige Liste der unterstützten Zeitzonen finden Sie in der PHP-Dokumentation.

Property	Type	Description
default_theme	string	Standarddesign. Mögliche Werte: `blue-theme` - *(Standard)* Blau; `blue-classic-theme` - Blau (klassisch); `dark-theme` - Dunkel; `dark-blue-theme` - Dunkelblau; `dark-classic-theme` - Dunkel (klassisch); `hc-light` - Hell mit hohem Kontrast; `hc-dark` - Dunkel mit hohem Kontrast.
search_limit	integer	Limit für Such- und Filterergebnisse. Standard: 1000.
max_overview_table_size	integer	Maximale Anzahl von Spalten und Zeilen in den Dashboard-Widgets „Datenübersicht“ und „Auslöserübersicht“. Standard: 50.
max_in_table	integer	Maximale Anzahl von Elementen, die innerhalb einer Tabellenzelle angezeigt werden. Standard: 50.
server_check_interval	integer	Warnung anzeigen, wenn der Zabbix Server nicht verfügbar ist. Mögliche Werte: 0 - Keine Warnung anzeigen; 10 - *(Standard)* Warnung anzeigen.
work_period	string	Arbeitszeit. Standard: 1-5,09:00-18:00.
show_technical_errors	integer	Technische Fehler (PHP/SQL) Benutzern ohne Super-Admin-Rechte sowie Benutzern anzeigen, die keiner Benutzergruppe mit aktiviertem Debug-Modus angehören. Mögliche Werte: 0 - *(Standard)* Keine technischen Fehler anzeigen; 1 - Technische Fehler anzeigen.
history_period	string	Maximaler Zeitraum für die Anzeige von Verlaufsdaten in den Dashboard-Widgets „Neueste Daten“, „Web“ und „Datenübersicht“. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300) oder einen Wert mit einem Zeitsuffix (z. B. 30s, 1m, 2h, 1d). Standard: 24h.
period_default	string	Standardzeitraum des Zeitfilters. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300) oder einen Wert mit einem Zeitsuffix (z. B. 30s, 1m, 2h, 1d, 1M, 1y), einschließlich der Unterstützung für Monate und Jahre. Standard: 1h.
max_period	string	Maximaler Zeitraum für den Zeitfilter. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300) oder einen Wert mit einem Zeitsuffix (z. B. 30s, 1m, 2h, 1d, 1M, 1y), einschließlich der Unterstützung für Monate und Jahre. Standard: 2y.
severity_color_0	string	Farbe für den Schweregrad „Nicht klassifiziert“ als hexadezimaler Farbcode. Standard: 97AAB3.
severity_color_1	string	Farbe für den Schweregrad „Information“ als hexadezimaler Farbcode. Standard: 7499FF.

Property	Type	Description
severity_color_2	string	Farbe für den Schweregrad „Warnung“ als hexadezimaler Farbcode. Standard: FFC859.
severity_color_3	string	Farbe für den Schweregrad „Durchschnittlich“ als hexadezimaler Farbcode. Standard: FFA059.
severity_color_4	string	Farbe für den Schweregrad „Hoch“ als hexadezimaler Farbcode. Standard: E97659.
severity_color_5	string	Farbe für den Schweregrad „Katastrophe“ als hexadezimaler Farbcode. Standard: E45959.
severity_name_0	string	Name für den Schweregrad „Nicht klassifiziert“. Standard: Nicht klassifiziert.
severity_name_1	string	Name für den Schweregrad „Information“. Standard: Information.
severity_name_2	string	Name für den Schweregrad „Warnung“. Standard: Warnung.
severity_name_3	string	Name für den Schweregrad „Durchschnittlich“. Standard: Durchschnittlich.
severity_name_4	string	Name für den Schweregrad „Hoch“. Standard: Hoch.
severity_name_5	string	Name für den Schweregrad „Katastrophe“. Standard: Katastrophe.
custom_color	integer	Benutzerdefinierte Farben für Ereignisstatus verwenden. Mögliche Werte: 0 - *(Standard)* Keine benutzerdefinierten Farben für Ereignisstatus verwenden; 1 - Benutzerdefinierte Farben für Ereignisstatus verwenden.
ok_period	string	Zeitraum für die Anzeige von OK-Auslösern. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300) oder einen Wert mit einem Zeitsuffix (z. B. 30s, 1m, 2h). Standard: 5m.
blink_period	string	Blinkdauer bei einer Änderung des Auslöserstatus. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300) oder einen Wert mit einem Zeitsuffix (z. B. 30s, 1m, 2h). Standard: 2m.
problem_unack_color	string	Farbe für nicht bestätigte PROBLEM-Ereignisse als hexadezimaler Farbcode. Standard: CC0000.
problem_ack_color	string	Farbe für bestätigte PROBLEM-Ereignisse als hexadezimaler Farbcode. Standard: CC0000.
ok_unack_color	string	Farbe für nicht bestätigte RESOLVED-Ereignisse als hexadezimaler Farbcode. Standard: 009900.
ok_ack_color	string	Farbe für bestätigte RESOLVED-Ereignisse als hexadezimaler Farbcode. Standard: 009900.

Property	Type	Description
problem_unack_style	integer	Blinken für nicht bestätigte PROBLEM-Ereignisse. Mögliche Werte: 0 - Kein Blinken anzeigen; 1 - *(Standard)* Blinken anzeigen.
problem_ack_style	integer	Blinken für bestätigte PROBLEM-Ereignisse. Mögliche Werte: 0 - Kein Blinken anzeigen; 1 - *(Standard)* Blinken anzeigen.
ok_unack_style	integer	Blinken für nicht bestätigte RESOLVED-Ereignisse. Mögliche Werte: 0 - Kein Blinken anzeigen; 1 - *(Standard)* Blinken anzeigen.
ok_ack_style	integer	Blinken für bestätigte RESOLVED-Ereignisse. Mögliche Werte: 0 - Kein Blinken anzeigen; 1 - *(Standard)* Blinken anzeigen.
url	string	Frontend-URL.
discovery_groupid	ID	ID der Hostgruppe, in der erkannte Hosts automatisch platziert werden.
default_inventory_mode	integer	Standardmodus für das Hostinventar. Mögliche Werte: -1 - *(Standard)* Deaktiviert; 0 - Manuell; 1 - Automatisch.
alert_usrgrpid	ID	ID der Benutzergruppe, an die die Alarmmeldung bei einem Ausfall der Datenbank gesendet wird. Bei „0“ wird die Alarmmeldung nicht gesendet.
snmptrap_logging	integer	Nicht zugeordnete SNMP-Traps protokollieren. Mögliche Werte: 0 - Nicht zugeordnete SNMP-Traps nicht protokollieren; 1 - *(Standard)* Nicht zugeordnete SNMP-Traps protokollieren.
login_attempts	integer	Anzahl fehlgeschlagener Anmeldeversuche, nach der das Anmeldeformular gesperrt wird. Standard: 5.
login_block	string	Zeitraum, während dessen das Anmeldeformular gesperrt wird, wenn die Anzahl der fehlgeschlagenen Anmeldeversuche den im Feld login_attempts festgelegten Wert überschreitet. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60) oder einen Wert mit einem Zeitsuffix (z. B. 30s, 1m). Standard: 30s.
validate_uri_schemes	integer	URI-Schemata validieren. Mögliche Werte: 0 - Nicht validieren; 1 - *(Standard)* Validieren.
uri_valid_schemes	string	Gültige URI-Schemata. Standard: http,https,ftp,file,mailto,tel,ssh.
x_frame_options	string	X-Frame-Options-HTTP-Header. Standard: SAMEORIGIN.

Property	Type	Description
iframe_sandboxing_enabled	integer	Iframe-Sandboxing verwenden. Mögliche Werte: 0 - Nicht verwenden; 1 - *(Standard)* Verwenden.
iframe_sandboxing_exceptions	string	Ausnahmen für Iframe-Sandboxing.
connect_timeout	string	Verbindungs-Timeout mit dem Zabbix Server. Möglicher Wertebereich: 1-30s. Standard: 3s. Eigenschaftsverhalten: - *required*
socket_timeout	string	Standard-Netzwerk-Timeout. Möglicher Wertebereich: 1-300s. Standard: 3s. Eigenschaftsverhalten: - *required*
media_type_test_timeout	string	Netzwerk-Timeout für den Medientyp-Test. Möglicher Wertebereich: 1-300s. Standard: 65s. Eigenschaftsverhalten: - *required*
item_test_timeout	string	Netzwerk-Timeout für Datenpunkttests. Möglicher Wertebereich: 1-600s. Standard: 60s. Eigenschaftsverhalten: - *required*
script_timeout	string	Netzwerk-Timeout für die Skriptausführung. Möglicher Wertebereich: 1-300s. Standard: 60s. Eigenschaftsverhalten: - *required*
report_test_timeout	string	Netzwerk-Timeout für den Test eines geplanten Berichts. Möglicher Wertebereich: 1-300s. Standard: 60s. Eigenschaftsverhalten: - *required*

Property	Type	Description
device_link_timeout	string	Timeout für das Hinzufügen eines Mobilgeräts zur Verwendung mit der Zabbix-Mobile-App; gilt für die Kommunikation zwischen der Zabbix-API und dem Bridge-Adapter. Möglicher Wertebereich: 1-300s. Standard: 60s. Eigenschaftsverhalten: - *required*
auditlog_enabled	integer	Festlegen, ob die Auditprotokollierung aktiviert werden soll. Mögliche Werte: 0 - Deaktivieren; 1 - *(Standard)* Aktivieren.
auditlog_mode	integer	Festlegen, ob die Auditprotokollierung von Aktivitäten der Low-Level-Discovery, Netzwerkerkennung und automatischen Registrierung aktiviert werden soll, die vom Server (Systembenutzer) ausgeführt werden. Mögliche Werte: 0 - Deaktivieren; 1 - *(Standard)* Aktivieren.
ha_failover_delay	string	Failover-Verzögerung in Sekunden. Standard: 1m. Eigenschaftsverhalten: - *read-only*
geomaps_tile_provider	string	Kartenkachel-Anbieter. Mögliche Werte: `OpenStreetMap.Mapnik` - *(Standard)* OpenStreetMap Mapnik; `OpenTopoMap` - OpenTopoMap; `Stamen.TonerLite` - Stamen Toner Lite; `Stamen.Terrain` - Stamen Terrain; `USGS.USTopo` - USGS US Topo; `USGS.USImagery` - USGS US Imagery. Unterstützt eine leere Zeichenfolge zur Angabe benutzerdefinierter Werte für `geomaps_tile_url`, `geomaps_max_zoom` und `geomaps_attribution`.
geomaps_tile_url	string	URL der Kartenkacheln. Eigenschaftsverhalten: - *supported*, wenn `geomaps_tile_provider` auf eine leere Zeichenfolge gesetzt ist
geomaps_max_zoom	integer	Maximale Zoomstufe der Karte. Mögliche Werte: 0-30. Eigenschaftsverhalten: - *supported*, wenn `geomaps_tile_provider` auf eine leere Zeichenfolge gesetzt ist
geomaps_attribution	string	Zuordnungstext der Karte. Eigenschaftsverhalten: - *supported*, wenn `geomaps_tile_provider` auf eine leere Zeichenfolge gesetzt ist

Property	Type	Description
vault_provider	integer	Vault-Anbieter. Mögliche Werte: 0 - *(Standard)* HashiCorp Vault; 1 - CyberArk Vault.
proxy_secrets_provider	integer	Geheime Makrowerte auflösen durch: Mögliche Werte: 0 - *(Standard)* nur den Server; 1 - Server und Proxys unabhängig voneinander.
timeout_zabbix_agent	string	Nicht mehr als `timeout_*` Sekunden für die Verarbeitung verwenden. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300) oder einen Wert mit einem Zeitsuffix (z. B. 30s, 1m). Akzeptiert auch Benutzermakros. Möglicher Wertebereich: 1-600s. Standard: 3s. Standard für `timeout_browser`: 60s. Eigenschaftsverhalten: - *required*
timeout_simple_check		
timeout_snmp_agent		
timeout_external_check		
timeout_db_monitor		
timeout_http_agent		
timeout_ssh_agent		
timeout_telnet_agent		
timeout_script		
timeout_browser		
timeout_telemetry_query		
apm_global_db	object	Konfiguration der globalen Datenquelle.

Globale Datenquelle

Das globale Datenquellenobjekt verfügt über die folgenden Eigenschaften:

Property	Type	Description
status	integer	Status der globalen Datenquelle. Mögliche Werte: 0 - *(default)* nicht konfiguriert; 1 - konfiguriert.
url	string	URL, die zum Herstellen der Verbindung mit der Datenquelle verwendet wird. Property behavior: - *required*, wenn der Benutzer `status` auf 1 (konfiguriert) ändert.
authentication_type	integer	Authentifizierungstyp. Mögliche Werte: 0 - *(default)* Verwendung eines Benutzernamens und Passworts; 1 - keine Authentifizierung verwendet. Property behavior: - *required*, wenn der Benutzer `status` auf 1 (konfiguriert) ändert.

Property	Type	Description
username	string	Benutzername, der zur Anmeldung bei der Datenquelle verwendet wird. Property behavior: - *supported*, wenn `status` auf 1 (konfiguriert) und `authentication_type` auf 0 (Benutzername und Passwort) gesetzt ist. - *required*, wenn der Benutzer `authentication_type` auf 0 (Benutzername und Passwort) ändert.
password	string	Passwort, das zur Anmeldung bei der Datenquelle verwendet wird. Der Wert dieses Parameters kann leer sein. Property behavior: - *supported*, wenn `status` auf 1 (konfiguriert) und `authentication_type` auf 0 (Benutzername und Passwort) gesetzt ist.
db	string	Datenbankname. Der Wert dieses Parameters kann leer sein. Property behavior: - *supported*, wenn `status` auf 1 (konfiguriert) gesetzt ist.
ssl_verify_peer	integer	Serverzertifikat überprüfen. Mögliche Werte: 0 - *(default)* deaktiviert; 1 - aktiviert. Property behavior: - *supported*, wenn `status` auf 1 (konfiguriert) gesetzt ist und `url` das Schema `https://` verwendet.
ssl_verify_host	integer	Host überprüfen. Mögliche Werte: 0 - deaktiviert; 1 - *(default)* aktiviert. Property behavior: - *supported*, wenn `status` auf 1 (konfiguriert) gesetzt ist und `url` das Schema `https://` verwendet.

settings.get

Beschreibung

`object settings.get(object parameters)`

Die Methode ermöglicht es Ihnen, ein Einstellungen-Objekt anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt nur einen Parameter.

Parameter	Type	Beschreibung
output	query	Dieser Parameter wird im Referenzkommentar beschrieben.

Rückgabewerte

`(object)` Gibt das Einstellungsobjekt zurück.

Beispiele

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "settings.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "alert_usrgrpid": "7",
        "apm_global_db": {
            "status": 1,
            "url": "https://localhost:8443",
            "authentication_type": 0,
            "username": "zabbix",
            "db": "",
            "ssl_verify_peer": 1,
            "ssl_verify_host": 1
        },
        "auditlog_enabled": "1",
        "auditlog_mode": "1",
        "blink_period": "2m",
        "connect_timeout": "3s",
        "custom_color": "0",
        "default_inventory_mode": "-1",
        "default_lang": "en_US",
        "default_theme": "blue-theme",
        "default_timezone": "system",
        "device_link_timeout": "60s",
        "discovery_groupid": "5",
        "geomaps_attribution": "",
        "geomaps_max_zoom": "0",
        "geomaps_tile_provider": "OpenStreetMap.Mapnik",
        "geomaps_tile_url": "",
        "ha_failover_delay": "1m",
        "history_period": "24h",
        "iframe_sandboxing_enabled": "1",
        "iframe_sandboxing_exceptions": "",
        "item_test_timeout": "60s",
        "login_attempts": "5",
        "login_block": "30s",
        "max_in_table": "50",
        "max_overview_table_size": "50",
        "max_period": "2y",
        "media_type_test_timeout": "65s",
        "ok_ack_color": "009900",
        "ok_ack_style": "1",
        "ok_period": "5m",
```

```
        "ok_unack_color": "009900",
        "ok_unack_style": "1",
        "period_default": "1h",
        "problem_ack_color": "CC0000",
        "problem_ack_style": "1",
        "problem_unack_color": "CC0000",
        "problem_unack_style": "1",
        "proxy_secrets_provider": "0",
        "report_test_timeout": "60s",
        "script_timeout": "60s",
        "search_limit": "1000",
        "server_check_interval": "10",
        "serverid": "019f4707-1a3e-70e6-aca2-286a2b696358",
        "severity_color_0": "97AAB3",
        "severity_color_1": "7499FF",
        "severity_color_2": "FFC859",
        "severity_color_3": "FFA059",
        "severity_color_4": "E97659",
        "severity_color_5": "E45959",
        "severity_name_0": "Nicht klassifiziert",
        "severity_name_1": "Information",
        "severity_name_2": "Warnung",
        "severity_name_3": "Durchschnittlich",
        "severity_name_4": "Hoch",
        "severity_name_5": "Katastrophe",
        "show_technical_errors": "0",
        "snmptrap_logging": "1",
        "socket_timeout": "3s",
        "timeout_browser": "60s",
        "timeout_db_monitor": "3s",
        "timeout_external_check": "3s",
        "timeout_http_agent": "3s",
        "timeout_script": "3s",
        "timeout_simple_check": "3s",
        "timeout_snmp_agent": "3s",
        "timeout_ssh_agent": "3s",
        "timeout_telemetry_query": "3s",
        "timeout_telnet_agent": "3s",
        "timeout_zabbix_agent": "3s",
        "uri_valid_schemes": "http,https,ftp,file,mailto,tel,ssh",
        "url": "",
        "validate_uri_schemes": "1",
        "vault_provider": "0",
        "work_period": "1-5,09:00-18:00",
        "x_frame_options": "SAMEORIGIN"
    },
    "id": 1
}
```


Quelle

CSettings::get() in *ui/include/classes/api/services/CSettings.php*.

settings.update

Beschreibung

`object settings.update(object settings)`

Mit dieser Methode können Sie vorhandene allgemeine Einstellungen aktualisieren.

Note:
Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` zu aktualisierende Eigenschaften der Einstellungen.

Rückgabewerte

`(array)` Gibt ein Array mit den Namen der aktualisierten Parameter zurück.

Beispiele

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "settings.update",
    "params": {
        "login_attempts": "1",
        "login_block": "1m"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        "login_attempts",
        "login_block"
    ],
    "id": 1
}
```

Quelle

CSettings::update() in *ui/include/classes/api/services/CSettings.php*.

Entdeckter Host

Diese Klasse ist für die Arbeit mit entdeckten Hosts vorgesehen.

Objektreferenzen:

- Entdeckter Host

Verfügbare Methoden:

- dhost.get - entdeckte Hosts abrufen

Entdecktes Host-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `dhost` API.

Entdeckter Host

Note:
Entdeckte Hosts werden vom Zabbix Server erstellt und können nicht über die API geändert werden.

Das Objekt des entdeckten Hosts enthält Informationen über einen Host, der durch eine Netzwerk-Erkennungsregel entdeckt wurde. Es hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
dhostid	ID	ID des entdeckten Hosts.

Eigenschaft	Typ	Beschreibung
druleid	ID	ID der Erkennungsregel, die den Host erkannt hat.
lastdown	timestamp	Zeitpunkt, zu dem der entdeckte Host zuletzt ausgefallen ist.
lastup	timestamp	Zeitpunkt, zu dem der entdeckte Host zuletzt verfügbar war.
status	integer	Gibt an, ob der entdeckte Host verfügbar oder nicht verfügbar ist. Ein Host ist verfügbar, wenn er mindestens einen aktiven entdeckten Dienst hat. Mögliche Werte: 0 - Host verfügbar; 1 - Host nicht verfügbar.

dhost.get

Beschreibung

`integer/array dhost.get(object parameters)`

Die Methode ermöglicht es Ihnen, entdeckte Hosts anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
dhostids	ID/array	Gibt nur entdeckte Hosts mit den angegebenen IDs zurück.
druleids	ID/array	Gibt nur entdeckte Hosts zurück, die durch die angegebenen Discovery-Regeln erstellt wurden.
dserviceids	ID/array	Gibt nur entdeckte Hosts zurück, auf denen die angegebenen Services ausgeführt werden.
selectDRules	query	Gibt eine Eigenschaft `drules` mit einem Array der Discovery-Regeln zurück, die den Host erkannt haben.
selectDServices	query	Gibt eine Eigenschaft `dservices` mit den auf dem Host laufenden entdeckten Services zurück. Unterstützt `count`.
limitSelects	integer	Begrenzt die Anzahl der von Unterabfragen zurückgegebenen Datensätze. Gilt für die folgenden Unterabfragen: `selectDServices` - die Ergebnisse werden nach `dserviceid` sortiert.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `dhostid`, `druleid`.
countOutput	boolean	Diese Parameter sind in der Referenzkommentierung beschrieben.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	

Parameter	Type	Beschreibung
startSearch	boolean	

Rückgabewerte

`(integer/array)` Kann die folgenden Dinge zurück geben:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Durch Suchlaufregel ermittelte Hosts abrufen

Rufen Sie alle Hosts und die ermittelten Dienste ab, die auf ihnen ausgeführt werden und durch die Suchlaufregel „4“ erkannt wurden.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "dhost.get",
    "params": {
        "output": "extend",
        "selectDServices": "extend",
        "druleids": "4"
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```

Siehe auch

- Erkannter Dienst
- Discovery-Regel

Quelle

CDHost::get() in *ui/include/classes/api/services/CDHost.php*.

Ereignis

Diese Klasse ist für die Arbeit mit Ereignissen vorgesehen.

Objektreferenzen:

- Event
- Event-Tag
- Medientyp-URL

Verfügbare Methoden:

- event.get - Ereignisse abrufen
- event.acknowledge - Ereignisse bestätigen

Ereignis-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `event` API.

Ereignis

> **Note:**
> Ereignisse werden vom Zabbix-Server erstellt und können nicht über die API geändert werden.

Das Ereignisobjekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
eventid	ID	ID des Ereignisses.
source	integer	Typ des Ereignisses. Mögliche Werte: 0 - Ereignis, das von einem Auslöser erstellt wurde; 1 - Ereignis, das von einer Discovery-Regel erstellt wurde; 2 - Ereignis, das durch aktive Agent-Autoregistrierung erstellt wurde; 3 - internes Ereignis; 4 - Ereignis, das bei einer Aktualisierung des Service-Status erstellt wurde.

Eigenschaft	Typ	Beschreibung
object	integer	Typ des Objekts, das mit dem Ereignis verknüpft ist. Mögliche Werte, wenn `source` auf ”event created by a trigger” gesetzt ist: 0 - Auslöser. Mögliche Werte, wenn `source` auf ”event created by a discovery rule” gesetzt ist: 1 - entdeckter Host; 2 - entdeckter Service. Mögliche Werte, wenn `source` auf ”event created by active agent autoregistration” gesetzt ist: 3 - automatisch registrierter Host. Mögliche Werte, wenn `source` auf ”internal event” gesetzt ist: 0 - Auslöser; 4 - Datenpunkt; 5 - LLD-Regel. Mögliche Werte, wenn `source` auf ”event created on service status update” gesetzt ist: 6 - Service.
objectid	ID	ID des verknüpften Objekts.
acknowledged	integer	Ob das Ereignis bestätigt wurde.
clock	timestamp	Zeitpunkt, zu dem das Ereignis erstellt wurde.
ns	integer	Nanosekunden zum Zeitpunkt der Erstellung des Ereignisses.
name	string	Aufgelöster Ereignisname.
value	integer	Status des verknüpften Objekts. Mögliche Werte, wenn `source` auf ”event created by a trigger” oder ”event created on service status update” gesetzt ist: 0 - OK; 1 - Problem. Mögliche Werte, wenn `source` auf ”event created by a discovery rule” gesetzt ist: 0 - Host oder Service verfügbar; 1 - Host oder Service nicht verfügbar; 2 - Host oder Service entdeckt; 3 - Host oder Service verloren. Mögliche Werte, wenn `source` auf ”internal event” gesetzt ist: 0 - Status ”normal”; 1 - Status ”unbekannt” oder ”nicht unterstützt”. Verhalten der Eigenschaft: - *unterstützt*, wenn `source` auf ”event created by a trigger”, ”event created by a discovery rule”, ”internal event” oder ”event created on service status update” gesetzt ist
severity	integer	Aktueller Schweregrad des Ereignisses. Mögliche Werte: 0 - nicht klassifiziert; 1 - Information; 2 - Warnung; 3 - durchschnittlich; 4 - hoch; 5 - Katastrophe.
r_eventid	ID	ID des Wiederherstellungsereignisses.

Eigenschaft	Typ	Beschreibung
c_eventid	ID	ID des Ereignisses, das verwendet wurde, um das aktuelle Ereignis anhand einer globalen Korrelationsregel zu überschreiben (zu schließen). Siehe `correlationid`, um die genaue Korrelationsregel zu identifizieren. Dieser Parameter ist nur definiert, wenn das Ereignis durch eine globale Korrelationsregel geschlossen wird.
cause_eventid	ID	ID des verursachenden Ereignisses.
correlationid	ID	ID der Korrelationsregel, die das Schließen des Problems ausgelöst hat. Dieser Parameter ist nur definiert, wenn das Ereignis durch eine globale Korrelationsregel geschlossen wird.
userid	ID	ID des Benutzers, der das Ereignis geschlossen hat (falls das Ereignis manuell geschlossen wurde).
suppressed	integer	Ob das Ereignis unterdrückt ist. Mögliche Werte: 0 - Ereignis ist im normalen Zustand; 1 - Ereignis ist unterdrückt.
opdata	string	Betriebsdaten mit erweiterten Makros.
urls	array	Aktive Medientyp-URLs.

Ereignis-Tag

Das Ereignis-Tag-Objekt hat die folgenden Eigenschaften.

Eigenschaft	Type	Beschreibung
tag	string	Name des Ereignis-Tags.
value	string	Wert des Ereignis-Tags.

URL des Medientyps

Das URL-Objekt des Medientyps hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
name	string	Name der in einem Medientyp definierten URL.
url	string	Wert der in einem Medientyp definierten URL.

Die Ergebnisse enthalten nur Einträge für aktive Medientypen mit aktiviertem Ereignismenüeintrag. In Eigenschaften verwendete Makros werden erweitert; wenn jedoch eine der Eigenschaften ein nicht erweitertes Makro enthält, werden beide Eigenschaften aus den Ergebnissen ausgeschlossen. Unterstützte Makros finden Sie unter *Unterstützte Makros*.

event.acknowledge

Beschreibung

`object event.acknowledge(object/array parameters)`

Diese Methode ermöglicht es Ihnen, Ereignisse zu aktualisieren. Die folgenden Aktualisierungsaktionen können ausgeführt werden:

- Ereignis schließen. Wenn das Ereignis bereits behoben ist, wird diese Aktion übersprungen.
- Ereignis bestätigen. Wenn das Ereignis bereits bestätigt ist, wird diese Aktion übersprungen.
- Bestätigung des Ereignisses aufheben. Wenn das Ereignis nicht bestätigt ist, wird diese Aktion übersprungen.
- Nachricht hinzufügen.
- Schweregrad des Ereignisses ändern. Wenn das Ereignis bereits denselben Schweregrad hat, wird diese Aktion übersprungen.
- Ereignis unterdrücken. Wenn das Ereignis bereits unterdrückt ist, wird diese Aktion übersprungen.
- Unterdrückung des Ereignisses aufheben. Wenn das Ereignis nicht unterdrückt ist, wird diese Aktion übersprungen.
- Ereignisbewertung auf Ursache ändern. Wenn das Ereignis bereits als Ursache eingestuft ist, wird diese Aktion übersprungen.
- Ereignisbewertung auf Symptom ändern. Wenn das Ereignis bereits als Symptom eingestuft ist, wird diese Aktion übersprungen.

Attention:
Es können nur Auslöser-Ereignisse aktualisiert werden.
 Es können nur Problemereignisse aktualisiert werden.
 Lese-/Schreibrechte für den Auslöser sind erforderlich, um das Ereignis zu schließen oder den Schweregrad des Ereignisses zu ändern.
 Um ein Ereignis zu schließen, muss das manuelle Schließen im Auslöser erlaubt sein.

Note:
Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Parameter mit den IDs der Ereignisse und Aktualisierungsvorgängen, die ausgeführt werden sollen.

Parameter	Typ	Beschreibung
eventids	ID/array	IDs der zu bestätigenden Ereignisse. Parameterverhalten: - *erforderlich*
action	integer	Aktion(en) zur Aktualisierung des Ereignisses. Mögliche Bitmaskenwerte: 1 - Problem schließen; 2 - Ereignis bestätigen; 4 - Nachricht hinzufügen; 8 - Schweregrad ändern; 16 - Bestätigung des Ereignisses aufheben; 32 - Ereignis unterdrücken; 64 - Unterdrückung des Ereignisses aufheben; 128 - Rang des Ereignisses in Ursache ändern; 256 - Rang des Ereignisses in Symptom ändern. Dies ist ein Bitmaskenfeld; jede Summe möglicher Bitmaskenwerte ist zulässig (zum Beispiel 6 zum Bestätigen eines Ereignisses und Hinterlassen einer Nachricht oder 34 zum Bestätigen und Unterdrücken eines Ereignisses). Parameterverhalten: - *erforderlich*
cause_eventid	ID	ID des Ursache-Ereignisses. Parameterverhalten: - *erforderlich*, wenn `action` das Bit „Rang des Ereignisses in Symptom ändern“ enthält
message	string	Text der Nachricht. Parameterverhalten: - *erforderlich*, wenn `action` das Bit „Nachricht hinzufügen“ enthält
severity	integer	Neuer Schweregrad für Ereignisse. Mögliche Werte: 0 - nicht klassifiziert; 1 - Information; 2 - Warnung; 3 - durchschnittlich; 4 - hoch; 5 - Katastrophe. Parameterverhalten: - *erforderlich*, wenn `action` das Bit „Schweregrad ändern“ enthält

Parameter	Typ	Beschreibung
suppress_until	integer	Unix-Zeitstempel, bis zu dem das Ereignis unterdrückt werden muss. Wenn der Wert auf „0“ gesetzt wird, ist die Unterdrückung unbefristet. Parameterverhalten: - *erforderlich*, wenn `action` das Bit „Ereignis unterdrücken“ enthält

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Ereignisse in der Eigenschaft `eventids` enthält.

Beispiele

Bestätigen eines Ereignisses

Bestätigen Sie ein einzelnes Ereignis und hinterlassen Sie eine Nachricht.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": "20427",
        "action": 6,
        "message": "Problem behoben."
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            "20427"
        ]
    },
    "id": 1
}
```

Ändern des Schweregrads eines Ereignisses

Ändern Sie den Schweregrad für mehrere Ereignisse und hinterlassen Sie eine Nachricht.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": ["20427", "20428"],
        "action": 12,
        "message": "Wartung erforderlich, um das Problem zu beheben.",
        "severity": 4
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            "20427",
```

```
            "20428"
        ]
    },
    "id": 1
}
```

Unterdrücken eines Ereignisses auf unbestimmte Zeit

Unterdrücken Sie ein einzelnes Ereignis ohne Ablaufzeit.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": "20427",
        "action": 32,
        "suppress_until": 0
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            20427
        ]
    },
    "id": 1
}
```

Aufhebung der Unterdrückung eines Ereignisses

Hebt die Unterdrückung eines einzelnen zuvor unterdrückten Ereignisses auf.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": "20427",
        "action": 64
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            20427
        ]
    },
    "id": 1
}
```

Ein Ereignis bestätigen und unterdrücken

Bestätigen Sie ein oder mehrere Ereignisse, fügen Sie eine Nachricht hinzu und unterdrücken Sie sie in derselben Anfrage.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": [
            "20427",
            "20428"
        ],
        "action": 38,
        "message": "Unterdrückung während geplanter Wartung.",
        "suppress_until": 0
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            20427,
            20428
        ]
    },
    "id": 1
}
```

Quelle

CEvent::acknowledge() in *ui/include/classes/api/services/CEvent.php*.

event.get

Beschreibung

`integer/array event.get(object parameters)`

Die Methode ermöglicht es Ihnen, Ereignisse anhand der angegebenen Parameter abzurufen.

Attention:
Diese Methode kann Ereignisse einer gelöschten Entität zurückgeben, wenn diese Ereignisse vom Housekeeper noch nicht entfernt wurden.

Note:
Diese Methode ist für Benutzer aller Typen verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Typ	Beschreibung
eventids	ID/array	Nur Events mit den angegebenen IDs zurückgeben.
groupids	ID/array	Nur Events zurückgeben, die von Objekten erstellt wurden, die zu den angegebenen Hostgruppen gehören.
hostids	ID/array	Nur Events zurückgeben, die von Objekten erstellt wurden, die zu den angegebenen Hosts gehören.
objectids	ID/array	Nur Events zurückgeben, die von den angegebenen Objekten erstellt wurden.

Parameter	Typ	Beschreibung
source	integer	Nur Events mit dem angegebenen Typ zurückgeben. Eine Liste der unterstützten Eventtypen finden Sie auf der Seite des Eventobjekts. Standard: 0 - Auslöser-Events.
object	integer	Nur Events zurückgeben, die von Objekten des angegebenen Typs erstellt wurden. Eine Liste der unterstützten Objekttypen finden Sie auf der Seite des Eventobjekts. Standard: 0 - Auslöser.
acknowledged	boolean	Wenn auf `true` gesetzt, nur bestätigte Events zurückgeben.
action	integer	Nur Events zurückgeben, für die die angegebenen Event-Aktualisierungsaktionen ausgeführt wurden. Für mehrere Aktionen verwenden Sie die Summe beliebiger zulässiger Bitmap-Werte als Bitmaske (zum Beispiel 34 für Bestätigen und Event unterdrücken).
action_userids	ID/array	Nur Events mit den angegebenen IDs von Benutzern zurückgeben, die die Event-Aktualisierungsaktionen ausgeführt haben.
suppressed	boolean	Wenn auf `true` gesetzt, nur unterdrückte Events zurückgeben.
symptom	boolean	Wenn auf `true` gesetzt, nur Symptom-Events zurückgeben.
severities	integer/array	Nur Events mit den angegebenen Event-Schweregraden zurückgeben. Gilt nur, wenn object ein Auslöser ist.
trigger_severities	integer/array	Nur Events mit den angegebenen Auslöser-Schweregraden zurückgeben. Gilt nur, wenn object ein Auslöser ist.
evaltype	integer	Tag-Auswertungsmethode. Mögliche Werte: 0 - *(Standard)* Und/Oder; 2 - Oder.
tags	array	Nur Events mit den angegebenen Tags zurückgeben. Format: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Ein leeres Array gibt alle Events zurück. Mögliche Operator-Werte: 0 - *(Standard)* Enthält; 1 - Entspricht; 2 - Enthält nicht; 3 - Entspricht nicht; 4 - Existiert; 5 - Existiert nicht.
eventid_from	string	Nur Events mit IDs größer oder gleich der angegebenen ID zurückgeben.
eventid_till	string	Nur Events mit IDs kleiner oder gleich der angegebenen ID zurückgeben.
time_from	timestamp	Nur Events zurückgeben, die zum angegebenen Zeitpunkt oder danach erstellt wurden.
time_till	timestamp	Nur Events zurückgeben, die zum angegebenen Zeitpunkt oder davor erstellt wurden.
problem_time_from	timestamp	Gibt nur Events zurück, die sich ab `problem_time_from` im Problemzustand befanden, unabhängig von ihrem aktuellen Status. Gilt nur, wenn die Quelle ein Auslöser-Event und das object ein Auslöser ist. Dieser Parameter ist obligatorisch, wenn `problem_time_till` angegeben ist.
problem_time_till	timestamp	Gibt nur Events zurück, die sich bis `problem_time_till` im Problemzustand befanden, unabhängig von ihrem aktuellen Status. Gilt nur, wenn die Quelle ein Auslöser-Event und das object ein Auslöser ist. Dieser Parameter ist obligatorisch, wenn `problem_time_from` angegeben ist.

Parameter	Typ	Beschreibung
value	integer/array	Nur Events mit den angegebenen Werten zurückgeben.
selectAcknowledges	query	Eine Eigenschaft `acknowledges` mit Event-Aktualisierungen zurückgeben. Event-Aktualisierungen werden in umgekehrt chronologischer Reihenfolge sortiert. Das Event-Aktualisierungsobjekt hat die folgenden Eigenschaften: `acknowledgeid` - `(ID)` ID der Bestätigung; `userid` - `(ID)` ID des Benutzers, der das Event aktualisiert hat; `clock` - `(timestamp)` Zeitpunkt der Aktualisierung des Events; `message` - `(string)` Text der Nachricht; `action` - `(integer)` ausgeführte Aktualisierungsaktion, siehe `event.acknowledge`; `old_severity` - `(integer)` Event-Schweregrad vor dieser Aktualisierungsaktion; `new_severity` - `(integer)` Event-Schweregrad nach dieser Aktualisierungsaktion; `suppress_until` - `(timestamp)` Zeitpunkt, bis zu dem das Event unterdrückt wird; `taskid` - `(ID)` ID der Aufgabe, wenn das aktuelle Event einen Statuswechsel durchläuft; `maintenanceid` - `(ID)` ID der Wartung; `username` - `(string)` Benutzername des Benutzers, der das Event aktualisiert hat; `name` - `(string)` Vorname des Benutzers, der das Event aktualisiert hat; `surname` - `(string)` Nachname des Benutzers, der das Event aktualisiert hat. Unterstützt `count`.
selectAlerts	query	Eine Eigenschaft `alerts` mit den vom Event generierten Alerts zurückgeben. Alerts werden in umgekehrt chronologischer Reihenfolge sortiert.
selectHosts	query	Eine Eigenschaft `hosts` mit Hosts zurückgeben, die das Objekt enthalten, das das Event erstellt hat. Wird nur für Events unterstützt, die von Auslösern, Datenpunkten oder LLD-Regeln generiert wurden.
selectRelatedObject	query	Eine Eigenschaft `relatedObject` mit dem Objekt zurückgeben, das das Event erstellt hat. Der zurückgegebene Objekttyp hängt vom Eventtyp ab.
selectSuppressionData	query	Eine Eigenschaft `suppression_data` mit der Liste aktiver Wartungen und manueller Unterdrückungen zurückgeben: `maintenanceid` - `(ID)` ID der Wartung; `userid` - `(ID)` ID des Benutzers, der das Event unterdrückt hat; `suppress_until` - `(integer)` Zeitpunkt, bis zu dem das Event unterdrückt wird.
selectTags	query	Eine Eigenschaft `tags` mit Event-Tags zurückgeben.
filter	object	Nur Ergebnisse zurückgeben, die exakt dem angegebenen Filter entsprechen. Akzeptiert ein Objekt, bei dem die Schlüssel Eigenschaftsnamen sind und die Werte entweder ein einzelner Wert oder ein Array von Werten sind, mit denen verglichen werden soll. Unterstützt keine Eigenschaften vom Datentyp `text`.
sortfield	string/array	Das Ergebnis nach den angegebenen Eigenschaften sortieren. Mögliche Werte: `eventid`, `objectid`, `clock`. Mögliche Werte bei Verwendung zusammen mit `groupBy`: `objectid`. Mögliche Werte bei Verwendung zusammen mit `countOutput` und `groupBy`: `objectid`, `rowscount`.

Parameter	Typ	Beschreibung
groupBy	string/array	Die Ergebnisse nach den angegebenen Eigenschaften gruppieren. Die angegebenen Eigenschaften werden in den Ergebnissen zurückgegeben. Mögliche Werte: `objectid`.
countOutput	boolean	Diese Parameter werden in den Referenzhinweisen beschrieben.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Gibt entweder zurück:

- ein Array von Objekten
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde, der Parameter `groupBy` jedoch nicht verwendet wurde
- ein Array von Objekten mit Aggregationsergebnissen, wenn der Parameter `groupBy` verwendet wurde

Beispiele

Abrufen von Auslöser-Ereignissen

Rufen Sie die neuesten Ereignisse von Auslöser ”22395” ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "event.get",
    "params": {
        "output": "extend",
        "selectAcknowledges": "extend",
        "selectSuppressionData": "extend",
        "selectTags": "extend",
        "objectids": "22395",
        "sortfield": ["clock", "eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

Antwort:

```
{
    "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",
            "suppressed": "0",
            "cause_eventid": "0",
            "acknowledges": [],
            "opdata": "Load averages(1m 5m 15m): (1.6328 3.0522 2.2515), # of CPUs: 2",
            "suppression_data": [],
            "tags": [
                {
                    "tag": "scope",
                    "value": "capacity"
                },
                {
                    "tag": "scope",
                    "value": "performance"
                },
                {
                    "tag": "component",
                    "value": "cpu"
                },
                {
                    "tag": "class",
                    "value": "os"
                },
                {
                    "tag": "target",
                    "value": "linux"
                }
            ],
            "urls": []
        },
        {
            "eventid": "4",
            "source": "0",
            "object": "0",
            "objectid": "22395",
            "clock": "1728657737",
            "value": "1",
            "acknowledged": "1",
            "ns": "460759366",
            "name": "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",
            "suppressed": "1",
            "cause_eventid": "0",
            "acknowledges": [
                {
                    "acknowledgeid": "1",
                    "userid": "1",
                    "clock": "1728657938",
                    "message": "Testing environment. Please disregard this alert.",
                    "action": "38",
                    "old_severity": "0",
                    "new_severity": "0",
                    "suppress_until": "1728744338",
                    "taskid": "0",
                    "maintenanceid": "0",
```

```
                    "username": "Admin",
                    "name": "Zabbix",
                    "surname": "Administrator"
                }
            ],
            "opdata": "Load averages(1m 5m 15m): (1.6328 3.0522 2.2515), # of CPUs: 2",
            "suppression_data": [
                {
                    "maintenanceid": "0",
                    "suppress_until": "1728744338",
                    "userid": "1"
                }
            ],
            "tags": [
                {
                    "tag": "scope",
                    "value": "capacity"
                },
                {
                    "tag": "scope",
                    "value": "performance"
                },
                {
                    "tag": "component",
                    "value": "cpu"
                },
                {
                    "tag": "class",
                    "value": "os"
                },
                {
                    "tag": "target",
                    "value": "linux"
                }
            ],
            "urls": []
        }
    ],
    "id": 1
}
```


Abrufen von Ereignissen nach Zeitraum

Rufen Sie alle Ereignisse ab, die zwischen dem 17. und 18. Oktober 2012 erstellt wurden, in umgekehrt chronologischer Reihenfolge.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "event.get",
    "params": {
        "output": "extend",
        "time_from": "1350432000",
        "time_till": "1350518400",
        "sortfield": ["clock", "eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

Antwort:


```
{
    "jsonrpc": "2.0",
```

```
    "result": [
        {
            "eventid": "20617",
            "source": "0",
            "object": "0",
            "objectid": "14282",
            "clock": "1350477816",
            "value": "1",
            "acknowledged": "0",
            "ns": "0",
            "name": "Weniger als 25 % frei im Verlaufs-Cache",
            "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": "Zabbix-Trapper-Prozesse sind zu mehr als 75 % ausgelastet",
            "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": "Zabbix-Trapper-Prozesse sind zu mehr als 75 % ausgelastet",
            "severity": "3",
            "r_eventid": "20616",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "opdata": "",
            "suppressed": "0",
            "urls": []
        }
    ],
```

```
    "id": 1
}
```

Abrufen von Ereignissen, die von einem angegebenen Benutzer bestätigt wurden

Abrufen von Ereignissen, die von Benutzer mit ID=10 bestätigt wurden

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "event.get",
    "params": {
        "output": "extend",
        "action": 2,
        "action_userids": [10],
        "selectAcknowledges": ["userid", "action"],
        "sortfield": ["eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "503",
            "source": "0",
            "object": "0",
            "objectid": "23162",
            "clock": "1747212236",
            "value": "1",
            "acknowledged": "1",
            "ns": "413470863",
            "name": "Number of installed packages has been changed",
            "severity": "2",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "suppressed": "0",
            "cause_eventid": "0",
            "acknowledges": [
                {
                    "userid": "10",
                    "action": "2"
                }
            ],
            "opdata": "",
            "urls": []
        }
    ],
    "id": 1
}
```

Abrufen der Top-5-Auslöser nach Anzahl der Problemereignisse

Rufen Sie die 5 wichtigsten Auslöser mit den Schweregraden „Warning“, „Average“, „High“ oder „Disaster“ zusammen mit der Anzahl der Problemereignisse innerhalb eines angegebenen Zeitraums ab.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "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
}
```

Siehe auch

- Alert
- Item
- Host
- LLD-Regel
- Service
- Auslöser

Quelle

CEvent::get() in *ui/include/classes/api/services/CEvent.php*.

Erkannter Dienst

Diese Klasse ist für die Arbeit mit erkannten Diensten vorgesehen.

Objektreferenzen:

- Erkannter Dienst

Verfügbare Methoden:

- dservice.get - erkannte Dienste abrufen

Entdecktes Dienstobjekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `dservice` API.

Erkannter Dienst

> **Note:**
> Erkannte Dienste werden vom Zabbix-Server erstellt und können nicht über die API geändert werden.

Das Objekt für erkannte Dienste enthält Informationen über einen Dienst, der durch eine Netzwerk-Erkennungsregel auf einem Host erkannt wurde. Es hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
dserviceid	ID	ID des erkannten Dienstes.
dcheckid	ID	ID der Erkennungsprüfung, die zum Erkennen des Dienstes verwendet wurde.
dhostid	ID	ID des erkannten Hosts, auf dem der Dienst ausgeführt wird.
dns	string	DNS des Hosts, auf dem der Dienst ausgeführt wird.
ip	string	IP-Adresse des Hosts, auf dem der Dienst ausgeführt wird.
lastdown	timestamp	Zeitpunkt, zu dem der erkannte Dienst zuletzt ausgefallen ist.
lastup	timestamp	Zeitpunkt, zu dem der erkannte Dienst zuletzt verfügbar war.
port	integer	Portnummer des Dienstes.
status	integer	Status des Dienstes. Mögliche Werte: 0 - Dienst verfügbar; 1 - Dienst nicht verfügbar.
value	string	Vom Dienst zurückgegebener Wert bei der Durchführung einer Zabbix-Agent-, SNMPv1-, SNMPv2- oder SNMPv3-Erkennungsprüfung.

dservice.get

Beschreibung

`integer/array dservice.get(object parameters)`

Die Methode ermöglicht es Ihnen, erkannte Dienste anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
dserviceids	ID/array	Gibt nur entdeckte Services mit den angegebenen IDs zurück.
dhostids	ID/array	Gibt nur entdeckte Services zurück, die zu den angegebenen entdeckten Hosts gehören.
dcheckids	ID/array	Gibt nur entdeckte Services zurück, die durch die angegebenen Discovery-Prüfungen erkannt wurden.
druleids	ID/array	Gibt nur entdeckte Services zurück, die durch die angegebenen Discovery-Regeln erkannt wurden.
selectDRules	query	Gibt eine Eigenschaft `drules` mit einem Array der Discovery-Regeln zurück, die den Service erkannt haben.

Parameter	Type	Beschreibung
selectDHosts	query	Gibt eine Eigenschaft `dhosts` mit einem Array der entdeckten Hosts zurück, zu denen der Service gehört.
selectHosts	query	Gibt eine Eigenschaft `hosts` mit den Hosts zurück, die dieselbe IP-Adresse und denselben Proxy wie der Service haben. Unterstützt `count`.
limitSelects	integer	Begrenzt die Anzahl der von Unterabfragen zurückgegebenen Datensätze. Gilt für die folgenden Unterabfragen: `selectHosts` - das Ergebnis wird nach `hostid` sortiert.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `dserviceid`, `dhostid`, `ip`.
countOutput	boolean	Diese Parameter sind in der Referenzkommentierung beschrieben.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Kann die folgenden Dinge zurück geben:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Auf einem Host erkannte Services abrufen

Rufen Sie alle erkannten Services ab, die auf dem erkannten Host „11“ gefunden wurden.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "dservice.get",
    "params": {
        "output": "extend",
        "dhostids": "11"
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```


Siehe auch

- Erkannter Host
- Discovery- Prüfung
- Host

Quelle

CDService::get() in *ui/include/classes/api/services/CDService.php*.

Gerät

Diese Klasse ist für die Arbeit mit mobilen Geräten zur Verwendung mit der Zabbix mobile app vorgesehen.

Objektreferenzen:

- Device

Verfügbare Methoden:

- device.get - Geräte abrufen
- device.offboard - Geräte entfernen (außer Betrieb nehmen)

Geräteobjekt

Das Geräteobjekt stellt ein mobiles Gerät zur Verwendung mit der Zabbix Mobile App dar.

Property	Type	Description
deviceid	ID	Geräte-ID (intern, z. B. 1, 2). Property behavior: - Nur lesbar
userid	ID	ID des Gerätebenutzers. Property behavior: - Konstant
uuid	string	Universell eindeutige Kennung (UUIDv7) des Geräts (extern, z. B. `019f930e-0ea2-7c1f-9dcc-0fbb60c40664`). Property behavior: - Nur lesbar
name	string	Gerätename. Property behavior: - Nur lesbar

Property	Type	Description
status	integer	Gerätestatus. Mögliche Werte: 0 - Neu 1 - Aktiv 2 - Nicht zugewiesen Property behavior: - Nur lesbar
activated_at	timestamp	Datum und Uhrzeit, zu denen das Gerät in der mobilen App hinzugefügt wurde, oder 0, wenn es noch nicht hinzugefügt wurde. Property behavior: - Nur lesbar
lastaccess	timestamp	Datum und Uhrzeit, zu denen sich das Gerät zuletzt in der mobilen App authentifiziert hat, oder 0, wenn es noch nicht authentifiziert wurde. Property behavior: - Nur lesbar

device.get

Diese Methode ermöglicht das Abrufen mobiler Geräte zur Verwendung mit der Zabbix Mobile App.

Die Methode ist für Benutzer mit den Rollen *User*, *Admin* und *Super admin* verfügbar, wenn `EnableMobileDevices` aktiviert ist und der Benutzer Zugriff auf Geräte hat.
Der Zugriff auf API-Methoden kann in der Konfiguration der Benutzerrolle entzogen werden.

Benutzer mit den Rollen *User* und *Admin* können nur ihre eigenen Geräte abrufen.

Methodensignatur: `integer/array device.get(object parameters)`

Methodenquelle: `CDevice::get()` in `ui/include/classes/api/services/CDevice.php`

Parameter

Parameter	Typ	Beschreibung
deviceids	ID/array	Gibt nur Geräte mit den angegebenen Geräte-IDs zurück.
userids	ID/array	Gibt nur Geräte zurück, die Benutzern mit den angegebenen Benutzer-IDs gehören.
roleids	ID/array	Gibt nur Geräte zurück, die Benutzern mit den angegebenen Benutzerrollen-IDs gehören.
usrgrpids	ID/array	Gibt nur Geräte zurück, die Benutzern mit den angegebenen Benutzergruppen-IDs gehören.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `deviceid`, `name`.
countOutput	boolean	Diese Parameter werden in der Referenzdokumentation beschrieben.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

Gibt entweder ein Array von Objekten oder die Anzahl der abgerufenen Objekte zurück (wenn `countOutput` verwendet wird).

Beispiele

Geräte abrufen

Rufen Sie die Geräte ab, die zu Benutzern mit den Benutzer-IDs 1, 2 und 3 gehören.
 Beachten Sie, dass Benutzer 2 (Gast) keine Geräte besitzt und Gerät 3 noch nicht zur mobilen Zabbix-App hinzugefügt oder darin authentifiziert wurde.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "device.get",
    "params": {
        "output": "extend",
        "userids": [1, 2, 3]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "deviceid": "1",
            "userid": "1",
            "uuid": "019f930e-0ea2-7c1f-9dcc-0fbb60c40664",
            "name": "Google Pixel 10",
            "status": "1",
            "activated_at": "1784533122",
            "lastaccess": "1784879933"
        },
        {
            "deviceid": "2",
            "userid": "1",
            "uuid": "019d00fb-9e87-7ad5-bb4e-978467254aff",
            "name": "Apple iPhone 15 Pro",
            "status": "1",
            "activated_at": "1784533225",
            "lastaccess": "0"
        },
        {
            "deviceid": "3",
            "userid": "3",
            "uuid": "019d00fb-9e87-7ad5-bb4e-6ed88a2297ee",
            "name": "",
            "status": "0",
            "activated_at": "0",
            "lastaccess": "0"
        }
    ],
    "id": 1
}
```

device.offboard

Mit dieser Methode können Sie mobile Geräte zur Verwendung mit der Zabbix Mobile App entfernen (Offboarding).

Die Methode ist für Benutzer mit den Rollen *User*, *Admin* und *Super admin* verfügbar, wenn `EnableMobileDevices` aktiviert ist und der Benutzer Zugriff auf Geräte hat. Der Zugriff auf API-Methoden kann in der Konfiguration der Benutzerrolle entzogen werden.

Benutzer mit den Rollen *User* und *Admin* können nur ihre eigenen Geräte entfernen.

Methodensignatur: `object device.offboard(object parameters)`

Methodenquelle: `CDevice::offboard()` in `ui/include/classes/api/services/CDevice.php`

Parameter

Parameter	Typ	Beschreibung
uuid	string	Universell eindeutiger Bezeichner des Geräts (UUIDv7). Entspricht `deviceid`.
force	boolean	Auf `true` setzen, um den Fehler „Gerät nicht gefunden“ zu ignorieren und das Gerät zwangsweise zu entfernen. Standard: `false`

Rückgabewerte

Gibt ein Objekt zurück, das die UUID des entfernten Geräts enthält.

Beispiele

Gerät entfernen

Entfernt ein Gerät mit der angegebenen UUID und ignoriert den Fehler „Gerät nicht gefunden“.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "device.offboard",
    "params": {
        "uuid": "019d0106-5eef-7154-a159-bb29e35d38e5",
        "force": true
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "uuid": "019d0106-5eef-7154-a159-bb29e35d38e5"
    },
    "id": 1
}
```

Graph

Diese Klasse ist für die Arbeit mit Graphen vorgesehen.

Objektreferenzen:

- Graph

Verfügbare Methoden:

- graph.create - neue Graphen erstellen
- graph.delete - Graphen löschen
- graph.get - Graphen abrufen
- graph.update - Graphen aktualisieren

Graph Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `graph` API.

Graph

Das Graph-Objekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
graphid	ID	ID des Graphen. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
height	integer	Höhe des Graphen in Pixeln. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
name	string	Name des Graphen. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
width	integer	Breite des Graphen in Pixeln. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
flags	integer	Herkunft des Graphen. Mögliche Werte: 0 - *(Standard)* ein einfacher Graph; 4 - ein aus einem Prototyp konvertierter Graph. Verhalten der Eigenschaft: - *schreibgeschützt*
graphtype	integer	Layout-Typ des Graphen. Mögliche Werte: 0 - *(Standard)* normal; 1 - gestapelt; 2 - Kreisdiagramm; 3 - explodiert.
percent_left	float	Linkes Perzentil. Standard: 0.
percent_right	float	Rechtes Perzentil. Standard: 0.
show_3d	integer	Gibt an, ob Kreisdiagramme und explodierte Graphen in 3D angezeigt werden. Mögliche Werte: 0 - *(Standard)* in 2D anzeigen; 1 - in 3D anzeigen.
show_legend	integer	Gibt an, ob die Legende im Graphen angezeigt wird. Mögliche Werte: 0 - ausblenden; 1 - *(Standard)* anzeigen.
show_work_period	integer	Gibt an, ob die Arbeitszeit im Graphen angezeigt wird. Mögliche Werte: 0 - ausblenden; 1 - *(Standard)* anzeigen.
show_triggers	integer	Gibt an, ob die Auslöser-Linie im Graphen angezeigt wird. Mögliche Werte: 0 - ausblenden; 1 - *(Standard)* anzeigen.

Eigenschaft	Typ	Beschreibung
templateid	ID	ID des übergeordneten Vorlagen-Graphen. Verhalten der Eigenschaft: - *schreibgeschützt*
yaxismax	float	Der feste Maximalwert für die Y-Achse. Standard: 100.
yaxismin	float	Der feste Minimalwert für die Y-Achse. Standard: 0.
ymax_itemid	ID	ID des Datenpunkts, der als Maximalwert für die Y-Achse verwendet wird. Wenn ein Benutzer keinen Zugriff auf den angegebenen Datenpunkt hat, wird der Graph so dargestellt, als wäre `ymax_type` auf „calculated“ gesetzt.
ymax_type	integer	Berechnungsmethode des Maximalwerts für die Y-Achse. Mögliche Werte: 0 - *(Standard)* calculated; 1 - fest; 2 - Datenpunkt.
ymin_itemid	ID	ID des Datenpunkts, der als Minimalwert für die Y-Achse verwendet wird. Wenn ein Benutzer keinen Zugriff auf den angegebenen Datenpunkt hat, wird der Graph so dargestellt, als wäre `ymin_type` auf „calculated“ gesetzt.
ymin_type	integer	Berechnungsmethode des Minimalwerts für die Y-Achse. Mögliche Werte: 0 - *(Standard)* calculated; 1 - fest; 2 - Datenpunkt.
uuid	string	Universell eindeutige Kennung, die verwendet wird, um importierte Graphen mit bereits vorhandenen zu verknüpfen. Wird automatisch erzeugt, wenn sie nicht angegeben wird. Verhalten der Eigenschaft: - *unterstützt*, wenn der Graph zu einer Vorlage gehört

graph.create

Beschreibung

`object graph.create(object/array graphs)`

Diese Methode ermöglicht es Ihnen, neue Diagramme zu erstellen.

Note:
Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellende Graphen.

Zusätzlich zu den Standard-Grapheigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
gitems	array	Graph-Elemente, die für den Graphen erstellt werden sollen. Parameterverhalten: - *erforderlich*

Return values

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Graphen unter der Eigenschaft `graphids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Graphen.

Beispiele

Erstellen eines Graphen

Erstellen Sie einen Graphen mit zwei Datenpunkten.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652"
        ]
    },
    "id": 1
}
```

Siehe auch

- Graph-Datenpunkt

Quelle

CGraph::create() in *ui/include/classes/api/services/CGraph.php*.

graph.delete

Beschreibung

`object graph.delete(array graphIds)`

Diese Methode ermöglicht es Ihnen, Diagramme zu löschen.

Note:
Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der zu löschenden Diagramme.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Graphen unter der Eigenschaft `graphids` enthält.

Beispiele

Mehrere Graphen löschen

Löschen Sie zwei Graphen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "graph.delete",
    "params": [
        "652",
        "653"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652",
            "653"
        ]
    },
    "id": 1
}
```

Quelle

CGraph::delete() in *ui/include/classes/api/services/CGraph.php*.

graph.get

Beschreibung

`integer/array graph.get(object parameters)`

Die Methode ermöglicht es Ihnen, Diagramme anhand der angegebenen Parameter abzurufen.

Note:
Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
graphids	ID/array	Gibt nur Graphen mit den angegebenen IDs zurück.
groupids	ID/array	Gibt nur Graphen zurück, die zu Hosts oder Vorlagen in den angegebenen Host-Gruppen oder Vorlagen-Gruppen gehören.

Parameter	Type	Beschreibung
templateids	ID/array	Gibt nur Graphen zurück, die zu den angegebenen Vorlagen gehören.
hostids	ID/array	Gibt nur Graphen zurück, die zu den angegebenen Hosts gehören.
itemids	ID/array	Gibt nur Graphen zurück, die die angegebenen Datenpunkte enthalten.
templated	boolean	Wenn auf `true` gesetzt, werden nur Graphen zurückgegeben, die zu Vorlagen gehören.
inherited	boolean	Wenn auf `true` gesetzt, werden nur Graphen zurückgegeben, die von einer Vorlage geerbt wurden.
expandName	flag	Erweitert Makros im Graphnamen.
selectDiscoveryData	query	Gibt eine `discoveryData`-Eigenschaft mit den Daten des Graph-Discovery-Objekts zurück. Das Graph-Discovery-Objekt verknüpft einen entdeckten Graphen mit einem Graphprototyp, aus dem er entdeckt wurde. Es hat die folgenden Eigenschaften: `parent_graphid` - `(ID)` ID des Graphprototyps, aus dem der Graph erstellt wurde; `status` - `(int)` Status der Graph-Discovery: 0 - *(Standard)* Graph ist entdeckt, 1 - Graph ist nicht mehr entdeckt; `ts_delete` - `(timestamp)` Zeitpunkt, zu dem ein Graph, der nicht mehr entdeckt wird, gelöscht wird.
selectDiscoveryRule	query	Gibt eine `discoveryRule`-Eigenschaft mit der Low-Level-Discovery-Regel zurück, die den Graphen erstellt hat.
selectHostGroups	query	Gibt eine `hostgroups`-Eigenschaft mit den Host-Gruppen zurück, zu denen der Graph gehört.
selectTemplateGroups	query	Gibt eine `templategroups`-Eigenschaft mit den Vorlagen-Gruppen zurück, zu denen der Graph gehört.
selectTemplates	query	Gibt eine `templates`-Eigenschaft mit den Vorlagen zurück, zu denen der Graph gehört.
selectHosts	query	Gibt eine `hosts`-Eigenschaft mit den Hosts zurück, zu denen der Graph gehört.
selectItems	query	Gibt eine `items`-Eigenschaft mit den im Graphen verwendeten Datenpunkten zurück.
selectGraphItems	query	Gibt eine `gitems`-Eigenschaft mit den im Graphen verwendeten Datenpunkten zurück.
filter	object	Gibt nur die Ergebnisse zurück, die exakt dem angegebenen Filter entsprechen. Akzeptiert ein Objekt, bei dem die Schlüssel Eigenschaftsnamen sind und die Werte entweder ein einzelner Wert oder ein Array von Werten sind, mit denen abgeglichen wird. Unterstützt keine Eigenschaften des `text`-Datentyps. Unterstützt zusätzliche Eigenschaften: `host` - technischer Name des Hosts, zu dem der Graph gehört; `hostid` - ID des Hosts, zu dem der Graph gehört.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `graphid`, `name`, `graphtype`.
countOutput	boolean	Diese Parameter sind in der Referenzkommentierung beschrieben.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Parameter	Type	Beschreibung
selectGraphDiscovery	query	Gibt eine `graphDiscovery`-Eigenschaft mit dem Graph-Discovery-Objekt zurück. Die Graph-Discovery-Objekte verknüpfen den Graphen mit einem Graphprototyp, aus dem er erstellt wurde. Diese Abfrage ist **veraltet**, bitte verwenden Sie stattdessen `selectDiscoveryData`.

Rückgabewerte

`(integer/array)` Kann die folgenden Dinge zurück geben:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Abrufen von Diagrammen von Hosts

Rufen Sie alle Diagramme vom Host „10107“ ab und sortieren Sie sie nach Namen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "graph.get",
    "params": {
        "output": "extend",
        "hostids": 10107,
        "sortfield": "name"
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```


Siehe auch

- Discovery-Regel
- Graph-Datenpunkt
- Datenpunkt
- Host
- Host-Gruppe
- Vorlage
- Vorlagengruppe

Quelle

CGraph::get() in *ui/include/classes/api/services/CGraph.php*.

graph.update

Beschreibung

`object graph.update(object/array graphs)`

Diese Methode ermöglicht es Ihnen, vorhandene Diagramme zu aktualisieren.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu aktualisierende Graph-Eigenschaften.

Die Eigenschaft `graphid` muss für jeden Graphen definiert sein, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Zusätzlich zu den Standard-Graph-Eigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
gitems	array	Graph-Elemente zum Ersetzen vorhandener Graph-Elemente. Wenn für ein Graph-Element die Eigenschaft `gitemid` definiert ist, wird es aktualisiert, andernfalls wird ein neues Graph-Element erstellt.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Graphen unter der Eigenschaft `graphids` enthält.

Beispiele

Festlegen des Maximums für die Y-Skala

Setzen Sie das Maximum der Y-Skala auf einen festen Wert von 100.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "graph.update",
    "params": {
        "graphid": "439",
        "ymax_type": 1,
        "yaxismax": 100
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "439"
```

```
        ]
    },
    "id": 1
}
```

Quelle

CGraph::update() in *ui/include/classes/api/services/CGraph.php*.

Graph-Datenpunkt

Diese Klasse ist für die Arbeit mit Graph-Datenpunkten ausgelegt.

Objektreferenzen:

- Graph-Datenpunkt

Verfügbare Methoden:

- graphitem.get - Graph-Datenpunkte abrufen

Graph-Item Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `graphitem` API.

Graph-Datenpunkt

> **Note:**
> Graph-Datenpunkte können nur über die `graph`-API geändert werden.

Das Graph-Datenpunkt-Objekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
gitemid	ID	ID des Graph-Datenpunkts. Verhalten der Eigenschaft: - *schreibgeschützt*
color	string	Zeichenfarbe des Graph-Datenpunkts als hexadezimaler Farbcode. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsoperationen
itemid	ID	ID des Datenpunkts. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsoperationen
calc_fnc	integer	Wert des Datenpunkts, der angezeigt wird. Mögliche Werte: 1 - Minimalwert; 2 - *(Standard)* Durchschnittswert; 4 - Maximalwert; 7 - alle Werte; 9 - letzter Wert, wird nur von Kreis- und Explosionsdiagrammen verwendet.
drawtype	integer	Zeichenstil des Graph-Datenpunkts. Mögliche Werte: 0 - *(Standard)* Linie; 1 - gefüllter Bereich; 2 - fette Linie; 3 - Punkt; 4 - gestrichelte Linie; 5 - Verlaufslinie.
graphid	ID	ID des Graphen, zu dem der Graph-Datenpunkt gehört.

Eigenschaft	Typ	Beschreibung
sortorder	integer	Position des Datenpunkts im Graphen. Standard: beginnt mit „0“ und erhöht sich mit jedem Eintrag um eins.
type	integer	Typ des Graph-Datenpunkts. Mögliche Werte: 0 - *(Standard)* einfach; 2 - Graph-Summe, wird nur von Kreis- und Explosionsdiagrammen verwendet.
yaxisside	integer	Seite des Graphen, auf der die Y-Skala des Graph-Datenpunkts gezeichnet wird. Mögliche Werte: 0 - *(Standard)* linke Seite; 1 - rechte Seite.

graphitem.get

Beschreibung

`integer/array graphitem.get(object parameters)`

Die Methode ermöglicht es Ihnen, Graph-Elemente anhand der angegebenen Parameter abzurufen.

Note:
Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
graphids	ID/array	Gibt nur Graph-Datenpunkte zurück, die zu den angegebenen Graphen gehören.
itemids	ID/array	Gibt nur Graph-Datenpunkte mit den angegebenen Datenpunkt-IDs zurück.
type	integer	Gibt nur Graph-Datenpunkte mit dem angegebenen Typ zurück. Eine Liste der unterstützten Typen von Graph-Datenpunkten finden Sie auf der Seite zum Graph-Datenpunkt-Objekt.
selectGraphs	query	Gibt eine Eigenschaft `graphs` mit einem Array von Graphen zurück, zu denen der Datenpunkt gehört.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `gitemid`.
countOutput	boolean	Diese Parameter werden in der Referenzkommentierung beschrieben.
editable	boolean	
limit	integer	
output	query	
preservekeys	boolean	
sortorder	string/array	

Rückgabewerte

`(integer/array)` Kann die folgenden Dinge zurück geben:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Abrufen von Graph-Datenpunkten aus einem Graphen

Rufen Sie alle in einem Graphen verwendeten Graph-Datenpunkte mit zusätzlichen Informationen über den Datenpunkt und den Host ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "graphitem.get",
    "params": {
        "output": "extend",
        "graphids": "387"
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```

Siehe auch

- Graph

Quelle

CGraphItem::get() in *ui/include/classes/api/services/CGraphItem.php*.

Graph-Prototyp

Diese Klasse ist für die Arbeit mit Graph-Prototypen ausgelegt.

Objektreferenzen:

- Graph-Prototyp

Verfügbare Methoden:

- graphprototype.create - neue Graph-Prototypen erstellen
- graphprototype.delete - Graph-Prototypen löschen
- graphprototype.get - Graph-Prototypen abrufen
- graphprototype.update - Graph-Prototypen aktualisieren

Graph-Prototyp-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `graphprototype` API.

Graph-Prototyp

Das Graph-Prototyp-Objekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
graphid	ID	ID des Graph-Prototyps. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
height	integer	Höhe des Graph-Prototyps in Pixeln. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
name	string	Name des Graph-Prototyps. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
width	integer	Breite des Graph-Prototyps in Pixeln. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
flags	integer	Herkunft des Graph-Prototyps. Mögliche Werte: 2 - ein Graph-Prototyp; 6 - ein entdeckter Graph-Prototyp Verhalten der Eigenschaft: - *schreibgeschützt*
graphtype	integer	Layout-Typ des Graph-Prototyps. Mögliche Werte: 0 - *(Standard)* normal; 1 - gestapelt; 2 - Kreisdiagramm; 3 - explodiert.
percent_left	float	Linkes Perzentil. Standard: 0.
percent_right	float	Rechtes Perzentil. Standard: 0.

Eigenschaft	Typ	Beschreibung
show_3d	integer	Gibt an, ob entdeckte Kreis- und explodierte Graphen in 3D angezeigt werden sollen. Mögliche Werte: 0 - *(Standard)* in 2D anzeigen; 1 - in 3D anzeigen.
show_legend	integer	Gibt an, ob die Legende im entdeckten Graphen angezeigt werden soll. Mögliche Werte: 0 - ausblenden; 1 - *(Standard)* anzeigen.
show_work_period	integer	Gibt an, ob die Arbeitszeit im entdeckten Graphen angezeigt werden soll. Mögliche Werte: 0 - ausblenden; 1 - *(Standard)* anzeigen.
templateid	ID	ID des Graph-Prototyps der übergeordneten Vorlage. Verhalten der Eigenschaft: - *schreibgeschützt*
yaxismax	float	Der feste Maximalwert für die Y-Achse.
yaxismin	float	Der feste Minimalwert für die Y-Achse.
ymax_itemid	ID	ID des Datenpunkts, der als Maximalwert für die Y-Achse verwendet wird. Wenn ein Benutzer keinen Zugriff auf den angegebenen Datenpunkt hat, wird der Graph so dargestellt, als wäre `ymax_type` auf „calculated“ gesetzt.
ymax_type	integer	Berechnungsmethode des Maximalwerts für die Y-Achse. Mögliche Werte: 0 - *(Standard)* calculated; 1 - fest; 2 - Datenpunkt.
ymin_itemid	ID	ID des Datenpunkts, der als Minimalwert für die Y-Achse verwendet wird. Wenn ein Benutzer keinen Zugriff auf den angegebenen Datenpunkt hat, wird der Graph so dargestellt, als wäre `ymin_type` auf „calculated“ gesetzt.
ymin_type	integer	Berechnungsmethode des Minimalwerts für die Y-Achse. Mögliche Werte: 0 - *(Standard)* calculated; 1 - fest; 2 - Datenpunkt.
discover	integer	Erkennungsstatus des Graph-Prototyps. Mögliche Werte: 0 - *(Standard)* neue Graphen werden erkannt; 1 - neue Graphen werden nicht erkannt und vorhandene Graphen werden als verloren markiert.
uuid	string	Universell eindeutige Kennung, die verwendet wird, um importierte Graph-Prototypen mit bereits vorhandenen zu verknüpfen. Wird automatisch erzeugt, wenn sie nicht angegeben ist. Verhalten der Eigenschaft: - *unterstützt*, wenn der Graph-Prototyp zu einer Vorlage gehört

graphprototype.create

Beschreibung

`object graphprototype.create(object/array graphPrototypes)`

Diese Methode ermöglicht es Ihnen, neue Diagrammvorlagen zu erstellen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellende Graph-Prototypen.

Zusätzlich zu den Standard-Eigenschaften von Graph-Prototypen akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
gitems	array	Graph-Elemente, die für die Graph-Prototypen erstellt werden sollen. Graph-Elemente können sich sowohl auf Datenpunkte als auch auf Datenpunkt-Prototypen beziehen, es muss jedoch mindestens ein Datenpunkt-Prototyp vorhanden sein. Parameterverhalten: - *erforderlich*

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Graphenprototypen unter der Eigenschaft `graphids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Graphenprototypen.

Beispiele

Erstellen eines Graphprototyps

Erstellen Sie einen Graphprototyp mit zwei Datenpunkten.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.create",
    "params": {
        "name": "Festplattennutzung {#FSNAME}",
        "width": 900,
        "height": 200,
        "gitems": [
            {
                "itemid": "22828",
                "color": "00AA00"
            },
            {
                "itemid": "22829",
                "color": "3333FF"
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
```

```
            "652"
        ]
    },
    "id": 1
}
```

Siehe auch

- Graph-Datenpunkt

Quelle

CGraphPrototype::create() in *ui/include/classes/api/services/CGraphPrototype.php*.

graphprototype.delete

Beschreibung

`object graphprototype.delete(array graphPrototypeIds)`

Diese Methode ermöglicht es Ihnen, Graphvorlagen zu löschen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der zu löschenden Graphprototypen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Graphenprototypen unter der Eigenschaft `graphids` enthält.

Beispiele

Mehrere Graphprototypen löschen

Löschen Sie zwei Graphprototypen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.delete",
    "params": [
        "652",
        "653"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652",
            "653"
        ]
    },
    "id": 1
}
```

Quelle

CGraphPrototype::delete() in *ui/include/classes/api/services/CGraphPrototype.php*.

graphprototype.get

Beschreibung

`integer/array graphprototype.get(object parameters)`

Die Methode ermöglicht es Ihnen, Graphvorlagen anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
discoveryids	ID/array	Gibt nur Graph-Prototypen zurück, die zu den angegebenen Discovery-Regeln gehören.
graphids	ID/array	Gibt nur Graph-Prototypen mit den angegebenen IDs zurück.
groupids	ID/array	Gibt nur Graph-Prototypen zurück, die zu Hosts oder Vorlagen in den angegebenen Host-Gruppen oder Vorlagen-Gruppen gehören.
hostids	ID/array	Gibt nur Graph-Prototypen zurück, die zu den angegebenen Hosts gehören.
inherited	boolean	Wenn auf `true` gesetzt, werden nur von einer Vorlage geerbte Graph-Prototypen zurückgegeben.
itemids	ID/array	Gibt nur Graph-Prototypen zurück, die die angegebenen Datenpunkt-Prototypen enthalten.
templated	boolean	Wenn auf `true` gesetzt, werden nur Graph-Prototypen zurückgegeben, die zu Vorlagen gehören.
templateids	ID/array	Gibt nur Graph-Prototypen zurück, die zu den angegebenen Vorlagen gehören.
selectDiscoveryData	query	Gibt eine `discoveryData`-Eigenschaft mit den Discovery-Objektdaten des Graph-Prototyps zurück. Das Discovery-Objekt des Graph-Prototyps verknüpft einen entdeckten Graph-Prototyp mit einem Graph-Prototyp, aus dem er entdeckt wurde. Es hat die folgenden Eigenschaften: `parent_graphid` - `(ID)` ID des Graph-Prototyps, aus dem der Graph erstellt wurde; `status` - `(int)` Status der Graph-Discovery: 0 - *(Standard)* Graph-Prototyp ist entdeckt, 1 - Graph-Prototyp ist nicht mehr entdeckt; `ts_delete` - `(timestamp)` Zeitpunkt, zu dem ein Graph-Prototyp, der nicht mehr entdeckt wird, gelöscht wird.
selectDiscoveryRule	query	Gibt eine Eigenschaft `discoveryRule` mit der LLD-Regel zurück, zu der der Graph-Prototyp gehört.
selectDiscoveryRulePrototype	query	Gibt eine Eigenschaft `discoveryRulePrototype` mit dem übergeordneten LLD-Regelprototyp zurück, zu dem der Graph-Prototyp gehört.
selectGraphItems	query	Gibt eine Eigenschaft `gitems` mit den im Graph-Prototyp verwendeten Graph-Elementen zurück.
selectHostGroups	query	Gibt eine Eigenschaft `hostgroups` mit den Host-Gruppen zurück, zu denen der Graph-Prototyp gehört.
selectHosts	query	Gibt eine Eigenschaft `hosts` mit den Hosts zurück, zu denen der Graph-Prototyp gehört.
selectItems	query	Gibt eine `items`-Eigenschaft mit den im Graph-Prototyp verwendeten Datenpunkten und Datenpunkt-Prototypen zurück.
selectTemplateGroups	query	Gibt eine Eigenschaft `templategroups` mit den Vorlagen-Gruppen zurück, zu denen der Graph-Prototyp gehört.
selectTemplates	query	Gibt eine Eigenschaft `templates` mit den Vorlagen zurück, zu denen der Graph-Prototyp gehört.

Parameter	Type	Beschreibung
filter	object	Gibt nur die Ergebnisse zurück, die exakt dem angegebenen Filter entsprechen. Akzeptiert ein Objekt, bei dem die Schlüssel Eigenschaftsnamen sind und die Werte entweder ein einzelner Wert oder ein Array von Werten sind, mit denen abgeglichen werden soll. Unterstützt keine Eigenschaften des `text`-Datentyps. Unterstützt zusätzliche Eigenschaften: `host` - technischer Name des Hosts, zu dem der Graph-Prototyp gehört; `hostid` - ID des Hosts, zu dem der Graph-Prototyp gehört.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `graphid`, `name`, `graphtype`, `discovered`.
countOutput	boolean	Diese Parameter sind im Referenzkommentar beschrieben.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Kann die folgenden Dinge zurück geben:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Abrufen von Graphprototypen aus einer LLD-Regel

Rufen Sie alle Graphprototypen aus einer LLD-Regel ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.get",
    "params": {
        "output": "extend",
        "discoveryids": "27426"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "graphid": "1017",
            "name": "Speicherplatzbelegung {#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
}
```


Siehe auch

- Discovery-Regel
- Graph-Datenpunkt
- Datenpunkt
- Host
- Host-Gruppe
- Vorlage
- Vorlagengruppe

Quelle

CGraphPrototype::get() in *ui/include/classes/api/services/CGraphPrototype.php*.

graphprototype.update

Beschreibung

`object graphprototype.update(object/array graphPrototypes)`

Diese Methode ermöglicht es Ihnen, vorhandene Diagrammvorlagen zu aktualisieren.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu aktualisierende Eigenschaften des Graphprototyps.

Die Eigenschaft `graphid` muss für jeden Graphprototyp definiert sein, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Zusätzlich zu den Standard-Eigenschaften des Graphprototyps akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
gitems	array	Graph-Elemente zum Ersetzen vorhandener Graph-Elemente. Wenn für ein Graph-Element die Eigenschaft `gitemid` definiert ist, wird es aktualisiert, andernfalls wird ein neues Graph-Element erstellt.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Graphenprototypen unter der Eigenschaft `graphids` enthält.

Beispiele

Ändern der Größe eines Graphenprototyps

Ändern Sie die Größe eines Graphenprototyps auf 1100 x 400 Pixel.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.update",
    "params": {
        "graphid": "439",
        "width": 1100,
        "height": 400
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "439"
        ]
    },
    "id": 1
}
```

Quelle

CGraphPrototype::update() in *ui/include/classes/api/services/CGraphPrototype.php*.

Hochverfügbarkeitsknoten

Diese Klasse ist für die Arbeit mit Server-Knoten konzipiert, die Teil eines Hochverfügbarkeitsclusters oder einer eigenständigen Server-Instanz sind.

Objektreferenzen:

- Hochverfügbarkeitsknoten

Verfügbare Methoden:

- hanode.get - Knoten abrufen

Objekt für Hochverfügbarkeitsknoten

Das folgende Objekt steht im Zusammenhang mit dem Betrieb eines Hochverfügbarkeitsclusters von Zabbix-Servern.

High availability node

> **Note:**
> Nodes are created by the Zabbix Server and cannot be modified via the API.

Das Objekt "High availability node" hat die folgenden Eigenschaften.

Property	Type	Description
ha_nodeid	ID	ID des Knotens.
name	string	Dem Knoten zugewiesener Name, unter Verwendung des Konfigurationseintrags HANodeName in `zabbix_server.conf`. Leer bei einem Server, der im Standalone-Modus läuft.
address	string	IP- oder DNS-Name, von dem aus der Knoten sich verbindet.
port	integer	Port, auf dem der Knoten läuft.

Property	Type	Description
lastaccess	integer	Heartbeat-Zeit, also Zeitpunkt der letzten Aktualisierung vom Knoten. UTC-Zeitstempel.
status	integer	Status des Knotens. Mögliche Werte: 0 - standby; 1 - manuell gestoppt; 2 - nicht verfügbar; 3 - aktiv.

hanode.get

Beschreibung

`integer/array hanode.get(object parameters)`

Die Methode ermöglicht es Ihnen, eine Liste von Knoten eines High-Availability-Clusters anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist nur für Benutzertypen vom Typ *Super admin* verfügbar. Weitere Informationen finden Sie unter User roles.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
ha_nodeids	ID/array	Gibt nur Knoten mit den angegebenen Knoten-IDs zurück.
filter	object	Gibt nur die Ergebnisse zurück, die exakt mit dem angegebenen Filter übereinstimmen. Akzeptiert ein Objekt, bei dem die Schlüssel Eigenschaftsnamen sind und die Werte entweder ein einzelner Wert oder ein Array von Werten sind, mit denen abgeglichen werden soll. Unterstützte Eigenschaften: `name`, `address`, `status`.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `name`, `lastaccess`, `status`.
countOutput	flag	Diese Parameter werden im Referenzkommentar beschrieben.
limit	integer	
output	query	
preservekeys	boolean	
sortorder	string/array	

Rückgabewerte

`(integer/array)` Kann die folgenden Dinge zurück geben:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Eine nach Status sortierte Liste von Knoten abrufen

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "preservekeys": true,
        "sortfield": "status",
        "sortorder": "DESC"
    },
    "id": 1
}
```

Antwort:

```
{
  "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
}
```

Eine Liste bestimmter Knoten anhand ihrer IDs abrufen

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "ha_nodeids": ["ckuo7i1nw000e0sajwfttc1mp", "ckuo7i1nv000c0sajz85xcrtt"]
    },
    "id": 1
}
```

Antwort:


```
{
    "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
}
```


Eine Liste gestoppter Knoten abrufen

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "output": ["ha_nodeid", "address", "port"],
        "filter": {
            "status": 1
        }
    },
    "id": 1
}
```

Antwort:


```
{
    "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
}
```


Anzahl der Standby-Knoten abrufen

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "countOutput": true,
        "filter": {
            "status": 0
        }
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": "3",
    "id": 1
}
```

Status von Knoten an bestimmten IP-Adressen prüfen

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "output": ["name", "status"],
        "filter": {
            "address": ["192.168.1.7", "192.168.1.13"]
        }
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "name": "node3",
            "status": "0"
        },
        {
            "name": "node-active",
            "status": "3"
        }
    ],
    "id": 1
}
```

Quelle

CHaNode::get() in *ui/include/classes/api/services/CHaNode.php*.

Host

Diese Klasse ist für die Arbeit mit Hosts konzipiert.

Objektreferenzen:

- Host
- Host-Inventar
- Host-Tag

Verfügbare Methoden:

- host.create - neue Hosts erstellen
- host.delete - Hosts löschen
- host.get - Hosts abrufen
- host.massadd - verknüpfte Objekte zu Hosts hinzufügen
- host.massremove - verknüpfte Objekte von Hosts entfernen
- host.update - Hosts aktualisieren

Host-Objekt

Die folgenden Objekte beziehen sich direkt auf die Host-API.

Host

Das Host-Objekt verfügt über die folgenden Eigenschaften.

Property	Type	Description
hostid	ID	ID des Hosts. Eigenschaftsverhalten: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
host	string	Technischer Name des Hosts. Eigenschaftsverhalten: - *erforderlich* für Erstellungsvorgänge
description	text	Beschreibung des Hosts.
flags	integer	Ursprung des Hosts. Mögliche Werte: 0 - ein einfacher Host; 4 - ein aus einem Prototyp konvertierter Host. Eigenschaftsverhalten: - *schreibgeschützt*
inventory_mode	integer	Modus zum Befüllen des Host-Inventars. Mögliche Werte: -1 - *(Standard)* deaktiviert; 0 - manuell; 1 - automatisch.
ipmi_authtype	integer	IPMI-Authentifizierungsalgorithmus. Mögliche Werte: -1 - *(Standard)* Standard; 0 - keiner; 1 - MD2; 2 - MD5 4 - direkt; 5 - OEM; 6 - RMCP+.
ipmi_password	string	IPMI-Passwort.
ipmi_privilege	integer	IPMI-Berechtigungsstufe. Mögliche Werte: 1 - Callback; 2 - *(Standard)* Benutzer; 3 - Operator; 4 - Administrator; 5 - OEM.
ipmi_username	string	IPMI-Benutzername.

Property	Type	Description
maintenance_from	timestamp	Startzeit der aktiven Wartung. Eigenschaftsverhalten: - *schreibgeschützt*
maintenance_status	integer	Status der aktiven Wartung. Mögliche Werte: 0 - *(Standard)* keine Wartung; 1 - Wartung aktiv. Eigenschaftsverhalten: - *schreibgeschützt*
maintenance_type	integer	Typ der aktiven Wartung. Mögliche Werte: 0 - *(Standard)* Wartung mit Datensammlung; 1 - Wartung ohne Datensammlung. Eigenschaftsverhalten: - *schreibgeschützt*
maintenanceid	ID	ID der Wartung, die derzeit für den Host aktiv ist. Eigenschaftsverhalten: - *schreibgeschützt*
name	string	Sichtbarer Name des Hosts. Standard: Wert der Eigenschaft `host`.
monitored_by	integer	Quelle, die zur Überwachung des Hosts verwendet wird. Mögliche Werte: 0 - *(Standard)* Zabbix-Server; 1 - Proxy; 2 - Proxy-Gruppe. Eigenschaftsverhalten: - *erforderlich*, wenn `select_server_for_monitoring` auf „Deaktiviert“ gesetzt ist.
proxyid	ID	ID des Proxy, der zur Überwachung des Hosts verwendet wird. Eigenschaftsverhalten: - *erforderlich*, wenn `monitored_by` auf „Proxy“ gesetzt ist
proxy_groupid	ID	ID der Proxy-Gruppe, die zur Überwachung des Hosts verwendet wird. Eigenschaftsverhalten: - *erforderlich*, wenn `monitored_by` auf „Proxy-Gruppe“ gesetzt ist
status	integer	Status und Funktion des Hosts. Mögliche Werte: 0 - *(Standard)* aktiviert; 1 - deaktiviert.
tls_connect	integer	Verbindungen zum Host. Mögliche Werte: 1 - *(Standard)* keine Verschlüsselung; 2 - PSK; 4 - Zertifikat.

Property	Type	Description
tls_accept	integer	Verbindungen vom Host. Mögliche Bitmap-Werte: 1 - *(Standard)* keine Verschlüsselung; 2 - PSK; 4 - Zertifikat. Dies ist ein Bitmaskenfeld; jede Summe möglicher Bitmap-Werte ist zulässig (zum Beispiel 6 für PSK und Zertifikat).
tls_issuer	string	Aussteller des Zertifikats.
tls_subject	string	Betreff des Zertifikats.
tls_psk_identity	string	PSK-Identität; darf nur mit genau einer PSK verknüpft sein (über Autoregistrierung, Hosts und Proxys hinweg). Fügen Sie keine vertraulichen Informationen in die PSK-Identität ein, da sie unverschlüsselt über das Netzwerk gesendet wird, um dem Empfänger mitzuteilen, welche PSK verwendet werden soll. Eigenschaftsverhalten: - *nur schreibbar* - *erforderlich*, wenn `tls_connect` auf „PSK“ gesetzt ist oder `tls_accept` das Bit „PSK“ enthält
tls_psk	string	Pre-Shared Key (PSK); muss mindestens 32 Hexadezimalziffern enthalten. Eigenschaftsverhalten: - *nur schreibbar* - *erforderlich*, wenn `tls_connect` auf „PSK“ gesetzt ist oder `tls_accept` das Bit „PSK“ enthält
active_available	integer	Verfügbarkeitsstatus der aktiven Host-Schnittstelle. Mögliche Werte: 0 - Schnittstellenstatus ist unbekannt; 1 - Schnittstelle ist verfügbar; 2 - Schnittstelle ist nicht verfügbar. Eigenschaftsverhalten: - *schreibgeschützt*
assigned_proxyid	ID	ID des vom Zabbix-Server zugewiesenen Proxy, wenn der Host von einer Proxy-Gruppe überwacht wird. Eigenschaftsverhalten: - *schreibgeschützt*

Host-Inventar

Das Host-Inventarobjekt hat die folgenden Eigenschaften.

Note:
Jede Eigenschaft hat ihre eigene eindeutige ID-Nummer, die verwendet wird, um Host-Inventarfelder mit Datenpunkten zu verknüpfen.

ID	Eigenschaft	Type	Beschreibung	Maximale Länge
4	alias	string	Alias.	128 Zeichen
11	asset_tag	string	Asset-Tag.	64 Zeichen
28	chassis	string	Gehäuse.	64 Zeichen
23	contact	string	Ansprechpartner.	65535 Zeichen für SQL-Datenbanken
32	contract_number	string	Vertragsnummer.	64 Zeichen
47	date_hw_decomm	string	Datum der HW-Außerbetriebnahme.	64 Zeichen
46	date_hw_expiry	string	Ablaufdatum der HW-Wartung.	64 Zeichen

ID	Eigenschaft	Type	Beschreibung	Maximale Länge
45	date_hw_install	string	Datum der HW-Installation.	64 Zeichen
44	date_hw_purchase	string	Datum des HW-Kaufs.	64 Zeichen
34	deployment_status	string	Bereitstellungsstatus.	64 Zeichen
14	hardware	string	Hardware.	255 Zeichen
15	hardware_full	string	Detaillierte Hardware.	65535 Zeichen für SQL-Datenbanken
39	host_netmask	string	Host-Subnetzmaske.	39 Zeichen
38	host_networks	string	Host-Netzwerke.	65535 Zeichen für SQL-Datenbanken
40	host_router	string	Host-Router.	39 Zeichen
30	hw_arch	string	HW-Architektur.	32 Zeichen
33	installer_name	string	Name des Installateurs.	64 Zeichen
24	location	string	Standort.	65535 Zeichen für SQL-Datenbanken
25	location_lat	string	Standortbreite.	16 Zeichen
26	location_lon	string	Standortlänge.	16 Zeichen
12	macaddress_a	string	MAC-Adresse A.	64 Zeichen
13	macaddress_b	string	MAC-Adresse B.	64 Zeichen
29	model	string	Modell.	64 Zeichen
3	name	string	Name.	128 Zeichen
27	notes	string	Notizen.	65535 Zeichen für SQL-Datenbanken
41	oob_ip	string	OOB-IP-Adresse.	39 Zeichen
42	oob_netmask	string	OOB-Host-Subnetzmaske.	39 Zeichen
43	oob_router	string	OOB-Router.	39 Zeichen
5	os	string	Name des Betriebssystems.	128 Zeichen
6	os_full	string	Vollständiger Name des Betriebssystems.	255 Zeichen
7	os_short	string	Kurzer Name des Betriebssystems.	128 Zeichen
61	poc_1_cell	string	Mobilnummer des primären POC.	64 Zeichen
58	poc_1_email	string	Primäre E-Mail.	128 Zeichen
57	poc_1_name	string	Name des primären POC.	128 Zeichen
63	poc_1_notes	string	Notizen zum primären POC.	65535 Zeichen für SQL-Datenbanken
59	poc_1_phone_a	string	Telefon A des primären POC.	64 Zeichen
60	poc_1_phone_b	string	Telefon B des primären POC.	64 Zeichen
62	poc_1_screen	string	Bildschirmname des primären POC.	64 Zeichen
68	poc_2_cell	string	Mobilnummer des sekundären POC.	64 Zeichen
65	poc_2_email	string	Sekundäre E-Mail.	128 Zeichen
64	poc_2_name	string	Name des sekundären POC.	128 Zeichen
70	poc_2_notes	string	Notizen zum sekundären POC.	65535 Zeichen für SQL-Datenbanken
66	poc_2_phone_a	string	Telefon A des sekundären POC.	64 Zeichen
67	poc_2_phone_b	string	Telefon B des sekundären POC.	64 Zeichen
69	poc_2_screen	string	Bildschirmname des sekundären POC.	64 Zeichen
8	serialno_a	string	Seriennummer A.	64 Zeichen
9	serialno_b	string	Seriennummer B.	64 Zeichen
48	site_address_a	string	Standortadresse A.	128 Zeichen
49	site_address_b	string	Standortadresse B.	128 Zeichen
50	site_address_c	string	Standortadresse C.	128 Zeichen
51	site_city	string	Standortstadt.	128 Zeichen
53	site_country	string	Standortland.	64 Zeichen
56	site_notes	string	Standortnotizen.	65535 Zeichen für SQL-Datenbanken
55	site_rack	string	Rack-Position am Standort.	128 Zeichen
52	site_state	string	Standortbundesland/-region.	64 Zeichen
54	site_zip	string	PLZ/Postleitzahl des Standorts.	64 Zeichen
16	software	string	Software.	255 Zeichen
18	software_app_a	string	Softwareanwendung A.	64 Zeichen
19	software_app_b	string	Softwareanwendung B.	64 Zeichen
20	software_app_c	string	Softwareanwendung C.	64 Zeichen
21	software_app_d	string	Softwareanwendung D.	64 Zeichen
22	software_app_e	string	Softwareanwendung E.	64 Zeichen
17	software_full	string	Softwaredetails.	65535 Zeichen für SQL-Datenbanken
10	tag	string	Tag.	64 Zeichen
1	type	string	Typ.	64 Zeichen
2	type_full	string	Typdetails.	64 Zeichen
35	url_a	string	URL A.	2048 Zeichen

ID	Eigenschaft	Type	Beschreibung	Maximale Länge
36	url_b	string	URL B.	2048 Zeichen
37	url_c	string	URL C.	2048 Zeichen
31	vendor	string	Anbieter.	64 Zeichen

Host-Tag

Das Host-Tag-Objekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
tag	string	Name des Host-Tags. Verhalten der Eigenschaft: - *erforderlich*
value	string	Wert des Host-Tags. Verhalten der Eigenschaft: - *schreibgeschützt*.
automatic	integer	Typ des Host-Tags. Mögliche Werte: 0 - *(Standard)* manuell (Tag wurde vom Benutzer erstellt); 1 - automatisch (Tag wurde durch Low-Level-Discovery erstellt) Verhalten der Eigenschaft: - *schreibgeschützt*
object	integer	Typ des Objekts, von dem das Tag geerbt wurde. Mögliche Werte: 0 - Vorlage. Verhalten der Eigenschaft: - *schreibgeschützt*.
objectid	ID	ID des Objekts, von dem das Tag geerbt wurde. Verhalten der Eigenschaft: - *schreibgeschützt*.

host.create

Beschreibung

`object host.create(object/array hosts)`

Diese Methode ermöglicht es Ihnen, neue Hosts zu erstellen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellende Hosts.

Zusätzlich zu den standardmäßigen Host-Eigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Typ	Beschreibung
groups	object/array	Host-Gruppen, zu denen der Host hinzugefügt werden soll. Für die Host-Gruppen darf nur die Eigenschaft `groupid` definiert sein. Parameterverhalten: - *erforderlich*
interfaces	object/array	Schnittstellen, die für den Host erstellt werden sollen.
tags	object/array	Host-Tags.
templates	object/array	Vorlagen, die mit dem Host verknüpft werden sollen. Für die Vorlagen darf nur die Eigenschaft `templateid` definiert sein.
macros	object/array	Benutzermakros, die für den Host erstellt werden sollen.
inventory	object	Eigenschaften des Host-Inventars.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Hosts unter der Eigenschaft `hostids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Hosts.

Beispiele

Erstellen eines Hosts

Erstellen Sie einen Host mit dem Namen „Linux server“ mit einer IP-Schnittstelle und Tags, fügen Sie ihn einer Gruppe hinzu, verknüpfen Sie eine Vorlage mit ihm und legen Sie die MAC-Adressen in der Host-Inventarisierung fest.

Anfrage:

```
{
    "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": "Breiten-, Längen- und Höhenkoordinaten"
            }
        ],
        "inventory_mode": 0,
        "inventory": {
            "macaddress_a": "01234",
            "macaddress_b": "56768"
        }
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "107819"
        ]
    },
    "id": 1
}
```

Erstellen eines Hosts mit SNMP-Schnittstelle

Erstellen Sie einen Host namens „SNMP host“ mit einer SNMPv3-Schnittstelle mit Details.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10658"
        ]
    },
    "id": 1
}
```

Erstellen eines Hosts mit PSK-Verschlüsselung

Erstellen Sie einen Host mit dem Namen "PSK host", für den PSK-Verschlüsselung konfiguriert ist. Beachten Sie, dass der Host vorab für die Verwendung von PSK konfiguriert sein muss.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10590"
        ]
    },
    "id": 1
}
```

Erstellen eines von einem Proxy überwachten Hosts

Erstellen Sie einen Host, der von einem Proxy mit der ID „1“ überwacht wird.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
```

```
        "host": "Host monitored by proxy",
        "groups": [
            {
                "groupid": "2"
            }
        ],
        "monitored_by": 1,
        "proxyid": 1
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10591"
        ]
    },
    "id": 1
}
```

Erstellen eines von einer Proxy-Gruppe überwachten Hosts

Erstellen Sie einen Host, der von der Proxy-Gruppe mit der ID „1“ überwacht wird.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "Host monitored by proxy group",
        "groups": [
            {
                "groupid": "2"
            }
        ],
        "monitored_by": 2,
        "proxy_groupid": 1
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10592"
        ]
    },
    "id": 1
}
```

Siehe auch

- Host-Gruppe
- Vorlage
- Benutzermakro
- Host-Schnittstelle
- Host-Inventar
- Host-Tag
- Proxy

- Proxy-Gruppe

Quelle

CHost::create() in *ui/include/classes/api/services/CHost.php*.

host.delete

Beschreibung

`object host.delete(array hosts)`

Diese Methode ermöglicht es Ihnen, Hosts zu löschen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der zu löschenden Hosts.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Hosts unter der Eigenschaft `hostids` enthält.

Beispiele

Mehrere Hosts löschen

Löschen Sie zwei Hosts.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "host.delete",
    "params": [
        "13",
        "32"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "13",
            "32"
        ]
    },
    "id": 1
}
```

Quelle

CHost::delete() in *ui/include/classes/api/services/CHost.php*.

host.get

Beschreibung

`integer/array host.get(object parameters)`

Die Methode ermöglicht es Ihnen, Hosts anhand der angegebenen Parameter abzurufen.

Note:
Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
groupids	ID/array	Gibt nur Hosts zurück, die zu den angegebenen Gruppen gehören.
dserviceids	ID/array	Gibt nur Hosts zurück, die mit den angegebenen erkannten Services verknüpft sind.
graphids	ID/array	Gibt nur Hosts zurück, die die angegebenen Graphen haben.
hostids	ID/array	Gibt nur Hosts mit den angegebenen Host-IDs zurück.
httptestids	ID/array	Gibt nur Hosts zurück, die die angegebenen Webprüfungen haben.
interfaceids	ID/array	Gibt nur Hosts zurück, die die angegebenen Schnittstellen verwenden.
itemids	ID/array	Gibt nur Hosts zurück, die die angegebenen Datenpunkte haben.
maintenanceids	ID/array	Gibt nur Hosts zurück, die von den angegebenen Wartungen betroffen sind.
monitored_hosts	flag	Gibt nur überwachte Hosts zurück.
proxyids	ID/array	Gibt nur Hosts zurück, die von den angegebenen Proxys überwacht werden.
proxy_groupids	ID/array	Gibt nur Hosts zurück, die von den angegebenen Proxy-Gruppen überwacht werden.
templated_hosts	flag	Gibt sowohl Hosts als auch Vorlagen zurück.
templateids	ID/array	Gibt nur Hosts zurück, die mit den angegebenen Vorlagen verknüpft sind.
triggerids	ID/array	Gibt nur Hosts zurück, die die angegebenen Auslöser haben.
with_items	flag	Gibt nur Hosts zurück, die Datenpunkte haben. Überschreibt die Parameter `with_monitored_items` und `with_simple_graph_items`.
with_item_prototypes	flag	Gibt nur Hosts zurück, die Datenpunktprototypen haben. Überschreibt den Parameter `with_simple_graph_item_prototypes`.
with_simple_graph_item_prototypes	flag	Gibt nur Hosts zurück, die Datenpunktprototypen haben, die für die Erstellung aktiviert sind und einen numerischen Informationstyp haben.
with_graphs	flag	Gibt nur Hosts zurück, die Graphen haben.
with_graph_prototypes	flag	Gibt nur Hosts zurück, die Graphprototypen haben.
with_httptests	flag	Gibt nur Hosts zurück, die Webprüfungen haben. Überschreibt den Parameter `with_monitored_httptests`.
with_monitored_httptests	flag	Gibt nur Hosts zurück, die aktivierte Webprüfungen haben.
with_monitored_items	flag	Gibt nur Hosts zurück, die aktivierte Datenpunkte haben. Überschreibt den Parameter `with_simple_graph_items`.
with_monitored_triggers	flag	Gibt nur Hosts zurück, die aktivierte Auslöser haben. Alle im Auslöser verwendeten Datenpunkte müssen ebenfalls aktiviert sein.
with_simple_graph_items	flag	Gibt nur Hosts zurück, die Datenpunkte mit numerischem Informationstyp haben.
with_triggers	flag	Gibt nur Hosts zurück, die Auslöser haben. Überschreibt den Parameter `with_monitored_triggers`.
withProblemsSuppressed	boolean	Wenn auf `true` gesetzt, werden nur Hosts mit unterdrückten Problemen zurückgegeben.

Parameter	Type	Beschreibung
evaltype	integer	Auswertungsmethode für Tags. Mögliche Werte: 0 - *(Standard)* Und/Oder; 2 - Oder.
severities	integer/array	Gibt Hosts zurück, die nur Probleme mit den angegebenen Schweregraden haben. Gilt nur, wenn das Problemobjekt ein Auslöser ist.
tags	object/array	Gibt nur Hosts mit den angegebenen Tags zurück. Format: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Ein leeres Array gibt alle Hosts zurück. Mögliche Werte für operator: 0 - *(Standard)* Enthält; 1 - Gleich; 2 - Enthält nicht; 3 - Ungleich; 4 - Existiert; 5 - Existiert nicht.
inheritedTags	boolean	Gibt Hosts zurück, die die angegebenen `tags` auch in allen mit ihnen verknüpften Vorlagen haben. Mögliche Werte: `true` - verknüpfte Vorlagen müssen die angegebenen `tags` ebenfalls haben; `false` - *(Standard)* Tags verknüpfter Vorlagen werden ignoriert.
selectDiscoveryData	query	Gibt eine Eigenschaft `discoveryData` mit den Objektdaten der Host-Erkennung zurück. Das Host-Erkennungsobjekt verknüpft einen erkannten Host mit einem Hostprototyp, über den er erkannt wurde. Es hat die folgenden Eigenschaften: `host` - `(string)` ID des Hosts; `parent_hostid` - `(string)` ID des Hostprototyps, aus dem der Host erstellt wurde; `status` - `(int)` Status der Host-Erkennung: 0 - *(Standard)* Host ist erkannt, 1 - Host ist nicht mehr erkannt; `ts_delete` - `(timestamp)` Zeitpunkt, zu dem ein nicht mehr erkannter Host gelöscht wird; `ts_disable` - `(timestamp)` Zeitpunkt, zu dem ein nicht mehr erkannter Host deaktiviert wird; `disable_source` - `(int)` Kennzeichen dafür, ob der Host durch eine LLD-Regel oder manuell deaktiviert wurde: 0 - *(Standard)* automatisch deaktiviert, 1 - durch eine LLD-Regel deaktiviert.
selectDiscoveryRule	query	Gibt eine Eigenschaft `discoveryRule` mit der Low-Level-Discovery-Regel zurück, die den Host erstellt hat (aus dem Hostprototyp in der VMware-Überwachung).
selectDiscoveryRules	query	Gibt eine Eigenschaft `discoveryRules` mit Host-LLD-Regeln zurück. Unterstützt `count`.
selectGraphs	query	Gibt eine Eigenschaft `graphs` mit Host-Graphen zurück. Unterstützt `count`.
selectHostGroups	query	Gibt eine Eigenschaft `hostgroups` mit den Daten der Host-Gruppen zurück, zu denen der Host gehört.
selectHttpTests	query	Gibt eine Eigenschaft `httpTests` mit Webszenarien des Hosts zurück. Unterstützt `count`.

Parameter	Type	Beschreibung
selectInterfaces	query	Gibt eine Eigenschaft `interfaces` mit Host-Schnittstellen zurück. Unterstützt `count`.
selectInventory	query	Gibt eine Eigenschaft `inventory` mit den Inventardaten des Hosts zurück.
selectItems	query	Gibt eine Eigenschaft `items` mit Host-Datenpunkten zurück. Unterstützt `count`.
selectMacros	query	Gibt eine Eigenschaft `macros` mit Host-Makros zurück.
selectParentTemplates	query	Gibt eine Eigenschaft `parentTemplates` mit Vorlagen zurück, mit denen der Host verknüpft ist. Zusätzlich zu den Feldern des Vorlagenobjekts enthält sie `link_type` - `(integer)` die Art, wie die Vorlage mit dem Host verknüpft ist. Mögliche Werte: 0 - *(Standard)* manuell verknüpft; 1 - automatisch durch LLD verknüpft. Unterstützt `count`.
selectDashboards	query	Gibt eine Eigenschaft `dashboards` zurück. Unterstützt `count`.
selectTags	query	Gibt eine Eigenschaft `tags` mit Host-Tags zurück.
selectInheritedTags	query	Gibt eine Eigenschaft `inheritedTags` mit Tags zurück, die auf allen mit dem Host verknüpften Vorlagen vorhanden sind.
selectTriggers	query	Gibt eine Eigenschaft `triggers` mit Host-Auslösern zurück. Unterstützt `count`.
selectValueMaps	query	Gibt eine Eigenschaft `valuemaps` mit Host-Wertzuordnungen zurück.
filter	object	Gibt nur Ergebnisse zurück, die exakt dem angegebenen Filter entsprechen. Akzeptiert ein Objekt, bei dem die Schlüssel Eigenschaftsnamen sind und die Werte entweder ein einzelner Wert oder ein Array von Werten sind, mit denen abgeglichen werden soll. Unterstützt keine Eigenschaften vom Datentyp `text`. Unterstützt zusätzliche Eigenschaften: Eigenschaften des Host interface.
limitSelects	integer	Begrenzt die Anzahl der von Unterabfragen zurückgegebenen Datensätze. Gilt für die folgenden Unterabfragen: `selectParentTemplates` - Ergebnisse werden nach `host` sortiert; `selectInterfaces`; `selectItems` - sortiert nach `name`; `selectDiscoveryRules` - sortiert nach `name`; `selectTriggers` - sortiert nach `description`; `selectGraphs` - sortiert nach `name`; `selectDashboards` - sortiert nach `name`.

Parameter	Type	Beschreibung
search	object	Gibt Ergebnisse zurück, die dem angegebenen Muster entsprechen (Groß-/Kleinschreibung wird nicht beachtet). Akzeptiert ein Objekt, bei dem die Schlüssel Eigenschaftsnamen sind und die Werte Zeichenfolgen sind, nach denen gesucht werden soll. Wenn keine zusätzlichen Optionen angegeben werden, wird eine Suche vom Typ `LIKE "%…%"` durchgeführt. Unterstützt nur Eigenschaften vom Datentyp `string` und `text`. Unterstützt zusätzliche Eigenschaften: Eigenschaften des Host interface.
searchInventory	object	Gibt Hosts zurück, deren Inventardaten dem angegebenen Muster entsprechen (Groß-/Kleinschreibung wird nicht beachtet). Akzeptiert ein Objekt, bei dem die Schlüssel Eigenschaftsnamen sind und die Werte Zeichenfolgen sind, nach denen gesucht werden soll. Wenn keine zusätzlichen Optionen angegeben werden, wird eine Suche vom Typ `LIKE "%…%"` durchgeführt. Unterstützt nur Eigenschaften vom Datentyp `string` und `text`.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `hostid`, `host`, `name`, `status`.
countOutput	boolean	Diese Parameter sind in der Referenzkommentierung beschrieben.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectDiscoveries	query	Gibt eine Eigenschaft `discoveries` mit Low-Level-Discovery-Regeln des Hosts zurück. Unterstützt `count`.
selectHostDiscovery	query	Diese Abfrage ist **veraltet**, bitte verwenden Sie stattdessen `selectDiscoveryRules`. Gibt eine Eigenschaft `hostDiscovery` mit den Objektdaten der Host-Erkennung zurück. Diese Abfrage ist **veraltet**, bitte verwenden Sie stattdessen `selectDiscoveryData`.

Rückgabewerte

`(integer/array)` Kann die folgenden Dinge zurück geben:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Daten nach Namen abrufen

Rufen Sie alle Daten zu zwei Hosts mit den Namen „Zabbix server“ und „Linux server“ ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "host.get",
    "params": {
        "filter": {
            "host": [
                "Zabbix server",
                "Linux server"
            ]
        }
    },
    "id": 1
}
```

Antwort:

```
{
    "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": "Der Zabbix-Monitoring-Server.",
            "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
}
```

Abrufen von Hostgruppen

Rufen Sie die Hostgruppen ab, deren Mitglied der Host „Zabbix server“ ist.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectHostGroups": "extend",
        "filter": {
            "host": [
                "Zabbix server"
            ]
        }
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```

Verknüpfte Vorlagen abrufen

Rufen Sie die IDs und Namen der mit dem Host „10084“ verknüpften Vorlagen ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
```

```
        "output": ["hostid"],
        "selectParentTemplates": [
            "templateid",
            "name"
        ],
        "hostids": "10084"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10084",
            "parentTemplates": [
                {
                    "name": "Linux",
                    "templateid": "10001"
                },
                {
                    "name": "Zabbix Server",
                    "templateid": "10047"
                }
            ]
        }
    ],
    "id": 1
}
```

Abrufen von Hosts nach Vorlage

Rufen Sie Hosts ab, mit denen die Vorlage „10001“ (*Linux by Zabbix Agent*) verknüpft ist.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid", "name"],
        "templateids": "10001"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10001",
            "hosts": [
                {
                    "hostid": "10084",
                    "name": "Zabbix server"
                },
                {
                    "hostid": "10603",
                    "name": "Host 1"
                },
                {
                    "hostid": "10604",
```

```
                    "name": "Host 2"
                }
            ]
        }
    ],
    "id": 1
}
```

Suche nach Host-Inventardaten

Rufen Sie Hosts ab, die „Linux“ im Host-Inventarfeld „OS“ enthalten.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": [
            "host"
        ],
        "selectInventory": [
            "os"
        ],
        "searchInventory": {
            "os": "Linux"
        }
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10084",
            "host": "Zabbix server",
            "inventory": {
                "os": "Linux Ubuntu"
            }
        },
        {
            "hostid": "10107",
            "host": "Linux server",
            "inventory": {
                "os": "Linux Mint"
            }
        }
    ],
    "id": 1
}
```

Suche nach Host-Tags

Rufen Sie Hosts ab, deren Tag „host-name“ gleich „linux-server“ ist.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectTags": "extend",
        "evaltype": 0,
```

```
        "tags": [
            {
                "tag": "host-name",
                "value": "linux-server",
                "operator": 1
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10085",
            "tags": [
                {
                    "tag": "host-name",
                    "value": "linux-server",
                    "automatic": "0"
                },
                {
                    "tag": "os",
                    "value": "rhel-7",
                    "automatic": "0"
                }
            ]
        }
    ],
    "id": 1
}
```

Rufen Sie Hosts ab, die diese Tags nicht nur auf Host-Ebene, sondern auch in ihren verknüpften übergeordneten Vorlagen haben.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "tags": [
            {
                "tag": "os",
                "value": "rhel-7",
                "operator": 1
            }
        ],
        "inheritedTags": true
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10623",
            "name": "PC room 1"
        },
```

```
        {
            "hostid": "10601",
            "name": "Office"
        }
    ],
    "id": 1
}
```

Rufen Sie Hosts ab, ausgenommen diejenigen mit dem Tag „component“.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectTags": "extend",
        "evaltype": 0,
        "tags": [
            {
                "tag": "component",
                "value": "",
                "operator": 5
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10084",
            "tags": [
                {
                    "tag": "class",
                    "value": "os",
                    "automatic": "0"
                },
                {
                    "tag": "target",
                    "value": "linux",
                    "automatic": "0"
                }
            ]
        },
        {
            "hostid": "10629",
            "tags": [
                {
                    "tag": "class",
                    "value": "application",
                    "automatic": "0"
                },
                {
                    "tag": "target",
                    "value": "browser",
                    "automatic": "0"
                }
            ]
        }
```

```
    ],
    "id": 1
}
```

Host mit Tags und Vorlagen-Tags suchen

Rufen Sie einen Host mit Tags und allen Tags ab, die mit übergeordneten Vorlagen verknüpft sind.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "hostids": 10502,
        "selectTags": ["tag", "value"],
        "selectInheritedTags": ["tag", "value"]
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```

Hosts nach Problem-Schweregrad suchen

Rufen Sie Hosts ab, die Probleme mit dem Schweregrad „Katastrophal“ haben.

Anfrage:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "host.get",
    "params": {
        "output": ["name"],
        "severities": 5
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10160",
            "name": "Zabbix server"
        }
    ],
    "id": 1
}
```

Rufen Sie Hosts ab, die Probleme mit den Schweregraden „Durchschnittlich“ und „Hoch“ haben.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "severities": [3, 4]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "20170",
            "name": "Database"
        },
        {
            "hostid": "20183",
            "name": "workstation"
        }
    ],
    "id": 1
}
```

Abrufen von Host-Datenpunkten

Rufen Sie Datenpunkte für den Host „Zabbix server“ ab. Die Anfrage ist auf die Datenpunkt-Eigenschaften `itemid`, `name` und `status` beschränkt. Aufgrund der großen Antwortgröße wird im Beispiel nur eine Teilmenge der Datenpunkte angezeigt.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectItems": ["itemid","name","status"],
        "filter": {
            "host": [
```

```
                "Zabbix server"
            ]
        }
    },
    "id": 1
}
```

Antwort:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "hostid": "10084",
      "items": [
        {
          "itemid": "42227",
          "name": "Host-Name des laufenden Zabbix Agent",
          "status": "0"
        },
        {
          "itemid": "42237",
          "name": "Zabbix Agent ping",
          "status": "0"
        },
        {
          "itemid": "42250",
          "name": "Version des laufenden Zabbix Agent",
          "status": "0"
        },
        {
          "itemid": "42251",
          "name": "Maximale Anzahl offener Dateideskriptoren",
          "status": "0"
        },
        {
          "itemid": "42252",
          "name": "Maximale Anzahl von Prozessen",
          "status": "0"
        },
        {
          "itemid": "69869",
          "name": "Schnittstelle enp0s3: Verworfen eingehende Pakete",
          "status": "0"
        },
        {
          "itemid": "69870",
          "name": "Schnittstelle enp0s3: Eingehende Pakete mit Fehlern",
          "status": "0"
        },
        {
          "itemid": "69871",
          "name": "Schnittstelle enp0s3: Empfangene Bits",
          "status": "0"
        },
        {
          "itemid": "69872",
          "name": "Schnittstelle enp0s3: Verworfen ausgehende Pakete",
          "status": "0"
        },
        {
          "itemid": "69873",
          "name": "Schnittstelle enp0s3: Ausgehende Pakete mit Fehlern",
```

```
            "status": "0"
        },
        {
            "itemid": "69874",
            "name": "Schnittstelle enp0s3: Gesendete Bits",
            "status": "0"
        },
        {
            "itemid": "42253",
            "name": "Anzahl der Prozesse",
            "status": "0"
        },
        {
            "itemid": "42254",
            "name": "Anzahl der laufenden Prozesse",
            "status": "0"
        },
        {
            "itemid": "42255",
            "name": "Systemstartzeit",
            "status": "0"
        }
      ]
    }
  ],
  "id": 1
}
```


Siehe auch

- Host-Gruppe
- Vorlage
- Benutzermakro
- Host-Schnittstelle
- Proxy
- Proxy-Gruppe

Quelle

CHost::get() in *ui/include/classes/api/services/CHost.php*.

host.massadd

Beschreibung

`object host.massadd(object parameters)`

Diese Methode ermöglicht es Ihnen, mehrere zusammengehörige Objekte gleichzeitig zu allen angegebenen Hosts hinzuzufügen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die IDs der zu aktualisierenden Hosts und die Objekte enthalten, die zu allen Hosts hinzugefügt werden sollen.

Die Methode akzeptiert die folgenden Parameter.

Parameter	Type	Beschreibung
hosts	object/array	Hosts, die aktualisiert werden sollen. Für die Hosts darf nur die Eigenschaft `hostid` definiert sein. Parameterverhalten: - *erforderlich*
groups	object/array	Host-Gruppen, die zu den angegebenen Hosts hinzugefügt werden sollen. Für die Host-Gruppen darf nur die Eigenschaft `groupid` definiert sein.
interfaces	object/array	Host-Schnittstellen, die für die angegebenen Hosts erstellt werden sollen.
macros	object/array	Benutzermakros, die für die angegebenen Hosts erstellt werden sollen.
templates	object/array	Vorlagen, die mit den angegebenen Hosts verknüpft werden sollen. Für die Vorlagen darf nur die Eigenschaft `templateid` definiert sein.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Hosts unter der Eigenschaft `hostids` enthält.

Beispiele

Makros hinzufügen

Fügen Sie zwei neue Makros zu zwei Hosts hinzu.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "host.massadd",
    "params": {
        "hosts": [
            {
                "hostid": "10160"
            },
            {
                "hostid": "10167"
            }
        ],
        "macros": [
            {
                "macro": "{$TEST1}",
                "value": "MACROTEST1"
            },
            {
                "macro": "{$TEST2}",
                "value": "MACROTEST2",
                "description": "Testbeschreibung"
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10160",
            "10167"
```

```
        ]
    },
    "id": 1
}
```

Siehe auch

- host.update
- Host-Gruppe
- Vorlage
- Benutzer- Makro
- Host- Schnittstelle

Quelle

CHost::massAdd() in *ui/include/classes/api/services/CHost.php*.

host.massremove

Beschreibung

`object host.massremove(object parameters)`

Diese Methode ermöglicht es Ihnen, zugehörige Objekte von mehreren Hosts zu entfernen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die IDs der zu aktualisierenden Hosts und die Objekte enthalten, die entfernt werden sollen.

Parameter	Type	Beschreibung
hostids	ID/array	IDs der Hosts, die aktualisiert werden sollen. Parameterverhalten: - *erforderlich*
groupids	ID/array	IDs der Host-Gruppen, aus denen die angegebenen Hosts entfernt werden sollen.
interfaces	object/array	Host-Schnittstellen, die von den angegebenen Hosts entfernt werden sollen. Im Host-Schnittstellenobjekt dürfen nur die Eigenschaften `ip`, `dns` und `port` definiert sein.
macros	string/array	Benutzermakros, die von den angegebenen Hosts gelöscht werden sollen.
templateids	ID/array	IDs der Vorlagen, deren Verknüpfung mit den angegebenen Hosts aufgehoben werden soll.
templateids_clear	ID/array	IDs der Vorlagen, deren Verknüpfung mit den angegebenen Hosts aufgehoben und die von ihnen entfernt werden sollen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Hosts unter der Eigenschaft `hostids` enthält.

Beispiele

Verknüpfung von Vorlagen aufheben

Heben Sie die Verknüpfung einer Vorlage von zwei Hosts auf und löschen Sie alle aus der Vorlage stammenden Entitäten.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "host.massremove",
    "params": {
        "hostids": ["69665", "69666"],
        "templateids_clear": "325"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "69665",
            "69666"
        ]
    },
    "id": 1
}
```

Siehe auch

- host.update
- Benutzer- Makro
- Host- Schnittstelle

Quelle

CHost::massRemove() in *ui/include/classes/api/services/CHost.php*.

host.update

Beschreibung

`object host.update(object/array hosts)`

Diese Methode ermöglicht es Ihnen, vorhandene Hosts zu aktualisieren.

Note:
Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu aktualisierende Host-Eigenschaften.

Die Eigenschaft `hostid` muss für jeden Host definiert sein, alle anderen Eigenschaften sind optional. Nur die angegebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Beachten Sie jedoch, dass die Aktualisierung des technischen Host-Namens auch den sichtbaren Namen des Hosts auf den Wert des technischen Host-Namens aktualisiert (falls er nicht separat angegeben wird).

Zusätzlich zu den standardmäßigen Host-Eigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Typ	Beschreibung
groups	object/array	Host-Gruppen, die die aktuellen Host-Gruppen ersetzen, denen der Host angehört. Alle Host-Gruppen, die nicht in der Anfrage aufgeführt sind, werden getrennt. Für die Host-Gruppen darf nur die Eigenschaft `groupid` definiert sein.
interfaces	object/array	Host-Schnittstellen, die die aktuellen Host-Schnittstellen ersetzen. Alle Schnittstellen, die nicht in der Anfrage aufgeführt sind, werden entfernt.

Parameter	Typ	Beschreibung
tags	object/array	Host-Tags, die die aktuellen Host-Tags ersetzen. Alle Tags, die nicht in der Anfrage aufgeführt sind, werden entfernt.
inventory	object	Eigenschaften des Host-Inventars.
macros	object/array	Benutzermakros, die die aktuellen Benutzermakros ersetzen. Alle Makros, die nicht in der Anfrage aufgeführt sind, werden entfernt.
templates	object/array	Vorlagen, die die aktuell verknüpften Vorlagen ersetzen. Alle Vorlagen, die nicht in der Anfrage aufgeführt sind, werden nur getrennt. Für die Vorlagen darf nur die Eigenschaft `templateid` definiert sein.
templates_clear	object/array	Vorlagen, die vom Host getrennt und daraus entfernt werden. Für die Vorlagen darf nur die Eigenschaft `templateid` definiert sein.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Hosts unter der Eigenschaft `hostids` enthält.

Beispiele

Aktivieren eines Hosts

Aktivieren Sie die Überwachung des Hosts, d. h. setzen Sie seinen Status auf „0“.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10126",
        "status": 0
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10126"
        ]
    },
    "id": 1
}
```

Aktivieren mehrerer Hosts

Aktivieren Sie die Überwachung von zwei Hosts, d. h. setzen Sie ihren Status auf „0“.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": [
        {
            "hostid": "10127",
            "status": 0
        },
        {
            "hostid": "10128",
            "status": 0
        }
```

```
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10127",
            "10128"
        ]
    },
    "id": 1
}
```

Verknüpfung von Vorlagen aufheben

Heben Sie die Verknüpfung von zwei Vorlagen mit dem Host auf und löschen Sie sie.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10126",
        "templates_clear": [
            {
                "templateid": "10124"
            },
            {
                "templateid": "10125"
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10126"
        ]
    },
    "id": 1
}
```

Host-Makros aktualisieren

Ersetzen Sie alle Host-Makros durch zwei neue.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10126",
        "macros": [
            {
                "macro": "{$PASS}",
                "value": "password"
            },
```

```
            {
                "macro": "{$DISC}",
                "value": "sda",
                "description": "Aktualisierte Beschreibung"
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10126"
        ]
    },
    "id": 1
}
```

Host-Inventar aktualisieren

Inventarmodus ändern und Standort hinzufügen

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10387",
        "inventory_mode": 0,
        "inventory": {
            "location": "Lettland, Riga"
        }
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10387"
        ]
    },
    "id": 1
}
```

Aktualisieren von Host-Tags

Ersetzen Sie alle Host-Tags durch ein neues.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10387",
        "tags": {
            "tag": "os",
            "value": "rhel-7"
        }
```

```
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10387"
        ]
    },
    "id": 1
}
```

Aktualisieren entdeckter Host-Makros

Wandeln Sie das von der Discovery-Regel erstellte „automatische“ Makro in ein „manuelles“ um und ändern Sie seinen Wert in „new-value“.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10387",
        "macros": {
            "hostmacroid": "5541",
            "value": "new-value",
            "automatic": "0"
        }
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10387"
        ]
    },
    "id": 1
}
```

Aktualisierung der Host-Verschlüsselung

Aktualisieren Sie den Host ”10590”, damit für Verbindungen vom Host zum Zabbix-Server nur noch PSK-Verschlüsselung verwendet wird, und ändern Sie die PSK-Identität sowie den PSK-Schlüssel. Beachten Sie, dass der Host vorab für die Verwendung von PSK konfiguriert sein muss.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10590"
        ]
    },
    "id": 1
}
```

Siehe auch

- host.massadd
- host.massremove
- Host-Gruppe
- Vorlage
- Benutzermakro
- Host-Schnittstelle
- Host-Inventar
- Host-Tag
- Proxy
- Proxy-Gruppe

Quelle

CHost::update() in *ui/include/classes/api/services/CHost.php*.

Host-Gruppe

Diese Klasse ist für die Arbeit mit Host-Gruppen vorgesehen.

Objektreferenzen:

- Host-Gruppe

Verfügbare Methoden:

- hostgroup.create - neue Host-Gruppen erstellen
- hostgroup.delete - Host-Gruppen löschen
- hostgroup.get - Host-Gruppen abrufen
- hostgroup.massadd - verknüpfte Objekte zu Host-Gruppen hinzufügen
- hostgroup.massremove - verknüpfte Objekte aus Host-Gruppen entfernen
- hostgroup.propagate - Berechtigungen und Tag-Filter an Untergruppen von Host-Gruppen weitergeben
- hostgroup.update - Host-Gruppen aktualisieren

Host-Gruppen-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `hostgroup` API.

Host-Gruppe

Das Objekt der Host-Gruppe hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
groupid	ID	ID der Host-Gruppe. Eigenschaftsverhalten: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge

Eigenschaft	Typ	Beschreibung
name	string	Name der Host-Gruppe. Eigenschaftsverhalten: - *erforderlich* für Erstellungsvorgänge
flags	integer	Herkunft der Host-Gruppe. Mögliche Werte: 0 - eine normale Host-Gruppe; 4 - eine entdeckte Host-Gruppe. Eigenschaftsverhalten: - *schreibgeschützt*
uuid	string	Universell eindeutige Kennung, die verwendet wird, um importierte Host-Gruppen mit bereits vorhandenen zu verknüpfen. Wird automatisch generiert, wenn sie nicht angegeben ist.

hostgroup.create

Beschreibung

`object hostgroup.create(object/array hostGroups)`

Diese Methode ermöglicht es Ihnen, neue Hostgruppen zu erstellen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellende Host-Gruppen.

Die Methode akzeptiert Host-Gruppen mit den Standard-Host-Gruppeneigenschaften.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Host-Gruppen unter der Eigenschaft `groupids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Wirtsgruppen.

Beispiele

Erstellen einer Host-Gruppe

Erstellen Sie eine Host-Gruppe mit dem Namen „Linux servers“.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.create",
    "params": {
        "name": "Linux servers"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107819"
        ]
    },
```

```
    "id": 1
}
```

Quelle

CHostGroup::create() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.delete

Beschreibung

`object hostgroup.delete(array hostGroupIds)`

Diese Methode ermöglicht das Löschen von Hostgruppen.

Eine Hostgruppe kann nicht gelöscht werden, wenn:

- sie Hosts enthält, die nur zu dieser Gruppe gehören;
- sie als intern markiert ist;
- sie von einem Hostprototyp verwendet wird;
- sie in einem globalen Skript verwendet wird;
- sie in einer Korrelationsbedingung verwendet wird.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Benutzerrolleneinstellungen widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der zu löschenden Hostgruppen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Host-Gruppen unter der Eigenschaft `groupids` enthält.

Beispiele

Mehrere Host-Gruppen löschen

Löschen Sie zwei Host-Gruppen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.delete",
    "params": [
        "107824",
        "107825"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107824",
            "107825"
        ]
    },
    "id": 1
}
```

Quelle

CHostGroup::delete() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.get

Beschreibung

`integer/array hostgroup.get(object parameters)`

Die Methode ermöglicht es Ihnen, Hostgruppen anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
graphids	ID/array	Gibt nur Host-Gruppen zurück, die Hosts mit den angegebenen Graphen enthalten.
groupids	ID/array	Gibt nur Host-Gruppen mit den angegebenen Host-Gruppen-IDs zurück.
hostids	ID/array	Gibt nur Host-Gruppen zurück, die die angegebenen Hosts enthalten.
maintenanceids	ID/array	Gibt nur Host-Gruppen zurück, die von den angegebenen Wartungen betroffen sind.
triggerids	ID/array	Gibt nur Host-Gruppen zurück, die Hosts mit den angegebenen Auslösern enthalten.
with_graphs	boolean	Gibt nur Host-Gruppen zurück, die Hosts mit Graphen enthalten.
with_graph_prototypes	boolean	Gibt nur Host-Gruppen zurück, die Hosts mit Graph-Prototypen enthalten.
with_hosts	boolean	Gibt nur Host-Gruppen zurück, die Hosts enthalten.
with_httptests	boolean	Gibt nur Host-Gruppen zurück, die Hosts mit Web-Prüfungen enthalten. Überschreibt den Parameter `with_monitored_httptests`.
with_items	boolean	Gibt nur Host-Gruppen zurück, die Hosts mit Datenpunkten enthalten. Überschreibt die Parameter `with_monitored_items` und `with_simple_graph_items`.
with_item_prototypes	boolean	Gibt nur Host-Gruppen zurück, die Hosts mit Datenpunkt-Prototypen enthalten. Überschreibt den Parameter `with_simple_graph_item_prototypes`.
with_simple_graph_item_prototypes	boolean	Gibt nur Host-Gruppen zurück, die Hosts mit Datenpunkt-Prototypen enthalten, die für die Erstellung aktiviert sind und einen numerischen Informationstyp haben.
with_monitored_httptests	boolean	Gibt nur Host-Gruppen zurück, die Hosts mit aktivierten Web-Prüfungen enthalten.
with_monitored_hosts	boolean	Gibt nur Host-Gruppen zurück, die überwachte Hosts enthalten.
with_monitored_items	boolean	Gibt nur Host-Gruppen zurück, die Hosts mit aktivierten Datenpunkten enthalten. Überschreibt den Parameter `with_simple_graph_items`.
with_monitored_triggers	boolean	Gibt nur Host-Gruppen zurück, die Hosts mit aktivierten Auslösern enthalten. Alle im Auslöser verwendeten Datenpunkte müssen ebenfalls aktiviert sein.
with_simple_graph_items	boolean	Gibt nur Host-Gruppen zurück, die Hosts mit numerischen Datenpunkten enthalten.
with_triggers	boolean	Gibt nur Host-Gruppen zurück, die Hosts mit Auslösern enthalten. Überschreibt den Parameter `with_monitored_triggers`.
selectDiscoveryRules	query	Gibt eine Eigenschaft `discoveryRules` mit den LLD-Regeln zurück, die die Host-Gruppe erkannt haben.

Parameter	Type	Beschreibung
selectDiscoveryData	query	Gibt eine Eigenschaft `discoveryData` mit den Host-Gruppen-Erkennungsobjekten zurück. Jedes Host-Gruppen-Erkennungsobjekt ist ein mit der erkannten Host-Gruppe verknüpfter Host-Gruppen-Prototyp und hat die folgenden Eigenschaften: `parent_group_prototypeid` - `(ID)` ID des Host-Gruppen-Prototyps, aus dem die Host-Gruppe erkannt wurde; `name` - `(string)` Name des Host-Gruppen-Prototyps; `ts_delete` - `(timestamp)` Zeitpunkt, zu dem die nicht mehr erkannte Host-Gruppe gelöscht wird; `status` - `(int)` Status der Host-Gruppen-Erkennung: 0 - *(Standard)* Host-Gruppe ist erkannt, 1 - Host-Gruppe wird nicht mehr erkannt.
selectHostPrototypes	query	Gibt eine Eigenschaft `hostPrototypes` mit Host-Prototypen zurück, die diese Host-Gruppe erkannt haben.
selectHosts	query	Gibt eine Eigenschaft `hosts` mit den Hosts zurück, die zur Host-Gruppe gehören. Unterstützt `count`.
limitSelects	integer	Begrenzt die Anzahl der von Unterabfragen zurückgegebenen Datensätze. Gilt für die folgenden Unterabfragen: `selectHosts` - Ergebnisse werden nach `host` sortiert.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `groupid`, `name`.
countOutput	boolean	Diese Parameter sind in der Referenzkommentierung beschrieben.
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	Gibt eine Eigenschaft `groupDiscoveries` mit den Host-Gruppen-Erkennungsobjekten zurück. Jedes Host-Gruppen-Erkennungsobjekt ist ein mit der erkannten Host-Gruppe verknüpfter Host-Gruppen-Prototyp. Diese Abfrage ist **veraltet**, bitte verwenden Sie stattdessen `selectDiscoveryData`.

Rückgabewerte

`(integer/array)` Gibt entweder Folgendes zurück:

- ein Array von Objekten
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde

Beispiele

Daten per Namen abrufen

Rufen Sie alle Daten zu zwei Host-Gruppen mit den Namen „Zabbix servers“ und „Linux servers“ ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.get",
    "params": {
        "output": "extend",
        "filter": {
            "name": [
                "Zabbix servers",
                "Linux servers"
            ]
        }
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "groupid": "2",
            "name": "Linux servers",
            "flags": "0",
            "uuid": "dc579cd7a1a34222933f24f52a68bcd8"
        },
        {
            "groupid": "4",
            "name": "Zabbix servers",
            "flags": "0",
            "uuid": "6f6799aa69e844b4b3918f779f2abf08"
        }
    ],
    "id": 1
}
```

Siehe auch

- Host

Quelle

CHostGroup::get() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.massadd

Beschreibung

`object hostgroup.massadd(object parameters)`

Diese Methode ermöglicht es Ihnen, mehrere zusammengehörige Objekte gleichzeitig zu allen angegebenen Hostgruppen hinzuzufügen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die IDs der zu aktualisierenden Host-Gruppen und die Objekte enthalten, die zu allen Host-Gruppen hinzugefügt werden.

Die Methode akzeptiert die folgenden Parameter.

Parameter	Type	Beschreibung
groups	object/array	Zu aktualisierende Host-Gruppen. Für die Host-Gruppen darf nur die Eigenschaft `groupid` definiert sein. Parameter behavior: - *erforderlich*
hosts	object/array	Hosts, die zu allen Host-Gruppen hinzugefügt werden sollen. Für die Hosts darf nur die Eigenschaft `hostid` definiert sein.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Host-Gruppen unter der Eigenschaft `groupids` enthält.

Beispiele

Hinzufügen von Hosts zu Host-Gruppen

Fügen Sie zwei Hosts zu Host-Gruppen mit den IDs 5 und 6 hinzu.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.massadd",
    "params": {
        "groups": [
            {
                "groupid": "5"
            },
            {
                "groupid": "6"
            }
        ],
        "hosts": [
            {
                "hostid": "30050"
            },
            {
                "hostid": "30001"
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "5",
            "6"
        ]
    },
    "id": 1
}
```

Siehe auch

- Host

Quelle

CHostGroup::massAdd() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.massremove

Beschreibung

`object hostgroup.massremove(object parameters)`

Diese Methode ermöglicht es Ihnen, zugehörige Objekte aus mehreren Hostgruppen zu entfernen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die IDs der zu aktualisierenden Hostgruppen und die Objekte enthalten, die entfernt werden sollen.

Parameter	Type	Beschreibung
groupids	ID/array	IDs der zu aktualisierenden Hostgruppen. Parameter behavior: - *erforderlich*
hostids	ID/array	IDs der Hosts, die aus allen Hostgruppen entfernt werden sollen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Host-Gruppen unter der Eigenschaft `groupids` enthält.

Beispiele

Entfernen von Hosts aus Host-Gruppen

Entfernen Sie zwei Hosts aus den angegebenen Host-Gruppen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.massremove",
    "params": {
        "groupids": [
            "5",
            "6"
        ],
        "hostids": [
            "30050",
            "30001"
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "5",
            "6"
        ]
    },
    "id": 1
}
```

Quelle

CHostGroup::massRemove() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.propagate

Beschreibung

`object hostgroup.propagate(object parameters)`

Diese Methode ermöglicht es Ihnen, Berechtigungen und Tag-Filter auf alle Untergruppen einer Hostgruppe anzuwenden.

> **Note:**
> Diese Methode ist nur für Benutzertypen vom Typ *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
groups	object/array	Zu propagierende Host-Gruppen. Für die Host-Gruppen muss die Eigenschaft `groupid` definiert sein. Parameterverhalten: - *erforderlich*
permissions	boolean	Auf „true“ setzen, um Berechtigungen zu propagieren. Parameterverhalten: - *erforderlich*, wenn `tag_filters` nicht gesetzt ist
tag_filters	boolean	Auf „true“ setzen, um Tag-Filter zu propagieren. Parameterverhalten: - *erforderlich*, wenn `permissions` nicht gesetzt ist

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der propagierten Host-Gruppen unter der Eigenschaft `groupids` enthält.

Beispiele

Berechtigungen und Tag-Filter einer Hostgruppe an ihre Untergruppen weitergeben.

Geben Sie Berechtigungen und Tag-Filter einer Hostgruppe an ihre Untergruppen weiter.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.propagate",
    "params": {
        "groups": [
            {
                "groupid": "6"
            }
        ],
        "permissions": true,
        "tag_filters": true
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
```

```
            "6",
        ]
    },
    "id": 1
}
```

Siehe auch

- hostgroup.update
- hostgroup.massadd
- Host

Quelle

CHostGroup::propagate() in *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.update

Beschreibung

`object hostgroup.update(object/array hostGroups)`

Diese Methode ermöglicht es Ihnen, vorhandene Hostgruppen zu aktualisieren.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` zu aktualisierende Host-Gruppen-Eigenschaften.

Die Eigenschaft `groupid` muss für jede Host-Gruppe definiert sein, alle anderen Eigenschaften sind optional. Nur die angegebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Host-Gruppen unter der Eigenschaft `groupids` enthält.

Beispiele

Umbenennen einer Host-Gruppe

Benennen Sie eine Host-Gruppe in „Linux hosts“ um.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.update",
    "params": {
        "groupid": "7",
        "name": "Linux hosts"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "7"
        ]
    },
    "id": 1
}
```

Umbenennen mehrerer Host-Gruppen

Benennen Sie zwei Host-Gruppen um, um Regionsinformationen in ihre Namen aufzunehmen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.update",
    "params": [
        {
            "groupid": "8",
            "name": "Europe/Linux hosts"
        },
        {
            "groupid": "9",
            "name": "Europe/Windows hosts"
        }
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "8",
            "9"
        ]
    },
    "id": 1
}
```

Quelle

CHostGroup::update() in *ui/include/classes/api/services/CHostGroup.php*.

Host-Prototyp

Diese Klasse ist für die Arbeit mit Host-Prototypen ausgelegt.

Objektreferenzen:

- Host-Prototyp
- Gruppenverknüpfung
- Gruppenprototyp
- Host-Prototyp-Tag
- Benutzerdefinierte Schnittstelle
    - Details der benutzerdefinierten Schnittstelle

Verfügbare Methoden:

- hostprototype.create - neue Host-Prototypen erstellen
- hostprototype.delete - Host-Prototypen löschen
- hostprototype.get - Host-Prototypen abrufen
- hostprototype.update - Host-Prototypen aktualisieren

Host-Prototyp-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `hostprototype` API.

Host-Prototyp

Das Host-Prototyp-Objekt hat die folgenden Eigenschaften.

Eigenschaft	Type	Beschreibung
hostid	ID	ID des Host-Prototyps. Property behavior: - *read-only* - *required* für Aktualisierungsvorgänge
host	string	Technischer Name des Host-Prototyps. Property behavior: - *required* für Erstellungsvorgänge - *read-only* für vererbte Objekte
name	string	Sichtbarer Name des Host-Prototyps. Standard: Wert der Eigenschaft `host`. Property behavior: - *read-only* für vererbte Objekte
status	integer	Status des Host-Prototyps. Mögliche Werte: 0 - *(Standard)* überwachter Host; 1 - nicht überwachter Host.
flags	integer	Herkunft des Host-Prototyps. Mögliche Werte: 2 - ein Host-Prototyp; 6 - ein entdeckter Host-Prototyp Property behavior: - *read-only*
inventory_mode	integer	Modus zum Befüllen des Host-Inventars. Mögliche Werte: -1 - *(Standard)* deaktiviert; 0 - manuell; 1 - automatisch.
templateid	ID	ID des übergeordneten Vorlagen-Host-Prototyps. Property behavior: - *read-only*
discover	integer	Erkennungsstatus des Host-Prototyps. Mögliche Werte: 0 - *(Standard)* neue Hosts werden erkannt; 1 - neue Hosts werden nicht erkannt und vorhandene Hosts werden als verloren markiert.
custom_interfaces	integer	Quelle der benutzerdefinierten Schnittstellen für Hosts, die durch den Host-Prototyp erstellt werden. Mögliche Werte: 0 - *(Standard)* Schnittstellen vom übergeordneten Host erben; 1 - benutzerdefinierte Schnittstellen des Host-Prototyps verwenden. Property behavior: - *read-only* für vererbte Objekte
uuid	string	Universell eindeutige Kennung, die verwendet wird, um importierte Host-Prototypen mit bereits vorhandenen zu verknüpfen. Wird automatisch generiert, wenn sie nicht angegeben ist. Property behavior: - *supported* wenn der Host-Prototyp zu einer Vorlage gehört

Gruppenverknüpfung

Das Objekt „Gruppenverknüpfung“ verknüpft einen Host-Prototyp mit einer Host-Gruppe. Es hat die folgenden Eigenschaften.

Eigenschaft	Type	Beschreibung
groupid	ID	ID der Host-Gruppe. Verhalten der Eigenschaft: - *erforderlich*

Gruppenprototyp

Das Gruppenprototyp-Objekt definiert eine Gruppe, die für einen erkannten Host erstellt wird. Es hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
group_prototypeid	ID	ID des Gruppenprototyps. Verhalten der Eigenschaft: - *schreibgeschützt*
name	string	Name des Gruppenprototyps. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsoperationen

Tag des Host-Prototyps

Das Tag-Objekt des Host-Prototyps hat die folgenden Eigenschaften.

Eigenschaft	Type	Beschreibung
tag	string	Name des Tags des Host-Prototyps. Property behavior: - *required*
value	string	Wert des Tags des Host-Prototyps. Property behavior: - *read-only*.
object	integer	Typ des Objekts, von dem das Tag geerbt wurde. Mögliche Werte: 0 - Vorlage. Property behavior: - *read-only*.
objectid	ID	ID des Objekts, von dem das Tag geerbt wurde. Property behavior: - *read-only*.

Benutzerdefinierte Schnittstelle

Benutzerdefinierte Schnittstellen werden unterstützt, wenn `custom_interfaces` des Host prototype object auf „benutzerdefinierte Schnittstellen von Host-Prototypen verwenden“ gesetzt ist. Das Objekt für benutzerdefinierte Schnittstellen hat die folgenden Eigenschaften.

Eigenschaft	Type	Beschreibung
type	integer	Schnittstellentyp. Mögliche Werte: 1 - Agent; 2 - SNMP; 3 - IPMI; 4 - JMX. Property behavior: - *erforderlich*
useip	integer	Gibt an, ob die Verbindung über IP hergestellt werden soll. Mögliche Werte: 0 - Verbindung über den DNS-Namen des Hosts herstellen; 1 - Verbindung über die IP-Adresse des Hosts herstellen. Property behavior: - *erforderlich*
ip	string	Von der Schnittstelle verwendete IP-Adresse. Kann Makros enthalten. Property behavior: - *erforderlich*, wenn `useip` auf „Verbindung über die IP-Adresse des Hosts herstellen“ gesetzt ist
dns	string	Von der Schnittstelle verwendeter DNS-Name. Kann Makros enthalten. Property behavior: - *erforderlich*, wenn `useip` auf „Verbindung über den DNS-Namen des Hosts herstellen“ gesetzt ist
port	string	Von der Schnittstelle verwendete Portnummer. Kann Benutzer- und LLD-Makros enthalten. Property behavior: - *erforderlich*
main	integer	Gibt an, ob die Schnittstelle auf dem Host als Standard verwendet wird. Nur eine Schnittstelle eines bestimmten Typs kann auf einem Host als Standard festgelegt werden. Mögliche Werte: 0 - nicht Standard; 1 - Standard. Property behavior: - *erforderlich*
details	object	Zusätzliches Objekt für Details benutzerdefinierter Schnittstellen. Property behavior: - *erforderlich*, wenn `type` auf „SNMP“ gesetzt ist

Details der benutzerdefinierten Schnittstelle

Das Detailobjekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
version	integer	Version der SNMP-Schnittstelle. Mögliche Werte: 1 - SNMPv1; 2 - SNMPv2c; 3 - SNMPv3. Eigenschaftsverhalten: - *erforderlich*
bulk	integer	Gibt an, ob Bulk-SNMP-Anfragen verwendet werden sollen. Mögliche Werte: 0 - keine Bulk-Anfragen verwenden; 1 - *(Standard)* - Bulk-Anfragen verwenden.
community	string	SNMP-Community. Eigenschaftsverhalten: - *erforderlich*, wenn `version` auf ”SNMPv1” oder ”SNMPv2c” gesetzt ist
max_repetitions	integer	Maximaler Wiederholungswert für native SNMP-Bulk-Anfragen (GetBulkRequest-PDUs). Wird nur für `discovery[]`- und `walk[]`-Datenpunkte in SNMPv2 und v3 verwendet. Standard: 10.
securityname	string	SNMPv3-Sicherheitsname. Eigenschaftsverhalten: - *unterstützt*, wenn `version` auf ”SNMPv3” gesetzt ist
securitylevel	integer	SNMPv3-Sicherheitsstufe. Mögliche Werte: 0 - *(Standard)* - noAuthNoPriv; 1 - authNoPriv; 2 - authPriv. Eigenschaftsverhalten: - *unterstützt*, wenn `version` auf ”SNMPv3” gesetzt ist
authpassphrase	string	SNMPv3-Authentifizierungs-Passphrase. Eigenschaftsverhalten: - *unterstützt*, wenn `version` auf ”SNMPv3” und `securitylevel` auf ”authNoPriv” oder ”authPriv” gesetzt ist
privpassphrase	string	SNMPv3-Datenschutz-Passphrase. Eigenschaftsverhalten: - *unterstützt*, wenn `version` auf ”SNMPv3” und `securitylevel` auf ”authPriv” gesetzt ist
authprotocol	integer	SNMPv3-Authentifizierungsprotokoll. Mögliche Werte: 0 - *(Standard)* - MD5; 1 - SHA1; 2 - SHA224; 3 - SHA256; 4 - SHA384; 5 - SHA512. Eigenschaftsverhalten: - *unterstützt*, wenn `version` auf ”SNMPv3” und `securitylevel` auf ”authNoPriv” oder ”authPriv” gesetzt ist

Eigenschaft	Typ	Beschreibung
privprotocol	integer	SNMPv3-Datenschutzprotokoll. Wird nur von SNMPv3-Schnittstellen verwendet. Mögliche Werte: 0 - *(Standard)* - DES; 1 - AES128; 2 - AES192; 3 - AES256; 4 - AES192C; 5 - AES256C. Eigenschaftsverhalten: - *unterstützt*, wenn `version` auf "SNMPv3" und `securitylevel` auf "authPriv" gesetzt ist
contextname	string	SNMPv3-Kontextname. Eigenschaftsverhalten: - *unterstützt*, wenn `version` auf "SNMPv3" gesetzt ist

hostprototype.create

Beschreibung

`object hostprototype.create(object/array hostPrototypes)`

Diese Methode ermöglicht es Ihnen, neue Host-Prototypen zu erstellen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellende Host-Prototypen.

Zusätzlich zu den Standard-Host-Prototyp-Eigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Typ	Beschreibung
groupLinks	array	Zu erstellende Gruppenverknüpfungen für den Host-Prototyp. Parameterverhalten: - *erforderlich*
ruleid	ID	ID der LLD-Regel, zu der der Host-Prototyp gehört. Parameterverhalten: - *erforderlich*
groupPrototypes	array	Zu erstellende Gruppenprototypen für den Host-Prototyp.
macros	object/array	Zu erstellende Benutzermakros für den Host-Prototyp.
tags	object/array	Host-Prototyp-Tags.
interfaces	object/array	Benutzerdefinierte Schnittstellen des Host-Prototyps.
templates	object/array	Vorlagen, die mit dem Host-Prototyp verknüpft werden sollen. Für die Vorlagen darf nur die Eigenschaft `templateid` definiert sein.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Host-Prototypen unter der Eigenschaft `hostids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Host-Prototypen.

Beispiele

Erstellen eines Host-Prototyps

Erstellen Sie einen Host-Prototypen "{#VM.NAME}" für die LLD-Regel "23542" mit einem Gruppen- Prototypen "{#HV.NAME}", dem Tag-Paar "datacenter": "{#DATACENTER.NAME}" und einer benutzerdefinierten SNMPv2-Schnittstelle 127.0.0.1:161 mit Community {$SNMP_COMMUNITY}. Verknüpfen Sie ihn mit der Host-Gruppe "2".

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10103"
        ]
    },
    "id": 1
}
```

Siehe auch

- Gruppenverknüpfung
- Gruppenprototyp
- Host-Prototyp-Tag
- Benutzerdefinierte Schnittstelle

- Benutzermakro

Quelle

CHostPrototype::create() in *ui/include/classes/api/services/CHostPrototype.php*.

hostprototype.delete

Beschreibung

`object hostprototype.delete(array hostPrototypeIds)`

Diese Methode ermöglicht es Ihnen, Host-Prototypen zu löschen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der zu löschenden Host-Prototypen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Host-Prototypen unter der Eigenschaft `hostids` enthält.

Beispiele

Mehrere Host-Prototypen löschen

Löschen Sie zwei Host-Prototypen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.delete",
    "params": [
        "10103",
        "10105"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10103",
            "10105"
        ]
    },
    "id": 1
}
```

Quelle

CHostPrototype::delete() in *ui/include/classes/api/services/CHostPrototype.php*.

hostprototype.get

Beschreibung

`integer/array hostprototype.get(object parameters)`

Die Methode ermöglicht es Ihnen, Host-Prototypen anhand der angegebenen Parameter abzurufen.

Note:
Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
hostids	ID/array	Gibt nur Host-Prototypen mit den angegebenen IDs zurück.
discoveryids	ID/array	Gibt nur Host-Prototypen zurück, die zu den angegebenen LLD-Regeln gehören.
inherited	boolean	Wenn auf `true` gesetzt, werden nur von einer Vorlage geerbte Datenpunkte zurückgegeben.
selectDiscoveryData	query	Gibt eine `discoveryData`-Eigenschaft mit den Objektdaten der Host-Prototyp-Erkennung zurück. Das Host-Prototyp-Erkennungsobjekt verknüpft einen erkannten Host-Prototyp mit einem Host-Prototyp, von dem er erkannt wurde. Es hat die folgenden Eigenschaften: `host` - `(string)` ID des Hosts; `parent_hostid` - `(string)` ID des Host-Prototyps, aus dem der Host-Prototyp erstellt wurde; `status` - `(int)` Erkennungsstatus des Host-Prototyps: 0 - *(Standard)* Host-Prototyp wurde erkannt, 1 - Host-Prototyp wird nicht mehr erkannt; `ts_delete` - `(timestamp)` Zeitpunkt, zu dem ein Host-Prototyp, der nicht mehr erkannt wird, gelöscht wird; `ts_disable` - `(timestamp)` Zeitpunkt, zu dem ein Host-Prototyp, der nicht mehr erkannt wird, deaktiviert wird; `disable_source` - `(int)` Kennzeichen, ob der Host-Prototyp durch eine LLD-Regel oder manuell deaktiviert wurde: 0 - *(Standard)* automatisch deaktiviert, 1 - durch eine LLD-Regel deaktiviert.
selectDiscoveryRule	query	Gibt eine `discoveryRule`-Eigenschaft mit der LLD-Regel zurück, zu der der Host-Prototyp gehört.
selectDiscoveryRulePrototype	query	Gibt eine `discoveryRulePrototype`-Eigenschaft mit der übergeordneten LLD-Regelvorlage zurück, zu der der Host-Prototyp gehört.
selectInterfaces	query	Gibt eine `interfaces`-Eigenschaft mit benutzerdefinierten Schnittstellen des Host-Prototyps zurück.
selectGroupLinks	query	Gibt eine `groupLinks`-Eigenschaft mit den Gruppenverknüpfungen des Host-Prototyps zurück.
selectGroupPrototypes	query	Gibt eine `groupPrototypes`-Eigenschaft mit den Gruppenvorlagen des Host-Prototyps zurück.
selectInheritedTags	query	Gibt eine `inheritedTags`-Eigenschaft mit Tags zurück, die sich auf verknüpften Vorlagen befinden.
selectMacros	query	Gibt eine `macros`-Eigenschaft mit Makros des Host-Prototyps zurück.
selectParentHost	query	Gibt eine `parentHost`-Eigenschaft mit dem Host zurück, zu dem der Host-Prototyp gehört.
selectTags	query	Gibt eine `tags`-Eigenschaft mit Tags des Host-Prototyps zurück.
selectTemplates	query	Gibt eine `templates`-Eigenschaft mit den mit dem Host-Prototyp verknüpften Vorlagen zurück. Unterstützt `count`.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `hostid`, `host`, `name`, `status`, `discovered`.
countOutput	boolean	Diese Parameter werden im Referenzkommentar beschrieben.
editable	boolean	
excludeSearch	boolean	

Parameter	Type	Beschreibung
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Gibt entweder Folgendes zurück:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Abrufen von Host-Prototypen aus einer LLD-Regel

Rufen Sie alle Host-Prototypen, ihre Gruppenverknüpfungen, Gruppenprototypen und Tags aus einer LLD-Regel ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.get",
    "params": {
        "output": "extend",
        "selectInterfaces": "extend",
        "selectGroupLinks": "extend",
        "selectGroupPrototypes": "extend",
        "selectTags": "extend",
        "discoveryids": "23554"
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```


Siehe auch

- Gruppenverknüpfung
- Gruppenprototyp
- Benutzermakro

Quelle

CHostPrototype::get() in *ui/include/classes/api/services/CHostPrototype.php*.

hostprototype.update

Beschreibung

`object hostprototype.update(object/array hostPrototypes)`

Diese Methode ermöglicht es Ihnen, vorhandene Host-Prototypen zu aktualisieren.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

(object/array) Zu aktualisierende Eigenschaften des Host-Prototyps.

Die Eigenschaft hostid muss für jeden Host-Prototyp definiert sein, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Zusätzlich zu den Standard-Host-Prototyp-Eigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Typ	Beschreibung
groupLinks	array	Gruppenverknüpfungen, um die aktuellen Gruppenverknüpfungen des Host-Prototyps zu ersetzen. Parameterverhalten: - *schreibgeschützt* für vererbte Objekte
groupPrototypes	array	Gruppenprototypen, um die vorhandenen Gruppenprototypen des Host-Prototyps zu ersetzen. Alle Gruppenprototypen, die nicht in der Anfrage aufgeführt sind, werden entfernt. Parameterverhalten: - *schreibgeschützt* für vererbte Objekte
macros	object/array	Benutzermakros, um die aktuellen Benutzermakros zu ersetzen. Alle Makros, die nicht in der Anfrage aufgeführt sind, werden entfernt.
tags	object/array	Host-Prototyp-Tags, um die aktuellen Tags zu ersetzen. Alle Tags, die nicht in der Anfrage aufgeführt sind, werden entfernt.
interfaces	object/array	Parameterverhalten: - *schreibgeschützt* für vererbte Objekte Host-Prototyp-benutzerdefinierte Schnittstellen, um die aktuellen Schnittstellen zu ersetzen. Das Objekt der benutzerdefinierten Schnittstelle muss alle seine Parameter enthalten. Alle Schnittstellen, die nicht in der Anfrage aufgeführt sind, werden entfernt. Parameterverhalten: - *unterstützt*, wenn custom_interfaces des Host-Prototyp-Objekts auf "use host prototypes custom interfaces" gesetzt ist - *schreibgeschützt* für vererbte Objekte
templates	object/array	Vorlagen, um die aktuell verknüpften Vorlagen zu ersetzen. Für die Vorlagen darf nur die Eigenschaft templateid definiert sein.

Rückgabewerte

(object) Gibt ein Objekt zurück, das die IDs der aktualisierten Host-Prototypen unter der Eigenschaft hostids enthält.

Beispiele

Deaktivieren eines Host-Prototyps

Deaktivieren Sie einen Host-Prototyp, d. h. setzen Sie seinen Status auf „1“.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.update",
    "params": {
        "hostid": "10092",
        "status": 1
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10092"
        ]
    },
    "id": 1
}
```

Tags von Host-Prototypen aktualisieren

Ersetzen Sie die Tags des Host-Prototyps durch neue.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.update",
    "params": {
        "hostid": "10092",
        "tags": [
            {
                "tag": "datacenter",
                "value": "{#DATACENTER.NAME}"
            },
            {
                "tag": "instance-type",
                "value": "{#INSTANCE_TYPE}"
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10092"
        ]
    },
    "id": 1
}
```

Benutzerdefinierte Schnittstellen des Host-Prototyps aktualisieren

Ersetzen Sie geerbte Schnittstellen durch benutzerdefinierte Schnittstellen des Host-Prototyps.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10092"
        ]
    },
    "id": 1
}
```

Siehe auch

- Gruppenverknüpfung
- Gruppenprototyp
- Host-Prototyp-Tag
- Benutzerdefinierte Schnittstelle
- Benutzer- Makro

Quelle

CHostPrototype::update() in *ui/include/classes/api/services/CHostPrototype.php*.

Host-Schnittstelle

Diese Klasse ermöglicht den Zugriff auf Host-Schnittstellen.

Objektreferenzen:

- Host-Schnittstelle
  - Details

Verfügbare Methode:

- hostinterface.get – Host-Schnittstellen abrufen

Host-Schnittstellen werden mithilfe von Host-Methoden erstellt, aktualisiert und gelöscht.

Host-Interface-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `hostinterface` API.

Host-Schnittstelle

Das Host-Schnittstellenobjekt verfügt über die folgenden Eigenschaften.

Property	Type	Description
interfaceid	ID	ID der Schnittstelle. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge

Property	Type	Description
available	integer	Verfügbarkeit der Host-Schnittstelle. Mögliche Werte: 0 - *(Standard)* unbekannt; 1 - verfügbar; 2 - nicht verfügbar. Verhalten der Eigenschaft: - *schreibgeschützt*
hostid	ID	ID des Hosts, zu dem die Schnittstelle gehört. Verhalten der Eigenschaft: - *konstant* - *erforderlich* für Erstellungsvorgänge
type	integer	Schnittstellentyp. Mögliche Werte: 1 - Agent; 2 - SNMP; 3 - IPMI; 4 - JMX. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
ip	string	Von der Schnittstelle verwendete IP-Adresse. Kann leer sein, wenn die Verbindung über DNS hergestellt wird. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
dns	string	Von der Schnittstelle verwendeter DNS-Name. Kann leer sein, wenn die Verbindung über IP hergestellt wird. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
port	string	Von der Schnittstelle verwendete Portnummer. Kann Benutzermakros enthalten. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
useip	integer	Gibt an, ob die Verbindung über IP hergestellt werden soll. Mögliche Werte: 0 - Verbindung über den DNS-Namen des Hosts herstellen; 1 - Verbindung über die IP-Adresse des Hosts herstellen. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
main	integer	Gibt an, ob die Schnittstelle als Standardschnittstelle auf dem Host verwendet wird. Pro Typ kann auf einem Host nur eine Schnittstelle als Standardschnittstelle festgelegt werden. Mögliche Werte: 0 - nicht Standard; 1 - Standard. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge

Property	Type	Description
details	object	Zusätzliches Details-Objekt für die Schnittstelle. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf "SNMP" gesetzt ist
disable_until	timestamp	Der nächste Abfragezeitpunkt einer nicht verfügbaren Host-Schnittstelle. Verhalten der Eigenschaft: - *schreibgeschützt*
error	string	Fehlertext, wenn die Host-Schnittstelle nicht verfügbar ist. Verhalten der Eigenschaft: - *schreibgeschützt*
errors_from	timestamp	Zeitpunkt, zu dem die Host-Schnittstelle nicht verfügbar wurde. Verhalten der Eigenschaft: - *schreibgeschützt*

Details

Das Objekt `details` hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
version	integer	Version der SNMP-Schnittstelle. Mögliche Werte: 1 - SNMPv1; 2 - SNMPv2c; 3 - SNMPv3. Verhalten der Eigenschaft: - *erforderlich*
bulk	integer	Gibt an, ob Bulk-SNMP-Anfragen verwendet werden sollen. Mögliche Werte: 0 - keine Bulk-Anfragen verwenden; 1 - (Standard) - Bulk-Anfragen verwenden.
community	string	SNMP-Community. Wird nur von SNMPv1- und SNMPv2-Schnittstellen verwendet. Verhalten der Eigenschaft: - *erforderlich*, wenn `version` auf "SNMPv1" oder "SNMPv2c" gesetzt ist
max_repetitions	integer	Maximaler Wiederholungswert für native SNMP-Bulk-Anfragen (GetBulkRequest-PDUs). Wird nur für `discovery[]`- und `walk[]`-Datenpunkte in SNMPv2 und v3 verwendet. Standard: 10.
securityname	string	SNMPv3-Sicherheitsname. Wird nur von SNMPv3-Schnittstellen verwendet.
securitylevel	integer	SNMPv3-Sicherheitsstufe. Wird nur von SNMPv3-Schnittstellen verwendet. Mögliche Werte: 0 - (Standard) - noAuthNoPriv; 1 - authNoPriv; 2 - authPriv.
authpassphrase	string	SNMPv3-Authentifizierungs-Passphrase. Wird nur von SNMPv3-Schnittstellen verwendet.

Eigenschaft	Typ	Beschreibung
privpassphrase	string	SNMPv3-Datenschutz-Passphrase. Wird nur von SNMPv3-Schnittstellen verwendet.
authprotocol	integer	SNMPv3-Authentifizierungsprotokoll. Wird nur von SNMPv3-Schnittstellen verwendet. Mögliche Werte: 0 - (Standard) - MD5; 1 - SHA1; 2 - SHA224; 3 - SHA256; 4 - SHA384; 5 - SHA512.
privprotocol	integer	SNMPv3-Datenschutzsprotokoll. Wird nur von SNMPv3-Schnittstellen verwendet. Mögliche Werte: 0 - (Standard) - DES; 1 - AES128; 2 - AES192; 3 - AES256; 4 - AES192C; 5 - AES256C.
contextname	string	SNMPv3-Kontextname. Wird nur von SNMPv3-Schnittstellen verwendet.

hostinterface.get

Beschreibung

`integer/array hostinterface.get(object parameters)`

Mit dieser Methode können Sie Host-Schnittstellen entsprechend den angegebenen Parametern abrufen.

> **Note:**
> Diese Methode steht Benutzern jeden Typs zur Verfügung. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
hostids	ID/array	Gibt nur Host-Schnittstellen zurück, die von den angegebenen Hosts verwendet werden.
interfaceids	ID/array	Gibt nur Host-Schnittstellen mit den angegebenen IDs zurück.
itemids	ID/array	Gibt nur Host-Schnittstellen zurück, die von den angegebenen Datenpunkten verwendet werden.
triggerids	ID/array	Gibt nur Host-Schnittstellen zurück, die von Datenpunkten in den angegebenen Auslösern verwendet werden.
selectItems	query	Gibt eine Eigenschaft `items` mit den Datenpunkten zurück, die die Schnittstelle verwenden. Unterstützt `count`.
selectHosts	query	Gibt eine Eigenschaft `hosts` mit einem Array von Hosts zurück, die die Schnittstelle verwenden.
limitSelects	integer	Begrenzt die Anzahl der Datensätze, die von Unterabfragen zurückgegeben werden. Gilt für die folgenden Unterabfragen: `selectItems`.

Parameter	Type	Beschreibung
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `interfaceid`, `dns`, `ip`.
countOutput	boolean	Diese Parameter werden im Referenzkommentar beschrieben.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Kann die folgenden Dinge zurück geben:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Host-Schnittstellen abrufen

Rufen Sie alle Daten zu den Schnittstellen ab, die vom Host „30057“ verwendet werden.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.get",
    "params": {
        "output": "extend",
        "hostids": "30057"
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```


Siehe auch

- Host
- Item

Quelle

CHostInterface::get() in *ui/include/classes/api/services/CHostInterface.php*.

Karte

Diese Klasse ist für die Arbeit mit Karten vorgesehen.

Objektreferenzen:

- Karte
- Kartenelement
  - Kartenelement Host
  - Kartenelement Host-Gruppe
  - Kartenelement Karte
  - Kartenelement Auslöser
  - Kartenelement Tag
  - Kartenelement URL
- Kartenverbindung
  - Auslöser der Kartenverbindung
  - Indikatoren der Kartenverbindung
- Karten-URL
- Kartenbenutzer
- Karten-Benutzergruppe
- Kartenformen
- Kartenlinien

Verfügbare Methoden:

- map.create - neue Karten erstellen
- map.delete - Karten löschen
- map.get - Karten abrufen
- map.update - Karten aktualisieren

Karten-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `map` API.

Karte

Das Kartenobjekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
sysmapid	ID	ID der Karte. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
height	integer	Höhe der Karte in Pixeln. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
name	string	Name der Karte. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
width	integer	Breite der Karte in Pixeln. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
backgroundid	ID	ID des Bildes, das als Hintergrund für die Karte verwendet wird.
background_scale	integer	Ob die proportionale Skalierung des Hintergrundbildes aktiviert werden soll. Mögliche Werte: 0 - Skalierung deaktivieren; 1 - *(Standard)* Bildskalierung aktivieren.
expand_macros	integer	Ob Makros in Beschriftungen beim Konfigurieren der Karte erweitert werden sollen. Mögliche Werte: 0 - *(Standard)* Makros nicht erweitern; 1 - Makros erweitern.
expandproblem	integer	Ob der Problem-Auslöser für Elemente mit genau einem Problem angezeigt wird. Mögliche Werte: 0 - immer die Anzahl der Probleme anzeigen; 1 - *(Standard)* den Problem-Auslöser anzeigen, wenn es nur ein Problem gibt.
grid_align	integer	Ob die Ausrichtung am Raster aktiviert werden soll. Mögliche Werte: 0 - Ausrichtung am Raster deaktivieren; 1 - *(Standard)* Ausrichtung am Raster aktivieren.
grid_show	integer	Ob das Raster auf der Karte angezeigt werden soll. Mögliche Werte: 0 - Raster nicht anzeigen; 1 - *(Standard)* Raster anzeigen.
grid_size	integer	Größe des Kartenrasters in Pixeln. Unterstützte Werte: 20, 40, 50, 75 und 100. Standard: 50.
highlight	integer	Ob die Hervorhebung von Symbolen aktiviert ist. Mögliche Werte: 0 - Hervorhebung deaktiviert; 1 - *(Standard)* Hervorhebung aktiviert.
iconmapid	ID	ID der Symbolzuordnung, die auf der Karte verwendet wird.

Eigenschaft	Typ	Beschreibung
label_format	integer	Ob erweiterte Beschriftungen aktiviert werden sollen. Mögliche Werte: 0 - *(Standard)* erweiterte Beschriftungen deaktivieren; 1 - erweiterte Beschriftungen aktivieren.
label_location	integer	Position der Beschriftung des Kartenelements. Mögliche Werte: 0 - *(Standard)* unten; 1 - links; 2 - rechts; 3 - oben.
label_string_host	string	Benutzerdefinierte Beschriftung für Host-Elemente. Verhalten der Eigenschaft: - *erforderlich*, wenn `label_type_host` auf „custom“ gesetzt ist
label_string_hostgroup	string	Benutzerdefinierte Beschriftung für Hostgruppen-Elemente. Verhalten der Eigenschaft: - *erforderlich*, wenn `label_type_hostgroup` auf „custom“ gesetzt ist
label_string_image	string	Benutzerdefinierte Beschriftung für Bildelemente. Verhalten der Eigenschaft: - *erforderlich*, wenn `label_type_image` auf „custom“ gesetzt ist
label_string_map	string	Benutzerdefinierte Beschriftung für Kartenelemente. Verhalten der Eigenschaft: - *erforderlich*, wenn `label_type_map` auf „custom“ gesetzt ist
label_string_trigger	string	Benutzerdefinierte Beschriftung für Auslöser-Elemente. Verhalten der Eigenschaft: - *erforderlich*, wenn `label_type_trigger` auf „custom“ gesetzt ist
label_type	integer	Beschriftungstyp des Kartenelements. Mögliche Werte: 0 - Beschriftung; 1 - IP-Adresse; 2 - *(Standard)* Elementname; 3 - nur Status; 4 - nichts.
label_type_host	integer	Beschriftungstyp für Host-Elemente. Mögliche Werte: 0 - Beschriftung; 1 - IP-Adresse; 2 - *(Standard)* Elementname; 3 - nur Status; 4 - nichts; 5 - benutzerdefiniert.
label_type_hostgroup	integer	Beschriftungstyp für Hostgruppen-Elemente. Mögliche Werte: 0 - Beschriftung; 2 - *(Standard)* Elementname; 3 - nur Status; 4 - nichts; 5 - benutzerdefiniert.

Eigenschaft	Typ	Beschreibung
label_type_image	integer	Beschriftungstyp für Hostgruppen-Elemente. Mögliche Werte: 0 - Beschriftung; 2 - *(Standard)* Elementname; 4 - nichts; 5 - benutzerdefiniert.
label_type_map	integer	Beschriftungstyp für Kartenelemente. Mögliche Werte: 0 - Beschriftung; 2 - *(Standard)* Elementname; 3 - nur Status; 4 - nichts; 5 - benutzerdefiniert.
label_type_trigger	integer	Beschriftungstyp für Auslöser-Elemente. Mögliche Werte: 0 - Beschriftung; 2 - *(Standard)* Elementname; 3 - nur Status; 4 - nichts; 5 - benutzerdefiniert.
markelements	integer	Ob Kartenelemente hervorgehoben werden sollen, deren Status sich kürzlich geändert hat. Mögliche Werte: 0 - *(Standard)* Elemente nicht hervorheben; 1 - Elemente hervorheben.
severity_min	integer	Mindestschweregrad der Auslöser, die auf der Karte angezeigt werden. Eine Liste der unterstützten Auslöser-Schweregrade finden Sie in der Auslöser-Eigenschaft `severity`.
show_element_label	integer	Wie Elementbeschriftungen standardmäßig angezeigt werden sollen. Mögliche Werte: 0 - immer anzeigen; 1 - *(Standard)* automatisch ausblenden.
show_link_label	integer	Wie Verknüpfungsbeschriftungen standardmäßig angezeigt werden sollen. Mögliche Werte: 0 - immer anzeigen; 1 - *(Standard)* automatisch ausblenden.
show_unack	integer	Wie Probleme angezeigt werden sollen. Mögliche Werte: 0 - *(Standard)* die Anzahl aller Probleme anzeigen; 1 - nur die Anzahl unbestätigter Probleme anzeigen; 2 - die Anzahl bestätigter und unbestätigter Probleme getrennt anzeigen.
userid	ID	ID des Benutzers, der Eigentümer der Karte ist.
private	integer	Typ der Kartenfreigabe. Mögliche Werte: 0 - öffentliche Karte; 1 - *(Standard)* private Karte.

Eigenschaft	Typ	Beschreibung
show_suppressed	integer	Ob unterdrückte Probleme angezeigt werden. Mögliche Werte: 0 - *(Standard)* unterdrückte Probleme ausblenden; 1 - unterdrückte Probleme anzeigen.

Kartenelement

Das Kartenelement-Objekt definiert ein Objekt, das auf einer Karte angezeigt wird. Es hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
selementid	ID	ID des Kartenelements. Verhalten der Eigenschaft: - *schreibgeschützt*
elements	array	Datenobjekt Element. Verhalten der Eigenschaft: - *erforderlich*, wenn `elementtype` auf ”host”, ”map”, ”trigger” oder ”host group” gesetzt ist
elementtype	integer	Typ des Kartenelements. Mögliche Werte: 0 - Host; 1 - Karte; 2 - Auslöser; 3 - Host-Gruppe; 4 - Bild. Verhalten der Eigenschaft: - *erforderlich*
iconid_off	ID	ID des Bildes, das verwendet wird, um das Element im Standardzustand anzuzeigen. Verhalten der Eigenschaft: - *erforderlich*
areatype	integer	Wie separate Hosts einer Host-Gruppe angezeigt werden sollen. Mögliche Werte: 0 - *(Standard)* das Element der Host-Gruppe nimmt die gesamte Karte ein; 1 - das Element der Host-Gruppe hat eine feste Größe.
elementsubtype	integer	Wie ein Host-Gruppen-Element auf einer Karte angezeigt werden soll. Mögliche Werte: 0 - *(Standard)* die Host-Gruppe als einzelnes Element anzeigen; 1 - jeden Host in der Gruppe separat anzeigen.
evaltype	integer	Bedingung für die Tag-Filterung von Kartenelementen Auswertungsmethode. Mögliche Werte: 0 - *(Standard)* Und/Oder; 2 - Oder.
height	integer	Höhe des Host-Gruppen-Elements mit fester Größe in Pixeln. Standard: 200.

Eigenschaft	Typ	Beschreibung
iconid_disabled	ID	ID des Bildes, das verwendet wird, um deaktivierte Kartenelemente anzuzeigen. Verhalten der Eigenschaft: - *unterstützt*, wenn `elementtype` auf ”host”, ”map”, ”trigger” oder ”host group” gesetzt ist
iconid_maintenance	ID	ID des Bildes, das verwendet wird, um Kartenelemente in Wartung anzuzeigen. Verhalten der Eigenschaft: - *unterstützt*, wenn `elementtype` auf ”host”, ”map”, ”trigger” oder ”host group” gesetzt ist
iconid_on	ID	ID des Bildes, das verwendet wird, um Kartenelemente mit Problemen anzuzeigen. Verhalten der Eigenschaft: - *unterstützt*, wenn `elementtype` auf ”host”, ”map”, ”trigger” oder ”host group” gesetzt ist
label	string	Beschriftung des Elements.
label_location	integer	Position der Beschriftung des Kartenelements. Mögliche Werte: -1 - *(Standard)* Standardposition; 0 - unten; 1 - links; 2 - rechts; 3 - oben.
permission	integer	Typ der Berechtigungsstufe. Mögliche Werte: -1 - keine; 2 - schreibgeschützt; 3 - Lesen/Schreiben.
show_label	integer	Wie die Elementbeschriftung angezeigt werden soll. Mögliche Werte: -1 - *(Standard)* Kartenstandard; 0 - immer anzeigen; 1 - automatisch ausblenden.
sysmapid	ID	ID der Karte, zu der das Element gehört. Verhalten der Eigenschaft: - *schreibgeschützt*
urls	array	URLs des Kartenelements. Das URL-Objekt des Kartenelements wird weiter unten im Detail beschrieben.
use_iconmap	integer	Gibt an, ob die Symbolzuordnung für Host-Elemente verwendet werden muss. Mögliche Werte: 0 - Symbolzuordnung nicht verwenden; 1 - *(Standard)* Symbolzuordnung verwenden.
viewtype	integer	Platzierungsalgorithmus für Host-Gruppen-Elemente. Mögliche Werte: 0 - *(Standard)* Raster.
width	integer	Breite des Host-Gruppen-Elements mit fester Größe in Pixeln. Standard: 200.

Eigenschaft	Typ	Beschreibung
x	integer	X-Koordinaten des Elements in Pixeln. Standard: 0.
y	integer	Y-Koordinaten des Elements in Pixeln. Standard: 0.
zindex	integer	Wert, der zum Anordnen von Kartenelementen verwendet wird (z-index). Standard: 0.

Kartenelement Host

Das Objekt „Kartenelement Host“ definiert ein Host-Element.

Eigenschaft	Type	Beschreibung
hostid	ID	ID des Hosts.

Kartenelement Host-Gruppe

Das Objekt des Kartenelements Host-Gruppe definiert ein Host-Gruppenelement.

Eigenschaft	Type	Beschreibung
groupid	ID	ID der Host-Gruppe.

Kartenelement Karte

Das Objekt „Kartenelement Karte“ definiert ein Kartenelement.

Eigenschaft	Typ	Beschreibung
sysmapid	ID	ID der Karte.

Kartenelement Auslöser

Das Objekt des Kartenelements Auslöser definiert ein oder mehrere Auslöser-Elemente.

Eigenschaft	Type	Beschreibung
triggerid	ID	ID des Auslösers.

Tag des Kartenelements

Das Tag-Objekt des Kartenelements hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
tag	string	Name des Tags des Kartenelements. Verhalten der Eigenschaft: - *erforderlich*

Eigenschaft	Typ	Beschreibung
operator	integer	Operator der Tag-Bedingung des Kartenelements. Mögliche Werte: 0 - *(Standard)* Enthält; 1 - Entspricht; 2 - Enthält nicht; 3 - Entspricht nicht; 4 - Existiert; 5 - Existiert nicht.
value	string	Wert des Tags des Kartenelements.

URL des Kartenelements

Das URL-Objekt des Kartenelements definiert einen anklickbaren Link, der für ein bestimmtes Kartenelement verfügbar ist. Es hat die folgenden Eigenschaften:

Eigenschaft	Typ	Beschreibung
sysmapelementurlid	ID	ID der URL des Kartenelements. Verhalten der Eigenschaft: - *schreibgeschützt*
name	string	Beschriftung des Links. Verhalten der Eigenschaft: - *erforderlich*
url	string	URL des Links. Verhalten der Eigenschaft: - *erforderlich*
selementid	ID	ID des Kartenelements, zu dem die URL gehört.

Kartenverknüpfung

Das Objekt der Kartenverknüpfung definiert eine Verknüpfung zwischen zwei Kartenelementen. Es hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
linkid	ID	ID der Kartenverknüpfung. Verhalten der Eigenschaft: - *schreibgeschützt*
sysmapid	ID	ID der Karte, zu der die Verknüpfung gehört.
selementid1	ID	ID des ersten Kartenelements, das an einem Ende verknüpft ist. Verhalten der Eigenschaft: - *erforderlich*
selementid2	ID	ID des ersten Kartenelements, das am anderen Ende verknüpft ist. Verhalten der Eigenschaft: - *erforderlich*
drawtype	integer	Zeichenstil der Verknüpfungslinie. Mögliche Werte: 0 - *(Standard)* Linie; 2 - fette Linie; 3 - gepunktete Linie; 4 - gestrichelte Linie.
color	string	Linienfarbe als hexadezimaler Farbcode. Standard: 000000.

Eigenschaft	Typ	Beschreibung
label	string	Bezeichnung der Verknüpfung.
show_label	integer	Wie die Bezeichnung der Verknüpfung angezeigt werden soll. Mögliche Werte: -1 - *(Standard)* Kartenstandard; 0 - immer anzeigen; 1 - automatisch ausblenden.
indicator_type	integer	Typ des Verknüpfungsindikators auswählen. Mögliche Werte: 0 - *(Standard)* statische Verknüpfung; 1 - Auslöser; 2 - Datenpunktwert.
linktriggers	array	Auslöser der Kartenverknüpfung, die als Indikatoren für den Verknüpfungsstatus verwendet werden. Das Objekt für den Auslöser der Kartenverknüpfung wird weiter unten im Detail beschrieben. Verhalten der Eigenschaft: - *erforderlich*, wenn `indicator_type` auf ”trigger” (1) gesetzt ist.
itemid	ID	ID des Datenpunkts. Verhalten der Eigenschaft: - *erforderlich*, wenn `indicator_type` auf ”item value” (2) gesetzt ist.
highlights	array	Hervorhebungen der Kartenverknüpfung, die als Indikatoren für den Verknüpfungsstatus verwendet werden. Das Objekt für die Indikatoren der Kartenverknüpfung wird weiter unten im Detail beschrieben. Verhalten der Eigenschaft: - *erforderlich*, wenn `indicator_type` auf ”item value” (2) gesetzt ist.
thresholds	array	Schwellenwerte der Kartenverknüpfung, die als Indikatoren für den Verknüpfungsstatus verwendet werden. Das Objekt für die Indikatoren der Kartenverknüpfung wird weiter unten im Detail beschrieben. Verhalten der Eigenschaft: - *erforderlich*, wenn `indicator_type` auf ”item value” (2) gesetzt ist.
permission	integer	Typ der Berechtigungsstufe. Mögliche Werte: -1 - keine; 2 - nur Lesen; 3 - Lesen und Schreiben.

Auslöser für Kartenverknüpfung

Das Objekt für den Auslöser einer Kartenverknüpfung definiert einen Statusindikator für eine Kartenverknüpfung basierend auf dem Zustand eines Auslösers. Es hat die folgenden Eigenschaften:

Eigenschaft	Typ	Beschreibung
triggerid	ID	ID des Auslösers, der als Verknüpfungsindikator verwendet wird. Verhalten der Eigenschaft: - *erforderlich*

Eigenschaft	Typ	Beschreibung
color	string	Farbe des Indikators als hexadezimaler Farbcode. Standard: DD0000.
drawtype	integer	Zeichenstil des Indikators. Mögliche Werte: 0 - *(Standard)* Linie; 2 - fette Linie; 3 - gepunktete Linie; 4 - gestrichelte Linie.

Indikatoren für Kartenverknüpfungen

Das Objekt für Indikatoren von Kartenverknüpfungen definiert einen Statusindikator für eine Kartenverknüpfung basierend auf dem Wert des Datenpunkts. Es hat die folgenden Eigenschaften:

Eigenschaft	Type	Beschreibung
pattern	string	Regulärer Ausdruck für den Vergleich. Nur für Hervorhebungen verfügbar.
sortorder	integer	Wird verwendet, um die Sortierreihenfolge von Hervorhebungen festzulegen. Nur für Hervorhebungen verfügbar.
threshold	string	Numerischer Wert für den Vergleich. Nur für Schwellenwerte verfügbar.
drawtype	integer	Zeichenstil des Indikators. Mögliche Werte: 0 - *(Standard)* Linie; 2 - fette Linie; 3 - gepunktete Linie; 4 - gestrichelte Linie.
color	string	Farbe des Indikators als hexadezimaler Farbcode. Standard: DD0000. Property behavior: - *erforderlich*

Karten-URL

Das Objekt „Karten-URL“ definiert einen anklickbaren Link, der für alle Elemente eines bestimmten Typs auf der Karte verfügbar ist. Es hat die folgenden Eigenschaften:

Eigenschaft	Typ	Beschreibung
sysmapurlid	ID	ID der Karten-URL. Verhalten von Eigenschaften: - *schreibgeschützt*
name	string	Link-Beschriftung. Verhalten von Eigenschaften: - *erforderlich*
url	string	Link-URL. Verhalten von Eigenschaften: - *erforderlich*

Eigenschaft	Typ	Beschreibung
elementtype	integer	Typ des Kartenelements, für das die URL verfügbar sein wird. Eine Liste der unterstützten Typen finden Sie in der Eigenschaft `type` des Kartenelements. Standard: 0.
sysmapid	ID	ID der Karte, zu der die URL gehört.

Kartenbenutzer

Liste der Kartenberechtigungen basierend auf Benutzern. Sie hat die folgenden Eigenschaften:

Eigenschaft	Type	Beschreibung
sysmapuserid	ID	ID des Kartenbenutzers. Property behavior: - *read-only*
userid	ID	ID des Benutzers. Property behavior: - *required*
permission	integer	Typ der Berechtigungsstufe. Mögliche Werte: 2 - nur Lesen; 3 - Lesen und Schreiben. Property behavior: - *required*

Karten-Benutzergruppe

Liste der Kartenberechtigungen basierend auf Benutzergruppen. Sie hat die folgenden Eigenschaften:

Eigenschaft	Type	Beschreibung
sysmapusrgrpid	ID	ID der Karten-Benutzergruppe. Property behavior: - *read-only*
usrgrpid	ID	ID der Benutzergruppe. Property behavior: - *required*
permission	integer	Typ der Berechtigungsstufe. Mögliche Werte: 2 - nur lesen; 3 - Lesen und Schreiben. Property behavior: - *required*

Kartenformen

Das Objekt für Kartenformen definiert eine geometrische Form (mit oder ohne Text), die auf einer Karte angezeigt wird. Es hat die folgenden Eigenschaften:

Eigenschaft	Typ	Beschreibung
sysmap_shapeid	ID	ID des Kartenform-Elements. Verhalten der Eigenschaft: - *schreibgeschützt*
type	integer	Typ des Kartenform-Elements. Mögliche Werte: 0 - Rechteck; 1 - Ellipse. Die Eigenschaft ist erforderlich, wenn neue Formen erstellt werden. Verhalten der Eigenschaft: - *erforderlich*
x	integer	X-Koordinaten der Form in Pixeln. Standard: 0.
y	integer	Y-Koordinaten der Form in Pixeln. Standard: 0.
width	integer	Breite der Form in Pixeln. Standard: 200.
height	integer	Höhe der Form in Pixeln. Standard: 200.
text	string	Text der Form.
font	integer	Schriftart des Textes innerhalb der Form. Mögliche Werte: 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 Standard: 9.
font_size	integer	Schriftgröße in Pixeln. Standard: 11.
font_color	string	Schriftfarbe. Standard: 000000.
text_halign	integer	Horizontale Ausrichtung des Textes. Mögliche Werte: 0 - zentriert; 1 - links; 2 - rechts. Standard: 0.

Eigenschaft	Typ	Beschreibung
text_valign	integer	Vertikale Ausrichtung des Textes. Mögliche Werte: 0 - mittig; 1 - oben; 2 - unten. Standard: 0.
border_type	integer	Typ des Rahmens. Mögliche Werte: 0 - keiner; 1 - ──────; 2 - ┅┅; 3 - ╴ ╴ ╴. Standard: 0.
border_width	integer	Breite des Rahmens in Pixeln. Standard: 0.
border_color	string	Rahmenfarbe. Standard: 000000.
background_color	string	Hintergrundfarbe (Füllfarbe). Standard: `(leer)`.
zindex	integer	Wert, der zum Anordnen aller Formen und Linien verwendet wird (z-index). Standard: 0.

Kartenlinien

Das Objekt für Kartenlinien definiert eine Linie, die auf einer Karte angezeigt wird. Es hat die folgenden Eigenschaften:

Eigenschaft	Typ	Beschreibung
sysmap_shapeid	ID	ID des Kartenelement-Formobjekts. Eigenschaftsverhalten: - *schreibgeschützt*
x1	integer	X-Koordinaten des ersten Linienpunkts in Pixeln. Standard: 0.
y1	integer	Y-Koordinaten des ersten Linienpunkts in Pixeln. Standard: 0.
x2	integer	X-Koordinaten des zweiten Linienpunkts in Pixeln. Standard: 200.
y2	integer	Y-Koordinaten des zweiten Linienpunkts in Pixeln. Standard: 200.

Eigenschaft	Typ	Beschreibung
line_type	integer	Typ der Linien. Mögliche Werte: 0 - keine; 1 - ————; 2 - ┅; 3 - – – –. Standard: 0.
line_width	integer	Breite der Linien in Pixeln. Standard: 0.
line_color	string	Linienfarbe. Standard: 000000.
zindex	integer	Wert, der zur Anordnung aller Formen und Linien verwendet wird (z-index). Standard: 0.

map.create

Beschreibung

`object map.create(object/array maps)`

Diese Methode ermöglicht es Ihnen, neue Maps zu erstellen.

Note:
Diese Methode ist für Benutzer aller Typen verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellende Karten.

Zusätzlich zu den Standard-Karteneigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
links	array	Kartenverknüpfungen, die auf der Karte erstellt werden sollen.
selements	array	Kartenelemente, die auf der Karte erstellt werden sollen.
urls	array	Karten-URLs, die auf der Karte erstellt werden sollen.
users	array	Freigaben für Kartenbenutzer, die auf der Karte erstellt werden sollen.
userGroups	array	Freigaben für Karten-Benutzergruppen, die auf der Karte erstellt werden sollen.
shapes	array	Kartenformen, die auf der Karte erstellt werden sollen.
lines	array	Kartenlinien, die auf der Karte erstellt werden sollen.

Note:
Um Kartenverknüpfungen zu erstellen, müssen Sie für ein Kartenelement `selementid` auf einen beliebigen Wert setzen und diesen Wert dann verwenden, um dieses Element in den Eigenschaften `selementid1` oder `selementid2` der Verknüpfungen zu referenzieren. Wenn das Element erstellt wird, wird dieser Wert durch die korrekte von Zabbix generierte ID ersetzt. Siehe Beispiel.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Maps unter der Eigenschaft `sysmapids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Maps.

Beispiele

Eine leere Karte erstellen

Erstellen Sie eine Karte ohne Elemente.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Map",
        "width": 600,
        "height": 600
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "8"
        ]
    },
    "id": 1
}
```

Eine Host-Karte erstellen

Erstellen Sie eine Karte mit zwei Host-Elementen und einer Verbindung zwischen ihnen. Beachten Sie die Verwendung der temporären Werte „selementid1“ und „selementid2“ im Kartenverknüpfungsobjekt, um auf Kartenelemente zu verweisen.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "9"
        ]
    },
    "id": 1
}
```

Eine Auslöser-Karte erstellen

Erstellen Sie eine Karte mit einem Auslöser-Element, das zwei Auslöser enthält.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "10"
        ]
    },
    "id": 1
}
```

Freigabe von Karten

Erstellen Sie eine Karte mit zwei Freigabetypen (Benutzer und Benutzergruppe).

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "9"
        ]
    },
    "id": 1
}
```

Kartenformen

Erstellen Sie eine Karte mit dem Kartennamen als Titel.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "10"
        ]
    },
    "id": 1
```

```
}
```

Kartenlinien

Erstellen Sie eine Kartenlinie.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "11"
        ]
    },
    "id": 1
}
```

Siehe auch

- Kartenelement
- Kartenverknüpfung
- Karten-URL
- Kartenbenutzer
- Kartenbenutzergruppe
- Kartenform
- Kartenlinie

Quelle

CMap::create() in *ui/include/classes/api/services/CMap.php*.

map.delete

Beschreibung

`object map.delete(array mapIds)`

Diese Methode ermöglicht es Ihnen, Karten zu löschen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der zu löschenden Karten.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Maps unter der Eigenschaft `sysmapids` enthält.

Beispiele

Mehrere Karten löschen

Löschen Sie zwei Karten.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "map.delete",
    "params": [
        "12",
        "34"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "12",
            "34"
        ]
    },
    "id": 1
}
```

Quelle

CMap::delete() in *ui/include/classes/api/services/CMap.php*.

map.get

Beschreibung

`integer/array map.get(object parameters)`

Die Methode ermöglicht es Ihnen, Maps anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
sysmapids	ID/array	Gibt nur Karten mit den angegebenen IDs zurück.
userids	ID/array	Gibt nur Karten zurück, die zu den angegebenen Benutzer-IDs gehören.
expandUrls	flag	Fügt globale Karten-URLs zu den entsprechenden Kartenelementen hinzu und erweitert Makros in allen URLs der Kartenelemente.
selectIconMap	query	Gibt eine Eigenschaft `iconmap` mit der auf der Karte verwendeten Symbolzuordnung zurück.

Parameter	Type	Beschreibung
selectLinks	query	Gibt eine Eigenschaft `links` mit den Kartenverknüpfungen zwischen Elementen zurück.
selectSelements	query	Gibt eine Eigenschaft `selements` mit den Kartenelementen zurück.
selectUrls	query	Gibt eine Eigenschaft `urls` mit den Karten-URLs zurück.
selectUsers	query	Gibt eine Eigenschaft `users` mit Benutzern zurück, für die die Karte freigegeben ist.
selectUserGroups	query	Gibt eine Eigenschaft `userGroups` mit Benutzergruppen zurück, für die die Karte freigegeben ist.
selectShapes	query	Gibt eine Eigenschaft `shapes` mit den Kartenformen zurück.
selectLines	query	Gibt eine Eigenschaft `lines` mit den Kartenlinien zurück.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `name`, `width`, `height`.
countOutput	boolean	Diese Parameter werden in der Referenzkommentierung beschrieben.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Kann die folgenden Dinge zurück geben:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Eine Karte abrufen

Rufen Sie alle Daten zur Karte „3“ ab.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "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
}
```

Siehe auch

- Symbolzuordnung
- Kartenelement
- Kartenverknüpfung
- Karten-URL
- Kartenbenutzer
- Karten-Benutzergruppe
- Kartenformen
- Kartenlinien

Quelle

CMap::get() in *ui/include/classes/api/services/CMap.php*.

map.update

Beschreibung

`object map.update(object/array maps)`

Diese Methode ermöglicht es Ihnen, vorhandene Maps zu aktualisieren.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu aktualisierende Karteneigenschaften.

Die Eigenschaft `mapid` muss für jede Karte definiert werden, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Zusätzlich zu den standardmäßigen Karteneigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
links	array	Kartenverknüpfungen, die die vorhandenen Verknüpfungen ersetzen.
selements	array	Kartenelemente, die die vorhandenen Elemente ersetzen.
urls	array	Karten-URLs, die die vorhandenen URLs ersetzen.
users	array	Kartenbenutzer-Freigaben, die die vorhandenen Elemente ersetzen.
userGroups	array	Freigaben für Karten-Benutzergruppen, die die vorhandenen Elemente ersetzen.
shapes	array	Kartenformen, die die vorhandenen Formen ersetzen.
lines	array	Kartenlinien, die die vorhandenen Linien ersetzen.

> **Note:**
> Um Kartenverknüpfungen zwischen neuen Kartenelementen zu erstellen, müssen Sie die `selementid` eines Elements auf einen beliebigen Wert setzen und dann diesen Wert verwenden, um in den Eigenschaften `selementid1` oder `selementid2` der Verknüpfungen auf dieses Element zu verweisen. Wenn das Element erstellt wird, wird dieser Wert durch die korrekte von Zabbix generierte ID ersetzt. Siehe Beispiel für map.create.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Maps unter der Eigenschaft `sysmapids` enthält.

Beispiele

Größe einer Karte ändern

Ändern Sie die Größe der Karte auf 1200x1200 Pixel.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "map.update",
    "params": {
        "sysmapid": "8",
        "width": 1200,
        "height": 1200
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "8"
        ]
    },
    "id": 1
}
```

Eigentümer der Karte ändern

Nur für Admins und Super-Admins verfügbar.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "map.update",
    "params": {
        "sysmapid": "9",
        "userid": "1"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "9"
        ]
    },
    "id": 1
}
```

Siehe auch

- Kartenelement
- Kartenverknüpfung
- Karten-URL
- Kartenbenutzer
- Kartenbenutzergruppe
- Kartenformen
- Kartenlinien

Quelle

CMap::update() in *ui/include/classes/api/services/CMap.php*.

Konfiguration

Diese Klasse dient dem Export und Import von Zabbix-Konfigurationsdaten.

Verfügbare Methoden:

- configuration.export - Konfigurationsdaten exportieren
- configuration.import - Konfigurationsdaten importieren
- configuration.importcompare - Importdatei mit den aktuellen Systemelementen vergleichen

configuration.export

Beschreibung

`string configuration.export(object parameters)`

Diese Methode ermöglicht es, Konfigurationsdaten als serialisierte Zeichenfolge zu exportieren.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die zu exportierenden Objekte und das zu verwendende Format definieren.

Parameter	Typ	Beschreibung
format	string	Format, in dem die Daten exportiert werden müssen. Mögliche Werte: `yaml` - YAML; `xml` - XML; `json` - JSON; `raw` - unverarbeitetes PHP-Array. Verhalten des Parameters: - *erforderlich*
prettyprint	boolean	Macht die Ausgabe durch Einrückung besser lesbar. Mögliche Werte: `true` - Einrückung hinzufügen; `false` - *(Standard)* keine Einrückung hinzufügen.

Parameter	Typ	Beschreibung
options	object	Zu exportierende Objekte. Das Objekt `options` hat die folgenden Parameter: `host_groups` - `(array)` IDs der zu exportierenden Hostgruppen; `hosts` - `(array)` IDs der zu exportierenden Hosts; `images` - `(array)` IDs der zu exportierenden Bilder; `maps` - `(array)` IDs der zu exportierenden Karten; `mediaTypes` - `(array)` IDs der zu exportierenden Medientypen; `template_groups` - `(array)` IDs der zu exportierenden Vorlagengruppen; `templates` - `(array)` IDs der zu exportierenden Vorlagen; `dashboards` - `(array)` IDs der zu exportierenden Dashboards; `global_regexes` - `(array)` IDs der zu exportierenden globalen regulären Ausdrücke. Benutzer vom Typ *Admin* und *User* dürfen nur diejenigen Objekte exportieren, für die sie über die Berechtigung *nur lesen* oder *lesen und schreiben* Berechtigung verfügen, sowie Bilder, jedoch keine Medientypen. Verhalten des Parameters: - *erforderlich*.

Rückgabewerte

`(string)` Gibt eine serialisierte Zeichenfolge zurück, welche die angeforderten Konfigurationsdaten enthält.

Beispiele

Exportieren einer Vorlage

Exportieren Sie die Konfiguration der Vorlage „10571“ als XML-Zeichenfolge.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "configuration.export",
    "params": {
        "options": {
            "templates": [
                "10571"
            ]
        },
        "format": "xml"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<zabbix_export><version>8.0</version><template_
    "id": 1
}
```


Quelle

CConfiguration::export() in *ui/include/classes/api/services/CConfiguration.php*.

configuration.import

Beschreibung

`boolean configuration.import(object parameters)`

Diese Methode ermöglicht es, Konfigurationsdaten aus einer serialisierten Zeichenfolge zu importieren.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die zu importierenden Daten und die Regeln enthalten, wie die Daten verarbeitet werden sollen.

Parameter	Type	Beschreibung
format	string	Format der serialisierten Zeichenfolge. Mögliche Werte: `yaml` - YAML; `xml` - XML; `json` - JSON. Parameterverhalten: - *erforderlich*
source	string	Serialisierte Zeichenfolge, die die Konfigurationsdaten enthält. Parameterverhalten: - *erforderlich*
rules	object	Regeln dafür, wie neue und vorhandene Objekte importiert werden sollen. Benutzer vom Typ *Admin* dürfen nur diejenigen Objekte importieren, für die sie über die Berechtigung *read-write* permission verfügen, sowie Karten. Beispielsweise können ein Host und seine Entitäten (Datenpunkte, Auslöser, Diagramme usw.) nur importiert werden, wenn die Benutzergruppe des Benutzers über Berechtigungen für die Hostgruppe verfügt, zu der der importierte Host gehören wird. Bilder und Medientypen können von Benutzern vom Typ *Admin* nicht importiert werden. Der Parameter `rules` wird in der folgenden Tabelle ausführlich beschrieben. Parameterverhalten: - *erforderlich*
returnMissingObjects	boolean	Regeln, ob das Antwortobjekt Informationen über fehlende Objekte enthalten soll; Standard: `false`.

> **Attention:**
> Wenn keine Regeln angegeben werden, wird die Konfiguration nicht aktualisiert.

Das Objekt `rules` unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
dashboards	object	Regeln für den Import von Dashboards. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Dashboards erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt und ein bestimmtes importierbares Dashboard im System vorhanden ist, wird es vollständig aktualisiert. Seiten und Widgets werden vollständig durch die neuen ersetzt; Standard: `false`.
discoveryRules	object	Regeln für den Import von LLD-Regeln. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue LLD-Regeln erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden vorhandene LLD-Regeln aktualisiert; Standard: `false`; `deleteMissing` - `(boolean)` wenn auf `true` gesetzt, werden LLD-Regeln, die in den importierten Daten nicht vorhanden sind, aus der Datenbank gelöscht; Standard: `false`.
global_regexes	object	Regeln für den Import globaler regulärer Ausdrücke. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue globale reguläre Ausdrücke erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden vorhandene globale reguläre Ausdrücke aktualisiert, wobei ihre Unterausdrücke vollständig überschrieben werden; Standard: `false`.
graphs	object	Regeln für den Import von Diagrammen. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Diagramme erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden vorhandene Diagramme aktualisiert; Standard: `false`; `deleteMissing` - `(boolean)` wenn auf `true` gesetzt, werden Diagramme, die in den importierten Daten nicht vorhanden sind, aus der Datenbank gelöscht; Standard: `false`.
host_groups	object	Regeln für den Import von Hostgruppen. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Hostgruppen erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden vorhandene Hostgruppen aktualisiert; Standard: `false`.
hosts	object	Regeln für den Import von Hosts. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Hosts erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden vorhandene Hosts aktualisiert; Standard: `false`.

Parameter	Type	Beschreibung
httptests	object	Regeln für den Import von Web-Szenarien. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Web-Szenarien erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden vorhandene Web-Szenarien aktualisiert; Standard: `false`; `deleteMissing` - `(boolean)` wenn auf `true` gesetzt, werden Web-Szenarien, die in den importierten Daten nicht vorhanden sind, aus der Datenbank gelöscht; Standard: `false`.
images	object	Regeln für den Import von Bildern. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Bilder erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden vorhandene Bilder aktualisiert; Standard: `false`.
items	object	Regeln für den Import von Datenpunkten. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Datenpunkte erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden vorhandene Datenpunkte aktualisiert; Standard: `false`; `deleteMissing` - `(boolean)` wenn auf `true` gesetzt, werden Datenpunkte, die in den importierten Daten nicht vorhanden sind, aus der Datenbank gelöscht; Standard: `false`.
maps	object	Regeln für den Import von Karten. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Karten erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden vorhandene Karten aktualisiert; Standard: `false`.
mediaTypes	object	Regeln für den Import von Medientypen. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Medientypen erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden vorhandene Medientypen aktualisiert; Standard: `false`.
templateDashboards	object	Regeln für den Import von Vorlagen-Dashboards. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Vorlagen-Dashboards erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden vorhandene Vorlagen-Dashboards aktualisiert; Standard: `false`; `deleteMissing` - `(boolean)` wenn auf `true` gesetzt, werden Vorlagen-Dashboards, die in den importierten Daten nicht vorhanden sind, aus der Datenbank gelöscht; Standard: `false`.
template_groups	object	Regeln für den Import von Vorlagengruppen. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Vorlagengruppen erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden vorhandene Vorlagengruppen aktualisiert; Standard: `false`.

Parameter	Type	Beschreibung
templateLinkage	object	Regeln für den Import von Vorlagenverknüpfungen. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden Vorlagen, die nicht mit dem zu importierenden Host oder der zu importierenden Vorlage verknüpft sind, aber in den importierten Daten vorhanden sind, verknüpft; Standard: `false`; `deleteMissing` - `(boolean)` wenn auf `true` gesetzt, werden Vorlagen, die mit dem zu importierenden Host oder der zu importierenden Vorlage verknüpft sind, aber in den importierten Daten nicht vorhanden sind, ohne Entfernen der von den getrennten Vorlagen geerbten Entitäten (Datenpunkte, Auslöser usw.) getrennt; Standard: `false`.
templates	object	Regeln für den Import von Vorlagen. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Vorlagen erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden vorhandene Vorlagen aktualisiert; Standard: `false`.
triggers	object	Regeln für den Import von Auslösern. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Auslöser erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden vorhandene Auslöser aktualisiert; Standard: `false`; `deleteMissing` - `(boolean)` wenn auf `true` gesetzt, werden Auslöser, die in den importierten Daten nicht vorhanden sind, aus der Datenbank gelöscht; Standard: `false`.
valueMaps	object	Regeln für den Import von Host- oder Vorlagen-Wertzuordnungen. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Wertzuordnungen erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden vorhandene Wertzuordnungen aktualisiert; Standard: `false`; `deleteMissing` - `(boolean)` wenn auf `true` gesetzt, werden Wertzuordnungen, die in den importierten Daten nicht vorhanden sind, aus der Datenbank gelöscht; Standard: `false`.

Rückgabewerte

`(boolean)` gibt `true` zurück, wenn der Importvorgang erfolgreich war.

Beispiele

Importieren einer Vorlage

Importieren Sie die in der XML-Zeichenfolge enthaltene Vorlagenkonfiguration. Falls in der XML-Zeichenfolge Datenpunkte oder Auslöser fehlen, werden diese aus der Datenbank gelöscht, und alles andere bleibt unverändert.

Anfrage:

```
{
    "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
}
```


Antwort:


```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```


Quelle

CConfiguration::import() in *ui/include/classes/api/services/CConfiguration.php*.

configuration.importcompare

Beschreibung

`array configuration.importcompare(object parameters)`

Diese Methode ermöglicht es Ihnen, die Importdatei mit den aktuellen Systemelementen zu vergleichen und zeigt an, was geändert wird, wenn diese Importdatei importiert wird.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die möglichen zu importierenden Daten und die Regeln enthalten, wie die Daten verarbeitet werden sollen.

Parameter	Type	Beschreibung
format	string	Format der serialisierten Zeichenfolge. Mögliche Werte: `yaml` - YAML; `xml` - XML; `json` - JSON. Parameterverhalten: - *erforderlich*
source	string	Serialisierte Zeichenfolge, die die Konfigurationsdaten enthält. Parameterverhalten: - *erforderlich*

Parameter	Type	Beschreibung
rules	object	Regeln dafür, wie neue und vorhandene Objekte verglichen werden sollen. Benutzer vom Typ *Admin* und *User* dürfen neue Objekte nur mit vorhandenen Objekten vergleichen, für die sie über die Berechtigung *nur lesen* oder *lesen und schreiben* permission verfügen. Der Parameter `rules` wird in der folgenden Tabelle ausführlich beschrieben. Parameterverhalten: - *erforderlich*
returnMissingObjects	boolean	Regeln, ob das Antwortobjekt Informationen über fehlende Objekte enthalten soll; Standard: `false`.

Attention:
Wenn keine Regeln angegeben werden, gibt es nichts zu aktualisieren und das Ergebnis ist leer.

Note:
Der Vergleich wird nur für Hostgruppen und Vorlagen durchgeführt. Auslöser und Diagramme werden nur für importierte Vorlagen verglichen, alle anderen werden als ”neu” betrachtet.

Das Objekt `rules` unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
dashboards	object	Regeln für den Import von Dashboards. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Dashboards erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt und ein bestimmtes importierbares Dashboard im System vorhanden ist, wird es vollständig aktualisiert. Seiten und Widgets werden vollständig durch die neuen ersetzt; Standard: `false`.
discoveryRules	object	Regeln für den Import von LLD-Regeln. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue LLD-Regeln erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden vorhandene LLD-Regeln aktualisiert; Standard: `false`; `deleteMissing` - `(boolean)` wenn auf `true` gesetzt, werden in den importierten Daten nicht vorhandene LLD-Regeln aus der Datenbank gelöscht; Standard: `false`.
global_regexes	object	Regeln für den Import globaler regulärer Ausdrücke. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue globale reguläre Ausdrücke erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden vorhandene globale reguläre Ausdrücke aktualisiert, wobei ihre Unterausdrücke vollständig überschrieben werden; Standard: `false`.

Parameter	Type	Beschreibung
graphs	object	Regeln für den Import von Diagrammen. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Diagramme erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden vorhandene Diagramme aktualisiert; Standard: `false`; `deleteMissing` - `(boolean)` wenn auf `true` gesetzt, werden in den importierten Daten nicht vorhandene Diagramme aus der Datenbank gelöscht; Standard: `false`.
host_groups	object	Regeln für den Import von Hostgruppen. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Hostgruppen erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden vorhandene Hostgruppen aktualisiert; Standard: `false`.
httptests	object	Regeln für den Import von Web-Szenarien. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Web-Szenarien erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden vorhandene Web-Szenarien aktualisiert; Standard: `false`; `deleteMissing` - `(boolean)` wenn auf `true` gesetzt, werden in den importierten Daten nicht vorhandene Web-Szenarien aus der Datenbank gelöscht; Standard: `false`.
images	object	Regeln für den Import von Bildern. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Bilder erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden vorhandene Bilder aktualisiert; Standard: `false`. Dieser Parameter hat keine Auswirkung auf die Ausgabe. Er ist nur aus Gründen der Konsistenz mit `configuration.import` zulässig.
items	object	Regeln für den Import von Datenpunkten. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Datenpunkte erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden vorhandene Datenpunkte aktualisiert; Standard: `false`; `deleteMissing` - `(boolean)` wenn auf `true` gesetzt, werden in den importierten Daten nicht vorhandene Datenpunkte aus der Datenbank gelöscht; Standard: `false`.
maps	object	Regeln für den Import von Karten. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Karten erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden vorhandene Karten aktualisiert; Standard: `false`. Dieser Parameter hat keine Auswirkung auf die Ausgabe. Er ist nur aus Gründen der Konsistenz mit `configuration.import` zulässig.

Parameter	Type	Beschreibung
mediaTypes	object	Regeln für den Import von Medientypen. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Medientypen erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden vorhandene Medientypen aktualisiert; Standard: `false`. Dieser Parameter hat keine Auswirkung auf die Ausgabe. Er ist nur aus Gründen der Konsistenz mit `configuration.import` zulässig.
templateDashboards	object	Regeln für den Import von Vorlagen-Dashboards. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Vorlagen-Dashboards erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden vorhandene Vorlagen-Dashboards aktualisiert; Standard: `false`; `deleteMissing` - `(boolean)` wenn auf `true` gesetzt, werden in den importierten Daten nicht vorhandene Vorlagen-Dashboards aus der Datenbank gelöscht; Standard: `false`.
template_groups	object	Regeln für den Import von Vorlagengruppen. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Vorlagengruppen erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden vorhandene Vorlagengruppen aktualisiert; Standard: `false`.
templateLinkage	object	Regeln für den Import von Vorlagenverknüpfungen. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden Vorlagen, die nicht mit dem zu importierenden Host oder der zu importierenden Vorlage verknüpft sind, aber in den importierten Daten vorhanden sind, verknüpft; Standard: `false`; `deleteMissing` - `(boolean)` wenn auf `true` gesetzt, werden Vorlagen, die mit dem zu importierenden Host oder der zu importierenden Vorlage verknüpft sind, aber in den importierten Daten nicht vorhanden sind, getrennt, ohne von den getrennten Vorlagen geerbte Entitäten (Datenpunkte, Auslöser usw.) zu entfernen; Standard: `false`.
templates	object	Regeln für den Import von Vorlagen. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Vorlagen erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden vorhandene Vorlagen aktualisiert; Standard: `false`.
triggers	object	Regeln für den Import von Auslösern. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Auslöser erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden vorhandene Auslöser aktualisiert; Standard: `false`; `deleteMissing` - `(boolean)` wenn auf `true` gesetzt, werden in den importierten Daten nicht vorhandene Auslöser aus der Datenbank gelöscht; Standard: `false`.

Parameter	Type	Beschreibung
valueMaps	object	Regeln für den Import von Wertzuordnungen für Host oder Vorlage. Unterstützte Parameter: `createMissing` - `(boolean)` wenn auf `true` gesetzt, werden neue Wertzuordnungen erstellt; Standard: `false`; `updateExisting` - `(boolean)` wenn auf `true` gesetzt, werden vorhandene Wertzuordnungen aktualisiert; Standard: `false`; `deleteMissing` - `(boolean)` wenn auf `true` gesetzt, werden in den importierten Daten nicht vorhandene Wertzuordnungen aus der Datenbank gelöscht; Standard: `false`.

Rückgabewerte

`(array)` Gibt ein Array mit den Änderungen in der Konfiguration zurück, die vorgenommen werden.

Beispiele

Vergleich des Imports einer Vorlage

Vergleichen Sie die in der XML-Zeichenfolge enthaltene Vorlage mit den aktuellen Systemelementen und zeigen Sie, was geändert wird, wenn diese Vorlage importiert wird.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "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
}
```


Quelle

CConfiguration::importcompare() in *ui/include/classes/api/services/CConfiguration.php*.

Korrelation

Diese Klasse ist für die Arbeit mit Korrelationen vorgesehen.

Objektreferenzen:

- Korrelation
- Korrelationsoperation
- Korrelationsfilter
  - Korrelationsfilterbedingung

Verfügbare Methoden:

- correlation.create - neue Korrelationen erstellen
- correlation.delete - Korrelationen löschen
- correlation.get - Korrelationen abrufen
- correlation.update - Korrelationen aktualisieren

Korrelationsobjekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `correlation` API.

Korrelation

Das Korrelationsobjekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
correlationid	ID	ID der Korrelation. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
name	string	Name der Korrelation. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
description	string	Beschreibung der Korrelation.
status	integer	Gibt an, ob die Korrelation aktiviert oder deaktiviert ist. Mögliche Werte: 0 - *(Standard)* aktiviert; 1 - deaktiviert.

Korrelationsoperation

Das Objekt für die Korrelationsoperation definiert eine Operation, die ausgeführt wird, wenn eine Korrelation ausgeführt wird. Es hat die folgenden Eigenschaften.

Property	Type	Description
type	integer	Typ der Operation. Mögliche Werte: 0 - alte Ereignisse schließen; 1 - neues Ereignis schließen. Verhalten der Eigenschaft: - *erforderlich*

Korrelationsfilter

Das Objekt Korrelationsfilter definiert eine Reihe von Bedingungen, die erfüllt sein müssen, um die konfigurierten Korrelationsoperationen auszuführen. Es hat die folgenden Eigenschaften.

Property	Type	Description
conditions	array	Menge von Filterbedingungen, die zum Filtern von Ergebnissen verwendet werden. Die Bedingungen werden in der Reihenfolge ihrer Platzierung in der Formel sortiert. Property behavior: - *required*
evaltype	integer	Auswertungsmethode der Filterbedingung. Mögliche Werte: 0 - And/Or; 1 - And; 2 - Or; 3 - Benutzerdefinierter Ausdruck. Property behavior: - *required*
eval_formula	string	Generierter Ausdruck, der zur Auswertung der Filterbedingungen verwendet wird. Der Ausdruck enthält IDs, die auf bestimmte Filterbedingungen über deren `formulaid` verweisen. Der Wert von `eval_formula` entspricht dem Wert von `formula` für Filter mit einem benutzerdefinierten Ausdruck. Property behavior: - *read-only*
formula	string	Vom Benutzer definierter Ausdruck, der zur Auswertung von Bedingungen von Filtern mit einem benutzerdefinierten Ausdruck verwendet wird. Der Ausdruck muss IDs enthalten, die auf bestimmte Filterbedingungen über deren `formulaid` verweisen. Die im Ausdruck verwendeten IDs müssen exakt mit den in den Filterbedingungen definierten übereinstimmen: Keine Bedingung darf ungenutzt bleiben oder ausgelassen werden. Property behavior: - *required* wenn `evaltype` auf ”custom expression” gesetzt ist

Korrelationsfilterbedingung

Das Objekt für die Korrelationsfilterbedingung definiert eine bestimmte Bedingung, die vor der Ausführung der Korrelationsoperationen geprüft werden muss.

Eigenschaft	Typ	Beschreibung
type	integer	Typ der Bedingung. Mögliche Werte: 0 - altes Ereignis-Tag; 1 - neues Ereignis-Tag; 2 - neue Ereignis-Hostgruppe; 3 - Ereignis-Tag-Paar; 4 - Wert des alten Ereignis-Tags; 5 - Wert des neuen Ereignis-Tags. Verhalten der Eigenschaft: - *erforderlich*
tag	string	Ereignis-Tag (alt oder neu). Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „altes Ereignis-Tag“, „neues Ereignis-Tag“, „Wert des alten Ereignis-Tags“ oder „Wert des neuen Ereignis-Tags“ gesetzt ist
groupid	ID	ID der Hostgruppe. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „neue Ereignis-Hostgruppe“ gesetzt ist
oldtag	string	Altes Ereignis-Tag. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „Ereignis-Tag-Paar“ gesetzt ist
newtag	string	Altes Ereignis-Tag. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „Ereignis-Tag-Paar“ gesetzt ist
value	string	Wert des Ereignis-Tags (alt oder neu). Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „Wert des alten Ereignis-Tags“ oder „Wert des neuen Ereignis-Tags“ gesetzt ist
formulaid	string	Beliebige eindeutige ID, die verwendet wird, um aus einem benutzerdefinierten Ausdruck auf die Bedingung zu verweisen. Darf nur Großbuchstaben enthalten. Die ID muss vom Benutzer definiert werden, wenn Filterbedingungen geändert werden, wird jedoch bei einer späteren Abfrage erneut generiert.
operator	integer	Bedingungsoperator. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „neue Ereignis-Hostgruppe“, „Wert des alten Ereignis-Tags“ oder „Wert des neuen Ereignis-Tags“ gesetzt ist

Note:
Um besser zu verstehen, wie Filter mit verschiedenen Ausdruckstypen verwendet werden, siehe die Beispiele auf den Seiten der Methoden correlation.get und correlation.create.

Die folgenden Operatoren und Werte werden für jeden Bedingungstyp unterstützt.

Bedingung	Bedingungsname	Unterstützte Operatoren	Erwarteter Wert
2	Hostgruppe	=, <>	Hostgruppen-ID.
4	Wert des alten Ereignis-Tags	=, <>, like, not like	string
5	Wert des neuen Ereignis-Tags	=, <>, like, not like	string

correlation.create

Beschreibung

`object correlation.create(object/array correlations)`

Diese Methode ermöglicht es Ihnen, neue Korrelationen zu erstellen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellende Korrelationen.

Zusätzlich zu den standardmäßigen Korrelationseigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
operations	array	Zu erstellende Korrelationsoperationen für die Korrelation. Parameterverhalten: - *erforderlich*
filter	object	Objekt des Korrelationsfilters für die Korrelation. Parameterverhalten: - *erforderlich*

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Korrelationen unter der Eigenschaft `correlationids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Korrelationen.

Beispiele

Eine neue Ereignis-Tag-Korrelation erstellen

Erstellen Sie eine Korrelation mit der Auswertungsmethode `AND/OR` mit einer Bedingung und einer Operation. Standardmäßig ist die Korrelation aktiviert.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
```

```
    "result": {
        "correlationids": [
            "1"
        ]
    },
    "id": 1
}
```

Verwenden eines benutzerdefinierten Ausdrucksfilters

Erstellen Sie eine Korrelation, die eine benutzerdefinierte Filterbedingung verwendet. Die Formel-IDs „A“ oder „B“ wurden willkürlich gewählt. Der Bedingungstyp ist „Host-Gruppe“ mit dem Operator „<>“.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "2"
        ]
    },
    "id": 1
}
```

Siehe auch

- Korrelationsfilter
- Korrelationsoperation

Quelle

CCorrelation::create() in *ui/include/classes/api/services/CCorrelation.php*.

correlation.delete

Beschreibung

`object correlation.delete(array correlationids)`

Diese Methode ermöglicht es Ihnen, Korrelationen zu löschen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der zu löschenden Korrelationen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Korrelationen unter der Eigenschaft `correlationids` enthält.

Beispiel

Mehrere Korrelationen löschen

Löschen Sie zwei Korrelationen.

Anrfage:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.delete",
    "params": [
        "1",
        "2"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

Quelle

CCorrelation::delete() in *ui/include/classes/api/services/CCorrelation.php*.

correlation.get

Beschreibung

`integer/array correlation.get(object parameters)`

Die Methode ermöglicht es Ihnen, Korrelationen anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
correlationids	ID/array	Gibt nur Korrelationen mit den angegebenen IDs zurück.
selectFilter	query	Gibt eine Eigenschaft `filter` mit den Korrelationsbedingungen zurück.
selectOperations	query	Gibt eine Eigenschaft `operations` mit den Korrelationsoperationen zurück.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `correlationid`, `name`, `status`.
countOutput	boolean	Diese Parameter sind in der Referenzkommentierung beschrieben.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Gibt entweder:

- ein Array von Objekten zurück;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Korrelationen abrufen

Rufen Sie alle konfigurierten Korrelationen zusammen mit den Korrelationsbedingungen und Operationen ab. Der Filter verwendet den Auswertungstyp „and/or“, daher ist die Eigenschaft `formula` leer und `eval_formula` wird automatisch generiert.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.get",
    "params": {
        "output": "extend",
        "selectOperations": "extend",
        "selectFilter": "extend"
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```

Siehe auch

- Korrelationsfilter
- Korrelationsoperation

Quelle

CCorrelation::get() in *ui/include/classes/api/services/CCorrelation.php*.

correlation.update

Beschreibung

`object correlation.update(object/array correlations)`

Diese Methode ermöglicht es Ihnen, vorhandene Korrelationen zu aktualisieren.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu aktualisierende Korrelationseigenschaften.

Die Eigenschaft `correlationid` muss für jede Korrelation definiert sein, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Zusätzlich zu den standardmäßigen Korrelationseigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Typ	Beschreibung
filter	object	Objekt Korrelationsfilter, das den aktuellen Filter ersetzt.
operations	array	Korrelationsoperationen, die bestehende Operationen ersetzen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Korrelationen unter der Eigenschaft `correlationids` enthält.

Beispiele

Korrelation deaktivieren

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.update",
    "params": {
        "correlationid": "1",
        "status": "1"
```

```
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "1"
        ]
    },
    "id": 1
}
```

Bedingungen ersetzen, aber die Auswertungsmethode beibehalten

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.update",
    "params": {
        "correlationid": "1",
        "filter": {
            "conditions": [
                {
                    "type": 3,
                    "oldtag": "error",
                    "newtag": "ok"
                }
            ]
        }
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "1"
        ]
    },
    "id": 1
}
```

Siehe auch

- Korrelationsfilter
- Korrelationsoperation

Quelle

CCorrelation::update() in *ui/include/classes/api/services/CCorrelation.php*.

LLD-Regel

Diese Klasse ist für die Arbeit mit Low-Level-Discovery-Regeln ausgelegt.

Objektreferenzen:

- LLD-Regel
  - HTTP-Header
  - HTTP-Abfragefeld

- LLD-Regelfilter
  - LLD-Regelfilterbedingung
- LLD-Makropfad
- LLD-Regel-Preprocessing
- LLD-Regel-Überschreibungen
  - LLD-Regel-Überschreibungsfilter
    - LLD-Regel-Überschreibungsfilterbedingung
  - LLD-Regel-Überschreibungsoperation
    - LLD-Regel-Überschreibungsoperation Status
    - LLD-Regel-Überschreibungsoperation Erkennung
    - LLD-Regel-Überschreibungsoperation Intervall
    - LLD-Regel-Überschreibungsoperation Verlauf
    - LLD-Regel-Überschreibungsoperation Trends
    - LLD-Regel-Überschreibungsoperation Schweregrad
    - LLD-Regel-Überschreibungsoperation Tag
    - LLD-Regel-Überschreibungsoperation Vorlage
    - LLD-Regel-Überschreibungsoperation Inventar

Verfügbare Methoden:

- discoveryrule.create - neue LLD-Regeln erstellen
- discoveryrule.delete - LLD-Regeln löschen
- discoveryrule.get - LLD-Regeln abrufen
- discoveryrule.update - LLD-Regeln aktualisieren

LLD-Regelobjekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `discoveryrule` API.

LLD-Regel

Das LLD-Regelobjekt verfügt über die folgenden Eigenschaften.

Property	Type	Description
itemid	ID	ID der LLD-Regel. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
delay	string	Aktualisierungsintervall der LLD-Regel. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300), einen Wert mit einem Zeitsuffix (z. B. 30s, 1m, 2h, 1d) oder optional ein oder mehrere benutzerdefinierte Intervalle, die durch Semikolons getrennt sind. Benutzerdefinierte Intervalle können eine Mischung aus flexiblen Intervallen und Zeitplanintervallen sein. Akzeptiert Benutzermakros. Bei Verwendung muss der Wert aus einem einzelnen Makro bestehen. Mehrere Makros oder Makros in Kombination mit Text werden nicht unterstützt. Flexible Intervalle können als zwei durch einen Schrägstrich getrennte Makros angegeben werden (z. B. `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`). Beispiel: `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „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) oder „Browser“ (22) gesetzt ist oder wenn `type` auf „Zabbix agent (active)“ (7) gesetzt ist und `key_` nicht „mqtt.get“ enthält

Property	Type	Description
hostid	ID	ID des Hosts, zu dem die LLD-Regel gehört. Verhalten der Eigenschaft: - *konstant* - *erforderlich* für Erstellungsvorgänge
flags	integer	Herkunft der Discovery-Regel. Mögliche Werte: 1 - eine LLD-Regel; 5 - eine aus einem Prototyp konvertierte LLD-Regel. Verhalten der Eigenschaft: - *schreibgeschützt*
interfaceid	ID	ID der Host-Schnittstelle der LLD-Regel. Verhalten der Eigenschaft: - *erforderlich*, wenn die LLD-Regel zu einem Host gehört und `type` auf „Zabbix agent“, „IPMI agent“, „JMX agent“ oder „SNMP agent“ gesetzt ist - *unterstützt*, wenn die LLD-Regel zu einem Host gehört und `type` auf „Simple check“, „External check“, „SSH agent“, „TELNET agent“ oder „HTTP agent“ gesetzt ist
key_	string	Schlüssel der LLD-Regel. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge - *schreibgeschützt* für geerbte Objekte
name	string	Name der LLD-Regel. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge - *schreibgeschützt* für geerbte Objekte
type	integer	Typ der LLD-Regel. Mögliche Werte: 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 - Abhängiger Datenpunkt; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser; 23 - Verschachtelt. Dieser Typ ist nur zulässig, wenn das übergeordnete Element der Regel eine Vorlage ist (von der angenommen wird, dass sie bei der Discovery mit einem Host-Prototyp verknüpft wird) oder ein entdeckter Host (der eine LLD-Regel an seiner Wurzel besitzt). Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge - *schreibgeschützt* für geerbte Objekte

Property	Type	Description
url	string	URL-Zeichenfolge. Unterstützt Benutzermakros, {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
allow_traps	integer	Ermöglicht das Befüllen des Werts analog zum Trapper-Datenpunkt. Mögliche Werte: 0 - *(Standard)* Keine eingehenden Daten akzeptieren; 1 - Eingehende Daten akzeptieren. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist
authtype	integer	Authentifizierungsmethode. Mögliche Werte, wenn `type` auf „SSH agent“ gesetzt ist: 0 - *(Standard)* Passwort; 1 - öffentlicher Schlüssel. Mögliche Werte, wenn `type` auf „HTTP agent“ gesetzt ist: 0 - *(Standard)* keine; 1 - Basic; 2 - NTLM; 3 - Kerberos; 4 - Digest. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „SSH agent“ oder „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte (wenn `type` auf „HTTP agent“ gesetzt ist)
description	string	Beschreibung der LLD-Regel.
error	string	Fehlertext, wenn beim Aktualisieren des Werts der LLD-Regel Probleme auftreten. Verhalten der Eigenschaft: - *schreibgeschützt*
follow_redirects	integer	Antwortweiterleitungen beim Abrufen von Daten verfolgen. Mögliche Werte: 0 - Weiterleitungen nicht verfolgen; 1 - *(Standard)* Weiterleitungen verfolgen. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
headers	array	Array von Headern, die bei einer HTTP-Anfrage gesendet werden. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
http_proxy	string	HTTP(S)-Proxy-Verbindungszeichenfolge. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte

Property	Type	Description
ipmi_sensor	string	IPMI-Sensor. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „IPMI agent“ gesetzt ist und `key_` nicht auf „ipmi.get“ gesetzt ist - *unterstützt*, wenn `type` auf „IPMI agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
jmx_endpoint	string	Benutzerdefinierte Verbindungszeichenfolge des JMX agent. Standard: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „JMX agent“ gesetzt ist
lifetime	string	Zeitraum, nach dem nicht mehr entdeckte Datenpunkte gelöscht werden. Akzeptiert eine Anzahl von Sekunden (z. B. 3600), einen Wert mit einem Zeitsuffix (z. B. `3600s`, `180m`, `24h`, `1d`) oder ein Benutzermakro. Standard: `7d`.
lifetime_type	integer	Szenario zum Löschen verlorener LLD-Ressourcen. Mögliche Werte: 0 - *(Standard)* Nach Erreichen des Lebensdauergrenzwerts löschen; 1 - Nicht löschen; 2 - Sofort löschen.
enabled_lifetime	string	Zeitraum, nach dem nicht mehr entdeckte Datenpunkte deaktiviert werden. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300), einen Wert mit einem Zeitsuffix (z. B. `30s`, `1m`, `2h`, `1d`) oder ein Benutzermakro. Standard: `0`.
enabled_lifetime_type	integer	Szenario zum Deaktivieren verlorener LLD-Ressourcen. Mögliche Werte: 0 - Nach Erreichen des Lebensdauergrenzwerts deaktivieren; 1 - Nicht deaktivieren; 2 - *(Standard)* Sofort deaktivieren.
master_itemid	ID	ID des Master-Datenpunkts. Eine Discovery-Regel kann kein Master-Datenpunkt für eine andere Discovery-Regel sein. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „Dependent item“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
output_format	integer	Gibt an, ob die Antwort in JSON konvertiert werden soll. Mögliche Werte: 0 - *(Standard)* Rohdaten speichern; 1 - In JSON konvertieren. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte

Property	Type	Description
params	string	Zusätzliche Parameter abhängig vom Typ der LLD-Regel: - ausgeführtes Skript für SSH- und Telnet-LLD-Regeln; - SQL-Abfrage für LLD-Regeln des Datenbankmonitors; - Formel für berechnete LLD-Regeln; - Skript für Skript- und Browser-LLD-Regeln. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „Database monitor“, „SSH agent“, „TELNET agent“, „Script“ oder „Browser“ gesetzt ist - *schreibgeschützt* für geerbte Objekte (wenn `type` auf „Script“ oder „Browser“ gesetzt ist)
parameters	object/array	Zusätzliche Parameter, wenn `type` auf „Script“ oder „Browser“ gesetzt ist. Array von Objekten mit den Eigenschaften `name` und `value`, wobei `name` eindeutig sein muss. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „Script“ oder „Browser“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
password	string	Passwort für die Authentifizierung. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „JMX agent“ gesetzt ist und `username` angegeben wurde - *unterstützt*, wenn `type` auf „Simple check“, „Database monitor“, „SSH agent“, „TELNET agent“ oder „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte (wenn `type` auf „HTTP agent“ gesetzt ist)
post_type	integer	Typ des in der Eigenschaft `posts` gespeicherten POST-Datenkörpers. Mögliche Werte: 0 - *(Standard)* Rohdaten; 2 - JSON-Daten; 3 - XML-Daten. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
posts	string	Daten des HTTP(S)-Anforderungskörpers. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „HTTP agent“ und `post_type` auf „JSON data“ oder „XML data“ gesetzt ist - *unterstützt*, wenn `type` auf „HTTP agent“ und `post_type` auf „Raw data“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
privatekey	string	Name der Datei mit dem privaten Schlüssel. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „SSH agent“ und `authtype` auf „public key“ gesetzt ist
publickey	string	Name der Datei mit dem öffentlichen Schlüssel. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „SSH agent“ und `authtype` auf „public key“ gesetzt ist
query_fields	array	Array von Abfragefeldern, die bei einer HTTP-Anfrage gesendet werden. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte

Property	Type	Description
request_method	integer	Typ der Anforderungsmethode. Mögliche Werte: 0 - *(Standard)* GET; 1 - POST; 2 - PUT; 3 - HEAD. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
retrieve_mode	integer	Welcher Teil der Antwort gespeichert werden soll. Mögliche Werte, wenn `request_method` auf „GET“, „POST“ oder „PUT“ gesetzt ist: 0 - *(Standard)* Body; 1 - Header; 2 - Body und Header werden gespeichert. Mögliche Werte, wenn `request_method` auf „HEAD“ gesetzt ist: 1 - Header. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
snmp_oid	string	SNMP-OID. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „SNMP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
ssl_cert_file	string	Dateipfad des öffentlichen SSL-Schlüssels. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
ssl_key_file	string	Dateipfad des privaten SSL-Schlüssels. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
ssl_key_password	string	Passwort für die SSL-Schlüsseldatei. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
state	integer	Status der LLD-Regel. Mögliche Werte: 0 - *(Standard)* normal; 1 - nicht unterstützt. Verhalten der Eigenschaft: - *schreibgeschützt*
status	integer	Status der LLD-Regel. Mögliche Werte: 0 - *(Standard)* aktivierte LLD-Regel; 1 - deaktivierte LLD-Regel.

Property	Type	Description
status_codes	string	Durch Kommas getrennte Bereiche erforderlicher HTTP-Statuscodes. Unterstützt außerdem Benutzermakros als Teil einer durch Kommas getrennten Liste. Beispiel: 200,200-{$M},{$M},200-400 Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
templateid	ID	ID der übergeordneten LLD-Regel der Vorlage. Verhalten der Eigenschaft: - *schreibgeschützt*
timeout	string	Zeitüberschreitung für die Abfrage von Datenpunkten. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300) oder einen Wert mit einem Zeitsuffix (z. B. 30s, 1m). Akzeptiert außerdem Benutzermakros. Möglicher Wertebereich: 1-600s. Standard: ”” - Proxy-/globale Einstellungen verwenden. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „Zabbix agent“ (0), „Simple check“ (3) gesetzt ist und `key_` nicht mit „`vmware.`“ oder „`icmpping`“ beginnt, „Zabbix agent (active)“ (7), „External check“ (10), „Database monitor“ (11), „SSH agent“ (13), „TELNET agent“ (14), „HTTP agent“ (19), „SNMP agent“ (20) gesetzt ist und `snmp_oid` mit „`walk[`“ oder „`get[`“ beginnt, „Script“ (21) oder „Browser“ (22) - *schreibgeschützt* für geerbte Objekte, - wenn der Standardwert angewendet wird und der Datenpunkt zu einem von einem nicht erreichbaren Proxy oder einer nicht erreichbaren Proxy-Gruppe überwachten Host gehört, wird die globale Zeitüberschreitung verwendet.
trapper_hosts	string	Zulässige Hosts. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „Zabbix trapper“ gesetzt ist oder wenn `type` auf „HTTP agent“ und `allow_traps` auf „Allow incoming data“ gesetzt ist. Wenn die Eigenschaft nicht angegeben wird, wird sie automatisch auf das Benutzermakro `{$TRAPPER.ALLOWED_HOSTS}` (Standardwert) gesetzt. Standardmäßig ist `{$TRAPPER.ALLOWED_HOSTS}` `127.0.0.1,::1`. Wenn die Eigenschaft auf eine leere Zeichenfolge gesetzt ist, werden eingehende Verbindungen von allen Hosts abgelehnt.
username	string	Benutzername für die Authentifizierung. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „SSH agent“ oder „TELNET agent“ gesetzt ist oder wenn `type` auf „JMX agent“ und `password` gesetzt ist - *unterstützt*, wenn `type` auf „Simple check“, „Database monitor“ oder „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte (wenn `type` auf „HTTP agent“ gesetzt ist)
uuid	string	Universell eindeutiger Bezeichner, der zum Verknüpfen importierter LLD-Regeln mit bereits vorhandenen Regeln verwendet wird. Wird automatisch generiert, wenn er nicht angegeben ist. Verhalten der Eigenschaft: - *unterstützt*, wenn die LLD-Regel zu einer Vorlage gehört

Property	Type	Description
verify_host	integer	Gibt an, ob überprüft werden soll, dass der Hostname für die Verbindung mit dem Hostnamen im Zertifikat des Hosts übereinstimmt. Mögliche Werte: 0 - *(Standard)* Nicht überprüfen; 1 - Überprüfen. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
verify_peer	integer	Gibt an, ob überprüft werden soll, dass das Zertifikat des Hosts authentisch ist. Mögliche Werte: 0 - *(Standard)* Nicht überprüfen; 1 - Überprüfen. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte

HTTP-Header

Das Header-Objekt hat die folgenden Eigenschaften:

Eigenschaft	Typ	Beschreibung
name	string	Name des HTTP-Headers. Eigenschaftsverhalten: - *erforderlich*
value	string	Wert des Headers. Eigenschaftsverhalten: - *erforderlich*

HTTP-Abfragefeld

Das Abfragefeldobjekt definiert einen Namen und einen Wert, die zur Angabe eines URL-Parameters verwendet werden. Es hat die folgenden Eigenschaften:

Eigenschaft	Typ	Beschreibung
name	string	Name des Parameters. Eigenschaftsverhalten: - *erforderlich*
value	string	Wert des Parameters. Eigenschaftsverhalten: - *erforderlich*

LLD-Regelfilter

Das LLD-Regelfilterobjekt definiert eine Reihe von Bedingungen, die zum Filtern erkannter Objekte verwendet werden können. Es hat die folgenden Eigenschaften:

Eigenschaft	Typ	Beschreibung
conditions	object/array	Menge von Filterbedingungen, die zum Filtern der Ergebnisse verwendet werden. Die Bedingungen werden in der Reihenfolge ihrer Platzierung in der Formel sortiert. Eigenschaftsverhalten: - *erforderlich*
evaltype	integer	Auswertungsmethode der Filterbedingung. Mögliche Werte: 0 - Und/Oder; 1 - Und; 2 - Oder; 3 - Benutzerdefinierter Ausdruck. Eigenschaftsverhalten: - *erforderlich*
eval_formula	string	Generierter Ausdruck, der zur Auswertung von Filterbedingungen verwendet wird. Der Ausdruck enthält IDs, die über `formulaid` auf bestimmte Filterbedingungen verweisen. Der Wert von `eval_formula` entspricht dem Wert von `formula` bei Filtern mit einem benutzerdefinierten Ausdruck. Eigenschaftsverhalten: - *schreibgeschützt*
formula	string	Benutzerdefinierter Ausdruck zur Auswertung von Bedingungen von Filtern mit einem benutzerdefinierten Ausdruck. Der Ausdruck muss IDs enthalten, die über `formulaid` auf bestimmte Filterbedingungen verweisen. Die im Ausdruck verwendeten IDs müssen exakt mit den in den Filterbedingungen definierten IDs übereinstimmen: Keine Bedingung darf ungenutzt bleiben oder ausgelassen werden. Eigenschaftsverhalten: - *erforderlich*, wenn `evaltype` auf „custom expression“ gesetzt ist

Filterbedingung für LLD-Regeln

Das Objekt für die Filterbedingung einer LLD-Regel definiert eine separate Prüfung, die für den Wert eines LLD-Makros durchgeführt wird. Es hat die folgenden Eigenschaften:

Eigenschaft	Type	Beschreibung
macro	string	LLD-Makro, für das die Prüfung durchgeführt wird. Property behavior: - *required*
value	string	Wert, mit dem verglichen werden soll. Property behavior: - *required* if `operator` is set to ”matches regular expression” or ”does not match regular expression”
formulaid	string	Beliebige eindeutige ID, die verwendet wird, um aus einem benutzerdefinierten Ausdruck auf die Bedingung zu verweisen. Darf nur Großbuchstaben enthalten. Die ID muss vom Benutzer definiert werden, wenn Filterbedingungen geändert werden, wird jedoch bei einer späteren Abfrage erneut generiert. Property behavior: - *required* if `evaltype` of LLD rule filter object is set to ”custom expression”

Eigenschaft	Type	Beschreibung
operator	integer	Bedingungsoperator. Mögliche Werte: 8 - *(Standard)* entspricht regulärem Ausdruck; 9 - entspricht nicht regulärem Ausdruck; 12 - existiert; 13 - existiert nicht.

Note:
Um besser zu verstehen, wie Filter mit verschiedenen Ausdruckstypen verwendet werden, siehe die Beispiele auf den Methodenseiten discoveryrule.get und discoveryrule.create.

LLD-Makropfad

Der LLD-Makropfad hat die folgenden Eigenschaften:

Eigenschaft	Typ	Beschreibung
lld_macro	string	LLD-Makro. Verhalten der Eigenschaft: - *erforderlich*
path	string	Selektor für den Wert, der dem entsprechenden Makro zugewiesen wird. Verhalten der Eigenschaft: - *erforderlich*

Vorverarbeitung der LLD-Regel

Das Vorverarbeitungsobjekt der LLD-Regel verfügt über die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
type	integer	Der Typ der Vorverarbeitungsoption. Mögliche Werte: 5 - Regulärer Ausdruck; 11 - XML XPath; 12 - JSONPath; 14 - Entspricht regulärem Ausdruck; 15 - Entspricht nicht regulärem Ausdruck; 16 - Auf Fehler in JSON prüfen; 17 - Auf Fehler in XML prüfen; 20 - Unveränderte Werte mit Heartbeat verwerfen; 21 - JavaScript; 23 - Prometheus in JSON; 24 - CSV in JSON; 25 - Ersetzen; 27 - XML in JSON; 28 - SNMP-Walk-Wert; 29 - SNMP-Walk in JSON; 30 - SNMP-Get-Wert. Verhalten der Eigenschaft: - *erforderlich*

Eigenschaft	Typ	Beschreibung
params	string	Zusätzliche Parameter, die von der Vorverarbeitungsoption verwendet werden. Mehrere Parameter werden durch das Newline-Zeichen (\n) getrennt. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „Regulärer Ausdruck“ (5), „XML XPath“ (11), „JSONPath“ (12), „Entspricht regulärem Ausdruck“ (14), „Entspricht nicht regulärem Ausdruck“ (15), „Auf Fehler in JSON prüfen“ (16), „Auf Fehler in XML prüfen“ (17), „Unveränderte Werte mit Heartbeat verwerfen“ (20), „JavaScript“ (21), „Prometheus in JSON“ (23), „CSV in JSON“ (24), „Ersetzen“ (25), „SNMP-Walk-Wert“ (28), „SNMP-Walk in JSON“ (29) oder „SNMP-Get-Wert“ (30) gesetzt ist
error_handler	integer	Aktionstyp, der bei einem Fehler in einem Vorverarbeitungsschritt verwendet wird. Mögliche Werte: 0 - Fehlermeldung wird vom Zabbix Server festgelegt; 1 - Wert verwerfen; 2 - Benutzerdefinierten Wert festlegen; 3 - Benutzerdefinierte Fehlermeldung festlegen.
error_handler_params	string	Parameter des Fehlerhandlers. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „Regulärer Ausdruck“ (5), „XML XPath“ (11), „JSONPath“ (12), „Entspricht regulärem Ausdruck“ (14), „Entspricht nicht regulärem Ausdruck“ (15), „Auf Fehler in JSON prüfen“ (16), „Auf Fehler in XML prüfen“ (17), „Prometheus in JSON“ (23), „CSV in JSON“ (24), „XML in JSON“ (27), „SNMP-Walk-Wert“ (28), „SNMP-Walk in JSON“ (29) oder „SNMP-Get-Wert“ (30) gesetzt ist Verhalten der Eigenschaft: - *erforderlich*, wenn `error_handler` auf „Benutzerdefinierten Wert festlegen“ oder „Benutzerdefinierte Fehlermeldung festlegen“ gesetzt ist

Die folgenden Parameter und Fehlerhandler werden für jeden Vorverarbeitungstyp unterstützt.

Vorverarbeitungstyp	Name	Parameter 1	Parameter 2	Parameter 3	Unterstützte Fehlerhandler
5	Regulärer Ausdruck	**M**uster[1]	Ausgabe[2]		0, 1, 2, 3
11	XML XPath	Pfad[3]			0, 1, 2, 3
12	JSONPath	Pfad[3]			0, 1, 2, 3
14	Entspricht regulärem Ausdruck	**M**uster[1]			0, 1, 2, 3
15	Entspricht nicht regulärem Ausdruck	**M**uster[1]			0, 1, 2, 3
16	Auf Fehler in JSON prüfen	Pfad[3]			0, 1, 2, 3

Vorverarbeitungstyp	Name	Parameter 1	Parameter 2	Parameter 3	Unterstützte Fehlerhandler
17	Auf Fehler in XML prüfen	Pfad[3]			0, 1, 2, 3
20	Unveränderte Werte mit Heartbeat verwerfen	Sekunden^4, 5^			
21	JavaScript	Skript[2]			
23	Prometheus in JSON	Muster^5, 6^			0, 1, 2, 3
24	CSV in JSON	Zeichen[2]	Zeichen[2]	0,1	0, 1, 2, 3
25	Ersetzen	Suchzeichenfolge[2]	Ersetzung[2]		
27	XML in JSON				0, 1, 2, 3
28	SNMP-Walk-Wert	OID[2]	Format: 0 - Unverändert 1 - UTF-8 aus Hex-STRING 2 - MAC aus Hex-STRING 3 - Ganzzahl aus BITS		0, 1, 2, 3
29	SNMP-Walk in JSON[7]	Feldname[2]	OID-Präfix[2]	Format: 0 - Unverändert 1 - UTF-8 aus Hex-STRING 2 - MAC aus Hex-STRING 3 - Ganzzahl aus BITS	0, 1, 2, 3
30	SNMP-Get-Wert	Format: 1 - UTF-8 aus Hex-STRING 2 - MAC aus Hex-STRING 3 - Ganzzahl aus BITS			0, 1, 2, 3

[1] regulärer Ausdruck
 [2] Zeichenfolge
 [3] JSONPath oder XML XPath
 [4] positive Ganzzahl (mit Unterstützung von Zeitsuffixen, z. B. 30s, 1m, 2h, 1d)
 [5] Benutzermakro
 [6] Prometheus-Muster gemäß der Syntax: `<metric name>{<label name>="<label value>", ...} == <value>`. Jede Komponente des Prometheus-Musters (Metrik, Labelname, Labelwert und Metrikwert) kann ein Benutzermakro sein.
 [7] Unterstützt mehrere durch ein Newline-Zeichen getrennte Datensätze im Format „Feldname,OID-Präfix,Format“.

LLD-Regelüberschreibungen

Das Objekt für LLD-Regelüberschreibungen definiert eine Reihe von Regeln (Filter, Bedingungen und Operationen), die verwendet werden, um Eigenschaften verschiedener Prototyp-Objekte zu überschreiben. Es hat die folgenden Eigenschaften:

Eigenschaft	Typ	Beschreibung
name	string	Eindeutiger Name der Überschreibung. Verhalten der Eigenschaft: - *erforderlich*
step	integer	Eindeutige Reihenfolgenummer der Überschreibung. Verhalten der Eigenschaft: - *erforderlich*
stop	integer	Verarbeitung der nächsten Überschreibungen stoppen, wenn eine Übereinstimmung vorliegt. Mögliche Werte: 0 - *(Standard)* Verarbeitung der Überschreibungen nicht stoppen; 1 - Verarbeitung der Überschreibungen stoppen, wenn der Filter übereinstimmt.
filter	object	Überschreibungsfilter.
operations	object/array	Überschreibungsoperationen.

Überschreibungsfilter für LLD-Regeln

Das Objekt für den Überschreibungsfilter von LLD-Regeln definiert eine Reihe von Bedingungen, bei deren Übereinstimmung mit dem erkannten Objekt die Überschreibung angewendet wird. Es hat die folgenden Eigenschaften:

Eigenschaft	Typ	Beschreibung
conditions	object/array	Menge von Bedingungen für Überschreibungsfilter, die zum Abgleichen der erkannten Objekte verwendet werden. Die Bedingungen werden in der Reihenfolge ihrer Platzierung in der Formel sortiert. Verhalten der Eigenschaft: - *erforderlich*
evaltype	integer	Auswertungsmethode der Bedingungen des Überschreibungsfilters. Mögliche Werte: 0 - Und/Oder; 1 - Und; 2 - Oder; 3 - Benutzerdefinierter Ausdruck. Verhalten der Eigenschaft: - *erforderlich*
eval_formula	string	Generierter Ausdruck, der zur Auswertung der Bedingungen des Überschreibungsfilters verwendet wird. Der Ausdruck enthält IDs, die über ihr `formulaid` auf bestimmte Bedingungen des Überschreibungsfilters verweisen. Der Wert von `eval_formula` entspricht dem Wert von `formula` bei Filtern mit einem benutzerdefinierten Ausdruck. Verhalten der Eigenschaft: - *schreibgeschützt*
formula	string	Benutzerdefinierter Ausdruck, der zur Auswertung der Bedingungen von Überschreibungsfiltern mit einem benutzerdefinierten Ausdruck verwendet wird. Der Ausdruck muss IDs enthalten, die über ihr `formulaid` auf bestimmte Bedingungen des Überschreibungsfilters verweisen. Die im Ausdruck verwendeten IDs müssen exakt mit den in den Bedingungen des Überschreibungsfilters definierten IDs übereinstimmen: Keine Bedingung darf ungenutzt bleiben oder ausgelassen werden. Verhalten der Eigenschaft: - *erforderlich*, wenn `evaltype` auf „custom expression“ gesetzt ist

Filterbedingung für LLD-Regel-Overrides

Das Objekt für die Filterbedingung von LLD-Regel-Overrides definiert eine separate Prüfung, die für den Wert eines LLD-Makros durchgeführt wird. Es hat die folgenden Eigenschaften:

Eigenschaft	Type	Beschreibung
macro	string	LLD-Makro, für das die Prüfung durchgeführt werden soll. Property behavior: - *required*
value	string	Wert, mit dem verglichen werden soll. Property behavior: - *required* if `operator` is set to ”matches regular expression” or ”does not match regular expression”
formulaid	string	Beliebige eindeutige ID, die verwendet wird, um aus einem benutzerdefinierten Ausdruck auf die Bedingung zu verweisen. Darf nur Großbuchstaben enthalten. Die ID muss vom Benutzer definiert werden, wenn Filterbedingungen geändert werden, wird jedoch bei einer späteren Abfrage erneut generiert. Property behavior: - *required* if `evaltype` of LLD rule override filter object is set to ”custom expression”
operator	integer	Bedingungsoperator. Mögliche Werte: 8 - *(Standard)* entspricht regulärem Ausdruck; 9 - entspricht nicht regulärem Ausdruck; 12 - existiert; 13 - existiert nicht.

LLD-Regel-Override-Operation

Die LLD-Regel-Override-Operation ist eine Kombination aus Bedingungen und Aktionen, die auf dem Prototyp-Objekt ausgeführt werden. Sie hat die folgenden Eigenschaften:

Eigenschaft	Type	Beschreibung
operationobject	integer	Typ des erkannten Objekts, für das die Aktion ausgeführt werden soll. Mögliche Werte: 0 - Datenpunkt-Prototyp; 1 - Auslöser-Prototyp; 2 - Graph-Prototyp; 3 - Host-Prototyp; 4 - Discovery-Prototyp. Property behavior: - *erforderlich*
operator	integer	Override-Bedingungs-operator. Mögliche Werte: 0 - *(Standard)* gleich; 1 - ungleich; 2 - enthält; 3 - enthält nicht; 8 - entspricht; 9 - entspricht nicht.
value	string	Muster zum Abgleich mit dem Namen des Datenpunkt-, Auslöser-, Graph- oder Host-Prototyps, abhängig vom ausgewählten Objekt.
opstatus	object	Statusobjekt der Override-Operation für Datenpunkt-, Auslöser- und Host-Prototyp-Objekte.

Eigenschaft	Type	Beschreibung
opdiscover	object	Objekt für den Discovery-Status der Override-Operation (alle Objekttypen).
opperiod	object	Objekt für den Zeitraum (Aktualisierungsintervall) der Override-Operation für das Datenpunkt-Prototyp-Objekt.
ophistory	object	Verlaufsobjekt der Override-Operation für das Datenpunkt-Prototyp-Objekt.
optrends	object	Trendobjekt der Override-Operation für das Datenpunkt-Prototyp-Objekt.
opseverity	object	Schweregradobjekt der Override-Operation für das Auslöser-Prototyp-Objekt.
optag	object/array	Tag-Objekt der Override-Operation für Auslöser- und Host-Prototyp-Objekte.
optemplate	object/array	Vorlagenobjekt der Override-Operation für das Host-Prototyp-Objekt.
opinventory	object	Inventarobjekt der Override-Operation für das Host-Prototyp-Objekt.

Status der LLD-Regel-Override-Operation

Status der LLD-Regel-Override-Operation, der für das erkannte Objekt festgelegt wird. Er hat die folgenden Eigenschaften:

Eigenschaft	Type	Beschreibung
status	integer	Überschreibt den Status für das ausgewählte Objekt. Mögliche Werte: 0 - Aktiviert erstellen; 1 - Deaktiviert erstellen. Property behavior: - *required*

LLD-Regel-Override-Operation „Discover“

Der Discover-Status der LLD-Regel-Override-Operation wird für das erkannte Objekt festgelegt. Sie hat die folgenden Eigenschaften:

Eigenschaft	Type	Beschreibung
discover	integer	Überschreibt den Discover-Status für das ausgewählte Objekt. Mögliche Werte: 0 - Ja, mit der Erkennung der Objekte fortfahren; 1 - Nein, neue Objekte werden nicht erkannt und vorhandene werden als verloren markiert. Property behavior: - *required*

Betriebszeitraum der LLD-Regelüberschreibung

Der Betriebszeitraum der LLD-Regelüberschreibung ist ein Aktualisierungsintervallwert, der für den ermittelten Datenpunkt festgelegt wird. Er verfügt über die folgenden Eigenschaften:

Property	Type	Description
delay	string	Überschreibt das Aktualisierungsintervall des Datenpunkt-Prototyps. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300), einen Wert mit einem Zeitsuffix (z. B. 30s, 1m, 2h, 1d) oder optional ein oder mehrere benutzerdefinierte Intervalle, die durch Semikolons getrennt sind. Benutzerdefinierte Intervalle können eine Mischung aus flexiblen und Zeitplanintervallen sein. Akzeptiert Benutzermakros oder LLD-Makros. Bei Verwendung muss der Wert aus einem einzelnen Makro bestehen. Mehrere Makros oder Makros in Kombination mit Text werden nicht unterstützt. Flexible Intervalle können als zwei durch einen Schrägstrich getrennte Makros geschrieben werden (z. B. {$FLEX_INTERVAL}/{$FLEX_PERIOD}). Beispiel: 1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr Verhalten der Eigenschaft: - *required*

Verlauf der Überschreibungsoperation der LLD-Regel

Wert der Verlaufsüberschreibungsoperation der LLD-Regel, der für den erkannten Datenpunkt festgelegt wird. Er hat die folgenden Eigenschaften:

Eigenschaft	Type	Beschreibung
history	string	Überschreibt den Verlauf des Datenpunktprototyps; dies ist eine Zeiteinheit dafür, wie lange die Verlaufsdaten gespeichert werden sollen. Akzeptiert auch Benutzermakro und LLD-Makro. Property behavior: - *required*

Trends von LLD-Regel-Override-Operationen

Der Wert der Trends von LLD-Regel-Override-Operationen wird für den erkannten Datenpunkt festgelegt. Er hat die folgenden Eigenschaften:

Property	Type	Description
trends	string	Überschreibt die Trends des Datenpunktprototyps; dies ist eine Zeiteinheit dafür, wie lange die Trenddaten gespeichert werden sollen. Akzeptiert auch Benutzermakro und LLD-Makro. Property behavior: - *required*

Schweregrad der Überschreibungsoperation einer LLD-Regel

Der Schweregradwert der Überschreibungsoperation einer LLD-Regel, der für den erkannten Auslöser festgelegt wird. Er hat die folgenden Eigenschaften:

Eigenschaft	Type	Beschreibung
severity	integer	Überschreibt den Schweregrad des Auslöser-Prototyps. Mögliche Werte: 0 - *(Standard)* nicht klassifiziert; 1 - Information; 2 - Warnung; 3 - Durchschnitt; 4 - Hoch; 5 - Katastrophe. Verhalten der Eigenschaft: - *erforderlich*

LLD-Regel-Override-Operation-Tag

Das Objekt LLD-Regel-Override-Operation-Tag enthält den Tag-Namen und den Wert, die dem erkannten Objekt zugewiesen werden. Es hat die folgenden Eigenschaften:

Property	Type	Description
tag	string	Neuer Tag-Name. Akzeptiert auch User-Makros und LLD-Makros. Property behavior: - *required*
value	string	Neuer Tag-Wert. Akzeptiert auch User-Makros und LLD-Makros.

Vorlage für die Überschreibungsoperation einer LLD-Regel

Objekt der Überschreibungsoperation einer LLD-Regel, das mit dem erkannten Host verknüpft ist.
Es hat die folgenden Eigenschaften:

Property	Type	Description
templateid	ID	Überschreibt die Vorlage der verknüpften Vorlagen des Host-Prototyps. Property behavior: - *required*

Inventar des LLD-Regel-Override-Vorgangs

Der Wert des Inventarmodus des LLD-Regel-Override-Vorgangs, der für den erkannten Host festgelegt wird. Er hat die folgenden Eigenschaften:

Eigenschaft	Type	Beschreibung
inventory_mode	integer	Überschreibt den Inventarmodus des Host-Prototyps. Mögliche Werte: -1 - deaktiviert; 0 - *(Standard)* manuell; 1 - automatisch. Property behavior: - *erforderlich*

discoveryrule.create

Beschreibung

```
object discoveryrule.create(object/array lldRules)
```

Diese Methode ermöglicht es Ihnen, neue LLD-Regeln zu erstellen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellende LLD-Regeln.

Zusätzlich zu den standardmäßigen LLD-Regel-Eigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Typ	Beschreibung
filter	object	LLD-Regelfilter für die LLD-Regel.
preprocessing	object/array	Optionen für die LLD-Regel-Vorverarbeitung.
lld_macro_paths	object/array	Optionen für lld_macro_path der LLD-Regel.
overrides	object/array	Optionen für LLD-Regel-Überschreibungen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten LLD-Regeln unter der Eigenschaft `itemids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen LLD-Regeln.

Beispiele

Erstellen einer LLD-Regel

Erstellen Sie eine Zabbix-Agent-LLD-Regel, um eingehängte Dateisysteme zu erkennen. Erkannte Datenpunkte werden alle 30 Sekunden aktualisiert.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

Verwenden eines Filters

Erstellen Sie eine LLD-Regel mit einer Reihe von Bedingungen, nach denen die Ergebnisse gefiltert werden. Die Bedingungen werden mithilfe des logischen Operators „and“ gruppiert.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
```

```
    "params": {
        "name": "Gefilterte LLD-Regel",
        "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

Erstellen einer LLD-Regel mit Makropfaden

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "LLD-Regel mit LLD-Makropfaden",
        "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

Verwenden eines benutzerdefinierten Ausdrucksfilters

Erstellen Sie eine LLD-Regel mit einem Filter, der einen benutzerdefinierten Ausdruck zur Auswertung der Bedingungen verwendet. Die LLD-Regel darf nur Objekte erkennen, deren Makrowert "{#MACRO1}" sowohl dem regulären Ausdruck "regex1" als auch "regex2" entspricht und deren Wert von "{#MACRO2}" entweder "regex3" oder "regex4" entspricht. Die Formel-IDs "A", "B", "C" und "D" wurden willkürlich gewählt.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Gefilterte LLD-Regel",
        "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

Benutzerdefinierte Abfragefelder und Header verwenden

Erstellen Sie eine LLD-Regel mit benutzerdefinierten Abfragefeldern und Headern.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "itemids": [
            "28336"
        ]
    },
    "id": 1
}
```

Erstellen einer LLD-Regel mit Vorverarbeitung

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Discovery-Regel mit Vorverarbeitung",
        "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

Erstellen einer LLD-Regel mit Überschreibungen

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Datenbank-Host erkennen",
        "key_": "lld.with.overrides",
        "hostid": "10001",
        "type": 0,
        "delay": "60s",
        "interfaceid": "1155",
        "overrides": [
            {
                "name": "MySQL-Host erkennen",
                "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": "Datenbank-Host",
                    "opstatus": {
                        "status": "0"
                    },
                    "optemplate": [
                        {
                            "templateid": "10170"
                        }
                    ],
                    "optag": [
                        {
                            "tag": "database",
                            "value": "mysql"
                        }
                    ]
                }
            ]
        },
        {
            "name": "PostgreSQL-Host erkennen",
            "step": "2",
            "stop": "1",
            "filter": {
                "evaltype": "0",
                "conditions": [
                    {
                        "macro": "{#UNIT.NAME}",
                        "operator": "8",
                        "value": "^postgresql\\.service$"
                    }
                ]
            },
            "operations": [
                {
                    "operationobject": "3",
                    "operator": "2",
                    "value": "Datenbank-Host",
                    "opstatus": {
                        "status": "0"
                    },
                    "optemplate": [
                        {
                            "templateid": "10263"
                        }
```

```
                        ],
                        "optag": [
                            {
                                "tag": "database",
                                "value": "postgresql"
                            }
                        ]
                    }
                ]
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "30980"
        ]
    },
    "id": 1
}
```

Skript-LLD-Regel erstellen

Erstellen Sie eine einfache Datenerfassung mit einer Skript-LLD-Regel.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

LLD-Regel mit einem angegebenen Zeitraum für die Deaktivierung und ohne Löschung erstellen

Erstellen Sie eine LLD-Regel mit einem benutzerdefinierten Zeitraum für die Deaktivierung einer Entität, nachdem sie nicht mehr

erkannt wird, mit der Einstellung, dass sie niemals gelöscht wird.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "46864"
        ]
    },
    "id": 1
}
```

Siehe auch

- LLD-Regelfilter
- LLD-Makropfade
- LLD-Regelvorverarbeitung

Quelle

CDiscoveryRule::create() in *ui/include/classes/api/services/CDiscoveryRule.php*.

discoveryrule.delete

Beschreibung

`object discoveryrule.delete(array lldRuleIds)`

Mit dieser Methode können Sie LLD-Regeln löschen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.
 Der Benutzer kann keine LLD-Regel löschen, die Entitäten über einen nicht zugänglichen Proxy erkennt. Weitere Informationen finden Sie unter Konfiguration von Benutzergruppen (*Proxy-Zugriffsliste*).

Parameter

`(array)` IDs der zu löschenden LLD-Regeln.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten LLD-Regeln unter der Eigenschaft `ruleids` enthält.

Beispiele

Mehrere LLD-Regeln löschen

Löschen Sie zwei LLD-Regeln.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.delete",
    "params": [
        "27665",
        "27668"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "ruleids": [
            "27665",
            "27668"
        ]
    },
    "id": 1
}
```

Quelle

CDiscoveryRule::delete() in *ui/include/classes/api/services/CDiscoveryRule.php*.

discoveryrule.get

Beschreibung

`integer/array discoveryrule.get(object parameters)`

Die Methode ermöglicht es Ihnen, LLD-Regeln anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
itemids	ID/array	Gibt nur LLD-Regeln mit den angegebenen IDs zurück.
groupids	ID/array	Gibt nur LLD-Regeln zurück, die zu den Hosts aus den angegebenen Gruppen gehören.
hostids	ID/array	Gibt nur LLD-Regeln zurück, die zu den angegebenen Hosts gehören.
inherited	boolean	Wenn auf `true` gesetzt, werden nur von einer Vorlage geerbte LLD-Regeln zurückgegeben.
interfaceids	ID/array	Gibt nur LLD-Regeln zurück, die die angegebenen Host-Schnittstellen verwenden.
monitored	boolean	Wenn auf `true` gesetzt, werden nur aktivierte LLD-Regeln zurückgegeben, die zu überwachten Hosts gehören.
templated	boolean	Wenn auf `true` gesetzt, werden nur LLD-Regeln zurückgegeben, die zu Vorlagen gehören.
templateids	ID/array	Gibt nur LLD-Regeln zurück, die zu den angegebenen Vorlagen gehören.
selectDiscoveryData	query	Gibt eine `discoveryData`-Eigenschaft mit den Objektdaten der LLD-Regelerkennung zurück. Das LLD-Regelerkennungsobjekt verknüpft eine erkannte LLD-Regel mit einem LLD-Regelprototyp, aus dem sie erkannt wurde.

Parameter	Type	Beschreibung
selectDiscoveryRulePrototypes	query	Gibt eine Eigenschaft `discoveryRulePrototypes` mit LLD-Regelprototypen zurück, die zur LLD-Regel gehören. Unterstützt `count`.
selectFilter	query	Gibt eine Eigenschaft `filter` mit Daten des von der LLD-Regel verwendeten Filters zurück.
selectGraphs	query	Gibt eine Eigenschaft `graphs` mit Graphprototypen zurück, die zur LLD-Regel gehören. Unterstützt `count`.
selectHostPrototypes	query	Gibt eine Eigenschaft `hostPrototypes` mit Host-Prototypen zurück, die zur LLD-Regel gehören. Unterstützt `count`.
selectHosts	query	Gibt eine Eigenschaft `hosts` mit einem Array von Hosts zurück, zu denen die LLD-Regel gehört.
selectItems	query	Gibt eine Eigenschaft `items` mit Datenpunktprototypen zurück, die zur LLD-Regel gehören. Unterstützt `count`.
selectTriggers	query	Gibt eine Eigenschaft `triggers` mit Auslöserprototypen zurück, die zur LLD-Regel gehören. Unterstützt `count`.
selectLLDMacroPaths	query	Gibt eine Eigenschaft `lld_macro_paths` mit einer Liste von LLD-Makros und Pfaden zu Werten zurück, die jedem entsprechenden Makro zugewiesen sind.
selectPreprocessing	query	Gibt eine Eigenschaft `preprocessing` mit Vorverarbeitungsoptionen der LLD-Regel zurück.
selectOverrides	query	Gibt eine Eigenschaft `lld_rule_overrides` mit einer Liste von Überschreibungsfiltern, Bedingungen und Operationen zurück, die auf Prototypobjekte angewendet werden.
filter	object	Gibt nur Ergebnisse zurück, die exakt mit dem angegebenen Filter übereinstimmen. Akzeptiert ein Objekt, bei dem die Schlüssel Eigenschaftsnamen sind und die Werte entweder ein einzelner Wert oder ein Array von Werten sind, mit denen abgeglichen wird. Unterstützt keine Eigenschaften des Datentyps `text`.
limitSelects	integer	Unterstützt zusätzliche Eigenschaften: `host` - technischer Name des Hosts, zu dem die LLD-Regel gehört. Begrenzt die Anzahl der von Unterabfragen zurückgegebenen Datensätze. Gilt für die folgenden Unterabfragen: `selectItems`, `selectGraphs`, `selectTriggers`.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `itemid`, `name`, `key_`, `delay`, `type`, `status`.
countOutput	boolean	Diese Parameter werden in der Referenzkommentierung beschrieben.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	

Parameter	Type	Beschreibung
startSearch	boolean	

Rückgabewerte

`(integer/array)` Gibt entweder Folgendes zurück:

- ein Array von Objekten
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Abrufen von Discovery-Regeln von einem Host

Rufen Sie alle Discovery-Regeln für eine bestimmte Host-ID ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "output": "extend",
        "hostids": "10202"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "27425",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10202",
            "name": "Erkennung von Netzwerkschnittstellen",
            "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": "Erkennung von Netzwerkschnittstellen, wie in dem globalen regulären Ausdruck \
            "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": "Erkennung eingehängter Dateisysteme",
            "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": "Erkennung von Dateisystemen verschiedener Typen, wie in dem globalen regulären
            "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
}
```

Abrufen von Filterbedingungen

Rufen Sie den Namen der LLD-Regel „24681“ und ihre Filterbedingungen ab. Der Filter verwendet den Auswertungstyp „and“, daher ist die Eigenschaft `formula` leer und `eval_formula` wird automatisch generiert.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "output": ["name"],
        "selectFilter": "extend",
        "itemids": ["24681"]
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```

LLD-Regel per URL abrufen

Rufen Sie die LLD-Regel für einen Host anhand des Werts des Regel-URL-Feldes ab. Es wird nur eine exakte Übereinstimmung der für die LLD-Regel definierten URL-Zeichenfolge unterstützt.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "hostids": "10257",
        "filter": {
            "type": 19,
            "url": "http://127.0.0.1/discoverer.php"
        }
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```


LLD-Regel mit Überschreibungen abrufen

Rufen Sie eine LLD-Regel ab, die verschiedene Überschreibungseinstellungen hat.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "output": ["name"],
        "itemids": "30980",
        "selectOverrides": ["name", "step", "stop", "filter", "operations"]
    },
    "id": 1
}
```

Antwort:


```
{
    "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
}
```


Siehe auch

- Graph-Prototyp
- Host
- Datenpunkt-Prototyp
- LLD-Regelfilter
- Auslöser-Prototyp

Quelle

CDiscoveryRule::get() in *ui/include/classes/api/services/CDiscoveryRule.php*.

discoveryrule.update

Beschreibung

`object discoveryrule.update(object/array lldRules)`

Diese Methode ermöglicht es Ihnen, vorhandene LLD-Regeln zu aktualisieren.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu aktualisierende Eigenschaften der LLD-Regel.

Die Eigenschaft `itemid` muss für jede LLD-Regel definiert sein, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Zusätzlich zu den standardmäßigen Eigenschaften der LLD-Regel akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
filter	object	LLD-Regelfilter zum Ersetzen des vorhandenen Filters.

Parameter	Type	Beschreibung
preprocessing	object/array	Optionen für die LLD-Regelvorverarbeitung zum Ersetzen der vorhandenen Vorverarbeitungsoptionen. Parameterverhalten: - *schreibgeschützt* für vererbte Objekte
lld_macro_paths	object/array	Optionen für lld_macro_path der LLD-Regel zum Ersetzen der vorhandenen Optionen für lld_macro_path. Parameterverhalten: - *schreibgeschützt* für vererbte Objekte
overrides	object/array	Optionen für LLD-Regelüberschreibungen zum Ersetzen der vorhandenen Überschreibungsoptionen. Parameterverhalten: - *schreibgeschützt* für vererbte Objekte

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten LLD-Regeln unter der Eigenschaft `itemids` enthält.

Beispiele

Hinzufügen eines Filters zu einer LLD-Regel

Fügen Sie einen Filter hinzu, damit der Inhalt des Makros *{#FSTYPE}* dem regulären Ausdruck *@File systems for discovery* entspricht.

Request:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "22450",
        "filter": {
            "evaltype": 1,
            "conditions": [
                {
                    "macro": "{#FSTYPE}",
                    "value": "@File systems for discovery"
                }
            ]
        }
    },
    "id": 1
}
```

Response:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "22450"
        ]
    },
    "id": 1
}
```

Hinzufügen von LLD-Makropfaden

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
```

```
    "params": {
        "itemid": "22450",
        "lld_macro_paths": [
            {
                "lld_macro": "{#MACRO1}",
                "path": "$.json.path"
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "22450"
        ]
    },
    "id": 1
}
```

Trapping deaktivieren

Deaktivieren Sie das LLD-Trapping für die Discovery-Regel.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "28336",
        "allow_traps": 0
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28336"
        ]
    },
    "id": 1
}
```

Vorverarbeitungsoptionen der LLD-Regel aktualisieren

Aktualisieren Sie eine LLD-Regel mit der Vorverarbeitungsregel „JSONPath“.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

LLD-Regelskript aktualisieren

Aktualisieren Sie ein LLD-Regelskript mit einem anderen Skript und entfernen Sie nicht benötigte Parameter, die vom vorherigen Skript verwendet wurden.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "23865",
        "parameters": [],
        "script": "Zabbix.log(3, 'Log test');\nreturn 1;"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Lebensdauer der LLD-Regel aktualisieren

Aktualisieren Sie die LLD-Regel, um eine nicht mehr entdeckte Entität nach 12 Stunden zu deaktivieren und nach 7 Tagen zu löschen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "46864",
        "lifetime_type": 0,
        "lifetime": "7d",
        "enabled_lifetime_type": 0,
        "enabled_lifetime": "12h"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "46864"
        ]
    },
    "id": 1
}
```

Quelle

CDiscoveryRule::update() in *ui/include/classes/api/services/CDiscoveryRule.php*.

LLD-Regelprototyp

Diese Klasse ist für die Arbeit mit Low-Level-Discovery-Regelprototypen vorgesehen.

Objektreferenzen:

- LLD-Regelprototyp
  - HTTP-Header
  - HTTP-Abfragefeld
- LLD-Regelprototyp-Filter
  - LLD-Regelprototyp-Filterbedingung
- LLD-Prototyp-Makropfad
- LLD-Regelprototyp-Vorverarbeitung
- LLD-Regelprototyp-Überschreibungen
  - LLD-Regelprototyp-Überschreibungsfilter
    - LLD-Regelprototyp-Überschreibungsfilterbedingung
  - LLD-Regelprototyp-Überschreibungsoperation
    - LLD-Regelprototyp-Überschreibungsoperationsstatus
    - LLD-Regelprototyp-Überschreibungsoperationserkennung
    - LLD-Regelprototyp-Überschreibungsoperationsintervall
    - LLD-Regelprototyp-Überschreibungsoperationsverlauf
    - LLD-Regelprototyp-Überschreibungsoperationstrends
    - LLD-Regelprototyp-Überschreibungsoperationsschweregrad
    - LLD-Regelprototyp-Überschreibungsoperationstag
    - LLD-Regelprototyp-Überschreibungsoperationsvorlage
    - LLD-Regelprototyp-Überschreibungsoperationsinventar

Verfügbare Methoden:

- discoveryruleprototype.create - neue LLD-Regelprototypen erstellen
- discoveryruleprototype.delete - LLD-Regelprototypen löschen
- discoveryruleprototype.get - LLD-Regelprototypen abrufen
- discoveryruleprototype.update - LLD-Regelprototypen aktualisieren

LLD-Regelprototyp-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der API `discoveryruleprototype`.

Prototyp des LLD-Regel

Das Objekt des Prototyps einer Low-Level-Discovery-Regel verfügt über die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
itemid	ID	ID des Prototyps der LLD-Regel. Eigenschaftsverhalten: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge

Eigenschaft	Typ	Beschreibung
ruleid	ID	ID der übergeordneten LLD-Regel bzw. des übergeordneten LLD-Regelprototyps. Eigenschaftsverhalten: - *schreibgeschützt* - *erforderlich* für Erstellungsvorgänge
delay	string	Aktualisierungsintervall des Prototyps der LLD-Regel. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300), einen Wert mit einem Zeitsuffix (z. B. 30s, 1m, 2h, 1d) oder optional ein oder mehrere benutzerdefinierte Intervalle, die durch Semikolons getrennt sind. Benutzerdefinierte Intervalle können eine Kombination aus flexiblen Intervallen und Zeitplanintervallen sein. Akzeptiert Benutzermakros. Bei Verwendung muss der Wert aus einem einzelnen Makro bestehen. Mehrere Makros oder Makros in Kombination mit Text werden nicht unterstützt. Flexible Intervalle können als zwei durch einen Schrägstrich getrennte Makros angegeben werden (z. B. `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`). Beispiel: `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` Eigenschaftsverhalten: - *erforderlich*, wenn `type` auf „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) oder „Browser“ (22) gesetzt ist oder wenn `type` auf „Zabbix agent (active)“ (7) gesetzt ist und `key_` nicht „mqtt.get“ enthält
hostid	ID	ID des Hosts, zu dem der Prototyp der LLD-Regel gehört. Eigenschaftsverhalten: - *konstant* - *erforderlich* für Erstellungsvorgänge
flags	integer	Ursprung des Prototyps der LLD-Regel. Mögliche Werte: 3 - ein Prototyp einer Low-Level-Discovery-Regel; 7 - ein entdeckter Prototyp einer Low-Level-Discovery-Regel. Eigenschaftsverhalten: - *schreibgeschützt*
interfaceid	ID	ID der Host-Schnittstelle des Prototyps der LLD-Regel. Eigenschaftsverhalten: - *erforderlich*, wenn der Prototyp der LLD-Regel zu einem Host gehört und `type` auf „Zabbix agent“, „IPMI agent“, „JMX agent“ oder „SNMP agent“ gesetzt ist - *unterstützt*, wenn der Prototyp der LLD-Regel zu einem Host gehört und `type` auf „Simple check“, „External check“, „SSH agent“, „TELNET agent“ oder „HTTP agent“ gesetzt ist
key_	string	Schlüssel des Prototyps der LLD-Regel. Mindestens ein LLD-Makro ist erforderlich. Eigenschaftsverhalten: - *erforderlich* für Erstellungsvorgänge - *schreibgeschützt* für geerbte Objekte

Eigenschaft	Typ	Beschreibung
name	string	Name des Prototyps der LLD-Regel. Eigenschaftsverhalten: - *erforderlich* für Erstellungsvorgänge - *schreibgeschützt* für geerbte Objekte
type	integer	Typ des Prototyps der LLD-Regel. Mögliche Werte: 0 - Zabbix agent; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix agent (active); 10 - External check; 11 - Database monitor; 12 - IPMI agent; 13 - SSH agent; 14 - TELNET agent; 16 - JMX agent; 18 - Dependent item; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser; 23 - Nested. Eigenschaftsverhalten: - *erforderlich* für Erstellungsvorgänge - *schreibgeschützt* für geerbte Objekte
url	string	URL-Zeichenfolge. Unterstützt Benutzermakros, {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}. Eigenschaftsverhalten: - *erforderlich*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
allow_traps	integer	Ermöglicht das Befüllen des Werts ähnlich wie bei einem Trapper-Datenpunkt. Mögliche Werte: 0 - *(Standard)* Keine eingehenden Daten akzeptieren; 1 - Eingehende Daten akzeptieren. Eigenschaftsverhalten: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist
authtype	integer	Authentifizierungsmethode. Mögliche Werte, wenn `type` auf „SSH agent“ gesetzt ist: 0 - *(Standard)* Passwort; 1 - öffentlicher Schlüssel. Mögliche Werte, wenn `type` auf „HTTP agent“ gesetzt ist: 0 - *(Standard)* keine; 1 - Basic; 2 - NTLM. Eigenschaftsverhalten: - *unterstützt*, wenn `type` auf „SSH agent“ oder „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte (wenn `type` auf „HTTP agent“ gesetzt ist)
description	string	Beschreibung des Prototyps der LLD-Regel.

Eigenschaft	Typ	Beschreibung
error	string	Fehlertext, wenn beim Aktualisieren des Werts des Prototyps der LLD-Regel Probleme auftreten. Eigenschaftsverhalten: - *schreibgeschützt*
follow_redirects	integer	Beim Abrufen von Daten Weiterleitungen der Antwort folgen. Mögliche Werte: 0 - Weiterleitungen nicht folgen; 1 - *(Standard)* Weiterleitungen folgen. Eigenschaftsverhalten: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
headers	array	Array von Headern, die bei einer HTTP-Anfrage gesendet werden. Eigenschaftsverhalten: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
http_proxy	string	Verbindungszeichenfolge des HTTP(S)-Proxys. Eigenschaftsverhalten: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
ipmi_sensor	string	IPMI-Sensor. Eigenschaftsverhalten: - *erforderlich*, wenn `type` auf „IPMI agent“ gesetzt ist und `key_` nicht auf „ipmi.get“ gesetzt ist - *unterstützt*, wenn `type` auf „IPMI agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
jmx_endpoint	string	Benutzerdefinierte Verbindungszeichenfolge des JMX agent. Standard: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Eigenschaftsverhalten: - *unterstützt*, wenn `type` auf „JMX agent“ gesetzt ist
lifetime	string	Zeitraum, nach dem Datenpunkte gelöscht werden, die nicht mehr entdeckt werden. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300), einen Wert mit einem Zeitsuffix (z. B. `30s`, `1m`, `2h`, `1d`) oder ein Benutzermakro. Standard: `7d`.
lifetime_type	integer	Szenario zum Löschen verlorener LLD-Ressourcen. Mögliche Werte: 0 - *(Standard)* Nach Erreichen des Lebensdauergrenzwerts löschen; 1 - Nicht löschen; 2 - Sofort löschen.
enabled_lifetime	string	Zeitraum, nach dem Datenpunkte deaktiviert werden, die nicht mehr entdeckt werden. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300), einen Wert mit einem Zeitsuffix (z. B. `30s`, `1m`, `2h`, `1d`) oder ein Benutzermakro. Standard: `0`.
enabled_lifetime_type	integer	Szenario zum Deaktivieren verlorener LLD-Ressourcen. Mögliche Werte: 0 - Nach Erreichen des Lebensdauergrenzwerts deaktivieren; 1 - Nicht deaktivieren; 2 - *(Standard)* Sofort deaktivieren.

Eigenschaft	Typ	Beschreibung
master_itemid	ID	ID des Master-Datenpunkts. Eine Discovery-Regel kann kein Master-Datenpunkt für eine andere Discovery-Regel sein. Eigenschaftsverhalten: - *erforderlich*, wenn `type` auf „Dependent item“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
output_format	integer	Soll die Antwort in JSON konvertiert werden? Mögliche Werte: 0 - *(Standard)* Rohdaten speichern; 1 - In JSON konvertieren. Eigenschaftsverhalten: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
params	string	Zusätzliche Parameter abhängig vom Typ des Prototyps der LLD-Regel: - ausgeführtes Skript für SSH- und Telnet-LLD-Regeln; - SQL-Abfrage für LLD-Regeln des Datenbankmonitors; - Formel für berechnete LLD-Regeln; - Skript für Skript- und Browser-LLD-Regeln. Eigenschaftsverhalten: - *erforderlich*, wenn `type` auf „Database monitor“, „SSH agent“, „TELNET agent“, „Script“ oder „Browser“ gesetzt ist - *schreibgeschützt* für geerbte Objekte (wenn `type` auf „Script“ oder „Browser“ gesetzt ist)
parameters	object/array	Zusätzliche Parameter, wenn `type` auf „Script“ oder „Browser“ gesetzt ist. Array von Objekten mit den Eigenschaften `name` und `value`, wobei `name` eindeutig sein muss. Eigenschaftsverhalten: - *unterstützt*, wenn `type` auf „Script“ oder „Browser“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
password	string	Passwort für die Authentifizierung. Eigenschaftsverhalten: - *erforderlich*, wenn `type` auf „JMX agent“ gesetzt ist und `username` festgelegt ist - *unterstützt*, wenn `type` auf „Simple check“, „Database monitor“, „SSH agent“, „TELNET agent“ oder „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte (wenn `type` auf „HTTP agent“ gesetzt ist)
post_type	integer	Typ des im Attribut `posts` gespeicherten Textkörpers der POST-Anfrage. Mögliche Werte: 0 - *(Standard)* Rohdaten; 2 - JSON-Daten; 3 - XML-Daten. Eigenschaftsverhalten: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte

Eigenschaft	Typ	Beschreibung
posts	string	Textkörperdaten der HTTP(S)-Anfrage. Eigenschaftsverhalten: - *erforderlich*, wenn `type` auf „HTTP agent“ und `post_type` auf „JSON data“ oder „XML data“ gesetzt ist - *unterstützt*, wenn `type` auf „HTTP agent“ und `post_type` auf „Raw data“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
privatekey	string	Name der Datei mit dem privaten Schlüssel. Eigenschaftsverhalten: - *erforderlich*, wenn `type` auf „SSH agent“ und `authtype` auf „public key“ gesetzt ist
publickey	string	Name der Datei mit dem öffentlichen Schlüssel. Eigenschaftsverhalten: - *erforderlich*, wenn `type` auf „SSH agent“ und `authtype` auf „public key“ gesetzt ist
query_fields	array	Array von Abfragefeldern, die bei einer HTTP-Anfrage gesendet werden. Eigenschaftsverhalten: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
request_method	integer	Typ der Anfragemethode. Mögliche Werte: 0 - *(Standard)* GET; 1 - POST; 2 - PUT; 3 - HEAD. Eigenschaftsverhalten: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
retrieve_mode	integer	Welcher Teil der Antwort gespeichert werden soll. Mögliche Werte, wenn `request_method` auf „GET“, „POST“ oder „PUT“ gesetzt ist: 0 - *(Standard)* Textkörper; 1 - Header; 2 - Sowohl Textkörper als auch Header werden gespeichert. Mögliche Werte, wenn `request_method` auf „HEAD“ gesetzt ist: 1 - Header. Eigenschaftsverhalten: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
snmp_oid	string	SNMP-OID. Eigenschaftsverhalten: - *erforderlich*, wenn `type` auf „SNMP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
ssl_cert_file	string	Dateipfad des öffentlichen SSL-Schlüssels. Eigenschaftsverhalten: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte

Eigenschaft	Typ	Beschreibung
ssl_key_file	string	Dateipfad des privaten SSL-Schlüssels. Eigenschaftsverhalten: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
ssl_key_password	string	Passwort für die SSL-Schlüsseldatei. Eigenschaftsverhalten: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
state	integer	Status des Prototyps der LLD-Regel. Mögliche Werte: 0 - *(Standard)* normal; 1 - nicht unterstützt. Eigenschaftsverhalten: - *schreibgeschützt*
status	integer	Status des Prototyps der LLD-Regel. Mögliche Werte: 0 - *(Standard)* aktivierter Prototyp der LLD-Regel; 1 - deaktivierter Prototyp der LLD-Regel.
status_codes	string	Durch Kommas getrennte Bereiche erforderlicher HTTP-Statuscodes. Unterstützt auch Benutzermakros als Teil einer durch Kommas getrennten Liste. Beispiel: 200,200-{$M},{$M},200-400 Eigenschaftsverhalten: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
templateid	ID	ID des übergeordneten LLD-Regelprototyps der Vorlage. Eigenschaftsverhalten: - *schreibgeschützt*
timeout	string	Zeitüberschreitung für die Anfrage zum Abrufen von Datenpunkten. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300) oder einen Wert mit einem Zeitsuffix (z. B. 30s, 1m). Akzeptiert auch Benutzermakros. Möglicher Wertebereich: 1-600s. Standard: ”” - Proxy-/globale Einstellungen verwenden. Eigenschaftsverhalten: - *unterstützt*, wenn `type` auf „Zabbix agent“ (0), „Simple check“ (3) gesetzt ist und `key_` nicht mit „`vmware.`“ oder „`icmpping`“ beginnt, „Zabbix agent (active)“ (7), „External check“ (10), „Database monitor“ (11), „SSH agent“ (13), „TELNET agent“ (14), „HTTP agent“ (19), „SNMP agent“ (20) gesetzt ist und `snmp_oid` mit „`walk[`“ oder „`get[`“ beginnt, „Script“ (21) oder „Browser“ (22) - *schreibgeschützt* für geerbte Objekte

Eigenschaft	Typ	Beschreibung
trapper_hosts	string	Zulässige Hosts. Eigenschaftsverhalten: - *unterstützt*, wenn `type` auf „Zabbix trapper“ gesetzt ist oder wenn `type` auf „HTTP agent“ und `allow_traps` auf „Allow incoming data“ gesetzt ist. Wenn die Eigenschaft nicht angegeben wird, wird sie automatisch auf das Benutzermakro `{$TRAPPER.ALLOWED_HOSTS}` gesetzt (Standardwert). Standardmäßig ist `{$TRAPPER.ALLOWED_HOSTS}` auf `127.0.0.1,::1` gesetzt. Wenn die Eigenschaft auf eine leere Zeichenfolge gesetzt ist, werden eingehende Verbindungen von allen Hosts abgelehnt.
username	string	Benutzername für die Authentifizierung. Eigenschaftsverhalten: - *erforderlich*, wenn `type` auf „SSH agent“ oder „TELNET agent“ gesetzt ist oder wenn `type` auf „JMX agent“ und `password` gesetzt ist - *unterstützt*, wenn `type` auf „Simple check“, „Database monitor“ oder „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte (wenn `type` auf „HTTP agent“ gesetzt ist)
uuid	string	Universell eindeutiger Bezeichner, der zum Verknüpfen importierter Prototypen von LLD-Regeln mit bereits vorhandenen Prototypen verwendet wird. Wird automatisch generiert, wenn er nicht angegeben wird. Eigenschaftsverhalten: - *unterstützt*, wenn der Prototyp der LLD-Regel zu einer Vorlage gehört
verify_host	integer	Gibt an, ob überprüft werden soll, dass der Hostname für die Verbindung mit dem Hostnamen im Zertifikat des Hosts übereinstimmt. Mögliche Werte: 0 - *(Standard)* Nicht überprüfen; 1 - Überprüfen. Eigenschaftsverhalten: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte
verify_peer	integer	Gibt an, ob überprüft werden soll, dass das Zertifikat des Hosts authentisch ist. Mögliche Werte: 0 - *(Standard)* Nicht überprüfen; 1 - Überprüfen. Eigenschaftsverhalten: - *unterstützt*, wenn `type` auf „HTTP agent“ gesetzt ist - *schreibgeschützt* für geerbte Objekte

HTTP-Header

Das Header-Objekt hat die folgenden Eigenschaften:

Eigenschaft	Typ	Beschreibung
name	string	Name des HTTP-Headers. Verhalten der Eigenschaft: - *erforderlich*

Eigenschaft	Typ	Beschreibung
value	string	Wert des Headers. Verhalten der Eigenschaft: - *erforderlich*

HTTP-Abfragefeld

Das Abfragefeldobjekt definiert einen Namen und einen Wert, die zur Angabe eines URL-Parameters verwendet werden. Es hat die folgenden Eigenschaften:

Eigenschaft	Typ	Beschreibung
name	string	Name des Parameters. Eigenschaftsverhalten: - *erforderlich*
value	string	Wert des Parameters. Eigenschaftsverhalten: - *erforderlich*

Filter für LLD-Regelprototypen

Das Filterobjekt für LLD-Regelprototypen definiert eine Reihe von Bedingungen, die zum Filtern entdeckter Objekte verwendet werden können. Es hat die folgenden Eigenschaften:

Eigenschaft	Typ	Beschreibung
conditions	object/array	Menge von Filterbedingungen, die zum Filtern von Ergebnissen verwendet werden. Die Bedingungen werden in der Reihenfolge ihrer Platzierung in der Formel sortiert. Verhalten der Eigenschaft: - *erforderlich*
evaltype	integer	Auswertungsmethode der Filterbedingung. Mögliche Werte: 0 - Und/Oder; 1 - Und; 2 - Oder; 3 - Benutzerdefinierter Ausdruck. Verhalten der Eigenschaft: - *erforderlich*
eval_formula	string	Generierter Ausdruck, der zur Auswertung von Filterbedingungen verwendet wird. Der Ausdruck enthält IDs, die bestimmte Filterbedingungen über ihre `formulaid` referenzieren. Der Wert von `eval_formula` entspricht dem Wert von `formula` für Filter mit einem benutzerdefinierten Ausdruck. Verhalten der Eigenschaft: - *schreibgeschützt*

Eigenschaft	Typ	Beschreibung
formula	string	Benutzerdefinierter Ausdruck, der zur Auswertung von Bedingungen von Filtern mit einem benutzerdefinierten Ausdruck verwendet wird. Der Ausdruck muss IDs enthalten, die bestimmte Filterbedingungen über ihre `formulaid` referenzieren. Die im Ausdruck verwendeten IDs müssen exakt mit den in den Filterbedingungen definierten IDs übereinstimmen: Keine Bedingung darf ungenutzt bleiben oder ausgelassen werden. Verhalten der Eigenschaft: - *erforderlich*, wenn `evaltype` auf „custom expression“ gesetzt ist

Filterbedingung für LLD-Regelprototypen

Das Objekt für die Filterbedingung eines LLD-Regelprototyps definiert eine separate Prüfung, die für den Wert eines LLD-Makros durchgeführt wird. Es hat die folgenden Eigenschaften:

Eigenschaft	Typ	Beschreibung
macro	string	LLD-Makro, für das die Prüfung durchgeführt wird. Verhalten der Eigenschaft: - *erforderlich*
value	string	Wert, mit dem verglichen wird. Verhalten der Eigenschaft: - *erforderlich*, wenn `operator` auf „entspricht regulärem Ausdruck“ oder „entspricht nicht regulärem Ausdruck“ gesetzt ist
formulaid	string	Beliebige eindeutige ID, die verwendet wird, um aus einem benutzerdefinierten Ausdruck auf die Bedingung zu verweisen. Sie darf nur Großbuchstaben enthalten. Die ID muss vom Benutzer definiert werden, wenn Filterbedingungen geändert werden, wird jedoch bei einer späteren Abfrage erneut generiert. Verhalten der Eigenschaft: - *erforderlich*, wenn `evaltype` des Objekts Filter für LLD-Regelprototypen auf „benutzerdefinierter Ausdruck“ gesetzt ist
operator	integer	Bedingungsoperator. Mögliche Werte: 8 - *(Standard)* entspricht regulärem Ausdruck; 9 - entspricht nicht regulärem Ausdruck; 12 - existiert; 13 - existiert nicht.

Note:
Um besser zu verstehen, wie Filter mit verschiedenen Ausdruckstypen verwendet werden, siehe die Beispiele auf den Seiten der Methoden discoveryruleprototype.get und discoveryruleprototype.create.

LLD-Makropfad

Der LLD-Makropfad hat die folgenden Eigenschaften:

Eigenschaft	Type	Beschreibung
lld_macro	string	LLD-Makro. Verhalten der Eigenschaft: - *erforderlich*

Eigenschaft	Type	Beschreibung
path	string	Selektor für den Wert, der dem entsprechenden Makro zugewiesen wird. Verhalten der Eigenschaft: - *erforderlich*

Vorverarbeitung des LLD-Regelprototyps

Das Vorverarbeitungsobjekt des LLD-Regelprototyps verfügt über die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
type	integer	Der Typ der Vorverarbeitungsoption. Mögliche Werte: 5 - Regulärer Ausdruck; 11 - XML XPath; 12 - JSONPath; 14 - Entspricht regulärem Ausdruck; 15 - Entspricht keinem regulären Ausdruck; 16 - Auf Fehler in JSON prüfen; 17 - Auf Fehler in XML prüfen; 20 - Unveränderte Werte mit Heartbeat verwerfen; 21 - JavaScript; 23 - Prometheus in JSON; 24 - CSV in JSON; 25 - Ersetzen; 27 - XML in JSON; 28 - SNMP-Walk-Wert; 29 - SNMP-Walk in JSON; 30 - SNMP-Get-Wert. Verhalten der Eigenschaft: - *erforderlich*
params	string	Zusätzliche Parameter, die von der Vorverarbeitungsoption verwendet werden. Mehrere Parameter werden durch das Zeilenumbruchzeichen (\n) getrennt. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „Regulärer Ausdruck“ (5), „XML XPath“ (11), „JSONPath“ (12), „Entspricht regulärem Ausdruck“ (14), „Entspricht keinem regulären Ausdruck“ (15), „Auf Fehler in JSON prüfen“ (16), „Auf Fehler in XML prüfen“ (17), „Unveränderte Werte mit Heartbeat verwerfen“ (20), „JavaScript“ (21), „Prometheus in JSON“ (23), „CSV in JSON“ (24), „Ersetzen“ (25), „SNMP-Walk-Wert“ (28), „SNMP-Walk in JSON“ (29) oder „SNMP-Get-Wert“ (30) gesetzt ist

Eigenschaft	Typ	Beschreibung
error_handler	integer	Aktionstyp, der bei einem Fehler in einem Vorverarbeitungsschritt verwendet wird. Mögliche Werte: 0 - Fehlermeldung wird vom Zabbix Server festgelegt; 1 - Wert verwerfen; 2 - Benutzerdefinierten Wert festlegen; 3 - Benutzerdefinierte Fehlermeldung festlegen. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „Regulärer Ausdruck“ (5), „XML XPath“ (11), „JSONPath“ (12), „Entspricht regulärem Ausdruck“ (14), „Entspricht keinem regulären Ausdruck“ (15), „Auf Fehler in JSON prüfen“ (16), „Auf Fehler in XML prüfen“ (17), „Prometheus in JSON“ (23), „CSV in JSON“ (24), „XML in JSON“ (27), „SNMP-Walk-Wert“ (28), „SNMP-Walk in JSON“ (29) oder „SNMP-Get-Wert“ (30) gesetzt ist
error_handler_params	string	Parameter des Fehlerhandlers. Verhalten der Eigenschaft: - *erforderlich*, wenn `error_handler` auf „Benutzerdefinierten Wert festlegen“ oder „Benutzerdefinierte Fehlermeldung festlegen“ gesetzt ist

Die folgenden Parameter und Fehlerhandler werden für jeden Vorverarbeitungstyp unterstützt.

Vorverarbeitungstyp	Name	Parameter 1	Parameter 2	Parameter 3	Unterstützte Fehlerhandler
5	Regulärer Ausdruck	pattern[1]	output[2]		0, 1, 2, 3
11	XML XPath	path[3]			0, 1, 2, 3
12	JSONPath	path[3]			0, 1, 2, 3
14	Entspricht regulärem Ausdruck	pattern[1]			0, 1, 2, 3
15	Entspricht keinem regulären Ausdruck	pattern[1]			0, 1, 2, 3
16	Auf Fehler in JSON prüfen	path[3]			0, 1, 2, 3
17	Auf Fehler in XML prüfen	path[3]			0, 1, 2, 3

Vorverarbeitungstyp	Name	Parameter 1	Parameter 2	Parameter 3	Unterstützte Fehlerhandler
20	Unveränderte Werte mit Heartbeat verwerfen	seconds^4, 5^			
21	JavaScript	script[2]			
23	Prometheus in JSON	pattern^5, 6^			0, 1, 2, 3
24	CSV in JSON	character[2]	character[2]	0,1	0, 1, 2, 3
25	Ersetzen	search string[2]	replacement[2]		
27	XML in JSON				0, 1, 2, 3
28	SNMP-Walk-Wert	OID[2]	Format: 0 - Unverändert 1 - UTF-8 aus Hex-STRING 2 - MAC aus Hex-STRING 3 - Integer aus BITS		0, 1, 2, 3
29	SNMP-Walk in JSON[7]	Field name[2]	OID prefix[2]	Format: 0 - Unverändert 1 - UTF-8 aus Hex-STRING 2 - MAC aus Hex-STRING 3 - Integer aus BITS	0, 1, 2, 3
30	SNMP-Get-Wert	Format: 1 - UTF-8 aus Hex-STRING 2 - MAC aus Hex-STRING 3 - Integer aus BITS			0, 1, 2, 3

[1] regulärer Ausdruck
 [2] Zeichenfolge
 [3] JSONPath oder XML XPath
 [4] positive Ganzzahl (mit Unterstützung von Zeitsuffixen, z. B. 30s, 1m, 2h, 1d)
 [5] Benutzermakro
 [6] Prometheus-Muster gemäß der Syntax: `<metric name>{<label name>="<label value>", ...} == <value>`. Jede Komponente des Prometheus-Musters (Metrik, Labelname, Labelwert und Metrikwert) kann ein Benutzermakro sein.
 [7] Unterstützt mehrere durch ein Zeilenumbruchzeichen getrennte Datensätze im Format „Field name,OID prefix,Format“.

Überschreibungen von LLD-Regelprototypen

Das Objekt für Überschreibungen von LLD-Regelprototypen definiert eine Reihe von Regeln (Filter, Bedingungen und Operationen), die verwendet werden, um Eigenschaften verschiedener Prototyp-Objekte zu überschreiben. Es hat die folgenden Eigenschaften:

Property	Type	Description
name	string	Eindeutiger Name der Überschreibung. Property behavior: *- required*

Property	Type	Description
step	integer	Eindeutige Reihenfolgenummer der Überschreibung. Property behavior: - *required*
stop	integer	Verarbeitung der nächsten Überschreibungen stoppen, wenn eine Übereinstimmung vorliegt. Mögliche Werte: 0 - *(Standard)* Verarbeitung der Überschreibungen nicht stoppen; 1 - Verarbeitung der Überschreibungen stoppen, wenn der Filter übereinstimmt.
filter	object	Überschreibungsfilter.
operations	object/array	Überschreibungsoperationen.

Filter für Überschreibungen von LLD-Regelprototypen

Das Filterobjekt für Überschreibungen von LLD-Regelprototypen definiert eine Reihe von Bedingungen, bei deren Übereinstimmung mit dem erkannten Objekt die Überschreibung angewendet wird. Es hat die folgenden Eigenschaften:

Property	Type	Description
conditions	object/array	Menge von Bedingungen für Überschreibungsfilter, die zum Abgleichen der erkannten Objekte verwendet werden. Die Bedingungen werden in der Reihenfolge ihrer Platzierung in der Formel sortiert. Property behavior: - *required*
evaltype	integer	Auswertungsmethode der Bedingungen des Überschreibungsfilters. Mögliche Werte: 0 - Und/Oder; 1 - Und; 2 - Oder; 3 - Benutzerdefinierter Ausdruck. Property behavior: - *required*
eval_formula	string	Generierter Ausdruck, der zur Auswertung der Bedingungen des Überschreibungsfilters verwendet wird. Der Ausdruck enthält IDs, die über ihr `formulaid` auf bestimmte Bedingungen des Überschreibungsfilters verweisen. Der Wert von `eval_formula` entspricht dem Wert von `formula` bei Filtern mit einem benutzerdefinierten Ausdruck. Property behavior: - *read-only*
formula	string	Benutzerdefinierter Ausdruck zur Auswertung der Bedingungen von Überschreibungsfiltern mit einem benutzerdefinierten Ausdruck. Der Ausdruck muss IDs enthalten, die über ihr `formulaid` auf bestimmte Bedingungen des Überschreibungsfilters verweisen. Die im Ausdruck verwendeten IDs müssen exakt mit den in den Bedingungen des Überschreibungsfilters definierten IDs übereinstimmen: Keine Bedingung darf ungenutzt bleiben oder ausgelassen werden. Property behavior: - *required* if `evaltype` is set to ”custom expression”

Filterbedingung für die Überschreibung von LLD-Regelprototypen

Das Objekt für die Filterbedingung der Überschreibung von LLD-Regelprototypen definiert eine separate Prüfung, die für den Wert eines LLD-Makros durchgeführt wird. Es hat die folgenden Eigenschaften:

Property	Type	Description
macro	string	LLD-Makro, für das die Prüfung durchgeführt wird. Property behavior: - *required*
value	string	Wert für den Vergleich. Property behavior: - *required* if `operator` is set to ”matches regular expression” or ”does not match regular expression”
formulaid	string	Beliebige eindeutige ID, die verwendet wird, um aus einem benutzerdefinierten Ausdruck auf die Bedingung zu verweisen. Darf nur Großbuchstaben enthalten. Die ID muss vom Benutzer definiert werden, wenn Filterbedingungen geändert werden, wird jedoch bei einer späteren Abfrage erneut generiert. Property behavior: - *required* if `evaltype` of LLD rule prototype override filter object is set to ”custom expression”
operator	integer	Bedingungsoperator. Mögliche Werte: 8 - *(Standard)* entspricht regulärem Ausdruck; 9 - entspricht nicht regulärem Ausdruck; 12 - existiert; 13 - existiert nicht.

Überschreibungsoperation für LLD-Regelprototypen

Die Überschreibungsoperation für LLD-Regelprototypen ist eine Kombination aus Bedingungen und Aktionen, die für das Prototyp-Objekt ausgeführt werden. Sie hat die folgenden Eigenschaften:

Eigenschaft	Type	Beschreibung
operationobject	integer	Typ des erkannten Objekts, für das die Aktion ausgeführt werden soll. Mögliche Werte: 0 - Datenpunkt-Prototyp; 1 - Auslöser-Prototyp; 2 - Graph-Prototyp; 3 - Host-Prototyp. Property behavior: - *required*
operator	integer	Operator der Überschreibungsbedingung. Mögliche Werte: 0 - *(Standard)* gleich; 1 - ungleich; 2 - enthält; 3 - enthält nicht; 8 - entspricht; 9 - entspricht nicht.
value	string	Muster zum Abgleich mit dem Namen des Datenpunkt-, Auslöser-, Graph- oder Host-Prototyps, abhängig vom ausgewählten Objekt.
opstatus	object	Statusobjekt der Überschreibungsoperation für Datenpunkt-, Auslöser- und Host-Prototyp-Objekte.
opdiscover	object	Objekt für den Erkennungsstatus der Überschreibungsoperation (alle Objekttypen).
opperiod	object	Objekt für den Zeitraum (Aktualisierungsintervall) der Überschreibungsoperation für Datenpunkt-Prototyp-Objekte.

Eigenschaft	Type	Beschreibung
ophistory	object	Verlaufsobjekt der Überschreibungsoperation für Datenpunkt-Prototyp-Objekte.
optrends	object	Trendobjekt der Überschreibungsoperation für Datenpunkt-Prototyp-Objekte.
opseverity	object	Schweregradobjekt der Überschreibungsoperation für Auslöser-Prototyp-Objekte.
optag	object/array	Tag-Objekt der Überschreibungsoperation für Auslöser- und Host-Prototyp-Objekte.
optemplate	object/array	Vorlagenobjekt der Überschreibungsoperation für Host-Prototyp-Objekte.
opinventory	object	Inventarobjekt der Überschreibungsoperation für Host-Prototyp-Objekte.

Status der Überschreibungsoperation des LLD-Regelprototyps

Status der Überschreibungsoperation des LLD-Regelprototyps, der für das entdeckte Objekt festgelegt wird. Er hat die folgenden Eigenschaften:

Property	Type	Beschreibung
status	integer	Überschreibt den Status für das ausgewählte Objekt. Mögliche Werte: 0 - Aktiviert erstellen; 1 - Deaktiviert erstellen. Property behavior: - *erforderlich*

LLD-Regelprototyp-Override-Operation „discover“

Der „discover“-Status der Override-Operation eines LLD-Regelprototyps wird für das als entdeckt gesetzte Objekt verwendet. Er hat die folgenden Eigenschaften:

Eigenschaft	Type	Beschreibung
discover	integer	Überschreibt den „discover“-Status für das ausgewählte Objekt. Mögliche Werte: 0 - Ja, die Objekte weiterhin entdecken; 1 - Nein, neue Objekte werden nicht entdeckt und bestehende werden als verloren markiert. Property behavior: - *erforderlich*

Betriebszeitraum der LLD-Regelprototypüberschreibung

Der Betriebszeitraum der LLD-Regelprototypüberschreibung ist ein Wert für das Aktualisierungsintervall, der für den ermittelten Datenpunkt festgelegt wird. Er verfügt über die folgenden Eigenschaften:

Property	Type	Description
delay	string	Überschreibt das Aktualisierungsintervall des Datenpunktprototyps. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300), einen Wert mit einem Zeitsuffix (z. B. 30s, 1m, 2h, 1d) oder optional ein oder mehrere benutzerdefinierte Intervalle, die jeweils durch Semikolons getrennt sind. Benutzerdefinierte Intervalle können eine Kombination aus flexiblen Intervallen und Zeitplanintervallen sein. Akzeptiert Benutzermakros oder LLD-Makros. Bei Verwendung muss der Wert aus einem einzelnen Makro bestehen. Mehrere Makros oder Makros in Kombination mit Text werden nicht unterstützt. Flexible Intervalle können als zwei durch einen Schrägstrich getrennte Makros geschrieben werden (z. B. `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`). Beispiel: `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` Verhalten der Eigenschaft: *- required*

Verlauf der Überschreibungsoperation des LLD-Regelprototyps

Wert der Überschreibungsoperation des LLD-Regelprototyps, der für den erkannten Datenpunkt festgelegt wird. Er hat die folgenden Eigenschaften:

Eigenschaft	Type	Beschreibung
history	string	Überschreibt den Verlauf des Datenpunktprototyps, also die Zeiteinheit dafür, wie lange die Verlaufsdaten gespeichert werden sollen. Akzeptiert auch Benutzermakro und LLD-Makro. Property behavior: *- erforderlich*

Trends der Überschreibungsoperation von LLD-Regelprototypen

Der Wert der Trends der Überschreibungsoperation von LLD-Regelprototypen wird für den erkannten Datenpunkt festgelegt. Er hat die folgenden Eigenschaften:

Eigenschaft	Type	Beschreibung
trends	string	Überschreibt die Trends des Datenpunktprototyps; dies ist eine Zeiteinheit dafür, wie lange die Trenddaten gespeichert werden sollen. Akzeptiert auch Benutzermakro und LLD-Makro. Property behavior: *- required*

Schweregrad der Überschreibungsoperation für LLD-Regelprototypen

Wert des Schweregrads der Überschreibungsoperation für LLD-Regelprototypen, der für den erkannten Auslöser festgelegt wird. Er hat die folgenden Eigenschaften:

Eigenschaft	Typ	Beschreibung
severity	integer	Überschreibt den Schweregrad des Auslöserprototyps. Mögliche Werte: 0 - *(Standard)* nicht klassifiziert; 1 - Information; 2 - Warnung; 3 - durchschnittlich; 4 - hoch; 5 - Katastrophe. Verhalten der Eigenschaft: - *erforderlich*

Tag einer Überschreibungsoperation für LLD-Regelprototypen

Das Tag-Objekt einer Überschreibungsoperation für LLD-Regelprototypen enthält den Tag-Namen und den Wert, die für das erkannte Objekt gesetzt werden. Es hat die folgenden Eigenschaften:

Property	Type	Description
tag	string	Neuer Tag-Name. Property behavior: - *required*
value	string	Neuer Tag-Wert.

Vorlage für die Überschreibungsoperation des LLD-Regelprototyps

Objekt der Vorlage für die Überschreibungsoperation des LLD-Regelprototyps, das mit dem erkannten Host verknüpft ist. Es hat die folgenden Eigenschaften:

Eigenschaft	Type	Beschreibung
templateid	ID	Überschreibt die Vorlage der verknüpften Vorlagen des Host-Prototyps. Verhalten der Eigenschaft: - *erforderlich*

Inventar des Überschreibungsvorgangs für LLD-Regelprototypen

Wert des Inventarmodus für den Überschreibungsvorgang eines LLD-Regelprototyps, der für den erkannten Host festgelegt wird. Er hat die folgenden Eigenschaften:

Eigenschaft	Type	Beschreibung
inventory_mode	integer	Den Inventarmodus des Host-Prototyps überschreiben. Mögliche Werte: -1 - deaktiviert; 0 - *(Standard)* manuell; 1 - automatisch. Verhalten der Eigenschaft: - *erforderlich*

discoveryruleprototype.create

Beschreibung

`object discoveryruleprototype.create(object/array lldRules)`

Diese Methode ermöglicht es Ihnen, neue LLD-Regelvorlagen zu erstellen.

Note:
Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellende LLD-Regelprototypen.

Zusätzlich zu den standardmäßigen Eigenschaften von LLD-Regelprototypen akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
filter	object	Filter für LLD-Regelprototypen für die LLD-Regel.
preprocessing	object/array	Optionen für die Vorverarbeitung von LLD-Regelprototypen.
lld_macro_paths	object/array	Optionen für lld_macro_path von LLD-Regelprototypen.
overrides	object/array	Optionen für Überschreibungen von LLD-Regelprototypen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten LLD-Regelprototypen unter der Eigenschaft `itemids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen LLD-Regelprototypen.

Beispiele

Erstellen eines LLD-Regelprototyps

Erstellen Sie einen LLD-Regelprototypen (Typ: Verschachtelt), um Tablespaces in einer Datenbankinstanz zu erkennen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.create",
    "params": {
        "name": "Tablespaces für {#DB} erkennen",
        "key_": "db.tablespace.discovery[{#DB}]",
        "hostid": "10084",
        "ruleid": "47251",
        "type": 23
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "47252"
        ]
    },
    "id": 1
}
```

Siehe auch

- LLD-Regelprototyp-Filter
- LLD-Makropfade
- LLD-Regelprototyp-Vorverarbeitung

Quelle

CDiscoveryRulePrototype::create() in *ui/include/classes/api/services/CDiscoveryRulePrototype.php*.

discoveryruleprototype.delete

Beschreibung

`object discoveryruleprototype.delete(array lldRuleIds)`

Diese Methode ermöglicht es Ihnen, LLD-Regelvorlagen zu löschen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der zu löschenden LLD-Regelprototypen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten LLD-Regelprototypen unter der Eigenschaft `ruleids` enthält.

Beispiele

Mehrere LLD-Regelprototypen löschen

Löschen Sie zwei LLD-Regelprototypen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.delete",
    "params": [
        "47252",
        "47253"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "ruleids": [
            "47252",
            "47253"
        ]
    },
    "id": 1
}
```

Quelle

CDiscoveryRulePrototype::delete() in *ui/include/classes/api/services/CDiscoveryRulePrototype.php*.

discoveryruleprototype.get

Beschreibung

`integer/array discoveryruleprototype.get(object parameters)`

Die Methode ermöglicht es Ihnen, LLD-Regelvorlagen anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
discoveryids	ID/array	Gibt nur LLD-Regelprototypen zurück, die zu den angegebenen LLD-Regeln oder LLD-Regelprototypen gehören.
itemids	ID/array	Gibt nur LLD-Regelprototypen mit den angegebenen IDs zurück.
groupids	ID/array	Gibt nur LLD-Regelprototypen zurück, die zu den Hosts aus den angegebenen Gruppen gehören.
hostids	ID/array	Gibt nur LLD-Regelprototypen zurück, die zu den angegebenen Hosts gehören.
inherited	boolean	Wenn auf `true` gesetzt, werden nur LLD-Regelprototypen zurückgegeben, die von einer Vorlage geerbt wurden.
interfaceids	ID/array	Gibt nur LLD-Regelprototypen zurück, die die angegebenen Host-Schnittstellen verwenden.
monitored	boolean	Wenn auf `true` gesetzt, werden nur aktivierte LLD-Regelprototypen zurückgegeben, die zu überwachten Hosts gehören.
templated	boolean	Wenn auf `true` gesetzt, werden nur LLD-Regelprototypen zurückgegeben, die zu Vorlagen gehören.
templateids	ID/array	Gibt nur LLD-Regelprototypen zurück, die zu den angegebenen Vorlagen gehören.
selectDiscoveryRule	query	Gibt eine Eigenschaft `discoveryRule` mit der übergeordneten LLD-Regel des LLD-Regelprototyps zurück. Wenn das übergeordnete Element ein Discovery-Regelprototyp ist, wird eine leere Antwort zurückgegeben.
selectDiscoveryRulePrototype	query	Gibt eine Eigenschaft `discoveryRulePrototype` mit dem übergeordneten LLD-Regelprototyp des LLD-Regelprototyps zurück. Wenn das übergeordnete Element eine Discovery-Regel (kein Prototyp) ist, wird eine leere Antwort zurückgegeben.
selectDiscoveryRulePrototypes	query	Gibt eine Eigenschaft `discoveryRulePrototypes` mit untergeordneten LLD-Regelprototypen zurück, die zum LLD-Regelprototyp gehören. Unterstützt `count`.
selectFilter	query	Gibt eine Eigenschaft `filter` mit Daten des Filters zurück, der vom LLD-Regelprototyp verwendet wird.
selectGraphs	query	Gibt eine Eigenschaft `graphs` mit Graphprototypen zurück, die zum LLD-Regelprototyp gehören. Unterstützt `count`.
selectHostPrototypes	query	Gibt eine Eigenschaft `hostPrototypes` mit Host-Prototypen zurück, die zum LLD-Regelprototyp gehören. Unterstützt `count`.
selectHosts	query	Gibt eine Eigenschaft `hosts` mit einem Array von Hosts zurück, zu denen der LLD-Regelprototyp gehört.
selectItems	query	Gibt eine Eigenschaft `items` mit Datenpunktprototypen zurück, die zum LLD-Regelprototyp gehören. Unterstützt `count`.
selectTriggers	query	Gibt eine Eigenschaft `triggers` mit Auslöserprototypen zurück, die zum LLD-Regelprototyp gehören. Unterstützt `count`.
selectLLDMacroPaths	query	Gibt eine Eigenschaft `lld_macro_paths` mit einer Liste von LLD-Makros und Pfaden zu Werten zurück, die jedem entsprechenden Makro zugewiesen sind.
selectPreprocessing	query	Gibt eine Eigenschaft `preprocessing` mit Vorverarbeitungsoptionen des LLD-Regelprototyps zurück.
selectOverrides	query	Gibt eine Eigenschaft `lld_rule_overrides` mit einer Liste von Override-Filtern, Bedingungen und Operationen zurück, die auf Prototyp-Objekte angewendet werden.

Parameter	Type	Beschreibung
filter	object	Gibt nur die Ergebnisse zurück, die exakt dem angegebenen Filter entsprechen. Akzeptiert ein Objekt, bei dem die Schlüssel Eigenschaftsnamen sind und die Werte entweder ein einzelner Wert oder ein Array von Werten sind, mit denen abgeglichen werden soll. Unterstützt keine Eigenschaften vom Datentyp `text`. Unterstützt zusätzliche Eigenschaften: `host` - technischer Name des Hosts, zu dem der LLD-Regelprototyp gehört.
limitSelects	integer	Begrenzt die Anzahl der von Unterabfragen zurückgegebenen Datensätze. Gilt für die folgenden Unterabfragen: `selectItems`, `selectGraphs`, `selectTriggers`.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `itemid`, `name`, `key_`, `delay`, `type`, `status`.
countOutput	boolean	Diese Parameter sind in der Referenzkommentierung beschrieben.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` gibt entweder Folgendes zurück:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Abrufen von Prototypen für Discovery-Regeln von einem Host

Rufen Sie alle Prototypen für Discovery-Regeln für eine bestimmte Host-ID ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.get",
    "params": {
        "hostids": "10084"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "47253",
            "type": "23",
            "snmp_oid": "",
```

```
            "hostid": "10084",
            "name": "Tablespaces für {#DB} ermitteln",
            "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
}
```


Siehe auch

- Graph-Prototyp
- Host
- Datenpunkt-Prototyp
- LLD-Regelprototyp-Filter
- Auslöser-Prototyp

Quelle

CDiscoveryRulePrototype::get() in *ui/include/classes/api/services/CDiscoveryRulePrototype.php*.

discoveryruleprototype.update

Beschreibung

`object discoveryruleprototype.update(object/array lldRules)`

Diese Methode ermöglicht es Ihnen, vorhandene LLD-Regelvorlagen zu aktualisieren.

Beachten Sie, dass die Aktualisierung bereits entdeckter Vorlagen eingeschränkt ist.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu aktualisierende Eigenschaften von LLD-Regelprototypen.

Die Eigenschaft `itemid` muss für jeden LLD-Regelprototyp definiert werden, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Zusätzlich zu den standardmäßigen Eigenschaften von LLD-Regelprototypen akzeptiert die Methode die folgenden Parameter.

Parameter	Typ	Beschreibung
filter	object	Filter für LLD-Regelprototypen zum Ersetzen des vorhandenen Filters.
preprocessing	object/array	Optionen für die Vorverarbeitung von LLD-Regelprototypen zum Ersetzen der vorhandenen Vorverarbeitungsoptionen. Parameterverhalten: - *schreibgeschützt* für vererbte Objekte
lld_macro_paths	object/array	Optionen für lld_macro_path von LLD-Regelprototypen zum Ersetzen der vorhandenen Optionen für lld_macro_path. Parameterverhalten: - *schreibgeschützt* für vererbte Objekte
overrides	object/array	Optionen für Überschreibungen von LLD-Regelprototypen zum Ersetzen der vorhandenen Überschreibungsoptionen. Parameterverhalten: - *schreibgeschützt* für vererbte Objekte

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten LLD-Regelprototypen unter der Eigenschaft `itemids` enthält.

Beispiele

Vorverarbeitungsoptionen des LLD-Regelprototyps aktualisieren

Aktualisieren Sie einen LLD-Regelprototyp mit einer JSONPath-Vorverarbeitungsregel. Request:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.update",
    "params": {
        "itemid": "47253",
        "preprocessing": [
            {
                "type": 12,
                "params": "$.tablespaces",
                "error_handler": 1
            }
        ]
```

```
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "47253"
        ]
    },
    "id": 1
}
```

Quelle

CDiscoveryRulePrototype::update() in *ui/include/classes/api/services/CDiscoveryRulePrototype.php*.

Medientyp

Diese Klasse wurde für die Arbeit mit Medientypen entwickelt.

Objektreferenzen:

- Medientyp
  - webhook-Parameter
  - Skriptparameter
- Nachrichtenvorlage

Verfügbare Methoden:

- mediatype.create - neue Medientypen erstellen
- mediatype.delete - Medientypen löschen
- mediatype.get - Medientypen abrufen
- mediatype.update - Medientypen aktualisieren

Medientyp Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `mediatype` API.

Medientyp

Das Medientypobjekt verfügt über die folgenden Eigenschaften.

Property	Type	Beschreibung
mediatypeid	ID	ID des Medientyps. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
name	string	Name des Medientyps. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge

Property	Type	Beschreibung
type	integer	Vom Medientyp verwendetes Transportverfahren. Mögliche Werte: 0 - E-Mail; 1 - Skript; 2 - SMS; 4 - Webhook; 5 - Push. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
exec_path	string	Name der Skriptdatei (z. B. notification.sh), die sich in dem Verzeichnis befindet, das im Server-Konfigurationsparameter AlertScriptsPath angegeben ist. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „Script“ gesetzt ist
gsm_modem	string	Name des seriellen Geräts des GSM-Modems. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „SMS“ gesetzt ist
passwd	string	Authentifizierungskennwort. Verhalten der Eigenschaft: - *unterstützt*, wenn `smtp_authentication` auf „Normal password“ gesetzt ist
provider	integer	E-Mail-Anbieter. Mögliche Werte: 0 - *(Standard)* Generisches SMTP; 1 - Gmail; 2 - Gmail-Relay; 3 - Office365; 4 - Office365-Relay.
smtp_email	string	E-Mail-Adresse, von der Benachrichtigungen gesendet werden. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „Email“ gesetzt ist
smtp_helo	string	SMTP-HELO. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „Email“ gesetzt ist
smtp_server	string	SMTP-Server. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „Email“ gesetzt ist
smtp_port	integer	Port des SMTP-Servers, zu dem eine Verbindung hergestellt werden soll. Standard: 25. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „Email“ gesetzt ist
smtp_security	integer	Zu verwendende Sicherheitsstufe der SMTP-Verbindung. Mögliche Werte: 0 - *(Standard)* Keine; 1 - STARTTLS; 2 - SSL/TLS. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „Email“ gesetzt ist

Property	Type	Beschreibung
smtp_verify_host	integer	SSL-Überprüfung des Hosts für SMTP. Mögliche Werte: 0 - *(Standard)* Nein; 1 - Ja. Verhalten der Eigenschaft: - *unterstützt*, wenn `smtp_security` auf „STARTTLS“ oder „SSL/TLS“ gesetzt ist
smtp_verify_peer	integer	SSL-Überprüfung des Peers für SMTP. Mögliche Werte: 0 - *(Standard)* Nein; 1 - Ja. Verhalten der Eigenschaft: - *unterstützt*, wenn `smtp_security` auf „STARTTLS“ oder „SSL/TLS“ gesetzt ist
smtp_authentication	integer	Zu verwendende SMTP-Authentifizierungsmethode. Mögliche Werte: 0 - *(Standard)* Keine; 1 - Normales Kennwort; 2 - OAuth-Token. OAuth-Authentifizierung ist für den E-Mail-Anbieter *Office365-Relay* nicht zulässig. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „Email“ gesetzt ist
redirection_url	string	Zabbix-Frontend-URL, zu der nach der OAuth-Autorisierung zurückgeleitet wird. Standard: Wert der API-Einstellungseigenschaft `url` mit dem Teil `zabbix.php?action=oauth.authorize` Verhalten der Eigenschaft: - *erforderlich*, wenn `smtp_authentication` auf „OAuth token“ gesetzt ist
client_id	string	Die beim OAuth-Autorisierungsserver registrierte Client-ID. Verhalten der Eigenschaft: - *erforderlich*, wenn `smtp_authentication` auf „OAuth token“ gesetzt ist
client_secret	string	Das beim OAuth-Autorisierungsserver registrierte Clientgeheimnis. Nur für Benutzer des Typs Super Admin zugänglich. Verhalten der Eigenschaft: - *erforderlich*, wenn `smtp_authentication` auf „OAuth token“ gesetzt ist
authorization_url	string	OAuth-URL mit Parametern zum Abrufen von Zugriffs- und Aktualisierungstoken. Verhalten der Eigenschaft: - *erforderlich*, wenn `smtp_authentication` auf „OAuth token“ gesetzt ist

Property	Type	Beschreibung
token_url	string	OAuth-URL zum Austauschen des Autorisierungstokens gegen Zugriffs- und Aktualisierungstoken. Diese URL wird vom Server auch zum Aktualisieren eines ungültigen Zugriffstokens verwendet. Verhalten der Eigenschaft: - *erforderlich*, wenn `smtp_authentication` auf „OAuth token“ gesetzt ist
tokens_status	integer	Bitmaske für den Status der Token. Mögliche Werte: 0 - (Standard) Beide Token enthalten einen ungültigen Wert; 1 - Das Zugriffstoken enthält einen gültigen Wert; 2 - Das Aktualisierungstoken enthält einen gültigen Wert; 3 - Beide Token enthalten einen gültigen Wert. Verhalten der Eigenschaft: - *unterstützt*, wenn `smtp_authentication` auf „OAuth token“ gesetzt ist
access_token	string	Wert des OAuth-Zugriffstokens. Verhalten der Eigenschaft: - *erforderlich*, wenn `smtp_authentication` auf „OAuth token“ gesetzt ist
access_token_updated	timestamp	Zeitstempel der letzten Änderung von `access_token`, die vom Server bei der Aktualisierung mit `refresh_token` oder über die API bei Tokenänderungen vorgenommen wurde. Verhalten der Eigenschaft: - *unterstützt*, wenn `smtp_authentication` auf „OAuth token“ gesetzt ist
access_expires_in	integer	Zeit in Sekunden, nach der `access_token` veraltet ist und eine Anfrage an `refresh_url` erforderlich wird. Wird vom Zabbix-Server bei der Aktualisierung von `access_token` oder von der API bei Tokenänderungen gesetzt. Der Zeitstempel wird durch Addition des Werts von `access_token_updated` berechnet. Verhalten der Eigenschaft: - *unterstützt*, wenn `smtp_authentication` auf „OAuth token“ gesetzt ist
refresh_token	string	Wert des OAuth-Aktualisierungstokens. Verhalten der Eigenschaft: - *erforderlich*, wenn `smtp_authentication` auf „OAuth token“ gesetzt ist
status	integer	Gibt an, ob der Medientyp aktiviert ist. Mögliche Werte: 0 - *(Standard)* Aktiviert; 1 - Deaktiviert.
username	string	Benutzername. Verhalten der Eigenschaft: - *unterstützt*, wenn `smtp_authentication` auf „Normal password“ gesetzt ist

Property	Type	Beschreibung
maxsessions	integer	Maximale Anzahl von Benachrichtigungen, die parallel verarbeitet werden können. Möglicher Wert, wenn `type` auf „SMS“ gesetzt ist: 1. Mögliche Werte, wenn `type` auf „Email“, „Script“ oder „Webhook“ gesetzt ist: 0-100. Standard: 1.
maxattempts	integer	Maximale Anzahl der Versuche, eine Benachrichtigung zu senden. Mögliche Werte: 1-100. Standard: 3.
attempt_interval	string	Intervall zwischen Wiederholungsversuchen. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 30, 60) oder einen Wert mit einem Zeitsuffix (z. B. `30s`, `1m`, `1h`). Mögliche Werte: 0-1h. Standard: 10s.
message_format	integer	Nachrichtenformat. Mögliche Werte: 0 - Nur Text; 1 - *(Standard)* HTML.
script	text	Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „Email“ gesetzt ist Text des Webhook-Skripts (JavaScript).
timeout	string	Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „Webhook“ gesetzt ist Zeitüberschreitung des Webhook-Skripts. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60) oder einen Wert mit einem Zeitsuffix (z. B. `30s`, `1m`). Mögliche Werte: 1-60s. Standard: 30s.
process_tags	integer	Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „Webhook“ gesetzt ist Verarbeitet JSON-Eigenschaftswerte in der Antwort des Webhook-Skripts als Tags. Diese Tags werden zu allen vorhandenen Problem-Tags hinzugefügt. Mögliche Werte: 0 - *(Standard)* Antwort des Webhook-Skripts ignorieren; 1 - Antwort des Webhook-Skripts als Tags verarbeiten. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „Webhook“ gesetzt ist

Property	Type	Beschreibung
show_event_menu	integer	Fügt einen Eintrag im Ereignismenü hinzu, der auf eine benutzerdefinierte URL verweist. Fügt außerdem die Eigenschaft `urls` zur Ausgabe von problem.get und event.get hinzu. Mögliche Werte: 0 - *(Standard)* Keinen Eintrag im Ereignismenü und keine Eigenschaft `urls` einschließen; 1 - Eintrag im Ereignismenü und Eigenschaft `urls` einschließen.
event_menu_url	string	Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „Webhook“ gesetzt ist URL, die im Eintrag des Ereignismenüs und in der von problem.get und event.get zurückgegebenen Eigenschaft `urls` verwendet wird.
event_menu_name	string	Verhalten der Eigenschaft: - *unterstützt*, wenn `show_event_menu` auf „Include event menu entry and `urls` property“ gesetzt ist Name, der für den Eintrag des Ereignismenüs und in der von problem.get und event.get zurückgegebenen Eigenschaft `urls` verwendet wird.
parameters	array	Verhalten der Eigenschaft: - *unterstützt*, wenn `show_event_menu` auf „Include event menu entry and `urls` property“ gesetzt ist Parameter für Webhook oder Skript.
description	text	Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „Webhook“ oder „Script“ gesetzt ist Beschreibung des Medientyps.

webhook-Parameter

webhook-Parameter haben die folgenden Eigenschaften.

Eigenschaft	Type	Beschreibung
name	string	Parametername. Eigenschaftsverhalten: - *erforderlich*
value	string	Parameterwert, unterstützt Makros. Unterstützte Makros sind auf der Seite Unterstützte Makros beschrieben.

Skriptparameter

Skriptparameter haben die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
sortorder	integer	Die Reihenfolge, in der Parameterwerte als Befehlszeilenargumente an das Skript übergeben werden, beginnend mit 0 als erstem Wert. Verhalten der Eigenschaft: - *erforderlich*
value	string	Parameterwert, unterstützt Makros. Unterstützte Makros werden auf der Seite Unterstützte Makros beschrieben.

Nachrichtenvorlage

Das Nachrichtenvorlagenobjekt definiert eine Vorlage, die als Standardnachricht für Aktionsoperationen zum Senden einer Benachrichtigung verwendet wird. Es hat die folgenden Eigenschaften.

Eigenschaft	Type	Beschreibung
eventsource	integer	Ereignisquelle. Mögliche Werte: 0 - Auslöser; 1 - Discovery; 2 - Autoregistrierung; 3 - Intern; 4 - Services. Verhalten der Eigenschaft: - *erforderlich*
recovery	integer	Operationsmodus. Mögliche Werte: 0 - Operationen; 1 - Wiederherstellungsoperationen; 2 - Aktualisierungsoperationen. Verhalten der Eigenschaft: - *erforderlich*
subject	string	Betreff der Nachricht.
message	string	Nachrichtentext.

mediatype.create

Beschreibung

`object mediatype.create(object/array mediaTypes)`

Mit dieser Methode können Sie neue Medientypen erstellen.

Note:
Diese Methode ist nur für den Benutzertyp *Super-Admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.
Wenn im Feature-Flag `$ZBX_FEATURE_FLAGS['media_type_denylist']` in der Frontend-Konfigurationsdatei Zustellungsmethoden angegeben sind, darf der Benutzer die Methode für die entsprechenden Medientypen nicht aufrufen.

Parameter

`(object/array)` Zu erstellende Medientypen.

Zusätzlich zu den Standard-Medientyp-Eigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
message_templates	array	Nachrichtenvorlagen, die für den Medientyp erstellt werden sollen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Medientypen unter der Eigenschaft `mediatypeids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Medientypen.

Beispiele

Erstellen eines E-Mail-Medientyps

Erstellen Sie einen neuen E-Mail-Medientyp mit einem benutzerdefinierten SMTP-Port und Nachrichtenvorlagen.

Anfrage:

```
{
    "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": "Problem: {EVENT.NAME}",
                "message": "Problem \"{EVENT.NAME}\" auf Host \"{HOST.NAME}\" begann um {EVENT.TIME}."
            },
            {
                "eventsource": "0",
                "recovery": "1",
                "subject": "Behoben in {EVENT.DURATION}: {EVENT.NAME}",
                "message": "Problem \"{EVENT.NAME}\" auf Host \"{HOST.NAME}\" wurde um {EVENT.RECOVERY.TIM
            },
            {
                "eventsource": "0",
                "recovery": "2",
                "subject": "Aktualisiertes Problem in {EVENT.AGE}: {EVENT.NAME}",
                "message": "{USER.FULLNAME} hat das Problem \"{EVENT.NAME}\" auf Host \"{HOST.NAME}\" am {
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "7"
        ]
    },
    "id": 1
}
```

Erstellen eines Skript-Medientyps

Erstellen Sie einen neuen Skript-Medientyp mit einem benutzerdefinierten Wert für die Anzahl der Versuche und das Intervall zwischen ihnen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.create",
    "params": {
        "type": "1",
        "name": "Push-Benachrichtigungen",
        "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "8"
        ]
    },
    "id": 1
}
```

Erstellen eines webhook-Medientyps

Erstellen Sie einen neuen webhook-Medientyp.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "9"
        ]
    },
    "id": 1
}
```

Quelle

CMediaType::create() in *ui/include/classes/api/services/CMediaType.php*.

mediatype.delete

Beschreibung

`object mediatype.delete(array mediaTypeIds)`

Mit dieser Methode können Sie Medientypen löschen.

Note:
Diese Methode ist nur für den Benutzertyp *Super-Admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.
 Wenn Übermittlungsmethoden im Feature-Flag `$ZBX_FEATURE_FLAGS['media_type_denylist']` in der Frontend-Konfigurationsdatei angegeben sind, darf der Benutzer die Methode für die entsprechenden Medientypen nicht aufrufen.

Parameter

`(array)` IDs der zu löschenden Medientypen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Medientypen unter der Eigenschaft `mediatypeids` enthält.

Beispiele

Mehrere Medientypen löschen

Löschen Sie zwei Medientypen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.delete",
    "params": [
        "3",
        "5"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "3",
            "5"
        ]
    },
    "id": 1
}
```

Quelle

CMediaType::delete() in *ui/include/classes/api/services/CMediaType.php*.

mediatype.get

Beschreibung

`integer/array mediatype.get(object parameters)`

Mit dieser Methode können Sie Medientypen entsprechend den angegebenen Parametern abrufen.

> **Note:**
> Diese Methode steht Benutzern jeden Typs zur Verfügung. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.
 Wenn Übermittlungsmethoden im Feature-Flag `$ZBX_FEATURE_FLAGS['media_type_denylist']` in der Frontend-Konfigurationsdatei angegeben sind, darf der Benutzer die Methode für die entsprechenden Medientypen nicht aufrufen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

> **Note:**
> Beim Anfordern benutzerbezogener Informationen von Medientypen können Benutzer des Typs *Admin* und *User* Daten nur über ihren eigenen Benutzer abrufen. Ein Beispiel finden Sie unter Abrufen von Medientypen als *Admin*.

Parameter	Type	Beschreibung
mediatypeids	ID/array	Gibt nur Medientypen mit den angegebenen IDs zurück.
mediaids	ID/array	Gibt nur Medientypen zurück, die von den angegebenen Medien verwendet werden.
userids	ID/array	Gibt nur Medientypen zurück, die von den angegebenen Benutzern verwendet werden.
selectActions	query	Gibt eine Eigenschaft `actions` mit den Aktionen zurück, die den Medientyp verwenden.
selectMessageTemplates	query	Gibt eine Eigenschaft `message_templates` mit einem Array von Medientyp-Nachrichten zurück. Parameterverhalten: - *unterstützt* für Benutzer des Typs *Super admin*
selectUsers	query	Gibt eine Eigenschaft `users` mit den Benutzern zurück, die den Medientyp verwenden. Siehe `user.get` für Einschränkungen je nach Benutzertyp.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `mediatypeid`.
filter	object	Gibt nur die Ergebnisse zurück, die exakt dem angegebenen Filter entsprechen. Akzeptiert ein Objekt, bei dem die Schlüssel Eigenschaftsnamen sind und die Werte entweder ein einzelner Wert oder ein Array von Werten sind, mit denen abgeglichen wird. Unterstützt keine Eigenschaften vom Datentyp `text`. Mögliche Eigenschaften des Media type object für Benutzer des Typs *Admin* und *User*: `mediatypeid`, `name`, `type`, `status`, `maxattempts`.

Parameter	Type	Beschreibung
output	query	Eigenschaften des Media type object, die zurückgegeben werden sollen. Benutzer des Typs *Admin* und *User* können nur die folgenden Eigenschaften abrufen: `mediatypeid`, `name`, `type`, `status`, `maxattempts`, `description`. Ein Beispiel finden Sie unter Abrufen von Medientypen als *Admin*. Standard: `extend`.
search	object	Gibt Ergebnisse zurück, die dem angegebenen Muster entsprechen (Groß-/Kleinschreibung wird nicht beachtet). Akzeptiert ein Objekt, bei dem die Schlüssel Eigenschaftsnamen sind und die Werte Zeichenfolgen sind, nach denen gesucht werden soll. Wenn keine zusätzlichen Optionen angegeben sind, wird eine Suche vom Typ `LIKE "%…%"` durchgeführt. Unterstützt nur Eigenschaften vom Datentyp `string` und `text`. Mögliche Eigenschaften des Media type object für Benutzer des Typs *Admin* und *User*: `name`, `description`.
countOutput	boolean	Diese Parameter werden in der Referenzkommentierung beschrieben.
editable	boolean	
excludeSearch	boolean	
limit	integer	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Kann die folgenden Dinge zurück geben:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Medientypen abrufen

Rufen Sie alle konfigurierten Medientypen ab. Das folgende Beispiel gibt zwei Medientypen zurück:

- E-Mail-Medientyp;
- SMS-Medientyp.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.get",
    "params": {
        "output": "extend",
        "selectMessageTemplates": "extend"
    },
    "id": 1
}
```

Antwort:

```
{
    "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": "Problem: {EVENT.NAME}",
                "message": "Problem begann um {EVENT.TIME} am {EVENT.DATE}\r\nProblemname: {EVENT.NAME
            },
            {
                "eventsource": "0",
                "recovery": "1",
                "subject": "Gelöst: {EVENT.NAME}",
                "message": "Problem wurde um {EVENT.RECOVERY.TIME} am {EVENT.RECOVERY.DATE} gelöst\r\n
            },
            {
                "eventsource": "0",
                "recovery": "2",
                "subject": "Aktualisiertes Problem: {EVENT.NAME}",
                "message": "{USER.FULLNAME} hat das Problem am {EVENT.UPDATE.DATE} um {EVENT.UPDATE.TI
            },
            {
                "eventsource": "1",
                "recovery": "0",
                "subject": "Discovery: {DISCOVERY.DEVICE.STATUS} {DISCOVERY.DEVICE.IPADDRESS}",
                "message": "Discovery-Regel: {DISCOVERY.RULE.NAME}\r\n\r\nGeräte-IP: {DISCOVERY.DEVICE
            },
            {
                "eventsource": "2",
                "recovery": "0",
                "subject": "Autoregistrierung: {HOST.HOST}",
                "message": "Host-Name: {HOST.HOST}\r\nHost-IP: {HOST.IP}\r\nAgent-Port: {HOST.PORT}"
            }
        ],
        "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\nHost: {HOST.NAME}\r\n{EVENT.DATE} {EVENT
                },
                {
                    "eventsource": "0",
                    "recovery": "1",
                    "subject": "",
                    "message": "GELÖST: {EVENT.NAME}\r\nHost: {HOST.NAME}\r\n{EVENT.DATE} {EVENT.TIME}"
                },
                {
                    "eventsource": "0",
                    "recovery": "2",
                    "subject": "",
                    "message": "{USER.FULLNAME} hat das Problem am {EVENT.UPDATE.DATE} um {EVENT.UPDATE.TI
                },
                {
                    "eventsource": "1",
                    "recovery": "0",
                    "subject": "",
                    "message": "Discovery: {DISCOVERY.DEVICE.STATUS} {DISCOVERY.DEVICE.IPADDRESS}"
                },
                {
                    "eventsource": "2",
                    "recovery": "0",
                    "subject": "",
                    "message": "Autoregistrierung: {HOST.HOST}\r\nHost-IP: {HOST.IP}\r\nAgent-Port: {HOST.
                }
            ],
            "parameters": []
```

```
        }
    ],
    "id": 1
}
```

Medientypen als *Admin* abrufen

Als Benutzer vom Typ *Admin* rufen Sie alle aktivierten Medientypen ab, zusammen mit den Benutzern, die diese Medientypen verwenden. Das folgende Beispiel gibt zwei Medientypen zurück:

- E-Mail-Medientyp mit einem Benutzer (nur der eigene Benutzer des Benutzers vom Typ *Admin*);
- SMS-Medientyp ohne Benutzer.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.get",
    "params": {
        "output": "extend",
        "filter": {
            "status": 0
        },
        "selectUsers": "extend"
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```

Skript- und webhook-Medientypen abrufen

Das folgende Beispiel gibt drei Medientypen zurück:

- Skript-Medientyp mit Parametern;
- Skript-Medientyp ohne Parameter;
- webhook-Medientyp mit Parametern.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.get",
    "params": {
        "output": ["mediatypeid", "name", "parameters"],
        "filter": {
            "type": [1, 4]
        }
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```

Siehe auch

- User

Quelle

CMediaType::get() in *ui/include/classes/api/services/CMediaType.php*.

mediatype.update

Beschreibung

`object mediatype.update(object/array mediaTypes)`

Mit dieser Methode können vorhandene Medientypen aktualisiert werden.

Note:
Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.
 Wenn Übermittlungsmethoden im Feature-Flag `$ZBX_FEATURE_FLAGS['media_type_denylist']` in der Frontend-Konfigurationsdatei angegeben sind, darf der Benutzer die Methode für die entsprechenden Medientypen nicht aufrufen.

Parameter

`(object/array)` Zu aktualisierende Medientyp-Eigenschaften.

Die Eigenschaft `mediatypeid` muss für jeden Medientyp definiert werden, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Zusätzlich zu den Standard-Medientyp-Eigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
message_templates	array	Nachrichtenvorlagen zum Ersetzen der aktuellen Nachrichtenvorlagen.

Rückgabewerte

(object) Gibt ein Objekt zurück, das die IDs der aktualisierten Medientypen unter der Eigenschaft "mediatypeids" enthält.

Beispiele

Aktivieren eines Medientyps

Aktivieren Sie einen Medientyp, d. h. setzen Sie seinen Status auf „0".

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.update",
    "params": {
        "mediatypeid": "6",
        "status": "0"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "6"
        ]
    },
    "id": 1
}
```

Quelle

CMediaType::update() in *ui/include/classes/api/services/CMediaType.php*.

MFA

Diese Klasse ist für die Arbeit mit MFA-Methoden (Multi-Faktor-Authentifizierung) vorgesehen.

Objektreferenzen:

- MFA

Verfügbare Methoden:

- mfa.create - neue MFA-Methoden erstellen
- mfa.delete - MFA-Methoden löschen
- mfa.get - MFA-Methoden abrufen
- mfa.update - MFA-Methoden aktualisieren

MFA-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `mfa` API.

MFA

Das MFA-Objekt (Multi-Faktor-Authentifizierung) hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
mfaid	ID	ID der MFA-Methode. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge

Eigenschaft	Typ	Beschreibung
type	integer	Typ der MFA-Methode. Mögliche Werte: 1 - TOTP (zeitbasierte Einmalpasswörter); 2 - Duo Universal Prompt.
name	string	Eindeutiger Name der MFA-Methode. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
hash_function	integer	Typ der Hash-Funktion zur Generierung von TOTP-Codes. Mögliche Werte: 1 - SHA-1; 2 - SHA-256; 3 - SHA-512. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „TOTP“ gesetzt ist
code_length	integer	Länge des Verifizierungscodes. Mögliche Werte: 6 - 6-stellig; 8 - 8-stellig. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „TOTP“ gesetzt ist
api_hostname	string	API-Hostname, der vom Duo-Authentifizierungsdienst bereitgestellt wird. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „Duo Universal Prompt“ gesetzt ist
clientid	string	Client-ID, die vom Duo-Authentifizierungsdienst bereitgestellt wird. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „Duo Universal Prompt“ gesetzt ist
client_secret	string	Client-Secret, das vom Duo-Authentifizierungsdienst bereitgestellt wird. Verhalten der Eigenschaft: - *nur schreibbar* - *erforderlich*, wenn `type` auf „Duo Universal Prompt“ gesetzt ist

mfa.create

Beschreibung

`object mfa.create(object/array MFA methods)`

Diese Methode ermöglicht es Ihnen, neue MFA-Methoden zu erstellen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellende MFA-Methoden.

Die Methode akzeptiert MFA-Methoden mit den Standard-MFA-Methodeneigenschaften.

Rückgabewerte

(`object`) Gibt ein Objekt zurück, das die IDs der erstellten MFA-Methoden unter der Eigenschaft `mfaids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Elemente.

Beispiele

Erstellen von MFA-Methoden

Erstellen Sie eine MFA-Methode „Zabbix TOTP“, die zeitbasierte Einmalpasswörter (TOTP) verwendet, wobei die Hash-Funktion zur Generierung von TOTP-Codes auf SHA-1 und die Länge des Verifizierungscodes auf 6 Ziffern festgelegt ist.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "mfa.create",
    "params": {
        "type": 1,
        "name": "Zabbix TOTP",
        "hash_function": 1,
        "code_length": 6
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mfaids": [
            "1"
        ]
    },
    "id": 1
}
```

Quelle

CMfa::create() in *ui/include/classes/api/services/CMfa.php*.

mfa.delete

Beschreibung

`object mfa.delete(array mfaids)`

Diese Methode ermöglicht es Ihnen, MFA-Methoden zu löschen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Superadmin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

(`array`) IDs der zu löschenden MFA-Methoden.

Rückgabewerte

(`object`) Gibt ein Objekt zurück, das die IDs der gelöschten MFA-Methoden unter der Eigenschaft `mfaids` enthält.

Beispiele

Löschen von MFA-Methoden

Löschen Sie eine MFA-Methode.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "mfa.delete",
```

```
    "params": [
        "2"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mfaids": [
            "2"
        ]
    },
    "id": 1
}
```

Quelle

CMfa::delete() in *ui/include/classes/api/services/CMfa.php*.

mfa.get

Beschreibung

`integer/array mfa.get(object parameters)`

Die Methode ermöglicht es Ihnen, MFA-Methoden anhand der angegebenen Parameter abzurufen.

Note:
Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
mfaids	ID/array	Gibt nur MFA-Methoden mit den angegebenen IDs zurück.
selectUsrgrps	query	Gibt eine `usrgrps`-Eigenschaft mit den MFA-Methoden zugeordneten Benutzergruppen zurück. Unterstützt `count`.
filter	object	Gibt nur die Ergebnisse zurück, die exakt dem angegebenen Filter entsprechen. Akzeptiert ein Objekt, bei dem die Schlüssel Eigenschaftsnamen sind und die Werte entweder ein einzelner Wert oder ein Array von Werten sind, mit denen abgeglichen werden soll. Unterstützte Eigenschaften: `mfaid` - ID der MFA-Methode; `type` - Typ der MFA-Methode.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `name`.

Parameter	Type	Beschreibung
search	object	Gibt Ergebnisse zurück, die dem angegebenen Muster entsprechen (Groß-/Kleinschreibung wird nicht beachtet). Akzeptiert ein Objekt, bei dem die Schlüssel Eigenschaftsnamen sind und die Werte Zeichenfolgen sind, nach denen gesucht werden soll. Wenn keine zusätzlichen Optionen angegeben werden, wird eine Suche vom Typ `LIKE "%…%"` durchgeführt. Unterstützte Eigenschaften: `name`.
countOutput	boolean	Diese Parameter sind in der Referenzkommentierung beschrieben.
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Gibt entweder:

- eine Reihe von Objekten zurück;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde zurück.

Beispiele

MFA-Methoden nach Namen finden

Rufen Sie alle MFA-Methoden ab, die „Zabbix“ in ihrem Namen enthalten.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "mfa.get",
    "params": {
        "output": "extend",
        "search": {
            "name": "Zabbix"
        }
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```

Quelle

CMfa::get() in *ui/include/classes/api/services/CMfa.php*.

mfa.update

Beschreibung

`object mfa.update(object/array MFA methods)`

Diese Methode ermöglicht es Ihnen, vorhandene MFA-Methoden zu aktualisieren.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu aktualisierende Eigenschaften der MFA-Methode.

Die Eigenschaft `mfaid` muss für jedes Element definiert werden, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Die Methode akzeptiert MFA-Methoden mit den Standard-Eigenschaften der MFA-Methode.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten MFA-Methoden unter der Eigenschaft `mfaids` enthält.

Beispiele

Eigenschaften der Methode aktualisieren

Aktualisieren Sie die Hash-Funktion zur Generierung von TOTP-Codes sowie die Länge des Verifizierungscodes für die MFA-Methode „Zabbix TOTP“, die zeitbasierte Einmalpasswörter (TOTP) verwendet.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "mfa.update",
    "params": {
        "mfaid": "1",
        "hash_function": 3,
        "code_length": 8
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mfaids": [
            "1"
        ]
    },
    "id": 1
}
```

Quelle

CMfa::update() in *ui/include/classes/api/services/CMfa.php*.

Modul

Diese Klasse ist für die Arbeit mit Frontend-Modulen ausgelegt.

Objektreferenzen:

- Module

Verfügbare Methoden:

- module.create - neue Module installieren
- module.delete - Module deinstallieren
- module.get - Module abrufen
- module.update - Module aktualisieren

Modul-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `module` API.

Modul

Das Modulobjekt hat die folgenden Eigenschaften.

Property	Type	Description
moduleid	ID	ID des Moduls, wie sie in der Datenbank gespeichert ist. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsoperationen
id	string	Eindeutige Modul-ID, wie sie von einem Entwickler in der Datei `manifest.json` des Moduls definiert wurde. Mögliche Werte für integrierte Module: siehe Beschreibung der Eigenschaft "type" in Dashboard widget. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsoperationen
relative_path	string	Pfad zum Verzeichnis des Moduls relativ zum Verzeichnis des Zabbix Frontend. Mögliche Werte: `widgets/*` - für integrierte Widget-Module; `modules/*` - für Module von Drittanbietern. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsoperationen
status	integer	Gibt an, ob das Modul aktiviert oder deaktiviert ist. Mögliche Werte: 0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
config	object	Modulkonfiguration.

module.create

Beschreibung

`object module.create(object/array modules)`

Mit dieser Methode können Sie neue Frontend-Module installieren.

Note:
Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.
 Wenn `$ZBX_FEATURE_FLAGS['modules_config_enabled']` in der Frontend-Konfigurationsdatei auf `false` gesetzt ist, darf der Benutzer diese Methode nicht aufrufen.

Attention:
Moduldateien müssen manuell in die korrekten Unterverzeichnisse entpackt werden, die der Eigenschaft `relative_path` der Module entsprechen.

Parameter

`(object/array)` Zu erstellende Module.

Die Methode akzeptiert Module mit den Standard-Moduleigenschaften.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der installierten Module unter der Eigenschaft `moduleids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Module.

Beispiele

Installieren eines Moduls

Installieren Sie ein Modul mit dem Status „Aktiviert“.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "module.create",
    "params": {
        "id": "example_module",
        "relative_path": "modules/example_module",
        "status": 1
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "moduleids": [
            "25"
        ]
    },
    "id": 1
}
```

Siehe auch

- Modul
- Frontend Module

Quelle

CModule::create() in *ui/include/classes/api/services/CModule.php*.

module.delete

Beschreibung

`object module.delete(array moduleids)`

Mit dieser Methode können Sie Module deinstallieren.

Note:
Diese Methode ist nur für den Benutzertyp *Super-Admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.
 Wenn `$ZBX_FEATURE_FLAGS['modules_config_enabled']` in der Frontend-Konfigurationsdatei auf `false` gesetzt ist, darf der Benutzer diese Methode nicht aufrufen.

Attention:
Moduldateien müssen manuell entfernt werden.

Parameter

`(array)` IDs der zu deinstallierenden Module.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der deinstallierten Module unter der Eigenschaft `moduleids` enthält.

Beispiele

Deinstallation mehrerer Module

Deinstallieren Sie die Module „27“ und „28“.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "module.delete",
    "params": [
        "27",
        "28"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "moduleids": [
            "27",
            "28"
        ]
    },
    "id": 1
}
```

Quelle

CModule::delete() in *ui/include/classes/api/services/CModule.php*.

module.get

Beschreibung

`integer/array module.get(object parameters)`

Mit dieser Methode können Sie Module entsprechend den angegebenen Parametern abrufen.

Note:
Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.
 Wenn `$ZBX_FEATURE_FLAGS['modules_config_enabled']` in der Frontend-Konfigurationsdatei auf `false` gesetzt ist, darf der Benutzer diese Methode nicht aufrufen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Typ	Beschreibung
moduleids	ID/array	Gibt nur Module mit den angegebenen IDs zurück.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `moduleid`, `relative_path`.
countOutput	boolean	Diese Parameter werden im Referenzkommentar beschrieben.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Gibt entweder:

- eine Reihe von Objekten zurück;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde zurück.

Beispiele

Abrufen eines Moduls anhand der ID

Rufen Sie alle Daten zu den Modulen „1“, „2“ und „25“ ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "module.get",
    "params": {
        "output": "extend",
        "moduleids": [
            "1",
            "2",
            "25"
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```

Siehe auch

- Module
- Dashboard Widget
- Frontend Module

Quelle

CModule::get() in *ui/include/classes/api/services/CModule.php*.

module.update

Beschreibung

`object module.update(object/array modules)`

Mit dieser Methode können Sie vorhandene Module aktualisieren.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.
 Wenn `$ZBX_FEATURE_FLAGS['modules_config_enabled']` in der Frontend-Konfigurationsdatei auf `false` gesetzt ist, darf der Benutzer diese Methode nicht aufrufen.

Parameter

`(object/array)` Zu aktualisierende Moduleigenschaften.

Die Eigenschaft `moduleid` muss für jedes Modul definiert sein, alle anderen Eigenschaften sind optional. Nur die angegebenen Eigenschaften werden aktualisiert.

Die Methode akzeptiert Module mit den Standard-Moduleigenschaften.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Module unter der Eigenschaft `moduleids` enthält.

Beispiele

Deaktivieren eines Moduls

Modul „25“ deaktivieren.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "module.update",
    "params": {
        "moduleid": "25",
        "status": 0
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "moduleids": [
            "25"
        ]
    },
    "id": 1
}
```


Siehe auch

- Modul
- Frontend Module

Quelle

CModule::update() in *ui/include/classes/api/services/CModule.php*.

Problem

Diese Klasse ist für die Arbeit mit Problemen ausgelegt.

Objektreferenzen:

- Problem
  - Medientyp-URL
- Problem-Tag

Verfügbare Methoden:

- problem.get - Probleme abrufen

Problem-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `problem` API.

Problem

> **Note:**
> Probleme werden vom Zabbix-Server erstellt und können nicht über die API geändert werden.

Das Problem-Objekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
eventid	ID	ID des Problemereignisses.
source	integer	Typ des Problemereignisses. Mögliche Werte: 0 - Ereignis, das von einem Auslöser erstellt wurde; 3 - internes Ereignis; 4 - Ereignis, das bei einer Aktualisierung des Servicestatus erstellt wurde.

Eigenschaft	Typ	Beschreibung
object	integer	Typ des Objekts, das mit dem Problemereignis verknüpft ist. Mögliche Werte, wenn `source` auf „Ereignis, das von einem Auslöser erstellt wurde“ gesetzt ist: 0 - Auslöser. Mögliche Werte, wenn `source` auf „internes Ereignis“ gesetzt ist: 0 - Auslöser; 4 - Datenpunkt; 5 - LLD-Regel. Mögliche Werte, wenn `source` auf „Ereignis, das bei einer Aktualisierung des Servicestatus erstellt wurde“ gesetzt ist: 6 - Service.
objectid	ID	ID des verknüpften Objekts.
clock	timestamp	Zeitpunkt, zu dem das Problemereignis erstellt wurde.
ns	integer	Nanosekunden zum Zeitpunkt der Erstellung des Problemereignisses.
r_eventid	ID	ID des Wiederherstellungsereignisses.
r_clock	timestamp	Zeitpunkt, zu dem das Wiederherstellungsereignis erstellt wurde.
r_ns	integer	Nanosekunden zum Zeitpunkt der Erstellung des Wiederherstellungsereignisses.
cause_eventid	ID	ID des Ursacheereignisses.
correlationid	ID	ID der Korrelationsregel, falls dieses Ereignis durch eine globale Korrelationsregel wiederhergestellt wurde.
userid	ID	ID des Benutzers, der das Problem geschlossen hat (falls das Problem manuell geschlossen wurde).
name	string	Aufgelöster Problemname.
acknowledged	integer	Bestätigungsstatus des Problems. Mögliche Werte: 0 - nicht bestätigt; 1 - bestätigt.
severity	integer	Aktueller Schweregrad des Problems. Mögliche Werte: 0 - nicht klassifiziert; 1 - Information; 2 - Warnung; 3 - durchschnittlich; 4 - hoch; 5 - Katastrophe.
suppressed	integer	Gibt an, ob das Problem unterdrückt ist. Mögliche Werte: 0 - Problem befindet sich im normalen Zustand; 1 - Problem ist unterdrückt.
opdata	string	Betriebsdaten mit erweiterten Makros.
urls	array	Aktive Medientyp-URLs.

URL des Medientyps

Das URL-Objekt des Medientyps hat die folgenden Eigenschaften.

Eigenschaft	Type	Beschreibung
name	string	Name der im Medientyp definierten URL.
url	string	Wert der im Medientyp definierten URL.

Die Ergebnisse enthalten nur Einträge für aktive Medientypen mit aktiviertem Ereignismenüeintrag. In Eigenschaften verwendete Makros werden expandiert, aber wenn eine der Eigenschaften ein nicht expandiertes Makro enthält, werden beide Eigenschaften aus den Ergebnissen ausgeschlossen. Unterstützte Makros finden Sie unter *Unterstützte Makros*.

Problem-Tag

Das Problem-Tag-Objekt hat die folgenden Eigenschaften.

Eigenschaft	Type	Beschreibung
tag	string	Name des Problem-Tags.
value	string	Wert des Problem-Tags.

problem.get

Beschreibung

`integer/array problem.get(object parameters)`

Die Methode ermöglicht es Ihnen, Probleme anhand der angegebenen Parameter abzurufen.

Diese Methode dient zum Abrufen ungelöster Probleme. Falls angegeben, ist es außerdem möglich, zusätzlich kürzlich gelöste Probleme abzurufen. Der Zeitraum, der festlegt, wie alt "kürzlich" ist, wird unter *Administration* > *Allgemein* definiert. Probleme, die vor diesem Zeitraum gelöst wurden, werden nicht in der Problem-Tabelle aufbewahrt. Um Probleme abzurufen, die weiter in der Vergangenheit gelöst wurden, verwenden Sie die Methode `event.get`.

Attention:
Diese Methode kann Probleme einer gelöschten Entität zurückgeben, wenn diese Probleme vom Housekeeper noch nicht entfernt wurden.

Note:
Diese Methode ist für Benutzer aller Typen verfügbar. Berechtigungen zum Aufruf der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Typ	Beschreibung
eventids	ID/array	Nur Probleme mit den angegebenen IDs zurückgeben.
groupids	ID/array	Nur Probleme zurückgeben, die von Objekten erstellt wurden, die zu den angegebenen Hostgruppen gehören.
hostids	ID/array	Nur Probleme zurückgeben, die von Objekten erstellt wurden, die zu den angegebenen Hosts gehören.
objectids	ID/array	Nur Probleme zurückgeben, die von den angegebenen Objekten erstellt wurden.
source	integer	Nur Probleme mit dem angegebenen Typ zurückgeben. Eine Liste der unterstützten Ereignistypen finden Sie auf der Seite des Problemereignisobjekts. Standard: 0 - Problem, erstellt durch einen Auslöser.
object	integer	Nur Probleme zurückgeben, die von Objekten des angegebenen Typs erstellt wurden. Eine Liste der unterstützten Objekttypen finden Sie auf der Seite des Problemereignisobjekts. Standard: 0 - Auslöser.
acknowledged	boolean	Wenn auf `true` gesetzt, nur bestätigte Probleme zurückgeben.
action	integer	Nur Probleme zurückgeben, für die die angegebenen Ereignis-Aktualisierungsaktionen ausgeführt wurden. Für mehrere Aktionen verwenden Sie eine Summe beliebiger zulässiger Bitmap-Werte als Bitmaske (zum Beispiel 34 für Bestätigung und Unterdrückung des Ereignisses).

Parameter	Typ	Beschreibung
action_userids	ID/array	Nur Probleme mit den angegebenen IDs von Benutzern zurückgeben, die die Ereignis-Aktualisierungsaktionen für das Problem ausgeführt haben.
suppressed	boolean	Wenn auf `true` gesetzt, nur unterdrückte Probleme zurückgeben.
symptom	boolean	Wenn auf `true` gesetzt, nur Symptom-Problemereignisse zurückgeben.
severities	integer/array	Nur Probleme mit den angegebenen Ereignisschweregraden zurückgeben. Gilt nur, wenn object ein Auslöser ist.
evaltype	integer	Tag-Auswertungsmethode. Mögliche Werte: 0 - *(Standard)* Und/Oder; 2 - Oder.
tags	array	Nur Probleme mit den angegebenen Tags zurückgeben. Format: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Ein leeres Array gibt alle Probleme zurück. Mögliche Operator-Werte: 0 - *(Standard)* Enthält; 1 - Ist gleich; 2 - Enthält nicht; 3 - Ist nicht gleich; 4 - Existiert; 5 - Existiert nicht.
recent	boolean	Wenn auf `true` gesetzt, aktuelle Probleme zurückgeben, einschließlich derjenigen, die innerhalb des `ok_period` behoben wurden, sowie aller nicht behobenen Probleme. Wenn auf `false` gesetzt, nur nicht behobene Probleme zurückgeben.
eventid_from	string	Nur Probleme mit IDs größer oder gleich der angegebenen ID zurückgeben.
eventid_till	string	Nur Probleme mit IDs kleiner oder gleich der angegebenen ID zurückgeben.
time_from	timestamp	Nur Probleme zurückgeben, die nach oder zum angegebenen Zeitpunkt erstellt wurden.
time_till	timestamp	Nur Probleme zurückgeben, die vor oder zum angegebenen Zeitpunkt erstellt wurden.
selectAcknowledges	query	Eine Eigenschaft `acknowledges` mit den Problemaktualisierungen zurückgeben. Problemaktualisierungen werden in umgekehrt chronologischer Reihenfolge sortiert. Das Problemaktualisierungsobjekt hat die folgenden Eigenschaften: `acknowledgeid` - `(ID)` ID der Aktualisierung; `userid` - `(ID)` ID des Benutzers, der das Ereignis aktualisiert hat; `eventid` - `(ID)` ID des aktualisierten Ereignisses; `clock` - `(timestamp)` Zeitpunkt, zu dem das Ereignis aktualisiert wurde; `message` - `(string)` Text der Nachricht; `action` - `(integer)` Typ der Aktualisierungsaktion (siehe `event.acknowledge`); `old_severity` - `(integer)` Ereignisschweregrad vor dieser Aktualisierungsaktion; `new_severity` - `(integer)` Ereignisschweregrad nach dieser Aktualisierungsaktion; `suppress_until` - `(timestamp)` Zeitpunkt, bis zu dem das Ereignis unterdrückt wird; `taskid` - `(ID)` ID der Aufgabe, wenn das aktuelle Ereignis einen Rangwechsel durchläuft; `maintenanceid` - `(ID)` ID der Wartung. Unterstützt `count`.

Parameter	Typ	Beschreibung
selectTags	query	Eine Eigenschaft `tags` mit den Problem-Tags zurückgeben. Ausgabeformat: `[{"tag": "<tag>", "value": "<value>"}, ...]`.
selectSuppressionData	query	Eine Eigenschaft `suppression_data` mit der Liste aktiver Wartungen und manueller Unterdrückungen zurückgeben: `maintenanceid` - `(ID)` ID der Wartung; `userid` - `(ID)` ID des Benutzers, der das Problem unterdrückt hat; `suppress_until` - `(integer)` Zeit, bis zu der das Problem unterdrückt ist.
filter	object	Nur diejenigen Ergebnisse zurückgeben, die exakt dem angegebenen Filter entsprechen. Akzeptiert ein Objekt, bei dem die Schlüssel Eigenschaftsnamen sind und die Werte entweder ein einzelner Wert oder ein Array von Werten sind, mit denen verglichen werden soll. Unterstützt keine Eigenschaften vom Datentyp `text`.
sortfield	string/array	Das Ergebnis nach den angegebenen Eigenschaften sortieren. Mögliche Werte: `eventid`.
countOutput	boolean	Diese Parameter werden in der Referenzbeschreibung erläutert.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Gibt entweder Folgendes zurück:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Abrufen von Auslöser-Problemereignissen

Rufen Sie aktuelle Ereignisse von Auslöser ”15112” ab.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "1245463",
            "source": "0",
            "object": "0",
            "objectid": "15112",
            "clock": "1472457242",
            "ns": "209442442",
            "r_eventid": "1245468",
            "r_clock": "1472457285",
            "r_ns": "125644870",
            "correlationid": "0",
            "userid": "1",
            "name": "Zabbix Agent auf localhost ist seit 5 Minuten nicht erreichbar",
            "acknowledged": "1",
            "severity": "3",
            "cause_eventid": "0",
            "suppressed": "1",
            "acknowledges": [
                {
                    "acknowledgeid": "14443",
                    "userid": "1",
                    "eventid": "1245463",
                    "clock": "1472457281",
                    "message": "problem solved",
                    "action": "6",
                    "old_severity": "0",
                    "new_severity": "0",
                    "suppress_until": "1472511600",
                    "taskid": "0",
                    "maintenanceid": "0"
                }
            ],
            "opdata": "",
            "suppression_data": [
                {
                    "maintenanceid": "15",
                    "suppress_until": "1472511600",
                    "userid": "0"
                }
            ],
            "tags": [
                {
                    "tag": "test-tag",
                    "value": "test-value"
                }
            ],
            "urls": []
        }
    ],
    "id": 1
}
```


Abrufen von Problemen, die von einem angegebenen Benutzer bestätigt wurden

Abrufen von Problemen, die von Benutzer mit ID=10 bestätigt wurden

Anfrage:


```
{
    "jsonrpc": "2.0",
    "method": "problem.get",
```

```
    "params": {
        "output": "extend",
        "action": 2,
        "action_userids": [10],
        "selectAcknowledges": ["userid", "action"],
        "sortfield": ["eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "1248566",
            "source": "0",
            "object": "0",
            "objectid": "15142",
            "clock": "1472457242",
            "ns": "209442442",
            "r_eventid": "1245468",
            "r_clock": "1472457285",
            "r_ns": "125644870",
            "correlationid": "0",
            "userid": "10",
            "name": "Zabbix agent on localhost is unreachable for 5 minutes",
            "acknowledged": "1",
            "severity": "3",
            "cause_eventid": "0",
            "suppressed": "0",
            "acknowledges": [
                {
                    "acknowledgeid": "9",
                    "userid": "10",
                    "action": "2"
                }
            ],
            "opdata": "",
            "urls": []
        }
    ],
    "id": 1
}
```

Siehe auch

- Alarm
- Datenpunkt
- Host
- LLD-Regel
- Auslöser

Quelle

CEvent::get() in *ui/include/classes/api/services/CProblem.php*.

Proxy

Diese Klasse ist für die Arbeit mit Proxys vorgesehen.

Objektreferenzen:

- Proxy

Verfügbare Methoden:

- proxy.create - neue Proxys erstellen
- proxy.delete - Proxys löschen
- proxy.get - Proxys abrufen
- proxy.update - Proxys aktualisieren

Proxy-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `proxy` API.

Proxy

Das Proxy-Objekt verfügt über die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
proxyid	ID	ID des Proxy. Eigenschaftsverhalten: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
name	string	Name des Proxy. Eigenschaftsverhalten: - *erforderlich* für Erstellungsvorgänge
proxy_groupid	ID	ID der Proxy-Gruppe. 0 - Proxy ist keiner Proxy-Gruppe zugewiesen.
local_address	string	Adresse für aktive Agents. IP-Adresse oder DNS-Name, zu dem eine Verbindung hergestellt werden soll. Eigenschaftsverhalten: - *erforderlich*, wenn `proxy_groupid` nicht 0 ist
local_port	string	Lokale Proxy-Portnummer, zu der eine Verbindung hergestellt werden soll. Benutzermakros werden unterstützt. Standard: `10051`. Eigenschaftsverhalten: - *unterstützt*, wenn `proxy_groupid` nicht 0 ist
operating_mode	integer	Typ des Proxy. Mögliche Werte: 0 - aktiver Proxy; 1 - passiver Proxy. Eigenschaftsverhalten: - *erforderlich* für Erstellungsvorgänge
description	text	Beschreibung des Proxy.
lastaccess	timestamp	Zeitpunkt, zu dem der Proxy zuletzt eine Verbindung zum Server hergestellt hat. Eigenschaftsverhalten: - *schreibgeschützt*
address	string	IP-Adresse oder DNS-Name, zu dem eine Verbindung hergestellt werden soll. Benutzermakros werden unterstützt. Eigenschaftsverhalten: - *erforderlich*, wenn der Betriebsmodus des Zabbix Proxy passiv ist

Eigenschaft	Typ	Beschreibung
port	string	Portnummer, zu der eine Verbindung hergestellt werden soll. Benutzermakros werden unterstützt. Standard: 10051. Eigenschaftsverhalten: - *unterstützt*, wenn der Betriebsmodus des Zabbix Proxy passiv ist
allowed_addresses	string	Durch Kommas getrennte IP-Adressen oder DNS-Namen des aktiven Zabbix Proxy.
tls_connect	integer	Verbindungen zum Proxy. Mögliche Werte: 1 - *(Standard)* Keine Verschlüsselung; 2 - PSK; 4 - Zertifikat.
tls_accept	integer	Verbindungen vom Proxy. Mögliche Bitmaskenwerte: 1 - *(Standard)* Keine Verschlüsselung; 2 - PSK; 4 - Zertifikat. Dies ist ein Bitmaskenfeld; jede Summe möglicher Bitmaskenwerte ist zulässig (beispielsweise 6 für PSK und Zertifikat).
tls_issuer	string	Aussteller des Zertifikats.
tls_subject	string	Betreff des Zertifikats.
tls_psk_identity	string	PSK-Identität; darf nur mit einem einzigen PSK verknüpft sein (über Autoregistrierung, Hosts und Proxys hinweg). Fügen Sie keine vertraulichen Informationen in die PSK-Identität ein, da sie unverschlüsselt über das Netzwerk gesendet wird, um dem Empfänger mitzuteilen, welcher PSK verwendet werden soll. Eigenschaftsverhalten: - *nur schreiben* - *erforderlich*, wenn `tls_connect` auf „PSK“ gesetzt ist oder `tls_accept` das Bit „PSK“ enthält
tls_psk	string	Pre-Shared Key (PSK); muss mindestens 32 Hexadezimalziffern enthalten. Eigenschaftsverhalten: - *nur schreiben* - *erforderlich*, wenn `tls_connect` auf „PSK“ gesetzt ist oder `tls_accept` das Bit „PSK“ enthält
custom_timeouts	integer	Gibt an, ob globale Zeitüberschreitungen für Datenpunkte auf Proxy-Ebene überschrieben werden sollen. Mögliche Werte: 0 - *(Standard)* globale Einstellungen verwenden; 1 - Zeitüberschreitungen überschreiben.

Eigenschaft	Typ	Beschreibung
timeout_zabbix_agent	string	Für die Verarbeitung nicht mehr als `timeout_*` Sekunden aufwenden. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300) oder einen Wert mit einem Zeitsuffix (z. B. 30s, 1m). Akzeptiert auch Benutzermakros. Wird nur verwendet, wenn `custom_timeouts` auf „Zeitüberschreitungen überschreiben“ gesetzt ist; andernfalls wird der Wert ignoriert. Möglicher Wertebereich: 1-600s. Eigenschaftsverhalten: - *erforderlich*, wenn `custom_timeouts` auf „Zeitüberschreitungen überschreiben“ gesetzt ist
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		
timeout_telemetry_query		
version	integer	Version des Proxy. Dreiteilige Zabbix-Versionsnummer, bei der für jeden Teil zwei Dezimalziffern verwendet werden, z. B. 60401 für Version 6.4.1, 70002 für Version 7.0.2 usw. 0 - Unbekannte Proxy-Version. Eigenschaftsverhalten: - *schreibgeschützt*
compatibility	integer	Version des Proxy im Vergleich zur Zabbix-Server-Version. Mögliche Werte: 0 - Nicht definiert; 1 - Aktuelle Version (Proxy und Server verfügen über dieselbe Hauptversion); 2 - Veraltete Version (Proxy-Version ist älter als die Server-Version, wird aber teilweise unterstützt); 3 - Nicht unterstützte Version (Proxy-Version ist älter als die vorherige LTS-Version des Servers oder die Hauptversion des Servers ist älter als die Hauptversion des Proxy). Eigenschaftsverhalten: - *schreibgeschützt*
state	integer	Status des Proxy. Mögliche Werte: 0 - Unbekannt; 1 - Offline; 2 - Online. Eigenschaftsverhalten: - *schreibgeschützt*

Application Performance Monitoring (APM)

Das APM-Objekt verfügt über die folgenden Eigenschaften.

Property	Type	Description
data_collection_status	integer	Gibt an, ob die Erfassung von Telemetriedaten auf dem Proxy aktiviert ist. Mögliche Werte: 0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
max_messages_per_second	integer	Maximale Anzahl von Nachrichten pro Sekunde, die der Proxy akzeptiert. Möglicher Wertebereich: 0 (unbegrenzt) – 2147483647. Standard: 0. Verhalten der Eigenschaft: - *unterstützt*, wenn `data_collection_status` auf „Enabled“ gesetzt ist
additional_resource_attributes	array	Array von Ressourcenattributen, die den eingehenden Telemetriedaten hinzugefügt werden. Wenn die eingehenden Daten bereits ein Ressourcenattribut mit demselben Namen enthalten, wird das eingehende Attribut anstelle des hier angegebenen beibehalten. Verhalten der Eigenschaft: - *unterstützt*, wenn `data_collection_status` auf „Enabled“ gesetzt ist

Ressourcenattribute

Das Objekt für Ressourcenattribute verfügt über die folgenden Eigenschaften.

Property	Type	Description
key	string	Name des Ressourcenattributs. Muss für den angegebenen `signal_type` eindeutig sein. Verhalten der Eigenschaft: - *erforderlich*
value	string	Wert des Ressourcenattributs. Kann leer sein.
signal_type	integer	Typ der Telemetriedaten, auf den das Attribut angewendet wird. Mögliche Werte: 0 - Traces; 1 - Metriken; 2 - Logs. Verhalten der Eigenschaft: - *erforderlich*

proxy.create

Beschreibung

`object proxy.create(object/array proxies)`

Diese Methode ermöglicht es Ihnen, neue Proxy zu erstellen.

Note:
Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

(object/array) Zu erstellende Proxys.

Zusätzlich zu den Standard-Proxy-Eigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Typ	Beschreibung
hosts	array	Hosts, die vom Proxy überwacht werden sollen. Wenn ein Host bereits von einem anderen Proxy überwacht wird, wird er dem aktuellen Proxy neu zugewiesen. Für die Hosts darf nur die Eigenschaft `hostid` definiert sein.
apm	object	Einstellungen für das Application Performance Monitoring (APM) des Proxys.

Rückgabewerte

(object) Gibt ein Objekt zurück, das die IDs der erstellten Proxys unter der Eigenschaft `proxyids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Proxys.

Beispiele

Einen aktiven Proxy erstellen

Erstellen Sie einen aktiven Proxy „Active proxy“ und weisen Sie ihm einen Host zur Überwachung zu.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Active proxy",
        "operating_mode": "0",
        "hosts": [
            {
                "hostid": "10279"
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10280"
        ]
    },
    "id": 1
}
```

Einen passiven Proxy erstellen

Erstellen Sie einen passiven Proxy „Passive proxy“ und weisen Sie ihm zwei Hosts zur Überwachung zu.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10284"
        ]
    },
    "id": 1
}
```

Erstellen eines Proxy und Hinzufügen zu einer Proxy-Gruppe

Erstellen Sie einen aktiven Proxy „Active proxy“ und fügen Sie ihn der Proxy-Gruppe mit der ID „1“ hinzu.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Active proxy",
        "proxy_groupid": "1",
        "operating_mode": "0"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "5"
        ]
    },
    "id": 1
}
```

Erstellen eines Proxy mit zertifikatsbasierter Verschlüsselung

Erstellen Sie einen aktiven Proxy mit zertifikatsbasierter Verschlüsselung. Der Proxy ist so konfiguriert, dass er den Host ”10785” überwacht.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Active proxy",
        "operating_mode": "0",
        "hosts": [
```

```
            {
                "hostid": "10785"
            }
        ],
        "tls_accept": "4",
        "tls_issuer": "CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com",
        "tls_subject": "CN=Zabbix server,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "2"
        ]
    },
    "id": 1
}
```

Erstellen eines Proxys mit aktiviertem APM

Erstellen Sie einen aktiven Proxy „Active proxy“ mit aktiviertem APM.
Der Proxy akzeptiert bis zu 1000 Nachrichten pro Sekunde, fügt das Ressourcenattribut `deployment.environment` zu Traces hinzu und fügt das Ressourcenattribut `cloud.region` zu Metriken hinzu.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Active proxy",
        "operating_mode": "0",
        "apm": {
            "data_collection_status": "1",
            "max_messages_per_second": "1000",
            "additional_resource_attributes": [
                {
                    "key": "deployment.environment",
                    "value": "production",
                    "signal_type": "0"
                },
                {
                    "key": "cloud.region",
                    "value": "eu-central-1",
                    "signal_type": "1"
                }
            ]
        }
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "3"
        ]
    },
```

```
    "id": 1
}
```

Siehe auch

- Host
- Proxy-Gruppe

Quelle

CProxy::create() in *ui/include/classes/api/services/CProxy.php*.

proxy.delete

Beschreibung

`object proxy.delete(array proxies)`

Diese Methode ermöglicht es Ihnen, Proxys zu löschen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen für Benutzerrollen widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der zu löschenden Proxys.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Proxys unter der Eigenschaft `proxyids` enthält.

Beispiele

Mehrere Proxys löschen

Löschen Sie zwei Proxys.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.delete",
    "params": [
        "10286",
        "10285"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10286",
            "10285"
        ]
    },
    "id": 1
}
```

Quelle

CProxy::delete() in *ui/include/classes/api/services/CProxy.php*.

proxy.get

Beschreibung

`integer/array proxy.get(object parameters)`

Mit dieser Methode können Sie Proxies anhand der angegebenen Parameter abrufen.

> **Note:**
> Diese Methode steht Benutzern jeden Typs zur Verfügung. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.
 Es werden nur Proxies zurückgegeben, die durch die *Proxy access list* der Benutzergruppen des Benutzers zugelassen sind.

Parameter

`(object)` Parameter zur Definition der gewünschten Ausgabe.

Die Methode unterstützt die folgenden Parameter.

Parameter	Typ	Beschreibung
proxyids	ID/array	Gibt nur Proxys mit den angegebenen IDs zurück.
proxy_groupids	ID/array	Gibt nur Proxys zurück, die zu den angegebenen Proxy-Gruppen gehören.
selectApm	query	Gibt eine `apm`-Eigenschaft mit den Einstellungen für das Application Performance Monitoring (APM) des Proxys zurück.
selectAssignedHosts	query	Gibt eine `assignedHosts`-Eigenschaft mit den dem Proxy zugewiesenen Hosts zurück. Unterstützt `count`.
selectHosts	query	Gibt eine `hosts`-Eigenschaft mit den vom Proxy überwachten Hosts zurück. Unterstützt `count`.
selectProxyGroup	query	Gibt eine `proxyGroup`-Eigenschaft mit dem Proxy-Gruppenobjekt zurück.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `proxyid`, `name`, `operating_mode`.
countOutput	boolean	Diese Parameter werden im Referenzkommentar beschrieben.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Gibt entweder:

- eine Reihe von Objekten zurück;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde zurück.

Beispiele

Alle Proxys abrufen

Rufen Sie alle konfigurierten Proxys und ihre Schnittstellen ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Antwort:


```
{
    "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
}
```


Siehe auch

- Host
- Proxy-Gruppe

Quelle

CProxy::get() in *ui/include/classes/api/services/CProxy.php*.

proxy.update

Beschreibung

`object proxy.update(object/array proxies)`

Diese Methode ermöglicht es Ihnen, vorhandene Proxy zu aktualisieren.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu aktualisierende Proxy-Eigenschaften.

Die Eigenschaft `proxyid` muss für jeden Proxy definiert sein. Alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Zusätzlich zu den Standard-Proxy-Eigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Description
hosts	array	Hosts, die vom Proxy überwacht werden sollen. Wenn ein Host bereits von einem anderen Proxy überwacht wird, wird er dem aktuellen Proxy neu zugewiesen. Für die Hosts darf nur die Eigenschaft `hostid` definiert sein.
apm	object	Einstellungen für das Application Performance Monitoring (APM) des Proxys.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Proxys unter der Eigenschaft `proxyids` enthält.

Beispiele

Von einem Proxy überwachte Hosts ändern

Aktualisieren Sie den Proxy, damit er die beiden angegebenen Hosts überwacht.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.update",
    "params": {
        "proxyid": "10293",
        "hosts": [
            {
                "hostid": "10294"
            },
            {
                "hostid": "10295"
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10293"
        ]
    },
    "id": 1
}
```

Proxy-Status ändern

Ändern Sie den Proxy in einen aktiven Proxy und benennen Sie ihn in „Active proxy“ um.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.update",
    "params": {
        "proxyid": "10293",
        "name": "Active proxy",
        "operating_mode": "0"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10293"
        ]
    },
    "id": 1
}
```

Einen Proxy zu einer Proxy-Gruppe hinzufügen

Aktualisieren Sie den Proxy mit der ID „5“ und fügen Sie ihn der Proxy-Gruppe mit der ID „1“ hinzu.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
```

```
        "proxyid": "5",
        "proxy_groupid": "1",
        "local_address": "127.0.0.1"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "5"
        ]
    },
    "id": 1
}
```

Siehe auch

- Host
- Proxy-Gruppe

Quelle

CProxy::update() in *ui/include/classes/api/services/CProxy.php*.

Proxy-Gruppe

Diese Klasse ist für die Arbeit mit Proxy-Gruppen vorgesehen.

Objektreferenzen:

- Proxy-Gruppe

Verfügbare Methoden:

- proxygroup.create - neue Proxy-Gruppen erstellen
- proxygroup.delete - Proxy-Gruppen löschen
- proxygroup.get - Proxy-Gruppen abrufen
- proxygroup.update - Proxy-Gruppen aktualisieren

Proxy-Gruppenobjekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `proxygroup` API.

Proxy-Gruppe

Das Objekt der Proxy-Gruppe verfügt über die folgenden Eigenschaften.

Property	Type	Description
proxy_groupid	ID	ID der Proxy-Gruppe. Verhalten der Eigenschaft: - *read-only* - *required* für Aktualisierungsvorgänge
name	string	Name der Proxy-Gruppe. Verhalten der Eigenschaft: - *required* für Erstellungsvorgänge
description	text	Beschreibung der Proxy-Gruppe.

Property	Type	Description
failover_delay	string	Zeitraum, während dessen ein Proxy in der Proxy-Gruppe mit dem Zabbix-Server kommunizieren muss, um als online zu gelten. Zeitsuffixe werden unterstützt (z. B. 30s, 1m). Benutzermakros werden unterstützt. Mögliche Werte: 10s-15m. Standard: 1m.
min_online	string	Mindestanzahl der Online-Proxys, die erforderlich ist, damit die Proxy-Gruppe online bleibt. Benutzermakros werden unterstützt. Möglicher Wertebereich: 1-1000. Standard: 1.
state	integer	Status der Proxy-Gruppe. Mögliche Werte: 0 - Unbekannt; 1 - Offline; 2 - Wird wiederhergestellt; 3 - Online; 4 - Wird schlechter. Verhalten der Eigenschaft: *- read-only*

proxygroup.create

Beschreibung

`object proxygroup.create(object/array proxyGroups)`

Diese Methode ermöglicht es Ihnen, neue Proxy-Gruppen zu erstellen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellende Proxy-Gruppen.

Die Methode akzeptiert Proxy-Gruppen mit den Standard-Proxy-Gruppeneigenschaften.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Proxygruppen unter der Eigenschaft `proxy_groupids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Proxygruppen.

Beispiele

Eine Proxy-Gruppe erstellen

Erstellen Sie eine Proxy-Gruppe mit benutzerdefinierten Einstellungen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.create",
    "params": {
        "name": "Proxy group",
```

```
        "failover_delay": "5m",
        "min_online": "10"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxy_groupids": [
            "5"
        ]
    },
    "id": 1
}
```

Quelle

CProxyGroup::create() in *ui/include/classes/api/services/CProxyGroup.php*.

proxygroup.delete

Beschreibung

`object proxygroup.delete(array proxyGroupIds)`

Diese Methode ermöglicht es Ihnen, Proxy-Gruppen zu löschen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der zu löschenden Proxy-Gruppen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Proxy-Gruppen unter der Eigenschaft `proxy_groupids` enthält.

Beispiele

Mehrere Proxy-Gruppen löschen

Löschen Sie zwei Proxy-Gruppen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.delete",
    "params": [
        "5",
        "10"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxy_groupids": [
            "5",
            "10"
        ]
```

```
    },
    "id": 1
}
```

Quelle

CProxyGroup::delete() in *ui/include/classes/api/services/CProxyGroup.php*.

proxygroup.get

Beschreibung

`integer/array proxygroup.get(object parameters)`

Mit dieser Methode können Sie Proxy-Gruppen entsprechend den angegebenen Parametern abrufen.

> **Note:**
> Diese Methode steht Benutzern jeden Typs zur Verfügung. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen. Es werden nur Proxy-Gruppen zurückgegeben, die durch die *Proxy access list* der Benutzergruppen des Benutzers zugelassen sind.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
proxy_groupids	ID/array	Gibt nur Proxy-Gruppen mit den angegebenen IDs zurück.
proxyids	ID/array	Gibt nur Proxy-Gruppen zurück, die die angegebenen Proxys enthalten.
selectProxies	query	Gibt eine `proxies`-Eigenschaft mit den Proxys zurück, die zur Proxy-Gruppe gehören. Unterstützt `count`.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `proxy_groupid`, `name`.
countOutput	boolean	Diese Parameter sind in der Referenzkommentierung beschrieben.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Gibt entweder:

- eine Reihe von Objekten zurück;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde zurück.

Beispiele

Alle Proxy-Gruppen abrufen

Rufen Sie alle konfigurierten Proxy-Gruppen mit Proxys ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.get",
    "params": {
        "output": "extend",
        "selectProxies": ["proxyid", "name"]
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```

Siehe auch

- Proxy

Quelle

CProxyGroup::get() in *ui/include/classes/api/services/CProxyGroup.php*.

proxygroup.update

Beschreibung

`object proxygroup.update(object/array proxyGroups)`

Diese Methode ermöglicht es Ihnen, vorhandene Proxygruppen zu aktualisieren.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu aktualisierende Proxy-Gruppeneigenschaften.

Die Eigenschaft `proxy_groupid` muss für jede Proxy-Gruppe definiert sein, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Die Methode akzeptiert Proxy-Gruppen mit den Standard-Proxy-Gruppeneigenschaften.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Proxy-Gruppen in der Eigenschaft `proxy_groupids` enthält.

Beispiele

Mindestanzahl online befindlicher Proxys ändern

Ändern Sie die Mindestanzahl online befindlicher Proxys, die erforderlich ist, damit die Gruppe als online gilt.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.update",
    "params": {
        "proxy_groupid": "5",
        "min_online": "3"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxy_groupids": [
            "5"
        ]
    },
    "id": 1
}
```

Quelle

CProxyGroup::update() in *ui/include/classes/api/services/CProxyGroup.php*.

Regulärer Ausdruck

Diese Klasse ist für die Arbeit mit globalen regulären Ausdrücken vorgesehen.

Objektreferenzen:

- Regulärer Ausdruck
- Ausdrücke

Verfügbare Methoden:

- regexp.create - neue reguläre Ausdrücke erstellen
- regexp.delete - reguläre Ausdrücke löschen
- regexp.get - reguläre Ausdrücke abrufen
- regexp.update - reguläre Ausdrücke aktualisieren

Regex-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `regexp` API.

Regulärer Ausdruck

Das globale Objekt für reguläre Ausdrücke hat die folgenden Eigenschaften.

Property	Type	Description
regexpid	ID	ID des regulären Ausdrucks. Property behavior: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge.
name	string	Name des regulären Ausdrucks. Property behavior: - *erforderlich* für Erstellungsoperationen.
description	text	Beschreibung des regulären Ausdrucks.
test_string	string	Testzeichenfolge.

Ausdrücke

Das Objekt expressions hat die folgenden Eigenschaften.

Property	Type	Description
expression	string	Regulärer Ausdruck. Property behavior: - *required*
expression_type	integer	Typ des regulären Ausdrucks. Mögliche Werte: 0 - Enthält Zeichenfolge; 1 - Enthält eine beliebige Teilzeichenfolge aus der Liste; 2 - Enthält keine Zeichenfolge; 3 - Entspricht regulärem Ausdruck; 4 - Entspricht nicht regulärem Ausdruck. Property behavior: - *required*
exp_delimiter	string	Trennzeichen des Ausdrucks. Standardwert: `","`. Mögliche Werte: `","` oder `"."` oder `"/"`. Property behavior: - *supported* if `expression_type` is set to *Contains any substring from list*.

Property	Type	Description
case_sensitive	integer	Groß-/Kleinschreibung beachten. Standardwert: 0. Mögliche Werte: 0 - Groß-/Kleinschreibung wird nicht beachtet; 1 - Groß-/Kleinschreibung wird beachtet.

regexp.create

Beschreibung

`object regexp.create(object/array regularExpressions)`

Diese Methode ermöglicht es Ihnen, neue globale reguläre Ausdrücke zu erstellen.

> **Note:**
> Diese Methode ist nur für Benutzertypen vom Typ *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Reguläre Ausdrücke, die erstellt werden sollen.

Zusätzlich zu den Standardeigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
expressions	array	Optionen für Ausdrücke. Parameterverhalten: *- erforderlich*

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten regulären Ausdrücke in der Eigenschaft `regexpids` enthält.

Beispiele

Erstellen eines neuen globalen regulären Ausdrucks.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.create",
    "params": {
        "name": "Storage devices for SNMP discovery",
        "description": "Global regular expression used to discover and identify memory resources on the mo
        "test_string": "/boot",
        "expressions": [
            {
                "expression": "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$"
                "expression_type": "4",
                "case_sensitive": "1"
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "regexpids": [
            "9"
        ]
    },
    "id": 1
}
```

Quelle

CRegexp::create() in *ui/include/classes/api/services/CRegexp.php*.

regexp.delete

Beschreibung

`object regexp.delete(array regexpids)`

Diese Methode ermöglicht es Ihnen, globale reguläre Ausdrücke zu löschen.

> **Note:**
> Diese Methode ist nur für Benutzertypen vom Typ *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der zu löschenden regulären Ausdrücke.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten regulären Ausdrücke in der Eigenschaft `regexpids` enthält.

Beispiele

Mehrere globale reguläre Ausdrücke löschen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.delete",
    "params": [
        "16",
        "17"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "regexpids": [
            "16",
            "17"
        ]
    },
    "id": 1
}
```

Quelle

CRegexp::delete() in *ui/include/classes/api/services/CRegexp.php*.

regexp.get

Beschreibung

`integer/array regexp.get(object parameters)`

Die Methode ermöglicht es Ihnen, globale reguläre Ausdrücke anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist nur für *Super Admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Typ	Beschreibung
regexpids	ID/array	Gibt nur reguläre Ausdrücke mit den angegebenen IDs zurück.
selectExpressions	query	Gibt eine Eigenschaft `expressions` zurück.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `regexpid`, `name`.
countOutput	boolean	Diese Parameter werden in der Referenzkommentierung beschrieben.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Gibt entweder Folgendes zurück:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Abrufen globaler regulärer Ausdrücke.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.get",
    "params": {
        "output": ["regexpid","name","description"],
        "selectExpressions": ["expression","expression_type"],
        "regexpids": [1,2],
        "preservekeys": true
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "1": {
            "regexpid": "1",
            "name": "File systems for discovery",
            "description": "",
            "expressions": [
                {
                    "expression": "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|n
                    "expression_type": "3"
                }
            ]
        },
        "2": {
            "regexpid": "2",
            "name": "Network interfaces for discovery",
            "description": "",
            "expressions": [
                {
                    "expression": "^Software Loopback Interface",
                    "expression_type": "4"
                },
                {
                    "expression": "^(In)?[Ll]oop[Bb]ack[0-9._]*$",
                    "expression_type": "4"
                },
                {
                    "expression": "^NULL[0-9.]*$",
                    "expression_type": "4"
                },
                {
                    "expression": "^[Ll]o[0-9.]*$",
                    "expression_type": "4"
                },
                {
                    "expression": "^[Ss]ystem$",
                    "expression_type": "4"
                },
                {
                    "expression": "^Nu[0-9.]*$",
                    "expression_type": "4"
                }
            ]
        }
    },
    "id": 1
}
```


Quelle

CRegexp::get() in *ui/include/classes/api/services/CRegexp.php*.

regexp.update

Beschreibung

`object regexp.update(object/array regularExpressions)`

Diese Methode ermöglicht es Ihnen, vorhandene globale reguläre Ausdrücke zu aktualisieren.

> **Note:**
> Diese Methode ist nur für Benutzertypen vom Typ *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

(object/array) Zu aktualisierende Eigenschaften des regulären Ausdrucks.

Die Eigenschaft regexpid muss für jedes Objekt definiert werden, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle übrigen bleiben unverändert.

Zusätzlich zu den Standardeigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Description
expressions	array	Optionen für Ausdrücke.

Rückgabewerte

(object) Gibt ein Objekt zurück, das die IDs der aktualisierten regulären Ausdrücke in der Eigenschaft regexpids enthält.

Beispiele

Aktualisierung des globalen regulären Ausdrucks für die Erkennung von Dateisystemen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.update",
    "params": {
        "regexpid": "1",
        "name": "Dateisysteme für die Erkennung",
        "description": "Globaler regulärer Ausdruck zum Erkennen und Identifizieren von Dateisystemen auf
        "test_string": "",
        "expressions": [
            {
                "expression": "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|zfs)$
                "expression_type": "3",
                "case_sensitive": "0"
            },
            {
                "expression": "^(ntfs|fat32|fat16)$",
                "expression_type": "3",
                "case_sensitive": "0"
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "regexpids": [
            "1"
        ]
    },
    "id": 1
}
```

Quelle

CRegexp::update() in *ui/include/classes/api/services/CRegexp.php*.

Rolle

Diese Klasse ist für die Arbeit mit Benutzerrollen vorgesehen.

Objektreferenzen:

- Role

- Rollenregeln
  - UI-Element
  - Service
  - Service-Tag
  - Modul
  - Aktion

Verfügbare Methoden:

- role.create - neue Benutzerrollen erstellen
- role.delete - Benutzerrollen löschen
- role.get - Benutzerrollen abrufen
- role.update - Benutzerrollen aktualisieren

Rollen-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `role` API.

Rolle

Das Rollenobjekt hat die folgenden Eigenschaften:

Eigenschaft	Typ	Beschreibung
roleid	ID	ID der Rolle. Eigenschaftsverhalten: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
name	string	Name der Rolle. Eigenschaftsverhalten: - *erforderlich* für Erstellungsvorgänge
type	integer	Benutzertyp. Mögliche Werte: 1 - *(Standard)* Benutzer; 2 - Admin; 3 - Super-Admin. Eigenschaftsverhalten: - *erforderlich* für Erstellungsvorgänge
readonly	integer	Gibt an, ob die Rolle schreibgeschützt ist. Mögliche Werte: 0 - *(Standard)* Nein; 1 - Ja. Eigenschaftsverhalten: - *schreibgeschützt*

Rollenregeln

Das Objekt für Rollenregeln verfügt über die folgenden Eigenschaften:

Eigenschaft	Typ	Beschreibung
ui	array	Array der UI-Element-Objekte.
ui.default_access	integer	Gibt an, ob der Zugriff auf neue UI-Elemente aktiviert ist. Mögliche Werte: 0 - Deaktiviert; 1 - *(Standard)* Aktiviert.

Eigenschaft	Typ	Beschreibung
profile.redirect.enforce	integer	Gibt an, ob die Weiterleitungs-URL auf Rollenebene nach der Authentifizierung erzwungen wird und vom Benutzer nicht überschrieben werden kann. Mögliche Werte: 0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
profile.redirect.url	string	URL, die für die Weiterleitung des Benutzers nach der Authentifizierung verwendet wird.
services.read.mode	integer	Schreibgeschützter Zugriff auf Services. Mögliche Werte: 0 - Schreibgeschützter Zugriff auf die Services, die durch die Eigenschaft `services.read.list` angegeben oder durch `services.read.tag` abgeglichen werden; 1 - *(Standard)* Schreibgeschützter Zugriff auf alle Services.
services.read.list	array	Array von Service-Objekten. Auf die angegebenen Services einschließlich untergeordneter Services erhält die Benutzerrolle schreibgeschützten Zugriff. Der schreibgeschützte Zugriff überschreibt keinen Lese-/Schreibzugriff auf die Services. Verhalten der Eigenschaft: - *unterstützt*, wenn `services.read.mode` auf "0" gesetzt ist.
services.read.tag	object	Array von Service-Tag-Objekten. Auf die durch Tags abgeglichenen Services einschließlich untergeordneter Services erhält die Benutzerrolle schreibgeschützten Zugriff. Der schreibgeschützte Zugriff überschreibt keinen Lese-/Schreibzugriff auf die Services. Verhalten der Eigenschaft: - *unterstützt*, wenn `services.read.mode` auf "0" gesetzt ist.
services.write.mode	integer	Lese-/Schreibzugriff auf Services. Mögliche Werte: 0 - *(Standard)* Lese-/Schreibzugriff auf die Services, die durch die Eigenschaft `services.write.list` angegeben oder durch `services.write.tag` abgeglichen werden; 1 - Lese-/Schreibzugriff auf alle Services.
services.write.list	array	Array von Service-Objekten. Auf die angegebenen Services einschließlich untergeordneter Services erhält die Benutzerrolle Lese-/Schreibzugriff. Der Lese-/Schreibzugriff überschreibt den schreibgeschützten Zugriff auf die Services. Verhalten der Eigenschaft: - *unterstützt*, wenn `services.write.mode` auf "0" gesetzt ist.
services.write.tag	object	Array von Service-Tag-Objekten. Auf die durch Tags abgeglichenen Services einschließlich untergeordneter Services erhält die Benutzerrolle Lese-/Schreibzugriff. Der Lese-/Schreibzugriff überschreibt den schreibgeschützten Zugriff auf die Services. Verhalten der Eigenschaft: - *unterstützt*, wenn `services.write.mode` auf "0" gesetzt ist.
modules	array	Array der Modul-Objekte.

Eigenschaft	Typ	Beschreibung
modules.default_access	integer	Gibt an, ob der Zugriff auf neue Module aktiviert ist. Mögliche Werte: 0 - Deaktiviert; 1 - *(Standard)* Aktiviert.
api.access	integer	Gibt an, ob der Zugriff auf die API aktiviert ist. Mögliche Werte: 0 - *(Standard)* Deaktiviert; 1 - Aktiviert.
api.mode	integer	Modus für die Behandlung der in der Eigenschaft `api` aufgeführten API-Methoden. Mögliche Werte: 0 - *(Standard)* Ausschlussliste; 1 - Einschlussliste. Verhalten der Eigenschaft: - *unterstützt*, wenn `api.access` auf "1" gesetzt ist.
api	array	Array von API-Methoden. Platzhalter werden unterstützt, zum Beispiel: `*` (jede Methode), `dashboard.*`, `*.export`. Wenn keine Methoden angegeben sind, wird der Zugriff auf alle API-Methoden verweigert, wenn `api.mode` auf "1" (Einschlussliste) gesetzt ist, und gewährt, wenn `api.mode` auf "0" (Ausschlussliste) gesetzt ist. Verhalten der Eigenschaft: - *unterstützt*, wenn `api.access` auf "1" gesetzt ist.
actions	array	Array der Aktion-Objekte.
actions.default_access	integer	Gibt an, ob der Zugriff auf neue Aktionen aktiviert ist. Mögliche Werte: 0 - Deaktiviert; 1 - *(Standard)* Aktiviert.
devices.access	integer	Gibt an, ob der Zugriff auf die eigenen Mobilgeräte des Benutzers zur Verwendung mit der Zabbix-Mobile-App aktiviert ist. Mögliche Werte: 0 - (*Standard für Benutzer und Administratoren*) Deaktiviert; 1 - (*Standard für Super-Administratoren*) Aktiviert. Verhalten der Eigenschaft: - *unterstützt*, wenn `EnableMobileDevices` aktiviert ist
devices.actions	array	Array von Geräteaktionen-Objekten. Verhalten der Eigenschaft: - *unterstützt*, wenn `devices.access` auf "1" gesetzt ist
devices.actions.default_access	integer	Gibt an, ob der Zugriff auf neue Geräteaktionen aktiviert ist. Mögliche Werte: 0 - (*Standard für Benutzer und Administratoren*) Deaktiviert; 1 - (*Standard für Super-Administratoren*) Aktiviert. Verhalten der Eigenschaft: - *unterstützt*, wenn `EnableMobileDevices` aktiviert ist

UI-Element

Das UI-Element-Objekt verfügt über die folgenden Eigenschaften:

Property	Type	Description
name	string	Name des UI-Elements. Mögliche Werte, wenn `type` des Role-Objekts auf „User“, „Admin“ oder „Super admin“ gesetzt ist: `monitoring.dashboard` - *Dashboards*; `monitoring.problems` - *Monitoring > Problems*; `monitoring.hosts` - *Monitoring > Hosts*; `monitoring.latest_data` - *Monitoring > Latest data*; `monitoring.maps` - *Monitoring > Maps*; `services.services` - *Services > Services*; `services.sla_report` - *Services > SLA report*; `inventory.overview` - *Inventory > Overview*; `inventory.hosts` - *Inventory > Hosts*; `reports.availability_report` - *Reports > Availability report*; `reports.top_triggers` - *Reports > Triggers top 100*. Mögliche Werte, wenn `type` des Role-Objekts auf „Admin“ oder „Super admin“ gesetzt ist: `monitoring.discovery` - *Monitoring > Discovery*; `services.sla` - *Services > SLA*; `reports.scheduled_reports` - *Reports > Scheduled reports*; `reports.notifications` - *Reports > Notifications*; `configuration.template_groups` - *Data collection > Template groups*; `configuration.host_groups` - *Data collection > Host groups*; `configuration.templates` - *Data collection > Templates*; `configuration.hosts` - *Data collection > Hosts*; `configuration.maintenance` - *Data collection > Maintenance*; `configuration.discovery` - *Data collection > Discovery*; `configuration.trigger_actions` - *Alerts > Actions > Trigger actions*; `configuration.service_actions` - *Alerts > Actions > Service actions*; `configuration.discovery_actions` - *Alerts > Actions > Discovery actions*; `configuration.autoregistration_actions` - *Alerts > Actions > Autoregistration actions*; `configuration.internal_actions` - *Alerts > Actions > Internal actions*. Mögliche Werte, wenn `type` des Role-Objekts auf „Super admin“ gesetzt ist: `reports.system_info` - *Reports > System information*; `reports.audit` - *Reports > Audit log*; `reports.action_log` - *Reports > Action log*; `configuration.event_correlation` - *Data collection > Event correlation*; `administration.media_types` - *Alerts > Media types*; `administration.scripts` - *Alerts > Scripts*; `administration.user_groups` - *Users > User groups*; `administration.user_roles` - *Users > User roles*; `administration.users` - *Users > Users*; `administration.api_tokens` - *Users > API tokens*; `administration.authentication` - *Users > Authentication*; `administration.linked_devices` - *Users > Devices*; `administration.general` - *Administration > General*; `administration.audit_log` - *Administration > Audit log*; `administration.housekeeping` - *Administration > Housekeeping*; `administration.proxy_groups` - *Administration > Proxy groups*; `administration.proxies` - *Administration > Proxies*; `administration.macros` - *Administration > Macros*; `administration.queue` - *Administration > Queue*. Property behavior: *- required*

Property	Type	Description
status	integer	Gibt an, ob der Zugriff auf das UI-Element aktiviert ist. Mögliche Werte: 0 - Deaktiviert; 1 - *(default)* Aktiviert.

Service

Eigenschaft	Typ	Beschreibung
serviceid	ID	ID des Service. Verhalten der Eigenschaft: - *erforderlich*

Service-Tag

Eigenschaft	Typ	Beschreibung
tag	string	Tag-Name. Wenn eine leere Zeichenfolge angegeben wird, wird das Service-Tag nicht für den Service-Abgleich verwendet. Verhalten der Eigenschaft: - *erforderlich*
value	string	Tag-Wert. Wenn kein Wert oder eine leere Zeichenfolge angegeben wird, wird nur der Tag-Name für den Service-Abgleich verwendet.

Modul

Das Modulobjekt hat die folgenden Eigenschaften:

Eigenschaft	Typ	Beschreibung
moduleid	ID	ID des Moduls. Eigenschaftsverhalten: - *erforderlich*
status	integer	Gibt an, ob der Zugriff auf das Modul aktiviert ist. Mögliche Werte: 0 - Deaktiviert; 1 - *(Standard)* Aktiviert.

Aktion

Das Aktionsobjekt verfügt über die folgenden Eigenschaften:

Property	Type	Description
name	string	Name der Aktion. Mögliche Werte, wenn `type` des Role-Objekts auf „User“, „Admin“ oder „Super admin“ gesetzt ist: `edit_dashboards` - Dashboards erstellen und bearbeiten; `edit_maps` - Karten erstellen und bearbeiten; `add_problem_comments` - Problemkommentare hinzufügen; `change_severity` - Problemschweregrad ändern; `acknowledge_problems` - Probleme bestätigen; `suppress_problems` - Probleme unterdrücken; `close_problems` - Probleme schließen; `execute_scripts` - Skripte ausführen; `manage_api_tokens` - API-Token verwalten; `change_problem_ranking` - Die Problemrangfolge von Ursache zu Symptom und umgekehrt ändern; `edit_own_media` - Benutzern erlauben, ihre eigenen Medien zu erstellen und zu bearbeiten. Mögliche Werte, wenn `type` des Role-Objekts auf „Admin“ oder „Super admin“ gesetzt ist: `edit_maintenance` - Wartungen erstellen und bearbeiten; `manage_scheduled_reports` - Geplante Berichte verwalten, `manage_sla` - SLA verwalten, `select_server_for_monitoring` - Benutzern erlauben, Hosts zu erstellen, die vom Server überwacht werden, sowie Hosts, die von den vom Server ausgeführten Netzwerkerkennungsregeln erkannt werden. Mögliche Werte, wenn `type` des Role-Objekts auf „User“ oder „Admin“ gesetzt ist: `invoke_execute_now` - Benutzern mit ausschließlich Leseberechtigungen für einen Host erlauben, Datenpunktprüfungen auszuführen. Mögliche Werte, wenn `type` des Role-Objekts auf „Super admin“ gesetzt ist: `edit_user_media` - Benutzern erlauben, Medien für andere Benutzer zu erstellen und zu bearbeiten. Verhalten von Eigenschaften: - *required*
status	integer	Gibt an, ob der Zugriff zum Ausführen der Aktion aktiviert ist. Mögliche Werte: 0 - Deaktiviert; 1 - *(default)* Aktiviert.

Geräteaktion

Das Geräteaktionsobjekt verfügt über die folgenden Eigenschaften:

Property	Type	Description
name	string	Name der Geräteaktion. Mögliche Werte, wenn `type` des Role-Objekts auf „User“, „Admin“ oder „Super admin“ gesetzt ist: `manage_own` – Benutzern erlauben, ihre eigenen Mobilgeräte hinzuzufügen oder zu entfernen. Mögliche Werte, wenn `type` des Role-Objekts auf „Super admin“ gesetzt ist: `manage_user` – Benutzern erlauben, Mobilgeräte für alle Benutzer hinzuzufügen oder zu entfernen. Eigenschaftsverhalten: - *required*
status	integer	Gibt an, ob der Zugriff zum Ausführen der Aktion aktiviert ist. Mögliche Werte: 0 – Deaktiviert; 1 – (*default*) Aktiviert.

role.create

Beschreibung

`object role.create(object/array roles)`

Diese Methode ermöglicht es Ihnen, neue Rollen zu erstellen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellende Rollen.

Zusätzlich zu den Standard-Rolleneigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
rules	array	Für die Rolle zu erstellende Rollenregeln.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Rollen unter der Eigenschaft `roleids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Rollen.

Beispiele

Erstellen einer Rolle

Erstellen Sie eine Rolle vom Typ „Benutzer“ mit verweigertem Zugriff auf zwei UI-Elemente.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "5"
        ]
    },
    "id": 1
}
```

Siehe auch

- Rollenregeln
- UI-Element
- Modul
- Aktion

Quelle

CRole::create() in *ui/include/classes/api/services/CRole.php*.

role.delete

Beschreibung

`object role.delete(array roleids)`

Diese Methode ermöglicht es Ihnen, Rollen zu löschen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der zu löschenden Rollen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Rollen unter der Eigenschaft `roleids` enthält.

Beispiele

Mehrere Benutzerrollen löschen

Löschen Sie zwei Benutzerrollen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "role.delete",
    "params": [
        "4",
```

```
        "5"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "4",
            "5"
        ]
    },
    "id": 1
}
```

Quelle

CRole::delete() in *ui/include/classes/api/services/CRole.php*.

role.get

Beschreibung

`integer/array role.get(object parameters)`

Die Methode ermöglicht es Ihnen, Rollen anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter zur Definition der gewünschten Ausgabe.

Die Methode unterstützt die folgenden Parameter.

Parameter	Typ	Beschreibung
roleids	ID/array	Gibt nur Rollen mit den angegebenen IDs zurück.
current	boolean	Wenn auf `true` gesetzt, wird nur die Rolle des aktuellen Benutzers zurückgegeben.
selectRules	query	Gibt eine Eigenschaft `rules` mit den Rollenregeln zurück.
selectUsers	query	Gibt eine Eigenschaft `users` mit den Benutzern zurück, denen die Rolle zugewiesen ist. Weitere Informationen zu Einschränkungen basierend auf dem Benutzertyp finden Sie unter `user.get`.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `roleid`, `name`.
countOutput	boolean	Diese Parameter werden im Referenzkommentar beschrieben.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	

Parameter	Typ	Beschreibung
startSearch	boolean	

Rückgabewerte

`(integer/array)` Gibt entweder Folgendes zurück:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Rollendaten abrufen

Rufen Sie die Rollendaten der Rolle „Super admin role“ und deren Zugriffsregeln ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "role.get",
    "params": {
        "output": "extend",
        "selectRules": "extend",
        "roleids": "3"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "roleid": "3",
            "name": "Super admin role",
            "type": "3",
            "readonly": "1",
            "rules": {
                "ui": [
                    {
                        "name": "monitoring.dashboard",
                        "status": "1"
                    },
                    {
                        "name": "monitoring.problems",
                        "status": "1"
                    },
                    {
                        "name": "monitoring.hosts",
                        "status": "1"
                    },
                    {
                        "name": "monitoring.latest_data",
                        "status": "1"
                    },
                    {
                        "name": "monitoring.maps",
                        "status": "1"
                    },
                    {
                        "name": "services.services",
                        "status": "1"
                    },
                    {
```

```
            "name": "services.sla_report",
            "status": "1"
        },
        {
            "name": "inventory.overview",
            "status": "1"
        },
        {
            "name": "inventory.hosts",
            "status": "1"
        },
        {
            "name": "reports.availability_report",
            "status": "1"
        },
        {
            "name": "reports.top_triggers",
            "status": "1"
        },
        {
            "name": "monitoring.discovery",
            "status": "1"
        },
        {
            "name": "services.sla",
            "status": "1"
        },
        {
            "name": "reports.scheduled_reports",
            "status": "1"
        },
        {
            "name": "reports.notifications",
            "status": "1"
        },
        {
            "name": "configuration.template_groups",
            "status": "1"
        },
        {
            "name": "configuration.host_groups",
            "status": "1"
        },
        {
            "name": "configuration.templates",
            "status": "1"
        },
        {
            "name": "configuration.hosts",
            "status": "1"
        },
        {
            "name": "configuration.maintenance",
            "status": "1"
        },
        {
            "name": "configuration.discovery",
            "status": "1"
        },
        {
            "name": "configuration.trigger_actions",
            "status": "1"
```

```
            },
            {
                "name": "configuration.service_actions",
                "status": "1"
            },
            {
                "name": "configuration.discovery_actions",
                "status": "1"
            },
            {
                "name": "configuration.autoregistration_actions",
                "status": "1"
            },
            {
                "name": "configuration.internal_actions",
                "status": "1"
            },
            {
                "name": "reports.system_info",
                "status": "1"
            },
            {
                "name": "reports.audit",
                "status": "1"
            },
            {
                "name": "reports.action_log",
                "status": "1"
            },
            {
                "name": "configuration.event_correlation",
                "status": "1"
            },
            {
                "name": "administration.media_types",
                "status": "1"
            },
            {
                "name": "administration.scripts",
                "status": "1"
            },
            {
                "name": "administration.user_groups",
                "status": "1"
            },
            {
                "name": "administration.user_roles",
                "status": "1"
            },
            {
                "name": "administration.users",
                "status": "1"
            },
            {
                "name": "administration.api_tokens",
                "status": "1"
            },
            {
                "name": "administration.authentication",
                "status": "1"
            },
            {
```

```
                    "name": "administration.general",
                    "status": "1"
                },
                {
                    "name": "administration.audit_log",
                    "status": "1"
                },
                {
                    "name": "administration.housekeeping",
                    "status": "1"
                },
                {
                    "name": "administration.proxy_groups",
                    "status": "1"
                },
                {
                    "name": "administration.proxies",
                    "status": "1"
                },
                {
                    "name": "administration.macros",
                    "status": "1"
                },
                {
                    "name": "administration.queue",
                    "status": "1"
                }
            ],
            "ui.default_access": "1",
            "profile.redirect.enforce": "0",
            "profile.redirect.url": "",
            "services.read.mode": "1",
            "services.read.list": [],
            "services.read.tag": {
                "tag": "",
                "value": ""
            },
            "services.write.mode": "1",
            "services.write.list": [],
            "services.write.tag": {
                "tag": "",
                "value": ""
            },
            "modules": [
                {
                    "moduleid": 1,
                    "status": "1"
                },
                {
                    "moduleid": 2,
                    "status": "1"
                },
                {
                    "moduleid": 3,
                    "status": "1"
                },
                {
                    "moduleid": 4,
                    "status": "1"
                },
                {
                    "moduleid": 5,
```

```
            "status": "1"
        },
        {
            "moduleid": 6,
            "status": "1"
        },
        {
            "moduleid": 7,
            "status": "1"
        },
        {
            "moduleid": 8,
            "status": "1"
        },
        {
            "moduleid": 9,
            "status": "1"
        },
        {
            "moduleid": 10,
            "status": "1"
        },
        {
            "moduleid": 11,
            "status": "1"
        },
        {
            "moduleid": 12,
            "status": "1"
        },
        {
            "moduleid": 13,
            "status": "1"
        },
        {
            "moduleid": 14,
            "status": "1"
        },
        {
            "moduleid": 15,
            "status": "1"
        },
        {
            "moduleid": 16,
            "status": "1"
        },
        {
            "moduleid": 17,
            "status": "1"
        },
        {
            "moduleid": 18,
            "status": "1"
        },
        {
            "moduleid": 19,
            "status": "1"
        },
        {
            "moduleid": 20,
            "status": "1"
        },
```

```
            {
                "moduleid": 21,
                "status": "1"
            },
            {
                "moduleid": 22,
                "status": "1"
            },
            {
                "moduleid": 23,
                "status": "1"
            },
            {
                "moduleid": 24,
                "status": "1"
            },
            {
                "moduleid": 25,
                "status": "1"
            },
            {
                "moduleid": 26,
                "status": "1"
            },
            {
                "moduleid": 27,
                "status": "1"
            },
            {
                "moduleid": 28,
                "status": "1"
            },
            {
                "moduleid": 29,
                "status": "1"
            },
            {
                "moduleid": 30,
                "status": "1"
            },
            {
                "moduleid": 31,
                "status": "1"
            },
            {
                "moduleid": 32,
                "status": "1"
            },
            {
                "moduleid": 33,
                "status": "1"
            }
        ],
        "modules.default_access": "1",
        "api.access": "1",
        "api.mode": "0",
        "api": [],
        "actions": [
            {
                "name": "edit_dashboards",
                "status": "1"
            },
```

```
            {
                "name": "edit_maps",
                "status": "1"
            },
            {
                "name": "acknowledge_problems",
                "status": "1"
            },
            {
                "name": "suppress_problems",
                "status": "1"
            },
            {
                "name": "close_problems",
                "status": "1"
            },
            {
                "name": "change_severity",
                "status": "1"
            },
            {
                "name": "add_problem_comments",
                "status": "1"
            },
            {
                "name": "execute_scripts",
                "status": "1"
            },
            {
                "name": "manage_api_tokens",
                "status": "1"
            },
            {
                "name": "edit_maintenance",
                "status": "1"
            },
            {
                "name": "manage_scheduled_reports",
                "status": "1"
            },
            {
                "name": "manage_sla",
                "status": "1"
            },
            {
                "name": "invoke_execute_now",
                "status": "1"
            },
            {
                "name": "change_problem_ranking",
                "status": "1"
            },
            {
                "name": "edit_own_media",
                "status": "1"
            },
            {
                "name": "edit_user_media",
                "status": "1"
            }
        ],
        "actions.default_access": "1"
```

```
            }
        }
    ],
    "id": 1
}
```

Siehe auch

- Rollenregeln
- Benutzer

Quelle

CRole::get() in *ui/include/classes/api/services/CRole.php*.

role.update

Beschreibung

`object role.update(object/array roles)`

Diese Methode ermöglicht es Ihnen, vorhandene Rollen zu aktualisieren.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu aktualisierende Rolleneigenschaften.

Die Eigenschaft `roleid` muss für jede Rolle definiert werden, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Zusätzlich zu den Standard-Rolleneigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
rules	array	Zugriffs-Regeln, die für die Rolle aktualisiert werden sollen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Rollen unter der Eigenschaft `roleids` enthält.

Beispiele

Deaktivieren der Möglichkeit, Skripte auszuführen

Aktualisieren Sie die Rolle mit der ID „5“ und deaktivieren Sie die Möglichkeit, Skripte auszuführen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "role.update",
    "params": [
        {
            "roleid": "5",
            "rules": {
                "actions": [
                    {
                        "name": "execute_scripts",
                        "status": "0"
                    }
                ]
            }
        }
    ],
```

```
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "5"
        ]
    },
    "id": 1
}
```

Zugriff auf die API einschränken

Aktualisieren Sie die Rolle mit der ID „5“ und verweigern Sie den Aufruf aller Methoden „create“, „update“ oder „delete“.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "role.update",
    "params": [
        {
            "roleid": "5",
            "rules": {
                "api.access": "1",
                "api.mode": "0",
                "api": ["*.create", "*.update", "*.delete"]
            }
        }
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "5"
        ]
    },
    "id": 1
}
```

Quelle

CRole::update() in *ui/include/classes/api/services/CRole.php*.

Script

Diese Klasse ist für die Arbeit mit Skripten vorgesehen.

Objektreferenzen:

- Script
  - webhook-Parameter
- Debug
  - Log-Eintrag

Verfügbare Methoden:

- script.create - neue Skripte erstellen
- script.delete - Skripte löschen

- script.execute - Skripte ausführen
- script.get - Skripte abrufen
- script.getscriptsbyevents - Skripte für Ereignisse abrufen
- script.getscriptsbyhosts - Skripte für Hosts abrufen
- script.update - Skripte aktualisieren

Skript-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `script` API.

Skript

Das Skriptobjekt verfügt über die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
scriptid	ID	ID des Skripts. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
name	string	Name des Skripts. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
type	integer	Skripttyp. Mögliche Werte, wenn `scope` auf „action operation“ gesetzt ist: 0 - Skript; 1 - IPMI; 2 - SSH; 3 - TELNET; 5 - Webhook. Mögliche Werte, wenn `scope` auf „manual host action“ oder „manual event action“ gesetzt ist: 6 - URL. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
command	string	Auszuführender Befehl. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „Script“, „IPMI“, „SSH“, „TELNET“ oder „Webhook“ gesetzt ist
scope	integer	Gültigkeitsbereich des Skripts. Mögliche Werte: 1 - action operation; 2 - manual host action; 4 - manual event action. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
execute_on	integer	Ort, an dem das Skript ausgeführt wird. Mögliche Werte: 0 - auf dem Zabbix Agent ausführen; 1 - auf dem Zabbix Server ausführen. Dies wird nur *unterstützt*, wenn die Ausführung globaler Skripte auf dem Zabbix Server aktiviert ist; 2 - *(Standard)* auf dem Zabbix Server oder Proxy ausführen. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „Script“ gesetzt ist

Eigenschaft	Typ	Beschreibung
menu_path	string	Durch Schrägstriche getrennte Ordner, die im Frontend beim Anklicken eines Hosts oder Ereignisses eine menüähnliche Navigation bilden. Verhalten der Eigenschaft: - *unterstützt*, wenn `scope` auf „manual host action“ oder „manual event action“ gesetzt ist
authtype	integer	Für den SSH-Skripttyp verwendete Authentifizierungsmethode. Mögliche Werte: 0 - Passwort; 1 - öffentlicher Schlüssel. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „SSH“ gesetzt ist
username	string	Für die Authentifizierung verwendeter Benutzername. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „SSH“ oder „TELNET“ gesetzt ist
password	string	Für SSH-Skripte mit passwortbasierter Authentifizierung und TELNET-Skripte verwendetes Passwort. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „SSH“ und `authtype` auf „password“ gesetzt ist oder `type` auf „TELNET“ gesetzt ist
publickey	string	Name der Datei mit dem öffentlichen Schlüssel, die für SSH-Skripte mit Authentifizierung über einen öffentlichen Schlüssel verwendet wird. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „SSH“ und `authtype` auf „public key“ gesetzt ist
privatekey	string	Name der Datei mit dem privaten Schlüssel, die für SSH-Skripte mit Authentifizierung über einen öffentlichen Schlüssel verwendet wird. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „SSH“ und `authtype` auf „public key“ gesetzt ist
port	string	Für SSH- und TELNET-Skripte verwendete Portnummer. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „SSH“ oder „TELNET“ gesetzt ist
groupid	ID	ID der Hostgruppe, auf der das Skript ausgeführt werden kann. Wenn der Wert auf „0“ gesetzt ist, ist das Skript für alle Hostgruppen verfügbar. Standard: 0.
usrgrpid	ID	ID der Benutzergruppe, die das Skript ausführen darf. Wenn der Wert auf „0“ gesetzt ist, ist das Skript für alle Benutzergruppen verfügbar. Standard: 0. Verhalten der Eigenschaft: - *unterstützt*, wenn `scope` auf „manual host action“ oder „manual event action“ gesetzt ist

Eigenschaft	Typ	Beschreibung
host_access	integer	Zum Ausführen des Skripts erforderliche Hostberechtigungen. Mögliche Werte: 2 - *(Standard)* Lesen; 3 - Schreiben. Verhalten der Eigenschaft: - *unterstützt*, wenn `scope` auf „manual host action“ oder „manual event action“ gesetzt ist
confirmation	string	Text des Bestätigungsdialogs. Der Dialog wird angezeigt, wenn versucht wird, das Skript über das Zabbix Frontend auszuführen. Verhalten der Eigenschaft: - *unterstützt*, wenn `scope` auf „manual host action“ oder „manual event action“ gesetzt ist
timeout	string	Zeitüberschreitung für die Ausführung des Webhook-Skripts in Sekunden. Zeitsuffixe werden unterstützt (z. B. `30s`, `1m`). Mögliche Werte: 1-60s. Standard: 30s. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „Webhook“ gesetzt ist
parameters	array	Array von Webhook-Eingabeparametern. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „Webhook“ gesetzt ist
description	string	Beschreibung des Skripts.
url	string	Benutzerdefinierte URL. Verhalten der Eigenschaft: - *erforderlich*, wenn `type` auf „URL“ gesetzt ist
new_window	integer	URL in einem neuen Fenster öffnen. Mögliche Werte: 0 - Nein; 1 - *(Standard)* Ja. Verhalten der Eigenschaft: - *unterstützt*, wenn `type` auf „URL“ gesetzt ist
manualinput	integer	Gibt an, ob das Skript vom Benutzer bereitgestellte Eingaben akzeptiert. Mögliche Werte: 0 - *(Standard)* Deaktiviert; 1 - Aktiviert; Verhalten der Eigenschaft: - *unterstützt*, wenn `scope` auf „manual host action“ oder „manual event action“ gesetzt ist
manualinput_prompt	string	Text der Eingabeaufforderung. Verhalten der Eigenschaft: - *erforderlich*, wenn `manualinput` auf „Enabled“ gesetzt ist
manualinput_validator	string	Zeichenfolgenfeld zur Validierung der vom Benutzer bereitgestellten Eingabe. Die Zeichenfolge besteht entweder aus einem regulären Ausdruck oder einer durch Kommas getrennten Wertemenge. Verhalten der Eigenschaft: - *erforderlich*, wenn `manualinput` auf „Enabled“ gesetzt ist

Eigenschaft	Typ	Beschreibung
manualinput_validator_type	integer	Bestimmt den erwarteten Typ der Benutzereingabe. Mögliche Werte: 0 - *(Standard)* String. Gibt an, dass manualinput_validator als regulärer Ausdruck behandelt werden soll; 1 - Liste. Gibt an, dass manualinput_validator als durch Kommas getrennte Liste möglicher Eingabewerte behandelt werden soll. Verhalten der Eigenschaft: - *unterstützt*, wenn `manualinput` auf „Enabled“ gesetzt ist
manualinput_default_value	string	Standardwert zum automatischen Ausfüllen der Benutzereingabe. Verhalten der Eigenschaft: - *unterstützt*, wenn `manualinput_validator_type` auf „String“ gesetzt ist

webhook-Parameter

Parameter, die beim Aufruf an das webhook-Skript übergeben werden, haben die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
name	string	Parametername. Verhalten der Eigenschaft: - *erforderlich*
value	string	Parameterwert. Unterstützt Makros.

Debug

Debug-Informationen des ausgeführten webhook-Skripts. Das Debug-Objekt hat die folgenden Eigenschaften.

Eigenschaft	Type	Beschreibung
logs	array	Array von Log-Einträgen.
ms	string	Dauer der Skriptausführung in Millisekunden.

Protokolleintrag

Das Protokolleintrag-Objekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
level	integer	Protokollebene.
ms	string	Die in Millisekunden verstrichene Zeit seit der Ausführung des Skripts, bevor der Protokolleintrag hinzugefügt wurde.
message	string	Protokollnachricht.

script.create

Beschreibung

`object script.create(object/array scripts)`

Diese Methode ermöglicht es Ihnen, neue Skripte zu erstellen.

Note:
Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

(object/array) Zu erstellende Skripte.

Die Methode akzeptiert Skripte mit den standardmäßigen Skript-Eigenschaften.

Rückgabewerte

(object) Gibt ein Objekt zurück, das die IDs der erstellten Skripte unter der Eigenschaft scriptids enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Skripte.

Beispiele

Ein webhook-Skript erstellen

Erstellen Sie ein webhook-Skript, das eine HTTP-Anfrage an einen externen Dienst sendet.

Anfrage:

```
{
    "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
        "scope": 1,
        "type": 5,
        "timeout": "40s",
        "parameters": [
            {
                "name": "token",
                "value": "{$WEBHOOK.TOKEN}"
            },
            {
                "name": "host",
                "value": "{HOST.HOST}"
            },
            {
                "name": "v",
                "value": "2.2"
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "3"
        ]
    },
    "id": 1
}
```

Ein SSH-Skript erstellen

Erstellen Sie ein SSH-Skript mit Public-Key-Authentifizierung, das auf einem Host ausgeführt werden kann und ein Kontextmenü hat.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "5"
        ]
    },
    "id": 1
}
```

Ein benutzerdefiniertes Skript erstellen

Erstellen Sie ein benutzerdefiniertes Skript, das einen Server neu startet. Das Skript benötigt Schreibzugriff auf den Host und fordert den Benutzer zu manueller Eingabe auf. Nach erfolgreicher Übermittlung der Eingabe zeigt das Skript eine Bestätigungsmeldung im Frontend an.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "4"
        ]
    },
    "id": 1
}
```

Ein Skript vom Typ URL erstellen

Erstellen Sie ein Skript vom Typ URL für den Geltungsbereich Host, das im selben Fenster geöffnet wird und einen Bestätigungstext

hat.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "56"
        ]
    },
    "id": 1
}
```

Ein Skript vom Typ URL mit manueller Eingabe erstellen

Erstellen Sie ein Skript vom Typ URL für den Ereignisbereich, das in einem neuen Fenster geöffnet wird und eine manuelle Eingabe hat.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "script.create",
    "params": {
        "name": "URL-Skript mit manueller Eingabe",
        "type": 6,
        "scope": 4,
        "url": "http://zabbix/ui/zabbix.php?action={MANUALINPUT}",
        "new_window": 1,
        "manualinput": 1,
        "manualinput_prompt": "Wählen Sie eine zu öffnende Seite aus:",
        "manualinput_validator": "dashboard.view,script.list,actionlog.list",
        "manualinput_validator_type": 1
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "57"
        ]
    },
    "id": 1
}
```

Quelle

CScript::create() in *ui/include/classes/api/services/CScript.php*.

script.delete

Beschreibung

`object script.delete(array scriptIds)`

Diese Methode ermöglicht es Ihnen, Skripte zu löschen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der zu löschenden Skripte.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Skripte unter der Eigenschaft `scriptids` enthält.

Beispiele

Mehrere Skripte löschen

Löschen Sie zwei Skripte.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "script.delete",
    "params": [
        "3",
        "4"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "3",
            "4"
        ]
    },
    "id": 1
}
```

Quelle

CScript::delete() in *ui/include/classes/api/services/CScript.php*.

script.execute

Beschreibung

`object script.execute(object parameters)`

Diese Methode ermöglicht es Ihnen, ein Skript auf einem Host oder für ein Ereignis auszuführen. Ausgenommen sind Skripte vom Typ URL; diese können nicht ausgeführt werden.

Note:
Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die ID des auszuführenden Skripts sowie entweder die ID des Hosts oder die ID des Ereignisses und den Wert manualinput enthalten.

Parameter	Type	Beschreibung
scriptid	ID	ID des auszuführenden Skripts. Parameterverhalten: - *erforderlich*
hostid	ID	ID des Hosts, auf dem das Skript ausgeführt wird. Parameterverhalten: - *erforderlich*, wenn `eventid` nicht gesetzt ist
eventid	ID	ID des Ereignisses, für das das Skript ausgeführt wird. Parameterverhalten: - *erforderlich*, wenn `hostid` nicht gesetzt ist
manualinput	string	Vom Benutzer bereitgestellter Wert, mit dem das Skript ausgeführt wird; ersetzt dabei das Makro {MANUALINPUT}.

Rückgabewerte

`(object)` Gibt das Ergebnis der Skriptausführung zurück.

Eigenschaft	Type	Beschreibung
response	string	Gibt an, ob das Skript erfolgreich ausgeführt wurde. Möglicher Wert – `success`.
value	string	Skriptausgabe.
debug	object	Enthält ein debug-Objekt, wenn ein webhook-Skript ausgeführt wird. Für andere Skripttypen enthält es ein leeres Objekt.

Beispiele

Ein webhook-Skript ausführen

Führen Sie ein webhook-Skript aus, das eine HTTP-Anfrage an einen externen Dienst sendet.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "script.execute",
    "params": {
        "scriptid": "4",
        "hostid": "30079"
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```


Ein benutzerdefiniertes Skript ausführen

Führen Sie ein „ping“-Skript auf einem Host aus.

Anfrage:


```
{
    "jsonrpc": "2.0",
    "method": "script.execute",
    "params": {
        "scriptid": "1",
        "hostid": "30079"
    },
    "id": 1
}
```


Antwort:


```
{
    "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
}
```


Ein benutzerdefiniertes Skript mit manueller Eingabe ausführen

Führen Sie ein „ping“-Skript mit dem Befehl „ping -c {MANUALINPUT} {HOST.CONN}; case $? in [01]) true;; *) false;; esac“ auf einem Host aus.

Anfrage:


```
{
    "jsonrpc": "2.0",
    "method": "script.execute",
    "params": {
        "scriptid": "7",
        "hostid": "30079",
        "manualinput": "2"
    },
    "id": 1
}
```


Antwort:


```
{
    "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
}
```

Quelle

CScript::execute() in *ui/include/classes/api/services/CScript.php*.

script.get

Beschreibung

`integer/array script.get(object parameters)`

Die Methode ermöglicht es Ihnen, Skripte anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
groupids	ID/array	Gibt nur Skripte zurück, die für die angegebenen Host-Gruppen ausgeführt werden können.
hostids	ID/array	Gibt nur Skripte zurück, die auf den angegebenen Hosts ausgeführt werden können.
scriptids	ID/array	Gibt nur Skripte mit den angegebenen IDs zurück.
usrgrpids	ID/array	Gibt nur Skripte zurück, die von Benutzern in den angegebenen Benutzergruppen ausgeführt werden können.
selectHostGroups	query	Gibt eine Eigenschaft `hostgroups` mit Host-Gruppen zurück, für die das Skript ausgeführt werden kann.
selectHosts	query	Gibt eine Eigenschaft `hosts` mit Hosts zurück, auf denen das Skript ausgeführt werden kann.
selectActions	query	Gibt eine Eigenschaft `actions` mit Aktionen zurück, mit denen das Skript verknüpft ist.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `scriptid`, `name`.
countOutput	boolean	Diese Parameter sind im Referenzkommentar beschrieben.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Gibt entweder Folgendes zurück:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Alle Skripte abrufen

Rufen Sie alle konfigurierten Skripte ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "script.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Antwort:

```
{
    "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": "Betriebssystem erkennen",
            "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": "Zu {HOST.NAME} gehen?",
            "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 mit Benutzereingabe",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Zabbix-Seite {MANUALINPUT} öffnen?",
            "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": "Wählen Sie eine zu öffnende Seite aus:",
            "manualinput_validator": "dashboard.view,script.list,actionlog.list",
            "manualinput_validator_type": "1",
            "parameters": []
        }
    ],
    "id": 1
}
```


Siehe auch

- Host
- Host-Gruppe

Quelle

CScript::get() in *ui/include/classes/api/services/CScript.php*.

script.getscriptsbyevents

Beschreibung

`object script.getscriptsbyevents(object parameters)`

Diese Methode ermöglicht es Ihnen, alle verfügbaren Skripte für das angegebene Ereignis abzurufen oder ein bestimmtes Skript, wenn die Skript-ID angegeben ist. Wenn manualinput angegeben ist, ersetzt es das Makro `{MANUALINPUT}` durch den angegebenen Wert.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Die Methode akzeptiert ein Objekt oder ein Array von Objekten mit den folgenden Parametern.

Parameter	Type	Beschreibung
eventid	ID	ID des Ereignisses, für das Skripte zurückgegeben werden sollen. Muss eindeutig sein. Parameterverhalten: - *erforderlich*
scriptid	ID	ID des zurückzugebenden Skripts.
manualinput	string	Wert des vom Benutzer bereitgestellten Makrowerts {MANUALINPUT}.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, bei dem Ereignis-IDs als Eigenschaften und Arrays mit verfügbaren Skripten als Werte enthalten sind. Wenn eine Skript-ID angegeben wird, ist der zugehörige Wert ein Array, das das entsprechende Skript enthält.

Note:
Die Methode erweitert Makros im Text `confirmation`, im Text `manualinput prompt` und in der `url` automatisch.
Wenn der Parameter `manualinput` angegeben wird, wird das Makro {MANUALINPUT} in den angegebenen Wert aufgelöst.

Beispiele

Skripte nach Ereignis-IDs abrufen

Rufen Sie alle Skripte ab, die für die Ereignisse „632“ und „614“ verfügbar sind.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyevents",
    "params": [
      {
         "eventid":  "632"
      },
      {
         "eventid":  "614"
      }
    ],
    "id": 1
}
```

Antwort:


```
{
    "jsonrpc": "2.0",
    "result": {
        "632": [
            {
                "scriptid": "3",
                "name": "Betriebssystem erkennen",
                "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": "Zabbix-Seite öffnen",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Möchten Sie die Seite *UNKNOWN* wirklich öffnen?",
            "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": "Zu öffnende Zabbix-Seite:",
            "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": "Betriebssystem erkennen",
            "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": "Zabbix-Seite öffnen",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Möchten Sie die Seite *UNKNOWN* wirklich öffnen?",
            "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": "Zu öffnende Zabbix-Seite:",
            "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
}
```


Bestimmtes Skript mit manualinput-Wert abrufen.

Rufen Sie das Skript mit der ID ”4” für das Ereignis ”632” mit dem manualinput-Wert ”dashboard.view” ab.

Anfrage:


```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyevents",
    "params": [
      {
        "eventid":  "632",
        "scriptid": "4",
        "manualinput": "dashboard.view"
      }
  ],
    "id": 1
}
```


Antwort:


```
{
    "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
}
```

Quelle

CScript::getScriptsByEvents() in *ui/include/classes/api/services/CScript.php*.

script.getscriptsbyhosts

Beschreibung

`object script.getscriptsbyhosts(object parameters)`

Diese Methode ermöglicht es Ihnen, alle verfügbaren Skripte auf dem angegebenen Host abzurufen oder ein bestimmtes Skript, wenn die Skript-ID angegeben ist. Wenn manualinput angegeben ist, wird das Makro `{MANUALINPUT}` durch den angegebenen Wert ersetzt.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Die Methode akzeptiert ein Objekt oder ein Array von Objekten mit den folgenden Parametern.

Parameter	Typ	Beschreibung
hostid	ID	ID des Host, für den Skripte zurückgegeben werden sollen. Muss eindeutig sein. Parameterverhalten: - *erforderlich*
scriptid	ID	ID des Skripts, das zurückgegeben werden soll.
manualinput	string	Wert des vom Benutzer bereitgestellten Makrowerts {MANUALINPUT}.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das Host-IDs als Eigenschaften und Arrays verfügbarer Skripte als Werte enthält. Wenn eine Skript-ID angegeben wird, ist der zugehörige Wert ein Array, das das jeweilige Skript enthält.

> **Note:**
> Die Methode erweitert Makros im Text `confirmation`, im Text `manualinput prompt` und in der `url` automatisch.
> Wenn der Parameter manualinput angegeben wird, wird das Makro {MANUALINPUT} in den angegebenen Wert aufgelöst.

Beispiele

Skripte nach Host-IDs abrufen

Rufen Sie alle Skripte ab, die auf den Hosts „30079“ und „30073“ verfügbar sind.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyhosts",
```

```
    "params": [
        {
            "hostid":  "30079"
        },
        {
            "hostid":  "30073"
        }
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "30079": [
            {
                "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_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": "Open Zabbix page",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Are you sure you want to open 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": "Zabbix page to open:",
            "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": "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_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": "Open Zabbix page",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Are you sure you want to open 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": "Zabbix page to open:",
            "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
}
```

Bestimmtes Skript mit manualinput-Wert abrufen.

Rufen Sie das Skript mit der ID "4" auf dem Host "30079" mit dem manualinput-Wert "dashboard.view" ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyhosts",
    "params": [
      {
        "hostid":  "30079",
        "scriptid": "4",
        "manualinput": "dashboard.view"
      }
  ],
    "id": 1
}
```

Antwort:

```
{
    "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
}
```

Quelle

CScript::getScriptsByHosts() in *ui/include/classes/api/services/CScript.php*.

script.update

Beschreibung

`object script.update(object/array scripts)`

Diese Methode ermöglicht es Ihnen, vorhandene Skripte zu aktualisieren.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` zu aktualisierende Skript-Eigenschaften.

Die Eigenschaft `scriptid` muss für jedes Skript definiert sein, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert. Eine Ausnahme ist die Änderung der Eigenschaft `type` von 5 (webhook) zu einem anderen Wert: Die Eigenschaft `parameters` wird geleert.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Skripte unter der Eigenschaft `scriptids` enthält.

Beispiele

Skriptbefehl ändern

Ändern Sie den Befehl des Skripts in „/bin/ping -c 10 {HOST.CONN} 2>&1“.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "script.update",
    "params": {
        "scriptid": "1",
        "command": "/bin/ping -c 10 {HOST.CONN} 2>&1"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "1"
        ]
    },
    "id": 1
}
```

Skriptbefehl ändern und manuelle Eingabe hinzufügen

Ändern Sie den Befehl des Skripts in ”/bin/ping -c {MANUALINPUT} {HOST.CONN} 2>&1”.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "script.update",
    "params": {
      "scriptid": "1",
      "command": "/bin/ping -c {MANUALINPUT} {HOST.CONN} 2>&1",
      "manualinput": "1",
```

```
        "manualinput_prompt": "Geben Sie die Anzahl der ICMP-Pakete an, die mit dem ping-Befehl gesendet wer
        "manualinput_validator": "^(?:[1-9]|10)$",
        "manualinput_validator_type": "0",
        "manualinput_default_value": "10"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "1"
        ]
    },
    "id": 1
}
```

Quelle

CScript::update() in *ui/include/classes/api/services/CScript.php*.

Serverinformationen

Diese Klasse dient zum Abrufen von Informationen über den Zabbix Server.

Verfügbare Methoden:

- serverinfo.get - ruft den Zeitpunkt der letzten Aktualisierung vom Zabbix Server ab

serverinfo.get

Mit dieser Methode können Sie die Zeit der letzten Aktualisierung des angegebenen Zabbix-Servers abrufen.

Die Methode ist für *User*, *Admin*, *Super admin* und nicht authentifizierte Benutzer verfügbar.

Methodensignatur: `object serverinfo.get(object parameters)`

Methodenquelle: `CServerInfo::get()` in `ui/include/classes/api/services/CServerInfo.php`

Parameter

Parameter	Typ	Beschreibung
serverid	ID	ID des Zabbix-Servers. Parameterverhalten: - Erforderlich
output	query	Dieser Parameter wird im Referenzkommentar beschrieben.

Rückgabewerte

Property	Type	Beschreibung
serverid	ID	ID des Zabbix-Servers.
lastaccess	timestamp	Zeitpunkt der letzten Aktualisierung durch den Zabbix-Server. Für einen Hochverfügbarkeitscluster (HA) wird hier der Wert `lastaccess` des aktiven Knotens angezeigt.

Beispiele

Serverinformationen abrufen

Ruft Informationen zum angegebenen Zabbix-Server ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "serverinfo.get",
    "params": {
        "serverid": "019fb1f9-22bd-74ea-bc03-51efb8b76aba",
        "output": "extend"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serverid": "019fb1f9-22bd-74ea-bc03-51efb8b76aba",
        "lastaccess": 1785337245
    },
    "id": 1
}
```

Service

Diese Klasse ist für die Arbeit mit IT-Infrastruktur-/Business-Services konzipiert.

Objektreferenzen:

- Service
- Statusregel
- Service-Tag
- Service-Alarm
- Problem-Tag

Verfügbare Methoden:

- service.create - neue Services erstellen
- service.delete - Services löschen
- service.get - Services abrufen
- service.update - Services aktualisieren

Dienst-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `service` API.

Service

Das Service-Objekt hat die folgenden Eigenschaften.

Property	Type	Description
serviceid	ID	ID des Service. Property behavior: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge

Property	Type	Description
algorithm	integer	Regel zur Statusberechnung. Nur anwendbar, wenn untergeordnete Services vorhanden sind. Mögliche Werte: 0 - Status auf OK setzen; 1 - am kritischsten, wenn alle untergeordneten Elemente Probleme haben; 2 - am kritischsten unter den untergeordneten Services. Property behavior: - *erforderlich* für Erstellvorgänge
name	string	Name des Service. Property behavior: - *erforderlich* für Erstellvorgänge
sortorder	integer	Position des Service, verwendet für die Sortierung. Mögliche Werte: 0-999. Property behavior: - *erforderlich* für Erstellvorgänge
weight	integer	Gewichtung des Service. Mögliche Werte: 0-1000000. Standard: 0.
propagation_rule	integer	Regel zur Statusweitergabe. Mögliche Werte: 0 - *(Standard)* Service-Status unverändert weitergeben - ohne Änderungen; 1 - den weitergegebenen Status um einen angegebenen `propagation_value` erhöhen (um 1 bis 5 Schweregrade); 2 - den weitergegebenen Status um einen angegebenen `propagation_value` verringern (um 1 bis 5 Schweregrade); 3 - diesen Service ignorieren - der Status wird überhaupt nicht an den übergeordneten Service weitergegeben; 4 - festen Service-Status mit einem angegebenen `propagation_value` setzen. Property behavior: - *erforderlich*, wenn `propagation_value` gesetzt ist

Property	Type	Description
propagation_value	integer	Wert der Statusweitergabe. Mögliche Werte, wenn `propagation_rule` auf "0" oder "3" gesetzt ist: 0 - Nicht klassifiziert. Mögliche Werte, wenn `propagation_rule` auf "1" oder "2" gesetzt ist: 1 - Information; 2 - Warnung; 3 - Durchschnittlich; 4 - Hoch; 5 - Katastrophe. Mögliche Werte, wenn `propagation_rule` auf "4" gesetzt ist: -1 - OK; 0 - Nicht klassifiziert; 1 - Information; 2 - Warnung; 3 - Durchschnittlich; 4 - Hoch; 5 - Katastrophe. Property behavior: - *erforderlich*, wenn `propagation_rule` gesetzt ist
status	integer	Ob sich der Service im OK- oder Problemzustand befindet. Wenn sich der Service in einem Problemzustand befindet, entspricht `status` entweder der Schwere des kritischsten Problems oder dem höchsten Status eines untergeordneten Service in einem Problemzustand. Wenn sich der Service im OK-Zustand befindet, ist `status` gleich -1. Property behavior: - *schreibgeschützt*
description	string	Beschreibung des Service.
uuid	string	Universell eindeutige Kennung, verwendet, um importierte Services mit bereits vorhandenen zu verknüpfen. Wird automatisch generiert, wenn nicht angegeben.
created_at	integer	Unix-Zeitstempel, wann der Service erstellt wurde.
readonly	integer	Zugriff auf den Service. Mögliche Werte: 0 - Lese- und Schreibzugriff; 1 - Nur Lesezugriff. Property behavior: - *schreibgeschützt*

Statusregel

Das Statusregel-Objekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
type	integer	Bedingung zum Setzen des Status (Neuer Status). Mögliche Werte: 0 - wenn mindestens (N) untergeordnete Services den Status (Status) oder höher haben; 1 - wenn mindestens (N%) der untergeordneten Services den Status (Status) oder höher haben; 2 - wenn weniger als (N) untergeordnete Services den Status (Status) oder niedriger haben; 3 - wenn weniger als (N%) der untergeordneten Services den Status (Status) oder niedriger haben; 4 - wenn die Gewichtung der untergeordneten Services mit dem Status (Status) oder höher mindestens (W) beträgt; 5 - wenn die Gewichtung der untergeordneten Services mit dem Status (Status) oder höher mindestens (N%) beträgt; 6 - wenn die Gewichtung der untergeordneten Services mit dem Status (Status) oder niedriger kleiner als (W) ist; 7 - wenn die Gewichtung der untergeordneten Services mit dem Status (Status) oder niedriger kleiner als (N%) ist. Wobei: - N (W) ist `limit_value`; - (Status) ist `limit_status`; - (Neuer Status) ist `new_status`. Verhalten der Eigenschaft: - *erforderlich*
limit_value	integer	Grenzwert. Mögliche Werte: - für N und W: 1-100000; - für N%: 1-100. Verhalten der Eigenschaft: - *erforderlich*
limit_status	integer	Grenzstatus. Mögliche Werte: -1 - OK; 0 - Nicht klassifiziert; 1 - Information; 2 - Warnung; 3 - Durchschnittlich; 4 - Hoch; 5 - Katastrophe. Verhalten der Eigenschaft: - *erforderlich*
new_status	integer	Neuer Statuswert. Mögliche Werte: 0 - Nicht klassifiziert; 1 - Information; 2 - Warnung; 3 - Durchschnittlich; 4 - Hoch; 5 - Katastrophe. Verhalten der Eigenschaft: - *erforderlich*

Service-Tag

Das Service-Tag-Objekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
tag	string	Name des Service-Tags. Verhalten der Eigenschaft: - *erforderlich*
value	string	Wert des Service-Tags.

Service-Alarm

Note:
Service-Alarme können nicht direkt über die Zabbix API erstellt, aktualisiert oder gelöscht werden.

Die Service-Alarm-Objekte repräsentieren eine Zustandsänderung eines Services. Sie haben die folgenden Eigenschaften.

Property	Type	Description
clock	timestamp	Zeitpunkt, zu dem die Zustandsänderung des Services stattgefunden hat.
value	integer	Status des Services. Siehe die Eigenschaft `status` des Services für eine Liste möglicher Werte.

Problem-Tag

Problem-Tags ermöglichen die Verknüpfung von Services mit Problemereignissen. Das Problem-Tag-Objekt hat die folgenden Eigenschaften.

Property	Type	Description
tag	string	Name des Problem-Tags. Property behavior: - *required*
operator	integer	Zuordnungsbedingung operator. Mögliche Werte: 0 - *(default)* Entspricht; 2 - Enthält.
value	string	Wert des Problem-Tags.

service.create

Beschreibung

`object service.create(object/array services)`

Mit dieser Methode können Sie neue Services erstellen.

Note:
Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` zu erstellende Services.

Zusätzlich zu den Standard-Service-Eigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Typ	Beschreibung
children	array	Untergeordnete Services, die mit dem Service verknüpft werden sollen. Für die untergeordneten Services darf nur die Eigenschaft `serviceid` definiert sein.
parents	array	Übergeordnete Services, die mit dem Service verknüpft werden sollen. Für die übergeordneten Services darf nur die Eigenschaft `serviceid` definiert sein.
tags	array	Service-Tags, die für den Service erstellt werden sollen.
problem_tags	array	Problem-Tags, die für den Service erstellt werden sollen.
status_rules	array	Statusregeln, die für den Service erstellt werden sollen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Services unter der Eigenschaft `serviceids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Services.

Beispiele

Erstellen eines Service

Erstellen Sie einen Service, der in den Problemstatus wechselt, wenn mindestens ein untergeordneter Service ein Problem hat.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "service.create",
    "params": {
        "name": "Server 1",
        "algorithm": 1,
        "sortorder": 1
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "5"
        ]
    },
    "id": 1
}
```

Quelle

CService::create() in *ui/include/classes/api/services/CService.php*.

service.delete

Beschreibung

`object service.delete(array serviceIds)`

Diese Methode ermöglicht es Ihnen, Dienste zu löschen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

(array) IDs der zu löschenden Services.

Rückgabewerte

(object) Gibt ein Objekt zurück, das die IDs der gelöschten Services unter der Eigenschaft serviceids enthält.

Beispiele

Mehrere Services löschen

Löschen Sie zwei Services.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "service.delete",
    "params": [
        "4",
        "5"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "4",
            "5"
        ]
    },
    "id": 1
}
```

Quelle

CService::delete() in *ui/include/classes/api/services/CService.php*.

service.get

Beschreibung

`integer/array service.get(object parameters)`

Die Methode ermöglicht es Ihnen, Services anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Berechtigungen zum Aufruf der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

(object) Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
serviceids	ID/array	Gibt nur Services mit den angegebenen IDs zurück.
parentids	ID/array	Gibt nur Services zurück, die mit den angegebenen übergeordneten Services verknüpft sind.
deep_parentids	flag	Gibt alle direkten und indirekten untergeordneten Services zurück. Wird zusammen mit parentids verwendet.
childids	ID/array	Gibt nur Services zurück, die mit den angegebenen untergeordneten Services verknüpft sind.

Parameter	Type	Beschreibung
evaltype	integer	Tag-Auswertungsmethode. Mögliche Werte: 0 - *(Standard)* Und/Oder; 2 - Oder.
tags	object/array	Gibt nur Services mit den angegebenen Tags zurück. Format: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Ein leeres Array gibt alle Services zurück. Mögliche Werte für operator: 0 - *(Standard)* Enthält; 1 - Entspricht; 2 - Enthält nicht; 3 - Entspricht nicht; 4 - Existiert; 5 - Existiert nicht.
problem_tags	object/array	Gibt nur Services mit den angegebenen Problem-Tags zurück. Format: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Ein leeres Array gibt alle Services zurück. Mögliche Werte für operator: 0 - *(Standard)* Enthält; 1 - Entspricht; 2 - Enthält nicht; 3 - Entspricht nicht; 4 - Existiert; 5 - Existiert nicht.
without_problem_tags	flag	Gibt nur Services ohne Problem-Tags zurück.
slaids	ID/array	Gibt nur Services zurück, die mit den angegebenen SLA(s) verknüpft sind.
selectChildren	query	Gibt eine `children`-Eigenschaft mit den untergeordneten Services zurück. Unterstützt `count`.
selectParents	query	Gibt eine `parents`-Eigenschaft mit den übergeordneten Services zurück. Unterstützt `count`.
selectTags	query	Gibt eine `tags`-Eigenschaft mit Service-Tags zurück. Unterstützt `count`.
selectProblemEvents	query	Gibt eine `problem_events`-Eigenschaft mit einem Array von Problem-Event-Objekten zurück. Das Problem-Event-Objekt hat die folgenden Eigenschaften: `eventid` - `(ID)` Event-ID; `severity` - `(string)` Aktueller Event-Schweregrad; `name` - `(string)` Name des gelösten Events. Unterstützt `count`.
selectProblemTags	query	Gibt eine `problem_tags`-Eigenschaft mit Problem-Tags zurück. Unterstützt `count`.
selectStatusRules	query	Gibt eine `status_rules`-Eigenschaft mit Statusregeln zurück. Unterstützt `count`.

Parameter	Type	Beschreibung
selectStatusTimeline	object/array	Gibt eine `status_timeline`-Eigenschaft zurück, die Statusänderungen des Service für die angegebenen Zeiträume enthält. Format `[{"period_from": "<period_from>", "period_to": "<period_to>"}, ...]` - wobei `period_from` ein Startdatum (einschließlich; Integer-Zeitstempel) und `period_to` ein Enddatum (ausschließlich; Integer-Zeitstempel) des gewünschten Zeitraums ist. Gibt ein Array von Einträgen zurück, das eine `start_value`-Eigenschaft und ein `alarms`-Array für die Statusänderungen innerhalb der angegebenen Zeiträume enthält.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `serviceid`, `name`, `status`, `sortorder`, `created_at`.
countOutput	boolean	Diese Parameter sind in der Referenzbeschreibung beschrieben.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Gibt entweder Folgendes zurück:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Abrufen aller Services

Rufen Sie alle Daten zu allen Services und ihren Abhängigkeiten ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "service.get",
    "params": {
        "output": "extend",
        "selectChildren": "extend",
        "selectParents": ["serviceid", "name"]
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```


Quelle

CService::get() in *ui/include/classes/api/services/CService.php*.

service.update

Beschreibung

`object service.update(object/array services)`

Diese Methode ermöglicht es Ihnen, vorhandene Services zu aktualisieren.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` zu aktualisierende Service-Eigenschaften.

Die Eigenschaft `serviceid` muss für jeden Service definiert sein, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Zusätzlich zu den Standard-Service-Eigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
children	array	Untergeordnete Services, die die aktuellen untergeordneten Services ersetzen. Für die untergeordneten Services darf nur die Eigenschaft `serviceid` definiert sein.
parents	array	Übergeordnete Services, die die aktuellen übergeordneten Services ersetzen. Für die übergeordneten Services darf nur die Eigenschaft `serviceid` definiert sein.
tags	array	Service-Tags, die die aktuellen Service-Tags ersetzen.
problem_tags	array	Problem-Tags, die die aktuellen Problem-Tags ersetzen.
status_rules	array	Statusregeln, die die aktuellen Statusregeln ersetzen.

Rückgabewerte

(`object`) Gibt ein Objekt zurück, das die IDs der aktualisierten Services unter der Eigenschaft `serviceids` enthält.

Beispiele

Festlegen des übergeordneten Service für einen Service

Legen Sie den Service mit der ID „3“ als übergeordneten Service für den Service mit der ID „5“ fest.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "service.update",
    "params": {
        "serviceid": "5",
        "parents": [
            {
                "serviceid": "3"
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "5"
        ]
    },
    "id": 1
}
```

Hinzufügen einer geplanten Ausfallzeit

Fügen Sie für den Service mit der ID „4“ eine Ausfallzeit hinzu, die wöchentlich von Montag 22:00 Uhr bis Dienstag 10:00 Uhr geplant ist.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "service.update",
    "params": {
        "serviceid": "4",
        "times": [
            {
                "type": "1",
                "ts_from": "165600",
                "ts_to": "201600"
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "4"
        ]
    },
```

```
    "id": 1
}
```

Quelle

CService::update() in *ui/include/classes/api/services/CService.php*.

SLA

Diese Klasse ist für die Arbeit mit SLA-Objekten (Service Level Agreement) vorgesehen, die zur Bewertung der Leistung der IT-Infrastruktur und von Geschäftsdiensten verwendet werden.

Objektreferenzen:

- SLA
- SLA-Zeitplan
- SLA-ausgeschlossene Ausfallzeit
- SLA-Service-Tag

Verfügbare Methoden:

- sla.create - neue SLAs erstellen
- sla.delete - SLAs löschen
- sla.get - SLAs abrufen
- sla.getsli - Daten zum Service Level Indicator (SLI) für SLAs abrufen
- sla.update - SLAs aktualisieren

SLA Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `sla` (Service Level Agreement) API.

SLA

Das SLA-Objekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
slaid	ID	ID des SLA. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
name	string	Name des SLA. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
period	integer	Berichtszeitraum des SLA. Mögliche Werte: 0 - täglich; 1 - wöchentlich; 2 - monatlich; 3 - vierteljährlich; 4 - jährlich. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
slo	float	Minimal akzeptables Service Level Objective, ausgedrückt als Prozentsatz. Wenn der Service Level Indicator (SLI) darunter fällt, wird das SLA als problematisch/nicht erfüllt betrachtet. Mögliche Werte: 0-100 (bis zu 4 Nachkommastellen). Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge

Eigenschaft	Typ	Beschreibung
effective_date	integer	Gültigkeitsdatum des SLA. Mögliche Werte: Datums-Zeitstempel in UTC.
timezone	string	Berichtszeitzone, zum Beispiel: `Europe/London`, `UTC`. Eine vollständige Liste der unterstützten Zeitzonen finden Sie in der PHP-Dokumentation. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
status	integer	Status des SLA. Mögliche Werte: 0 - *(Standard)* SLA deaktiviert; 1 - SLA aktiviert.
description	string	Beschreibung des SLA.

SLA-Zeitplan

Das SLA-Zeitplan-Objekt definiert Zeiträume, in denen die verbundenen Services planmäßig betriebsbereit sein sollen. Es hat die folgenden Eigenschaften.

Eigenschaft	Type	Beschreibung
period_from	integer	Startzeit des wiederkehrenden wöchentlichen Zeitraums (einschließlich). Mögliche Werte: Anzahl der Sekunden (gezählt ab Sonntag). Property behavior: - *erforderlich*
period_to	integer	Endzeit des wiederkehrenden wöchentlichen Zeitraums (ausschließlich). Mögliche Werte: Anzahl der Sekunden (gezählt ab Sonntag). Property behavior: - *erforderlich*

SLA-ausgeschlossene Ausfallzeit

Das Objekt für ausgeschlossene Ausfallzeit definiert Zeiträume, in denen die verbundenen Services planmäßig außer Betrieb sind, ohne den SLI zu beeinflussen, z. B. während geplanter Wartungsarbeiten. Es hat die folgenden Eigenschaften.

Eigenschaft	Type	Beschreibung
name	string	Name der ausgeschlossenen Ausfallzeit. Property behavior: - *required*
period_from	integer	Startzeit der ausgeschlossenen Ausfallzeit (einschließlich). Mögliche Werte: Zeitstempel. Property behavior: - *required*

Eigenschaft	Type	Beschreibung
period_to	integer	Endzeit der ausgeschlossenen Ausfallzeit (ausschließlich). Mögliche Werte: Zeitstempel. Property behavior: - *required*

SLA-Service-Tag

Das SLA-Service-Tag-Objekt verknüpft Services, die in die Berechnungen für das SLA einbezogen werden sollen. Es hat die folgenden Eigenschaften.

Eigenschaft	Type	Beschreibung
tag	string	Name des SLA-Service-Tags. Verhalten der Eigenschaft: - *erforderlich*
operator	integer	Operator des SLA-Service-Tags. Mögliche Werte: 0 - *(Standard)* Gleich; 2 - Enthält.
value	string	Wert des SLA-Service-Tags.

sla.create

Beschreibung

`object sla.create(object/array SLAs)`

Diese Methode ermöglicht es Ihnen, neue SLA-Objekte zu erstellen.

Note:
Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellende SLA-Objekte.

Zusätzlich zu den standardmäßigen SLA-Eigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Typ	Beschreibung
service_tags	array	Zu erstellende SLA-Service-Tags für die SLA. Parameterverhalten: - *erforderlich*
schedule	array	Zu erstellender SLA-Zeitplan für die SLA. Die Angabe eines leeren Parameters wird als 24x7-Zeitplan interpretiert. Standard: 24x7-Zeitplan.
excluded_downtimes	array	Zu erstellende ausgeschlossene SLA-Ausfallzeiten für die SLA.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten SLAs unter der Eigenschaft `slaids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen SLAs.

Beispiele

Erstellen eines SLA

Weisen Sie an, einen SLA-Eintrag zu erstellen für: * die Verfolgung der Verfügbarkeit von SQL-Engine-bezogenen Services; * einen benutzerdefinierten Zeitplan für alle Wochentage, wobei die letzte Stunde am Samstag ausgeschlossen ist; * ein Gültigkeitsdatum am letzten Tag des Jahres 2022; * mit einer geplanten Ausfallzeit von 1 Stunde und 15 Minuten, beginnend um Mitternacht am 4. Juli; * die Berechnung des wöchentlichen SLA-Berichts wird aktiviert sein; * das minimal akzeptable SLO beträgt 99,9995 %.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "sla.create",
    "params": [
        {
            "name": "Datenbank-Verfügbarkeit",
            "slo": "99.9995",
            "period": "1",
            "timezone": "America/Toronto",
            "description": "Stellen Sie eine ausgezeichnete Verfügbarkeit für die wichtigsten Datenbank-En
            "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": "Einführung des Software-Versions-Upgrades",
                    "period_from": "1648760400",
                    "period_to": "1648764900"
                }
            ]
        }
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

Quelle

CSla::create() in *ui/include/classes/api/services/CSla.php*.

sla.delete

Beschreibung

`object sla.delete(array slaids)`

Diese Methode ermöglicht es Ihnen, SLA-Einträge zu löschen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der zu löschenden SLAs.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten SLAs unter der Eigenschaft `slaids` enthält.

Beispiele

Mehrere SLAs löschen

Löschen Sie zwei SLA-Einträge.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "sla.delete",
    "params": [
        "4",
        "5"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "4",
            "5"
        ]
    },
    "id": 1
}
```

Quelle

CSla::delete() in *ui/include/classes/api/services/CSla.php*.

sla.get

Beschreibung

`integer/array sla.get(object parameters)`

Die Methode ermöglicht es Ihnen, SLA-Objekte anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Typ	Beschreibung
slaids	ID/array	Gibt nur SLAs mit den angegebenen IDs zurück.
serviceids	ID/array	Gibt nur SLAs zurück, die den angegebenen Services entsprechen.
selectSchedule	query	Gibt eine Eigenschaft `schedule` mit SLA-Zeitplänen zurück. Unterstützt `count`.
selectExcludedDowntimes	query	Gibt eine Eigenschaft `excluded_downtimes` mit ausgeschlossenen SLA-Ausfallzeiten zurück. Unterstützt `count`.
selectServiceTags	query	Gibt eine Eigenschaft `service_tags` mit SLA-Service-Tags zurück. Unterstützt `count`.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `slaid`, `name`, `period`, `slo`, `effective_date`, `timezone`, `status`, `description`.
countOutput	boolean	Diese Parameter sind in der Referenzkommentierung beschrieben.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Gibt entweder Folgendes zurück:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Abrufen aller SLAs

Rufen Sie alle Daten zu allen SLAs und ihren Eigenschaften ab.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "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
}
```


Quelle

CSla:get() in *ui/include/classes/api/services/CSla.php*.

sla.getsli

Beschreibung

`object sla.getsli(object parameters)`

Diese Methode ermöglicht es Ihnen, die Service Level Indicator (SLI)-Daten für ein Service Level Agreement (SLA) zu berechnen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die SLA-ID, Berichtszeiträume und optional die IDs der Services enthalten, für die der SLI berechnet werden soll.

Parameter	Type	Beschreibung
slaid	ID	ID des SLA, für das Verfügbarkeitsinformationen zurückgegeben werden sollen. Parameter behavior: - *required*
period_from	timestamp	Startzeitstempel (einschließlich), für den der SLI gemeldet werden soll. Mögliche Werte: Unix-Zeitstempel.
period_to	timestamp	Endzeitstempel (einschließlich), für den der SLI gemeldet werden soll. Mögliche Werte: Unix-Zeitstempel.
periods	integer	Anzahl der zu meldenden Zeiträume. Mögliche Werte: 1-100
serviceids	ID/array	IDs der Services, für die der SLI zurückgegeben werden soll.

Aufteilung von Zeiträumen

Die folgende Tabelle veranschaulicht die Anordnung der zurückgegebenen Zeitraumabschnitte basierend auf Parameterkombinationen.

Note:
Zurückgegebene Zeiträume liegen nicht vor dem ersten verfügbaren Zeitraum (basierend auf dem Gültigkeitsdatum des SLA) und überschreiten nicht den aktuellen Zeitraum.

Parameter			Zurückgegebene Zeiträume
period_from	**period_to**	**periods**	
-	-	-	Letzte 20 Zeiträume, einschließlich des aktuellen.
-	-	N	Letzte N Zeiträume.
-	angegeben	-	Letzte 20 Zeiträume vor `period_to`.
-	angegeben	N	Letzte N Zeiträume vor `period_to`.
angegeben	-	-	Erste 20 Zeiträume beginnend mit `period_from`.
angegeben	-	N	Erste N Zeiträume beginnend mit `period_from`.
angegeben	angegeben	-	Bis zu 100 Zeiträume im angegebenen Bereich.
angegeben	angegeben	N	N Zeiträume im angegebenen Bereich.

Rückgabewerte

`(object)` Gibt die Ergebnisse der Berechnung zurück.

Eigenschaft	Type	Beschreibung
periods	array	Liste der gemeldeten Zeiträume. Jeder gemeldete Zeitraum wird als Objekt dargestellt, das aus Folgendem besteht: - `period_from` - `(timestamp)` Startzeitstempel (einschließlich) des gemeldeten Zeitraums. - `period_to` - `(timestamp)` Endzeitstempel (ausschließlich) des gemeldeten Zeitraums. Die Zeiträume sind nach `period_from` sortiert, wobei die frühesten Zeiträume zuerst erscheinen.
serviceids	array	Liste der Service-IDs in den gemeldeten Zeiträumen. Die Sortierreihenfolge der Liste ist nicht definiert. Auch dann nicht, wenn der Parameter `serviceids` an die Methode `sla.getsli` übergeben wurde.

Eigenschaft	Type	Beschreibung
sli	array	SLI-Daten (als **zweidimensionales Array**) für jeden gemeldeten Zeitraum und Service. Der Index der Eigenschaft `periods` wird als **erste** Dimension der Eigenschaft `sli` verwendet. Der Index der Eigenschaft `serviceids` wird als **zweite** Dimension der Eigenschaft `sli` verwendet.

SLI-Daten

Die für jeden gemeldeten Zeitraum und Service zurückgegebenen SLI-Daten bestehen aus:

Eigenschaft	Type	Beschreibung
uptime	integer	Zeitspanne, die der Service während der geplanten Verfügbarkeit im Status *OK* verbracht hat, abzüglich der ausgeschlossenen Ausfallzeiten.
downtime	integer	Zeitspanne, die der Service während der geplanten Verfügbarkeit im Status *not OK* verbracht hat, abzüglich der ausgeschlossenen Ausfallzeiten.
sli	float	SLI (in Prozent der gesamten Verfügbarkeitszeit), basierend auf uptime und downtime.
error_budget	integer	Fehlerbudget (in Sekunden), basierend auf dem SLI und dem SLO.
excluded_downtimes	array	Array der ausgeschlossenen Ausfallzeiten in diesem Berichtszeitraum. Jedes Objekt enthält die folgenden Parameter: - `name` - (`string`) Name der ausgeschlossenen Ausfallzeit. - `period_from` - (`timestamp`) Startzeitstempel (einschließlich) der ausgeschlossenen Ausfallzeit. - `period_to` - (`timestamp`) Endzeitstempel (ausschließlich) der ausgeschlossenen Ausfallzeit. Ausgeschlossene Ausfallzeiten werden nach `period_from` sortiert, wobei die frühesten Zeiträume zuerst erscheinen.

Beispiele

Berechnung des SLI für ein tägliches SLA

Rufen Sie SLI-Daten für Services mit den IDs „1“ und „4“ ab, die mit dem SLA mit der ID „1“ verknüpft sind. Rufen Sie Daten für einen einzelnen Zeitraum bis „1761861599“ (30. Okt. 2025 23:59:59 GMT+0200) ab. Da der Berichtszeitraum des SLA täglich ist, werden SLI-Daten von „1761775200“ (30. Okt. 2025 00:00:00 GMT+0200) bis „1761861600“ (31. Okt. 2025 00:00:00 GMT+0200) abgerufen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "sla.getsli",
    "params": {
        "slaid": "1",
        "serviceids": [
            1,
            4
        ],
        "periods": 1,
        "period_to": 1761861599
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```


Berechnung des SLI für eine monatliche SLA

Rufen Sie SLI-Daten für Services mit den IDs ”50”, ”60” und ”70” ab, die mit der SLA mit der ID ”5” verknüpft sind. Rufen Sie Daten für drei Zeiträume ab, beginnend mit ”1635724800” (01. Nov. 2021 00:00:00 UTC). Da der Berichtszeitraum der SLA monatlich ist, werden SLI-Daten für die folgenden drei Monate abgerufen:

- Von ”1635724800” (01. Nov. 2021 00:00:00 UTC) bis ”1638316800” (01. Dez. 2021 00:00:00 UTC)
- Von ”1638316800” (01. Dez. 2021 00:00:00 UTC) bis ”1640995200” (01. Jan. 2022 00:00:00 UTC)
- Von ”1640995200” (01. Jan. 2022 00:00:00 UTC) bis ”1643673600” (01. Feb. 2022 00:00:00 UTC)

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "sla.getsli",
    "params": {
        "slaid": "5",
        "serviceids": [
            50,
            60,
            70
        ],
        "periods": 3,
        "period_from": 1635724800
```

```
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```

Quelle

CSla::getSli() in *ui/include/classes/api/services/CSla.php*

sla.update

Beschreibung

`object sla.update(object/array slaids)`

Diese Methode ermöglicht es Ihnen, bestehende SLA-Einträge zu aktualisieren.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu aktualisierende SLA-Eigenschaften.

Die Eigenschaft `slaid` muss für jede SLA definiert werden, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Zusätzlich zu den standardmäßigen SLA-Eigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
service_tags	array	SLA-Service-Tags, die die aktuellen SLA-Service-Tags ersetzen.
schedule	array	SLA-Zeitplan, der den aktuellen ersetzt. Wird der Parameter leer angegeben, wird dies als 24x7-Zeitplan interpretiert.
excluded_downtimes	array	SLA-Ausschlusszeiten, die die aktuellen ersetzen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten SLAs in der Eigenschaft `slaids` enthält.

Beispiele

Aktualisieren von Service-Tags

Legen Sie fest, dass die SLA mit der ID „5“ in monatlichen Intervallen für NoSQL-bezogene Services berechnet wird, ohne ihren Zeitplan oder ausgeschlossene Ausfallzeiten zu ändern; setzen Sie das SLO auf 95 %.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

Ändern des Zeitplans eines SLA

Schalten Sie das SLA mit der ID „5“ auf einen 24x7-Zeitplan um.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "sla.update",
    "params": {
        "slaid": "5",
        "schedule": []
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

Ändern der ausgeschlossenen Ausfallzeiten für ein SLA

Fügen Sie eine geplante 4-stündige Ausfallzeit für ein RAM-Upgrade am 6. April 2022 hinzu, wobei ein zuvor vorhandenes geplantes Software-Upgrade am 4. Juli für das SLA mit der ID „5“ beibehalten wird (muss neu definiert werden).

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

Quelle

CSla::update() in *ui/include/classes/api/services/CSla.php*.

Symbolzuordnung

Diese Klasse ist für die Arbeit mit Symbolzuordnungen vorgesehen.

Objektreferenzen:

- Symbolzuordnung
- Symbolzuordnungseintrag

Verfügbare Methoden:

- iconmap.create - neue Symbolzuordnungen erstellen
- iconmap.delete - Symbolzuordnungen löschen
- iconmap.get - Symbolzuordnungen abrufen
- iconmap.update - Symbolzuordnungen aktualisieren

Icon Map Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der`iconmap` API.

Symbolzuordnung

Das Symbolzuordnungsobjekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
iconmapid	ID	ID der Symbolzuordnung. Eigenschaftsverhalten: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge

Eigenschaft	Typ	Beschreibung
default_iconid	ID	ID des Standardsymbols. Eigenschaftsverhalten: - *erforderlich* für Erstellungsvorgänge
name	string	Name der Symbolzuordnung. Eigenschaftsverhalten: - *erforderlich* für Erstellungsvorgänge

Symbolzuordnung

Das Symbolzuordnungsobjekt definiert ein bestimmtes Symbol, das für Hosts mit einem bestimmten Wert eines Inventarfeldes verwendet wird. Es hat die folgenden Eigenschaften.

Eigenschaft	Type	Beschreibung
iconid	ID	ID des Symbols, das von der Symbolzuordnung verwendet wird. Property behavior: - *required*
expression	string	Ausdruck, mit dem das Inventarfeld abgeglichen wird. Property behavior: - *required*
inventory_link	integer	ID des Host-Inventarfeldes. Eine Liste der unterstützten Inventarfelder finden Sie im Host-Inventar-Objekt. Property behavior: - *required*
sortorder	integer	Position der Symbolzuordnung in der Symbolkarte. Property behavior: - *read-only*

iconmap.create

Beschreibung

`object iconmap.create(object/array iconMaps)`

Diese Methode ermöglicht es Ihnen, neue Icon-Maps zu erstellen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellende Symbolzuordnungen.

Zusätzlich zu den Standard-Eigenschaften von Symbolzuordnungen akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
mappings	array	Symbolzuordnungen, die für die Symbolzuordnung erstellt werden sollen. Parameterverhalten: - *erforderlich*

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Icon Maps unter der Eigenschaft `iconmapids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Icon Maps.

Beispiele

Eine Symbolzuordnung erstellen

Erstellen Sie eine Symbolzuordnung, um Hosts verschiedener Typen anzuzeigen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.create",
    "params": {
        "name": "Type icons",
        "default_iconid": "2",
        "mappings": [
            {
                "inventory_link": 1,
                "expression": "server",
                "iconid": "3"
            },
            {
                "inventory_link": 1,
                "expression": "switch",
                "iconid": "4"
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "iconmapids": [
            "2"
        ]
    },
    "id": 1
}
```

Siehe auch

- Symbolzuordnung

Quelle

CIconMap::create() in *ui/include/classes/api/services/CIconMap.php*.

iconmap.delete

Beschreibung

`object iconmap.delete(array iconMapIds)`

Diese Methode ermöglicht es Ihnen, Icon-Maps zu löschen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

(array) IDs der zu löschenden Symbolzuordnungen.

Rückgabewerte

(object) Gibt ein Objekt zurück, das die IDs der gelöschten Icon Maps unter der Eigenschaft iconmapids enthält.

Beispiele

Mehrere Symbolzuordnungen löschen

Löschen Sie zwei Symbolzuordnungen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.delete",
    "params": [
        "2",
        "5"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "iconmapids": [
            "2",
            "5"
        ]
    },
    "id": 1
}
```

Quelle

CIconMap::delete() in *ui/include/classes/api/services/CIconMap.php*.

iconmap.get

Beschreibung

`integer/array iconmap.get(object parameters)`

Die Methode ermöglicht es Ihnen, Icon-Zuordnungen anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

(object) Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
iconmapids	ID/array	Gibt nur Symbolzuordnungen mit den angegebenen IDs zurück.
sysmapids	ID/array	Gibt nur Symbolzuordnungen zurück, die in den angegebenen Karten verwendet werden.
selectMappings	query	Gibt eine Eigenschaft `mappings` mit den verwendeten Symbolzuordnungen zurück.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `iconmapid`, `name`.
countOutput	boolean	Diese Parameter sind in der Referenzkommentierung beschrieben.

Parameter	Type	Beschreibung
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Kann die folgenden Dinge zurück geben:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Eine Symbolzuordnung abrufen

Rufen Sie alle Daten zur Symbolzuordnung „3“ ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.get",
    "params": {
        "iconmapids": "3",
        "output": "extend",
        "selectMappings": "extend"
    },
    "id": 1
}
```

Antwort:

```
{
    "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-Typ-Symbole",
            "default_iconid": "2"
```

```
        }
    ],
    "id": 1
}
```

Siehe auch

- Symbolzuordnung

Quelle

CIconMap::get() in *ui/include/classes/api/services/CIconMap.php*.

iconmap.update

Beschreibung

`object iconmap.update(object/array iconMaps)`

Diese Methode ermöglicht es Ihnen, vorhandene Icon-Maps zu aktualisieren.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu aktualisierende Eigenschaften der Symbolzuordnung.

Die Eigenschaft `iconmapid` muss für jede Symbolzuordnung definiert werden, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Zusätzlich zu den standardmäßigen Eigenschaften der Symbolzuordnung akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Description
mappings	array	Symbolzuordnungen zum Ersetzen der vorhandenen Symbolzuordnungen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Icon Maps unter der Eigenschaft `iconmapids` enthält.

Beispiele

Symbolzuordnung umbenennen

Benennen Sie eine Symbolzuordnung in „OS-Symbole“ um.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.update",
    "params": {
        "iconmapid": "1",
        "name": "OS icons"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "iconmapids": [
            "1"
        ]
```

```
    },
    "id": 1
}
```

Siehe auch

- Symbolzuordnung

Quelle

CIconMap::update() in *ui/include/classes/api/services/CIconMap.php*.

Token

Diese Klasse ist für die Arbeit mit Tokens vorgesehen.

Objektreferenzen:

- Token

Verfügbare Methoden:

- token.create - neue Tokens erstellen
- token.delete - Tokens löschen
- token.get - Tokens abrufen
- token.update - Tokens aktualisieren
- token.generate - Tokens generieren

Token-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `token` API.

Token

Das Token-Objekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
tokenid	ID	ID des Tokens. Eigenschaftsverhalten: - *schreibgeschützt* - für Aktualisierungsvorgänge *erforderlich*
name	string	Name des Tokens. Eigenschaftsverhalten: - für Erstellungsvorgänge *erforderlich*
description	text	Beschreibung des Tokens.
userid	ID	ID des Benutzers, dem das Token zugewiesen wurde. Standard: *aktueller Benutzer*. Eigenschaftsverhalten: - *konstant*
lastaccess	timestamp	Datum und Uhrzeit der letzten Authentifizierung des Tokens. ”0”, wenn das Token nie authentifiziert wurde. Eigenschaftsverhalten: - *schreibgeschützt*
status	integer	Token-Status. Mögliche Werte: 0 - *(Standard)* aktiviertes Token; 1 - deaktiviertes Token.

Eigenschaft	Typ	Beschreibung
expires_at	timestamp	Ablaufdatum und -uhrzeit des Tokens. ”0” für Tokens ohne Ablaufdatum.
created_at	timestamp	Erstellungsdatum und -uhrzeit des Tokens. Eigenschaftsverhalten: - *schreibgeschützt*
creator_userid	ID	ID des Benutzers, der das Token erstellt hat. Eigenschaftsverhalten: - *schreibgeschützt*

token.create

Beschreibung

`object token.create(object/array tokens)`

Diese Methode ermöglicht es Ihnen, neue Tokens zu erstellen.

Note:
Die Berechtigung *API-Tokens verwalten* ist erforderlich, damit die Benutzerrolle Tokens für andere Benutzer verwalten kann.

Attention:
Ein mit dieser Methode erstelltes Token muss außerdem generiert werden, bevor es verwendet werden kann.

Parameter

`(object/array)` Zu erstellende Token.

Die Methode akzeptiert Token mit den Standard-Token-Eigenschaften.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Token unter der Eigenschaft `tokenids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Token.

Beispiele

Ein Token erstellen

Erstellen Sie ein aktiviertes Token, das nie abläuft und den Benutzer mit der ID 2 authentifiziert.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "token.create",
    "params": {
        "name": "Your token",
        "userid": "2"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "188"
        ]
    },
```

```
    "id": 1
}
```

Erstellen Sie ein deaktiviertes Token, das am 21. Januar 2021 abläuft. Dieses Token authentifiziert den aktuellen Benutzer.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "token.create",
    "params": {
        "name": "Your token",
        "status": "1",
        "expires_at": "1611238072"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "189"
        ]
    },
    "id": 1
}
```

Quelle

CToken::create() in *ui/include/classes/api/services/CToken.php*.

token.delete

Beschreibung

`object token.delete(array tokenids)`

Diese Methode ermöglicht es Ihnen, Tokens zu löschen.

> **Note:**
> Die Berechtigung *API-Tokens verwalten* ist für die Benutzerrolle erforderlich, um Tokens für andere Benutzer zu verwalten.

Parameter

`(array)` IDs der zu löschenden Token.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Token unter der Eigenschaft `tokenids` enthält.

Beispiele

Mehrere Token löschen

Löschen Sie zwei Token.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "token.delete",
    "params": [
        "188",
        "192"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "188",
            "192"
        ]
    },
    "id": 1
}
```

Quelle

CToken::delete() in *ui/include/classes/api/services/CToken.php*.

token.generate

Beschreibung

`object token.generate(array tokenids)`

Diese Methode ermöglicht es Ihnen, Tokens zu generieren.

Note:
Die Berechtigung *API-Tokens verwalten* ist für die Benutzerrolle erforderlich, um Tokens für andere Benutzer zu verwalten.

Attention:
Ein Token kann mit dieser Methode nur generiert werden, wenn es erstellt wurde.

Parameter

`(array)` IDs der zu generierenden Token.

Rückgabewerte

`(array)` Gibt ein Array von Objekten zurück, das die ID des generierten Tokens in der Eigenschaft `tokenid` und die generierte Autorisierungszeichenfolge in der Eigenschaft `token` enthält.

Eigenschaft	Type	Beschreibung
tokenid	ID	ID des Tokens.
token	string	Die generierte Autorisierungszeichenfolge für dieses Token.

Beispiele

Mehrere Token generieren

Generieren Sie zwei Token.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "token.generate",
    "params": [
        "1",
        "2"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
```

```
        {
            "tokenid": "1",
            "token": "bbcfce79a2d95037502f7e9a534906d3466c9a1484beb6ea0f4e7be28e8b8ce2"
        },
        {
            "tokenid": "2",
            "token": "fa1258a83d518eabd87698a96bd7f07e5a6ae8aeb8463cae33d50b91dd21bd6d"
        }
    ],
    "id": 1
}
```

Quelle

CToken::generate() in *ui/include/classes/api/services/CToken.php*.

token.get

Beschreibung

`integer/array token.get(object parameters)`

Die Methode ermöglicht es Ihnen, Tokens anhand der angegebenen Parameter abzurufen.

> **Note:**
> Nur der Benutzertyp *Super admin* darf Tokens für andere Benutzer anzeigen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
tokenids	ID/array	Gibt nur Token mit den angegebenen IDs zurück.
userids	ID/array	Gibt nur Token zurück, die für die angegebenen Benutzer erstellt wurden.
token	string	Gibt nur Token zurück, die für das angegebene *Auth token* erstellt wurden.
valid_at	timestamp	Gibt nur Token zurück, die zum angegebenen Datum und zur angegebenen Uhrzeit gültig (nicht abgelaufen) sind.
expired_at	timestamp	Gibt nur Token zurück, die zum angegebenen Datum und zur angegebenen Uhrzeit abgelaufen (nicht gültig) sind.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `tokenid`, `name`, `lastaccess`, `status`, `expires_at`, `created_at`.
countOutput	boolean	Diese Parameter werden im Referenzkommentar beschrieben.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Gibt entweder Folgendes zurück:

- ein Array von Objekten;

- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Ein Token abrufen

Rufen Sie alle Daten für das Token mit der ID „2“ ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "token.get",
    "params": {
        "output": "extend",
        "tokenids": "2"
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```

Quelle

CToken::get() in *ui/include/classes/api/services/CToken.php*.

token.update

Beschreibung

`object token.update(object/array tokens)`

Diese Methode ermöglicht es Ihnen, vorhandene Tokens zu aktualisieren.

> **Note:**
> Die Berechtigung *API-Tokens verwalten* ist erforderlich, damit die Benutzerrolle Tokens für andere Benutzer verwalten kann.

Parameter

`(object/array)` Zu aktualisierende Token-Eigenschaften.

Die Eigenschaft `tokenid` muss für jeden Token definiert sein, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Die Methode akzeptiert Token mit den Standard-Token-Eigenschaften.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Token unter der Eigenschaft `tokenids` enthält.

Beispiele

Ablauf des Tokens entfernen

Entfernen Sie das Ablaufdatum des Tokens.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "token.update",
    "params": {
        "tokenid": "2",
        "expires_at": "0"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "2"
        ]
    },
    "id": 1
}
```

Quelle

CToken::update() in *ui/include/classes/api/services/CToken.php*.

Trend

Diese Klasse ist für die Arbeit mit Trenddaten vorgesehen.

Objektreferenzen:

- Float-Trend
- Integer-Trend

Verfügbare Methoden:

- trend.get - Trends abrufen

Trend-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `trend` API.

> **Note:**
> Trend objects differ depending on the item's type of information. They are created by the Zabbix server and cannot be modified via the API.

Float-Trend

Das Float-Trend-Objekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
clock	timestamp	Zeitstempel einer Stunde, für die der Wert berechnet wurde. Zum Beispiel bedeutet der Zeitstempel „04:00:00“ Werte, die für den Zeitraum „04:00:00-04:59:59“ berechnet wurden.
itemid	ID	ID des zugehörigen Datenpunkts.
num	integer	Anzahl der Werte, die für die Stunde verfügbar waren.
value_min	float	Stündlicher Minimalwert.
value_avg	float	Stündlicher Durchschnittswert.
value_max	float	Stündlicher Maximalwert.

Integer-Trend

Das Integer-Trend-Objekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
clock	timestamp	Zeitstempel einer Stunde, für die der Wert berechnet wurde. Zum Beispiel bedeutet der Zeitstempel „04:00:00“ Werte, die für den Zeitraum „04:00:00-04:59:59“ berechnet wurden.
itemid	ID	ID des zugehörigen Datenpunkts.
num	integer	Anzahl der Werte, die für die Stunde verfügbar waren.
value_min	integer	Stündlicher Minimalwert.
value_avg	integer	Stündlicher Durchschnittswert.
value_max	integer	Stündlicher Maximalwert.

trend.get

Beschreibung

`integer/array trend.get(object parameters)`

Die Methode ermöglicht es Ihnen, Trenddaten anhand der angegebenen Parameter abzurufen.

Note:
Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Typ	Beschreibung
itemids	ID/array	Gibt nur Trends mit den angegebenen Datenpunkt-IDs zurück.
time_from	timestamp	Gibt nur Werte zurück, die nach oder zum angegebenen Zeitpunkt erfasst wurden.
time_till	timestamp	Gibt nur Werte zurück, die vor oder zum angegebenen Zeitpunkt erfasst wurden.
countOutput	boolean	Zählt die Anzahl der abgerufenen Objekte.
limit	integer	Begrenzt die Anzahl der abgerufenen Objekte.
output	query	Legt die Eigenschaften des Trend-Objekts fest, die zurückgegeben werden sollen.

Rückgabewerte

`(integer/array)` Gibt entweder Folgendes zurück:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Abrufen von Datenpunkt-Trenddaten

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "trend.get",
    "params": {
        "output": [
            "itemid",
            "clock",
            "num",
            "value_min",
            "value_avg",
```

```
            "value_max"
        ],
        "itemids": [
            "23715"
        ],
        "limit": "1"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "23715",
            "clock": "1446199200",
            "num": "60",
            "value_min": "0.165",
            "value_avg": "0.2168",
            "value_max": "0.35"
        }
    ],
    "id": 1
}
```

Quelle

CTrend::get() in *ui/include/classes/api/services/CTrend.php*.

Verlauf

Diese Klasse ist für die Arbeit mit Verlaufsdaten vorgesehen.

Objektreferenzen:

- Float-Verlauf
- Integer-Verlauf
- String-Verlauf
- Text-Verlauf
- Log-Verlauf

Verfügbare Methoden:

- history.clear - Verlaufsdaten löschen
- history.get - Verlaufsdaten abrufen
- history.push - Verlaufsdaten an den Zabbix-Server senden

Geschichts Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit dem `history` API.

> **Note:**
> History-Objekte unterscheiden sich je nach Art der Information. Sie werden vom Zabbix-Server erstellt und können nicht über die API geändert werden.

Float-Verlauf

Das Float-Verlaufsobjekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
clock	timestamp	Zeitpunkt, zu dem dieser Wert empfangen wurde.
itemid	ID	ID des zugehörigen Datenpunkts.
ns	integer	Nanosekunden, zu denen der Wert empfangen wurde.

Eigenschaft	Typ	Beschreibung
value	float	Empfangener Wert.

Ganzzahlverlauf

Das Ganzzahlverlaufsobjekt hat die folgenden Eigenschaften.

Eigenschaft	Type	Beschreibung
clock	timestamp	Zeitpunkt, zu dem dieser Wert empfangen wurde.
itemid	ID	ID des zugehörigen Datenpunkts.
ns	integer	Nanosekunden, zu denen der Wert empfangen wurde.
value	integer	Empfangener Wert.

String-Verlauf

Das String-Verlaufsobjekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
clock	timestamp	Zeitpunkt, zu dem dieser Wert empfangen wurde.
itemid	ID	ID des zugehörigen Datenpunkts.
ns	integer	Nanosekunden zum Zeitpunkt des Empfangs des Werts.
value	string	Empfangener Wert.

Textverlauf

Das Textverlaufsobjekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
id	ID	ID des Verlaufseintrags.
clock	timestamp	Zeitpunkt, zu dem dieser Wert empfangen wurde.
itemid	ID	ID des zugehörigen Datenpunkts.
ns	integer	Nanosekunden, zu denen der Wert empfangen wurde.
value	text	Empfangener Wert.

Log-Verlauf

Das Log-Verlaufsobjekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
id	ID	ID des Verlaufseintrags.
clock	timestamp	Zeitpunkt, zu dem dieser Wert empfangen wurde.
itemid	ID	ID des zugehörigen Datenpunkts.
logeventid	integer	ID des Windows-Ereignisprotokolleintrags.
ns	integer	Nanosekunden zum Zeitpunkt des Empfangs des Werts.
severity	integer	Stufe des Windows-Ereignisprotokolleintrags.
source	string	Quelle des Windows-Ereignisprotokolleintrags.
timestamp	timestamp	Zeit des Windows-Ereignisprotokolleintrags.
value	text	Empfangener Wert.

Binärverlauf

Das Binärverlaufsobjekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
clock	timestamp	Zeitpunkt, zu dem dieser Wert empfangen wurde.
itemid	ID	ID des zugehörigen Datenpunkts.
ns	integer	Nanosekunden, zu denen der Wert empfangen wurde.

Eigenschaft	Typ	Beschreibung
value	text	Empfangener Wert.

JSON-Verlauf

Das JSON-Verlaufsobjekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
clock	timestamp	Zeitpunkt, zu dem dieser Wert empfangen wurde.
itemid	ID	ID des zugehörigen Datenpunkts.
ns	integer	Nanosekunden zum Zeitpunkt des Empfangs des Werts.
value	text	Empfangener Wert.

history.clear

Beschreibung

`object history.clear(array itemids)`

Diese Methode ermöglicht es Ihnen, den Verlauf von Datenpunkten zu löschen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der zu löschenden Datenpunkte.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Datenpunkte unter der Eigenschaft `itemids` enthält.

Beispiele

Verlauf löschen

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "history.clear",
    "params": [
        "10325",
        "13205"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "10325",
            "13205"
        ]
    },
    "id": 1
}
```

Quelle

CHistory::clear() in *ui/include/classes/api/services/CHistory.php*.

history.get

Beschreibung

`integer/array history.get(object parameters)`

Die Methode ermöglicht es Ihnen, Verlaufsdaten gemäß den angegebenen Parametern abzurufen.

Attention:
Diese Methode kann historische Daten einer gelöschten Entität zurückgeben, wenn diese Daten vom Housekeeper noch nicht entfernt wurden.

Note:
Diese Methode ist für Benutzer aller Typen verfügbar. Berechtigungen zum Aufruf der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Description
history	integer	History-Objekttypen, die zurückgegeben werden sollen. Mögliche Werte: 0 - numerischer Float; 1 - Zeichen; 2 - Log; 3 - *(Standard)* numerisch ohne Vorzeichen; 4 - Text; 5 - binär; 6 - JSON.
hostids	ID/array	Nur History von den angegebenen Hosts zurückgeben.
itemids	ID/array	Nur History von den angegebenen Datenpunkten zurückgeben.
maxValueSize	integer	Maximale Anzahl von Bytes, die in der Eigenschaft `value` zurückgegeben werden sollen. Möglicher Wertebereich: 1-134217728 (1 Byte bis 128 MiB). Wenn auf `null` gesetzt, wird kein Limit angewendet. Standard: 65536 (64 KiB). Parameterverhalten: - *unterstützt*, wenn `history` auf ”binary” oder ”JSON” gesetzt ist
time_from	timestamp	Nur Werte zurückgeben, die nach oder zu dem angegebenen Zeitpunkt empfangen wurden.
time_till	timestamp	Nur Werte zurückgeben, die vor oder zu dem angegebenen Zeitpunkt empfangen wurden.
sortfield	string/array	Das Ergebnis nach den angegebenen Eigenschaften sortieren. Mögliche Werte: `itemid`, `clock`, `ns`. Unterstützt keine Sortierung nach `ns`, wenn History-Daten in ClickHouse gespeichert sind.

Parameter	Type	Description
search	object	Ergebnisse zurückgeben, die dem angegebenen Muster entsprechen (Groß-/Kleinschreibung wird nicht berücksichtigt). Akzeptiert ein Objekt, bei dem die Schlüssel Eigenschaftsnamen und die Werte Suchzeichenfolgen sind. Wenn keine zusätzlichen Optionen angegeben sind, wird eine `LIKE "%…%"`-Suche ausgeführt. Unterstützt keine Suche nach `value`, wenn `history` auf 6 (JSON) gesetzt ist.
countOutput	boolean	Diese Parameter werden in der Referenzdokumentation beschrieben.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Kann die folgenden Dinge zurück geben:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Abrufen von Datenpunkt-Verlaufsdaten

Gibt die 10 zuletzt empfangenen Werte eines numerischen (float) Datenpunkts zurück.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "history.get",
    "params": {
        "output": "extend",
        "history": 0,
        "itemids": "23296",
        "sortfield": "clock",
        "sortorder": "DESC",
        "limit": 10
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```


Quelle

CHistory::get() in *ui/include/classes/api/services/CHistory.php*.

history.push

Beschreibung

`object history.push(object/array itemHistoryData)`

Mit dieser Methode können Datenpunkt-Verlaufsdaten an den Zabbix-Server gesendet werden.

Note:
Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu sendende Datenpunkt-Verlaufsdaten.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
itemid	ID	ID des zugehörigen Datenpunkts. Parameter behavior: - *erforderlich*, wenn `host` und `key` nicht gesetzt sind
host	string	Technischer Name des Hosts. Parameter behavior: - *erforderlich*, wenn `itemid` nicht gesetzt ist
key	string	Datenpunktschlüssel. Parameter behavior: - *erforderlich*, wenn `itemid` nicht gesetzt ist
value	mixed	Datenpunktwert. Parameter behavior: - *erforderlich*
clock	timestamp	Zeitpunkt, zu dem der Wert empfangen wurde.
ns	integer	Nanosekunden, zu denen der Wert empfangen wurde.

Rückgabewerte

`(object)` Gibt das Ergebnis des Datensendevorgangs zurück.

Beispiele

Verlaufsdaten von Datenpunkten senden

Senden Sie Verlaufsdaten von Datenpunkten für die Datenpunkte „10600“, „10601“ und „999999“ an den Zabbix Server.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "response": "success",
        "data": [
            {
                "itemid": "10600"
            },
            {
                "itemid": "10600"
            },
            {
                "itemid": "10601",
                "error": "Datenpunkt ist deaktiviert."
            },
            {
                "error": "Keine Berechtigungen für das referenzierte Objekt oder es existiert nicht."
            }
        ]
    },
    "id": 1
}
```

Siehe auch

- Trapper Items
- HTTP Agent items
- Host
- Item

Quelle

CHistory::push() in *ui/include/classes/api/services/CHistory.php*.

Vorlage

Diese Klasse ist für die Arbeit mit Vorlagen ausgelegt.

Objektreferenzen:

- Vorlage
- Vorlagen-Tag

Verfügbare Methoden:

- template.create - neue Vorlagen erstellen
- template.delete - Vorlagen löschen
- template.get - Vorlagen abrufen
- template.massadd - verknüpfte Objekte zu Vorlagen hinzufügen
- template.massremove - verknüpfte Objekte aus Vorlagen entfernen
- template.update - Vorlagen aktualisieren

Template-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `template` API.

Vorlage

Das Vorlagenobjekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
templateid	ID	ID der Vorlage. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
description	text	Beschreibung der Vorlage.
host	string	Technischer Name der Vorlage. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
name	string	Sichtbarer Name der Vorlage. Standard: Wert der Eigenschaft `host`.
readme	text	Vorlagenspezifische Konfigurationsanweisungen, die im Host-Assistenten angezeigt werden. Unterstützt Markdown-Formatierung.
uuid	string	Universell eindeutige Kennung, die verwendet wird, um importierte Vorlagen mit bereits vorhandenen zu verknüpfen. Wird automatisch generiert, wenn sie nicht angegeben wird.
vendor_name	string	Name des Vorlagenanbieters. Bei Erstellungsvorgängen sollten sowohl `vendor_name` als auch `vendor_version` entweder gesetzt oder leer gelassen werden. Bei Aktualisierungsvorgängen kann `vendor_version` leer gelassen werden, wenn es in der Datenbank einen Wert hat.
vendor_version	string	Version des Vorlagenanbieters. Bei Erstellungsvorgängen sollten sowohl `vendor_name` als auch `vendor_version` entweder gesetzt oder leer gelassen werden. Bei Aktualisierungsvorgängen kann `vendor_name` leer gelassen werden, wenn es in der Datenbank einen Wert hat.
wizard_ready	integer	Gibt an, ob die Vorlage im Host-Assistenten zur Auswahl verfügbar ist. Mögliche Werte: 0 - *(Standard)* Nicht verfügbar. 1 - Verfügbar.

Vorlagen-Tag

Das Vorlagen-Tag-Objekt hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
tag	string	Name des Vorlagen-Tags. Verhalten der Eigenschaft: - *erforderlich*
value	string	Wert des Vorlagen-Tags. Verhalten der Eigenschaft: - *schreibgeschützt*.
object	integer	Typ des Objekts, von dem das Tag geerbt wurde. Mögliche Werte: 0 - Vorlage. Verhalten der Eigenschaft: - *schreibgeschützt*.

Eigenschaft	Typ	Beschreibung
objectid	ID	ID des Objekts, von dem das Tag geerbt wurde. Verhalten der Eigenschaft: - *schreibgeschützt*.

template.create

Beschreibung

`object template.create(object/array templates)`

Diese Methode ermöglicht es Ihnen, neue Vorlagen zu erstellen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellende Vorlagen.

Zusätzlich zu den Standard-Vorlageneigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Typ	Beschreibung
groups	object/array	Vorlagengruppen, zu denen die Vorlage hinzugefügt werden soll. Für die Vorlagengruppen darf nur die Eigenschaft `groupid` definiert sein. Parameterverhalten: - *erforderlich*
tags	object/array	Vorlagen-Tags.
templates	object/array	Vorlagen, die mit der Vorlage verknüpft werden sollen. Für die Vorlagen darf nur die Eigenschaft `templateid` definiert sein.
macros	object/array	Benutzermakros, die für die Vorlage erstellt werden sollen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Vorlagen unter der Eigenschaft `templateids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Vorlagen.

Beispiele

Erstellen einer Vorlage

Erstellen Sie eine Vorlage mit Tags und verknüpfen Sie zwei Vorlagen mit dieser Vorlage.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "11117"
        ]
    },
    "id": 1
}
```

Quelle

CTemplate::create() in *ui/include/classes/api/services/CTemplate.php*.

template.get

Beschreibung

`integer/array template.get(object parameters)`

Die Methode ermöglicht es Ihnen, Vorlagen anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
templateids	ID/array	Gibt nur Vorlagen mit den angegebenen Vorlagen-IDs zurück.
groupids	ID/array	Gibt nur Vorlagen zurück, die zu den angegebenen Vorlagengruppen gehören.
parentTemplateids	ID/array	Gibt nur Vorlagen zurück, mit denen die angegebene Vorlage verknüpft ist.
hostids	ID/array	Gibt nur Vorlagen zurück, die mit den angegebenen Hosts/Vorlagen verknüpft sind.
graphids	ID/array	Gibt nur Vorlagen zurück, die die angegebenen Diagramme enthalten.
itemids	ID/array	Gibt nur Vorlagen zurück, die die angegebenen Datenpunkte enthalten.
triggerids	ID/array	Gibt nur Vorlagen zurück, die die angegebenen Auslöser enthalten.
with_items	flag	Gibt nur Vorlagen zurück, die Datenpunkte haben.
with_triggers	flag	Gibt nur Vorlagen zurück, die Auslöser haben.
with_graphs	flag	Gibt nur Vorlagen zurück, die Diagramme haben.
with_httptests	flag	Gibt nur Vorlagen zurück, die Webszenarien haben.

Parameter	Type	Beschreibung
evaltype	integer	Auswertungsmethode für Tags. Mögliche Werte: 0 - *(Standard)* Und/Oder; 2 - Oder.
tags	object/array	Gibt nur Vorlagen mit den angegebenen Tags zurück. Format: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Ein leeres Array gibt alle Vorlagen zurück. Mögliche Werte für operator: 0 - *(Standard)* Enthält; 1 - Gleich; 2 - Enthält nicht; 3 - Ungleich; 4 - Existiert; 5 - Existiert nicht.
inheritedTags	boolean	Gibt Vorlagen zurück, die die angegebenen `tags` auch in verknüpften Vorlagen haben. Mögliche Werte: `true` - verknüpfte Vorlagen müssen die angegebenen Tags ebenfalls haben; `false` - *(Standard)* Tags aus verknüpften Vorlagen werden ignoriert.
selectInheritedTags	query	Gibt eine Eigenschaft `inheritedTags` mit Tags zurück, die sich auf verknüpften Vorlagen befinden.
selectTags	query	Gibt Vorlagen-Tags in der Eigenschaft `tags` zurück.
selectDiscoveryRules	query	Gibt eine Eigenschaft `discoveryRules` mit LLD-Regeln der Vorlage zurück. Unterstützt `count`.
selectHosts	query	Gibt die Hosts zurück, die mit der Vorlage verknüpft sind, in der Eigenschaft `hosts`. Unterstützt `count`.
selectTemplateGroups	query	Gibt die Vorlagengruppen, zu denen die Vorlage gehört, in der Eigenschaft `templategroups` zurück.
selectTemplates	query	Gibt Vorlagen zurück, mit denen die angegebene Vorlage verknüpft ist, in der Eigenschaft `templates`. Unterstützt `count`.
selectParentTemplates	query	Gibt Vorlagen zurück, die mit der angegebenen Vorlage verknüpft sind, in der Eigenschaft `parentTemplates`. Unterstützt `count`.
selectHttpTests	query	Gibt die Webszenarien aus der Vorlage in der Eigenschaft `httpTests` zurück. Unterstützt `count`.
selectItems	query	Gibt Datenpunkte aus der Vorlage in der Eigenschaft `items` zurück. Unterstützt `count`.
selectTriggers	query	Gibt Auslöser aus der Vorlage in der Eigenschaft `triggers` zurück. Unterstützt `count`.
selectGraphs	query	Gibt Diagramme aus der Vorlage in der Eigenschaft `graphs` zurück. Unterstützt `count`.
selectMacros	query	Gibt die Makros aus der Vorlage in der Eigenschaft `macros` zurück.

Parameter	Type	Beschreibung
selectDashboards	query	Gibt Dashboards aus der Vorlage in der Eigenschaft `dashboards` zurück. Unterstützt `count`.
selectValueMaps	query	Gibt eine Eigenschaft `valuemaps` mit Wertezuordnungen der Vorlage zurück.
limitSelects	integer	Begrenzt die Anzahl der von Unterabfragen zurückgegebenen Datensätze. Gilt für die folgenden Unterabfragen: `selectTemplates` - Ergebnisse werden nach `name` sortiert; `selectHosts` - sortiert nach `host`; `selectParentTemplates` - sortiert nach `host`; `selectItems` - sortiert nach `name`; `selectDiscoveryRules` - sortiert nach `name`; `selectTriggers` - sortiert nach `description`; `selectGraphs` - sortiert nach `name`; `selectDashboards` - sortiert nach `name`.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `hostid`, `host`, `name`, `status`.
countOutput	boolean	Diese Parameter werden in der Referenzbeschreibung beschrieben.
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	Gibt Low-Level-Discoverys aus der Vorlage in der Eigenschaft `discoveries` zurück. Unterstützt `count`. Diese Abfrage ist **veraltet**, bitte verwenden Sie stattdessen `selectDiscoveryRules`.

Rückgabewerte

`(integer/array)` Gibt entweder Folgendes zurück:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Vorlagen nach Namen abrufen

Rufen Sie alle Daten zu zwei Vorlagen mit den Namen „Linux by Zabbix Agent“ und „Windows by Zabbix Agent“ ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": "extend",
        "filter": {
            "host": [
                "Linux by Zabbix agent",
```

```
                "Windows by Zabbix agent"
            ]
        }
    },
    "id": 1
}
```

Antwort:


```
{
    "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": "Dies ist eine offizielle Linux-Vorlage. Sie erfordert Zabbix Agent 8.0 oder ne
            "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": "## Überblick\r\n\r\nDies ist eine offizielle Linux-Vorlage. Sie erfordert Zabbix Ag
        },
        {
            "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": "Dies ist eine offizielle Windows-Vorlage. Sie erfordert Zabbix Agent 8.0 oder
            "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": "## Überblick\r\n\r\nDies ist eine offizielle Windows-Vorlage. Sie erfordert Zabbix
        }
    ],
    "id": 1
}
```


Abrufen von Vorlagengruppen

Rufen Sie die Vorlagengruppen ab, deren Mitglied die Vorlage „Linux by Zabbix Agent“ ist.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": ["hostid"],
        "selectTemplateGroups": "extend",
        "filter": {
            "host": [
                "Linux by Zabbix agent"
            ]
        }
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10001",
            "templategroups": [
                {
                    "groupid": "10",
                    "name": "Templates/Operating systems",
                    "uuid": "846977d1dfed4968bc5f8bdb363285bc"
                }
            ]
        }
    ],
    "id": 1
}
```

Hosts nach Vorlage abrufen

Rufen Sie Hosts ab, mit denen die Vorlage „10001“ (*Linux by Zabbix agent*) verknüpft ist.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": "templateid",
        "templateids": "10001",
```

```
        "selectHosts": ["hostid", "name"]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10001",
            "hosts": [
                {
                    "hostid": "10084",
                    "name": "Zabbix server"
                },
                {
                    "hostid": "10603",
                    "name": "Host 1"
                },
                {
                    "hostid": "10604",
                    "name": "Host 2"
                }
            ]
        }
    ],
    "id": 1
}
```

Suche nach Vorlagen-Tags

Rufen Sie Vorlagen ab, die das Tag "host-name" mit dem Wert "{HOST.NAME}" haben.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": ["hostid"],
        "selectTags": "extend",
        "evaltype": 0,
        "tags": [
            {
                "tag": "host-name",
                "value": "{HOST.NAME}",
                "operator": 1
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10402",
            "tags": [
                {
                    "tag": "host-name",
                    "value": "{HOST.NAME}"
```

```
                }
            ]
        }
    ],
    "id": 1
}
```

Siehe auch

- Vorlagengruppe
- Vorlage
- Benutzermakro
- Host-Schnittstelle

Quelle

CTemplate::get() in *ui/include/classes/api/services/CTemplate.php*.

template.massadd

Beschreibung

`object template.massadd(object parameters)`

Diese Methode ermöglicht es Ihnen, mehrere zusammengehörige Objekte gleichzeitig zu den angegebenen Vorlagen hinzuzufügen.

Note:
Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die IDs der zu aktualisierenden Vorlagen und die Objekte enthalten, die den Vorlagen hinzugefügt werden sollen.

Die Methode akzeptiert die folgenden Parameter.

Parameter	Type	Beschreibung
templates	object/array	Vorlagen, die aktualisiert werden sollen. Für die Vorlagen darf nur die Eigenschaft `templateid` definiert sein. Parameter behavior: - *erforderlich*
groups	object/array	Vorlagengruppen, zu denen die angegebenen Vorlagen hinzugefügt werden sollen. Für die Vorlagengruppen darf nur die Eigenschaft `groupid` definiert sein.
macros	object/array	Benutzermakros, die für die angegebenen Vorlagen erstellt werden sollen.
templates_link	object/array	Vorlagen, die mit den angegebenen Vorlagen verknüpft werden sollen. Für die Vorlagen darf nur die Eigenschaft `templateid` definiert sein.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Vorlagen unter der Eigenschaft `templateids` enthält.

Beispiele

Eine Gruppe mit Vorlagen verknüpfen

Fügen Sie die Vorlagengruppe „2“ zu zwei Vorlagen hinzu.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "template.massadd",
    "params": {
        "templates": [
            {
                "templateid": "10085"
            },
            {
                "templateid": "10086"
            }
        ],
        "groups": [
            {
                "groupid": "2"
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10085",
            "10086"
        ]
    },
    "id": 1
}
```

Zwei Vorlagen mit einer Vorlage verknüpfen

Verknüpfen Sie die Vorlagen „10106“ und „10104“ mit der Vorlage „10073“.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "template.massadd",
    "params": {
        "templates": [
            {
                "templateid": "10073"
            }
        ],
        "templates_link": [
            {
                "templateid": "10106"
            },
            {
                "templateid": "10104"
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "templateids": [
            "10073"
        ]
    },
    "id": 1
}
```

Siehe auch

- template.update
- Host
- Vorlagengruppe
- Benutzermakro

Quelle

CTemplate::massAdd() in *ui/include/classes/api/services/CTemplate.php*.

template.massremove

Beschreibung

`object template.massremove(object parameters)`

Diese Methode ermöglicht es Ihnen, verknüpfte Objekte aus mehreren Vorlagen zu entfernen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Berechtigungen zum Aufruf der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die IDs der zu aktualisierenden Vorlagen und die Objekte enthalten, die entfernt werden sollen.

Parameter	Type	Beschreibung
templateids	ID/array	IDs der zu aktualisierenden Vorlagen. Parameterverhalten: - *erforderlich*
groupids	ID/array	IDs der Vorlagengruppen, aus denen die angegebenen Vorlagen entfernt werden sollen.
macros	string/array	IDs der Benutzermakros, die aus den angegebenen Vorlagen gelöscht werden sollen.
templateids_clear	ID/array	IDs der Vorlagen, deren Verknüpfung mit den angegebenen Vorlagen aufgehoben und aus diesen entfernt werden soll (upstream).
templateids_link	ID/array	IDs der Vorlagen, deren Verknüpfung mit den angegebenen Vorlagen aufgehoben werden soll (upstream).

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Vorlagen unter der Eigenschaft `templateids` enthält.

Beispiele

Vorlagen aus einer Gruppe entfernen

Entfernen Sie zwei Vorlagen aus der Gruppe „2“.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "template.massremove",
    "params": {
        "templateids": [
            "10085",
```

```
            "10086"
        ],
        "groupids": "2"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10085",
            "10086"
        ]
    },
    "id": 1
}
```

Verknüpfung von Vorlagen mit einem Host aufheben

Heben Sie die Verknüpfung der Vorlagen „10106“ und „10104“ mit der Vorlage „10085“ auf.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "template.massremove",
    "params": {
        "templateids": "10085",
        "templateids_link": [
            "10106",
            "10104"
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10085"
        ]
    },
    "id": 1
}
```

Siehe auch

- template.update
- Benutzermakro

Quelle

CTemplate::massRemove() in *ui/include/classes/api/services/CTemplate.php*.

template.update

Beschreibung

`object template.update(object/array templates)`

Diese Methode ermöglicht es Ihnen, vorhandene Vorlagen zu aktualisieren.

Note:
Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu aktualisierende Vorlageneigenschaften.

Die Eigenschaft `templateid` muss für jede Vorlage definiert sein, alle anderen Eigenschaften sind optional. Nur die angegebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Zusätzlich zu den Standard-Vorlageneigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
groups	object/array	Vorlagengruppen, die die aktuellen Vorlagengruppen ersetzen, denen die Vorlagen angehören. Für die Vorlagengruppen darf nur die Eigenschaft `groupid` definiert sein.
tags	object/array	Vorlagen-Tags, die die aktuellen Vorlagen-Tags ersetzen.
macros	object/array	Benutzermakros, die die aktuellen Benutzermakros in den angegebenen Vorlagen ersetzen.
templates	object/array	Vorlagen, die die aktuell verknüpften Vorlagen ersetzen. Vorlagen, die nicht übergeben werden, werden nur getrennt. Für die Vorlagen darf nur die Eigenschaft `templateid` definiert sein.
templates_clear	object/array	Vorlagen, die von den angegebenen Vorlagen getrennt und daraus entfernt werden. Für die Vorlagen darf nur die Eigenschaft `templateid` definiert sein.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Vorlagen unter der Eigenschaft `templateids` enthält.

Beispiele

Ändern der Standard-Eigenschaften einer Vorlage

Ändern Sie den technischen Namen der Vorlage in „Linux by Zabbix agent Custom“, den sichtbaren Namen in „My template“ und aktualisieren Sie die Beschreibung der Vorlage.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
```

```
    "id": 1
}
```

Vorlagengruppen aktualisieren

Ersetzen Sie alle Vorlagengruppen für die angegebene Vorlage durch eine andere.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "groups": [
            {
                "groupid": "24"
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

Aktualisieren mehrerer Vorlagengruppen

Ersetzen Sie alle Vorlagengruppen für mehrere Vorlagen durch andere.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086",
            "10087",
            "10088"
        ]
    },
    "id": 1
}
```

Aktualisieren von Vorlagen-Tags

Ersetzen Sie alle Vorlagen-Tags durch ein anderes.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "tags": [
            {
                "tag": "host-name",
                "value": "{HOST.NAME}"
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

Aktualisieren von Makros in Vorlagen

Ersetzen Sie alle Makros der Vorlage durch ein anderes.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
```

```
    "params": {
        "templateid": "10086",
        "macros": [
            {
                "macro": "{$MY_MACRO}",
                "value": "new_value"
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

Mehrere Vorlagenmakros aktualisieren

Ersetzen Sie alle Benutzermakros in mehreren Vorlagen durch das angegebene Benutzermakro.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": [
            {
                "templateid": "10087",
                "macros": [
                    {
                        "macro": "{$AGENT.TIMEOUT}",
                        "value": "5m",
                        "description": "Zeitüberschreitung, nach der der Agent als nicht verfügbar betrach
                    }
                ]
            },
            {
                "templateid": "10088",
                "macros": [
                    {
                        "macro": "{$AGENT.TIMEOUT}",
                        "value": "5m",
                        "description": "Zeitüberschreitung, nach der der Agent als nicht verfügbar betrach
                    }
                ]
            }
        ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10087",
            "10088"
```

```
        ]
    },
    "id": 1
}
```

Verknüpfte Vorlagen einer Vorlage aktualisieren

Heben Sie die Verknüpfung aller Vorlagen von der angegebenen Vorlage auf (ohne sie zu löschen) und verknüpfen Sie stattdessen eine andere Vorlage damit.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "templates": [
            {
                "templateid": "10001"
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

Mehrere mit Vorlagen verknüpfte Vorlagen bereinigen

Verknüpfen Sie eine bestimmte verknüpfte Vorlage von den angegebenen Vorlagen ab und bereinigen Sie sie.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": [
        {
            "templateid": "10087",
            "templates_clear": [
                {
                    "templateid": "10001"
                }
            ]
        },
        {
            "templateid": "10088",
            "templates_clear": [
                {
                    "templateid": "10001"
                }
            ]
        }
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10087",
            "10088"
        ]
    },
    "id": 1
}
```

Quelle

CTemplate::update() in *ui/include/classes/api/services/CTemplate.php*.

vorlage.delete

Beschreibung

`object template.delete(array templateIds)`

Diese Methode ermöglicht es Ihnen, Vorlagen zu löschen.

Beim Löschen einer Vorlage werden auch alle zugehörigen Entitäten (Datenpunkte, Auslöser, Diagramme usw.) von allen Hosts oder Vorlagen entfernt, mit denen sie verknüpft ist. Um diese Entitäten zu behalten, lösen Sie die Vorlage zunächst von den betreffenden Hosts oder Vorlagen mit den Methoden `host.update`, `host.massremove`, `template.update` oder `template.massremove` ab.

Note:
Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der zu löschenden Vorlagen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Vorlagen unter der Eigenschaft `templateids` enthält.

Beispiele

Mehrere Vorlagen löschen

Löschen Sie zwei Vorlagen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "template.delete",
    "params": [
        "13",
        "32"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "13",
            "32"
        ]
    },
```

```
    "id": 1
}
```

Quelle

CTemplate::delete() in *ui/include/classes/api/services/CTemplate.php*.

Vorlagen-Dashboard

Diese Klasse ist für die Arbeit mit Vorlagen-Dashboards konzipiert.

Objektreferenzen:

- Vorlagen-Dashboard
- Vorlagen-Dashboard-Seite
    - Vorlagen-Dashboard-Widget
        * Vorlagen-Dashboard-Widget-Feld

Verfügbare Methoden:

- templatedashboard.create - neue Vorlagen-Dashboards erstellen
- templatedashboard.delete - Vorlagen-Dashboards löschen
- templatedashboard.get - Vorlagen-Dashboards abrufen
- templatedashboard.update - Vorlagen-Dashboards aktualisieren

Vorlagen Dashboard-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `templatedashboard` API.

Vorlage-Dashboard

Das Objekt des Vorlage-Dashboards hat die folgenden Eigenschaften.

Property	Type	Description
dashboardid	ID	ID des Vorlage-Dashboards. Property behavior: - *read-only* - *required* für Aktualisierungsoperationen
name	string	Name des Vorlage-Dashboards. Property behavior: - *required* für Erstellungsoperationen
templateid	ID	ID der Vorlage, zu der das Dashboard gehört. Property behavior: - *constant* - *required* für Erstellungsoperationen
display_period	integer	Standardintervall der Diashow (in Sekunden). Mögliche Werte: 10, 30, 60, 120, 600, 1800, 3600. Standard: 30.
auto_start	integer	Diashow automatisch starten. Mögliche Werte: 0 - Diashow nicht automatisch starten; 1 - *(Standard)* Diashow automatisch starten.
uuid	string	Universell eindeutige Kennung, die zum Verknüpfen importierter Vorlage-Dashboards mit bereits vorhandenen verwendet wird. Wird automatisch generiert, wenn nicht angegeben.

Vorlage-Dashboard-Seite

Das Objekt der Vorlage-Dashboard-Seite hat die folgenden Eigenschaften.

Property	Type	Description
dashboard_pageid	ID	ID der Dashboard-Seite. Property behavior: - *read-only*
name	string	Name der Dashboard-Seite. Standard: leerer String.
display_period	integer	Intervall der Dashboard-Diashow (in Sekunden). Mögliche Werte: 0, 10, 30, 60, 120, 600, 1800, 3600. Standard: 0 (verwendet das Standard-Diashow-Intervall).
widgets	array	Array der Objekte vom Typ template dashboard widget.

Widget für Vorlagen-Dashboards

Das Widget-Objekt für Vorlagen-Dashboards hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
widgetid	ID	ID des Dashboard-Widgets. Verhalten der Eigenschaft: - *schreibgeschützt*

Eigenschaft	Typ	Beschreibung
type	string	Typ des Dashboard-Widgets. Mögliche Werte: actionlog - Aktionsprotokoll; clock - Uhr; discovery - Discovery-Status; favgraphs - Bevorzugte Graphen; favmaps - Bevorzugte Karten; gauge - Messanzeige; graph - Graph (klassisch); graphprototype - Graph-Prototyp; honeycomb - Wabenansicht; hostavail - Host-Verfügbarkeit; hostcard - Host-Karte; hostnavigator - Host-Navigator; itemcard - Datenpunkt-Karte; itemnavigator - Datenpunkt-Navigator; item - Datenpunkt-Wert; map - Karte; navtree - Karten-Navigationsbaum; piechart - Kreisdiagramm; plaintext - Klartext; problemhosts - Problem-Hosts; problems - Probleme; problemsbysv - Probleme nach Schweregrad; scatterplot - Streudiagramm; slareport - SLA-Bericht; svggraph - Graph; systeminfo - Systeminformationen; tophosts - Top-Hosts; topitems - Top-Datenpunkte; toptriggers - Top-Auslöser; trigover - Auslöser-Übersicht; url - URL; web - Web-Überwachung. Verhalten der Eigenschaft: - *erforderlich*
name	string	Benutzerdefinierter Widget-Name.
x	integer	Eine horizontale Position von der linken Seite des Dashboards aus. Mögliche Werte reichen von 0 bis 71.
y	integer	Eine vertikale Position vom oberen Rand des Dashboards aus. Mögliche Werte reichen von 0 bis 63.
width	integer	Die Widget-Breite. Mögliche Werte reichen von 1 bis 72.
height	integer	Die Widget-Höhe. Mögliche Werte reichen von 1 bis 64.
view_mode	integer	Der Widget-Anzeigemodus. Mögliche Werte: 0 - *(Standard)* Standard-Widget-Ansicht; 1 - mit ausgeblendeter Kopfzeile;
fields	array	Array von Objekten des Typs Feld des Widgets für Vorlagen-Dashboards.

Feld für Vorlage-Dashboard-Widget

Das Feldobjekt für Vorlage-Dashboard-Widgets hat die folgenden Eigenschaften.

Property	Type	Description
type	integer	Typ des Widget-Feldes. Mögliche Werte: 0 - Ganzzahl; 1 - Zeichenfolge; 4 - Datenpunkt; 5 - Datenpunkt-Prototyp; 6 - Graph; 7 - Graph-Prototyp; 8 - Karte; 9 - Service; 10 - SLA; 11 - Benutzer; 12 - Aktion; 13 - Medientyp. Property behavior: - *required*
name	string	Name des Widget-Feldes. Mögliche Werte: siehe Dashboard-Widget-Felder. Beachten Sie, dass einige Host-bezogene Parameter (z. B. *Host-Gruppen*, *Host-Gruppen ausschließen* und *Hosts* im Widget *Probleme*, *Host-Gruppen* im Widget *Host-Verfügbarkeit* usw.) beim Konfigurieren des Widgets in einem Vorlage-Dashboard nicht verfügbar sind. Der Grund dafür ist, dass Vorlage-Dashboards nur Daten von dem Host anzeigen, mit dem die Vorlage verknüpft ist. Property behavior: - *required*
value	mixed	Wert des Widget-Feldes abhängig vom Typ. Mögliche Werte: siehe Dashboard-Widget-Felder. Beachten Sie, dass einige Host-bezogene Parameter (z. B. *Host-Gruppen*, *Host-Gruppen ausschließen* und *Hosts* im Widget *Probleme*, *Host-Gruppen* im Widget *Host-Verfügbarkeit* usw.) beim Konfigurieren des Widgets in einem Vorlage-Dashboard nicht verfügbar sind. Der Grund dafür ist, dass Vorlage-Dashboards nur Daten von dem Host anzeigen, mit dem die Vorlage verknüpft ist. Property behavior: - *required*

templatedashboard.create

Beschreibung

`object templatedashboard.create(object/array templateDashboards)`

Diese Methode ermöglicht es Ihnen, neue Vorlagen-Dashboards zu erstellen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellende Vorlagen-Dashboards.

Zusätzlich zu den Standard-Eigenschaften von Vorlagen-Dashboards akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
pages	array	Zu erstellende Vorlagen-Dashboard-Seiten für das Dashboard. Die Dashboard-Seiten werden in derselben Reihenfolge angeordnet, wie sie angegeben sind. Parameterverhalten: - *erforderlich*

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Vorlagen-Dashboards in der Eigenschaft `dashboardids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Vorlagen-Dashboards.

Beispiele

Erstellen eines Vorlagen-Dashboards

Erstellen Sie ein Vorlagen-Dashboard mit dem Namen „Graphs“ mit einem Graph-Widget auf einer einzelnen Dashboard-Seite.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "32"
        ]
    },
    "id": 1
}
```

Siehe auch

- Vorlagen-Dashboard-Seite
- Vorlagen-Dashboard-Widget
- Vorlagen-Dashboard-Widget-Feld

Quelle

CTemplateDashboard::create() in *ui/include/classes/api/services/CTemplateDashboard.php*.

templatedashboard.delete

Beschreibung

`object templatedashboard.delete(array templateDashboardIds)`

Diese Methode ermöglicht es Ihnen, Vorlagen-Dashboards zu löschen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der zu löschenden Vorlagen-Dashboards.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Vorlagen-Dashboards in der Eigenschaft `dashboardids` enthält.

Beispiele

Mehrere Vorlagen-Dashboards löschen

Löschen Sie zwei Vorlagen-Dashboards.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.delete",
    "params": [
        "45",
        "46"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "45",
            "46"
        ]
    },
    "id": 1
}
```

Quelle

CTemplateDashboard::delete() in *ui/include/classes/api/services/CTemplateDashboard.php*.

templatedashboard.get

Beschreibung

`integer/array templatedashboard.get(object parameters)`

Die Methode ermöglicht es Ihnen, Vorlagen-Dashboards anhand der angegebenen Parameter abzurufen.

Note:
Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
dashboardids	ID/array	Gibt nur Vorlagen-Dashboards mit den angegebenen IDs zurück.
templateids	ID/array	Gibt nur Vorlagen-Dashboards zurück, die zu den angegebenen Vorlagen gehören.
selectPages	query	Gibt eine Eigenschaft `pages` mit Vorlagen-Dashboard-Seiten in der korrekten Reihenfolge zurück.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `dashboardid`, `name`.
countOutput	boolean	Diese Parameter sind in der Referenzkommentierung beschrieben.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Gibt entweder Folgendes zurück:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Abrufen von Vorlagen-Dashboards

Rufen Sie alle Vorlagen-Dashboards mit Widgets für eine angegebene Vorlage ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.get",
    "params": {
        "output": "extend",
        "selectPages": "extend",
        "templateids": "10001"
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```


Siehe auch

- Vorlagen-Dashboard-Seite
- Vorlagen-Dashboard-Widget
- Vorlagen-Dashboard-Widget-Feld

Quelle

CTemplateDashboard::get() in *ui/include/classes/api/services/CTemplateDashboard.php*.

templatedashboard.update

Beschreibung

`object templatedashboard.update(object/array templateDashboards)`

Diese Methode ermöglicht es Ihnen, vorhandene Vorlagen-Dashboards zu aktualisieren.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

(object/array) Zu aktualisierende Eigenschaften der Vorlagen-Dashboards.

Die Eigenschaft dashboardid muss für jedes Dashboard angegeben werden, alle anderen Eigenschaften sind optional. Nur die angegebenen Eigenschaften werden aktualisiert.

Zusätzlich zu den standardmäßigen Eigenschaften von Vorlagen-Dashboards akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
pages	array	Vorlagen-Dashboard-Seiten zum Ersetzen der vorhandenen Dashboard-Seiten. Dashboard-Seiten werden über die Eigenschaft dashboard_pageid aktualisiert. Neue Dashboard-Seiten werden für Objekte ohne die Eigenschaft dashboard_pageid erstellt, und vorhandene Dashboard-Seiten werden gelöscht, wenn sie nicht wiederverwendet werden. Dashboard-Seiten werden in derselben Reihenfolge angeordnet, wie sie angegeben sind. Nur die angegebenen Eigenschaften der Dashboard-Seiten werden aktualisiert. Für die Eigenschaft pages ist mindestens ein Dashboard-Seitenobjekt erforderlich.

Rückgabewerte

(object) Gibt ein Objekt zurück, das die IDs der aktualisierten Vorlage-Dashboards unter der Eigenschaft dashboardids enthält.

Beispiele

Umbenennen eines Vorlagen-Dashboards

Benennen Sie ein Vorlagen-Dashboard in „Performance graphs“ um.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.update",
    "params": {
        "dashboardid": "23",
        "name": "Performance graphs"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "23"
        ]
    },
    "id": 1
}
```

Aktualisieren von Vorlagen-Dashboard-Seiten

Benennen Sie die erste Dashboard-Seite um, ersetzen Sie die Widgets auf der zweiten Dashboard-Seite und fügen Sie als dritte eine neue Seite hinzu. Löschen Sie alle anderen Dashboard-Seiten.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.update",
    "params": {
        "dashboardid": "2",
        "pages": [
```

```
            {
                "dashboard_pageid": 1,
                "name": "Renamed Page"
            },
            {
                "dashboard_pageid": 2,
                "widgets": [
                    {
                        "type": "clock",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3
                    }
                ]
            },
            {
                "display_period": 60
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

Siehe auch

- Vorlagen-Dashboard-Widget
- Feld des Vorlagen-Dashboard-Widgets

Quelle

CTemplateDashboard::update() in *ui/include/classes/api/services/CTemplateDashboard.php*.

Vorlagengruppe

Diese Klasse ist für die Arbeit mit Vorlagengruppen vorgesehen.

Objektreferenzen:

- Vorlagengruppe

Verfügbare Methoden:

- templategroup.create - neue Vorlagengruppen erstellen
- templategroup.delete - Vorlagengruppen löschen
- templategroup.get - Vorlagengruppen abrufen
- templategroup.massadd - verknüpfte Objekte zu Vorlagengruppen hinzufügen
- templategroup.massremove - verknüpfte Objekte aus Vorlagengruppen entfernen
- templategroup.propagate - Berechtigungen auf Untergruppen von Vorlagengruppen übertragen
- templategroup.update - Vorlagengruppen aktualisieren

Vorlagen Gruppen-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `templategroup` API.

Vorlagengruppe

Das Objekt der Vorlagengruppe hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
groupid	ID	ID der Vorlagengruppe. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
name	string	Name der Vorlagengruppe. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
uuid	string	Universell eindeutige Kennung, die verwendet wird, um importierte Vorlagengruppen mit bereits vorhandenen zu verknüpfen. Wird automatisch generiert, wenn sie nicht angegeben wird.

templategroup.create

Beschreibung

`object templategroup.create(object/array templateGroups)`

Diese Methode ermöglicht es Ihnen, neue Vorlagengruppen zu erstellen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellende Vorlagengruppen. Die Methode akzeptiert Vorlagengruppen mit den standardmäßigen Eigenschaften von Vorlagengruppen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Vorlagengruppen in der Eigenschaft `groupids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Vorlagengruppen.

Beispiele

Erstellen einer Vorlagengruppe

Erstellen Sie eine Vorlagengruppe mit dem Namen „Templates/Databases“.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.create",
    "params": {
        "name": "Templates/Databases"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107820"
        ]
    },
```

```
    "id": 1
}
```

Quelle

CTemplateGroup::create() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.delete

Beschreibung

`object templategroup.delete(array templateGroupIds)`

Diese Methode ermöglicht es Ihnen, Vorlagengruppen zu löschen.

Eine Vorlagengruppe kann nicht gelöscht werden, wenn sie Vorlagen enthält, die nur zu dieser Gruppe gehören.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der zu löschenden Vorlagengruppen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Vorlagengruppen in der Eigenschaft `groupids` enthält.

Beispiele

Mehrere Vorlagengruppen löschen

Löschen Sie zwei Vorlagengruppen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.delete",
    "params": [
        "107814",
        "107815"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107814",
            "107815"
        ]
    },
    "id": 1
}
```

Quelle

CTemplateGroup::delete() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.get

Beschreibung

`integer/array templategroup.get(object parameters)`

Die Methode ermöglicht es Ihnen, Vorlagengruppen anhand der angegebenen Parameter abzurufen.

Note:
Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
graphids	ID/array	Gibt nur Vorlagengruppen zurück, die Vorlagen mit den angegebenen Graphen enthalten.
groupids	ID/array	Gibt nur Vorlagengruppen mit den angegebenen Vorlagengruppen-IDs zurück.
templateids	ID/array	Gibt nur Vorlagengruppen zurück, die die angegebenen Vorlagen enthalten.
triggerids	ID/array	Gibt nur Vorlagengruppen zurück, die Vorlagen mit den angegebenen Auslösern enthalten.
with_graphs	boolean	Gibt nur Vorlagengruppen zurück, die Vorlagen mit Graphen enthalten.
with_graph_prototypes	boolean	Gibt nur Vorlagengruppen zurück, die Vorlagen mit Graphprototypen enthalten.
with_httptests	boolean	Gibt nur Vorlagengruppen zurück, die Vorlagen mit Webprüfungen enthalten.
with_items	boolean	Gibt nur Vorlagengruppen zurück, die Vorlagen mit Datenpunkten enthalten. Überschreibt die Parameter `with_simple_graph_items`.
with_item_prototypes	boolean	Gibt nur Vorlagengruppen zurück, die Vorlagen mit Datenpunktprototypen enthalten. Überschreibt den Parameter `with_simple_graph_item_prototypes`.
with_simple_graph_item_prototypes	boolean	Gibt nur Vorlagengruppen zurück, die Vorlagen mit Datenpunktprototypen enthalten, die für die Erstellung aktiviert sind und einen numerischen Informationstyp haben.
with_simple_graph_items	boolean	Gibt nur Vorlagengruppen zurück, die Vorlagen mit numerischen Datenpunkten enthalten.
with_templates	boolean	Gibt nur Vorlagengruppen zurück, die Vorlagen enthalten.
with_triggers	boolean	Gibt nur Vorlagengruppen zurück, die Vorlagen mit Auslösern enthalten.
selectTemplates	query	Gibt eine Eigenschaft `templates` mit den Vorlagen zurück, die zur Vorlagengruppe gehören. Unterstützt `count`.
limitSelects	integer	Begrenzt die Anzahl der Datensätze, die von Unterabfragen zurückgegeben werden. Gilt für die folgenden Unterabfragen: `selectTemplates` - Ergebnisse werden nach `template` sortiert.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `groupid`, `name`.
countOutput	boolean	Diese Parameter sind im Referenzkommentar beschrieben.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	

Parameter	Type	Beschreibung
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Gibt entweder Folgendes zurück:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Daten nach Namen abrufen

Rufen Sie alle Daten zu zwei Vorlagengruppen mit den Namen „Templates/Databases“ und „Templates/Modules“ ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.get",
    "params": {
        "output": "extend",
        "filter": {
            "name": [
                "Templates/Databases",
                "Templates/Modules"
            ]
        }
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "groupid": "13",
            "name": "Templates/Databases",
            "uuid": "748ad4d098d447d492bb935c907f652f"
        },
        {
            "groupid": "8",
            "name": "Templates/Modules",
            "uuid": "57b7ae836ca64446ba2c296389c009b7"
        }
    ],
    "id": 1
}
```

Siehe auch

- Template

Quelle

CTemplateGroup::get() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.massadd

Beschreibung

`object templategroup.massadd(object parameters)`

Diese Methode ermöglicht es Ihnen, mehrere zugehörige Objekte gleichzeitig zu allen angegebenen Vorlagengruppen hinzuzufügen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die IDs der zu aktualisierenden Vorlagengruppen und die Objekte enthalten, die zu allen Vorlagengruppen hinzugefügt werden sollen.

Die Methode akzeptiert die folgenden Parameter.

Parameter	Type	Beschreibung
groups	object/array	Zu aktualisierende Vorlagengruppen. Für die Vorlagengruppen darf nur die Eigenschaft `groupid` definiert sein. Parameter behavior: - *required*
templates	object/array	Vorlagen, die zu allen Vorlagengruppen hinzugefügt werden sollen. Für die Vorlagen darf nur die Eigenschaft `templateid` definiert sein. Parameter behavior: - *required*

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Vorlagengruppen in der Eigenschaft `groupids` enthält.

Beispiele

Hinzufügen von Vorlagen zu Vorlagengruppen

Fügen Sie zwei Vorlagen zu Vorlagengruppen mit den IDs 12 und 13 hinzu.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.massadd",
    "params": {
        "groups": [
            {
                "groupid": "12"
            },
            {
                "groupid": "13"
            }
        ],
        "templates": [
            {
                "templateid": "10486"
            },
            {
                "templateid": "10487"
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "12",
            "13"
        ]
    },
    "id": 1
}
```

Siehe auch

- Template

Quelle

CTemplateGroup::massAdd() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.massremove

Beschreibung

`object templategroup.massremove(object parameters)`

Diese Methode ermöglicht es Ihnen, zugehörige Objekte aus mehreren Vorlagen-Gruppen zu entfernen.

Note:
Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die IDs der zu aktualisierenden Vorlagengruppen und der zu entfernenden Objekte enthalten.

Parameter	Type	Beschreibung
groupids	ID/array	IDs der zu aktualisierenden Vorlagengruppen. Parameter behavior: - *required*
templateids	ID/array	IDs der Vorlagen, die aus allen Vorlagengruppen entfernt werden sollen. Parameter behavior: - *required*

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Vorlagengruppen in der Eigenschaft `groupids` enthält.

Beispiele

Entfernen von Vorlagen aus Vorlagengruppen

Entfernen Sie zwei Vorlagen aus den angegebenen Vorlagengruppen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.massremove",
    "params": {
        "groupids": [
            "5",
            "6"
        ],
```

```
        "templateids": [
            "30050",
            "30001"
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "5",
            "6"
        ]
    },
    "id": 1
}
```

Quelle

CTemplateGroup::massRemove() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.propagate

Beschreibung

`object templategroup.propagate(object parameters)`

Diese Methode ermöglicht es Ihnen, Berechtigungen auf alle Untergruppen von Vorlagen-Gruppen anzuwenden.

> **Note:**
> Diese Methode ist nur für Benutzertypen vom Typ *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Typ	Beschreibung
groups	object/array	Vorlagengruppen, die weitergegeben werden sollen. Für die Vorlagengruppen darf nur die Eigenschaft `groupid` definiert sein. Parameterverhalten: - *erforderlich*
permissions	boolean	Auf `true` setzen, wenn Berechtigungen weitergegeben werden sollen. Parameterverhalten: - *erforderlich*

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der propagierten Vorlagengruppen in der Eigenschaft `groupids` enthält.

Beispiele

Berechtigungen der Vorlagengruppe an ihre Untergruppen weitergeben.

Geben Sie die Berechtigungen der Vorlagengruppe an ihre Untergruppen weiter.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.propagate",
    "params": {
        "groups": [
            {
                "groupid": "15"
            }
        ],
        "permissions": true
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "15",
        ]
    },
    "id": 1
}
```

Siehe auch

- templategroup.update
- templategroup.massadd
- Template

Quelle

CTemplateGroup::propagate() in *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.update

Beschreibung

`object templategroup.update(object/array templateGroups)`

Diese Methode ermöglicht es Ihnen, vorhandene Vorlagengruppen zu aktualisieren.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` zu aktualisierende Eigenschaften der Vorlagengruppe.

Die Eigenschaft `groupid` muss für jede Vorlagengruppe definiert sein, alle anderen Eigenschaften sind optional. Nur die angegebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Vorlagengruppen in der Eigenschaft `groupids` enthält.

Beispiele

Umbenennen einer Vorlagengruppe

Benennen Sie eine Vorlagengruppe in „Templates/Databases“ um

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.update",
```

```
    "params": {
        "groupid": "7",
        "name": "Templates/Databases"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "7"
        ]
    },
    "id": 1
}
```

Quelle

CTemplateGroup::update() in *ui/include/classes/api/services/CTemplateGroup.php*.

Wartung

Diese Klasse ist für die Arbeit mit Wartungen vorgesehen.

Objektreferenzen:

- Wartung
- Zeitperiode
- Problem-Tag

Verfügbare Methoden:

- maintenance.create - neue Wartungen erstellen
- maintenance.delete - Wartungen löschen
- maintenance.get - Wartungen abrufen
- maintenance.update - Wartungen aktualisieren

Wartungs-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `maintenance` API.

Wartung

Das Wartungsobjekt hat die folgenden Eigenschaften.

Property	Type	Description
maintenanceid	ID	ID der Wartung. Property behavior: - *schreibgeschützt* - *erforderlich* für Aktualisierungsoperationen
name	string	Name der Wartung. Property behavior: - *erforderlich* für Erstellungsoperationen
active_since	timestamp	Zeitpunkt, ab dem die Wartung aktiv wird (einschließlich). Der angegebene Wert wird auf Minuten abgerundet. Property behavior: - *erforderlich* für Erstellungsoperationen

Property	Type	Description
active_till	timestamp	Zeitpunkt, bis zu dem die Wartung aktiv ist (ausschließlich). Der angegebene Wert wird auf Minuten abgerundet. Property behavior: - *erforderlich* für Erstellungsoperationen
description	string	Beschreibung der Wartung.
maintenance_type	integer	Typ der Wartung. Mögliche Werte: 0 - *(Standard)* mit Datenerfassung; 1 - ohne Datenerfassung.
tags_evaltype	integer	Problem-Tag Auswertungsmethode. Mögliche Werte: 0 - *(Standard)* Und/Oder; 2 - Oder.

Zeitperiode

Das Objekt für die Zeitperiode wird verwendet, um Zeiträume zu definieren, in denen die Wartung wirksam werden muss. Es hat die folgenden Eigenschaften.

Eigenschaft	Type	Beschreibung
period	integer	Dauer des Wartungszeitraums in Sekunden. Der angegebene Wert wird auf Minuten abgerundet. Mögliche Werte: 300 - 86399940. Standard: 3600.
timeperiod_type	integer	Typ der Zeitperiode. Mögliche Werte: 0 - *(Standard)* nur einmal; 2 - täglich; 3 - wöchentlich; 4 - monatlich.
start_date	timestamp	Datum, an dem der Wartungszeitraum wirksam werden muss. Der angegebene Wert wird auf Minuten abgerundet. Standard: aktuelles Datum. Property behavior: - *unterstützt*, wenn `timeperiod_type` auf ”nur einmal” gesetzt ist
start_time	integer	Uhrzeit des Tages, zu der die Wartung beginnt, in Sekunden. Der angegebene Wert wird auf Minuten abgerundet. Standard: 0. Property behavior: - *unterstützt*, wenn `timeperiod_type` auf ”täglich”, ”wöchentlich” oder ”monatlich” gesetzt ist

Eigenschaft	Type	Beschreibung
every	integer	Für tägliche und wöchentliche Zeiträume definiert `every` die Tages- oder Wochenintervalle, in denen die Wartung wirksam werden muss. Standardwert, wenn `timeperiod_type` auf ”täglich” oder ”wöchentlich” gesetzt ist: 1. Für monatliche Zeiträume, wenn `day` gesetzt ist, definiert die Eigenschaft `every` den Tag des Monats, an dem die Wartung wirksam werden muss. Standardwert, wenn `timeperiod_type` auf ”monatlich” gesetzt ist und `day` gesetzt ist: 1. Für monatliche Zeiträume, wenn `dayofweek` gesetzt ist, definiert die Eigenschaft `every` die Woche des Monats, in der die Wartung wirksam werden muss. Mögliche Werte, wenn `timeperiod_type` auf ”monatlich” gesetzt ist und `dayofweek` gesetzt ist: 1 - *(Standard)* erste Woche; 2 - zweite Woche; 3 - dritte Woche; 4 - vierte Woche; 5 - letzte Woche. Property behavior: - *unterstützt*, wenn `timeperiod_type` auf ”täglich”, ”wöchentlich” oder ”monatlich” gesetzt ist
dayofweek	integer	Wochentage, an denen die Wartung wirksam werden muss. Mögliche Bitmap-Werte: 1 - Montag; 2 - Dienstag; 4 - Mittwoch; 8 - Donnerstag; 16 - Freitag; 32 - Samstag; 64 - Sonntag. Dies ist ein Bitmaskenfeld; jede Summe der möglichen Bitmap-Werte ist zulässig (zum Beispiel 21 für Montag, Mittwoch und Freitag). Property behavior: - *erforderlich*, wenn `timeperiod_type` auf ”wöchentlich” gesetzt ist oder wenn `timeperiod_type` auf ”monatlich” gesetzt ist und `day` nicht gesetzt ist
day	integer	Tag des Monats, an dem die Wartung wirksam werden muss. Property behavior: - *erforderlich*, wenn `timeperiod_type` auf ”monatlich” gesetzt ist und `dayofweek` nicht gesetzt ist

Eigenschaft	Type	Beschreibung
month	integer	Monate, in denen die Wartung wirksam werden muss. Mögliche Bitmap-Werte: 1 - Januar; 2 - Februar; 4 - März; 8 - April; 16 - Mai; 32 - Juni; 64 - Juli; 128 - August; 256 - September; 512 - Oktober; 1024 - November; 2048 - Dezember. Dies ist ein Bitmaskenfeld; jede Summe der möglichen Bitmap-Werte ist zulässig (zum Beispiel 585 für Januar, April, Juli und Oktober). Property behavior: - *erforderlich*, wenn `timeperiod_type` auf ”monatlich” gesetzt ist

Ereignisname

Das Ereignisnamenobjekt wird verwendet, um festzulegen, welche Probleme unterdrückt werden müssen, wenn die Wartung in Kraft tritt.
Ereignisnamen können nur angegeben werden, wenn `maintenance_type` des Wartungsobjekts auf „with data collection“ gesetzt ist.
Es verfügt über die folgenden Eigenschaften.

Property	Type	Description
operator	integer	Bedingungs-operator. Mögliche Werte: 2 - *(Standard)* Enthält; 3 - Enthält nicht.
value	string	Wert des Ereignisnamens. Verhalten der Eigenschaft: - *erforderlich*

Problem-Tag

Das Problem-Tag-Objekt wird verwendet, um festzulegen, welche Probleme unterdrückt werden müssen, sobald das Wartungsfenster wirksam wird. Tags können nur angegeben werden, wenn `maintenance_type` des Maintenance-Objekts auf „with data collection“ gesetzt ist. Es verfügt über die folgenden Eigenschaften.

Property	Type	Description
tag	string	Name des Problem-Tags. Verhalten der Eigenschaft: - *erforderlich*
operator	integer	Bedingungs-Operator. Mögliche Werte: 0 - Gleich; 1 - Ungleich; 2 - *(Standard)* Enthält; 3 - Enthält nicht.

Property	Type	Description
value	string	Wert des Problem-Tags.

maintenance.create

Beschreibung

`object maintenance.create(object/array maintenances)`

Diese Methode ermöglicht es Ihnen, neue Wartungen zu erstellen.

Note:
Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Wartungen, die erstellt werden sollen.

Zusätzlich zu den Standard-Wartungseigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
groups	object/array	Hostgruppen, die in die Wartung aufgenommen werden sollen. Für die Hostgruppen darf nur die Eigenschaft `groupid` definiert sein. Parameterverhalten: - *erforderlich*, wenn `hosts` nicht gesetzt ist
hosts	object/array	Hosts, die in die Wartung aufgenommen werden sollen. Für die Hosts darf nur die Eigenschaft `hostid` definiert sein. Parameterverhalten: - *erforderlich*, wenn `groups` nicht gesetzt ist
triggers	object/array	Auslöser, die in die Wartung aufgenommen werden sollen. Für die Auslöser darf nur die Eigenschaft `triggerid` definiert sein. Parameterverhalten: - *erforderlich*, wenn `groups` und `hosts` nicht gesetzt sind - *unterstützt*, wenn maintenance_type=0 (mit Datensammlung)
event_names	object/array	Ereignisnamen, um Probleme zu definieren, die in die Wartung aufgenommen werden sollen. Parameterverhalten: - *unterstützt*, wenn maintenance_type=0 (mit Datensammlung)
tags	object/array	Problem-Tags, um Probleme zu definieren, die in die Wartung aufgenommen werden sollen. Parameterverhalten: - *unterstützt*, wenn `maintenance_type` des Wartungsobjekts auf „mit Datensammlung“ gesetzt ist
timeperiods	object/array	Wartungs-zeitintervalle. Parameterverhalten: - *erforderlich*

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Maintenances unter der Eigenschaft `maintenanceids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Maintenances.

Beispiele

Erstellen einer Wartung

Erstellen Sie eine Wartung mit Datensammlung für die Hostgruppe mit der ID „20“ sowie mit den Problem-Tags **service:mysqld** und **error**.
Sie muss vom 17.03.2026 bis zum 17.03.2027 aktiv sein, jeden Sonntag um 18:00 Uhr in Kraft treten und eine Stunde dauern.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.create",
    "params": {
        "name": "Sunday maintenance",
        "active_since": 1773720240,
        "active_till": 1805256240,
        "tags_evaltype": 0,
        "groups": [
            {"groupid": "20"}
        ],
        "timeperiods": [
            {
                "period": 3600,
                "timeperiod_type": 3,
                "start_time": 64800,
                "every": 1,
                "dayofweek": 64
            }
        ],
        "tags": [
            {
                "tag": "service",
                "operator": "0",
                "value": "mysqld"
            },
            {
                "tag": "error",
                "operator": "2",
                "value": ""
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "3"
        ]
    },
    "id": 1
}
```

Ad-hoc-Wartung erstellen

Erstellen Sie eine einstündige Ad-hoc-Wartung mit Datensammlung für die Hostgruppe mit der ID „4“, den Auslöser mit der ID „12345“ und den Ereignisnamen **Printer: ink low**.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.create",
    "params": {
```

```
        "name": "Ad-hoc maintenance",
        "active_since": 1792057740,
        "active_till": 1792061340,
        "tags_evaltype": 0,
        "groups": [
            {"groupid": "4"}
        ],
        "triggers": [
            {"triggerid": "12345"}
        ],
        "timeperiods": [
            {
                "period": 3600,
                "timeperiod_type": 0,
                "start_date": 1792057740
            }
        ],
        "event_names": [
            {
                "operator": "2",
                "value": "Printer: ink low"
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "4"
        ]
    },
    "id": 1
}
```

Siehe auch

- Host
- Host-Gruppe
- Auslöser
- Ereignisname
- Problem-Tag
- Zeitraum

Quelle

CMaintenance::create() in *ui/include/classes/api/services/CMaintenance.php*.

maintenance.delete

Beschreibung

`object maintenance.delete(array maintenanceIds)`

Diese Methode ermöglicht es Ihnen, Wartungszeiträume zu löschen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

(array) IDs der zu löschenden Wartungszeiträume.

Rückgabewerte

(object) Gibt ein Objekt zurück, das die IDs der gelöschten Wartungszeiträume unter der Eigenschaft maintenanceids enthält.

Beispiele

Mehrere Wartungszeiträume löschen

Löschen Sie zwei Wartungszeiträume.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.delete",
    "params": [
        "3",
        "1"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "3",
            "1"
        ]
    },
    "id": 1
}
```

Quelle

CMaintenance::delete() in *ui/include/classes/api/services/CMaintenance.php*.

maintenance.get

Beschreibung

`integer/array maintenance.get(object parameters)`

Die Methode ermöglicht es Ihnen, Wartungen anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

(object) Parameter zur Definition der gewünschten Ausgabe.

Die Methode unterstützt die folgenden Parameter.

Parameter	Typ	Beschreibung
groupids	ID/array	Gibt nur Wartungen zurück, die den angegebenen Hostgruppen zugewiesen sind.
hostids	ID/array	Gibt nur Wartungen zurück, die den angegebenen Hosts zugewiesen sind.
triggerids	ID/array	Gibt nur Wartungen zurück, die den angegebenen Auslösern zugewiesen sind.
maintenanceids	ID/array	Gibt nur Wartungen mit den angegebenen IDs zurück.
selectHostGroups	query	Gibt eine Eigenschaft hostgroups mit den der Wartung zugewiesenen Hostgruppen zurück.

Parameter	Typ	Beschreibung
selectHosts	query	Gibt eine Eigenschaft `hosts` mit den der Wartung zugewiesenen Hosts zurück.
selectTriggers	query	Gibt eine Eigenschaft `triggers` mit den der Wartung zugewiesenen Auslösern zurück.
selectEventNames	query	Gibt eine Eigenschaft `event_names` mit den der Wartung zugewiesenen Ereignisnamen zurück.
selectTags	query	Gibt eine Eigenschaft `tags` mit den Problem-Tags der Wartung zurück.
selectTimeperiods	query	Gibt eine Eigenschaft `timeperiods` mit den Zeiträumen der Wartung zurück.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `maintenanceid`, `name`, `maintenance_type`, `active_since`, `active_till`.
countOutput	boolean	Diese Parameter werden im Referenzkommentar beschrieben.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Gibt entweder Folgendes zurück:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Wartungen abrufen

Rufen Sie alle konfigurierten Wartungen, die Daten zum zugewiesenen Wartungsumfang (Hostgruppe, Host, Auslöser, Ereignisname, Tags) sowie die definierten Zeiträume ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.get",
    "params": {
        "output": "extend",
        "selectHostGroups": "extend",
        "selectHosts": "extend",
        "selectTriggers": "extend",
        "selectEventNames": "extend",
        "selectTags": "extend",
        "selectTimeperiods": "extend"
    },
    "id": 1
}
```

Antwort:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "maintenanceid": "1",
      "name": "Regelmäßige DB-Wartung",
```

```
      "maintenance_type": "0",
      "description": "",
      "active_since": "1791936000",
      "active_till": "2107555200",
      "tags_evaltype": "0",
      "hostgroups": [
        {
          "groupid": "20",
          "name": "Datenbanken",
          "flags": "0",
          "uuid": "748ad4d098d447d492bb935c907f652f"
        }
      ],
      "triggers": [],
      "event_names": [],
      "hosts": [],
      "tags": [],
      "timeperiods": [
        {
          "timeperiod_type": "4",
          "every": "1",
          "month": "4095",
          "dayofweek": "0",
          "day": "1",
          "start_time": "64800",
          "period": "3600",
          "start_date": "0"
        }
      ]
    },
    {
      "maintenanceid": "2",
      "name": "Ad-hoc: Drucker: Tinte LEER",
      "maintenance_type": "0",
      "description": "",
      "active_since": "1792057740",
      "active_till": "1792061340",
      "tags_evaltype": "0",
      "hostgroups": [
        {
          "groupid": "4",
          "name": "Zabbix-Server",
          "flags": "0",
          "uuid": "6f6799aa69e844b4b3918f779f2abf08"
        }
      ],
      "triggers": [],
      "event_names": [
        {
          "operator": "2",
          "value": "Drucker: Tinte LEER"
        }
      ],
      "hosts": [],
      "tags": [],
      "timeperiods": [
        {
          "timeperiod_type": "0",
          "every": "1",
          "month": "0",
          "dayofweek": "0",
          "day": "0",
```

```
            "start_time": "0",
            "period": "3600",
            "start_date": "1792057740"
          }
        ]
      }
  ],
  "id": 1
}
```


Siehe auch

- Host
- Host-Gruppe
- Auslöser
- Ereignisname
- Problem-Tag
- Zeitraum

Quelle

CMaintenance::get() in *ui/include/classes/api/services/CMaintenance.php*.

maintenance.update

Beschreibung

`object maintenance.update(object/array maintenances)`

Diese Methode ermöglicht es Ihnen, vorhandene Wartungen zu aktualisieren.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu aktualisierende Wartungseigenschaften.

Die Eigenschaft `maintenanceid` muss für jede Wartung definiert werden. Alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Zusätzlich zu den Standard-Wartungseigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
groups	object/array	Hostgruppen, die die aktuellen Gruppen ersetzen. Für die Hostgruppen darf nur die Eigenschaft `groupid` definiert sein. Parameterverhalten: - *erforderlich*, wenn `hosts` nicht festgelegt ist
hosts	object/array	Hosts, die die aktuellen Hosts ersetzen. Für die Hosts darf nur die Eigenschaft `hostid` definiert sein. Parameterverhalten: - *erforderlich*, wenn `groups` nicht festgelegt ist
triggers	object/array	Auslöser, die die aktuellen Auslöser ersetzen. Für die Auslöser darf nur die Eigenschaft `triggerid` definiert sein. Parameterverhalten: - *erforderlich*, wenn `groups` und `hosts` nicht festgelegt sind - *unterstützt*, wenn maintenance_type=0 (mit Datensammlung)

Parameter	Type	Beschreibung
event_names	object/array	Ereignisnamen, die die aktuellen Ereignisnamen ersetzen. Parameterverhalten: - *unterstützt*, wenn maintenance_type=0 (mit Datensammlung)
tags	object/array	Problem-Tags, die die aktuellen Tags ersetzen. Parameterverhalten: - *unterstützt*, wenn `maintenance_type` des Wartungsobjekts auf „mit Datensammlung“ gesetzt ist
timeperiods	object/array	Wartungs-Zeitintervalle, die die aktuellen Zeitintervalle ersetzen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Wartung unter der Eigenschaft `maintenanceids` enthält.

Beispiele

Zuweisen verschiedener Hosts

Ersetzen Sie die Hosts, die derzeit der Wartung zugewiesen sind, durch zwei andere.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.update",
    "params": {
        "maintenanceid": "3",
        "hosts": [
            {"hostid": "10085"},
            {"hostid": "10084"}
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "3"
        ]
    },
    "id": 1
}
```

Siehe auch

- Zeitperiode

Quelle

CMaintenance::update() in *ui/include/classes/api/services/CMaintenance.php*.

Web-Szenario

Diese Klasse ist für die Arbeit mit Web-Szenarien konzipiert.

Objektreferenzen:

- Web-Szenario
- Szenarioschritt
  - HTTP-Feld
- Web-Szenario-Tag

Verfügbare Methoden:

- httptest.create - neue Web-Szenarien erstellen
- httptest.delete - Web-Szenarien löschen
- httptest.get - Web-Szenarien abrufen
- httptest.update - Web-Szenarien aktualisieren

Web-Szenario-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `webcheck` API.

Webszenario

Das Webszenario-Objekt verfügt über die folgenden Eigenschaften.

Property	Type	Description
httptestid	ID	ID des Webszenarios. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
hostid	ID	ID des Hosts, zu dem das Webszenario gehört. Verhalten der Eigenschaft: - *konstant* - *erforderlich* für Erstellungsvorgänge
name	string	Name des Webszenarios. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
agent	string	User-Agent-Zeichenfolge, die vom Webszenario verwendet wird. Standard: Zabbix
authentication	integer	Authentifizierungsmethode, die vom Webszenario verwendet wird. Mögliche Werte: 0 - *(Standard)* keine; 1 - einfache HTTP-Authentifizierung; 2 - NTLM-Authentifizierung; 3 - Kerberos-Authentifizierung; 4 - Digest-Authentifizierung.
delay	string	Ausführungsintervall des Webszenarios. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300), einen Wert mit einem Zeitsuffix (z. B. `30s`, `1m`, `2h`, `1d`) oder ein Benutzermakro. Standard: 1m.
headers	array	HTTP-Header, die beim Ausführen einer Anfrage gesendet werden.
http_password	string	Passwort für die einfache HTTP-, NTLM-, Kerberos- oder Digest-Authentifizierung.
http_proxy	string	Proxy, der vom Webszenario verwendet wird, angegeben als *http://[username[:password]@]proxy.example.com[:port]*.
http_user	string	Benutzername für die einfache HTTP-, NTLM-, Kerberos- oder Digest-Authentifizierung.
retries	integer	Anzahl der Versuche, die ein Webszenario zur Ausführung jedes Schritts unternimmt, bevor es fehlschlägt. Standard: 1.
ssl_cert_file	string	Name der SSL-Zertifikatsdatei, die für die Clientauthentifizierung verwendet wird (muss im PEM-Format vorliegen).
ssl_key_file	string	Name der privaten SSL-Schlüsseldatei, die für die Clientauthentifizierung verwendet wird (muss im PEM-Format vorliegen).

Property	Type	Description
ssl_key_password	string	Passwort des privaten SSL-Schlüssels.
status	integer	Gibt an, ob das Webszenario aktiviert ist. Mögliche Werte: 0 - *(Standard)* aktiviert; 1 - deaktiviert.
templateid	ID	ID des übergeordneten Webszenarios der Vorlage. Verhalten der Eigenschaft: - *schreibgeschützt*
variables	array	Variablen des Webszenarios.
verify_host	integer	Gibt an, ob überprüft werden soll, dass der Hostname der Verbindung mit dem im Zertifikat des Hosts angegebenen Hostnamen übereinstimmt. Mögliche Werte: 0 - *(Standard)* Hostüberprüfung überspringen; 1 - Host überprüfen.
verify_peer	integer	Gibt an, ob überprüft werden soll, dass das Zertifikat des Hosts authentisch ist. Mögliche Werte: 0 - *(Standard)* Peer-Überprüfung überspringen; 1 - Peer überprüfen.
uuid	string	Global eindeutige Kennung, die verwendet wird, um importierte Webszenarien mit bereits vorhandenen Webszenarien zu verknüpfen. Wird automatisch generiert, wenn sie nicht angegeben wird. Verhalten der Eigenschaft: - *unterstützt*, wenn das Webszenario zu einer Vorlage gehört

Szenarioschritt

Das Objekt des Szenarioschritts definiert eine bestimmte Prüfung eines Webszenarios. Es verfügt über die folgenden Eigenschaften.

Property	Type	Beschreibung
name	string	Name des Szenarioschritts. Verhalten der Eigenschaft: - *erforderlich*
no	integer	Sequenznummer des Schritts in einem Webszenario. Verhalten der Eigenschaft: - *erforderlich*
url	string	Zu prüfende URL. Verhalten der Eigenschaft: - *erforderlich*
follow_redirects	integer	Gibt an, ob HTTP-Weiterleitungen verfolgt werden sollen. Mögliche Werte: 0 - Weiterleitungen nicht verfolgen; 1 - *(Standard)* Weiterleitungen verfolgen.
headers	array	HTTP-Header, die beim Ausführen einer Anfrage gesendet werden. Die Header des Szenarioschritts überschreiben die für das Webszenario angegebenen Header.
posts	string/array	HTTP-POST-Variablen als Zeichenfolge (unverarbeitete POST-Daten) oder als Array von HTTP-Feldern (Formulardaten).
required	string	Text, der in der Antwort enthalten sein muss.

Property	Type	Beschreibung
retrieve_mode	integer	Teil der HTTP-Antwort, den der Szenarioschritt abrufen muss. Mögliche Werte: 0 - *(Standard)* nur Body; 1 - nur Header; 2 - Header und Body.
status_codes	string	Bereiche der erforderlichen HTTP-Statuscodes, durch Kommas getrennt.
timeout	string	Zeitüberschreitung für die Anfrage in Sekunden. Akzeptiert eine Anzahl von Sekunden (z. B. 1, 60, 300), einen Wert mit einem Zeitsuffix (z. B. 30s, 1m, 1h) oder ein Benutzermakro. Standard: 15s. Maximum: 1h. Minimum: 1s.
variables	array	Variablen des Szenarioschritts.
query_fields	array	Abfragefelder - Array von HTTP-Feldern, die beim Ausführen einer Anfrage zur URL hinzugefügt werden.

HTTP-Feld

Das HTTP-Feldobjekt definiert den Namen und den Wert, die verwendet werden, um die Webszenario-Variablen, HTTP-Header sowie POST-Felder oder Abfragefelder anzugeben. Es hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
name	string	Name des Headers/der Variable/des POST- oder GET-Feldes. Eigenschaftsverhalten: - *erforderlich*
value	string	Wert des Headers/der Variable/des POST- oder GET-Feldes. Eigenschaftsverhalten: - *erforderlich*

Tag des Webszenarios

Das Tag-Objekt des Webszenarios hat die folgenden Eigenschaften.

Eigenschaft	Type	Beschreibung
tag	string	Name des Webszenario-Tags. Property behavior: - *required*
value	string	Wert des Webszenario-Tags. Property behavior: - *read-only*.
object	integer	Typ des Objekts, von dem das Tag geerbt wurde. Mögliche Werte: 0 - Vorlage; 1 - Host. Property behavior: - *read-only*.
objectid	ID	ID des Objekts, von dem das Tag geerbt wurde. Property behavior: - *read-only*.

httptest.create

Beschreibung

`object httptest.create(object/array webScenarios)`

Diese Methode ermöglicht es Ihnen, neue Webszenarien zu erstellen.

> **Note:**
> Beim Erstellen eines Webszenarios wird automatisch ein Satz von Web-Monitoring-Datenpunkten erstellt.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Die Berechtigung zum Aufrufen der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellende Webszenarien.

Zusätzlich zu den Standard-Webszenario- Eigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
steps	array	Szenarioschritte. Parameterverhalten: - *erforderlich*
tags	array	Webszenario-Tags.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Webszenarien unter der Eigenschaft `httptestids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Webszenarien.

Beispiele

Erstellen eines Webszenarios

Erstellen Sie ein Webszenario, um die Unternehmens-Homepage zu überwachen. Das Szenario hat zwei Schritte, um die Homepage und die Seite „Über uns“ zu prüfen und sicherzustellen, dass sie den HTTP-Statuscode 200 zurückgeben.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "httptestids": [
            "5"
        ]
    },
    "id": 1
}
```

Siehe auch

- Szenarioschritt

Quelle

CHttpTest::create() in *ui/include/classes/api/services/CHttpTest.php*.

httptest.delete

Beschreibung

`object httptest.delete(array webScenarioIds)`

Diese Methode ermöglicht es Ihnen, Web-Szenarien zu löschen.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der zu löschenden Webszenarien.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Webszenarien unter der Eigenschaft `httptestids` enthält.

Beispiele

Mehrere Webszenarien löschen

Löschen Sie zwei Webszenarien.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "httptest.delete",
    "params": [
        "2",
        "3"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "httptestids": [
            "2",
            "3"
        ]
    },
    "id": 1
}
```

Quelle

CHttpTest::delete() in *ui/include/classes/api/services/CHttpTest.php*.

httptest.get

Beschreibung

`integer/array httptest.get(object parameters)`

Die Methode ermöglicht es Ihnen, Webszenarien anhand der angegebenen Parameter abzurufen.

Note:
Diese Methode ist für Benutzer aller Typen verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
groupids	ID/array	Gibt nur Webszenarien zurück, die zu den angegebenen Host-Gruppen gehören.
hostids	ID/array	Gibt nur Webszenarien zurück, die zu den angegebenen Hosts gehören.
httptestids	ID/array	Gibt nur Webszenarien mit den angegebenen IDs zurück.
inherited	boolean	Wenn auf `true` gesetzt, werden nur Webszenarien zurückgegeben, die von einer Vorlage geerbt wurden.
inheritedTags	boolean	Gibt Webszenarien zurück, die die angegebenen `tags` auch in Vorlage/Host/verknüpften Vorlagen haben. Mögliche Werte: `true` - Vorlage/Host/verknüpfte Vorlagen müssen die angegebenen Tags ebenfalls haben; `false` - *(Standard)* Tags aus Vorlage/Host/verknüpften Vorlagen werden ignoriert.
monitored	boolean	Wenn auf `true` gesetzt, werden nur aktivierte Webszenarien zurückgegeben, die zu überwachten Hosts gehören.
selectInheritedTags	query	Gibt eine Eigenschaft `inheritedTags` mit Tags zurück, die sich auf Vorlage/Host/verknüpften Vorlagen befinden.
templated	boolean	Wenn auf `true` gesetzt, werden nur Webszenarien zurückgegeben, die zu Vorlagen gehören.
templateids	ID/array	Gibt nur Webszenarien zurück, die zu den angegebenen Vorlagen gehören.
expandName	flag	Erweitert Makros im Namen des Webszenarios.
expandStepName	flag	Erweitert Makros in den Namen der Szenarioschritte.
evaltype	integer	Tag-Auswertungsmethode. Mögliche Werte: 0 - *(Standard)* Und/Oder; 2 - Oder.
tags	array	Gibt nur Webszenarien mit den angegebenen Tags zurück. Format: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Ein leeres Array gibt alle Webszenarien zurück. Mögliche Werte für operator: 0 - *(Standard)* Enthält; 1 - Gleich; 2 - Enthält nicht; 3 - Ungleich; 4 - Existiert; 5 - Existiert nicht.

Parameter	Type	Beschreibung
selectHosts	query	Gibt die Hosts, zu denen das Webszenario gehört, als Array in der Eigenschaft `hosts` zurück.
selectSteps	query	Gibt Webszenarioschritte in der Eigenschaft `steps` zurück. Unterstützt `count`.
selectTags	query	Gibt Webszenario-Tags in der Eigenschaft `tags` zurück.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `httptestid`, `name`.
countOutput	boolean	Diese Parameter sind in der Referenzkommentierung beschrieben.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Gibt entweder Folgendes zurück:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Abrufen eines Webszenarios

Rufen Sie alle Daten zum Webszenario „9“ ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "httptest.get",
    "params": {
        "output": "extend",
        "selectSteps": "extend",
        "httptestids": "9"
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```


Siehe auch

- Host
- Szenarioschritt

Quelle

CHttpTest::get() in *ui/include/classes/api/services/CHttpTest.php*.

httptest.update

Beschreibung

`object httptest.update(object/array webScenarios)`

Diese Methode ermöglicht es Ihnen, vorhandene Webszenarien zu aktualisieren.

> **Note:**
> Diese Methode ist nur für die Benutzertypen *Admin* und *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu aktualisierende Eigenschaften des Webszenarios.

Die Eigenschaft `httptestid` muss für jedes Webszenario definiert sein, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Zusätzlich zu den Standard-Webszenario- Eigenschaften akzeptiert die Methode die folgenden Parameter.

Parameter	Type	Beschreibung
steps	array	Szenarioschritte zum Ersetzen vorhandener Schritte.
tags	array	Webszenario-Tags.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der aktualisierten Webszenarien unter der Eigenschaft `httptestid` enthält.

Beispiele

Aktivieren eines Webszenarios

Aktivieren Sie ein Webszenario, d. h. setzen Sie seinen Status auf „0“.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "httptest.update",
    "params": {
        "httptestid": "5",
        "status": 0
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "httptestids": [
            "5"
        ]
    },
    "id": 1
}
```

Siehe auch

- Szenarioschritt

Quelle

CHttpTest::update() in *ui/include/classes/api/services/CHttpTest.php*.

Wertezuordnung

Diese Klasse ist für die Arbeit mit Wertezuordnungen konzipiert.

Objektreferenzen:

- Wertezuordnung
  - Wertezuordnungen

Verfügbare Methoden:

- valuemap.create - neue Wertezuordnungen erstellen
- valuemap.delete - Wertezuordnungen löschen
- valuemap.get - Wertezuordnungen abrufen
- valuemap.update - Wertezuordnungen aktualisieren

Werte-Karten-Objekt

Die folgenden Objekte stehen in direktem Zusammenhang mit der `valuemap` API.

Wertezuordnung

Das Objekt der Wertezuordnung hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
valuemapid	ID	ID der Wertezuordnung. Verhalten der Eigenschaft: - *schreibgeschützt* - *erforderlich* für Aktualisierungsvorgänge
hostid	ID	ID des Hosts oder der Vorlage, zu dem bzw. zu der die Wertezuordnung gehört. Verhalten der Eigenschaft: - *konstant* - *erforderlich* für Erstellungsvorgänge
name	string	Name der Wertezuordnung. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
mappings	array	Wertezuordnungen für die aktuelle Wertezuordnung. Das Zuordnungsobjekt wird weiter unten im Detail beschrieben. Verhalten der Eigenschaft: - *erforderlich* für Erstellungsvorgänge
uuid	string	Universell eindeutige Kennung, die verwendet wird, um importierte Wertezuordnungen mit bereits vorhandenen zu verknüpfen. Wird automatisch generiert, wenn sie nicht angegeben wird. Verhalten der Eigenschaft: - *unterstützt*, wenn die Wertezuordnung zu einer Vorlage gehört

Wertezuordnungen

Das Objekt für Wertezuordnungen definiert Wertezuordnungen der Wertetabelle. Es hat die folgenden Eigenschaften.

Eigenschaft	Typ	Beschreibung
type	integer	Typ der Zuordnungsübereinstimmung. Mögliche Werte: 0 - *(Standard)* die Zuordnung wird angewendet, wenn der Wert gleich ist; 1 - die Zuordnung wird angewendet, wenn der Wert größer oder gleich ist[1]; 2 - die Zuordnung wird angewendet, wenn der Wert kleiner oder gleich ist[1]; 3 - die Zuordnung wird angewendet, wenn der Wert im Bereich liegt (Bereiche sind inklusive; mehrere Bereiche, durch Komma getrennt, können definiert werden)[1]; 4 - die Zuordnung wird angewendet, wenn der Wert einem regulären Ausdruck entspricht[2]; 5 - wenn keine Übereinstimmungen gefunden werden, wird die Zuordnung nicht angewendet und der Standardwert verwendet. Wenn `type` auf ”0”, ”1”, ”2”, ”3”, ”4” gesetzt ist, darf `value` nicht leer sein. Wenn `type` auf ”5” gesetzt ist, muss `value` leer sein.
value	string	Ursprünglicher Wert. Eigenschaftsverhalten: - *erforderlich*, wenn `type` auf ”1”, ”2”, ”3”, ”4” gesetzt ist - *unterstützt*, wenn `type` auf ”5” gesetzt ist
newvalue	string	Wert, dem der ursprüngliche Wert zugeordnet wird. Eigenschaftsverhalten: - *erforderlich*

[1] nur für Datenpunkte mit dem Werttyp ”numeric unsigned”, ”numeric float” unterstützt.
 [2] nur für Datenpunkte mit dem Werttyp ”character” unterstützt.

valuemap.create

Beschreibung

`object valuemap.create(object/array valuemaps)`

Diese Methode ermöglicht es Ihnen, neue Wertemaps zu erstellen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle entzogen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu erstellende Wertezuordnungen.

Die Methode akzeptiert Wertezuordnungen mit den standardmäßigen Eigenschaften von Wertezuordnungen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der erstellten Wertezuordnungen in der Eigenschaft `valuemapids` enthält. Die Reihenfolge der zurückgegebenen IDs entspricht der Reihenfolge der übergebenen Wertezuordnungen.

Beispiele

Erstellen einer Wertezuordnung

Erstellen Sie eine Wertezuordnung mit zwei Zuordnungen.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "1"
        ]
    },
    "id": 1
}
```

Quelle

CValueMap::create() in *ui/include/classes/api/services/CValueMap.php*.

valuemap.delete

Beschreibung

`object valuemap.delete(array valuemapids)`

Diese Methode ermöglicht es Ihnen, Wertemaps zu löschen.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Super admin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(array)` IDs der zu löschenden Wertezuordnungen.

Rückgabewerte

`(object)` Gibt ein Objekt zurück, das die IDs der gelöschten Wertezuordnungen in der Eigenschaft `valuemapids` enthält.

Beispiele

Mehrere Wertezuordnungen löschen

Löschen Sie zwei Wertezuordnungen.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.delete",
```

```
    "params": [
        "1",
        "2"
    ],
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

Quelle

CValueMap::delete() in *ui/include/classes/api/services/CValueMap.php*.

valuemap.get

Beschreibung

`integer/array valuemap.get(object parameters)`

Die Methode ermöglicht es Ihnen, Wertemaps anhand der angegebenen Parameter abzurufen.

> **Note:**
> Diese Methode ist für Benutzer aller Typen verfügbar. Die Berechtigung zum Aufruf der Methode kann in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object)` Parameter, die die gewünschte Ausgabe definieren.

Die Methode unterstützt die folgenden Parameter.

Parameter	Type	Beschreibung
valuemapids	ID/array	Gibt nur Wertezuordnungen mit den angegebenen IDs zurück.
selectMappings	query	Gibt die Wertezuordnungen für die aktuelle Wertezuordnung in der Eigenschaft `mappings` zurück. Unterstützt `count`.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Mögliche Werte: `valuemapid`, `name`.
countOutput	boolean	Diese Parameter werden im Referenzkommentar beschrieben.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Rückgabewerte

`(integer/array)` Gibt entweder Folgendes zurück:

- ein Array von Objekten;
- die Anzahl der abgerufenen Objekte, wenn der Parameter `countOutput` verwendet wurde.

Beispiele

Wertzuordnungen abrufen

Rufen Sie alle konfigurierten Wertzuordnungen ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```

Rufen Sie eine Wertzuordnung zusammen mit ihren Zuordnungen ab.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.get",
    "params": {
        "output": "extend",
        "selectMappings": "extend",
        "valuemapids": ["4"]
    },
    "id": 1
}
```

Antwort:

```
{
    "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
}
```


Quelle

CValueMap::get() in *ui/include/classes/api/services/CValueMap.php*.

valuemap.update

Beschreibung

`object valuemap.update(object/array valuemaps)`

Diese Methode ermöglicht es Ihnen, vorhandene Wertemaps zu aktualisieren.

> **Note:**
> Diese Methode ist nur für den Benutzertyp *Superadmin* verfügbar. Berechtigungen zum Aufrufen der Methode können in den Einstellungen der Benutzerrolle widerrufen werden. Weitere Informationen finden Sie unter Benutzerrollen.

Parameter

`(object/array)` Zu aktualisierende Eigenschaften der Wertezuordnung.

Die Eigenschaft `valuemapid` muss für jede Wertezuordnung definiert sein, alle anderen Eigenschaften sind optional. Nur die übergebenen Eigenschaften werden aktualisiert, alle anderen bleiben unverändert.

Rückgabewerte

(`object`) Gibt ein Objekt zurück, das die IDs der aktualisierten Wertezuordnungen in der Eigenschaft `valuemapids` enthält.

Beispiele

Namen der Wertezuordnung ändern

Ändern Sie den Namen der Wertezuordnung in „Device status“.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.update",
    "params": {
        "valuemapid": "2",
        "name": "Device status"
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "2"
        ]
    },
    "id": 1
}
```

Zuordnungen für eine Wertzuordnung ändern.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.update",
    "params": {
        "valuemapid": "2",
        "mappings": [
            {
                "type": "0",
                "value": "0",
                "newvalue": "Online"
            },
            {
                "type": "0",
                "value": "1",
                "newvalue": "Offline"
            }
        ]
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "2"
        ]
    },
    "id": 1
}
```

Quelle

CValueMap::update() in *ui/include/classes/api/services/CValueMap.php*.

Anhang 1. Referenzkommentar

Notation Methodensignatur

Jede Seite einer API-Methode enthält die Methodensignatur.

Beispiel: `object action.create(object/array actions)`

Signatur	Beschreibung
`object`	Der Datentyp der Daten, die die Methode zurückgibt.
`action.create`	Der vollständige Name der Methode.
`object/array`	Der Datentyp der Daten, die die Methode akzeptiert.
`actions`	Eine kurze Zusammenfassung der von der Methode akzeptierten Daten.

Datentypen

Die Zabbix-API unterstützt die folgenden Datentypen als Eingabe:

Typ	Beschreibung
ID	Eine eindeutige Kennung, mit der auf eine Entität verwiesen wird.
Boolean	Ein Boolescher Wert (entweder `true` oder `false`).
Flag	Ein Wert, der als `true` betrachtet wird, wenn er übergeben wird, und ungleich `null` ist; andernfalls wird der Wert als `false` betrachtet.
Integer	Eine ganze Zahl.
Float	Eine Gleitkommazahl.
String	Eine Textzeichenfolge.
Text	Eine längere Textzeichenfolge.
Timestamp	Ein Unix-Zeitstempel.
Array	Eine geordnete Wertefolge (ein einfaches Array).
Object	Ein assoziatives Array.
Query	Ein Wert, der die zurückzugebenden Daten definiert. Der Wert kann als Array von Eigenschaftsnamen (um nur bestimmte Eigenschaften zurückzugeben) oder als einer der vordefinierten Werte definiert werden: `extend` – gibt alle Objekteigenschaften zurück; `count` – gibt die Anzahl der abgerufenen Datensätze zurück, wird nur von bestimmten Unterauswahlen unterstützt.

Attention:
Die Zabbix-API gibt Werte immer nur als Zeichenfolgen oder Arrays zurück.

Verhalten von Eigenschaften

Einige der Objekteigenschaften sind mit kurzen Bezeichnungen versehen, um ihr Verhalten zu beschreiben. Die folgenden Bezeichnungen werden verwendet:

- ***schreibgeschützt*** - der Wert der Eigenschaft wird automatisch gesetzt und kann vom Benutzer nicht definiert oder geändert werden, auch nicht unter bestimmten Bedingungen (z. B. *schreibgeschützt* für geerbte oder entdeckte Objekte);
- ***nur schreibbar*** - der Wert der Eigenschaft kann gesetzt, danach jedoch nicht mehr abgerufen werden;
- ***konstant*** - der Wert der Eigenschaft kann beim Erstellen eines Objekts gesetzt, danach jedoch nicht mehr geändert werden;
- ***unterstützt*** - der Wert der Eigenschaft muss nicht gesetzt werden, darf jedoch unter bestimmten Bedingungen gesetzt werden (z. B. *unterstützt*, wenn `type` auf ”Simple check”, ”External check”, ”SSH agent”, ”TELNET agent” oder ”HTTP agent” gesetzt ist); beachten Sie jedoch, dass *unterstützte* Eigenschaften unabhängig von den Bedingungen weiterhin auf ihre Standardwerte gesetzt werden können;
- ***erforderlich*** - der Wert der Eigenschaft muss für alle Operationen (außer get-Operationen) oder unter bestimmten Bedingungen gesetzt werden (z. B. *erforderlich* für create-Operationen; *erforderlich*, wenn `operationtype` auf ”global script” gesetzt ist und `opcommand_hst` nicht gesetzt ist).

Note:
Bei update-Operationen gilt eine Eigenschaft als „gesetzt“, wenn sie während der update-Operation gesetzt wird.

Eigenschaften, die nicht mit Bezeichnungen versehen sind, sind optional.

Verhalten von Parametern

Einige der Operationsparameter sind mit kurzen Bezeichnungen versehen, um ihr Verhalten für die Operation zu beschreiben. Die folgenden Bezeichnungen werden verwendet:

- ***schreibgeschützt*** - der Wert des Parameters wird automatisch gesetzt und kann vom Benutzer nicht definiert oder geändert werden, auch nicht unter bestimmten Bedingungen (z. B. *schreibgeschützt* für geerbte Objekte oder entdeckte Objekte);
- ***nur schreiben*** - der Wert des Parameters kann gesetzt, danach jedoch nicht mehr abgerufen werden;
- ***unterstützt*** - der Wert des Parameters muss nicht gesetzt werden, darf jedoch unter bestimmten Bedingungen gesetzt werden (z. B. *unterstützt*, wenn `operating_mode` des Proxy-Objekts auf ”passive proxy” gesetzt ist); beachten Sie jedoch, dass *unterstützte* Parameter unabhängig von den Bedingungen weiterhin auf ihre Standardwerte gesetzt werden können;
- ***erforderlich*** - der Wert des Parameters muss gesetzt werden.

Parameter, die nicht mit Bezeichnungen versehen sind, sind optional.

Reservierter ID-Wert ”0” Der reservierte ID-Wert ”0” kann zum Filtern von Elementen und zum Entfernen referenzierter Objekte verwendet werden. Um zum Beispiel einen referenzierten Proxy von einem Host zu entfernen, sollte die proxyid auf 0 gesetzt werden (”proxyid”: ”0”) oder zum Filtern von Hosts, die von der Server-Option proxyids überwacht werden, sollte der Wert auf 0 gesetzt werden (”proxyids”: ”0”).“

Allgemeine Parameter der Methode ”get” Die folgenden Parameter werden von allen `get`-Methoden unterstützt:

Parameter	Type	Beschreibung
countOutput	boolean	Gibt die Anzahl der Datensätze im Ergebnis anstelle der eigentlichen Daten zurück.
editable	boolean	Wenn auf `true` gesetzt, werden nur Objekte zurückgegeben, für die der Benutzer Schreibrechte hat. Standard: `false`.
excludeSearch	boolean	Gibt Ergebnisse zurück, die nicht den im Parameter `search` angegebenen Kriterien entsprechen.
filter	object	Gibt nur die Ergebnisse zurück, die exakt dem angegebenen Filter entsprechen. Akzeptiert ein Objekt, bei dem die Schlüssel Eigenschaftsnamen sind (z. B. Host-Objekteigenschaften in `host.get`, Datenpunkt-Objekteigenschaften in `item.get` usw.) und die Werte entweder ein einzelner Wert oder ein Array von Werten sind, mit denen verglichen werden soll. Unterstützt keine Eigenschaften des Datentyps `text`. Beachten Sie, dass einige Methoden für diesen Parameter eine spezielle Funktionalität haben, die auf der Methodenseite beschrieben ist (z. B. unterstützt der Parameter `filter` in host.get auch Eigenschaften von Host-Schnittstellen).
limit	integer	Begrenzt die Anzahl der zurückgegebenen Datensätze.
output	query	Objekteigenschaften, die zurückgegeben werden sollen. Beachten Sie, dass die Objekt-ID (d. h. `hostid`, `itemid` usw.) immer in der Antwort enthalten ist, auch wenn sie im Parameter `output` nicht angegeben ist. Standard: `extend`.
preservekeys	boolean	Verwendet IDs als Schlüssel im resultierenden Array.

Parameter	Type	Beschreibung
search	object	Gibt Ergebnisse zurück, die dem angegebenen Muster entsprechen (Groß-/Kleinschreibung wird nicht beachtet). Akzeptiert ein Objekt, bei dem die Schlüssel Eigenschaftsnamen sind (z. B. Host-Objekteigenschaften in `host.get`, Datenpunkt-Objekteigenschaften in `item.get` usw.) und die Werte Zeichenfolgen sind, nach denen gesucht werden soll. Wenn keine zusätzlichen Optionen angegeben sind, wird eine Suche vom Typ `LIKE "%…%"` durchgeführt. Unterstützt nur Eigenschaften des Datentyps `string` und `text`. Beachten Sie, dass einige Methoden für diesen Parameter eine spezielle Funktionalität haben, die auf der Methodenseite beschrieben ist (z. B. unterstützt der Parameter `search` in host.get auch Eigenschaften von Host-Schnittstellen).
searchByAny	boolean	Wenn auf `true` gesetzt, werden Ergebnisse zurückgegeben, die einem beliebigen der im Parameter `filter` oder `search` angegebenen Kriterien entsprechen, anstatt allen Kriterien. Standard: `false`.
searchWildcardsEnabled	boolean	Wenn auf `true` gesetzt, wird die Verwendung von "*" als Platzhalterzeichen im Parameter `search` aktiviert. Standard: `false`.
sortfield	string/array	Sortiert das Ergebnis nach den angegebenen Eigenschaften. Eine Liste der Eigenschaften, die zum Sortieren verwendet werden können, finden Sie in der Beschreibung der jeweiligen API-get-Methode. Makros werden vor dem Sortieren nicht expandiert. Wenn kein Wert angegeben ist, werden die Daten unsortiert zurückgegeben.
sortorder	string/array	Reihenfolge der Sortierung. Wenn ein Array übergeben wird, wird jeder Wert der entsprechenden Eigenschaft zugeordnet, die im Parameter `sortfield` angegeben ist. Mögliche Werte: `ASC` - *(Standard)* aufsteigend; `DESC` - absteigend.
startSearch	boolean	Der Parameter `search` vergleicht den Anfang von Feldern, führt also stattdessen eine Suche vom Typ `LIKE "…%"` aus. Wird ignoriert, wenn `searchWildcardsEnabled` auf `true` gesetzt ist.

Flags für den Entitätsursprung Get-Methoden geben eine Eigenschaft `flags` für Entitäten zurück, die mit Low-Level-Discovery zusammenhängen (LLD-Regel/LLD-Regelprototyp, Datenpunkt/Datenpunktprototyp usw.). Diese Eigenschaft ist nützlich, um anzugeben, ob die Entität entdeckt wurde oder nicht, da die Bearbeitung für entdeckte Entitäten eingeschränkt ist.

Die Eigenschaft `flags` gibt ein Ergebnis zurück, das auf einer Kombination (Operation „+“) dieser Werte basiert:

Value	Description
0	Basisentität (Datenpunkt, Auslöser, Graph, Host)
1	Low-Level-Discovery-Regel
2	Beliebiger Prototyp (Datenpunktprototyp, Auslöserprototyp, LLD-Regelprototyp usw.)
4	Entdeckte Entität (entdeckter Datenpunkt, Auslöser, Graph, Host, LLD-Regel)

Der von der Eigenschaft `flags` zurückgegebene **kombinierte** Wert kann sein:

Value	Combination of	Description
0	0	Normale Entität (Datenpunkt, Auslöser, Graph, Host).
2	2	Entitätsprototyp (Datenpunktprototyp, Auslöserprototyp usw.).
6	2+4	Entdeckter Datenpunkt, Auslöser, Graph, Host (aus Prototyp konvertiert).
1	1	Low-Level-Discovery-Regel.
3	1+2	Low-Level-Discovery-Regelprototyp.
5	1+4	Entdeckte Low-Level-Discovery-Regel (aus Prototyp konvertiert).

Value	Combination of	Description
7	1+2+4	Entdeckter Low-Level-Discovery-Regelprototyp.

Beispiele Benutzerberechtigungsprüfung

Hat der Benutzer die Berechtigung, auf Hosts zu schreiben, deren Namen mit ”MySQL” oder ”Linux” beginnen? Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "countOutput": true,
        "search": {
            "host": ["MySQL", "Linux"]
        },
        "editable": true,
        "startSearch": true,
        "searchByAny": true
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": "0",
    "id": 1
}
```

Note:
Keine Ergebnisse bedeuten, es gibt keine Host mit Lese-/Schreib- Berechtigungen.

Mismatch-Zählung

Zählt die Anzahl der Hosts, deren Namen die Teilzeichenkette ”ubuntu” nicht enthalten.

Anfrage:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "countOutput": true,
        "search": {
            "host": "ubuntu"
        },
        "excludeSearch": true
    },
    "id": 1
}
```

Antwort:

```
{
    "jsonrpc": "2.0",
    "result": "44",
    "id": 1
}
```

Suche nach Hosts mithilfe von Platzhaltern

Suche nach Hosts, deren Name das Wort ”Server” enthält und die Schnittstellenports ”10050” or ”10071” haben. Sortieren Sie das Ergebnis nach dem Hostnamen in absteigender Reihenfolge und beschränke die Ausgabe auf 5 Hosts.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "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
}
```

Suche nach Hosts mit Wildcards und ”preservekeys”

Wenn Sie der vorherigen Anfrage den Parameter ”preservekeys” hinzufügen, wird das Ergebnis als assoziatives Array zurückgegeben, wobei die Schlüssel die IDs der Objekte sind.

Anfrage:

```
{
    "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
}
```

Antwort:

```
{
    "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
}
```


Anhang 2. Änderungen von 7.4 zu 8.0

Inkompatible Änderungen

Andere Änderungen und Fehlerbehebungen

Anhang 3. Änderungen in 8.0

8.0.1 Diese Version ist noch nicht veröffentlicht.

Erweiterungen

Übersicht Obwohl Zabbix eine Vielzahl von Funktionen bietet, gibt es immer Raum für zusätzliche Funktionalität. Erweiterungen sind eine praktische Möglichkeit, die Monitoring-Fähigkeiten von Zabbix zu verändern und zu erweitern, ohne den Quellcode zu ändern.

Sie können die Funktionalität von Zabbix entweder mithilfe integrierter Erweiterungsoptionen (Trapper-Datenpunkte, Benutzerparameter usw.) oder durch die Verwendung oder Erstellung benutzerdefinierter Erweiterungen (ladbare Module, Plugins usw.) erweitern.

Dieser Abschnitt bietet eine Übersicht mit Verweisen auf alle Optionen zur Erweiterung von Zabbix.

Datenerfassung mit benutzerdefinierten Befehlen Trapper-Datenpunkte

Trapper-Datenpunkte sind Datenpunkte, die eingehende Daten annehmen, anstatt sie abzufragen. Trapper-Datenpunkte sind nützlich, um bestimmte Daten an den Zabbix Server oder Proxy zu senden, zum Beispiel periodische Verfügbarkeits- und Leistungsdaten bei lang laufenden Benutzerskripten. Das Senden von Daten an den Zabbix Server oder Proxy ist mit dem Dienstprogramm Zabbix sender oder dem Zabbix sender-Protokoll möglich. Das Senden von Daten an den Zabbix Server ist außerdem mit der API-Methode `history.push` möglich.

Externe Prüfungen

Eine externe Prüfung ist ein Datenpunkt zum Ausführen von Prüfungen durch das Starten einer ausführbaren Datei, zum Beispiel eines Shell-Skripts oder einer Binärdatei.

Externe Prüfungen werden vom Zabbix Server oder Proxy ausgeführt (wenn der Host vom Proxy überwacht wird) und erfordern keinen auf dem überwachten Host laufenden Agent.

Benutzerparameter

Ein Benutzerparameter ist ein benutzerdefinierter Befehl (der einem benutzerdefinierten Schlüssel zugeordnet ist), der bei der Ausführung die benötigten Daten vom Host abrufen kann, auf dem der Zabbix Agent läuft. Benutzerparameter sind nützlich, um Agent- oder Agent-2-Datenpunkte zu konfigurieren, die in Zabbix nicht vordefiniert sind.

`system.run[]`-Datenpunkte des Zabbix Agent

`system.run[]`-Datenpunkte des Zabbix Agent dienen zum Ausführen eines benutzerdefinierten Befehls (verknüpft mit einem vordefinierten Schlüssel `system.run[]`, zum Beispiel `system.run[myscript.sh]`) auf dem Host, auf dem der Zabbix Agent ausgeführt wird.

> **Note:**
> `system.run[]`-Datenpunkte sind standardmäßig deaktiviert und müssen, falls sie verwendet werden, aktiviert (zugelassen und in der Konfigurationsdatei des Zabbix Agent oder Zabbix Agent 2 definiert werden (`AllowKey` oder `AllowKeyRegexp` Konfigurationsparameter).

> **Attention:**
> Benutzerdefinierte Befehle in Datenpunkten wie externen Prüfungen, Benutzerparametern und `system.run[]`-Zabbix-Agent-Datenpunkten werden unter dem OS-Benutzer ausgeführt, der zum Ausführen von Zabbix-Komponenten verwendet wird. Um diese Befehle auszuführen, muss dieser Benutzer über die erforderlichen Berechtigungen verfügen.

HTTP-Agent-Datenpunkte

Ein HTTP-Agent-Datenpunkt dient zum Ausführen von Datenanforderungen über HTTP/HTTPS. HTTP-Agent-Datenpunkte sind nützlich, um Anforderungen an HTTP-Endpunkte zu senden und Daten aus Diensten wie *Elasticsearch* und *OpenWeatherMap* abzurufen, den Status der Zabbix-API oder den Status eines Apache- oder NGINX-Webservers zu überprüfen usw. HTTP-Agent-Datenpunkte (mit aktiviertem Trapping) können auch als Trapper-Datenpunkte fungieren.

Skript-Datenpunkte

Ein Skript-Datenpunkt ist ein Datenpunkt zum Ausführen von benutzerdefiniertem JavaScript-Code, der Daten über HTTP/HTTPS abruft. Skript-Datenpunkte sind nützlich, wenn die von HTTP-Agent-Datenpunkten bereitgestellte Funktionalität nicht ausreicht. Zum Beispiel kann in anspruchsvollen Datenerfassungsszenarien, die mehrere Schritte oder eine komplexe Logik erfordern, ein Skript-Datenpunkt so konfiguriert werden, dass er einen HTTP-Aufruf ausführt, dann die empfangenen Daten verarbeitet und anschließend den transformierten Wert an einen zweiten HTTP-Aufruf übergibt.

> **Note:**
> HTTP-Agent-Datenpunkte und Skript-Datenpunkte werden von Zabbix Server und Proxy unterstützt und erfordern keinen laufenden Agent auf dem überwachten Host.

Erweiterte Erweiterungen Ladbare Module

Ladbare Module, in C geschrieben, sind eine vielseitige und leistungsorientierte Option zur Erweiterung der Funktionalität von Zabbix-Komponenten (Server, Proxy, Agent) auf UNIX-Plattformen. Ein ladbares Modul ist im Grunde eine Shared Library, die vom Zabbix-Daemon verwendet und beim Start geladen wird. Die Bibliothek sollte bestimmte Funktionen enthalten, damit ein Zabbix-Prozess erkennen kann, dass die Datei tatsächlich ein Modul ist, das er laden und mit dem er arbeiten kann.

Ladbare Module bieten eine Reihe von Vorteilen, darunter die Möglichkeit, neue Metriken hinzuzufügen oder andere Logik zu implementieren (zum Beispiel den Zabbix-Export von Verlaufsdaten), hohe Leistung sowie die Möglichkeit, die bereitgestellte Funktionalität zu entwickeln, zu nutzen und zu teilen. Dies trägt zu einer reibungslosen Wartung bei und hilft dabei, neue Funktionalität einfacher und unabhängig von der Zabbix-Kerncodebasis bereitzustellen.

Ladbare Module sind besonders in einer komplexen Monitoring-Umgebung nützlich. Bei der Überwachung eingebetteter Systeme, bei einer großen Anzahl überwachter Parameter oder bei umfangreichen Skripten mit komplexer Logik oder langer Startzeit wirken sich Erweiterungen wie Benutzerparameter, `system.run[]`-Zabbix-Agent-Datenpunkte und externe Prüfungen auf die Leistung aus. Ladbare Module bieten eine Möglichkeit, die Zabbix-Funktionalität zu erweitern, ohne Einbußen bei der Leistung hinnehmen zu müssen.

Plugins

Plugins erweitern die Monitoring-Funktionen von Zabbix Agent 2. Sie sind in der Programmiersprache Go geschrieben und bieten eine Alternative zu ladbaren Modulen (geschrieben in C).

Es ist möglich, ein eigenes Plugin (eigenständigen Datensammler) für Zabbix Agent 2 zu entwickeln. Siehe Developer center.

Anpassung von Warnmeldungen Webhooks

Ein webhook ist ein Zabbix-Medientyp, der eine Möglichkeit bietet, die Zabbix-Benachrichtigungsfunktionen auf externe Software wie Helpdesk-Systeme, Chats oder Messenger zu erweitern. Ähnlich wie Skript-Datenpunkte sind webhooks nützlich, um HTTP-Aufrufe mit benutzerdefiniertem JavaScript-Code auszuführen, zum Beispiel um Benachrichtigungen an verschiedene Plattformen wie Microsoft Teams, Discord und Jira zu senden. Es ist auch möglich, einige Daten zurückzugeben (zum Beispiel zu erstellten Helpdesk-Tickets), die dann in Zabbix angezeigt werden.

Vorhandene webhooks sind im Zabbix-Git-Repository verfügbar. Informationen zur Entwicklung benutzerdefinierter webhooks finden Sie unter Webhook-Entwicklungsrichtlinien.

Warnskripte

Ein Warnskript ist ein Zabbix-Medientyp, der die Möglichkeit bietet, eine alternative Methode (Skript) zur Verarbeitung von Zabbix-Warnungen zu erstellen. Warnskripte sind nützlich, wenn Sie mit den vorhandenen Medientypen zum Senden von Warnungen in Zabbix nicht zufrieden sind.

Anpassung des Frontends Benutzerdefinierte Themes

Es ist möglich, das visuelle Erscheinungsbild des Zabbix Frontend durch benutzerdefinierte Themes zu ändern. Siehe die Anleitung zum Erstellen und Anwenden Ihrer eigenen Themes.

Frontend-Module

Frontend-Module bieten eine Möglichkeit, die Funktionalität des Zabbix-Frontend zu erweitern, indem Module von Drittanbietern hinzugefügt oder eigene Module entwickelt werden. Mit Frontend-Modulen können Sie neue Menüpunkte sowie die entsprechenden Ansichten, Aktionen usw. hinzufügen.

Globale Skripte Ein globales Skript ist eine benutzerdefinierte Befehlsmenge, die auf einem Überwachungsziel ausgeführt werden kann (durch den Shell-Interpreter (/bin/sh)), abhängig vom konfigurierten Geltungsbereich und den Benutzerberechtigungen. Globale Skripte können für die folgenden Aktionen konfiguriert werden:

- Aktions-Operation
- Manuelle Host-Aktion
- Manuelle Ereignisaktion

Globale Skripte sind in vielen Fällen nützlich. Wenn sie beispielsweise für Aktionsoperationen oder manuelle Host-Aktionen konfiguriert sind, können Sie globale Skripte verwenden, um Remote-Befehle wie den Neustart einer Anwendung (Webserver, Middleware, CRM usw.) oder das Freigeben von Speicherplatz (Entfernen älterer Dateien, Bereinigen von `/tmp` usw.) automatisch oder manuell auszuführen. Oder, ein weiteres Beispiel: Wenn sie für manuelle Ereignisaktionen konfiguriert sind, können Sie globale Skripte verwenden, um Problem-Tickets in externen Systemen zu verwalten.

Globale Skripte können durch den Zabbix Server, Proxy oder Agent ausgeführt werden.

> **Attention:**
> Benutzerdefinierte Befehle werden mit dem Betriebssystembenutzer ausgeführt, der zum Ausführen der Zabbix-Komponenten verwendet wird. Um diese Befehle auszuführen, muss dieser Benutzer über die erforderlichen Berechtigungen verfügen.

Zabbix API Die Zabbix API ist eine HTTP-basierte API, die Teil des Zabbix Frontend ist. Mit der Zabbix API können Sie die folgenden Vorgänge ausführen:

- Die Konfiguration von Zabbix programmgesteuert abrufen und ändern.
- Die Zabbix-Konfiguration importieren und exportieren.
- Auf Verlaufs- und Trenddaten von Zabbix zugreifen.
- Anwendungen für die Zusammenarbeit mit Zabbix konfigurieren.

- Zabbix mit Software von Drittanbietern integrieren.
- Routineaufgaben automatisieren.

Die Zabbix API besteht aus einer Vielzahl von Methoden, die nominell in separate APIs gruppiert sind. Jede Methode führt eine bestimmte Aufgabe aus. Informationen zu den verfügbaren Methoden sowie einen Überblick über die von der Zabbix API bereitgestellten Funktionen finden Sie in der Zabbix API-Methodenreferenz.

Ladbare Module

Übersicht

Ladbare Module bieten eine leistungsorientierte Möglichkeit, die Funktionalität von Zabbix zu erweitern.

Sie können die Funktionalität von Zabbix auf viele Arten erweitern, zum Beispiel mit Benutzerparametern, externen Prüfungen und Zabbix-Agent-Datenpunkten vom Typ system.run. Diese funktionieren sehr gut, haben jedoch einen wesentlichen Nachteil, nämlich fork(). Zabbix muss jedes Mal einen neuen Prozess forken, wenn es eine Benutzermetrik verarbeitet, was sich negativ auf die Leistung auswirkt. Normalerweise ist das kein großes Problem, kann jedoch zu einem ernsthaften Thema werden, wenn eingebettete Systeme überwacht werden, eine große Anzahl überwachter Parameter vorhanden ist oder umfangreiche Skripte mit komplexer Logik oder langer Startzeit verwendet werden.

Die Unterstützung ladbarer Module bietet Möglichkeiten, den Zabbix-Agent, den Server und den Proxy zu erweitern, ohne dabei Leistung einzubüßen.

Ein ladbares Modul ist im Grunde eine Shared Library, die vom Zabbix-Daemon verwendet und beim Start geladen wird. Die Bibliothek sollte bestimmte Funktionen enthalten, damit ein Zabbix-Prozess erkennen kann, dass die Datei tatsächlich ein Modul ist, das er laden und verwenden kann.

Ladbare Module bieten eine Reihe von Vorteilen. Hohe Leistung und die Möglichkeit, beliebige Logik zu implementieren, sind sehr wichtig, aber der vielleicht wichtigste Vorteil ist die Möglichkeit, Zabbix-Module zu entwickeln, zu verwenden und zu teilen. Dies trägt zu einer problemlosen Wartung bei und hilft dabei, neue Funktionalität einfacher und unabhängig von der Zabbix-Kerncodebasis bereitzustellen.

Die Lizenzierung von Modulen und ihre Distribution in Binärform unterliegen der AGPL-3.0-Lizenz (Module werden zur Laufzeit mit Zabbix gelinkt und verwenden Zabbix-Header; der gesamte Zabbix-Code steht seit Zabbix 7.0 unter der AGPL-3.0-Lizenz). Die Binärkompatibilität wird von Zabbix nicht garantiert.

Die Stabilität der Modul-API wird während eines Zabbix-LTS-Zyklus (Long Term Support) Release garantiert. Die Stabilität der Zabbix-API wird nicht garantiert (technisch ist es möglich, interne Zabbix-Funktionen aus einem Modul heraus aufzurufen, es gibt jedoch keine Garantie, dass solche Module funktionieren).

Modul-API

Damit eine Shared Library als Zabbix-Modul behandelt wird, sollte sie mehrere Funktionen implementieren und exportieren. Derzeit gibt es sechs Funktionen in der Zabbix-Modul-API, von denen nur eine verpflichtend ist und die anderen fünf optional sind.

Obligatorische Schnittstelle

Die einzige obligatorische Funktion ist `zbx_module_api_version()`:

```
int zbx_module_api_version(void);
```

Diese Funktion sollte die von diesem Modul implementierte API-Version zurückgeben, und damit das Modul geladen werden kann, muss diese Version mit der von Zabbix unterstützten Modul-API-Version übereinstimmen. Die von Zabbix unterstützte Version der Modul-API ist `ZBX_MODULE_API_VERSION`. Diese Funktion sollte daher diese Konstante zurückgeben. Die alte Konstante `ZBX_MODULE_API_VERSION_ONE`, die zu diesem Zweck verwendet wurde, ist jetzt so definiert, dass sie `ZBX_MODULE_API_VERSION` entspricht, um die Quellcodekompatibilität zu erhalten, ihre Verwendung wird jedoch nicht empfohlen.

Optionale Schnittstelle

Die optionalen Funktionen sind `zbx_module_init()`, `zbx_module_item_list()`, `zbx_module_item_timeout()`, `zbx_module_history_write_cbs()` und `zbx_module_uninit()`:

```
int zbx_module_init(void);
```

Diese Funktion sollte die erforderliche Initialisierung für das Modul durchführen, falls eine solche erforderlich ist. Bei Erfolg sollte sie `ZBX_MODULE_OK` zurückgeben. Andernfalls sollte sie `ZBX_MODULE_FAIL` zurückgeben. Im letzteren Fall wird Zabbix nicht gestartet.

```
ZBX_METRIC  *zbx_module_item_list(void);
```

Diese Funktion sollte eine Liste der vom Modul unterstützten Datenpunkte zurückgeben. Jeder Datenpunkt ist in einer ZBX_METRIC-Struktur definiert, siehe den Abschnitt unten für Details. Die Liste wird durch eine ZBX_METRIC-Struktur mit dem Feld "key" auf NULL abgeschlossen.

```
void    zbx_module_item_timeout(int timeout);
```

Wenn das Modul zbx_module_item_list() exportiert, wird diese Funktion von Zabbix verwendet, um die Timeout-Einstellungen in der Zabbix-Konfigurationsdatei festzulegen, die von den durch das Modul implementierten Datenpunkt-Prüfungen eingehalten werden sollen. Hier ist der Parameter "timeout" in Sekunden angegeben.

```
ZBX_HISTORY_WRITE_CBS   zbx_module_history_write_cbs(void);
```

Diese Funktion sollte Callback-Funktionen zurückgeben, die der Zabbix Server zum Exportieren des Verlaufs verschiedener Datentypen verwendet. Die Callback-Funktionen werden als Felder der Struktur ZBX_HISTORY_WRITE_CBS bereitgestellt; Felder können NULL sein, wenn das Modul am Verlauf eines bestimmten Typs nicht interessiert ist.

```
int zbx_module_uninit(void);
```

Diese Funktion sollte die erforderliche Deinitialisierung durchführen, falls eine solche erforderlich ist, z. B. freigegebene Ressourcen freigeben, Dateideskriptoren schließen usw.

Alle Funktionen werden beim Start von Zabbix einmal aufgerufen, wenn das Modul geladen wird, mit Ausnahme von zbx_module_uninit(), das beim Herunterfahren von Zabbix einmal aufgerufen wird, wenn das Modul entladen wird.

Definieren von Datenpunkten

Jeder Datenpunkt wird in einer ZBX_METRIC-Struktur definiert:

```
typedef struct
{
    char        *key;
    unsigned    flags;
    int     (*function)();
    char        *test_param;
}
ZBX_METRIC;
```

Hier ist **key** der Datenpunkt-Schlüssel (z. B. "dummy.random"), **flags** entweder CF_HAVEPARAMS oder 0 (je nachdem, ob der Datenpunkt Parameter akzeptiert oder nicht), **function** eine C-Funktion, die den Datenpunkt implementiert (z. B. zbx_module_dummy_random), und **test_param** die Parameterliste, die verwendet werden soll, wenn der Zabbix-Agent mit dem -p-Flag gestartet wird (z. B. "1,1000", kann NULL sein). Eine Beispieldefinition könnte so aussehen:

```
static ZBX_METRIC keys[] =
{
    { "dummy.random", CF_HAVEPARAMS, zbx_module_dummy_random, "1,1000" },
    { NULL }
}
```

Jede Funktion, die einen Datenpunkt implementiert, sollte zwei Zeigerparameter akzeptieren: der erste vom Typ AGENT_REQUEST und der zweite vom Typ 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;
}
```

Diese Funktionen sollten SYSINFO_RET_OK zurückgeben, wenn der Wert des Datenpunkts erfolgreich ermittelt wurde. Andernfalls sollten sie SYSINFO_RET_FAIL zurückgeben. Siehe das unten stehende Beispielmodul "dummy" für Details dazu, wie Informationen aus AGENT_REQUEST gelesen und wie Informationen in AGENT_RESULT gesetzt werden.

Bereitstellung von Callbacks für den History-Export

Attention:
Der History-Export über ein Modul wird von Zabbix Proxy nicht mehr unterstützt.

Ein Modul kann Funktionen zum Exportieren von Verlaufsdaten nach Typ angeben: Numerisch (float), Numerisch (unsigned), Zeichen, Text und Log:

```
typedef struct
{
    void    (*history_float_cb)(const ZBX_HISTORY_FLOAT *history, int history_num);
    void    (*history_integer_cb)(const ZBX_HISTORY_INTEGER *history, int history_num);
    void    (*history_string_cb)(const ZBX_HISTORY_STRING *history, int history_num);
    void    (*history_text_cb)(const ZBX_HISTORY_TEXT *history, int history_num);
    void    (*history_log_cb)(const ZBX_HISTORY_LOG *history, int history_num);
}
ZBX_HISTORY_WRITE_CBS;
```

Jede dieser Funktionen sollte das `history`-Array mit `history_num` Elementen als Argumente übernehmen. Abhängig vom zu exportierenden History-Datentyp ist `history` jeweils ein Array der folgenden Strukturen:

```
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;
```

Callbacks werden von den History-Synchronisierungsprozessen des Zabbix-Servers am Ende des History-Synchronisationsvorgangs verwendet, nachdem die Daten in die Zabbix-Datenbank geschrieben und im Wertcache gespeichert wurden.

> **Attention:**
> Im Falle eines internen Fehlers im History-Export-Modul wird empfohlen, das Modul so zu implementieren, dass es nicht die gesamte Überwachung blockiert, bis es sich erholt, sondern stattdessen Daten verwirft und den Zabbix-Server weiterlaufen lässt.

Erstellen von Modulen

Module sind derzeit dafür vorgesehen, innerhalb des Zabbix-Quellbaums erstellt zu werden, da die Modul-API von einigen Datenstrukturen abhängt, die in Zabbix-Headerdateien definiert sind.

Der wichtigste Header für ladbare Module ist `include/module.h`, der diese Datenstrukturen definiert. Weitere erforderliche System-Header, die dazu beitragen, dass `include/module.h` ordnungsgemäß funktioniert, sind `stdlib.h` und `stdint.h`.

Mit diesen Informationen ist alles bereit, damit das Modul erstellt werden kann. Das Modul sollte `stdlib.h`, `stdint.h` und `module.h` einbinden, und das Build-Skript sollte sicherstellen, dass sich diese Dateien im Include-Pfad befinden. Weitere Details finden Sie im untenstehenden Beispielmodul "dummy".

Ein weiterer nützlicher Header ist `include/zbxcommon.h`, der die Funktion `zabbix_log()` definiert, die für Protokollierungs- und Debugging-Zwecke verwendet werden kann.

Konfigurationsparameter

Zabbix Agent, Server und Proxy unterstützen zwei Parameter für den Umgang mit Modulen:

- `LoadModulePath` - vollständiger Pfad zum Speicherort ladbarer Module
- `LoadModule` - beim Start zu ladende(s) Modul(e).

Die Module müssen sich in einem durch `LoadModulePath` angegebenen Verzeichnis befinden, oder der Pfad muss dem Modulnamen vorangestellt werden. Wenn der vorangestellte Pfad absolut ist (mit / beginnt), wird `LoadModulePath` ignoriert. Es sind mehrere `LoadModule`-Parameter zulässig.

Um beispielsweise den Zabbix Agent zu erweitern, könnten wir die folgenden Parameter hinzufügen:

```
LoadModulePath=/usr/local/lib/zabbix/agent/
LoadModule=mariadb.so
LoadModule=apache.so
LoadModule=kernel.so
LoadModule=/usr/local/lib/zabbix/dummy.so
```

Beim Start des Agents lädt dieser die Module mariadb.so, apache.so und kernel.so aus dem Verzeichnis `/usr/local/lib/zabbix/agent`, während dummy.so aus `/usr/local/lib/zabbix` geladen wird. Der Agent kann nicht gestartet werden, wenn ein Modul fehlt, die Berechtigungen nicht ausreichen oder eine gemeinsam genutzte Bibliothek kein Zabbix-Modul ist.

Frontend-Konfiguration

Ladbare Module werden von Zabbix Agent, Server und Proxy unterstützt. Daher hängt der Datenpunkt-Typ im Zabbix Frontend davon ab, wo das Modul geladen wird. Wenn das Modul in den Agent geladen wird, sollte der Datenpunkt-Typ "Zabbix agent" oder "Zabbix agent (active)" sein. Wenn das Modul in Server oder Proxy geladen wird, sollte der Datenpunkt-Typ "Simple check" sein.

Der History-Export über Zabbix-Module erfordert keine Frontend-Konfiguration. Wenn das Modul erfolgreich vom Server geladen wird und die Funktion `zbx_module_history_write_cbs()` bereitstellt, die mindestens eine nicht-NULL-Callback-Funktion zurückgibt, wird der History-Export automatisch aktiviert.

Dummy-Modul

Zabbix enthält ein Beispielmodul, das in der Programmiersprache C geschrieben ist. Das Modul befindet sich unter `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
```

Das Modul ist gut dokumentiert und kann als Vorlage für Ihre eigenen Module verwendet werden.

Nachdem `./configure` wie oben beschrieben im Stammverzeichnis des Zabbix-Quellbaums ausgeführt wurde, führen Sie einfach `make` aus, um `dummy.so` zu erstellen.

```
/*
** 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"

/* die Variable speichert die Timeout-Einstellung für die Verarbeitung von Datenpunkten */
static int  item_timeout = 0;

/* das Modul SOLLTE interne Funktionen als static definieren und ein Namensschema verwenden, das sich von
/* Zabbix-Symbolen (zbx_*) und den API-Funktionen ladbarer Module (zbx_module_*) unterscheidet, um Konflik
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;
}
```

```c
/******************************************************************************
 *                                                                            *
 * 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             *
 *                                                                            *
 ******************************************************************************/
```

```c
int zbx_module_init(void)
{
    /* initialization for dummy.random */
    srand(time(NULL));

    return ZBX_MODULE_OK;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_uninit                                                *
 *                                                                            *
 * Purpose: the function is called on agent shutdown                          *
 *          It should be used to cleanup used resources if there are any      *
 *                                                                            *
 * Return value: ZBX_MODULE_OK - success                                      *
 *               ZBX_MODULE_FAIL - function failed                            *
 *                                                                            *
 ******************************************************************************/
int zbx_module_uninit(void)
{
    return ZBX_MODULE_OK;
}

/******************************************************************************
 *                                                                            *
 * Functions: dummy_history_float_cb                                          *
 *            dummy_history_integer_cb                                        *
 *            dummy_history_string_cb                                         *
 *            dummy_history_text_cb                                           *
 *            dummy_history_log_cb                                            *
 *                                                                            *
 * Purpose: callback functions for storing historical data of types float,    *
 *          integer, string, text and log respectively in external storage    *
 *                                                                            *
 * Parameters: history     - array of historical data                         *
 *             history_num - number of elements in history array              *
 *                                                                            *
 ******************************************************************************/
static void dummy_history_float_cb(const ZBX_HISTORY_FLOAT *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_integer_cb(const ZBX_HISTORY_INTEGER *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_string_cb(const ZBX_HISTORY_STRING *history, int history_num)
{
    int i;
```

```
    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_text_cb(const ZBX_HISTORY_TEXT *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

static void dummy_history_log_cb(const ZBX_HISTORY_LOG *history, int history_num)
{
    int i;

    for (i = 0; i < history_num; i++)
    {
        /* do something with history[i].itemid, history[i].clock, history[i].ns, history[i].value, ... */
    }
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_history_write_cbs                                     *
 *                                                                            *
 * Purpose: returns a set of module functions Zabbix will call to export      *
 *          different types of historical data                                *
 *                                                                            *
 * Return value: structure with callback function pointers (can be NULL if    *
 *               module is not interested in data of certain types)           *
 *                                                                            *
 ******************************************************************************/
ZBX_HISTORY_WRITE_CBS   zbx_module_history_write_cbs(void)
{
    static ZBX_HISTORY_WRITE_CBS    dummy_callbacks =
    {
        dummy_history_float_cb,
        dummy_history_integer_cb,
        dummy_history_string_cb,
        dummy_history_text_cb,
        dummy_history_log_cb,
    };

    return dummy_callbacks;
}
```

Das Modul exportiert drei neue Datenpunkte:

- `dummy.ping` - gibt immer '1' zurück
- `dummy.echo[param1]` - gibt den ersten Parameter unverändert zurück, zum Beispiel gibt `dummy.echo[ABC]` ABC zurück
- `dummy.random[param1, param2]` - gibt eine Zufallszahl innerhalb des Bereichs param1-param2 zurück, zum Beispiel `dummy.random[1,1000000]`

Einschränkungen

Die Unterstützung für ladbare Module ist nur für die Unix-Plattform implementiert. Das bedeutet, dass sie für Windows-Agenten nicht funktioniert.

In einigen Fällen muss ein Modul möglicherweise modulbezogene Konfigurationsparameter aus `zabbix_agentd.conf` lesen. Dies

wird derzeit nicht unterstützt. Wenn Ihr Modul einige Konfigurationsparameter verwenden muss, sollten Sie wahrscheinlich das Einlesen einer modulspezifischen Konfigurationsdatei implementieren.

Frontend-Module

Übersicht

Es ist möglich, die Funktionalität des Zabbix Frontend durch das Hinzufügen von Modulen von Drittanbietern oder durch die Entwicklung eigener Module zu erweitern, ohne den Quellcode von Zabbix ändern zu müssen.

Beachten Sie, dass der Modulcode mit denselben Berechtigungen wie der Zabbix-Quellcode ausgeführt wird. Das bedeutet:

- Module von Drittanbietern können schädlich sein. Sie müssen den Modulen vertrauen, die Sie installieren.
- Fehler im Code eines Moduls von Drittanbietern können das Frontend zum Absturz bringen. In diesem Fall entfernen Sie einfach den Modulcode aus dem Frontend. Sobald Sie das Zabbix Frontend neu laden, wird eine Meldung angezeigt, dass einige Module fehlen. Gehen Sie zu Modulverwaltung (unter *Administration* > *Allgemein* > *Module*) und klicken Sie erneut auf *Verzeichnis scannen*, um nicht vorhandene Module aus der Datenbank zu entfernen.

Installation

Bitte lesen Sie immer das Installationshandbuch für ein bestimmtes Modul. Es wird empfohlen, neue Module nacheinander zu installieren, um Fehler leichter zu erkennen.

Unmittelbar bevor Sie ein Modul installieren:

- Stellen Sie sicher, dass Sie das Modul aus einer vertrauenswürdigen Quelle heruntergeladen haben. Die Installation schädlichen Codes kann zu Folgen wie Datenverlust führen.
- Verschiedene Versionen desselben Moduls (gleiche ID) können parallel installiert werden, aber jeweils nur eine Version kann gleichzeitig aktiviert sein.

Schritte zur Installation eines Moduls:

- Entpacken Sie Ihr Modul in einen eigenen Ordner im Verzeichnis `modules` der Zabbix-Frontend-Installation (zum Beispiel `zabbix/ui/modules`).
- Stellen Sie sicher, dass Ihr Modulverzeichnis mindestens die Datei `manifest.json` enthält.
- Navigieren Sie zu Modulverwaltung und klicken Sie auf die Schaltfläche *Verzeichnis scannen*.
- Das neue Modul wird zusammen mit Version, Autor, Beschreibung und Status in der Liste angezeigt.
- Aktivieren Sie das Modul, indem Sie auf seinen Status klicken.

Fehlerbehebung:

Problem	Lösung
Modul wurde nicht in der Liste angezeigt	Wenn Ihr Modul nicht in der Liste angezeigt wurde, stellen Sie sicher, dass `manifest.json` in `modules/your-module/` vorhanden ist. Falls nicht, haben Sie das Modul möglicherweise in das falsche Verzeichnis entpackt. Falls doch, ist das Modul möglicherweise nicht mit Ihrer Zabbix-Version kompatibel. Prüfen Sie außerdem, ob der Webserver-Benutzer mindestens Lese- und Suchzugriff (`r-x`) auf den Modulordner und alle Unterverzeichnisse sowie Lesezugriff (`r--`) auf alle darin enthaltenen Dateien hat.
Frontend abgestürzt	Der Modulcode ist nicht mit der aktuellen Zabbix-Version oder der Serverkonfiguration kompatibel. Bitte löschen Sie die Moduldateien und laden Sie das Frontend neu. Sie werden einen Hinweis sehen, dass einige Module fehlen. Gehen Sie zu Modulverwaltung und klicken Sie erneut auf *Verzeichnis scannen*, um nicht vorhandene Module aus der Datenbank zu entfernen.
Fehlermeldung über identischen Namespace, ID oder Aktionen erscheint	Das neue Modul hat versucht, einen Namespace, eine ID oder Aktionen zu registrieren, die bereits von anderen aktivierten Modulen registriert sind. Deaktivieren Sie das konfliktverursachende Modul (in der Fehlermeldung genannt), bevor Sie das neue aktivieren.
Technische Fehlermeldungen erscheinen	Melden Sie Fehler dem Entwickler des Moduls.

Entwicklung von Modulen

Informationen zur Entwicklung benutzerdefinierter Module finden Sie im Developer center.

Anhänge

Überblick Dieser Abschnitt enthält Referenzmaterialien, Konfigurationsparameter, technische Spezifikationen und Verfahrensanleitungen, die das Haupthandbuch ergänzen.

Wichtige Referenz

- Installation und Einrichtung - Erstellung der Datenbank sowie Installation und Konfiguration von Server und Agent.
- Prozesskonfiguration - Konfigurationsparameter für Zabbix Server, Proxy, Agent und mehr.
- Datenpunkte - Referenz zu Datenpunktschlüsseln und unterstützten Datentypen.
- Makros - Syntax und Verwendung integrierter und benutzerdefinierter Makros.

Installation und Einrichtung

Dieser Abschnitt behandelt die Erstellung der Datenbank, die Sicherheitskonfiguration, plattformspezifische Einrichtungsverfahren und die Konfiguration von Funktionen.

Bitte verwenden Sie die Seitenleiste, um das für Ihre Umgebung relevante Verfahren zu finden.

Datenbankerstellung

Übersicht

Während der Installation von Zabbix Server oder Proxy muss eine Zabbix-Datenbank erstellt werden.

Dieser Abschnitt enthält Anweisungen zum Erstellen einer Zabbix-Datenbank. Für jede unterstützte Datenbank steht ein separater Satz von Anweisungen zur Verfügung.

> **Note:**
> Um die Sicherheit der Datenbank zu verbessern, indem Datenbankrollen/-benutzer mit minimalen Berechtigungen erstellt werden, siehe die Best Practices zum Erstellen von Datenbanken für die jeweilige unterstützte Datenbank:

> - MySQL/MariaDB
> - PostgreSQL/TimescaleDB
>
> Informationen zum Konfigurieren sicherer TLS-Verbindungen finden Sie unter Sichere Verbindung zur Datenbank.

UTF-8 ist die einzige von Zabbix unterstützte Kodierung. Es ist bekannt, dass sie ohne Sicherheitsprobleme funktioniert. Benutzer sollten sich bewusst sein, dass bei der Verwendung einiger anderer Kodierungen bekannte Sicherheitsprobleme bestehen. Informationen zum Wechsel zu UTF-8 finden Sie unter Reparieren des Zeichensatzes und der Sortierung der Zabbix-Datenbank. Siehe auch Grenzen der Filterung mit utf8mb4-Sortierungen.

> **Note:**
> Wenn Sie aus dem Zabbix Git repository installieren, müssen Sie vor dem Fortfahren mit den nächsten Schritten den folgenden Befehl ausführen:

 `make dbschema`

MySQL/MariaDB

Der Zeichensatz utf8mb4 mit der Sortierung utf8mb4_bin wird unterstützt, damit Zabbix Server/Proxy ordnungsgemäß mit MySQL/MariaDB-Datenbanken arbeiten können.
 Um zukünftige Probleme zu vermeiden, wird dringend empfohlen, sowohl für neue als auch für bestehende Installationen den Zeichensatz utf8mb4 mit der Sortierung utf8mb4_bin zu verwenden. Um Datenbanken, die den Zeichensatz utf8mb3 und die Sortierung utf8mb3_bin verwenden, in den Zeichensatz utf8mb4 und die Sortierung utf8mb4_bin zu konvertieren, befolgen Sie die Anweisungen unter Reparieren des Zeichensatzes und der Sortierung der Zabbix-Datenbank.

Deterministische Auslöser müssen während des Imports des Schemas erstellt werden. Unter MySQL und MariaDB muss dazu `GLOBAL log_bin_trust_function_creators = 1` gesetzt werden, wenn die binäre Protokollierung aktiviert ist,

keine Superuser-Berechtigungen vorhanden sind und `log_bin_trust_function_creators = 1` nicht in der MySQL-Konfigurationsdatei gesetzt ist.

Wenn Sie Zabbix aus Paketen oder Containern installieren, beachten Sie die jeweiligen Anweisungen.

Wenn Sie Zabbix aus den Quellen installieren:

- Erstellen und konfigurieren Sie eine Datenbank und einen Benutzer. Verwenden Sie aus Sicherheitsgründen ein starkes, eindeutiges Passwort für den Zabbix-Datenbankbenutzer.

```
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;
```

- Importieren Sie die Daten in die Datenbank und legen Sie den Zeichensatz utf8mb4 als Standard fest (vorausgesetzt, Sie befinden sich im Stammverzeichnis der Zabbix-Quellen). Für eine Zabbix-Proxy-Datenbank sollte nur `schema.sql` importiert werden (weder images.sql noch data.sql).

```
cd database/mysql
mysql -uzabbix -p<password> zabbix < schema.sql
#### hier anhalten, wenn Sie eine Datenbank für den Zabbix-Proxy erstellen
mysql -uzabbix -p<password> zabbix < images.sql
mysql -uzabbix -p<password> --default-character-set=utf8mb4 zabbix < data.sql
```

`log_bin_trust_function_creators` kann deaktiviert werden, nachdem das Schema erfolgreich importiert wurde:

```
mysql -uroot -p<password>

mysql> SET GLOBAL log_bin_trust_function_creators = 0;
mysql> quit;
```

PostgreSQL

Sie benötigen einen Datenbankbenutzer mit Berechtigungen zum Erstellen von Datenbankobjekten.

Wenn Sie Zabbix aus Paketen oder Containern installieren, beachten Sie die jeweiligen Anweisungen.

Wenn Sie Zabbix aus den Quellen installieren:

- Erstellen Sie einen Datenbankbenutzer.

Der folgende Shell-Befehl erstellt den Benutzer `zabbix`. Geben Sie bei der Aufforderung ein Passwort ein und wiederholen Sie das Passwort (beachten Sie, dass Sie möglicherweise zuerst nach dem sudo-Passwort gefragt werden):

```
sudo -u postgres createuser --pwprompt zabbix
```

- Erstellen Sie eine Datenbank.

Der folgende Shell-Befehl erstellt die Datenbank `zabbix` (letzter Parameter) mit dem zuvor erstellten Benutzer als Eigentümer (`-O zabbix`).

```
sudo -u postgres createdb -O zabbix -E Unicode -T template0 zabbix
```

- Importieren Sie das anfängliche Schema und die Daten (unter der Annahme, dass Sie sich im Stammverzeichnis der Zabbix-Quellen befinden). Für eine Zabbix-Proxy-Datenbank sollte nur `schema.sql` importiert werden (weder `images.sql` noch `data.sql`).

```
cd database/postgresql
cat schema.sql | sudo -u zabbix psql zabbix
#### hier anhalten, wenn Sie die Datenbank für Zabbix-Proxy erstellen
cat images.sql | sudo -u zabbix psql zabbix
cat data.sql | sudo -u zabbix psql zabbix
```

Attention:
Die obigen Befehle sind als Beispiel angegeben und funktionieren in den meisten GNU/Linux-Installationen. Sie können je nach Konfiguration Ihres Systems bzw. Ihrer Datenbank andere Befehle verwenden, zum Beispiel:

 `psql -U <username>`

 Wenn Sie Probleme bei der Einrichtung der Datenbank haben, wenden Sie sich bitte an Ihren Datenbankadministrator.

TimescaleDB

Anweisungen zum Erstellen und Konfigurieren von TimescaleDB finden Sie in einem separaten Abschnitt.

Elasticsearch und ClickHouse

Zabbix kann Verlaufsdaten auch in Elasticsearch oder ClickHouse speichern. Anweisungen zur Einrichtung finden Sie auf den verlinkten Seiten.

SQLite

Die Verwendung von SQLite wird nur für **Zabbix Proxy** unterstützt!

Die Datenbank wird automatisch erstellt, wenn sie nicht existiert.

Kehren Sie zum Installationsabschnitt zurück.

Reparieren des Zeichensatzes und der Sortierung der Zabbix-Datenbank

MySQL/MariaDB

Historisch wurde bei MySQL und seinen Derivaten `utf8` als Alias für utf8mb3 verwendet - MySQLs eigene 3-Byte-Implementierung von UTF8, das 4-Byte ist. Ab MySQL 8.0.28 und MariaDB 10.6.1 ist der Zeichensatz `utf8mb3` veraltet, und zu einem späteren Zeitpunkt wird seine Unterstützung entfernt, während `utf8` zu einem Verweis auf `utf8mb4` wird. Seit Zabbix 6.0 wird `utf8mb4` unterstützt. Um zukünftige Probleme zu vermeiden, wird dringend empfohlen, `utf8mb4` zu verwenden. Ein weiterer Vorteil der Umstellung auf `utf8mb4` ist die Unterstützung zusätzlicher Unicode-Zeichen.

Warning:
Da Versionen vor Zabbix 6.0 `utf8mb4` nicht kennen, stellen Sie bitte sicher, dass Sie zuerst den Zabbix Server und das DB-Schema auf 6.0.x oder neuer aktualisieren, bevor Sie die `utf8mb4`-Konvertierung ausführen.

1. Prüfen Sie den Zeichensatz und die Sortierung der Datenbank.

Zum Beispiel:

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| latin2                   | latin2 _general_ci   |
+--------------------------+----------------------+
```

Oder:

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| utf8                     | utf8_bin             |
+--------------------------+----------------------+
```

Wie wir sehen, ist der Zeichensatz hier nicht `utf8mb4` und die Sortierung ist nicht `utf8mb4\_bin`, daher müssen wir sie korrigieren.

2. Stoppen Sie Zabbix.

3. Erstellen Sie eine Sicherungskopie der Datenbank!

4. Korrigieren Sie den Zeichensatz und die Sortierung auf Datenbankebene:

```
alter database <your DB name> character set utf8mb4 collate utf8mb4_bin;
```

Korrigierte Werte:

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| utf8mb4                  | utf8mb4_bin          |
+--------------------------+----------------------+
```

5. Laden Sie das Skript , um Zeichensatz und Sortierung auf Tabellen- und Spaltenebene zu korrigieren:

```
mysql <your DB name> < utf8mb4_convert.sql
```

6. Führen Sie das Skript aus:

```
                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;
```

Bitte beachten Sie, dass `utf8mb4` voraussichtlich etwas mehr Speicherplatz auf der Festplatte benötigt.

7. Wenn keine Fehler aufgetreten sind, können Sie eine Sicherungskopie der Datenbank mit der korrigierten Datenbank erstellen.

8. Starten Sie Zabbix.

Datenbank-Upgrade auf Primärschlüssel

Übersicht

Dieser Abschnitt enthält Anweisungen zum manuellen Upgrade von Tabellen in bestehenden Installationen auf Primärschlüssel.

Das Upgrade auf Primärschlüssel optimiert, wie Daten indiziert und abgerufen werden, wodurch Abfragen möglicherweise schneller ausgeführt werden und Speicherplatz gespart wird. Außerdem verbessert es die Datenverwaltung und Synchronisierung in Cluster-Setups, unterstützt die Skalierung und stellt sicher, dass das System auch dann zuverlässig bleibt, wenn einige Server ausfallen.

> **Attention:**
> Die auf dieser Seite bereitgestellten Anweisungen sind für fortgeschrittene Benutzer gedacht und müssen möglicherweise an Ihre spezifische Konfiguration angepasst werden. Das Upgrade auf Primärschlüssel kann zeitaufwendig und ressourcenintensiv sein. Stellen Sie sicher, dass genügend freier Speicherplatz verfügbar ist; je nach Größe Ihrer Datenbank und der gespeicherten Daten kann der Vorgang bis zu 2,5-mal so viel Speicherplatz benötigen, wie derzeit von den History-Tabellen verwendet wird.

Primärschlüssel werden seit Zabbix 6.0 für alle Tabellen in Neuinstallationen verwendet.

Es gibt kein automatisches Datenbank-Upgrade auf Primärschlüssel; bestehende Installationen können jedoch manuell aktualisiert werden, **nachdem** der Zabbix Server auf 6.0 oder neuer aktualisiert wurde.

> **Attention:**
> Seit Zabbix 7.0 führt das Upgrade von Tabellen auf Primärschlüssel auch dazu, dass Tabellen Double-Precision-Datentypen verwenden.

 Wenn Sie Zabbix 7.0 (oder neuer) verwenden, nutzen die Tabellen bereits Double Precision. Die Anweisungen auf dieser Seite können jedoch weiterhin verwendet werden, um Tabellen auf Primärschlüssel zu aktualisieren, ohne Tabellen zu beeinflussen, die bereits Double Precision verwenden.

 Wenn Sie Zabbix 6.4 (oder älter) verwenden, sollten Sie erwägen, die Tabellen zuerst auf Double Precision zu aktualisieren. Weitere Informationen finden Sie unter Upgrading to numeric values of extended range in der Zabbix 7.0-Dokumentation.

Anweisungen sind verfügbar für:

- MariaDB/MySQL
- PostgreSQL
- TimescaleDB

Wichtige Hinweise

So führen Sie das Datenbank-Upgrade durch:

1. Stoppen Sie den Zabbix Server.

Es wird dringend empfohlen, den Zabbix Server für die Dauer des Upgrades anzuhalten. Falls jedoch unbedingt erforderlich, können Sie das Upgrade auch durchführen, während der Server läuft (nur für MySQL, MariaDB und PostgreSQL ohne TimescaleDB).

2. Erstellen Sie eine Sicherung Ihrer Datenbank.
3. Installieren Sie das neueste Paket zabbix-sql-scripts, das mit Ihrer Zabbix-Version kompatibel ist (z. B. für RHEL: `dnf install zabbix-sql-scripts`).
4. Führen Sie die Skripte für Ihre Datenbank aus.
5. Starten Sie den Zabbix Server.

Warning:
Führen Sie die Skripte nur für die Server-Datenbank aus. Der Proxy profitiert nicht von diesem Upgrade.

Wenn die Datenbank Partitionen verwendet, wenden Sie sich an den DB-Administrator oder den Zabbix-Support, um Hilfe zu erhalten.

CSV-Dateien können nach einem erfolgreichen Upgrade auf Primärschlüssel entfernt werden.

Optional kann das Zabbix Frontend in den Wartungsmodus versetzt werden.

MariaDB/MySQL

Export und Import müssen in tmux/screen ausgeführt werden, um sicherzustellen, dass die Sitzung nicht unterbrochen wird.

Siehe auch: Wichtige Hinweise

MySQL 8.4+ mit mysqlsh

Diese Methode kann mit einem laufenden Zabbix Server verwendet werden, es wird jedoch empfohlen, den Server für die Dauer des Upgrades zu stoppen. Die MySQL Shell (*mysqlsh*) muss installiert sein und eine Verbindung zur DB herstellen können.

- Melden Sie sich in der MySQL-Konsole als root an (empfohlen) oder als beliebiger Benutzer mit FILE-Rechten.
- Starten Sie MySQL mit aktivierter Variable local_infile.
- Benennen Sie alte Tabellen um und erstellen Sie neue Tabellen, indem Sie `history_upgrade_prepare.sql` ausführen.

```
mysql -uzabbix -p<password> zabbix < /usr/share/zabbix/sql-scripts/mysql/option-patches/history_upgrade_pr
```

- Exportieren und importieren Sie die Daten.

Verbinden Sie sich über mysqlsh. Wenn eine Socket-Verbindung verwendet wird, muss möglicherweise der Pfad angegeben werden.

```
sudo mysqlsh -uroot -S /run/mysqld/mysqld.sock --no-password -Dzabbix
```

Wechseln Sie mit folgendem Befehl in den JavaScript-Modus:

```
\js
```

Führen Sie dann den folgenden Code aus (CSVPATH kann bei Bedarf geändert werden):

```
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" });
```

Wenn die Meldung "JavaScript is not supported" angezeigt wird, unterstützt Ihre MySQL Shell-Installation kein JS. Installieren Sie in diesem Fall das offizielle MySQL Shell-Paket von Oracle (oder erstellen Sie es aus dem Quellcode), damit der JavaScript-Modus aktiviert ist.

- Folgen Sie den Anweisungen nach der Migration, um die alten Tabellen zu löschen.

MariaDB/MySQL 8.4+ ohne mysqlsh

Diese Upgrade-Methode nimmt mehr Zeit in Anspruch und sollte nur verwendet werden, wenn ein Upgrade mit *mysqlsh* nicht möglich ist.

Tabellen-Upgrade

- Melden Sie sich in der MySQL-Konsole als root an (empfohlen) oder als beliebiger Benutzer mit FILE-Rechten.
- Wenn Sie die Migration mit einem laufenden Zabbix-Server durchführen, starten Sie MySQL mit aktivierter Variable local_infile.
- Benennen Sie die alten Tabellen um und erstellen Sie neue Tabellen, indem Sie `history_upgrade_prepare.sql` ausführen:

```
mysql -uzabbix -p<password> zabbix < /usr/share/zabbix/sql-scripts/mysql/option-patches/history_upgrade_pr
```

Migration mit gestopptem Server

`max_execution_time` (in MySQL) oder `max_statement_time` (in MariaDB) muss vor der Migration der Daten deaktiviert werden, um ein Timeout während der Migration zu vermeiden.

Für MySQL:

```
SET @@max_execution_time=0;
```

Für 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;
```

Befolgen Sie die Anweisungen nach der Migration, um die alten Tabellen zu löschen.

Migration mit laufendem Server

Prüfen Sie, für welche Pfade Import/Export aktiviert ist:

```
mysql> SELECT @@secure_file_priv;
+-----------------------+
| @@secure_file_priv    |
+-----------------------+
| /var/lib/mysql-files/ |
+-----------------------+
```

Wenn der Wert von *secure_file_priv* ein Pfad zu einem Verzeichnis ist, wird der Export/Import für Dateien in diesem Verzeichnis durchgeführt. Passen Sie in diesem Fall die Pfade zu den Dateien in den Abfragen entsprechend an oder setzen Sie den Wert von *secure_file_priv* für die Dauer des Upgrades auf eine leere Zeichenfolge.

Wenn der Wert von *secure_file_priv* leer ist, kann der Export/Import von jedem Speicherort aus durchgeführt werden.

Wenn der Wert von *secure_file_priv* NULL ist, setzen Sie ihn auf den Pfad, der die exportierten Tabellendaten enthält (im obigen Beispiel '/var/lib/mysql-files/').

Weitere Informationen finden Sie in der MySQL-Dokumentation oder in der MariaDB-Dokumentation.

`max_execution_time` (in MySQL) oder `max_statement_time` (in MariaDB) muss vor dem Export der Daten deaktiviert werden, um ein Timeout während des Exports zu vermeiden.

Für MySQL:

```
SET @@max_execution_time=0;
```

Für 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
```

Befolgen Sie die Anweisungen nach der Migration, um die alten Tabellen zu löschen.

PostgreSQL

Export und Import müssen in tmux/screen durchgeführt werden, um sicherzustellen, dass die Sitzung nicht getrennt wird. Bei Installationen mit TimescaleDB überspringen Sie diesen Abschnitt und fahren Sie mit PostgreSQL + TimescaleDB fort.

Siehe auch: Wichtige Hinweise

Tabellenupgrade

- Benennen Sie Tabellen mit `history_upgrade_prepare.sql` um:

```
sudo -u zabbix psql zabbix < /usr/share/zabbix/sql-scripts/postgresql/option-patches/history_upgrade_prepa
```

Migration mit gestopptem Server

- Exportieren Sie den aktuellen Verlauf, importieren Sie ihn in die temporäre Tabelle und fügen Sie dann die Daten unter Ignorieren von Duplikaten in die neuen Tabellen ein:

```
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;
```

Siehe Tipps zur Verbesserung der INSERT-Performance: PostgreSQL: Bulk Loading Huge Amounts of Data, Checkpoint Distance and Amount of WAL.

- Folgen Sie den Anweisungen nach der Migration, um die alten Tabellen zu löschen.

Migration mit laufendem Server

- Exportieren Sie den aktuellen Verlauf, importieren Sie ihn in die temporäre Tabelle und fügen Sie die Daten dann unter Ignorieren von Duplikaten in die neuen Tabellen ein:

```
\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;
```

- Folgen Sie den Anweisungen nach der Migration, um die alten Tabellen zu löschen.

PostgreSQL + TimescaleDB

Export und Import müssen in tmux/screen durchgeführt werden, um sicherzustellen, dass die Sitzung nicht getrennt wird. Der Zabbix Server sollte während des Upgrades heruntergefahren sein.

Siehe auch: Wichtige Hinweise

- Benennen Sie Tabellen mit `history_upgrade_prepare.sql` um.
  - Wenn die Komprimierung aktiviert ist (bei der Standardinstallation), führen Sie das Skript aus `/usr/share/zabbix/sql-script` aus:

    ```
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    ```

  - Wenn die Komprimierung deaktiviert ist, führen Sie das Skript aus `/usr/share/zabbix/sql-scripts/postgresql/timesca` aus:

    ```
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
    ```

- Führen Sie die TimescaleDB-Hypertable-Migrationsskripte basierend auf den Komprimierungseinstellungen aus:
  - Wenn die Komprimierung aktiviert ist (bei der Standardinstallation), führen Sie die Skripte aus `/usr/share/zabbix/sql-script` aus:

    ```
    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
    ```

  - Wenn die Komprimierung deaktiviert ist, führen Sie die Skripte aus `/usr/share/zabbix/sql-scripts/postgresql/timesca` aus:

```
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
```

Siehe auch: Tipps zur Verbesserung der INSERT-Performance.

- Befolgen Sie die Anweisungen nach der Migration, um die alten Tabellen zu löschen.

Nach der Migration

Führen Sie für alle Datenbanken nach Abschluss der Migration Folgendes durch:

- Überprüfen Sie, dass alles wie erwartet funktioniert.
- Löschen Sie die alten Tabellen:

```
DROP TABLE history_old;
DROP TABLE history_uint_old;
DROP TABLE history_str_old;
DROP TABLE history_log_old;
DROP TABLE history_text_old;
```

- Löschen Sie bei TimescaleDB zusätzlich die folgende alte Tabelle:

```
DROP TABLE trends_old;
```

Siehe auch

- Vorbereiten der Auditlog-Tabelle für die Partitionierung

Vorbereitung der auditlog-Tabelle für die Partitionierung

Überblick

Einige Datenbanken (zum Beispiel MySQL) erfordern, dass die Partitionierungsspalte Teil der eindeutigen Einschränkung der Tabelle ist. Daher muss zum Partitionieren der Tabelle `auditlog` nach Zeit der Primärschlüssel von `auditid` in einen zusammengesetzten Schlüssel `auditid` + `clock` geändert werden.

Dieser Abschnitt enthält Anweisungen zum Ändern des Primärschlüssels der Tabelle `auditlog`.

> **Attention:**
> Die auf dieser Seite bereitgestellten Anweisungen sind für fortgeschrittene Benutzer gedacht. Beachten Sie, dass diese Anweisungen möglicherweise an Ihre spezifische Konfiguration angepasst werden müssen. Das Ändern des Primärschlüssels kann außerdem mit zukünftigen Upgrade-Patches inkompatibel sein, sodass zukünftige Upgrades möglicherweise manuell durchgeführt werden müssen.

 Das Ändern des Primärschlüssels kann je nach Größe der Tabelle `auditlog` ein ressourcenintensiver und zeitaufwändiger Vorgang sein. Es wird empfohlen, den Zabbix Server anzuhalten und das Zabbix Frontend für die Dauer der Änderung in den Wartungsmodus zu versetzen. Falls jedoch unbedingt erforderlich, gibt es eine Möglichkeit, den Primärschlüssel ohne Ausfallzeit zu ändern (siehe unten).

Die Partitionierung der Tabelle `auditlog` kann beispielsweise das Housekeeping in großen Umgebungen verbessern. Obwohl das Zabbix-Housekeeping derzeit partitionierte Tabellen nicht nutzen kann (außer bei TimescaleDB), können Sie das Zabbix-Housekeeping deaktivieren und Partitionen mithilfe von Skripten löschen.

Seit Zabbix 7.0 wurde die Tabelle `auditlog` für TimescaleDB in eine Hypertabelle umgewandelt, wodurch der Housekeeper Daten in Chunks löschen kann. Informationen zum Upgrade der bestehenden Tabelle `auditlog` auf eine Hypertabelle finden Sie unter Upgrading TimescaleDB schema.

MySQL

Wichtige Hinweise zum Neuerstellen von Indizes

MySQL erstellt Indizes für den Primärschlüssel während der `ALTER TABLE`-Operation automatisch neu. Es wird jedoch dringend empfohlen, Indizes zusätzlich manuell mit der Anweisung `OPTIMIZE TABLE` neu zu erstellen, um eine optimale Datenbankleistung sicherzustellen.

Das Neuerstellen von Indizes kann vorübergehend zusätzlichen Speicherplatz in einer Größenordnung erfordern, die der von der Tabelle selbst belegten Größe entspricht. Um die aktuelle Größe von Daten und Indizes zu ermitteln, können Sie die folgenden Anweisungen ausführen:

```
ANALYZE TABLE auditlog;
SHOW TABLE STATUS LIKE 'auditlog';
```

Wenn der verfügbare Speicherplatz ein Problem darstellt, folgen Sie den Anweisungen unter Primärschlüssel ohne Ausfallzeit ändern. Es sind auch andere Optionen verfügbar:

- Eine Erhöhung des MySQL-Parameters `sort_buffer_size` kann dazu beitragen, den Speicherplatzverbrauch beim manuellen Neuerstellen von Indizes zu verringern. Allerdings kann die Änderung dieser Variable den gesamten Speicherverbrauch der Datenbank beeinflussen.
- Erwägen Sie, Speicherplatz freizugeben, indem Sie möglicherweise unnötige Daten löschen.
- Erwägen Sie, den Parameter *Data storage period* housekeeper vor dem Ausführen des housekeeper zu verringern.

Ändern des Primärschlüssels mit Ausfallzeit

1. Entfernen Sie den aktuellen Primärschlüssel der Tabelle `auditlog` und fügen Sie den neuen Primärschlüssel hinzu.

```
ALTER TABLE auditlog DROP PRIMARY KEY, ADD PRIMARY KEY (auditid, clock);
```

2. Erstellen Sie die Indizes neu (optional, aber sehr empfohlen, siehe Wichtige Hinweise zum Neuerstellen von Indizes).

```
OPTIMIZE TABLE auditlog;
```

Ändern des Primärschlüssels ohne Ausfallzeit

Die manuelle Methode zum Ändern des Primärschlüssels wird hier beschrieben. Alternativ können Sie das Toolkit pt-online-schema-change von Percona verwenden. Dieses Toolkit führt die folgenden Aktionen automatisch aus und minimiert dabei gleichzeitig den Speicherplatz, der für die Änderung der Tabelle `auditlog` benötigt wird.

1. Erstellen Sie eine neue Tabelle mit dem neuen Primärschlüssel und erstellen Sie Indizes.

```
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. Tauschen Sie die Tabellen aus.

```
RENAME TABLE auditlog TO auditlog_old, auditlog_new TO auditlog;
```

3. Kopieren Sie die Daten von der alten Tabelle in die neue Tabelle.

```
INSERT INTO auditlog SELECT * FROM auditlog_old;
```

Dies kann in Blöcken erfolgen (mehrere `INSERT INTO`-Anweisungen mit `WHERE clock`-Klauseln nach Bedarf), um eine übermäßige Ressourcennutzung zu vermeiden.

4. Löschen Sie die alte Tabelle.

```
DROP TABLE auditlog_old;
```

PostgreSQL

Wichtige Hinweise zum Neuerstellen von Indizes

PostgreSQL erstellt Indizes für den Primärschlüssel während der `ALTER TABLE`-Operation automatisch neu. Es wird jedoch dringend empfohlen, Indizes auch manuell mit der Anweisung `REINDEX TABLE CONCURRENTLY` neu zu erstellen, um eine optimale Datenbankleistung sicherzustellen.

Das Neuerstellen von Indizes kann vorübergehend bis zum Dreifachen des aktuell von Indizes belegten Speicherplatzes auf dem Datenträger erfordern. Um die aktuelle Größe der Indizes zu ermitteln, können Sie die folgende Abfrage ausführen:

```
SELECT pg_size_pretty(pg_indexes_size('auditlog'));
```

Wenn der verfügbare Speicherplatz ein Problem darstellt, befolgen Sie die Anweisungen unter Primärschlüssel ohne Ausfallzeit ändern. Es sind auch andere Optionen verfügbar:

- Eine Erhöhung des PostgreSQL-Parameters `maintenance_work_mem` kann helfen, den Speicherplatzverbrauch beim manuellen Neuerstellen von Indizes zu reduzieren. Allerdings kann die Änderung dieser Variable die gesamte Speichernutzung der Datenbank beeinflussen.
- Wenn Sie über eine weitere Festplatte oder einen Tablespace mit mehr verfügbarem Speicherplatz verfügen, können Sie den temporären Speicherort für das Neuerstellen des Index ändern. Sie können den PostgreSQL-Parameter `temp_tablespaces` festlegen, um einen anderen Tablespace für temporäre Objekte anzugeben.
- Erwägen Sie, Speicherplatz freizugeben, indem Sie möglicherweise unnötige Daten löschen.
- Erwägen Sie, den Parameter *Data storage period* housekeeper vor der Ausführung des housekeeper zu verringern.

Ändern des Primärschlüssels mit Ausfallzeit

1. Entfernen Sie den aktuellen Primärschlüssel der Tabelle `auditlog` und fügen Sie den neuen Primärschlüssel hinzu.

```
ALTER TABLE auditlog DROP CONSTRAINT auditlog_pkey;
ALTER TABLE auditlog ADD PRIMARY KEY (auditid,clock);
```

2. Indizes neu aufbauen (optional, aber dringend empfohlen, siehe Wichtige Hinweise zum Neuerstellen von Indizes).

```
REINDEX TABLE CONCURRENTLY auditlog;
```

Ändern des Primärschlüssels ohne Ausfallzeit

Hier wird die manuelle Methode zum Ändern des Primärschlüssels beschrieben. Alternativ kann die Erweiterung `pg_repack` in Betracht gezogen werden, um eine neue Tabelle zu erstellen, Daten zu kopieren und Tabellen zu tauschen.

1. Erstellen Sie eine neue Tabelle mit dem neuen Primärschlüssel und legen Sie Indizes an.

```
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. Tauschen Sie die Tabellen aus.

```
ALTER TABLE auditlog RENAME TO auditlog_old;
ALTER TABLE auditlog_new RENAME TO auditlog;
```

3. Kopieren Sie die Daten von der alten Tabelle in die neue Tabelle.

```
INSERT INTO auditlog SELECT * FROM auditlog_old;
```

Dies kann in Blöcken erfolgen (mehrere `INSERT INTO`-Anweisungen mit `WHERE clock`-Klauseln nach Bedarf), um eine übermäßige Ressourcennutzung zu vermeiden.

4. Löschen Sie die alte Tabelle.

```
DROP TABLE auditlog_old;
```

Siehe auch

- Datenbank-Upgrade auf Primärschlüssel

TimescaleDB-Einrichtung

Übersicht

Zabbix unterstützt TimescaleDB, eine auf PostgreSQL basierende Datenbanklösung, die Daten automatisch in zeitbasierte Chunks partitioniert, um bei Skalierung eine schnellere Leistung zu ermöglichen.

> **Warning:**
> Derzeit wird TimescaleDB von Zabbix Proxy nicht unterstützt.

Die Anweisungen auf dieser Seite können für die folgenden Szenarien verwendet werden:

- Erstellen einer TimescaleDB-Datenbank oder Migration von bestehenden PostgreSQL-Tabellen zu TimescaleDB (siehe Konfiguration).
- Aktualisierung eines bestehenden TimescaleDB-Datenbankschemas beim Upgrade von Zabbix (siehe Upgrade des TimescaleDB-Schemas).

Konfiguration

Voraussetzungen: TimescaleDB-Erweiterung einer unterstützten Version, installiert auf dem Datenbank-Server. Installationsanweisungen finden Sie in der TimescaleDB-Dokumentation.

> **Warning:**
> Bevor Sie TimescaleDB installieren, installieren Sie eine unterstützte PostgreSQL-Version aus dem Repository von official PostgreSQL.

Aktivieren Sie die TimescaleDB-Erweiterung für die jeweilige Datenbank durch Ausführen von:

```
echo "CREATE EXTENSION IF NOT EXISTS timescaledb CASCADE;" | sudo -u postgres psql zabbix
```

Zum Ausführen dieses Befehls sind Datenbankadministratorrechte erforderlich.

> **Note:**
> Wenn Sie ein anderes Datenbankschema als 'public' verwenden, müssen Sie dem obigen Befehl eine SCHEMA-Klausel hinzufügen. Z. B.:
 `echo "CREATE EXTENSION IF NOT EXISTS timescaledb SCHEMA yourschema CASCADE;" | sudo -u postgres psql zabbix`

Führen Sie dann das Skript `postgresql/timescaledb/schema.sql` aus. Bei Neuinstallationen muss das Skript ausgeführt werden, nachdem die reguläre PostgreSQL-Datenbank mit dem initialen Schema/den initialen Daten erstellt wurde (siehe Erstellung der Datenbank).

```
cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/schema.sql | sudo -u zabbix psql zabbix
```

> **Attention:**
> Bitte ignorieren Sie Warnmeldungen, die darauf hinweisen, dass die Best Practices beim Ausführen des Skripts `schema.sql` unter TimescaleDB Version 2.9.0 und höher nicht eingehalten werden. Ungeachtet dieser Warnung wird die Konfiguration erfolgreich abgeschlossen.

Die Migration vorhandener Verlaufs-, Trend- und Audit-Log-Daten kann sehr viel Zeit in Anspruch nehmen. Zabbix Server und Frontend müssen für die Dauer der Migration heruntergefahren sein.

Das Skript `schema.sql` setzt die folgenden Housekeeping-Parameter:

- Überschreiben des Verlaufszeitraums von Datenpunkten
- Überschreiben des Trendzeitraums von Datenpunkten

Um partitioniertes Housekeeping für Verlauf und Trends zu verwenden, müssen beide Optionen aktiviert sein. Es ist auch möglich, die Überschreibung einzeln zu aktivieren, entweder nur für den Verlauf oder nur für Trends.

Für PostgreSQL und TimescaleDB setzt das Skript `postgresql/timescaledb/schema.sql` zwei zusätzliche Parameter:

- Komprimierung aktivieren
- Datensätze komprimieren, die älter als 7 Tage sind

Um komprimierte Daten erfolgreich durch den Housekeeper zu entfernen, müssen sowohl die Optionen *Verlaufsspeicherzeitraum für Datenpunkte überschreiben* als auch *Trend-Speicherzeitraum für Datenpunkte überschreiben* aktiviert sein. Wenn die Überschreibung deaktiviert ist und Tabellen komprimierte Chunks enthalten, entfernt der Housekeeper keine Daten aus diesen Tabellen, und in den Abschnitten *Housekeeping* und *System information* werden Warnungen über eine fehlerhafte Konfiguration angezeigt.

Alle diese Parameter können nach der Installation unter *Administration* > *Housekeeping* geändert werden.

Note:
Möglicherweise möchten Sie das von TimescaleDB bereitgestellte Tool timescaledb-tune ausführen, um die PostgreSQL-Konfigurationsparameter in Ihrer `postgresql.conf` zu optimieren.

Upgrade des TimescaleDB-Schemas

Beim Upgrade von Zabbix auf eine Version, die neue TimescaleDB-Hypertabellen enthält, konfiguriert der Zabbix Server diese Hypertabellen nicht automatisch (zum Beispiel beim Upgrade von Zabbix 6.4 auf 8.0, da in den Versionen 7.0.0 und 7.0.2 neue Hypertabellen eingeführt wurden).

Gehen Sie wie folgt vor, um neue TimescaleDB-Hypertabellen zu konfigurieren:

1. Starten Sie den Zabbix Server; dadurch wird die bestehende Datenbank aktualisiert.
2. Prüfen Sie in der Server-Logdatei, dass das Datenbank-Upgrade abgeschlossen ist; sobald es abgeschlossen ist, stoppen Sie den Zabbix Server. Beachten Sie, dass der Server eine Warnung protokolliert, wenn er versucht, die Komprimierung für eine Tabelle zu aktivieren, die keine Hypertabelle ist.
3. Führen Sie das Skript `postgresql/timescaledb/schema.sql` aus; dadurch werden die neuen TimescaleDB-Hypertabellen konfiguriert. Beachten Sie, dass sich seit Zabbix 7.0.0 Speicherort und Name des Skripts von `postgresql/timescaledb` zu `postgresql/timescaledb/schema.sql` geändert haben.

Attention:
Bitte ignorieren Sie Warnmeldungen, die darauf hinweisen, dass beim Ausführen des Skripts `schema.sql` unter TimescaleDB Version 2.9.0 und höher die Best Practices nicht eingehalten werden. Ungeachtet dieser Warnung wird die Konfiguration erfolgreich abgeschlossen.

TimescaleDB-Komprimierung

Die native TimescaleDB-Komprimierung wird für alle Zabbix-Tabellen unterstützt, die TimescaleDB-Hypertables sind. Während des Upgrades oder der Migration zu TimescaleDB kann die anfängliche Komprimierung großer Tabellen sehr viel Zeit in Anspruch nehmen.

Beachten Sie, dass die Komprimierung unter der Timescale Community-Lizenz ”timescale” unterstützt wird und unter der Apache-2.0-Lizenz ”apache” nicht unterstützt wird. Wenn Zabbix erkennt, dass die Komprimierung nicht unterstützt wird, wird eine Warnmeldung in das Zabbix-Server-Log geschrieben, und Benutzer können die Komprimierung im Frontend nicht aktivieren.

Note:
Benutzern wird empfohlen, sich vor der Verwendung der Komprimierung mit der TimescaleDB-Dokumentation vertraut zu machen.

Beachten Sie, dass die Komprimierung bestimmte Einschränkungen mit sich bringt, insbesondere:

- Änderungen an komprimierten Chunks (Einfügungen, Löschungen, Aktualisierungen) sind nicht zulässig
- Schemaänderungen für komprimierte Tabellen sind nicht zulässig.

Die Komprimierungseinstellungen können im Block *Verlauf, Trends und Audit-Log-Komprimierung* im Abschnitt *Administration* > *Housekeeping* des Zabbix-Frontends geändert werden.

Parameter	Standard	Kommentare
Komprimierung aktivieren	Aktiviert	Das Aktivieren oder Deaktivieren des Kontrollkästchens aktiviert/deaktiviert die Komprimierung nicht sofort. Da die Komprimierung vom Housekeeper verarbeitet wird, werden die Änderungen innerhalb von bis zu 2 mal `HousekeepingFrequency` Stunden wirksam (festgelegt in `zabbix_server.conf`) Nach dem Deaktivieren der Komprimierung werden neue Chunks, die in den Komprimierungszeitraum fallen, nicht komprimiert. Bereits komprimierte Daten bleiben jedoch komprimiert. Um zuvor komprimierte Chunks zu dekomprimieren, folgen Sie den Anweisungen in der TimescaleDB-Dokumentation. Beim Upgrade von älteren Zabbix-Versionen mit TimescaleDB-Unterstützung ist die Komprimierung standardmäßig nicht aktiviert.
Datensätze komprimieren, die älter sind als	7d	Dieser Parameter darf nicht kleiner als 7 Tage sein. Aufgrund der Unveränderlichkeit komprimierter Chunks werden alle verspätet eintreffenden Daten (z. B. durch einen Proxy verzögerte Daten), die älter als dieser Wert sind, verworfen.

Für eine bessere Leistung bei der Aktualisierung von Trends sollten Sie den Wert `chunk_time_interval` für die Tabellen `trends` und `trends_uint` von 30 Tagen auf 7 Tage oder weniger verringern, abhängig davon, wie viele Datenpunkte Trends verwenden. Der Zweck dieser Einstellung besteht darin, den Best Practices von TimescaleDB zu entsprechen und sicherzustellen, dass die Chunk-Größe innerhalb der verfügbaren Systemressourcen bleibt.

Elasticsearch-Einrichtung

Zabbix kann Verlaufsdaten in Elasticsearch als Alternative zu einer relationalen Datenbank speichern.

> **Attention:**
> Die Unterstützung von Elasticsearch ist derzeit experimentell.

Diese Anleitung beschreibt die Einrichtung der unterstützten Versionen von Elasticsearch. Wenn Sie eine andere Version verwenden, funktioniert möglicherweise nicht alles wie vorgesehen.

Elasticsearch kann die folgenden Werttypen speichern:

Item value type	Database table	Elasticsearch type
Numeric (unsigned)	history_uint	uint
Numeric (float)	history	dbl
Character	history_str	str
Log	history_log	log
Text	history_text	text
Binary	history_bin	not supported by Zabbix
JSON	history_json	json

> **Note:**
> Elasticsearch akzeptiert keine JSON-Arrays. Ein JSON-Wert muss entweder ein einzelnes Objekt oder eine Menge von Objekten sein.

Wichtige Hinweise

- Elasticsearch erfordert libcurl. Weitere Informationen finden Sie unter Anforderungen.
- Der Housekeeper löscht keine Daten aus Elasticsearch. Um zu steuern, wie lange Daten aufbewahrt werden, siehe Index Lifecycle Management.
- Zabbix berechnet oder speichert Trends in Elasticsearch **nicht**. Erwägen Sie, den Speicherzeitraum für Historien zu verlängern, um ältere Daten zu erhalten.

- Wenn Elasticsearch verwendet wird, sind Bereichsabfragen, die Werte aus der Datenbank abrufen, durch den Zeitstempel des Datenspeicherzeitraums begrenzt.
- Elasticsearch wird für Zabbix Proxies nicht unterstützt.

Wenn Elasticsearch noch nicht installiert ist, lesen Sie vor dem Fortfahren die offizielle Installationsanleitung.

Elasticsearch konfigurieren

Um Verlaufsdaten in Elasticsearch zu speichern, müssen Sie:

- Für jeden Werttyp, den Sie speichern möchten, einen Index erstellen – dort speichert Elasticsearch die Daten, ähnlich wie in einer Tabelle in einer relationalen Datenbank.
- Für jeden Index ein Mapping definieren – dieses legt die Struktur der Daten fest, ähnlich wie ein Tabellenschema.
- Eine Ingest-Pipeline einrichten, um Werte vor der Speicherung zu verarbeiten (erforderlich für JSON-Werte und datumsbasierte Indizes).

Elasticsearch kann Daten in einem einzelnen Index pro Werttyp oder über mehrere datumsbasierte Indizes hinweg speichern. Beide Ansätze werden unten beschrieben.

Verlauf in einem einzelnen Index speichern

Bei diesem Ansatz werden alle Verlaufsdaten für einen bestimmten Werttyp in einen einzelnen Index geschrieben (z. B. `uint` oder `text`).

Um einen Index für den Werttyp *Numerisch (vorzeichenlos)* zu erstellen, senden Sie die folgende Anfrage (mit `/uint` in der URL) an Ihre Elasticsearch-Instanz:

```
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 antwortet mit einer Bestätigung, dass der Index erstellt wurde:

```
{"acknowledged": true, "shards_acknowledged": true, "index": "uint"}
```

Ähnliche Anfragen müssen für jeden zusätzlichen Werttyp gesendet werden, den Sie in Elasticsearch speichern möchten.

> **Note:**
> Mappings für alle Werttypen sind im Zabbix source repository verfügbar.

Zum Beispiel, um einen Index für den Werttyp *Text* zu erstellen:

```
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"
         }
      }
   }
}'
```

JSON-Werttyp

Im Gegensatz zu anderen Werttypen erfordern JSON-Werte vor der Speicherung eine zusätzliche Verarbeitung.

Der unten stehende Index verwendet separate Felder für geparste und rohe Werte, daher wird eine Ingest-Pipeline benötigt, um jeden Wert als JSON zu parsen und im richtigen Feld zu speichern.

Um einen Index für den Werttyp *JSON* zu erstellen, senden Sie die folgende Anfrage (mit `/json` in der URL) an Ihre Elasticsearch-Instanz.

```
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 }
    }
  }
}'
```

Erstellen Sie dann die Ingest-Pipeline:

```
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 antwortet mit einer Bestätigung, dass die Ingest-Pipeline erstellt wurde:

```
{"acknowledged": true}
```

Speicherung von History in datumsbasierten Indizes

Anstatt alle History-Daten in einen einzelnen Index zu schreiben (z. B. `uint`), kann Elasticsearch diese Daten auf mehrere datumsbasierte Indizes verteilen (z. B. `uint-2026-01-01`, `uint-2026-01-02`). Dadurch lassen sich Datenvolumen und Aufbewahrung im Laufe der Zeit einfacher verwalten.

Um dies zu aktivieren, müssen Sie:

- Für jeden Werttyp, den Sie speichern möchten, eine Indexvorlage erstellen - diese teilt Elasticsearch mit, welches Mapping angewendet werden soll, wenn automatisch ein neuer datumsbasierter Index erstellt wird.
- Für jeden Werttyp eine Ingest-Pipeline erstellen - sie verarbeitet jeden eingehenden Wert und leitet ihn an den richtigen datumsbasierten Index weiter.

Index-Vorlagen

Um eine Vorlage für den `text`-Index zu erstellen, senden Sie eine Anfrage mit den folgenden Details:

- Verwenden Sie `_template/text_template` in der URL Ihrer Elasticsearch-Instanz.
- Verwenden Sie `"text*"` im Feld `"index_patterns"`, um den Index-Namen abzugleichen.
- Verwenden Sie ein Mapping für den Werttyp `text` (siehe Mappings im Zabbix-Quellcode-Repository).

```
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"
         }
      }
   }
}'
```

Vorlage für den `json`-Index:

```
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 }
      }
   }
}'
```

Ingest-Pipelines

Um eine Ingest-Pipeline für den Index `text` zu erstellen:

- Verwenden Sie `_ingest/pipeline/text-pipeline` in der URL Ihrer Elasticsearch-Instanz.
- Fügen Sie einen Prozessor `date_index_name` hinzu, um jeden Wert anhand seines Zeitstempels an den korrekten datumsbasierten Index weiterzuleiten.

```
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"
         }
      }
   ]
}'
```

Für den Index `json` muss die Pipeline den JSON-Wert vor der Weiterleitung an den korrekten Index außerdem parsen:

```
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"
         }
      }
   ]
}'
```

Konfigurieren des Zabbix-Servers

Setzen Sie in der Konfigurationsdatei Ihres Zabbix-Servers (`zabbix_server.conf`) den Parameter `HistoryProvider`.

Um beispielsweise Werte der Typen *Character*, *Log*, *Text* und *JSON* in Elasticsearch zu speichern (während *Numeric*-Werte in einer Datenbank verbleiben):

```
HistoryProvider=elasticsearch;value_types="str,log,text,json",url=http://localhost:9200
```

Wenn Sie datumsbasierte Indizes verwenden, fügen Sie dem Parameter `date_index=1` hinzu:

```
HistoryProvider=elasticsearch;value_types="str,log,text,json",url=http://localhost:9200,date_index=1
```

Starten Sie nach den Änderungen den Zabbix-Server neu:

```
systemctl restart zabbix-server
```

Konfigurieren des Zabbix Frontends

Legen Sie in Ihrer Frontend-Konfigurationsdatei von Zabbix (`zabbix.conf.php`) die Variable `$HISTORY_PROVIDERS` entsprechend der Server-Konfiguration fest:

```
$HISTORY_PROVIDERS[] = [
  'types' => ['str','log','text','json'],
  'provider' => 'elasticsearch',
  'url' => 'http://localhost:9200'
];
```

Fehlerbehebung

Die folgenden Schritte können Ihnen bei der Fehlerbehebung Ihrer Elasticsearch-Konfiguration helfen:

1. Prüfen Sie die Protokolle von Elasticsearch oder dem Zabbix-Server auf Fehler.
2. Um langsame Abfragen zu identifizieren, verwenden Sie die Option `log_slow_queries` im Zabbix-Server-Konfigurationsparameter `HistoryProvider`.
3. Vergewissern Sie sich, dass Elasticsearch den Zugriff vom Zabbix-Server und vom Zabbix-Frontend erlaubt.
4. Vergewissern Sie sich, dass `auto_create_index` aktiviert ist:

```
curl -X GET \
 "http://localhost:9200/_cluster/settings?include_defaults=true&filter_path=**.auto_create_index"

#### {"defaults": {"action": {"auto_create_index": "false"} } }
```

Um es zu aktivieren, senden Sie die folgende Anfrage:

```
curl -X PUT \
 http://localhost:9200/_cluster/settings \
 -H 'content-type:application/json' \
 -d '{
   "persistent": {
      "action.auto_create_index": "true"
   }
}'

#### {"acknowledged": true, "persistent": {"action": {"auto_create_index": "true"} }, "transient": {} }
```

5. Vergewissern Sie sich, dass Mappings, Vorlagen und Ingest-Pipelines korrekt sind, indem Sie GET-Anfragen an die jeweiligen URLs senden:

```
curl -X GET http://localhost:9200/json
curl -X GET http://localhost:9200/_template/json*
curl -X GET http://localhost:9200/_ingest/pipeline/json*
```

Sie können die empfangenen Antworten mit den erwarteten Antworten in der Elasticsearch-API-Dokumentation vergleichen.

6. Prüfen Sie, ob sich Shards in einem fehlgeschlagenen Zustand befinden; ein Neustart von Elasticsearch kann das Problem möglicherweise beheben.

7. Fragen Sie Elasticsearch ab, um zu prüfen, ob die von Zabbix gesammelten Daten gespeichert wurden, zum Beispiel:

```
curl 'http://localhost:9200/json/_search' \
 -H 'Content-Type: application/json' \
 -d '{
   "query": {
     "term": {
       "itemid": 42269
     }
   }
 }'
```

8. Wenn Sie Ihre Elasticsearch-Konfiguration zurücksetzen und neu beginnen müssen, können Sie alle Indizes, Vorlagen und Ingest-Pipelines löschen:

```
curl -X DELETE "http://localhost:9200/_all"
curl -X DELETE "http://localhost:9200/_template/*"
curl -X DELETE "http://localhost:9200/_ingest/pipeline/*"
```

ClickHouse-Einrichtung

Zabbix kann Verlaufsdaten als Alternative zu einer relationalen Datenbank in ClickHouse speichern.

> **Note:**
> ClickHouse kann auch als Telemetriedatenquelle für das Application Performance Monitoring (APM) verwendet werden. Weitere Informationen finden Sie unter *Administration* > *Datenquelle*.

Diese Anleitung behandelt die Einrichtung der unterstützten Versionen von ClickHouse. Wenn Sie eine andere Version verwenden, funktionieren einige Funktionen möglicherweise nicht wie vorgesehen.

ClickHouse kann die folgenden Werttypen speichern:

Datenpunktwerttyp	Datenbanktabelle	ClickHouse-Typ
Numerisch (vorzeichenlos)	history_uint	uint
Numerisch (Gleitkomma)	history	dbl
Zeichen	history_str	str
Protokoll	history_log	log
Text	history_text	text
Binär	history_bin	von Zabbix nicht unterstützt
JSON	history_json	json

> **Note:**
> ClickHouse akzeptiert keine JSON-Arrays. Ein JSON-Wert muss entweder aus einem einzelnen Objekt oder aus einer Gruppe von Objekten bestehen. Außerdem behandelt ClickHouse JSON-Schlüssel mit NULL genauso wie fehlende Schlüssel.

Wichtige Hinweise

- Der Housekeeper löscht keine Daten aus ClickHouse. Um zu steuern, wie lange Daten aufbewahrt werden, konfigurieren Sie den Aufbewahrungszeitraum für ClickHouse-Daten in ClickHouse.
- Zabbix **berechnet oder speichert keine Trends** in ClickHouse. Erwägen Sie, den Aufbewahrungszeitraum für den Verlauf zu verlängern, um ältere Daten aufzubewahren.
- Wenn Sie Ihre Verlaufsdaten aus einer vorhandenen Zabbix-Datenbank (MySQL oder PostgreSQL) nach ClickHouse migrieren möchten, finden Sie weitere Informationen im ClickHouse-Schema und in den Migrationsskripten für den Verlauf.
- ClickHouse wird nicht als Verlaufsspeicher für Zabbix-Proxys unterstützt, kann jedoch als Telemetriedatenquelle für einen Zabbix-Proxy mit aktiviertem APM verwendet werden.
- Wenn Sie über einen ClickHouse-Cluster verfügen, lesen Sie Konfigurieren von ClickHouse-Clustern.

Konfigurieren von ClickHouse

Sie müssen eine Zabbix-Datenbank und einen Benutzer erstellen und konfigurieren sowie das Datenbankschema importieren.

Diese Anleitung enthält Anweisungen für Docker- oder Paket-Installationen von ClickHouse.

Docker

1. Erstellen und konfigurieren Sie die Zabbix-Datenbank und den Benutzer beim Ausführen des ClickHouse-Containers:

```
sudo docker run -d \
  --name clickhouse \
  -e CLICKHOUSE_DB=zabbix \
  -e CLICKHOUSE_USER=zabbix \
  -e CLICKHOUSE_DEFAULT_ACCESS_MANAGEMENT=1 \
  -e CLICKHOUSE_PASSWORD=<password> \
  -p 8123:8123/tcp \
  -p 9000:9000/tcp \
  --cap-add=SYS_NICE \
  --cap-add=NET_ADMIN \
  --cap-add=IPC_LOCK \
  --ulimit nofile=262144:262144 \
  -v clickhouse_data:/var/lib/clickhouse \
  clickhouse/clickhouse-server:26.4
```

2. Bestätigen Sie, dass ClickHouse funktioniert und Sie eine Verbindung herstellen können:

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --query "SELECT version()"

#### 26.4.4.38
```

3. Importieren Sie das Datenbankschema mit dem Skript `history_all.sh` aus Ihrem Zabbix-Verzeichnis:

```
./usr/share/zabbix/sql-scripts/clickhouse/history_all.sh \
  --user zabbix \
  --password <password> \
  --db zabbix \
  --server http://localhost:8123
```

> **Note:**
> Der Datenspeicherzeitraum (Time-To-Live, oder TTL) für ClickHouse beträgt standardmäßig 31 Tage. Um ihn zu ändern, verwenden Sie die Option `--ttl` beim Importieren des Datenbankschemas (z. B. `--ttl 604800` für 7 Tage), oder konfigurieren Sie ihn später.

4. Überprüfen Sie, ob die Tabellen erstellt wurden:

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "SHOW TABLES FROM zabbix"

#### history
#### history_json
#### history_log
#### history_str
#### history_text
#### history_uint
```

Pakete

1. Starten Sie ClickHouse:

```
sudo service clickhouse-server start
```

2. Erstellen und konfigurieren Sie die Zabbix-Datenbank und den Benutzer:

```
clickhouse-client --user default
```

```
localhost :) CREATE DATABASE IF NOT EXISTS zabbix
localhost :) CREATE USER IF NOT EXISTS zabbix IDENTIFIED WITH sha256_password BY '<password>'
localhost :) GRANT CREATE, ALTER, DROP, INSERT, SELECT, UPDATE, OPTIMIZE ON zabbix.* TO zabbix
localhost :) quit
```

3. Importieren Sie das Datenbankschema mit dem Skript `history_all.sh` aus Ihrem Zabbix-Verzeichnis:

```
./usr/share/zabbix/sql-scripts/clickhouse/history_all.sh \
  --user zabbix \
  --password <password> \
  --db zabbix \
  --server http://localhost:8123
```

Note:
Der Datenspeicherzeitraum (Time-To-Live, oder TTL) für ClickHouse beträgt standardmäßig 31 Tage. Um ihn zu ändern, verwenden Sie die Option `--ttl` beim Importieren des Datenbankschemas (z. B. `--ttl 604800` für 7 Tage), oder konfigurieren Sie ihn später.

4. Überprüfen Sie, ob die Tabellen erstellt wurden:

```
clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "SHOW TABLES FROM zabbix"

#### history
#### history_json
#### history_log
#### history_str
#### history_text
#### history_uint
```

Konfigurieren des Zabbix Server

Setzen Sie in Ihrer Zabbix-Server-Konfigurationsdatei (`zabbix_server.conf`) den Parameter `HistoryProvider`.

Um beispielsweise alle unterstützten Werttypen in ClickHouse zu speichern:

```
HistoryProvider=clickhouse;value_types="uint,dbl,str,log,text,json",url=http://localhost:8123,db=zabbix,us
```

Starten Sie nach den Änderungen den Zabbix Server neu:

```
systemctl restart zabbix-server
```

Zabbix-Frontend konfigurieren

Legen Sie in Ihrer Frontend-Konfigurationsdatei von Zabbix (`zabbix.conf.php`) die Variable `$HISTORY_PROVIDERS` entsprechend der Server-Konfiguration fest:

```
$HISTORY_PROVIDERS[] = [
  'types' => ['uint','dbl','str','log','text','json'],
  'provider' => 'clickhouse',
  'url' => 'http://localhost:8123',
  'db' => 'zabbix',
  'username' => 'zabbix',
  'password' => '<password>'
];
```

Zusätzliche Konfiguration

Die folgenden Schritte sind optional. Sie benötigen sie nicht für eine grundlegende Einrichtung.

Konfigurieren des ClickHouse-Datenspeicherzeitraums

Der Datenspeicherzeitraum (Time-To-Live, oder TTL) für ClickHouse beträgt standardmäßig 31 Tage. Um ihn zu ändern, führen Sie die folgenden Befehle aus.

Die folgenden Beispiele verwenden Docker. Wenn Sie ClickHouse mithilfe von Paketen installiert haben, führen Sie die Abfragen direkt im ClickHouse-Client aus.

1. Ändern Sie die Tabelle (ersetzen Sie `history_json` und 3600 durch die Werte, die Sie benötigen):

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "ALTER TABLE zabbix.history_json MODIFY TTL clock_ns + toIntervalSecond(3600)"
```

2. Wenden Sie die Änderungen sofort an:

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "OPTIMIZE TABLE zabbix.history_json MATERIALIZE TTL"
```

3. Überprüfen Sie, ob sich der Datenspeicherzeitraum geändert hat:

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "SHOW CREATE TABLE zabbix.history_json"

#### CREATE TABLE zabbix.history_json
#### (
####     `itemid` UInt64,
####     `clock_ns` DateTime64(9),
####     `value` JSON,
####     `value_str` String
#### )
#### ENGINE = MergeTree
#### PARTITION BY toDate(clock_ns)
#### PRIMARY KEY (itemid, clock_ns)
#### ORDER BY (itemid, clock_ns)
#### TTL clock_ns + toIntervalSecond(3600)
#### SETTINGS index_granularity = 8192
```

4. Starten Sie den Zabbix-Server neu, um den Datenspeicherzeitraum in *Administration* > *Housekeeping* zu aktualisieren:

```
systemctl restart zabbix-server
```

Konfigurieren von ClickHouse-Clustern

Dieser Abschnitt beschreibt die Konfigurationsschritte für ClickHouse-Cluster mit Replikation und Sharding.

Die folgenden Schritte verwenden Docker und das ClickHouse `cluster_2S_2R` aus der ClickHouse-Dokumentation (ClickHouse 26.4.4.38 mit ClickHouse Keeper).

Bevor Sie beginnen, wird empfohlen, einen Load Balancer vor Ihren ClickHouse-Cluster für Zabbix zu schalten. Die Konfiguration von Zabbix Server und Frontend unterstützt nur eine einzelne ClickHouse-URL, sodass sie nur auf einen Knoten verweisen kann. Wenn dieser Knoten ausfällt, leitet ein Load Balancer Zabbix stattdessen über dieselbe URL auf einen funktionsfähigen Knoten um.

1. Erstellen Sie die Zabbix-Datenbank:

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "CREATE DATABASE IF NOT EXISTS zabbix ON CLUSTER cluster_2S_2R ENGINE = Replicated('/clickhouse/
```

> **Attention:**
> Die `ON CLUSTER`-Klausel erreicht nur Knoten, die zum Zeitpunkt der Ausführung des Befehls Teil des Clusters sind. Wenn Sie Ihrem Cluster später Knoten hinzufügen, führen Sie diesen Befehl erneut aus.

2. Erstellen und konfigurieren Sie den Zabbix-Benutzer:

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
```

```
  --user default \
  --query "CREATE USER IF NOT EXISTS zabbix ON CLUSTER cluster_2S_2R IDENTIFIED WITH sha256_password BY '<

sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "GRANT ON CLUSTER cluster_2S_2R CREATE, ALTER, DROP, INSERT, SELECT, UPDATE, OPTIMIZE ON zabbix.
```

3. Importieren Sie das Datenbankschema mit dem Skript `history_all.sh` aus Ihrem Zabbix-Verzeichnis:

```
./usr/share/zabbix/sql-scripts/clickhouse/history_all.sh \
  --user zabbix \
  --password <password> \
  --db zabbix \
  --server http://localhost:8123 \
  --engine "ReplicatedMergeTree()"
```

Note:
Die Datenspeicherungsdauer (Time-To-Live, oder TTL) für ClickHouse beträgt standardmäßig 31 Tage. Um sie zu ändern, verwenden Sie beim Import des Datenbankschemas die Option `--ttl` (z. B. `--ttl 604800` für 7 Tage), oder konfigurieren Sie sie später.

Note:
Wenn Ihr ClickHouse-Cluster explizite Replikatpfade erfordert (z. B. wenn Ihre ClickHouse-Serverkonfiguration keine Standardpfade für replizierte Tabellen festlegt), führen Sie die Schema-Skripte (`history_schema.sh`, `history_uint_schema.sh` usw.) einzeln aus und legen Sie für jede Tabelle einen anderen Pfad fest:

```
./usr/share/zabbix/sql-scripts/clickhouse/history_schema.sh \
  --user zabbix \
  --password <password> \
  --db zabbix \
  --server http://localhost:8123 \
  --engine "ReplicatedMergeTree('/clickhouse/tables/{shard}/history', '{replica}')"
```

Weitere Informationen zur ClickHouse-Replikation finden Sie in der ClickHouse-Dokumentation unter Replicating data und Replicated* table engines.

4. Konfigurieren Sie das Sharding für jeden Tabellentyp. Benennen Sie die Tabellen um und erstellen Sie eine `Distributed`-Tabelle mit dem ursprünglichen Namen, die auf die umbenannte Tabelle verweist. Diese `Distributed`-Tabelle ist die Tabelle, aus der Zabbix tatsächlich liest und in die es schreibt; die Tabelle verteilt Daten automatisch auf die Shards. Zum Beispiel für die Tabelle `history`:

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "RENAME TABLE zabbix.history TO zabbix.history_local"

sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "CREATE TABLE zabbix.history AS zabbix.history_local ENGINE = Distributed('cluster_2S_2R', 'zabb
```

Note:
Weitere Informationen zum ClickHouse-Sharding finden Sie in der ClickHouse-Dokumentation unter Distributed table engine.

5. Konfigurieren Sie Zabbix Server.

6. Konfigurieren Sie das Zabbix Frontend.

7. Überprüfen Sie, ob die Zabbix-Daten über die Shards verteilt sind. Die Zeilenanzahl von `zabbix.history_local` kann sich zwischen wiederholten Prüfungen ändern, da Zabbix Server kontinuierlich neue Daten einfügt. Führen Sie diesen Befehl auf jedem Knoten aus:

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "SELECT count(*) FROM zabbix.history_local"

#### 2446 (clickhouse-01)
#### 2471 (clickhouse-02)
#### 2494 (clickhouse-03)
```

8. Prüfen Sie, ob die Replikation synchron ist (`active_replicas` sollte mit `total_replicas` übereinstimmen):

```
sudo docker exec -it clickhouse-01 \
  clickhouse-client \
  --user default \
  --query "SELECT database, table, is_leader, total_replicas, active_replicas FROM system.replicas WHERE d

#### zabbix  history_json_local  1 2 2
#### zabbix  history_local       1 2 2
#### zabbix  history_log_local   1 2 2
#### zabbix  history_str_local   1 2 2
#### zabbix  history_text_local  1 2 2
#### zabbix  history_uint_local  1 2 2
```

Fehlerbehebung

Die folgenden Schritte können Ihnen bei der Fehlerbehebung Ihrer ClickHouse-Einrichtung helfen:

1. Prüfen Sie die Protokolle von ClickHouse oder dem Zabbix Server auf Fehler.
2. Um langsame Abfragen zu identifizieren, verwenden Sie die Option `log_slow_queries` im Zabbix Server-Konfigurationsparameter `HistoryProvider`.
3. Vergewissern Sie sich, dass ClickHouse den Zugriff vom Zabbix Server und vom Zabbix Frontend aus zulässt.
4. Fragen Sie ClickHouse ab, um zu prüfen, ob die von Zabbix gesammelten Daten gespeichert werden, zum Beispiel:

```
sudo docker exec -it clickhouse \
  clickhouse-client \
  --user zabbix \
  --password <password> \
  --query "SELECT * FROM zabbix.history_uint WHERE itemid = 42269"
```

Distributionsspezifische Hinweise zur Einrichtung von NGINX für Zabbix

RHEL

Unter RHEL 8, 9 und 10 ist NGINX in AppStream enthalten – es sind keine zusätzlichen Repositories erforderlich. Aktivieren und installieren Sie NGINX einfach mit:

```
dnf module list nginx
dnf -y install nginx
```

Unter RHEL 7 kann NGINX aus dem CentOS-7-Extras-Repository installiert werden (bei Verwendung von CentOS):

```
yum -y install nginx
```

Wenn Sie die neuesten Upstream-Builds bevorzugen, konfigurieren Sie das offizielle NGINX-Repository unter `/etc/yum.repos.d/nginx.rep`

```
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

In SUSE Linux Enterprise Server 15 müssen Sie `php-fpm` konfigurieren (der Pfad zur Konfigurationsdatei kann je nach Service Pack leicht variieren):

```
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
```

SAML-Einrichtung mit Microsoft Entra ID

Übersicht

Dieser Abschnitt enthält Richtlinien für die Konfiguration von Single Sign-on und der Benutzerbereitstellung in Zabbix aus Microsoft Entra ID (früher Microsoft Azure Active Directory) mithilfe der SAML-2.0-Authentifizierung.

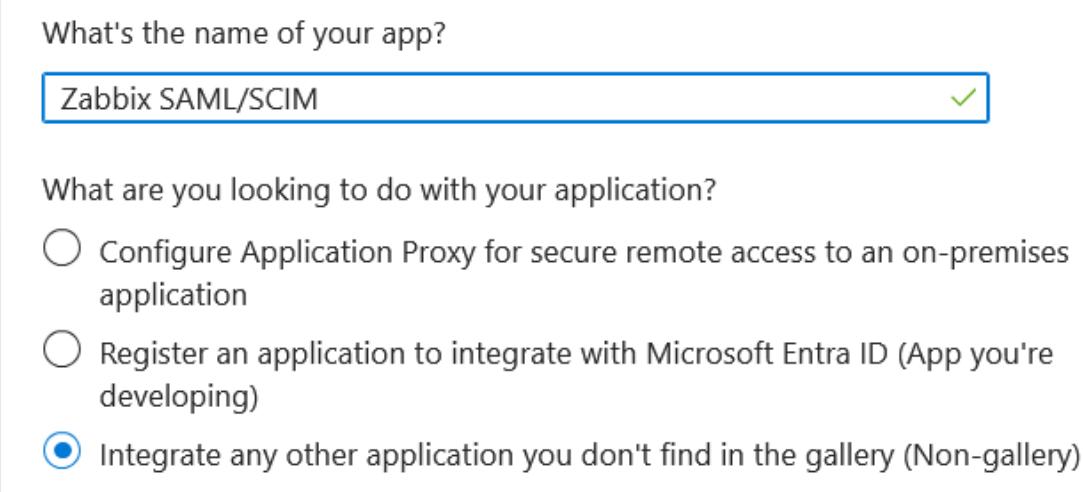
Microsoft Entra ID-Konfiguration

Erstellen einer Anwendung

1. Melden Sie sich im Microsoft Entra admin center bei Microsoft Entra ID an. Zu Testzwecken können Sie ein kostenloses Testkonto in Microsoft Entra ID erstellen.

2. Wählen Sie im Microsoft Entra admin center *Applications* > *Enterprise applications* > *New application* > *Create your own application* aus.

3. Fügen Sie den Namen Ihrer App hinzu und wählen Sie die Option *Integrate any other application...* aus. Klicken Sie anschließend auf *Create*.

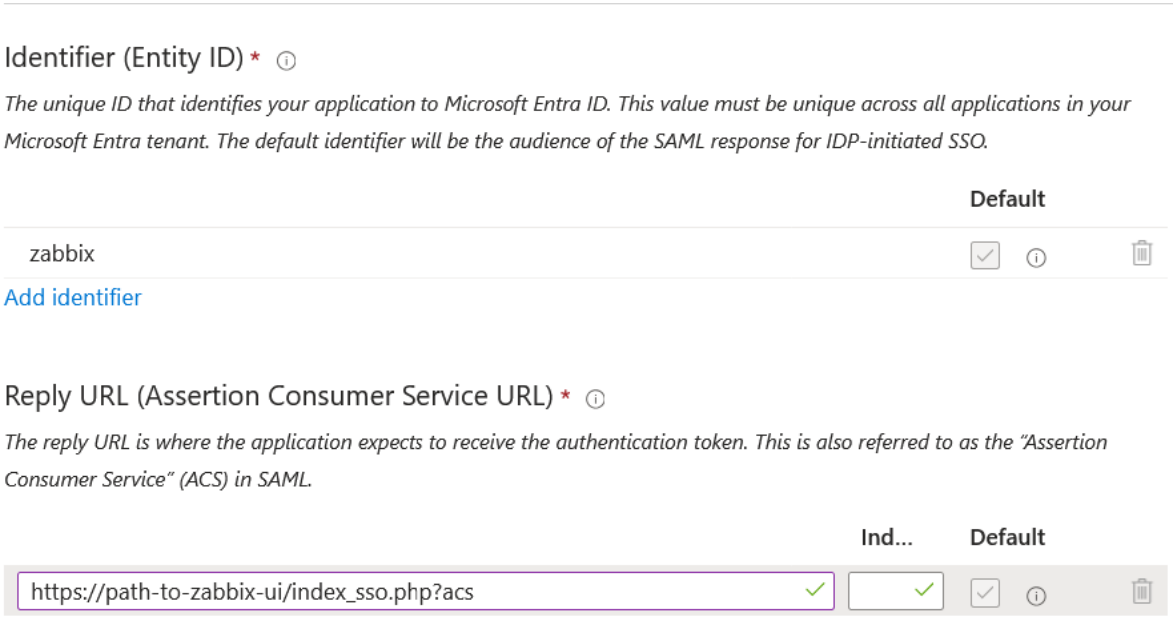


Einrichten des Single Sign-On

1. Gehen Sie auf der Anwendungsseite zu *Single Sign-On einrichten* und klicken Sie auf *Erste Schritte*. Wählen Sie anschließend *SAML* aus.

2. Bearbeiten Sie die *grundlegende SAML-Konfiguration*:

- Legen Sie unter *Bezeichner (Entitäts-ID)* einen eindeutigen Namen fest, anhand dessen Ihre App in Microsoft Entra ID identifiziert wird, zum Beispiel `zabbix`;
- Legen Sie unter *Antwort-URL (Assertion Consumer Service-URL)* den Single-Sign-On-Endpunkt von Zabbix fest: `https://<path-to-za`



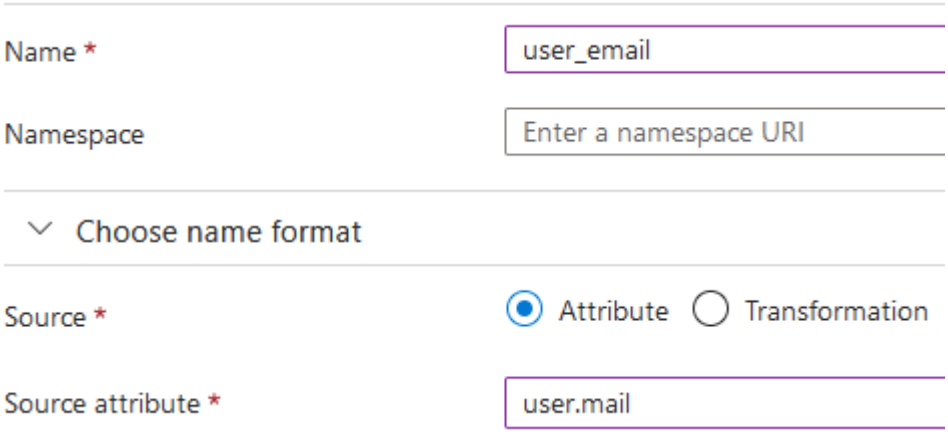
Beachten Sie, dass `https` erforderlich ist. Um dies mit Zabbix zu verwenden, fügen Sie der Frontend-Konfigurationsdatei die folgende Zeile hinzu:

```
$SSO['SETTINGS'] = ['use_proxy_headers' => true];
```

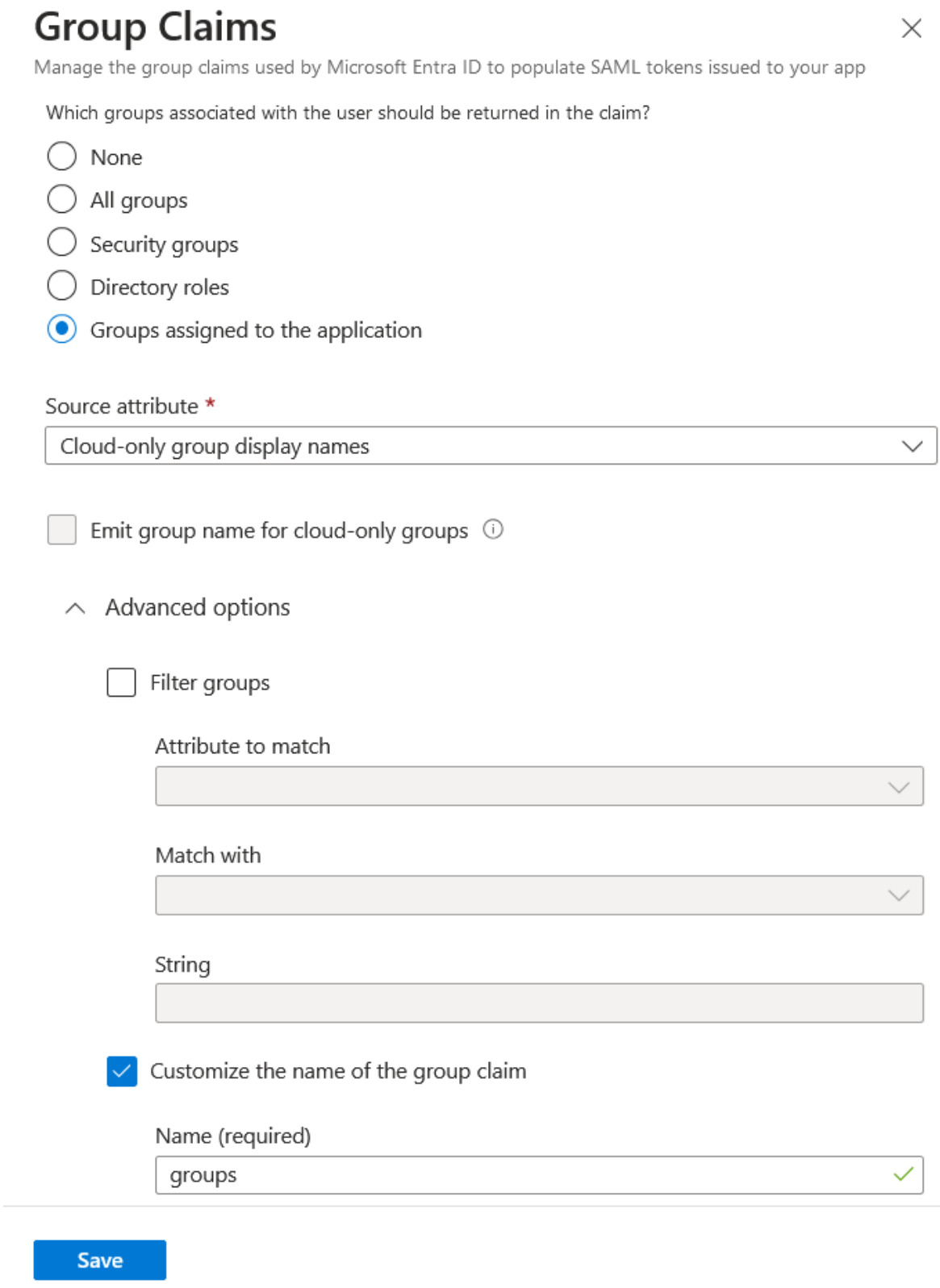
3. Bearbeiten Sie *Attribute und Ansprüche*. Sie müssen alle Attribute hinzufügen, die Sie an Zabbix übergeben möchten (user_name, user_lastname, user_email, user_mobile, groups).

Die Attributnamen sind frei wählbar. Es können andere Attributnamen verwendet werden. Sie müssen jedoch mit dem jeweiligen Feldwert in den Zabbix-SAML-Einstellungen übereinstimmen.

- Klicken Sie auf *Neuen Anspruch hinzufügen*, um ein Attribut hinzuzufügen:



- Klicken Sie auf *Gruppenanspruch hinzufügen*, um ein Attribut für die Übergabe von Gruppen an Zabbix hinzuzufügen:



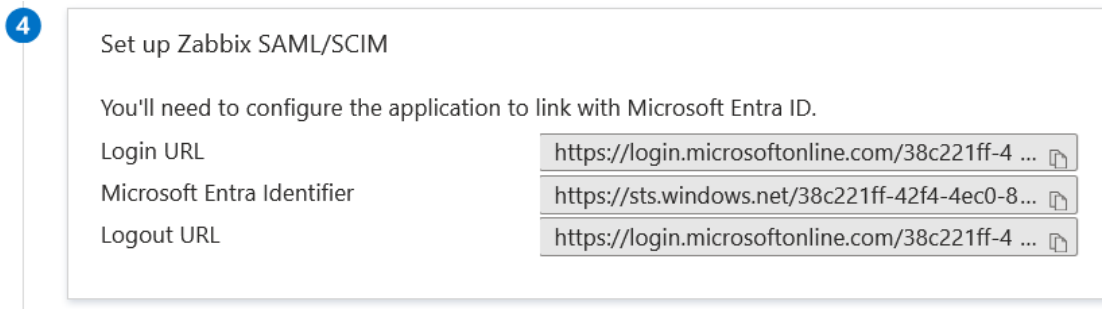
Bei diesem Anspruch ist es wichtig, dass die Gruppennamen (nicht die Gruppen-IDs) über das ausgewählte *Quellattribut* an Zabbix übergeben werden. Andernfalls funktioniert die JIT-Benutzerbereitstellung nicht ordnungsgemäß.

4. Laden Sie unter *SAML-Zertifikate* das von Entra ID bereitgestellte Base64-Zertifikat herunter und legen Sie es im Verzeichnis `conf/certs` der Zabbix-Frontend-Installation ab (der Speicherort kann je nach Betriebssystem und Quelle Ihrer Zabbix-Installation abweichen).

Legen Sie die Berechtigungen auf 644 fest, indem Sie folgenden Befehl ausführen:

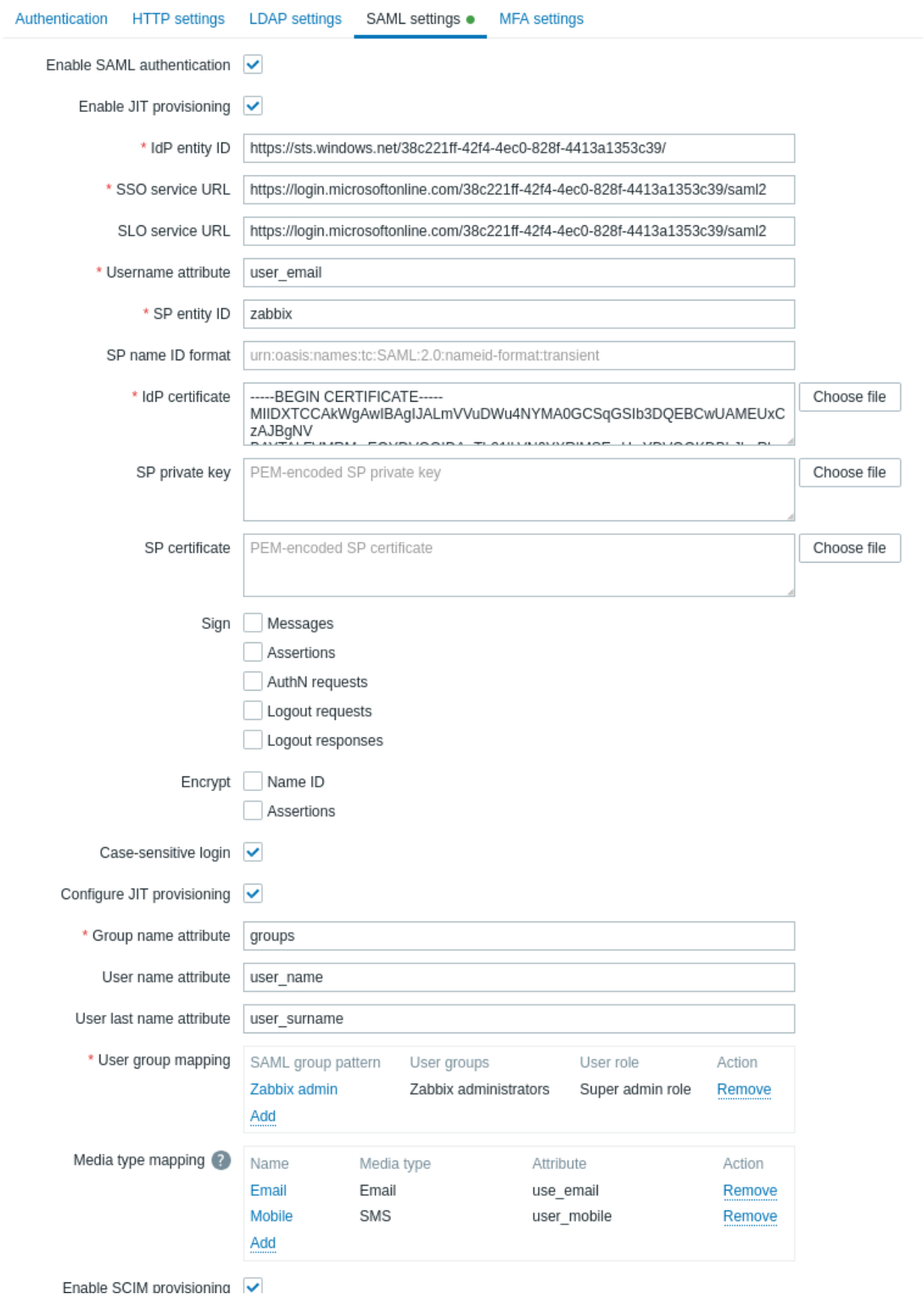
```
chmod 644 entra.cer
```

5. Verwenden Sie die Werte unter *<your app name> einrichten* in Entra ID, um die SAML-Authentifizierung von Zabbix zu konfigurieren (siehe nächsten Abschnitt):



Zabbix-Konfiguration

1. Rufen Sie in Zabbix die SAML-Einstellungen auf und füllen Sie die Konfigurationsoptionen anhand der Entra-ID-Konfiguration aus:



Zabbix-Feld	Setup-Feld in Entra ID	Beispielwert
IdP entity ID	Microsoft Entra identifier	
SSO service URL	Login URL	
SLO service URL	Logout URL	
SP entity ID	Identifier (Entity ID)	
Username attribute	Custom attribute (claim)	`user_email`
Group name attribute	Custom attribute (claim)	`groups`
User name attribute	Custom attribute (claim)	`user_name`

Zabbix-Feld	Setup-Feld in Entra ID	Beispielwert
User last name attribute	Custom attribute (claim)	`user_lastname`

2. Fügen Sie das von Entra ID bereitgestellte Base64-Zertifikat hinzu.

Wenn `$SSO['CERT_STORAGE'] = 'database'` in der Frontend-Konfigurationsdatei (`zabbix.conf.php`) festgelegt ist, können Sie den Zertifikatstext einfügen oder die Zertifikatsdatei während der SAML-Konfiguration im Frontend hochladen. Auf dem Dateisystem sind keine Dateien erforderlich.

Wenn `$SSO['CERT_STORAGE'] = 'file'` in der Konfigurationsdatei festgelegt ist, muss das Zertifikat im Dateisystem verfügbar gemacht werden (standardmäßig in `ui/conf/certs` oder unter dem in der Konfigurationsdatei festgelegten Pfad), und das Frontend speichert Zertifikate nicht in der Datenbank.
Beachten Sie, dass bei nicht gesetztem oder auskommentiertem `$SSO['CERT_STORAGE']` die Dateispeicherung verwendet wird und Zertifikate aus `ui/conf/certs` gelesen werden.

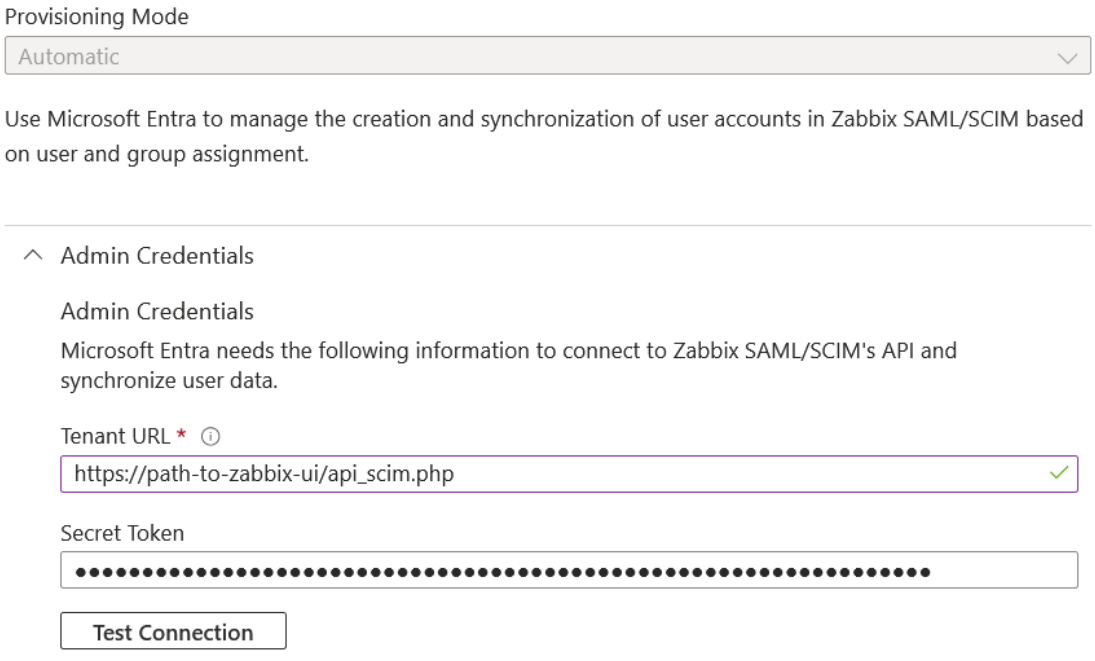
Außerdem muss die Zuordnung von Benutzergruppen konfiguriert werden.
Die Zuordnung von Medien ist optional.

3. Klicken Sie auf die Schaltfläche *Update*, um diese Einstellungen zu speichern.

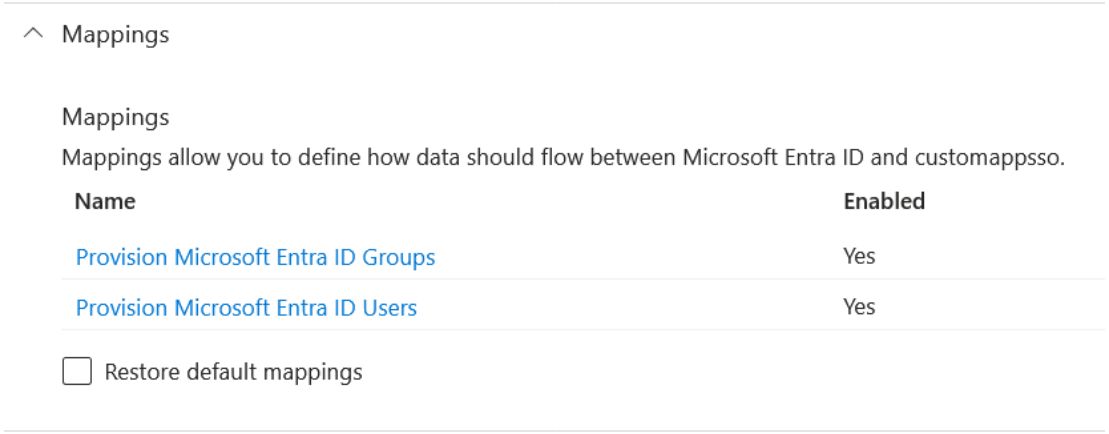
SCIM-Benutzerbereitstellung

1. Öffnen Sie auf der Anwendungsseite Ihrer Entra ID im Hauptmenü die Seite *Provisioning*. Klicken Sie auf *Get started* und wählen Sie dann den Modus *Automatic provisioning*:

- In *Tenant URL* setzen Sie den folgenden Wert: `https://<path-to-zabbix-ui>/api_scim.php`
- In *Secret token* geben Sie ein Zabbix-API-Token mit Super-Admin-Berechtigungen ein.
- Klicken Sie auf *Test connection*, um zu prüfen, ob die Verbindung hergestellt wurde.

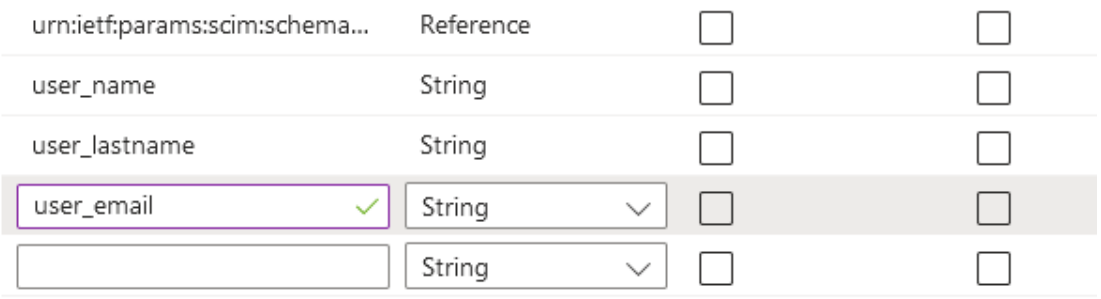


2. Nun können Sie alle Attribute hinzufügen, die per SCIM an Zabbix übergeben werden. Klicken Sie dazu auf *Mappings* und anschließend auf *Provision Microsoft Entra ID Users*.



Aktivieren Sie am unteren Ende der Attributzuordnungsliste *Show advanced options* und klicken Sie dann auf *Edit attribute list for customappsso*.

Fügen Sie am unteren Ende der Attributliste eigene Attribute mit dem Typ 'String' hinzu:



Speichern Sie die Liste.

3. Nun können Sie Zuordnungen für die hinzugefügten Attribute erstellen. Klicken Sie am unteren Ende der Attributzuordnungsliste auf *Add New Mapping* und erstellen Sie die Zuordnungen wie unten gezeigt:



Wenn alle Zuordnungen hinzugefügt wurden, speichern Sie die Liste der Zuordnungen.



4. Als Voraussetzung für die Benutzerbereitstellung in Zabbix müssen Benutzer und Gruppen in Entra ID konfiguriert sein.

Dazu gehen Sie zum *Microsoft Entra admin center* und fügen auf den jeweiligen Seiten *Users* und *Groups* Benutzer/Gruppen hinzu.

5. Wenn Benutzer und Gruppen in Entra ID erstellt wurden, können Sie im Menü *Users and groups* Ihrer Anwendung diese der App hinzufügen.

6. Gehen Sie zum Menü *Provisioning* Ihrer App und klicken Sie auf *Start provisioning*, um Benutzer an Zabbix bereitzustellen.

Beachten Sie, dass die Users-PATCH-Anforderung in Entra ID keine Änderungen an Medien unterstützt.

Signieren von Authentifizierungsanforderungen

Es ist möglich, Entra ID so zu konfigurieren, dass die Signatur signierter Authentifizierungsanforderungen überprüft wird.

Damit dies funktioniert, erstellen Sie öffentliche/private Schlüssel:

```
openssl req -x509 -newkey rsa:4096 -keyout /usr/share/zabbix/conf/certs/request-sign.key -out /usr/share/z
```

Weisen Sie Berechtigungen zu:

```
chown apache /usr/share/zabbix/conf/certs/request-sign.key
chmod 400 /usr/share/zabbix/conf/certs/request-sign.key
```

Aktualisieren Sie die Frontend-Konfiguration von Zabbix, indem Sie Folgendes hinzufügen:

```
$SSO['SP_KEY'] = 'conf/certs/request-sign.key';
$SSO['SP_CERT'] = 'conf/certs/request-sign.crt';
```

Fehlerbehebung

Bei Browsern von MicroSoft Edge können Authentifizierungsprobleme auftreten, wenn ein Benutzer versucht, sich über SAML bei Zabbix anzumelden, und bereits mit dem MicroSoft Edge-Profil angemeldet ist. Ein Anzeichen für ein solches Problem kann sein, dass sich der Benutzer mit MicroSoft Edge im privaten Modus bei Zabbix anmelden kann.

Um Authentifizierungsprobleme in diesem Fall zu vermeiden, kann es erforderlich sein, `requestedAuthnContext` in der Frontend-Konfigurationsdatei von Zabbix auf `false` zu setzen.

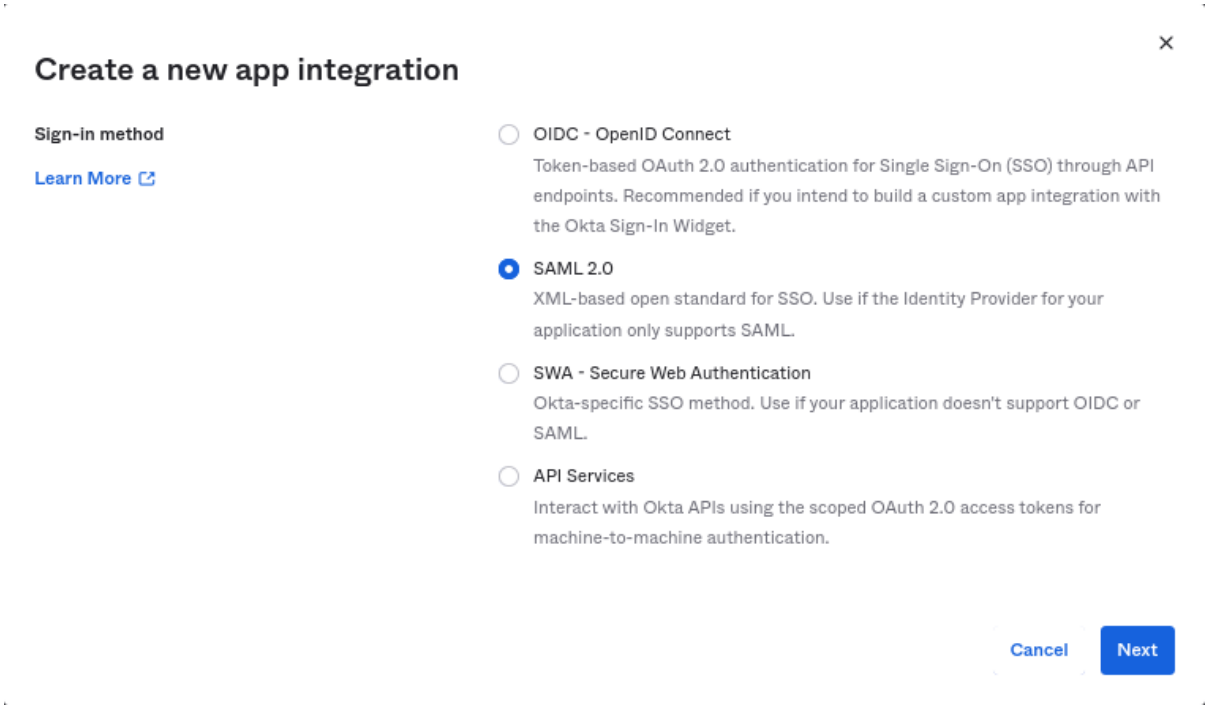
```
$SSO['SETTINGS'] = [
    'security' => [
        'requestedAuthnContext' => false
    ]
];
```

SAML-Einrichtung mit Okta

Dieser Abschnitt enthält Richtlinien zur Konfiguration von Okta, um die SAML-2.0-Authentifizierung und die Benutzerbereitstellung für Zabbix zu aktivieren.

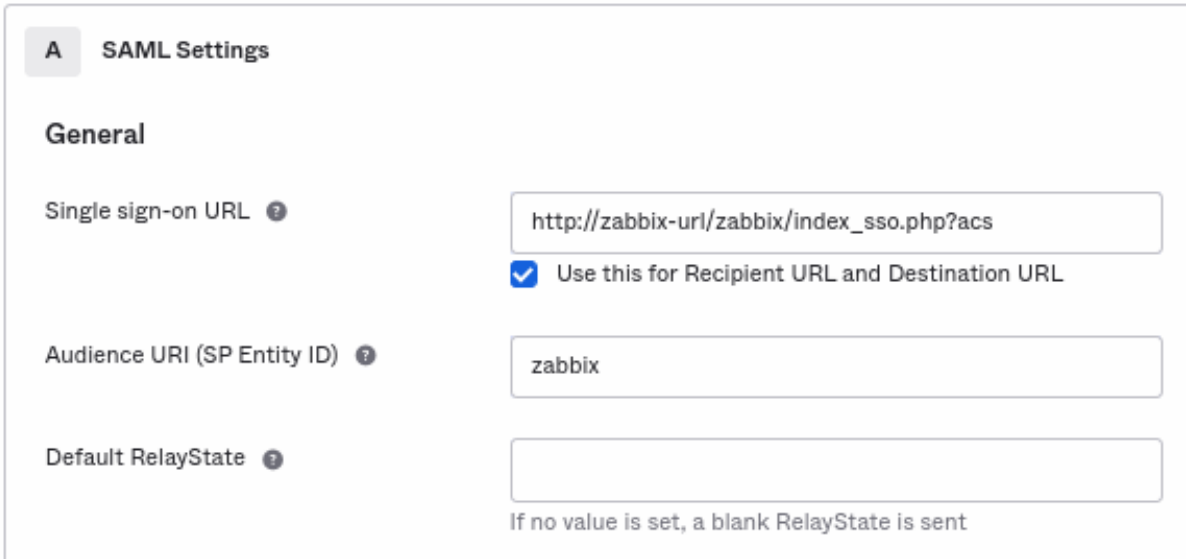
Okta-Konfiguration

1. Gehen Sie zu https://developer.okta.com/signup/ und registrieren Sie sich bzw. melden Sie sich bei Ihrem Konto an.
2. Navigieren Sie in der Okta-Weboberfläche zu *Applications* > *Applications*.
3. Klicken Sie auf *Create App Integration*.

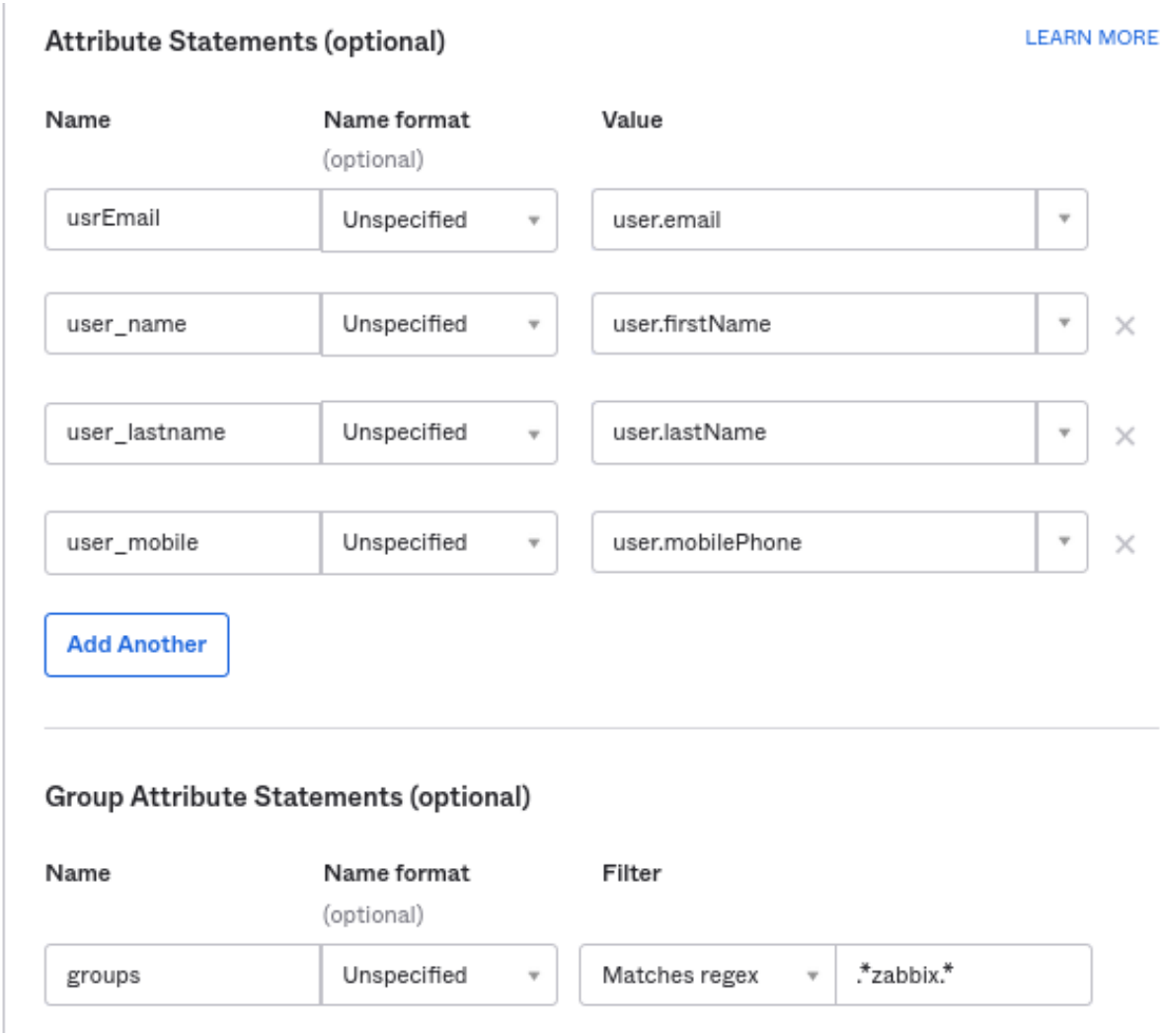


Wählen Sie "SAML 2.0" als Anmeldemethode und klicken Sie auf *Next*.

4. Geben Sie in den allgemeinen Einstellungen den App-Namen ein und klicken Sie auf *Next*.
5. Geben Sie in der SAML-Konfiguration die unten angegebenen Werte ein und klicken Sie dann auf *Next*.



- In **General** hinzufügen:
  - *Single sign-on URL*: `http://<your-zabbix-url>/zabbix/index_sso.php?acs<br>` Beachten Sie die Verwendung von "http" und nicht "https", damit der Parameter `acs` in der Anfrage nicht abgeschnitten wird. Das Kontrollkästchen *Use this for Recipient URL and Destination URL* sollte ebenfalls aktiviert sein.
  - *Audience URI (SP Entity ID)*: `zabbix<br>` Beachten Sie, dass dieser Wert innerhalb der SAML-Assertion als eindeutige Kennung des Service Providers verwendet wird (bei Nichtübereinstimmung wird der Vorgang abgelehnt). In diesem Feld kann eine URL oder eine beliebige Zeichenfolge angegeben werden.
  - *Default RelayState*:
 Lassen Sie dieses Feld leer; wenn eine benutzerdefinierte Weiterleitung erforderlich ist, kann sie in Zabbix in den Einstellungen *Users* > *Users* hinzugefügt werden.
  - Füllen Sie die übrigen Felder nach Ihren Wünschen aus.
- In **Attribute Statements/Group Attribute Statements** hinzufügen:



Diese Attributanweisungen werden in die mit Zabbix geteilten SAML-Assertions eingefügt.

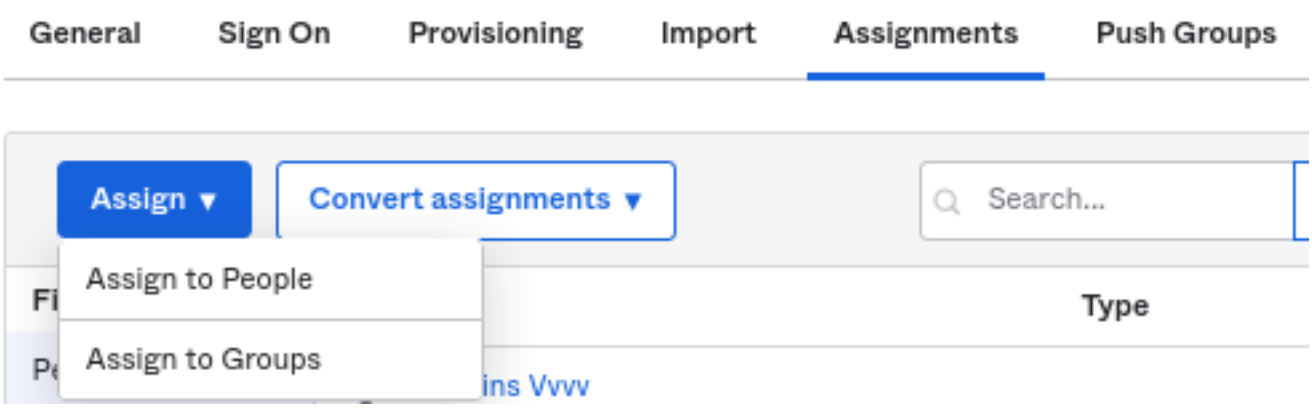
Die hier verwendeten Attributnamen sind beliebige Beispiele. Sie können andere Attributnamen verwenden; sie müssen jedoch mit dem jeweiligen Feldwert in den Zabbix-SAML-Einstellungen übereinstimmen.

Wenn Sie die SAML-Anmeldung bei Zabbix *ohne* JIT-Bereitstellung von Benutzern konfigurieren möchten, ist nur das E-Mail-Attribut erforderlich.

Note:
Wenn Sie eine verschlüsselte Verbindung verwenden möchten, generieren Sie die privaten und öffentlichen Verschlüsselungszertifikate und laden Sie anschließend das öffentliche Zertifikat in Okta hoch. Das Formular zum Hochladen des Zertifikats wird angezeigt, wenn *Assertion Encryption* auf "Encrypted" gesetzt ist (klicken Sie auf *Show Advanced Settings*, um diesen Parameter zu finden).

6. Wählen Sie im nächsten Tab "I'm a software vendor. I'd like to integrate my app with Okta" aus und klicken Sie auf *Finish*.

7. Navigieren Sie zum Tab "Assignments" der neu erstellten Anwendung und klicken Sie auf die Schaltfläche *Assign* und wählen Sie dann im Dropdown-Menü *Assign to People* aus.



8. Weisen Sie in dem angezeigten Popup die App den Personen zu, die SAML 2.0 zur Authentifizierung bei Zabbix verwenden werden, und klicken Sie dann auf *Save and go back*.

9. Laden Sie das IdP-Zertifikat herunter. Setzen Sie die Berechtigungen auf 644, indem Sie Folgendes ausführen:

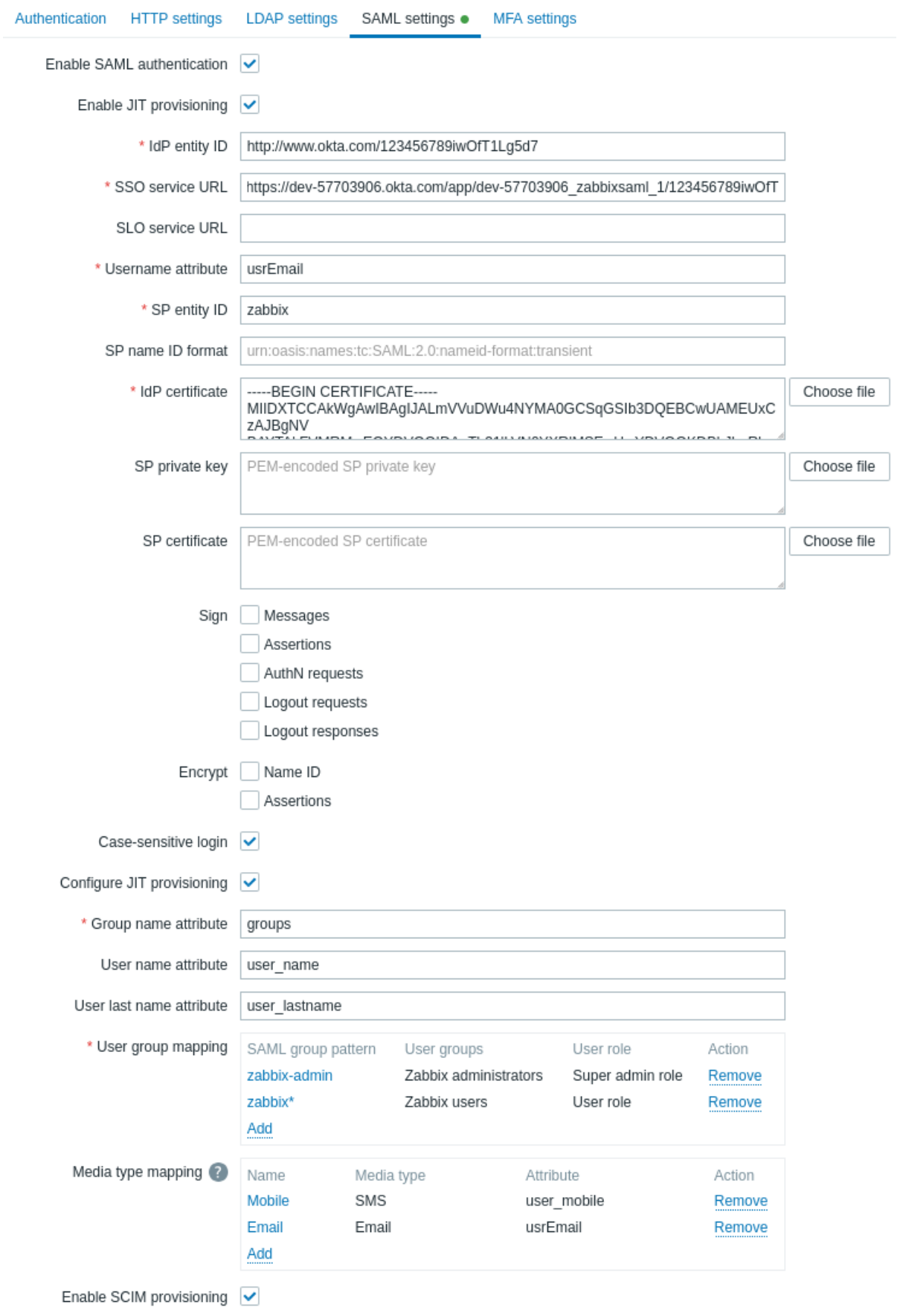
```
chmod 644 idp.crt
```

10. Navigieren Sie zum Tab "Sign On" und klicken Sie auf die Schaltfläche *View Setup Instructions*.

Die **Anweisungen** zur Einrichtung werden in einem neuen Tab geöffnet; lassen Sie diesen Tab während der Konfiguration von Zabbix geöffnet.

Zabbix-Konfiguration

1. Rufen Sie in Zabbix die SAML-Einstellungen auf und füllen Sie die Konfigurationsoptionen anhand der Einrichtungsanweisungen von Okta aus:



Zabbix-Feld	Einrichtungsfeld in Okta	Beispielwert
IdP entity ID	Identity Provider Issuer	
SSO service URL	Identity Provider Single Sign-On URL	
Username attribute	Attribute name	`usrEmail`
SP entity ID	Audience URI	`zabbix`
Group name attribute	Attribute name	`groups`

Zabbix-Feld	Einrichtungsfeld in Okta	Beispielwert
User name attribute	Attribute name	`user_name`
User last name attribute	Attribute name	`user_lastname`

Außerdem müssen die Zuordnung von Benutzergruppen und Medien konfiguriert werden.

2. Fügen Sie das in den Okta-SAML-Einrichtungsanweisungen bereitgestellte Base64-Zertifikat hinzu.

Wenn `$SSO['CERT_STORAGE'] = 'database'` in der Frontend-Konfigurationsdatei (`zabbix.conf.php`) festgelegt ist, können Sie den Zertifikatstext einfügen oder die Zertifikatsdatei während der SAML-Konfiguration im Frontend hochladen – auf dem Dateisystem sind keine Dateien erforderlich.

Wenn `$SSO['CERT_STORAGE'] = 'file'` in der Konfigurationsdatei festgelegt ist, muss das Zertifikat im Dateisystem verfügbar gemacht werden (standardmäßig in `ui/conf/certs` oder unter dem in der Konfigurationsdatei konfigurierten Pfad), und das Frontend speichert keine Zertifikate in der Datenbank. Beachten Sie, dass, wenn `$SSO['CERT_STORAGE']` nicht festgelegt oder auskommentiert ist, die Dateispeicherung verwendet wird und Zertifikate aus `ui/conf/certs` gelesen werden.

3. Wenn *Assertion Encryption* in Okta auf „Encrypted“ gesetzt wurde, muss in Zabbix ebenfalls das Kontrollkästchen „Assertions“ des Parameters *Encrypt* aktiviert werden.

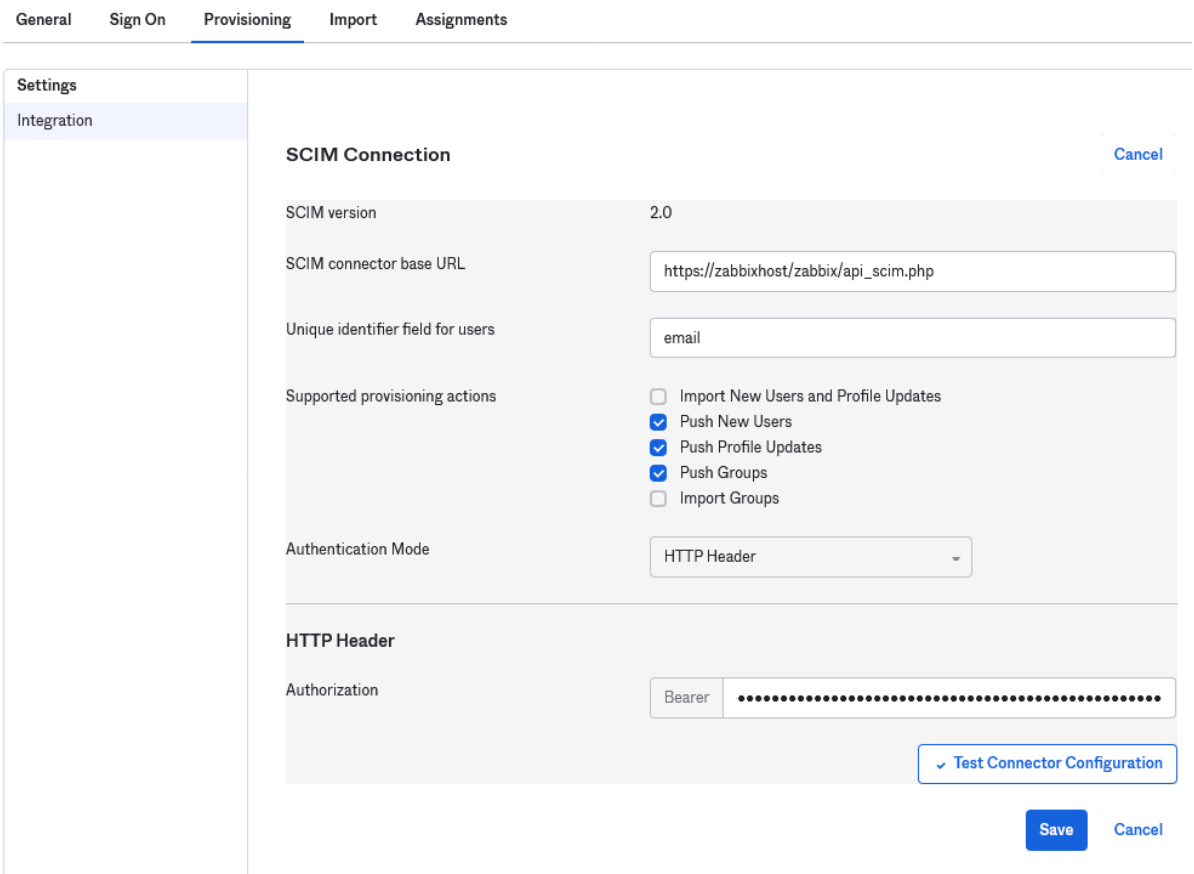
4. Klicken Sie auf die Schaltfläche *Update*, um diese Einstellungen zu speichern.

SCIM-Bereitstellung

1. Um die SCIM-Bereitstellung zu aktivieren, gehen Sie in Okta bei der Anwendung zu *General* > *App Settings*.

Aktivieren Sie das Kontrollkästchen *Enable SCIM provisioning*. Daraufhin wird eine neue Registerkarte *Provisioning* angezeigt.

2. Wechseln Sie zur Registerkarte „Provisioning“, um eine SCIM-Verbindung einzurichten: - Geben Sie unter *SCIM connector base URL* den Pfad zum Zabbix-Frontend an und hängen Sie `api_scim.php` daran an, z. B.:
 `https://<your-zabbix-url>/zabbix/api_s` - *Unique identifier field for users*: `email` - *Authentication mode*: `HTTP header` - Geben Sie unter *Authorization* ein gültiges API-Token mit Super-Admin-Rechten ein.



Attention:
Wenn Authentifizierungsprobleme auftreten, lesen Sie Weiterleitung des Authorization-Headers.

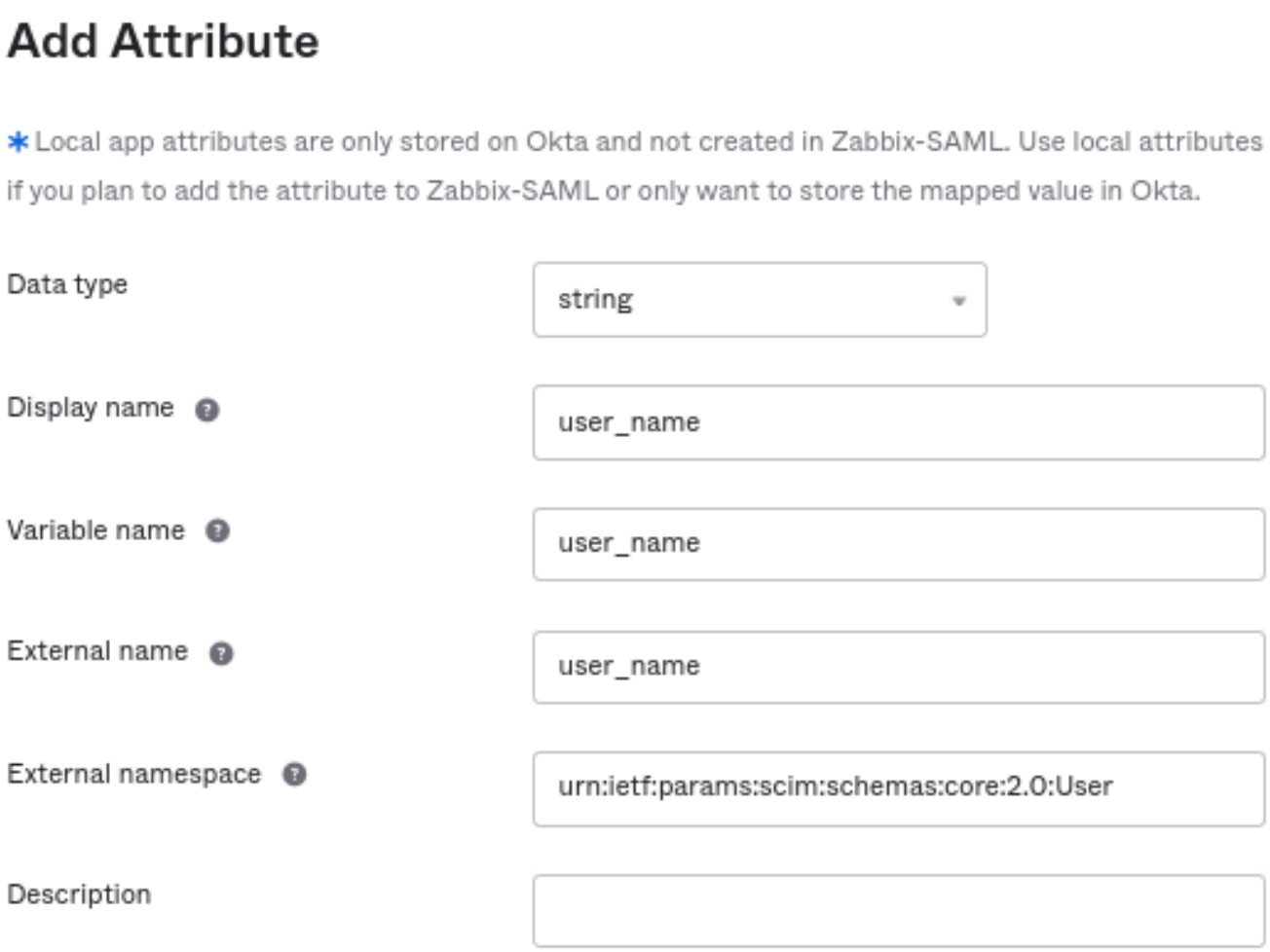
3. Klicken Sie auf *Test Connector Configuration*, um die Verbindung zu testen. Wenn alles korrekt ist, wird eine Erfolgsmeldung angezeigt.

4. Stellen Sie unter *Provisioning* > *To App* sicher, dass die folgenden Kontrollkästchen aktiviert sind: - Create Users - Update User Attributes - Deactivate Users

Dadurch wird sichergestellt, dass diese Anforderungstypen an Zabbix gesendet werden.

5. Stellen Sie sicher, dass alle in SAML definierten Attribute auch in SCIM definiert sind. Sie können den Profileditor für Ihre Anwendung unter *Provisioning* > *To App* öffnen, indem Sie auf *Go to Profile Editor* klicken.

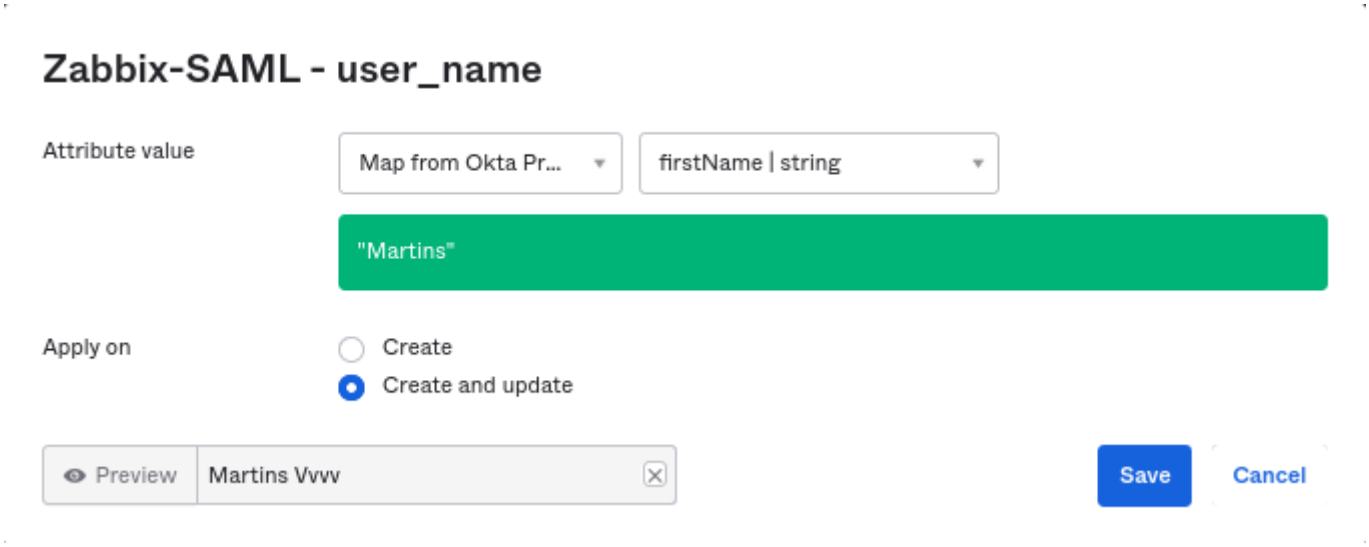
Klicken Sie auf *Add Attribute*. Tragen Sie für *Display name*, *Variable name* und *External name* den Namen des SAML-Attributs ein, z. B. `user_name`.



External namespace sollte mit dem Benutzerschema übereinstimmen: `urn:ietf:params:scim:schemas:core:2.0:User`

6. Wechseln Sie bei Ihrer Anwendung zu *Provisioning* > *To App* > *Attribute Mappings*. Klicken Sie unten auf *Show Unmapped Attributes*. Die neu hinzugefügten Attribute werden angezeigt.

7. Ordnen Sie jedes hinzugefügte Attribut zu.



8. Fügen Sie Benutzer auf der Registerkarte „Assignments“ hinzu. Die Benutzer müssen zuvor unter *Directory* > *People* hinzugefügt worden sein. Alle diese Zuweisungen werden als Anforderungen an Zabbix gesendet.

9. Fügen Sie Gruppen auf der Registerkarte „Push Groups“ hinzu. Das Muster für die Zuordnung von Benutzergruppen in den Zabbix-SAML-Einstellungen muss mit einer hier angegebenen Gruppe übereinstimmen. Wenn keine Übereinstimmung vorliegt, kann der Benutzer nicht in Zabbix erstellt werden.

Informationen zu Gruppenmitgliedern werden jedes Mal gesendet, wenn eine Änderung vorgenommen wird.

SAML-Einrichtung mit OneLogin

Übersicht

Dieser Abschnitt enthält Richtlinien für die Konfiguration von Single Sign-on und der Benutzerbereitstellung in Zabbix aus OneLogin mithilfe der SAML-2.0-Authentifizierung.

OneLogin-Konfiguration

Erstellen einer Anwendung

1. Melden Sie sich bei Ihrem Konto bei OneLogin an. Zu Testzwecken können Sie in OneLogin ein kostenloses Entwicklerkonto erstellen.

2. Navigieren Sie in der OneLogin-Weboberfläche zu *Applications* > *Applications*.

3. Klicken Sie auf ”Add App” und suchen Sie nach der passenden App. Die Richtlinien auf dieser Seite basieren auf dem App-Beispiel *SCIM Provisioner with SAML (SCIM v2 Enterprise, full SAML)*.

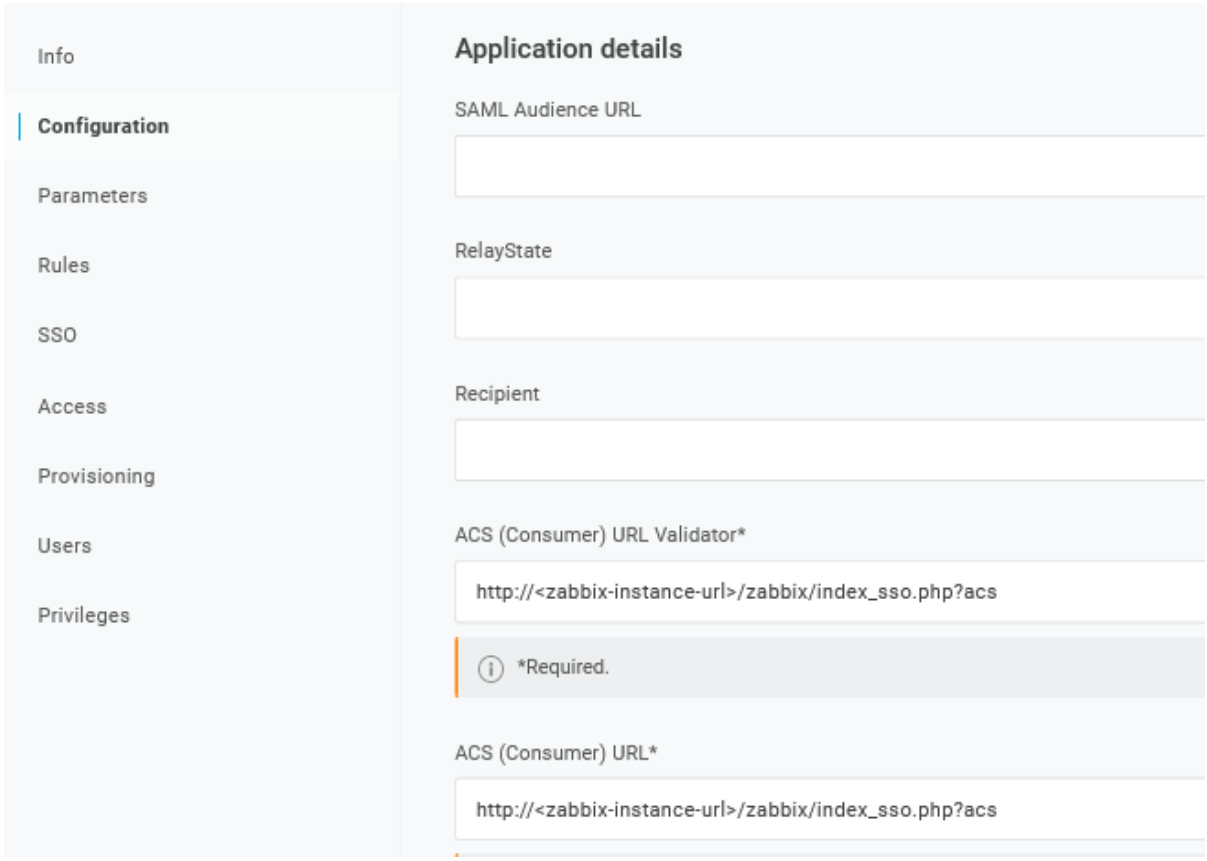
4. Zunächst möchten Sie möglicherweise den Anzeigenamen Ihrer App anpassen. Möglicherweise möchten Sie auch das Symbol und die App-Details hinzufügen. Klicken Sie danach auf *Save*.

SSO-/SCIM-Bereitstellung einrichten

1. Navigieren Sie zu *Configuration* > *Application details* und legen Sie den Zabbix-Endpunkt für einmaliges Anmelden `http://<zabbix-instance-url>/zabbix/index_sso.php?acs` als Wert für die folgenden Felder fest:

- *ACS (Consumer) URL Validator*
- *ACS (Consumer) URL*

Beachten Sie die Verwendung von „http“ und nicht „https“, damit der Parameter `acs` nicht aus der Anfrage entfernt wird.



Es ist auch möglich, „https“ zu verwenden. Damit dies mit Zabbix funktioniert, fügen Sie der Frontend-Konfigurationsdatei (`zabbix.conf.php`) die folgende Zeile hinzu:

```
$SSO['SETTINGS'] = ['use_proxy_headers' => true];
```

Belassen Sie die anderen Optionen bei ihren Standardwerten.

2. Navigieren Sie zu *Configuration* > *API connection* und legen Sie die folgenden Werte fest:

- *SCIM Base URL*: `https://<zabbix-instance-url>/zabbix/api_scim.php`
- *SCIM JSON Template*: Muss alle benutzerdefinierten Attribute enthalten, die Sie über SCIM an Zabbix übergeben möchten, z. B. `user_name`, `user_lastname`, `user_email` und `user_mobile`:

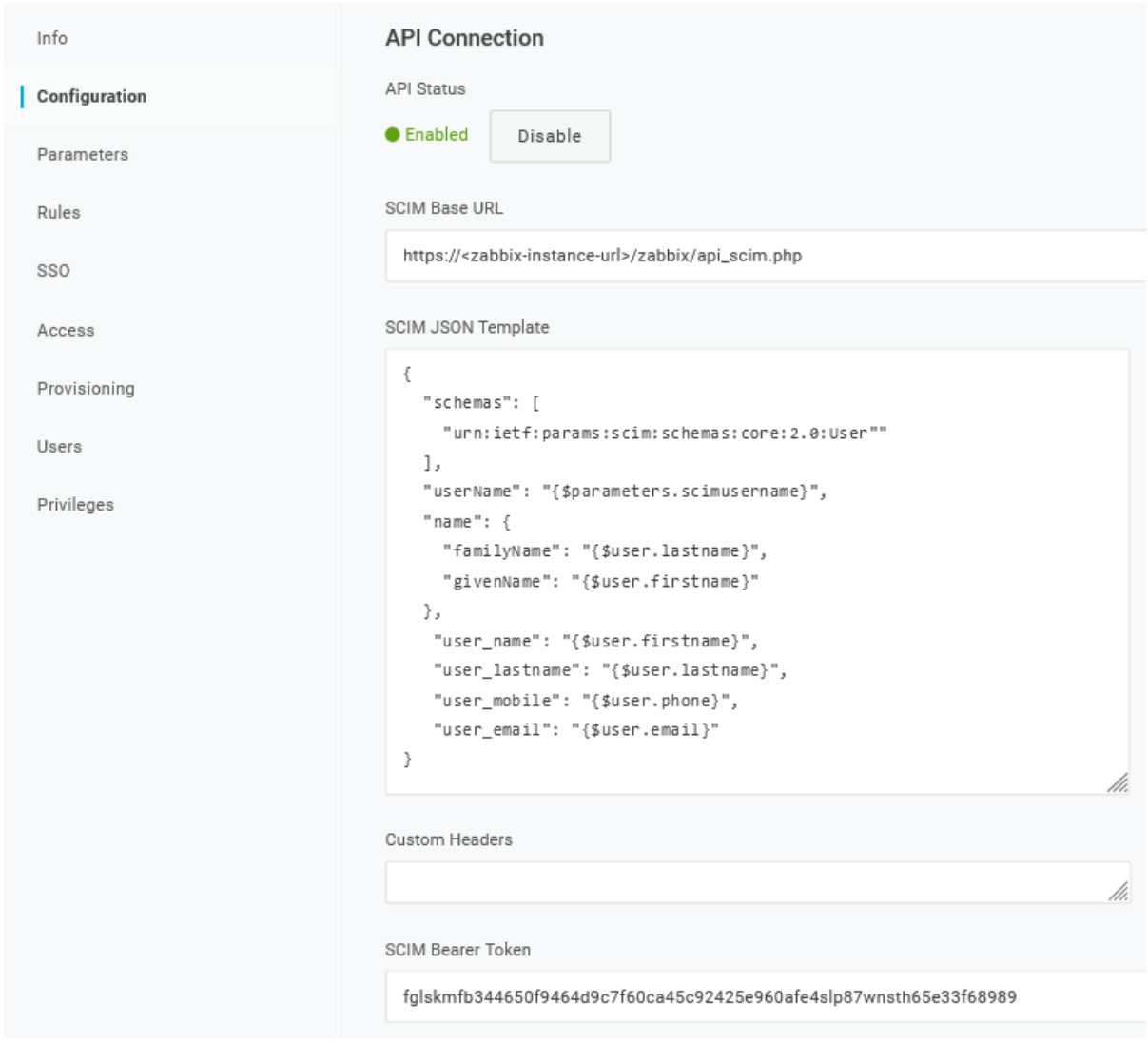
```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User"
  ],
  "userName": "{$parameters.scimusername}",
  "name": {
    "familyName": "{$user.lastname}",
    "givenName": "{$user.firstname}"
  },
   "user_name": "{$user.firstname}",
   "user_lastname": "{$user.lastname}",
   "user_mobile": "{$user.phone}",
   "user_email": "{$user.email}"
}
```

Die Attributnamen sind beliebig. Es können andere Attributnamen verwendet werden. Sie müssen jedoch mit dem jeweiligen Feldwert in den Zabbix-SAML-Einstellungen übereinstimmen.

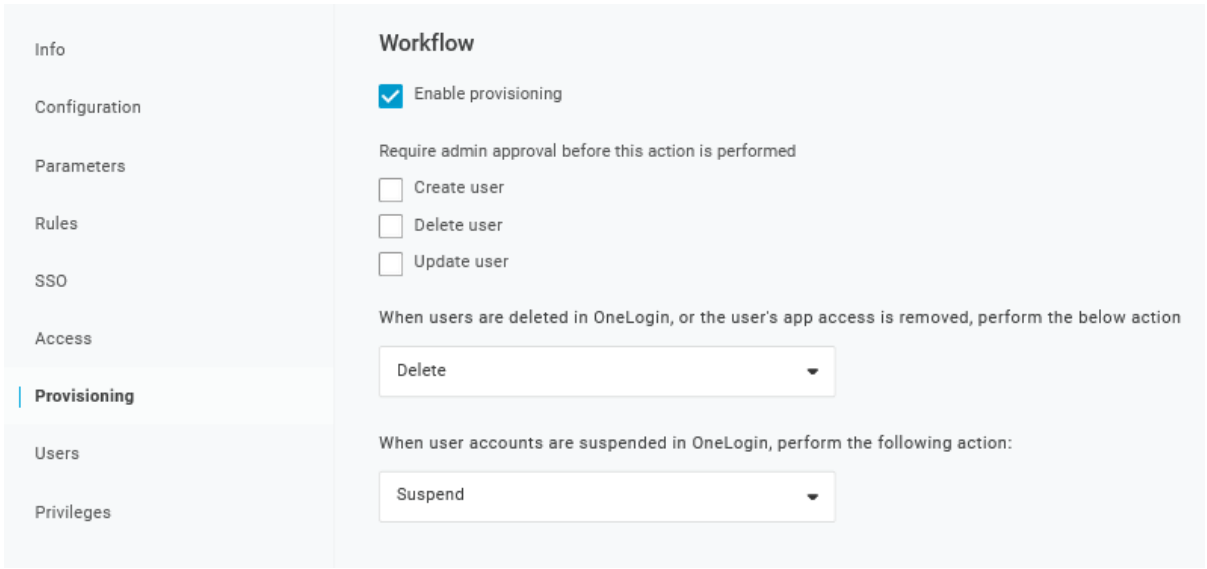
Beachten Sie, dass OneLogin für die Benutzerbereitstellung in der Antwort ein Attribut „name“ mit „givenName“ und „familyName“ empfangen muss, auch wenn dies vom Dienstanbieter nicht angefordert wurde. Daher muss dies im Schema im Abschnitt für die Anwendungskonfiguration angegeben werden.

- *SCIM Bearer Token*: Geben Sie ein Zabbix-API-Token mit Super-Admin-Berechtigungen ein.

Klicken Sie auf *Enable*, um die Verbindung zu aktivieren.

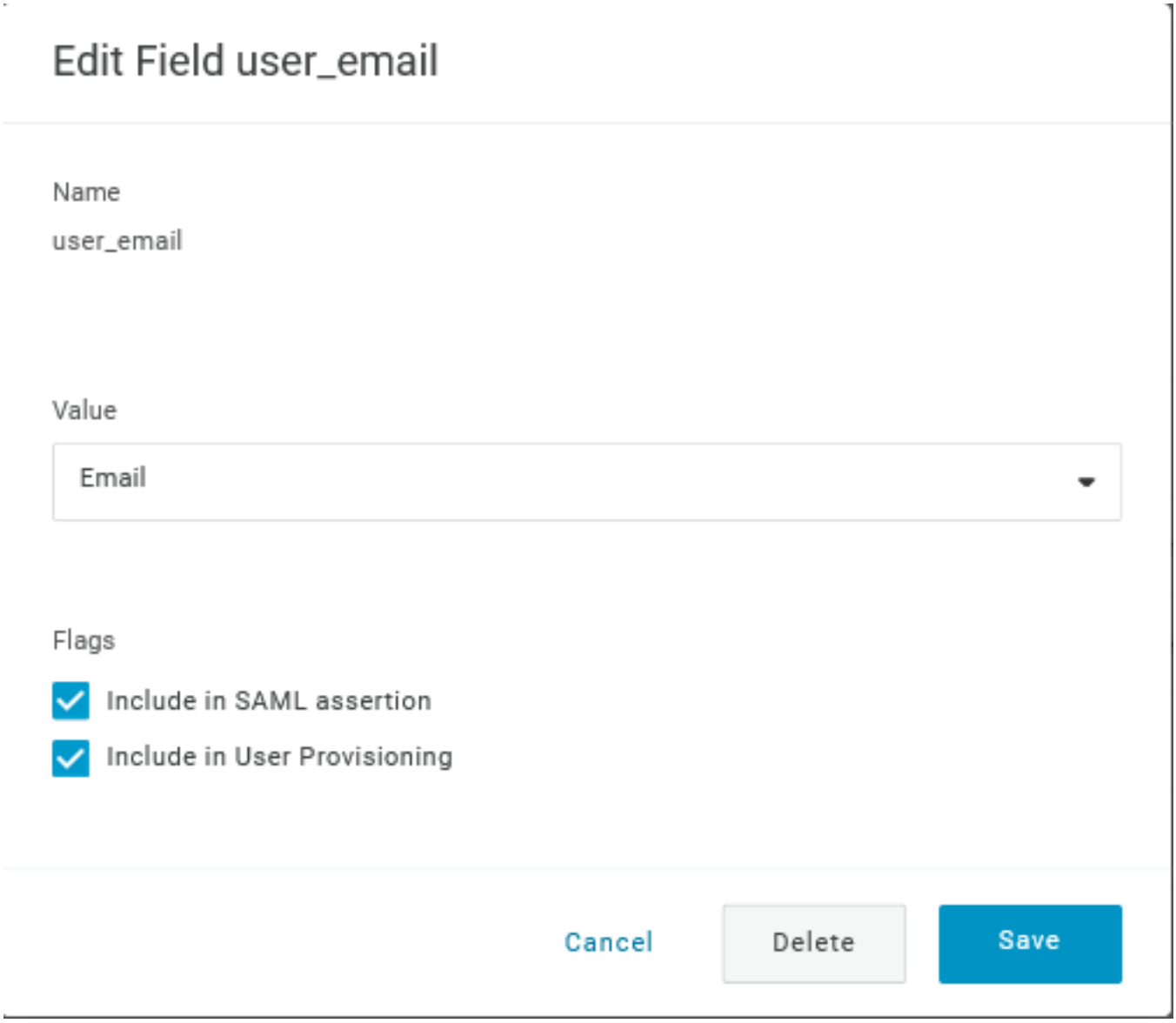


3. Aktivieren Sie auf der Seite *Provisioning* die Option *Provisioning*:



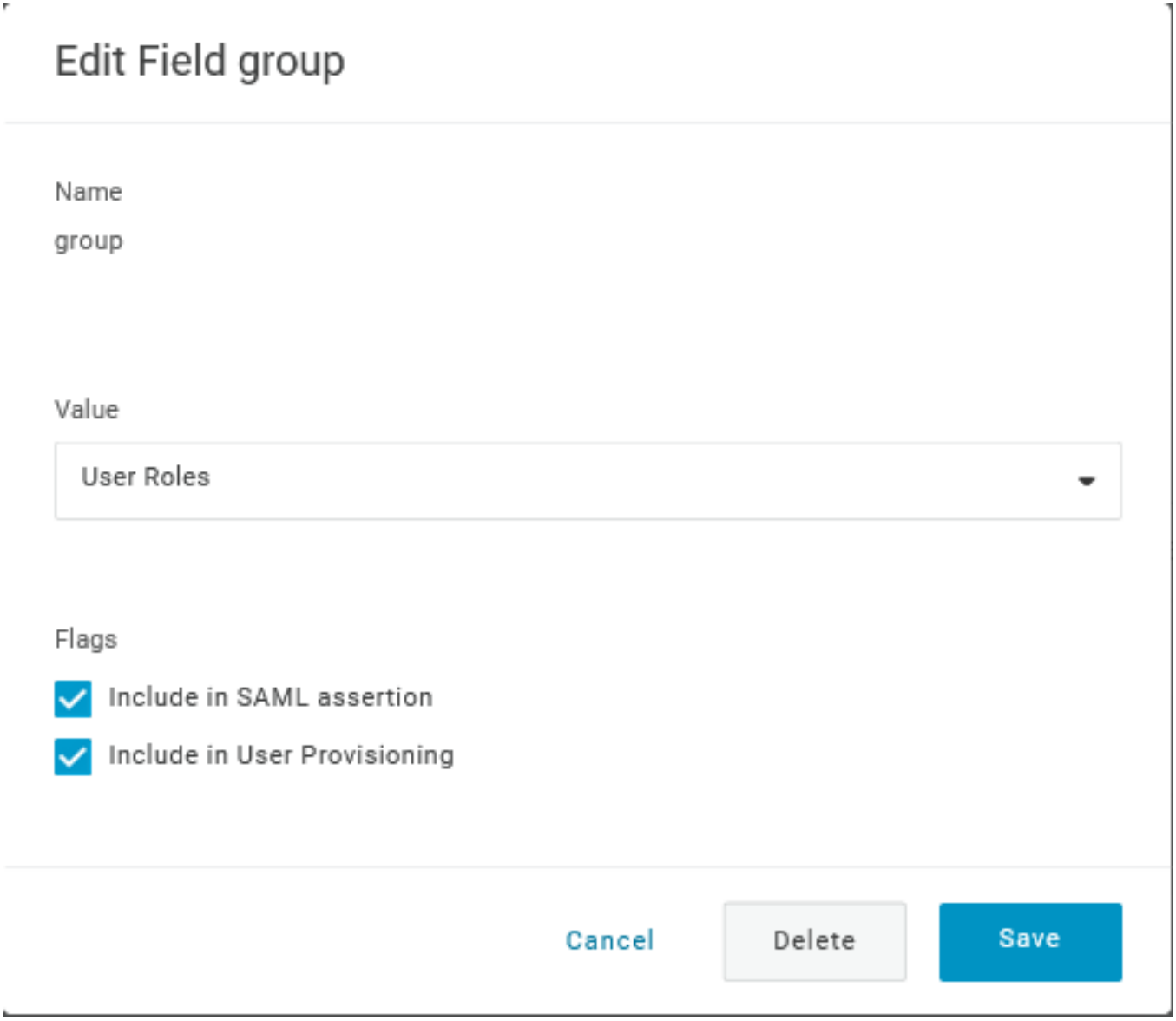
4. Die Seite *Parameters* enthält eine Liste der Standardparameter:

- Stellen Sie sicher, dass „scimusername“ mit dem Benutzeranmeldewert in OneLogin übereinstimmt (z. B. der E-Mail-Adresse).
- Aktivieren Sie für den Parameter „Groups“ die Option *Include in User Provisioning*.
- Klicken Sie auf „+“, um die für SAML-Assertions und die Benutzerbereitstellung erforderlichen benutzerdefinierten Parameter zu erstellen, z. B. `user_name`, `user_lastname`, `user_email` und `user_mobile`:

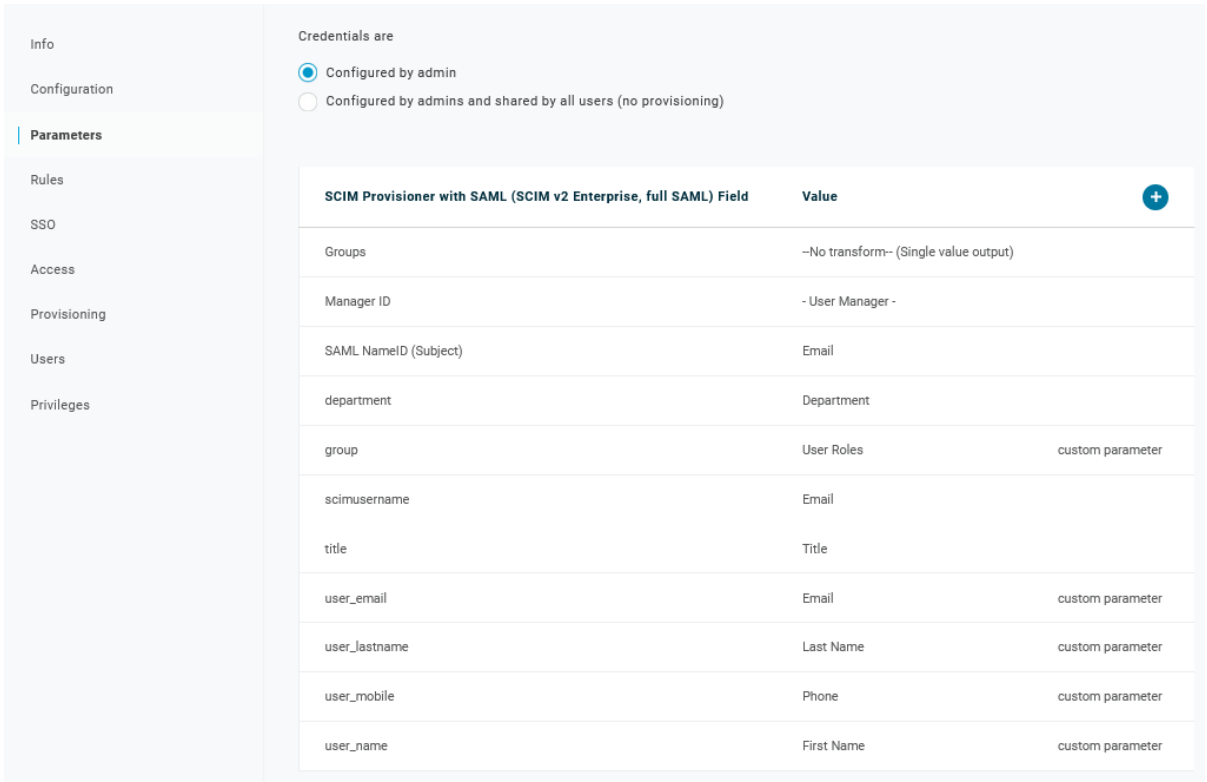


Achten Sie beim Hinzufügen eines Parameters darauf, sowohl die Optionen *Include in SAML assertion* als auch *Include in User Provisioning* zu aktivieren.

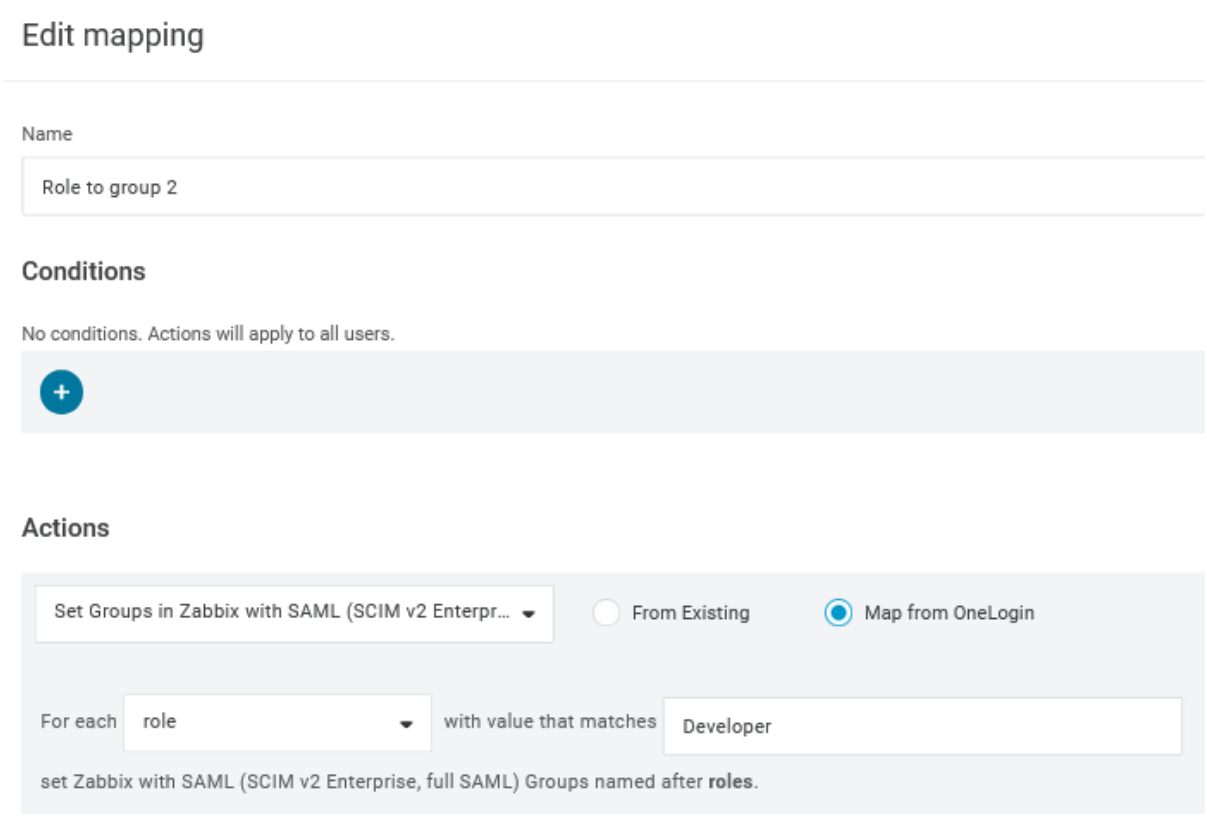
- Fügen Sie einen Parameter „group“ hinzu, der den Benutzerrollen in OneLogin entspricht. Die Benutzerrollen werden als durch ein Semikolon ; getrennte Zeichenfolge übergeben. Die OneLogin-Benutzerrollen werden zum Erstellen von Benutzergruppen in Zabbix verwendet:



Überprüfen Sie die Liste der Parameter:



5. Erstellen Sie auf der Seite *Rules* Benutzerrollenzuordnungen zum Standardparameter „Groups“.



Sie können einen regulären Ausdruck verwenden, um bestimmte Rollen als Gruppen zu übergeben. Die Rollennamen dürfen kein ; enthalten, da OneLogin dieses Zeichen als Trennzeichen verwendet, wenn ein Attribut mit mehreren Rollen gesendet wird.

6. Laden Sie das IdP-Zertifikat herunter. Legen Sie die Berechtigungen mit folgendem Befehl auf 644 fest:

```
chmod 644 idp.crt
```

Zabbix-Konfiguration

1. Rufen Sie in Zabbix die SAML-Einstellungen auf und füllen Sie die Konfigurationsoptionen basierend auf der OneLogin-Konfiguration aus:



Zabbix-Feld	Setup-Feld in OneLogin	Beispielwert
IdP entity ID	Issuer URL (siehe Registerkarte „SSO“ Ihrer Anwendung in OneLogin)	

Zabbix-Feld	Setup-Feld in OneLogin	Beispielwert
SSO service URL	SAML 2.0 Endpoint (HTTP) (siehe Registerkarte „SSO“ Ihrer Anwendung in OneLogin)	
SLO service URL	SLO Endpoint (HTTP) (siehe Registerkarte „SSO“ Ihrer Anwendung in OneLogin)	
Username attribute	Custom parameter	`user_email`
Group name attribute	Custom parameter	`group`
User name attribute	Custom parameter	`user_name`
User last name attribute	Custom parameter	`user_lastname`

Außerdem muss die Zuordnung von Benutzergruppen konfiguriert werden.
Die Medienzuordnung ist optional.
Klicken Sie auf *Update*, um diese Einstellungen zu speichern.

2. Fügen Sie das von OneLogin bereitgestellte Base64-Zertifikat hinzu.

Wenn `$SSO['CERT_STORAGE'] = 'database'` in der Frontend-Konfigurationsdatei (`zabbix.conf.php`) festgelegt ist, können Sie den Zertifikatstext einfügen oder die Zertifikatdatei während der SAML-Konfiguration im Frontend hochladen. Auf dem Dateisystem werden keine Dateien benötigt.

Wenn `$SSO['CERT_STORAGE'] = 'file'` in der Konfigurationsdatei festgelegt ist, muss das Zertifikat im Dateisystem verfügbar gemacht werden (standardmäßig in `ui/conf/certs` oder in dem in der Konfigurationsdatei konfigurierten Pfad), und das Frontend speichert keine Zertifikate in der Datenbank.
Beachten Sie, dass die Dateispeicherung angenommen wird und Zertifikate aus `ui/conf/certs` gelesen werden, wenn `$SSO['CERT_STORAGE']` nicht festgelegt oder auskommentiert ist.

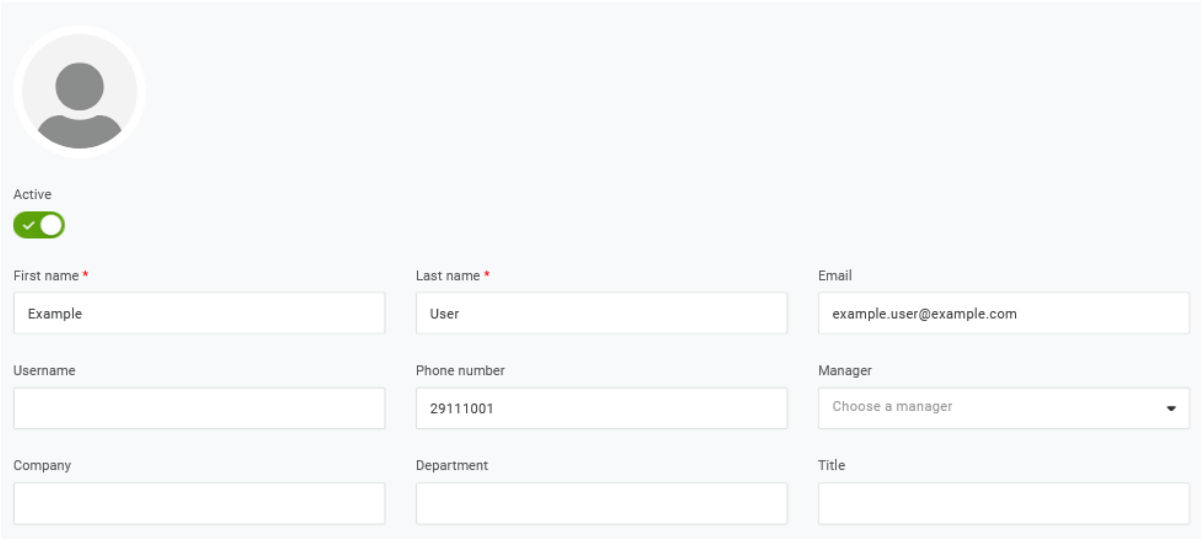
Sie können den Zertifikatdownload in OneLogin unter *Applications* > *SSO* aufrufen, indem Sie unter dem aktuellen Zertifikat auf *View details* klicken.

3. Klicken Sie auf die Schaltfläche *Update*, um diese Einstellungen zu speichern.

SCIM-Benutzerbereitstellung

Wenn die Benutzerbereitstellung aktiviert ist, ist es nun möglich, Benutzer und ihre Rollen in OneLogin hinzuzufügen/zu aktualisieren und sie sofort in Zabbix bereitzustellen.

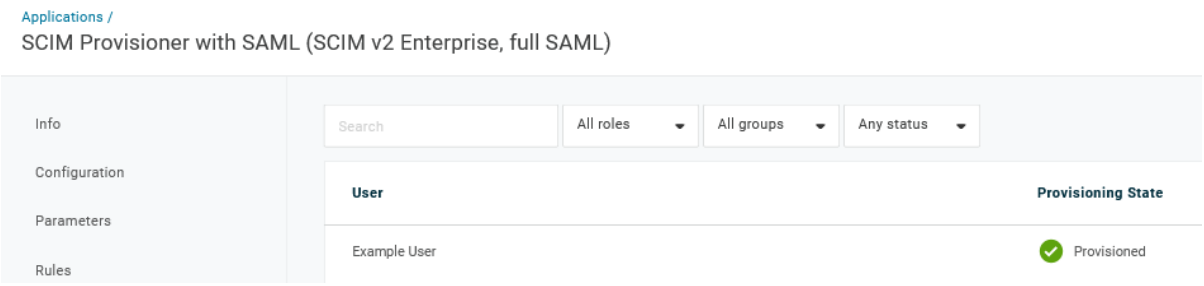
Sie können beispielsweise einen neuen Benutzer erstellen:



Fügen Sie ihn einer Benutzerrolle und der Anwendung hinzu, die den Benutzer bereitstellen wird:



Beim Speichern des Benutzers wird er in Zabbix bereitgestellt. Unter *Application* > *Users* können Sie den Bereitstellungsstatus der aktuellen Anwendungsbenutzer überprüfen:



Wenn die Bereitstellung erfolgreich war, ist der Benutzer in der Zabbix-Benutzerliste sichtbar.



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

Einrichten geplanter Berichte

Übersicht

Dieser Abschnitt enthält Anweisungen zur Installation des Zabbix-Webservice und zur Konfiguration von Zabbix, um die Erstellung von geplanten Berichten zu ermöglichen.

Installation

Ein neuer Prozess des Zabbix-Webservice und der Google Chrome-Browser sollten installiert werden, um die Erstellung geplanter Berichte zu ermöglichen. Der Webservice kann auf demselben Rechner installiert werden, auf dem der Zabbix Server installiert ist, oder auf einem anderen Rechner. Der Google Chrome-Browser sollte auf demselben Rechner installiert werden, auf dem der Webservice installiert ist.

Das offizielle Paket zabbix-web-service ist im Zabbix repository verfügbar. Der Browser Google Chrome ist in diesen Paketen nicht enthalten und muss separat installiert werden.

Zum Kompilieren des Zabbix-Webdienstes aus den Quellen siehe Installing Zabbix web service.

Nach der Installation führen Sie `zabbix_web_service` auf dem Rechner aus, auf dem der Webdienst installiert ist:

```
zabbix_web_service
```

Konfiguration

Um eine ordnungsgemäße Kommunikation zwischen allen beteiligten Elementen sicherzustellen, müssen die Parameter der Serverkonfiguration und der Frontend-Konfiguration korrekt konfiguriert sein.

Zabbix Server

Die folgenden Parameter in der Zabbix-Server-Konfigurationsdatei müssen aktualisiert werden: `WebServiceURL` und `StartReportWriters`.

WebServiceURL

Dieser Parameter ist erforderlich, um die Kommunikation mit dem Webservice zu aktivieren.
Die URL muss das Format `http[s]://host:port/report` haben.

- Standardmäßig lauscht der Webservice auf Port 10053.
  Ein anderer Port kann in der Webservice-Konfigurationsdatei angegeben werden.
- Die Angabe des Pfads `/report` ist obligatorisch (der Pfad ist fest vorgegeben und kann nicht geändert werden).

Beispiel:

```
WebServiceURL=http://localhost:10053/report
```

Attention:
Es wird dringend empfohlen, die Verschlüsselung zwischen Zabbix Server und Zabbix Webservice mithilfe von Zertifikaten einzurichten.
Standardmäßig werden zwischen Zabbix Server und Zabbix Webservice übertragene Daten nicht verschlüsselt, was zu unbefugtem Zugriff führen kann.

StartReportWriters

Dieser Parameter legt fest, wie viele Prozesse für die Berichtserstellung gestartet werden sollen.
Wenn er nicht festgelegt ist oder den Wert 0 hat, ist die Berichtserstellung deaktiviert.
Abhängig von der Anzahl und Häufigkeit der benötigten Berichte können 1 bis 100 Prozesse für die Berichtserstellung aktiviert werden.

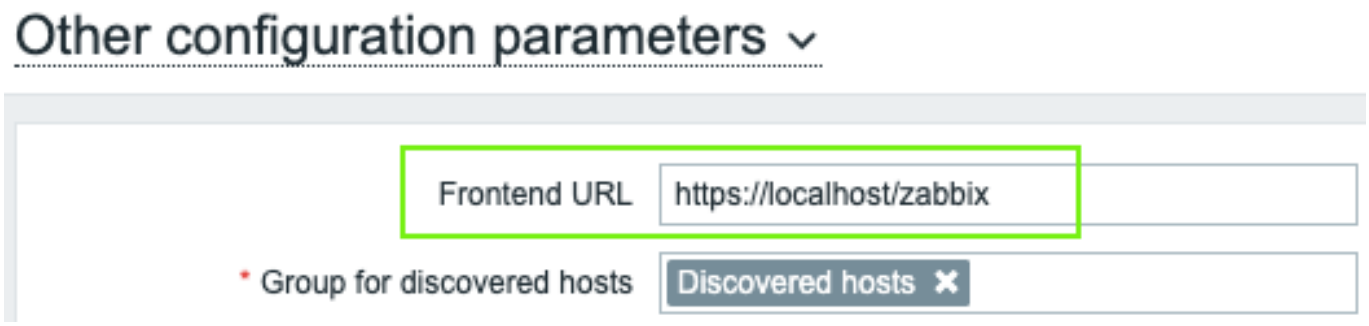
Beispiel:

```
StartReportWriters=3
```

Zabbix Frontend

Ein Parameter *Frontend URL* sollte festgelegt werden, um die Kommunikation zwischen dem Zabbix Frontend und dem Zabbix-Webservice zu aktivieren:

- Gehen Sie zum Frontend-Menüabschnitt *Administration* > *General* > *Other*
- Geben Sie im Parameter *Frontend URL* die vollständige URL der Zabbix-Weboberfläche an



Note:
Sobald der Einrichtungsprozess abgeschlossen ist, sollten Sie einen Testbericht konfigurieren und senden, um sicherzustellen, dass alles korrekt funktioniert.

Datenpunkte

Dieser Abschnitt enthält Referenzinformationen und technische Hinweise zu Datenpunkten, einschließlich Parameterdetails für bestimmte Datenpunktschlüssel, plattformspezifischer Details und Hinweise zu speziellen Datenpunktkonfigurationen.

Bitte verwenden Sie die Seitenleiste, um die für Ihren Anwendungsfall relevanten Informationen zu finden.

vm.memory.size-Parameter

Übersicht

Dieser Abschnitt enthält einige Parameterdetails für den Agent-Datenpunkt vm.memory.size[<mode>].

Parameter

Die folgenden Parameter sind für diesen Datenpunkt verfügbar:

- **active** - Speicher, der derzeit verwendet wird oder bis vor Kurzem verwendet wurde und sich daher im RAM befindet
- **anon** - Speicher, der keiner Datei zugeordnet ist (kann nicht daraus erneut eingelesen werden)
- **available** - verfügbarer Speicher, je nach Plattform unterschiedlich berechnet (siehe Tabelle unten)
- **buffers** - Cache für Dinge wie Dateisystem-Metadaten
- **cached** - Cache für verschiedene Dinge
- **exec** - ausführbarer Code, typischerweise aus einer (Programm-)Datei
- **file** - Cache für Inhalte kürzlich aufgerufener Dateien
- **free** - Speicher, der jeder Entität, die Speicher anfordert, unmittelbar zur Verfügung steht
- **inactive** - Speicher, der als nicht verwendet markiert ist
- **pavailable** - Speicher 'available' als Prozentsatz von 'total' (berechnet als `available/total*100`)
- **pinned** - dasselbe wie 'wired'
- **pused** - Speicher 'used' als Prozentsatz von 'total' (berechnet als `used/total*100`)
- **shared** - Speicher, auf den mehrere Prozesse gleichzeitig zugreifen können

- **slab** - Gesamtmenge des Speichers, den der Kernel zum Zwischenspeichern von Datenstrukturen für den eigenen Gebrauch verwendet
- **total** - insgesamt verfügbarer physischer Speicher
- **used** - verwendeter Speicher, je nach Plattform unterschiedlich berechnet (siehe Tabelle unten)
- **wired** - Speicher, der so markiert ist, dass er immer im RAM verbleibt. Er wird niemals auf den Datenträger verschoben.

Warning:
Einige dieser Parameter sind plattformspezifisch und möglicherweise auf Ihrer Plattform nicht verfügbar. Siehe Zabbix agent items für Details.

Plattformspezifische Berechnung von **available** und **used**:

Platform	**"available"**	**"used"**
AIX	free + cached	tatsächlich verwendeter Speicher
FreeBSD	inactive + cached + free	active + wired + cached
HP UX	free	total - free
Linux<3.14	free + buffers + cached	total - free
Linux 3.14+ (auch nach 3.10 auf RHEL 7 zurückportiert)	/proc/meminfo, siehe "MemAvailable" in der Linux-Kernel-Dokumentation für Details. Beachten Sie, dass free + buffers + cached nicht mehr gleich 'available' ist, da nicht der gesamte Page-Cache freigegeben werden kann und der Low-Watermark in der Berechnung verwendet wird.	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:
Die Summe von *vm.memory.size[used]* und *vm.memory.size[available]* entspricht nicht zwangsläufig dem Gesamtwert. Zum Beispiel unter FreeBSD:
 * Active-, inactive-, wired- und cached-Speicher werden als verwendet betrachtet, weil sie nützliche Informationen enthalten.
 * Gleichzeitig werden inactive-, cached- und free-Speicher als verfügbar betrachtet, weil diese Speicherarten Prozessen, die mehr Speicher anfordern, sofort zugewiesen werden können.
Damit ist inactive-Speicher gleichzeitig verwendet und verfügbar. Aus diesem Grund ist der Datenpunkt *vm.memory.size[used]* nur für Informationszwecke vorgesehen, während *vm.memory.size[available]* für die Verwendung in Auslösern vorgesehen ist.

Siehe auch

1. Zusätzliche Details zur Speicherberechnung in verschiedenen Betriebssystemen

Mindestberechtigungsstufe für Windows-Agent-Datenpunkte

Übersicht

Bei der Überwachung von Systemen mit einem Agent ist es eine bewährte Praxis, Metriken von dem Host zu erfassen, auf dem der Agent installiert ist. Um das Prinzip der geringsten Rechte anzuwenden, muss festgelegt werden, welche Metriken vom Agent erfasst werden.

Die Tabelle in diesem Dokument ermöglicht es Ihnen, die minimalen Rechte für einen garantiert korrekten Betrieb des Zabbix Agent auszuwählen.

Wenn für den Betrieb des Agent anstelle von „LocalSystem" ein anderer Benutzer ausgewählt wird, muss der neue Benutzer für den Betrieb des Agent als Windows-Dienst über das Recht „Als Dienst anmelden" aus „Lokale Richtlinie→Zuweisen von Benutzerrechten" sowie über das Recht zum Erstellen, Schreiben und Löschen der Zabbix-Agent-Protokolldatei verfügen. Ein Active-Directory-Benutzer muss der Gruppe *Leistungsüberwachungsbenutzer* hinzugefügt werden.

Note:
Bei der Arbeit mit den Rechten eines Agent auf Basis der Gruppe „technisch minimal akzeptabel" ist es erforderlich, im Voraus Rechte für zu überwachende Objekte bereitzustellen.

Von Agent unterstützte Standard-Datenpunkte unter Windows

Item key	User group	
	Empfohlen	Minimal technisch akzeptabel (Funktionalität ist eingeschränkt)
agent.hostname	Guests	Guests
agent.ping	Guests	Guests
agent.variant	Guests	Guests
agent.version	Guests	Guests
log	Administrators	Guests
log.count	Administrators	Guests
logrt	Administrators	Guests
logrt.count	Administrators	Guests
net.dns	Guests	Guests
net.dns.perf	Guests	Guests
net.dns.record	Guests	Guests
net.if.discovery	Guests	Guests
net.if.in	Guests	Guests
net.if.out	Guests	Guests
net.if.total	Guests	Guests
net.tcp.listen	Guests	Guests
net.tcp.port	Guests	Guests
net.tcp.service	Guests	Guests
net.tcp.service.perf	Guests	Guests
net.udp.service	Guests	Guests
net.udp.service.perf	Guests	Guests
proc.num	Administrators	Guests
system.cpu.discovery	Performance Monitor Users	Performance Monitor Users
system.cpu.load	Performance Monitor Users	Performance Monitor Users
system.cpu.num	Guests	Guests
system.cpu.util	Performance Monitor Users	Performance Monitor Users
system.hostname	Guests	Guests
system.localtime	Guests	Guests
system.run	Administrators	Guests
system.sw.arch	Guests	Guests
system.swap.size	Guests	Guests
system.uname	Guests	Guests
system.uptime	Performance Monitor Users	Performance Monitor Users
vfs.dir.count	Administrators	Guests
vfs.dir.get	Administrators	Guests
vfs.dir.size	Administrators	Guests
vfs.file.cksum	Administrators	Guests
vfs.file.contents	Administrators	Guests
vfs.file.exists	Administrators	Guests
vfs.file.md5sum	Administrators	Guests
vfs.file.regexp	Administrators	Guests
vfs.file.regmatch	Administrators	Guests
vfs.file.size	Administrators	Guests
vfs.file.time	Administrators	Guests
vfs.fs.discovery	Administrators	Guests
vfs.fs.size	Administrators	Guests
vfs.fs.get	Administrators	Guests
vm.memory.size	Guests	Guests
web.page.get	Guests	Guests
web.page.perf	Guests	Guests
web.page.regexp	Guests	Guests
zabbix.stats	Guests	Guests

Windows-spezifische Datenpunkt-Schlüssel

Item key	User group	
	Empfohlen	Mindestens technisch akzeptabel (Funktionalität ist eingeschränkt)
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
vm.vmemory.size	Guests	Guests

Kodierung zurückgegebener Werte

Der Zabbix Server erwartet, dass jeder zurückgegebene Textwert in der UTF8-Kodierung vorliegt. Dies gilt für jede Art von Prüfungen: Zabbix Agent, SSH, Telnet usw.

Unterschiedliche überwachte Systeme/Geräte und Prüfungen können nicht-ASCII-Zeichen im Wert zurückgeben. Für solche Fälle enthalten fast alle möglichen **zabbix**-Schlüssel einen zusätzlichen Datenpunkt-Schlüsselparameter - **<encoding>**. Dieser Schlüsselparameter ist optional, sollte jedoch angegeben werden, wenn der zurückgegebene Wert nicht in UTF8 kodiert ist und nicht-ASCII-Zeichen enthält. Andernfalls kann das Ergebnis unerwartet und unvorhersehbar sein.

Im Folgenden wird das Verhalten bei unterschiedlichen Datenbank-Backends in solchen Fällen beschrieben.

MySQL

Wenn ein Wert in einer Nicht-UTF8-Kodierung ein Nicht-ASCII-Zeichen enthält, werden dieses Zeichen und die folgenden Zeichen verworfen, wenn die Datenbank diesen Wert speichert. Es werden keine Warnmeldungen in die `zabbix_server.log` geschrieben.
 Gilt mindestens für MySQL Version 5.1.61.

PostgreSQL

Wenn ein Wert in einer Nicht-UTF8-Kodierung ein Nicht-ASCII-Zeichen enthält, führt dies zu einer fehlgeschlagenen SQL-Abfrage (PGRES_FATAL_ERROR:ERROR invalid byte sequence for encoding), und die Daten werden nicht gespeichert. Eine entsprechende Warnmeldung wird in die Datei `zabbix_server.log` geschrieben.
 Gilt mindestens für PostgreSQL Version 9.1.3.

Unterstützung großer Dateien

Unterstützung großer Dateien, oft als LFS abgekürzt, bezeichnet die Fähigkeit, mit Dateien größer als 2 GB auf 32-Bit-Betriebssystemen zu arbeiten. Die Unterstützung großer Dateien betrifft mindestens die Überwachung von Protokolldateien und alle vfs.file.* Datenpunkte. Die Unterstützung großer Dateien hängt von den Fähigkeiten eines Systems zum Zeitpunkt der Zabbix-Kompilierung ab, ist jedoch auf einem 32-Bit-Solaris aufgrund der Inkompatibilität mit procfs und swapctl vollständig deaktiviert.

Sensor

Übersicht

Jeder Sensorchip erhält sein eigenes Verzeichnis im sysfs-Baum `/sys/devices`. Um alle Sensorchips zu finden, ist es einfacher, den Geräte-Symlinks aus `/sys/class/hwmon/hwmon*` zu folgen, wobei * eine reelle Zahl ist (0,1,2,...).

Die Sensorwerte befinden sich entweder im Verzeichnis `/sys/class/hwmon/hwmon*/` für virtuelle Geräte oder im Verzeichnis `/sys/class/hwmon/hwmon*/device` für nicht-virtuelle Geräte. Eine Datei mit dem Namen `name`, die sich in den Verzeichnissen `hwmon*` oder `hwmon*/device` befindet, enthält den Namen des Chips, der dem Namen des vom Sensorchip verwendeten Kernel-Treibers entspricht.

Es gibt nur einen Sensorwert pro Datei. Das allgemeine Schema für die Benennung der Dateien, die Sensorwerte in einem der oben genannten Verzeichnisse enthalten, lautet: `<type><number>_<item>`, wobei

- **type** - bei Sensorchips ist dies "in" (Spannung), "temp" (Temperatur), "fan" (Lüfter) usw.
- **item** - "input" (gemessener Wert), "max" (oberer Schwellenwert), "min" (unterer Schwellenwert) usw.
- **number** - wird immer für Elemente verwendet, die mehr als einmal vorhanden sein können (beginnt normalerweise bei 1, außer bei Spannungen, die bei 0 beginnen). Wenn sich Dateien nicht auf ein bestimmtes Element beziehen, haben sie einen einfachen Namen ohne Nummer.

Die Informationen zu den auf dem Host verfügbaren Sensoren können mit den Werkzeugen **sensors-detect** und **sensors** ermittelt werden (lm-sensors package). **Sensors-detect** hilft dabei festzustellen, welche Module für die verfügbaren Sensoren erforderlich sind. Wenn die Module geladen sind, kann das Programm **sensors** verwendet werden, um die Werte aller Sensorchips anzuzeigen. Die von diesem Programm verwendete Bezeichnung der Sensorwerte kann vom allgemeinen Benennungsschema (<type><number>_<item>) abweichen:

- wenn es eine Datei namens <type><number>_label gibt, wird die Bezeichnung in dieser Datei anstelle des Namens <type><number><item> verwendet;
- wenn es keine Datei <type><number>_label gibt, sucht das Programm in /etc/sensors.conf (kann auch /etc/sensors3.conf oder eine andere Datei sein) nach einer Namensersetzung.

Diese Bezeichnung ermöglicht es dem Benutzer festzustellen, welche Art von Hardware verwendet wird. Wenn es weder eine Datei <type><number>_label noch eine Bezeichnung in der Konfigurationsdatei gibt, kann der Hardwaretyp anhand des Namensattributs (hwmon*/device/name) bestimmt werden. Die tatsächlichen Sensornamen, die zabbix_agent akzeptiert, können durch Ausführen des Programms **sensors** mit dem Parameter -u (sensors -u) ermittelt werden.

Im Programm **sensors** werden die verfügbaren Sensoren nach Bustyp getrennt (ISA-Adapter, PCI-Adapter, SPI-Adapter, virtuelles Gerät, ACPI-Schnittstelle, HID-Adapter).

Ermitteln von Sensor-IDs

Sensorbezeichnungen, wie sie vom Befehl sensors ausgegeben werden, können nicht immer direkt verwendet werden, da die Bezeichnung je nach Chip-Hersteller variiert. Zum Beispiel könnte die Ausgabe von sensors die folgenden Zeilen enthalten:

```
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)
```

Davon entspricht nur ein Anzeigename direkt einer internen ID:

```
zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,in0]
2.240000
```

Der Versuch, andere angezeigte Bezeichnungen zu verwenden (wie Vcore oder +12V), funktioniert nicht:

```
zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,Vcore]
ZBX_NOTSUPPORTED
```

Um die interne Sensor-ID zu finden, die Zabbix akzeptiert, führen Sie sensors -u aus. In der Ausgabe kann Folgendes zu sehen sein:

```
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
...
```

Die Bezeichnung Vcore entspricht also der ID in1, und +12V entspricht in4. Gemäß der Kernel-hwmon-sysfs-Spezifikation sind dies Spannungen an Chip-Pins und müssen allgemein gesprochen möglicherweise skaliert werden.

```
zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,in1]
1.301000
```

Nicht nur Spannungs- (in), sondern auch Strom- (curr), Temperatur- (temp) und Lüfterdrehzahlwerte (fan) können von Zabbix abgerufen werden.

Kompatibilität

Veraltete Distributionen

Sensorwerte werden aus dem Verzeichnis `/proc/sys/dev/sensors` abgerufen:

- **device** - Gerätename (wenn <mode> verwendet wird, ist dies ein regulärer Ausdruck)
- **sensor** - Sensorname (wenn <mode> verwendet wird, ist dies ein regulärer Ausdruck)
- **mode** - mögliche Werte: `avg`, `max`, `min` (wenn dieser Parameter weggelassen wird, werden device und sensor wörtlich behandelt)

Beispielschlüssel: `sensor[w83781d-i2c-0-2d,temp1]`.

Aktuelle Distributionen

Sensorwerte werden aus dem Verzeichnis `/sys/class/hwmon` abgerufen:

- **device** - Gerätename (kein regulärer Ausdruck). Der Gerätename kann der tatsächliche Name des Geräts sein (z. B. `0000:00:18.3`) oder der mit dem Programm sensors ermittelte Name (z. B. `k8temp-pci-00c3`). Es liegt am Benutzer, zu entscheiden, welcher Name verwendet werden soll.
- **sensor** - Sensorname (kein regulärer Ausdruck).
- **mode** - mögliche Werte: `avg`, `max`, `min` (wenn dieser Parameter weggelassen wird, werden device und sensor wörtlich behandelt).

Beispielschlüssel:

`sensor[k8temp-pci-00c3,temp,max]` oder `sensor[0000:00:18.3,temp1]`.

`sensor[smsc47b397-isa-0880,in,avg]` oder `sensor[smsc47b397.2176,in1]`.

Hinweise zum memtype-Parameter in proc.mem-Datenpunkten

Übersicht

Der Parameter **memtype** wird auf Linux-, AIX-, FreeBSD- und Solaris-Plattformen unterstützt.

Auf allen diesen Plattformen werden drei gängige Werte von `memtype` unterstützt: `pmem`, `rss` und `vsize`. Zusätzlich werden auf einigen Plattformen plattformspezifische `memtype`-Werte unterstützt.

AIX

In der Tabelle finden Sie die für den Parameter `memtype` unter AIX unterstützten Werte.

Unterstützter Wert	Beschreibung	Quelle in der Struktur procentry64	Versucht, kompatibel zu sein mit
`vsize` [1]	Größe des virtuellen Speichers	`pi_size`	
`pmem`	Prozentsatz des realen Speichers	`pi_prm`	`ps -o pmem`
`rss`	Größe des residenten Speichers	`pi_trss + pi_drss`	`ps -o rssize`
`size`	Größe des Prozesses (Code + Daten)	`pi_dvm`	Spalte SIZE von `ps gvw`
`dsize`	Datengröße	`pi_dsize`	
`tsize`	Textgröße (Code)	`pi_tsize`	Spalte TSIZ von `ps gvw`
`sdsize`	Datengröße aus der gemeinsam genutzten Bibliothek	`pi_sdsize`	
`drss`	Residente Datensatzgröße	`pi_drss`	
`trss`	Residente Textgröße	`pi_trss`	

Hinweise für AIX:

1. Versuchen Sie bei der Auswahl der Parameter für den `proc.mem[]`-Datenpunktschlüssel unter AIX, möglichst genaue Kriterien für die Prozessauswahl anzugeben. Andernfalls besteht das Risiko, dass unerwünschte Prozesse im Ergebnis von `proc.mem[]` mitgezählt werden.

Beispiel:

```
$ zabbix_agentd -t proc.mem[,,,NonExistingProcess,rss]
proc.mem[,,,NonExistingProcess,rss]           [u|2879488]
```

Dieses Beispiel zeigt, dass die Angabe ausschließlich des Parameters für die Befehlszeile (abzugleichender regulärer Ausdruck) dazu führt, dass der Zabbix Agent sich selbst mitzählt. Das ist wahrscheinlich nicht das gewünschte Verhalten.

2. Verwenden Sie `ps -ef` nicht zum Anzeigen von Prozessen – dieser Befehl zeigt nur Nicht-Kernel-Prozesse an. Verwenden Sie `ps -Af`, um alle Prozesse anzuzeigen, die vom Zabbix Agent erkannt werden.

3. Sehen wir uns am Beispiel von `topasrec` an, wie der Zabbix Agent `proc.mem[]` Prozesse auswählt.

```
$ 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[]` verfügt über folgende Argumente:

`proc.mem[<name>,<user>,<mode>,<cmdline>,<memtype>]`

wobei: - Das erste Kriterium ist ein Prozessname (Argument `<name>`). In unserem Beispiel wird der Zabbix Agent ihn als `topasrec` erkennen. Für eine Übereinstimmung müssen Sie entweder `topasrec` angeben oder das Feld leer lassen. - Das zweite Kriterium ist ein Benutzername (Argument `<user>`). Für eine Übereinstimmung müssen Sie entweder `root` angeben oder das Feld leer lassen. - Das dritte Kriterium, das bei der Prozessauswahl verwendet wird, ist ein Argument `<cmdline>`. Der Zabbix Agent wird dessen Wert als `/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` erkennen. Für eine Übereinstimmung müssen Sie entweder einen regulären Ausdruck angeben, der auf diese Zeichenfolge passt, oder das Feld leer lassen.

Die Argumente `<mode>` und `<memtype>` werden angewendet, nachdem die drei oben genannten Kriterien verwendet wurden.

FreeBSD

In der Tabelle finden Sie die für den Parameter `memtype` unter FreeBSD unterstützten Werte.

Unterstützter Wert	Beschreibung	Quelle in der Struktur kinfo_proc	Versucht, kompatibel zu sein mit
`vsize`	Größe des virtuellen Speichers	`kp_eproc.e_vm.vm_map_size` oder `ki_size`	`ps -o vsz`
`pmem`	Prozentsatz des realen Speichers	aus `rss` berechnet	`ps -o pmem`
`rss`	Größe des residenten Speichers	`kp_eproc.e_vm.vm_rssize` oder `ki_rssize`	`ps -o rss`
`size` [1]	Größe des Prozesses (Code + Daten + Stack)	`tsize` + `dsize` + `ssize`	
`tsize`	Größe des Textes (Code)	`kp_eproc.e_vm.vm_tsize` oder `ki_tsize`	`ps -o tsiz`
`dsize`	Datengröße	`kp_eproc.e_vm.vm_dsize` oder `ki_dsize`	`ps -o dsiz`
`ssize`	Stackgröße	`kp_eproc.e_vm.vm_ssize` oder `ki_ssize`	`ps -o ssiz`

Linux

Die für den Parameter `memtype` unter Linux unterstützten Werte finden Sie in der Tabelle.

Unterstützter Wert	Beschreibung	Quelle in der Datei /proc/<pid>/status
`vsize` [1]	Größe des virtuellen Speichers	`VmSize`
`pmem`	Prozentsatz des realen Speichers	(`VmRSS`/`total_memory`) * 100
`rss`	Größe des residenten Speichers	`VmRSS`
`data`	Größe des Datensegments	`VmData`
`exe`	Größe des Codesegments	`VmExe`
`hwm`	Maximale Größe des residenten Speichers	`VmHWM`
`lck`	Größe des gesperrten Speichers	`VmLck`
`lib`	Größe der gemeinsam genutzten Bibliotheken	`VmLib`
`peak`	Maximale Größe des virtuellen Speichers	`VmPeak`
`pin`	Größe der angehefteten Seiten	`VmPin`
`pte`	Größe der Seitentabelleneinträge	`VmPTE`
`size`	Größe der Code-, Daten- und Stacksegmente des Prozesses	`VmExe` + `VmData` + `VmStk`
`stk`	Größe des Stacksegments	`VmStk`
`swap`	Größe des verwendeten Swap-Speichers	`VmSwap`

Hinweise für Linux:

1. Nicht alle memtype-Werte werden von älteren Linux-Kerneln unterstützt. Beispielsweise unterstützen Linux-2.4-Kernel die Werte hwm, pin, peak, pte und swap nicht.
2. Wir haben festgestellt, dass die Selbstüberwachung des aktiven Prüfprozesses des Zabbix Agent mit proc.mem[...,...,...,...,da einen Wert anzeigt, der 4 kB größer ist als der in der Zeile VmData der Datei /proc/<pid>/status des Agent gemeldete Wert. Zum Zeitpunkt der Selbstmessung vergrößert sich das Datensegment des Agent um 4 kB und kehrt anschließend zu seiner vorherigen Größe zurück.

Solaris

In der Tabelle finden Sie die für den Parameter memtype unter Solaris unterstützten Werte.

Unterstützter Wert	Beschreibung	Quelle in der psinfo-Struktur	Versucht, kompatibel zu sein mit
vsize [1]	Größe des Prozessabbilds	pr_size	ps -o vsz
pmem	Prozentsatz des realen Speichers	pr_pctmem	ps -o pmem
rss	Größe des residenten Satzes Sie kann unterschätzt werden (siehe die Beschreibung von rss in man ps).	pr_rssize	ps -o rss

Fußnoten

[1] Standardwert.

Hinweise zur Auswahl von Prozessen in proc.mem- und proc.num-Datenpunkten

Prozesse, die ihre Befehlszeile verändern

Einige Programme verwenden das Verändern ihrer Befehlszeile als Methode, um ihre aktuelle Aktivität anzuzeigen. Ein Benutzer kann die Aktivität sehen, indem er die Befehle ps und top ausführt. Beispiele für solche Programme sind *PostgreSQL*, *Sendmail*, *Zabbix*.

Sehen wir uns ein Beispiel unter Linux an. Nehmen wir an, wir möchten eine Anzahl von Zabbix-Agent-Prozessen überwachen.

Der Befehl ps zeigt die relevanten Prozesse wie folgt an:

```
$ 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]
...
```

Die Auswahl von Prozessen nach Name und Benutzer erfüllt ihren Zweck:

```
$ zabbix_get -s localhost -k 'proc.num[zabbix_agentd,zabbix]'
6
```

Benennen wir nun die ausführbare Datei zabbix_agentd in zabbix_agentd_30 um und starten sie neu.

ps zeigt nun Folgendes an:

```
$ 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]
...
```

Nun liefert die Auswahl von Prozessen nach Name und Benutzer ein falsches Ergebnis:

```
$ zabbix_get -s localhost -k 'proc.num[zabbix_agentd_30,zabbix]'
1
```

Warum führt ein einfaches Umbenennen der ausführbaren Datei in einen längeren Namen zu einem so anderen Ergebnis?

Der Zabbix-Agent beginnt mit der Prüfung des Prozessnamens. Die Datei `/proc/<pid>/status` wird geöffnet und die Zeile `Name` wird geprüft. In unserem Fall lauten die `Name`-Zeilen:

```
$ 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
```

Der Prozessname in der Datei `status` wird auf 15 Zeichen gekürzt.

Ein ähnliches Ergebnis ist mit dem Befehl `ps` zu sehen:

```
$ 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
 ...
```

Offensichtlich ist das nicht gleich unserem `name`-Parameterwert `zabbix_agentd_30` von `proc.num[]`. Nachdem der Zabbix-Agent den Prozessnamen aus der Datei `status` nicht abgleichen konnte, greift er auf die Datei `/proc/<pid>/cmdline` zurück.

Wie der Agent die Datei "cmdline" sieht, lässt sich durch Ausführen des folgenden Befehls veranschaulichen:

```
$ 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
```

Die Dateien `/proc/<pid>/cmdline` enthalten in unserem Fall unsichtbare, nicht druckbare Null-Bytes, die in der Sprache *C* zum Beenden von Zeichenketten verwendet werden. Die Null-Bytes werden in diesem Beispiel als "<NUL>" dargestellt.

Der Zabbix-Agent prüft "cmdline" für den Hauptprozess und entnimmt `zabbix_agentd_30`, was mit unserem `name`-Parameterwert `zabbix_agentd_30` übereinstimmt. Daher wird der Hauptprozess vom Datenpunkt `proc.num[zabbix_agentd_30,zabbi` gezählt.

Bei der Prüfung des nächsten Prozesses entnimmt der Agent `zabbix_agentd_30: collector [idle 1 sec]` aus der Datei `cmdline`, und dies entspricht nicht unserem `name`-Parameter `zabbix_agentd_30`. Daher wird nur der Hauptprozess gezählt, der seine Befehlszeile nicht verändert. Andere Agent-Prozesse verändern ihre Befehlszeile und werden ignoriert.

Dieses Beispiel zeigt, dass der Parameter `name` in diesem Fall nicht in `proc.mem[]` und `proc.num[]` zur Auswahl von Prozessen verwendet werden kann.

Note:
Beim Datenpunkt `proc.get[]` verwendet der Zabbix-Agent, wenn er "cmdline" auf den Prozessnamen prüft, nur den Teil des Namens ab dem letzten Schrägstrich bis zum ersten Leerzeichen oder Doppelpunkt. Der aus der Datei cmdline erhaltene Prozessname wird nur verwendet, wenn sein Anfang vollständig mit dem gekürzten Prozessnamen in der Datei `status` übereinstimmt. Der Algorithmus ist sowohl für den Prozessnamen im Filter als auch in der JSON-Ausgabe derselbe.

Die Verwendung des Parameters `cmdline` mit einem geeigneten regulären Ausdruck liefert ein korrektes Ergebnis:

```
$ zabbix_get -s localhost -k 'proc.num[,zabbix,,zabbix_agentd_30[ :]]'
6
```

Seien Sie vorsichtig bei der Verwendung von proc.get[]-, proc.mem[]- und proc.num[]-Datenpunkten zur Überwachung von Programmen, die ihre Befehlszeilen verändern.

Bevor Sie die Parameter name und cmdline in proc.get[]-, proc.mem[]- und proc.num[]-Datenpunkten verwenden, sollten Sie die Parameter möglicherweise mit dem Datenpunkt proc.num[] und dem Befehl ps testen.

Linux-Kernel-Threads

Threads können in proc.get[]-, proc.mem[]- und proc.num[]-Datenpunkten nicht mit dem Parameter cmdline ausgewählt werden

Nehmen wir als Beispiel einen der Kernel-Threads:

```
$ ps -ef| grep kthreadd
root         2     0  0 09:33 ?        00:00:00 [kthreadd]
```

Er kann mit dem Prozessparameter name ausgewählt werden:

```
$ zabbix_get -s localhost -k 'proc.num[kthreadd,root]'
1
```

Die Auswahl über den Prozessparameter cmdline funktioniert jedoch nicht:

```
$ zabbix_get -s localhost -k 'proc.num[,root,,kthreadd]'
0
```

Der Grund dafür ist, dass der Zabbix Agent den im Parameter cmdline angegebenen regulären Ausdruck verwendet und ihn auf den Inhalt von /proc/<pid>/cmdline des Prozesses anwendet. Bei Kernel-Threads sind ihre Dateien /proc/<pid>/cmdline leer. Daher liefert der Parameter cmdline niemals eine Übereinstimmung.

Zählung von Threads in proc.mem[]- und proc.num[]-Datenpunkten

Linux-Kernel-Threads werden vom proc.num[]-Datenpunkt gezählt, melden jedoch keinen Speicher im proc.mem[]-Datenpunkt. Zum Beispiel:

```
$ 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.
```

Aber was passiert, wenn es einen Benutzerprozess mit demselben Namen wie ein Kernel-Thread gibt? Dann könnte es so aussehen:

```
$ 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[] zählte sowohl den Kernel-Thread als auch den Benutzerprozess. proc.mem[] meldet Speicher nur für den Benutzerprozess und zählt den Speicher des Kernel-Threads so, als wäre er 0. Dies unterscheidet sich von dem oben beschriebenen Fall, in dem ZBX_NOTSUPPORTED gemeldet wurde.

Seien Sie vorsichtig bei der Verwendung von proc.mem[]- und proc.num[]-Datenpunkten, wenn der Programmname zufällig mit einem der Threads übereinstimmt.

Bevor Sie Parameter in proc.mem[]- und proc.num[]-Datenpunkte eintragen, sollten Sie die Parameter möglicherweise mit dem proc.num[]-Datenpunkt und dem Befehl ps testen.

Implementierungsdetails der Prüfungen net.tcp.service und net.udp.service

Die Implementierung der Prüfungen net.tcp.service und net.udp.service wird auf dieser Seite für verschiedene Dienste beschrieben, die im Parameter service angegeben sind.

Parameter des Datenpunkts net.tcp.service

ftp

Stellt eine TCP-Verbindung her und erwartet, dass die ersten 4 Zeichen der Antwort ”220 ” sind, und sendet dann ”QUIT\r\n”. Standardmäßig wird Port 21 verwendet, wenn kein Port angegeben ist.

http

Stellt eine TCP-Verbindung her, ohne etwas zu erwarten oder zu senden. Standardmäßig wird Port 80 verwendet, wenn kein Port angegeben ist.

https

Verwendet libcurl (und funktioniert nur damit), überprüft nicht die Authentizität des Zertifikats, überprüft nicht den Host-Namen im SSL-Zertifikat und ruft nur den Antwort-Header ab (HEAD-Anfrage). Standardmäßig wird Port 443 verwendet, wenn kein Port angegeben ist.

imap

Stellt eine TCP-Verbindung her und erwartet, dass die ersten 4 Zeichen der Antwort ”* OK” sind, und sendet dann ”a1 LOGOUT\r\n”. Standardmäßig wird Port 143 verwendet, wenn kein Port angegeben ist.

ldap

Öffnet eine Verbindung zu einem LDAP-Server und führt eine LDAP-Suchoperation mit dem Filter (objectClass=*) aus. Erwartet das erfolgreiche Abrufen des ersten Attributs des ersten Eintrags. Standardmäßig wird Port 389 verwendet, wenn kein Port angegeben ist.

nntp

Stellt eine TCP-Verbindung her und erwartet, dass die ersten 3 Zeichen der Antwort ”200” oder ”201” sind, und sendet dann ”QUIT\r\n”. Standardmäßig wird Port 119 verwendet, wenn kein Port angegeben ist.

pop

Stellt eine TCP-Verbindung her und erwartet, dass die ersten 3 Zeichen der Antwort ”+OK” sind, und sendet dann ”QUIT\r\n”. Standardmäßig wird Port 110 verwendet, wenn kein Port angegeben ist.

smtp

Stellt eine TCP-Verbindung her und erwartet, dass die ersten 3 Zeichen der Antwort ”220” sind, gefolgt von einem Leerzeichen, einem Zeilenende oder einem Bindestrich. Die Zeilen mit einem Bindestrich gehören zu einer mehrzeiligen Antwort, und die Antwort wird erneut gelesen, bis eine Zeile ohne Bindestrich empfangen wird. Dann wird ”QUIT\r\n” gesendet. Standardmäßig wird Port 25 verwendet, wenn kein Port angegeben ist.

ssh

Stellt eine TCP-Verbindung her. Wenn die Verbindung hergestellt wurde, tauschen beide Seiten eine Identifikationszeichenfolge aus (SSH-major.minor-XXXX), wobei major und minor Protokollversionen sind und XXXX eine Zeichenfolge ist. Zabbix prüft, ob eine der Spezifikation entsprechende Zeichenfolge gefunden wird, und sendet dann die Zeichenfolge ”SSH-major.minor-zabbix_agent\r\n” zurück oder bei Nichtübereinstimmung ”0\n”. Standardmäßig wird Port 22 verwendet, wenn kein Port angegeben ist.

tcp

Stellt eine TCP-Verbindung her, ohne etwas zu erwarten oder zu senden. Im Gegensatz zu den anderen Prüfungen muss hier der Port-Parameter angegeben werden.

telnet

Stellt eine TCP-Verbindung her und erwartet eine Anmeldeaufforderung (’:’ am Ende). Standardmäßig wird Port 23 verwendet, wenn kein Port angegeben ist.

Datenpunkt-Parameter net.udp.service

ntp

Sendet ein SNTP-Paket über UDP und validiert die Antwort gemäß RFC 4330, Abschnitt 5. Standardmäßig wird Port 123 verwendet, wenn kein Port angegeben ist.

proc.get-Parameter

Übersicht

Der Datenpunkt **proc.get**[<name>,<user>,<cmdline>,<mode>] wird unter Linux, Windows, FreeBSD, OpenBSD und NetBSD unterstützt.

Die Liste der vom Datenpunkt zurückgegebenen Prozessparameter variiert je nach Betriebssystem und dem Wert des Arguments „mode“.

Linux

Die folgenden Prozessparameter werden unter Linux für jeden Modus zurückgegeben:

mode=process	mode=thread	mode=summary
pid: PID	pid: PID	name: Prozessname
ppid: übergeordnete PID	ppid: übergeordnete PID	processes: Anzahl der Prozesse
name: Prozessname	name: Prozessname	vsize: Größe des virtuellen Speichers
cmdline: Befehl mit Argumenten	user: Benutzer (real), unter dem der Prozess ausgeführt wird	pmem: Prozentsatz des realen Speichers
user: Benutzer (real), unter dem der Prozess ausgeführt wird	group: Gruppe (real), unter der der Prozess ausgeführt wird	rss: Größe des residenten Speichers
group: Gruppe (real), unter der der Prozess ausgeführt wird	uid: Benutzer-ID	data: Größe des Datensegments
uid: Benutzer-ID	gid: ID der Gruppe, unter der der Prozess ausgeführt wird	exe: Größe des Codesegments
gid: ID der Gruppe, unter der der Prozess ausgeführt wird	tid: Thread-ID	lib: Größe der gemeinsam genutzten Bibliotheken
vsize: Größe des virtuellen Speichers	tname: Threadname	lck: Größe des gesperrten Speichers
pmem: Prozentsatz des realen Speichers	cputime_user: gesamte im Benutzermodus verbrachte CPU-Zeit (Wert in Taktzyklen ausgedrückt – durch `getconf CLK_TCK` oder `sysconf(_SC_CLK_TCK)` teilen, um Sekunden zu erhalten)	pin: Größe der angehefteten Seiten
rss: Größe des residenten Speichers	cputime_system: gesamte im Systemmodus verbrachte CPU-Zeit (Wert in Taktzyklen ausgedrückt – durch `getconf CLK_TCK` oder `sysconf(_SC_CLK_TCK)` teilen, um Sekunden zu erhalten)	pte: Größe der Seitentabelleneinträge
data: Größe des Datensegments	state: Threadstatus	size: Größe der Code-, Daten- und Stacksegmente des Prozesses
exe: Größe des Codesegments	ctx_switches: Anzahl der Kontextwechsel	stk: Größe des Stacksegments
hwm: maximale Größe des residenten Speichers	page_faults: Anzahl der Seitenfehler	swap: Größe des verwendeten Swap-Speichers
lck: Größe des gesperrten Speichers		cputime_user: gesamte im Benutzermodus verbrachte CPU-Zeit (Wert in Taktzyklen ausgedrückt – durch `getconf CLK_TCK` oder `sysconf(_SC_CLK_TCK)` teilen, um Sekunden zu erhalten)
lib: Größe der gemeinsam genutzten Bibliotheken		cputime_system: gesamte im Systemmodus verbrachte CPU-Zeit (Wert in Taktzyklen ausgedrückt – durch `getconf CLK_TCK` oder `sysconf(_SC_CLK_TCK)` teilen, um Sekunden zu erhalten)
peak: maximale Größe des virtuellen Speichers		ctx_switches: Anzahl der Kontextwechsel
pin: Größe der angehefteten Seiten		threads: Anzahl der Threads
pte: Größe der Seitentabelleneinträge		page_faults: Anzahl der Seitenfehler
size: Größe der Code-, Daten- und Stacksegmente des Prozesses		pss: proportionaler Speicheranteil
stk: Größe des Stacksegments		
swap: Größe des verwendeten Swap-Speichers		

mode=process	mode=thread	mode=summary
cputime_user: gesamte im Benutzermodus verbrachte CPU-Zeit (Wert in Taktzyklen ausgedrückt – durch `getconf CLK_TCK` oder `sysconf(_SC_CLK_TCK)` teilen, um Sekunden zu erhalten) cputime_system: gesamte im Systemmodus verbrachte CPU-Zeit (Wert in Taktzyklen ausgedrückt – durch `getconf CLK_TCK` oder `sysconf(_SC_CLK_TCK)` teilen, um Sekunden zu erhalten) state: Prozessstatus (transparent aus procfs abgerufen, Langform) ctx_switches: Anzahl der Kontextwechsel threads: Anzahl der Threads page_faults: Anzahl der Seitenfehler pss: proportionaler Speicheranteil		

BSD-basierte Betriebssysteme

Die folgenden Prozessparameter werden unter FreeBSD, OpenBSD und NetBSD für jeden Modus zurückgegeben:

mode=process	mode=thread	mode=summary
pid: PID	pid: PID	name: Prozessname
ppid: übergeordnete PID	ppid: übergeordnete PID	processes: Anzahl der Prozesse
jid: ID der Jail (nur FreeBSD)	jid: ID der Jail (nur FreeBSD)	vsize: Größe des virtuellen Speichers
jname: Name der Jail (nur FreeBSD)	jname: Name der Jail (nur FreeBSD)	pmem: Prozentsatz des realen Speichers (nur FreeBSD)
name: Prozessname	name: Prozessname	rss: Resident Set Size
cmdline: Befehl mit Argumenten	user: Benutzer (real), unter dem der Prozess läuft	size: Größe des Prozesses (Code + Daten + Stack)
user: Benutzer (real), unter dem der Prozess läuft	group: Gruppe (real), unter der der Prozess läuft	tsize: Größe des Textsegments (Code)
group: Gruppe (real), unter der der Prozess läuft	uid: Benutzer-ID	dsize: Datengröße
uid: Benutzer-ID	gid: ID der Gruppe, unter der der Prozess läuft	ssize: Stack-Größe
gid: ID der Gruppe, unter der der Prozess läuft	tid: Thread-ID	cputime_user: gesamte im Benutzermodus verbrachte CPU-Zeit (in Sekunden angegeben)
vsize: Größe des virtuellen Speichers	tname: Thread-Name	cputime_system: gesamte im Systemmodus verbrachte CPU-Zeit (in Sekunden angegeben)
pmem: Prozentsatz des realen Speichers (nur FreeBSD)	cputime_user: gesamte im Benutzermodus verbrachte CPU-Zeit (in Sekunden angegeben)	ctx_switches: Anzahl der Kontextwechsel
rss: Resident Set Size	cputime_system: gesamte im Systemmodus verbrachte CPU-Zeit (in Sekunden angegeben)	threads: Anzahl der Threads (für NetBSD nicht unterstützt)
size: Größe des Prozesses (Code + Daten + Stack)	state: Thread-Status	stk: Größe des Stack-Segments
tsize: Größe des Textsegments (Code)	ctx_switches: Anzahl der Kontextwechsel	page_faults: Anzahl der Seitenfehler
dsize: Datengröße	io_read_op: Anzahl der Male, die das System eine Eingabe durchführen musste	fds: Anzahl der Dateideskriptoren (nur OpenBSD)

mode=process	mode=thread	mode=summary
ssize: Stack-Größe	io_write_op: Anzahl der Male, die das System eine Ausgabe durchführen musste	swap: Größe des verwendeten Swap-Speichers
cputime_user: gesamte im Benutzermodus verbrachte CPU-Zeit (in Sekunden angegeben)		io_read_op: Anzahl der Male, die das System eine Eingabe durchführen musste
cputime_system: gesamte im Systemmodus verbrachte CPU-Zeit (in Sekunden angegeben)		io_write_op: Anzahl der Male, die das System eine Ausgabe durchführen musste
state: Prozessstatus (disk sleep/running/sleeping/tracing stop/zombie/other)		
ctx_switches: Anzahl der Kontextwechsel		
threads: Anzahl der Threads (für NetBSD nicht unterstützt)		
page_faults: Anzahl der Seitenfehler		
fds: Anzahl der Dateideskriptoren (nur OpenBSD)		
swap: Größe des verwendeten Swap-Speichers		
io_read_op: Anzahl der Male, die das System eine Eingabe durchführen musste		
io_write_op: Anzahl der Male, die das System eine Ausgabe durchführen musste		

Windows

Die folgenden Prozessparameter werden unter Windows für jeden Modus zurückgegeben:

mode=process	mode=thread	mode=summary
pid: PID	pid: PID	name: Prozessname
ppid: übergeordnete PID	ppid: übergeordnete PID	processes: Anzahl der Prozesse
name: Prozessname	name: Prozessname	vmsize: Größe des virtuellen Speichers
user: Benutzer, unter dem der Prozess ausgeführt wird	user: Benutzer, unter dem der Prozess ausgeführt wird	wkset: Größe des Working Sets des Prozesses
sid: Benutzer-SID	sid: Benutzer-SID	cputime_user: gesamte im Benutzermodus verbrachte CPU-Zeit (in Millisekunden angegeben)
vmsize: Größe des virtuellen Speichers	tid: Thread-ID	cputime_system: gesamte im Systemmodus verbrachte CPU-Zeit (in Millisekunden angegeben)
wkset: Größe des Working Sets des Prozesses		threads: Anzahl der Threads
cputime_user: gesamte im Benutzermodus verbrachte CPU-Zeit (in Millisekunden angegeben)		page_faults: Anzahl der Seitenfehler
cputime_system: gesamte im Systemmodus verbrachte CPU-Zeit (in Millisekunden angegeben)		handles: Anzahl der Handles
threads: Anzahl der Threads		io_read_b: gelesene IO-Bytes
page_faults: Anzahl der Seitenfehler		io_write_b: geschriebene IO-Bytes
handles: Anzahl der Handles		io_read_op: IO-Leseoperationen
io_read_b: gelesene IO-Bytes		io_write_op: IO-Schreiboperationen
io_write_b: geschriebene IO-Bytes		io_other_b: übertragene IO-Bytes, außer Lese- und Schreiboperationen
io_read_op: IO-Leseoperationen		io_other_op: IO-Operationen, außer Lese- und Schreiboperationen

mode=process	mode=thread	mode=summary
io_write_op: IO-Schreiboperationen io_other_b: übertragene IO-Bytes, außer Lese- und Schreiboperationen io_other_op: IO-Operationen, außer Lese- und Schreiboperationen		

Einstellungen für nicht erreichbare/nicht verfügbare Host-Schnittstellen

Übersicht

Mehrere Konfigurations-Parameter legen fest, wie sich der Zabbix-Server verhalten soll, wenn eine Agent-Prüfung (Zabbix, SNMP, IPMI, JMX) fehlschlägt und eine Host-Schnittstelle nicht mehr erreichbar ist.

Nicht erreichbare Schnittstelle

Eine Host-Schnittstelle wird von Zabbix nach einer fehlgeschlagenen Prüfung (Netzwerkfehler, Zeitüberschreitung) durch Zabbix-, SNMP-, IPMI- oder JMX-Agents als nicht erreichbar behandelt. Seit Zabbix 6.2.0 wirken sich auch aktive Zabbix-Agent-Prüfungen auf die Verfügbarkeit der Schnittstelle aus. Wenn aktive Prüfungen nicht mehr verfügbar sind, tragen sie zum gesamten Verfügbarkeitsstatus der Agent-Schnittstelle bei.

Ab dem Zeitpunkt, an dem eine Schnittstelle nicht erreichbar wird, legt **UnreachableDelay** fest, wie oft sie mithilfe eines der Datenpunkte (einschließlich LLD-Regeln) erneut geprüft wird. Diese erneuten Prüfungen werden von Unreachable-Pollern (oder IPMI-Pollern für IPMI-Prüfungen) durchgeführt. Standardmäßig beträgt das Intervall zwischen aufeinanderfolgenden Erreichbarkeitsprüfungen 15 Sekunden.

Attention:
Prüfungen, die von asynchronen Pollern durchgeführt werden, werden nicht zu Unreachable-Pollern verschoben.

Im Zabbix-Server-Log wird die Nichterreichbarkeit durch Meldungen wie die folgenden angezeigt:

```
Zabbix agent item "system.cpu.load[percpu,avg1]" on host "New host" failed: first network error, wait for
Zabbix agent item "system.cpu.load[percpu,avg15]" on host "New host" failed: another network error, wait f
```

Die Logmeldungen geben den genauen fehlgeschlagenen Datenpunkt und seinen Typ (Zabbix agent) an.

Note:
Der Parameter *Timeout* beeinflusst ebenfalls, wie früh eine Schnittstelle während der Nichterreichbarkeit erneut geprüft wird. Wenn Timeout auf 20 Sekunden und UnreachableDelay auf 30 Sekunden gesetzt ist, erfolgt die nächste Prüfung 50 Sekunden nach dem ersten Versuch.

Der Parameter **UnreachablePeriod** definiert die Gesamtdauer der Phase der Nichterreichbarkeit. Standardmäßig beträgt UnreachablePeriod 45 Sekunden. Dieser Wert sollte um ein Vielfaches größer als UnreachableDelay sein, damit sichergestellt ist, dass eine Schnittstelle mehrfach erneut geprüft wird, bevor sie als nicht verfügbar markiert wird.

Ein interner Datenpunkt, zabbix[host,active_agent,available], ermöglicht die Überwachung der Verfügbarkeit aktiver Prüfungen in Szenarien mit Nichterreichbarkeit.

Schnittstelle wieder auf verfügbar setzen

Wenn der Nichterreichbarkeitszeitraum vorbei ist, wird die Schnittstelle erneut abgefragt, wobei die Priorität für den Datenpunkt verringert wird, der die Schnittstelle in den Zustand „nicht erreichbar“ versetzt hat. Wenn die nicht erreichbare Schnittstelle wieder erscheint, kehrt die Überwachung automatisch zum Normalzustand zurück:

```
Fortsetzen der Zabbix-Agent-Prüfungen auf Host "New host": Verbindung wiederhergestellt
```

Note:
Sobald die Schnittstelle verfügbar wird, fragt der Host nicht sofort alle seine Datenpunkte ab, und zwar aus zwei Gründen:
- Dies könnte den Host überlasten.
- Der Zeitpunkt der Wiederherstellung der Schnittstelle stimmt nicht immer mit dem geplanten Zeitpunkt des Datenpunkt-Abfrageintervalls überein.

Daher werden Datenpunkte, nachdem die Schnittstelle verfügbar geworden ist, nicht sofort abgefragt, sondern für ihre nächste Abfragerunde neu eingeplant.

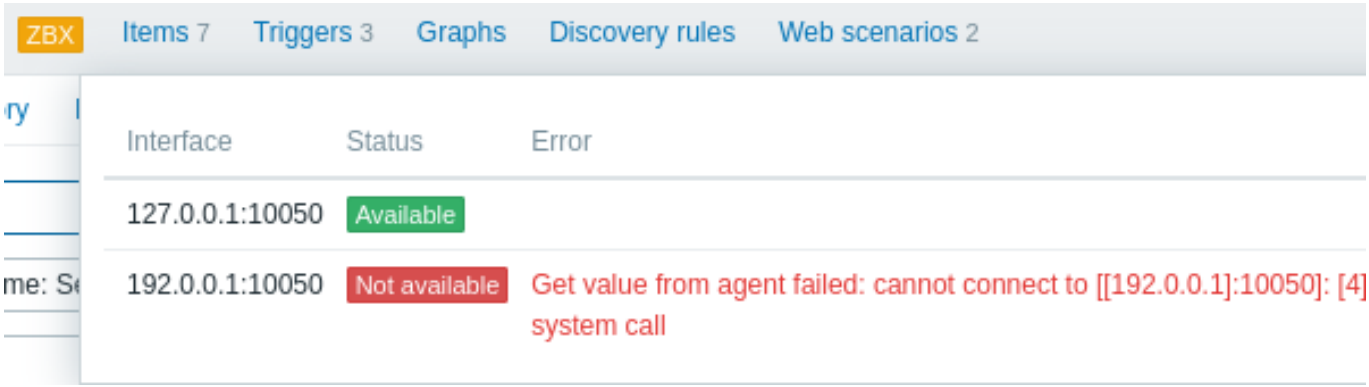
Nicht verfügbare Schnittstelle

Nachdem die UnreachablePeriod abgelaufen ist und die Schnittstelle nicht wieder erschienen ist, wird die Schnittstelle als nicht verfügbar behandelt.

Im Server-Log wird dies durch Meldungen wie diese angezeigt:

```
temporarily disabling Zabbix agent checks on host "New host": interface unavailable
```

und im Frontend wechselt das Verfügbarkeitssymbol des Hosts von grün/grau zu gelb/rot (die Details zur nicht verfügbaren Schnittstelle sind im Hinweisfeld zu sehen, das angezeigt wird, wenn sich der Mauszeiger auf dem Verfügbarkeitssymbol des Hosts befindet):



Der Parameter **UnavailableDelay** legt fest, wie oft eine Schnittstelle während ihrer Nichtverfügbarkeit geprüft wird.

Standardmäßig beträgt er 60 Sekunden (in diesem Fall bedeutet also „temporarily disabling“ aus der obigen Log-Meldung, dass Prüfungen für eine Minute deaktiviert werden).

Wenn die Verbindung zur Schnittstelle wiederhergestellt ist, kehrt auch die Überwachung automatisch in den Normalzustand zurück:

```
enabling Zabbix agent checks on host "New host": interface became available
```

Remote-Überwachung von Zabbix-Statistiken

Übersicht

Es ist möglich, einige interne Metriken des Zabbix-Servers und des Proxy für eine andere Zabbix-Instanz oder ein Drittanbieter-Tool aus der Ferne zugänglich zu machen. Dies kann nützlich sein, damit Supporter/Dienstleister die Zabbix-Server/Proxys ihrer Kunden aus der Ferne überwachen können oder damit in Organisationen, in denen Zabbix nicht das primäre Überwachungstool ist, die internen Zabbix-Metriken von einem Drittanbietersystem in einem übergeordneten Monitoring-Setup überwacht werden können.

Die internen Zabbix-Statistiken werden für einen konfigurierbaren Satz von Adressen bereitgestellt, die im neuen Parameter 'StatsAllowedIP' server/proxy aufgeführt sind. Anfragen werden nur von diesen Adressen akzeptiert.

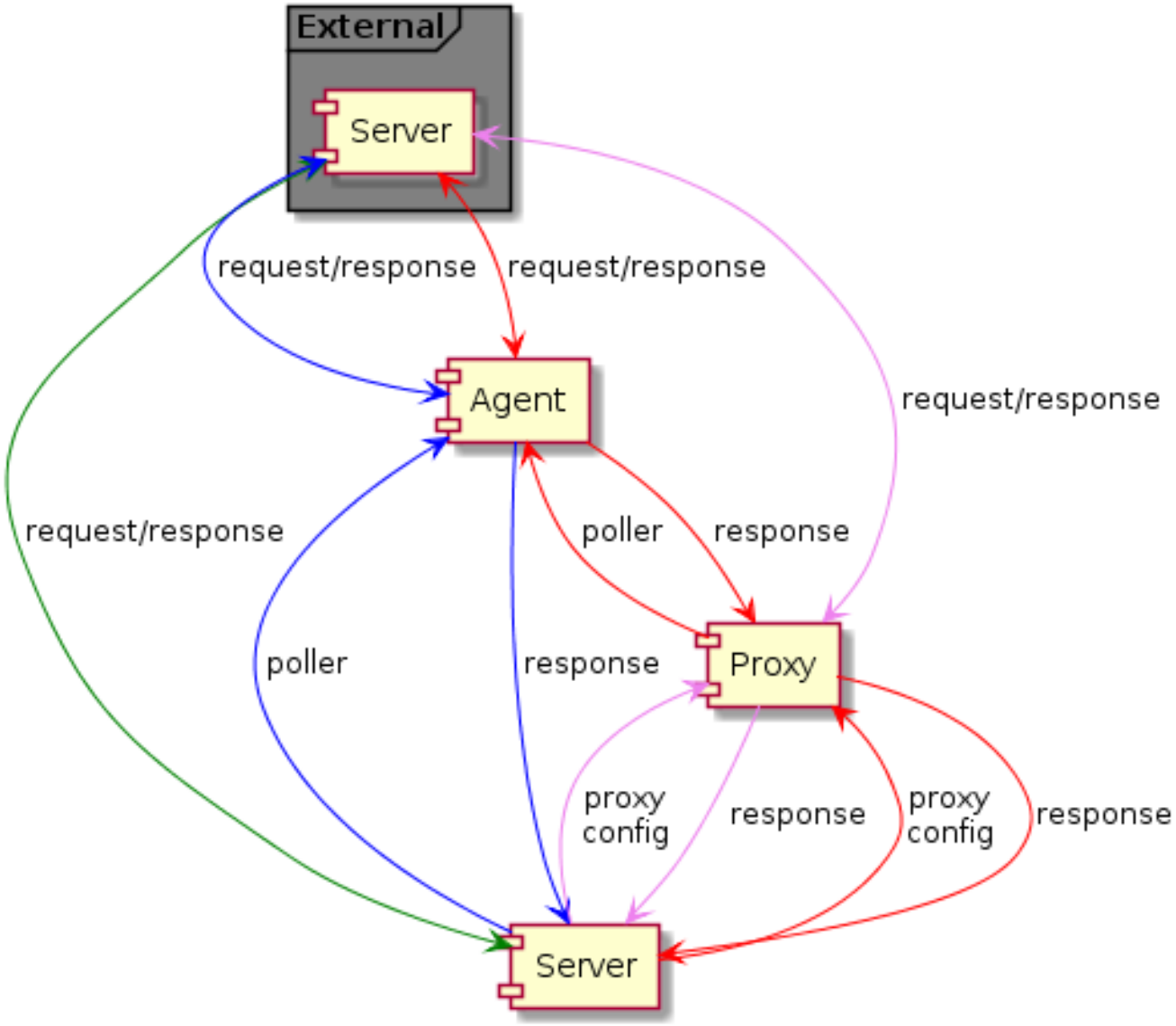
Datenpunkte

Um die Abfrage interner Statistiken auf einer anderen Zabbix-Instanz zu konfigurieren, können Sie zwei Datenpunkte verwenden:

- interner Datenpunkt `zabbix[stats,<ip>,<port>]` - für direkte Remote-Abfragen von Zabbix Server/Proxy. <ip> und <port> werden verwendet, um die Zielinstanz zu identifizieren.
- Agent-Datenpunkt `zabbix.stats[<ip>,<port>]` - für Agent-basierte Remote-Abfragen von Zabbix Server/Proxy. <ip> und <port> werden verwendet, um die Zielinstanz zu identifizieren.

Siehe auch: Interne Datenpunkte, Zabbix-Agent-Datenpunkte

Das folgende Diagramm veranschaulicht die Verwendung des jeweiligen Datenpunkts je nach Kontext.



- - Server → externe Zabbix-Instanz (`zabbix[stats,<ip>,<port>]`)
- - Server → Proxy → externe Zabbix-Instanz (`zabbix[stats,<ip>,<port>]`)
- - Server → Agent → externe Zabbix-Instanz (`zabbix.stats[<ip>,<port>]`)
- - Server → Proxy → Agent → externe Zabbix-Instanz (`zabbix.stats[<ip>,<port>]`)

Um sicherzustellen, dass die Zielinstanz Abfragen durch die externe Instanz zulässt, tragen Sie die Adresse der externen Instanz im Parameter 'StatsAllowedIP' auf der Zielinstanz ein.

Verfügbare Metriken

Die Statistik-Datenpunkte erfassen die Statistiken gesammelt und geben ein JSON zurück, das die Grundlage für abhängige Datenpunkte bildet, aus dem diese ihre Daten beziehen. Die folgenden internen Metriken werden von einem der beiden Datenpunkte zurückgegeben:

- `zabbix[boottime]`
- `zabbix[hosts]`
- `zabbix[items]`
- `zabbix[items_unsupported]`
- `zabbix[preprocessing]` (nur Server)
- `zabbix[preprocessing_queue]` (nur Server)
- `zabbix[process,<type>,<mode>,<state>]` (nur prozesstypbasierte Statistiken)
- `zabbix[rcache,<cache>,<mode>]`
- `zabbix[requiredperformance]`
- `zabbix[triggers]` (nur Server)
- `zabbix[uptime]`

- zabbix[vcache,buffer,<mode>] (nur Server)
- zabbix[vcache,cache,<parameter>]
- zabbix[version]
- zabbix[vmware,buffer,<mode>]
- zabbix[wcache,<cache>,<mode>] (Cache-Typ „trends“ nur Server)

Vorlagen

Vorlagen sind für die entfernte Überwachung interner Metriken von Zabbix Server oder Proxy aus einer externen Instanz verfügbar:

- Zustand des entfernten Zabbix Server
- Zustand des entfernten Zabbix Proxy

Beachten Sie, dass zur Verwendung einer Vorlage für die entfernte Überwachung mehrerer externer Instanzen für jede Überwachung einer externen Instanz ein separater Host erforderlich ist.

Trapper-Prozess

Der Empfang interner Metrikanfragen von einer externen Zabbix-Instanz wird vom Trapper-Prozess verarbeitet, der die Anfrage validiert, die Metriken sammelt, den JSON-Datenpuffer erstellt und das vorbereitete JSON zurücksendet, zum Beispiel vom Server:

```
{
  "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
      }
    }
  }
}
```

Interne Warteschlangen-Datenpunkte

Es gibt außerdem zwei weitere Datenpunkte, die speziell die Remote-Abfrage interner Warteschlangenstatistiken auf einer anderen

Zabbix-Instanz ermöglichen:

- `zabbix[stats,<ip>,<port>,queue,<from>,<to>]` interner Datenpunkt – für direkte interne Warteschlangenabfragen an einen entfernten Zabbix Server/Proxy
- `zabbix.stats[<ip>,<port>,queue,<from>,<to>]` Agent-Datenpunkt – für agentbasierte interne Warteschlangenabfragen an einen entfernten Zabbix Server/Proxy

Siehe auch: Interne Datenpunkte, Zabbix-Agent-Datenpunkte

Konfigurieren von Kerberos mit Zabbix

Übersicht

Kerberos-Authentifizierung kann in der Web-Überwachung und in HTTP-Datenpunkten in Zabbix verwendet werden.

Diese Seite beschreibt ein Beispiel für die Konfiguration von Kerberos für den Zabbix Server, damit dieser die Web-Überwachung von `www.example.com` mit einem Kerberos-Prinzipal für den Zabbix-Prozess unter Debian/Ubuntu durchführen kann.

Konfiguration

1. KDC und Client-Dienstprogramme installieren:

```
sudo apt update
sudo apt install krb5-kdc krb5-admin-server krb5-user
```

Beantworten Sie während der Paketinstallation die Eingabeaufforderungen, zum Beispiel:

```
Standard-Kerberos-Realm Version 5: EXAMPLE.COM
Kerberos-Server für Ihren Realm: localhost (oder Ihr FQDN)
Administrativer Server für Ihren Kerberos-Realm: localhost (oder Ihr FQDN)
```

2. Einen benutzerfreundlichen Hostnamen zuordnen (optional, für lokale Tests).

Bearbeiten Sie /etc/hosts und fügen Sie einen Eintrag für Ihren DC und Webserver hinzu, falls Sie kein DNS verwenden:

```
sudo vi /etc/hosts
```

Beispielzeile:

```
192.168.1.100  dc01.example.com dc01
```

3. Den Kerberos-Client und den KDC-Realm konfigurieren:

```
sudo vi /etc/krb5.conf
```

Beispieleinstellungen:

```
[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
```

Wenn Sie `.localdomain` oder andere nicht öffentliche Namen verwenden möchten, fügen Sie explizite Domain→Realm-Zuordnungen hinzu, damit die Zuordnung von Hostnamen zu Realms funktioniert. Nicht übereinstimmende Einträge verursachen Fehler vom Typ `Server not found in Kerberos database`.

4. Die Kerberos-Datenbank initialisieren (einmalig, auf dem KDC-Host). Legen Sie bei der Eingabeaufforderung ein sicheres Master-Passwort fest:

```
sudo krb5_newrealm
```

5. Den Principal `HTTP/host.fqdn@REALM` mit dem exakten Hostnamen erstellen, den die Clients verwenden; bevorzugt in Kleinbuchstaben (z. B. `HTTP/web.example.com@EXAMPLE.COM`). Eine Abweichung bei Groß-/Kleinschreibung oder Name verursacht den Fehler `Server not found in Kerberos database`.

```
sudo kadmin.local
```

In `kadmin.local`:

```
addprinc kerb-admin@EXAMPLE.COM      # administrativer Principal
addprinc -randkey HTTP/dc01.example.com@EXAMPLE.COM
ktadd -k /etc/apache2/http.keytab HTTP/dc01.example.com@EXAMPLE.COM
quit
```

Verschieben Sie die Keytab-Datei auf den Web-Host (oder belassen Sie sie lokal, wenn es sich um denselben Rechner handelt) und legen Sie Berechtigungen fest, die von Apache verwendet werden können:

```
chown www-data:www-data /etc/apache2/http.keytab
chmod 600 /etc/apache2/http.keytab
#### überprüfen
sudo -u www-data -k /etc/apache2/http.keytab
```

6. Das Apache-GSSAPI-Modul installieren und aktivieren:

```
sudo apt install libapache2-mod-auth-gssapi
sudo a2enmod auth_gssapi
sudo a2enmod headers
sudo systemctl restart apache2
```

> **Attention:**
> Nicht alle Versionen von mod_auth_gssapi unterstützen jede `Gssapi*`-Direktive. Wenn Apache mit `Invalid command 'GssapiCredStore'` fehlschlägt, entfernen Sie die nicht unterstützte Direktive oder aktualisieren Sie das Modul.

7. Einen VirtualHost konfigurieren (`DocumentRoot` bzw. den Pfad zu Ihrer Zabbix-Benutzeroberfläche anpassen):

```
sudo vi /etc/apache2/sites-available/zabbix.conf
```

In `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>
```

Apache neu starten:

```
sudo systemctl restart apache2
```

8. KDC-Dienste aktivieren und starten sowie die offenen Ports überprüfen (KDC-Host):

```
sudo systemctl enable --now krb5-kdc krb5-admin-server
ss -tnlp | grep :80    # oder: sudo netstat -tnlp | grep :80
```

9. Ein TGT zu Testzwecken abrufen (als der Benutzer ausführen, der das Ticket verwenden wird).

In der Ticketliste sollte krbtgt/EXAMPLE.COM@EXAMPLE.COM angezeigt werden. Führen Sie kinit als derselbe Betriebssystembenutzer aus, der das Ticket benötigt (z. B. zabbix für Webprüfungen oder www-data/Apache für interaktive Browser-SSO-Tests). Tickets, die einem anderen Betriebssystembenutzer ausgestellt wurden, sind nicht sichtbar, sofern KRB5CCNAME und die Berechtigungen nicht angepasst werden.

```
kinit kerb-admin@EXAMPLE.COM
klist
```

10. Den SPNEGO-Austausch mit curl testen (von einem Client mit einem gültigen TGT). Eine Antwort 200 OK (oder eine Weiterleitung zur Anwendung) zeigt an, dass SPNEGO erfolgreich war:

```
curl -v --negotiate -u : http://dc01.example.com/
```

11. Optional: Wenn die Zabbix-Benutzeroberfläche HTTP-authentifizierte Anmeldungen akzeptieren soll, aktivieren Sie die HTTP-Authentifizierung im Zabbix-Frontend.

Setzen Sie in der Frontend-Konfigurationsdatei (zabbix.conf.php):

```
$ZBX_FEATURE_FLAGS['http_auth_enabled'] = true;
```

Anschließend konfigurieren Sie die HTTP-Authentifizierung in der Weboberfläche.

12. Browserkonfiguration (Firefox dient als Beispiel): Setzen Sie network.negotiate-auth.trusted-uris auf den oder die Hosts, die Negotiate ausführen (dc01.example.com), damit der Browser Kerberos-Tokens automatisch sendet.

In about:config:

```
network.negotiate-auth.trusted-uris = dc01.example.com
```

Wenn Sie nun http://dc01.example.com aufrufen, sollten Sie direkt bei Zabbix angemeldet werden, ohne dass das Anmeldeformular angezeigt wird.

13. Schlüssel und Tickets aktuell halten. Die standardmäßige Lebensdauer eines Kerberos-Tickets beträgt ungefähr 10 Stunden. Fügen Sie einen Cron-Job oder einen systemd-Timer hinzu, um Ablaufzeiten zu vermeiden:

```
####für den Webdienst
kinit -kt /etc/apache2/http.keytab HTTP/dc01.example.com@EXAMPLE.COM
####für den Monitoring-Benutzer
kinit -kt /var/lib/zabbix/kerb.keytab kerb-admin@EXAMPLE.COM
```

14. Prüfungen zur Wartung:

- klist -k /etc/apache2/http.keytab - überprüfen, ob der Service-Principal in der Keytab-Datei vorhanden ist.
- sudo tail -f /var/log/apache2/error.log - auf GSSAPI-Fehler achten (gss_acquire_cred[_from]() failed to get server creds bedeutet, dass ein Problem mit der Keytab-Datei oder den Berechtigungen vorliegt oder der Principal fehlt).
- Wenn curl --negotiate den Status 401/403 zurückgibt, bedeutet dies häufig, dass der falsche Principal verwendet wird, kein Ticket vorhanden ist, der Host-Header nicht übereinstimmt oder ein Problem mit den Dateisystemberechtigungen vorliegt. Überprüfen Sie die Protokolle und die Domain-Zuordnungen in /etc/krb5.conf.

Hinweise zu Sicherheit und Dateiberechtigungen

Keytab-Dateien dürfen nur für das Konto lesbar sein, das sie benötigt. Beispielberechtigungen: 0400, Eigentümer zabbix:zabbix für eine Keytab-Datei des Benutzers zabbix oder 0440 und root:www-data für eine Apache-Keytab-Datei.

Vermeiden Sie es, langlebige Klartextpasswörter auf dem Host zu speichern. Verwenden Sie nach Möglichkeit Keytabs oder Computerprinzipale, die einer Domäne beigetreten sind.

Überprüfen Sie beim Ausführen von Tests oder Skripten, die KRB5CCNAME setzen oder Keytabs kopieren, nach dem Vorgang nochmals Eigentümer und Berechtigungen – wenn ein Webserver Anmeldedaten zurückweist, liegt häufig ein Problem mit den Dateiberechtigungen vor.

modbus.get-Parameter

Übersicht

Die folgende Tabelle enthält Details zu den Parametern des modbus.get-Datenpunkts.

Parameter

Parameter	Beschreibung	Standardwerte	Beispiel
endpoint	Protokoll und Adresse des Endpunkts, definiert als `protocol://connection_string` Mögliche Protokollwerte: *rtu*, *ascii* (nur Agent 2), *tcp* Format der Verbindungszeichenfolge: mit *tcp* - `address:port` mit serieller Leitung: *rtu*, *ascii* - `port_name:speed:params` wobei 'speed' - 1200, 9600 usw. 'params' - Datenbits (5,6,7 oder 8), Parität (n,e oder o für keine/gerade/ungerade), Stoppbits (1 oder 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	Modbus-Adresse des Geräts, für das sie bestimmt ist (1 bis 247), siehe MODBUS Messaging Implementation Guide (Seite 23) Ein tcp-Gerät (kein GW) ignoriert dieses Feld	seriell: 1 tcp: 255 (0xFF)	2
function	Leer oder Wert einer unterstützten Funktion: 1 - Read Coil, 2 - Read Discrete Input, 3 - Read Holding Registers, 4 - Read Input Registers	leer	3
address	Adresse des ersten Registers, der ersten Coil oder des ersten Eingangs. Wenn 'function' leer ist, sollte sich 'address' in folgendem Bereich befinden: Coil - 00001 - 09999 Discrete input - 10001 - 19999 Input register - 30001 - 39999 Holding register - 40001 - 49999 Wenn 'function' nicht leer ist, liegt das Feld 'address' im Bereich von 0 bis 65535 und wird unverändert verwendet (PDU)	leere function: 00001 nicht-leere function: 0	9999
count	Anzahl der aufeinanderfolgenden 'type', die vom Gerät gelesen werden, wobei gilt: für Coil oder Discrete input ist 'type' = 1 Bit in anderen Fällen: (count*sizeof(type))/2 = tatsächliche Anzahl der zu lesenden Register Wenn 'offset' nicht 0 ist, wird der Wert zur 'tatsächlichen Anzahl' addiert Der zulässige Bereich für die 'tatsächliche Anzahl' ist 1:65535	1	2
type	Datentyp: für Read Coil und Read Discrete Input - *bit* für Read Holding Registers und Read Input Registers: *int8* - 8 Bit *uint8* - 8 Bit (vorzeichenlos) *int16* - 16 Bit *uint16* - 16 Bit (vorzeichenlos) *int32* - 32 Bit *uint32* - 32 Bit (vorzeichenlos) *float* - 32 Bit *uint64* - 64 Bit (vorzeichenlos) *double* - 64 Bit	bit uint16	uint64

Parameter	Beschreibung	Standardwerte	Beispiel
endianness	Endianness-Typ: *be* - Big Endian *le* - Little Endian *mbe* - Mid-Big Endian *mle* - Mid-Little Endian Einschränkungen: für 1 Bit - be für 8 Bit - be,le für 16 Bit - be,le	be	le
offset	Anzahl der Register, beginnend bei 'address', deren Ergebnis verworfen wird. Die Größe jedes Registers beträgt 16 Bit (erforderlich zur Unterstützung von Geräten, die keinen wahlfreien Lesezugriff unterstützen).	0	4

Benutzerdefinierte Namen für Leistungsindikatoren für VMware erstellen

Übersicht

Der VMware-Performance-Counter-Pfad hat das Format `group/counter[rollup]`, wobei:

- `group` - die Performance-Counter-Gruppe, zum Beispiel *cpu*
- `counter` - der Name des Performance-Counters, zum Beispiel *usagemhz*
- `rollup` - der Rollup-Typ des Performance-Counters, zum Beispiel *average*

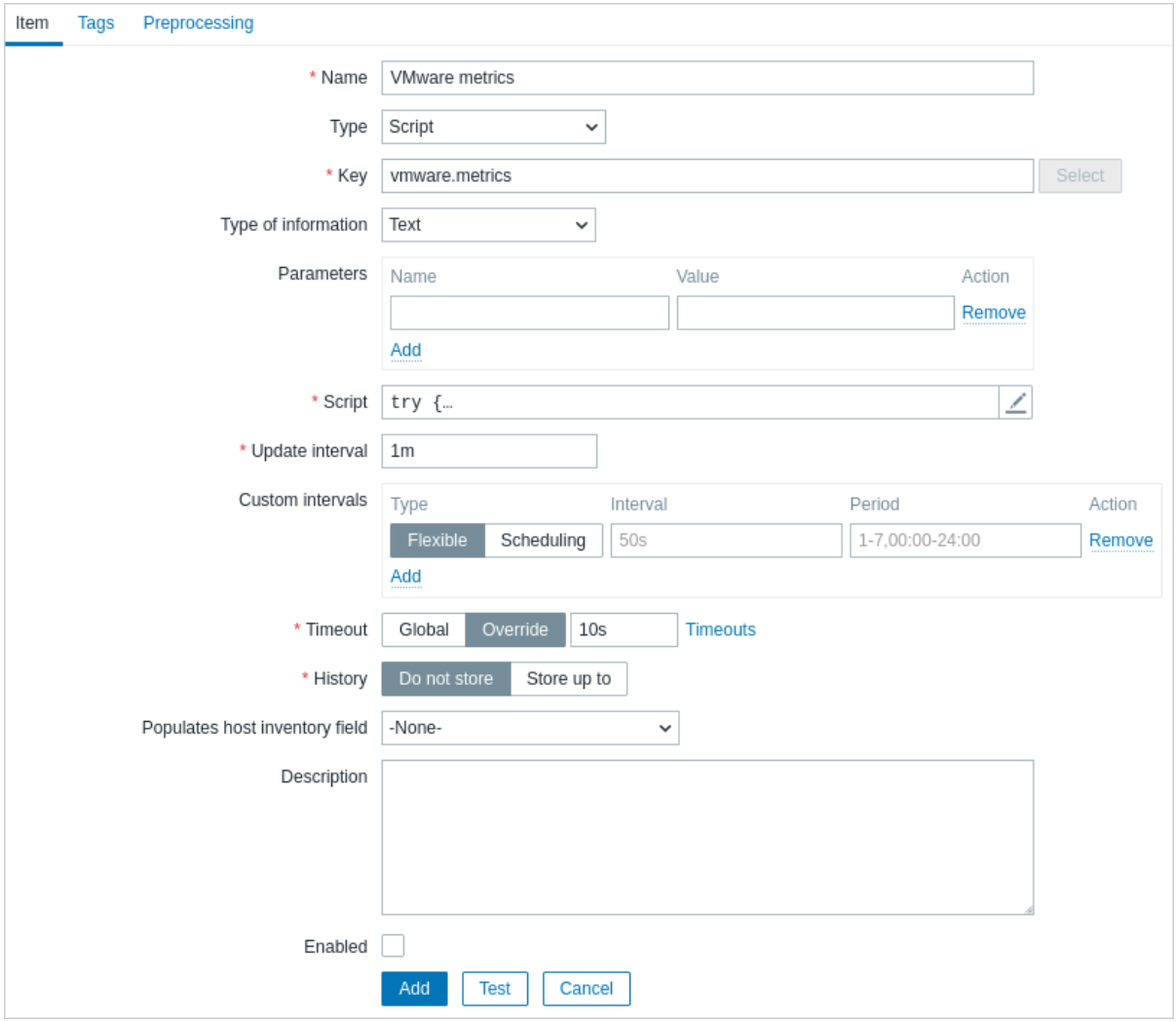
Das obige Beispiel würde also den folgenden Counter-Pfad ergeben: `cpu/usagemhz[average]`

Die Beschreibungen der Performance-Counter-Gruppen, die Namen der Counter und die Rollup-Typen finden Sie in der VMware-Dokumentation.

Es ist möglich, interne Namen zu ermitteln und benutzerdefinierte Performance-Counter-Namen zu erstellen, indem Sie ein Skript-Datenpunkt in Zabbix verwenden.

Konfiguration

1. Erstellen Sie auf dem Haupt-VMware-Host, auf dem der **eventlog[]**-Datenpunkt vorhanden ist, einen deaktivierten Script-Datenpunkt mit den folgenden Parametern:



- *Name*: VMware metrics
- *Type*: Script
- *Key*: vmware.metrics
- *Type of information*: Text
- *Script*: Kopieren Sie das unten bereitgestellte Skript und fügen Sie es ein
- *Timeout*: 10
- *History*: Nicht speichern
- *Enabled*: nicht markiert

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;
```

Nachdem der Datenpunkt konfiguriert wurde, klicken Sie auf die Schaltfläche *Test* und dann auf *Get value*.



Kopieren Sie das empfangene XML in einen beliebigen XML-Formatter und suchen Sie die gewünschte Metrik.

Ein Beispiel für XML für eine Metrik:

```
<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>
```

Verwenden Sie XPath, um den Counter-Pfad aus dem empfangenen XML zu extrahieren. Für das obige Beispiel lautet der XPath:

field	xPath	value
group	//groupInfo[../key=6]/key	cpu
counter	//nameInfo[../key=6]/key	usagemhz
rollup	//rollupType[../key=6]	average

Der resultierende Performance-Counter-Pfad ist in diesem Fall: `cpu/usagemhz[average]`

Rückgabewerte für system.sw.packages.get

Übersicht

Dieser Abschnitt enthält Details zu den Rückgabewerten für den Zabbix-Agent-Datenpunkt `system.sw.packages.get`.

Details

Die Ausgabe dieses Datenpunkts ist ein Array von Objekten, die jeweils die folgenden Eigenschaften enthalten:

- **name** - Paketname
- **manager** - Paketmanager, der diese Daten gemeldet hat (`rpm`, `dpkg`, `pacman`, `pkgtool` oder `portage`)
- **version** - Paketversion
- **size** - unkomprimierte Paketgröße in Byte (falls nicht verfügbar, auf 0 gesetzt)
- **arch** - Paketarchitektur
- **buildtime** - ein Objekt mit 2 Einträgen:
  - **timestamp** - UNIX-Zeitstempel, wann das Paket erstellt wurde (falls nicht verfügbar, auf 0 gesetzt)
  - **value** - menschenlesbares Datum und Uhrzeit, wann das Paket erstellt wurde (falls nicht verfügbar, auf leere Zeichenfolge gesetzt)
- **installtime** - ein Objekt mit 2 Einträgen:
  - **timestamp** - UNIX-Zeitstempel, wann das Paket installiert wurde (falls nicht verfügbar, auf 0 gesetzt)
  - **value** - menschenlesbares Datum und Uhrzeit, wann das Paket installiert wurde (falls nicht verfügbar, auf leere Zeichenfolge gesetzt)

Beispiel:

```
[
    {
        "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": ""
        }
    }
]
```

Rückgabewerte für net.dns.get

Übersicht

Dieser Abschnitt enthält Details zu den Rückgabewerten für den Zabbix Agent 2-Datenpunkt `net.dns.get`.

Details

Die Ausgabe dieses Datenpunkts ist ein Objekt, das DNS-Record-Informationen auf Grundlage der im Datenpunkt-Schlüssel angegebenen Parameter enthält.

Beispielsweise kann der Datenpunkt `net.dns.get[,example.com]` das folgende JSON einer abgelehnten Abfrage zurückgeben:

```
{
    "flags": [
        "RA"
    ],
    "query_time": "0.019030",
    "question_section": [
        {
            "qclass": "IN",
            "qname": "example.com.",
            "qtype": "SOA"
        }
    ],
    "response_code": "REFUSED",
    "zbx_error_code": 0
}
```

Durch Angabe der IP-Adresse des DNS-Servers kann der Datenpunkt `net.dns.get[192.0.2.1,example.com]` das folgende JSON zurückgeben:

```
{
    "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
}
```


Wenn ein Verbindungsproblem vorliegt, kann der Datenpunkt `net.dns.get[192.0.2.1,example.com]` einen Fehler zurückgeben:


```
{
    "zbx_error_code": -1,
    "zbx_error_msg": "Communication error: read udp 192.0.2.1:12345->192.0.2.1:53: i/o timeout"
}
```


Folgende Arten von Fehlercodes sind möglich:

Scenario	"zbx_error_code"	"zbx_error_msg"
Keine Fehler, und die DNS-Antwort wurde empfangen und verarbeitet.	0	
DNS ist ausgefallen.	-1	"Communication error"
Fehler tritt während der JSON-Analyse auf	-2	"Received unexpected response"

Mit zusätzlichen Parametern kann der Datenpunkt `net.dns.get[192.0.2.1,example.com,ANY,5,5,tcp,"cdflag,rdflag,dnssec,r` das folgende JSON zurückgeben:


```
{
    "additional_section": [
        {
            "extended_rcode": 32768,
            "name": ".",
            "rdata": {
                "options": [
                {
                    "code": 0,
                    "nsid": "67 70 64 6e 73 2d 6c 70 70"
                }
                ]
            },
            "rdlength": 13,
            "type": "OPT",
            "udp_payload": 512
        }
    ],
```

```
    "answer_section": [
        {
            "class": "IN",
            "name": "example.com.",
            "rdata": {
                "a": "192.0.2.1"
            },
            "rdlength": 4,
            "ttl": 19308,
            "type": "A"
        },
        {
            "class": "IN",
            "name": "example.com.",
            "rdata": {
                "algorithm": 13,
                "expiration": 1704715951,
                "inception": 1702910624,
                "key_tag": 21021,
                "labels": 2,
                "orig_ttl": 86400,
                "signature": "HVBOBcJJQy0S08J3f8kviPj8UkEUj7wmyiMyQqPSWgQIY9SCEJ5plq6KuxJmtAek1txZWXDo+6tp
                "signer_name": "example.com.",
                "type_covered": "A"
            },
            "rdlength": 95,
            "ttl": 19308,
            "type": "RRSIG"
        }
    ],
    "flags": [
        "RD",
        "RA",
        "AD",
        "CD"
    ],
    "query_time": "0.058221",
    "question_section": [
        {
            "qclass": "IN",
            "qname": "example.com.",
            "qtype": "ANY"
        }
    ],
    "response_code": "NOERROR",
    "zbx_error_code": 0
}
```


Siehe auch

Weitere Informationen über DNS-Einträge finden Sie unter:

- Domänennamen - Implementierung und Spezifikation
- Domain Name System (DNS) Parameter

Rückgabewerte für net.if.get

Übersicht

Dieser Abschnitt enthält Details zu den Rückgabewerten für den Datenpunkt `net.if.get`. Der Datenpunkt wird unter Linux von Zabbix Agent und Zabbix Agent 2 unterstützt.

Details

Dieser Datenpunkt gibt detaillierte Informationen zu Netzwerkschnittstellen zurück, deren Namen dem angegebenen regulären Ausdruck entsprechen. Die Ausgabe ist als JSON-Objekt strukturiert, das die Arrays `config` und `values` enthält.

Die Ausgabe enthält die folgenden Eigenschaften:

- **name** - Schnittstellenname, aus `/proc/net/dev` gelesen;
- **ifalias** - Schnittstellenalias, aus `/sys/class/net/<name>/ifalias` gelesen;
- **mac** - MAC-Adresse der Schnittstelle, aus `/sys/class/net/<name>/address` gelesen;
- **type** - Schnittstellentyp (`physical`, `virtual`, `loopback`);
- **speed** - die Verbindungsgeschwindigkeit in Mbit/s, Ethernet-spezifisch, aus `/sys/class/net/<name>/speed` gelesen;
- **duplex** - letzter oder aktueller Duplex-Wert der Schnittstelle (`half`, `full`), Ethernet-spezifisch, aus `/sys/class/net/<name>/duple` gelesen;
- **administrative_state** - konfigurierter Status (`down`, `up`), aus `/sys/class/net/<name>/flags` gelesen;
- **operational_state** - RFC2863-Betriebsstatus der Schnittstelle (`unknown`, `notpresent`, `down`, `lowerlayerdown`, `testing`, `dormant`, `up`), aus `sys/class/net/<name>/operstate` gelesen;
- **carrier** - Vorhandensein einer physischen Verbindung (0 - physische Verbindung ist down, 1 - physische Verbindung ist up), aus `/sys/class/net/<name>/carrier` gelesen;
- **carrier_changes** - Gesamtzahl der Änderungen des Verbindungsstatus von up zu down und umgekehrt, aus `/sys/class/net/<name>` gelesen;
- **carrier_up_count** - Anzahl, wie oft die Verbindung up geworden ist, aus `/sys/class/net/<name>/carrier_up_count` gelesen;
- **carrier_down_count** - Anzahl, wie oft die Verbindung down geworden ist, aus `/sys/class/net/<name>/carrier_down_count` gelesen;
- **in** - eingehender Datenverkehr, aus `/proc/net/dev` gelesen. Enthält Schnittstellenzähler:
  - **bytes** - Anzahl fehlerfreier Bytes;
  - **packets** - Anzahl fehlerfreier Pakete;
  - **errors** - Gesamtzahl fehlerhafter Pakete;
  - **dropped** - Anzahl empfangener, aber nicht verarbeiteter Pakete + Anzahl der vom Host verpassten Pakete;
  - **overruns** - Anzahl der Überlaufereignisse;
  - **frame** - Anzahl der Pakete mit Framing-Fehlern;
  - **compressed** - Anzahl korrekt empfangener komprimierter Pakete;
  - **multicast** - empfangene Multicast-Pakete.
- **out** - ausgehender Datenverkehr, aus `/proc/net/dev` gelesen. Enthält Schnittstellenzähler:
  - **bytes** - Anzahl fehlerfreier Bytes;
  - **packets** - Anzahl fehlerfreier Pakete;
  - **errors** - Gesamtzahl fehlerhafter Pakete;
  - **dropped** - Anzahl empfangener, aber nicht verarbeiteter Pakete + Anzahl der vom Host verpassten Pakete;
  - **overruns** - Anzahl der Überlaufereignisse;
  - **collisions** - Anzahl der Kollisionen bei Paketübertragungen;
  - **carrier** - Anzahl der Frame-Übertragungsfehler aufgrund eines Carrier-Verlusts während der Übertragung;
  - **compressed** - Anzahl übertragener komprimierter Pakete.

Nicht verfügbare Metriken werden aus der JSON-Ausgabe ausgeschlossen. Im folgenden Beispiel ist für die Schnittstelle "lo" kein Alias konfiguriert, daher wird die Eigenschaft `ifalias` ausgeschlossen.

Beispiel:


```
{
    "config": [
        {
            "name": "lo",
            "mac": "00:00:00:00:00:00",
            "type": "loopback",
            "administrative_state": "up",
            "operational_state": "unknown",
            "carrier": 1
        },
        {
            "name": "enp3s0",
            "ifalias": "Wired Ethernet connection",
            "mac": "8c:16:45:e3:1d:58",
            "type": "physical",
            "duplex": "unknown",
            "administrative_state": "up",
            "operational_state": "down",
```

```
            "carrier": 0
        },
        {
            "name": "wlp5s0",
            "ifalias": "Wi-Fi network adapter",
            "mac": "34:e1:2d:72:67:c6",
            "type": "physical",
            "administrative_state": "up",
            "operational_state": "up",
            "carrier": 1
        }
    ],
    "values": [
        {
            "name": "lo",
            "mac": "00:00:00:00:00:00",
            "carrier": 1,
            "carrier_changes": 0,
            "carrier_up_count": 0,
            "carrier_down_count": 0,
            "in": {
                "bytes": 1040047,
                "packets": 10784,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "frame": 0,
                "compressed": 0,
                "multicast": 0
            },
            "out": {
                "bytes": 1040047,
                "packets": 10784,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "collisions": 0,
                "carrier": 0,
                "compressed": 0
            }
        },
        {
            "name": "enp3s0",
            "ifalias": "Wired Ethernet connection",
            "mac": "8c:16:45:e3:1d:58",
            "carrier": 0,
            "carrier_changes": 1,
            "carrier_up_count": 0,
            "carrier_down_count": 1,
            "in": {
                "bytes": 0,
                "packets": 0,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "frame": 0,
                "compressed": 0,
                "multicast": 0
            },
            "out": {
                "bytes": 0,
                "packets": 0,
```

```
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "collisions": 0,
                "carrier": 0,
                "compressed": 0
            }
        },
        {
            "name": "wlp5s0",
            "ifalias": "Wi-Fi network adapter",
            "mac": "34:e1:2d:72:67:c6",
            "carrier": 1,
            "carrier_changes": 6,
            "carrier_up_count": 3,
            "carrier_down_count": 3,
            "in": {
                "bytes": 165537298,
                "packets": 216182,
                "errors": 0,
                "dropped": 291,
                "overruns": 0,
                "frame": 0,
                "compressed": 0,
                "multicast": 0
            },
            "out": {
                "bytes": 252574347,
                "packets": 246607,
                "errors": 0,
                "dropped": 0,
                "overruns": 0,
                "collisions": 0,
                "carrier": 0,
                "compressed": 0
            }
        }
    ]
}
```


Siehe auch

Weitere Informationen zur sysfs-ABI für Netzwerkgeräte finden Sie unter:

- Linux-Kernel-Dokumentation

Rückgabewerte für vfs.dev.get

Übersicht

Dieser Abschnitt enthält Details zu den Rückgabewerten für den `vfs.dev.get`-Datenpunkt. Der Datenpunkt wird unter Linux von Zabbix Agent und Zabbix Agent 2 sowie unter AIX von Zabbix Agent unterstützt.

Details

Dieser Datenpunkt liefert detaillierte Informationen zu Blockgeräten, deren Namen dem angegebenen regulären Ausdruck entsprechen.

Die Ausgabe ist als JSON-Objekt strukturiert, das `config` und, abhängig vom `<mode>`-Parameter, `values`-Arrays enthält.

Das `config`-Array enthält die Eigenschaften, die für den ausgewählten Modus verfügbar sind. Das `values`-Array wird nur für Modi zurückgegeben, die Statistiken zu Blockgeräten bereitstellen.

Nicht verfügbare Metriken werden aus der JSON-Ausgabe ausgeschlossen.

Die folgenden Abschnitte beschreiben die für jeden Modus zurückgegebenen Eigenschaften.

disks mode

`vfs.dev.get[disks,.*]` gibt Konfigurationsdaten von Blockgeräten zurück.

Eigenschaften:

- **name** - Name des Blockgeräts;
- **devid** - Kennung des Blockgeräts, zum Beispiel eine auf WWN basierende oder vom Kernel generierte Geräte-ID;
- **type** - Typ des Blockgeräts (`disk`, `partition`, `rom`);
- **path** - Pfad zum Knoten des Blockgeräts;
- **model** - Gerätemodell;
- **serial** - Seriennummer des Geräts;
- **wwn** - World Wide Name (WWN) des Geräts;
- **size_bytes** - Gerätegröße in Bytes;
- **logical_block_size** - logische Blockgröße in Bytes;
- **physical_block_size** - physische Blockgröße in Bytes.

Beispiel für Linux:


```
{
  "config": [
    {
      "name": "sda",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY",
      "type": "disk",
      "path": "/dev/sda",
      "model": " ST2000LM015-2E81",
      "wwn": "naa.5000c500b2791423",
      "size_bytes": 2000398934016,
      "logical_block_size": 512,
      "physical_block_size": 4096
    },
    {
      "name": "sdb",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0",
      "type": "disk",
      "path": "/dev/sdb",
      "model": "DataTraveler 3.0",
      "size_bytes": 61991813632,
      "logical_block_size": 512,
      "physical_block_size": 512
    },
    {
      "name": "nvme0n1",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633",
      "type": "disk",
      "path": "/dev/nvme0n1",
      "model": "SAMSUNG MZVLW256HEHP-000L7",
      "serial": "S35ENX1K753633",
      "wwn": "eui.002538b781b7dc55",
      "size_bytes": 256060514304,
      "logical_block_size": 512,
      "physical_block_size": 512
    }
  ]
}
```


Beispiel für AIX:


```
{
  "config": [
    {
      "name": "cd0",
      "type": "disk",
      "path": "/dev/cd0",
      "size_bytes": 0,
```

```
      "logical_block_size": 512
    },
    {
      "name": "hdisk1",
      "type": "disk",
      "devid": "48333321360050768OC818022F000000000015EA404214503IBMfcp05VDASD03AIXvscsi",
      "path": "/dev/hdisk1",
      "size_bytes": 51136000000,
      "logical_block_size": 512
    },
    {
      "name": "hdisk0",
      "type": "disk",
      "devid": "48333321360050768OC818022F000000000015EA304214503IBMfcp05VDASD03AIXvscsi",
      "path": "/dev/hdisk0",
      "size_bytes": 0,
      "logical_block_size": 512
    }
  ]
}
```


Geräte-Modus

`vfs.dev.get[devices,.*]` gibt Konfigurationsdaten von Blockgeräten zurück, einschließlich Partitionen.

Eigenschaften:

- **name** - Name des Blockgeräts.
- **devid** - Kennung des Blockgeräts.
- **type** - Typ des Blockgeräts (`disk`, `partition`, `rom`).
- **partitions** - Partitionen auf dem Gerät. Enthält Partitionsnamen und deren Größen in Bytes.

Beispiel für Linux:


```
{
  "config": [
    {
      "name": "sda",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY",
      "type": "disk",
      "partitions": {
        "sda1": 16759808,
        "sda2": 2000381018112
      }
    },
    {
      "name": "sdb",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0",
      "type": "disk",
      "partitions": {
        "sdb1": 27631878144,
        "sdb2": 34359768576,
        "sdb3": 32256
      }
    },
    {
      "name": "nvme0n1",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633",
      "type": "disk",
      "partitions": {
        "nvme0n1p1": 209715200,
        "nvme0n1p2": 16777216,
        "nvme0n1p3": 255060869120,
        "nvme0n1p4": 770703360
      }
```

```
    }
  ]
}
```

disk_stats mode

`vfs.dev.get[disk_stats,.*]` gibt Konfigurationsdaten von Blockgeräten und Festplattenstatistiken zurück.

Eigenschaften:

- **name** - Name des Blockgeräts.
- **devid** - Kennung des Blockgeräts.
- **type** - Typ des Blockgeräts (`disk`, `partition`, `rom`).
- **size_bytes** - Gerätegröße in Bytes.
- **logical_block_size** - logische Blockgröße in Bytes.
- **stats** - I/O-Statistiken des Blockgeräts. Enthält:
  - **reads_completed** - Anzahl der abgeschlossenen Lesevorgänge;
  - **writes_completed** - Anzahl der abgeschlossenen Schreibvorgänge;
  - **bytes_read** - Anzahl der gelesenen Bytes;
  - **bytes_written** - Anzahl der geschriebenen Bytes;
  - **io_time_ms** - gesamte für I/O aufgewendete Zeit in Millisekunden.

Beispiel für Linux

```
{
  "config": [
    {
      "name": "sda",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY",
      "type": "disk",
      "size_bytes": 61991813632
    },
    {
      "name": "sdb",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0",
      "type": "disk",
      "size_bytes": 61991813632
    },
    {
      "name": "nvme0n1",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633",
      "type": "disk",
      "size_bytes": 256060514304
    }
  ],
  "values": [
    {
      "name": "sda",
      "stats": {
        "reads_completed": 871,
        "writes_completed": 0,
        "bytes_read": 13045760,
        "bytes_written": 0,
        "io_time_ms": 1256
      }
    },
    {
      "name": "sdb",
      "stats": {
        "reads_completed": 51017,
        "writes_completed": 10682,
        "bytes_read": 10514627,
        "bytes_written": 1530472,
        "io_time_ms": 284988
      }
```

```
    },
    {
      "name": "nvme0n1",
      "stats": {
        "reads_completed": 1259,
        "writes_completed": 7,
        "bytes_read": 49698,
        "bytes_written": 6,
        "io_time_ms": 243
      }
    }
  ]
}
```


Beispiel für AIX:


```
{
  "config": [
    {
      "name": "cd0",
      "type": "disk",
      "path": "/dev/cd0",
      "size_bytes": 0,
      "logical_block_size": 2048
    },
    {
      "name": "hdisk1",
      "type": "disk",
      "devid": "48333321360050768 0C818022F000000000015EA404214503IBMfcp05VDASD03AIXvscsi",
      "path": "/dev/hdisk1",
      "size_bytes": 51136000000,
      "logical_block_size": 512
    },
    {
      "name": "hdisk0",
      "type": "disk",
      "devid": "48333321360050768 0C818022F000000000015EA304214503IBMfcp05VDASD03AIXvscsi",
      "path": "/dev/hdisk0",
      "size_bytes": 0,
      "logical_block_size": 512
    }
  ],
  "values": [
    {
      "name": "cd0",
      "stats": {
        "reads_completed": 21,
        "writes_completed": 0,
        "bytes_read": 75776,
        "bytes_written": 0,
        "io_time_ms": 0
      }
    },
    {
      "name": "hdisk1",
      "stats": {
        "reads_completed": 35888,
        "writes_completed": 1165147,
        "bytes_read": 593699840,
        "bytes_written": 10009646592,
        "io_time_ms": 69
      }
    },
```

```
    {
      "name": "hdisk0",
      "stats": {
        "reads_completed": 3,
        "writes_completed": 0,
        "bytes_read": 12800,
        "bytes_written": 0,
        "io_time_ms": 0
      }
    }
  ]
}
```

device_stats mode

`vfs.dev.get[device_stats,.*]` gibt Konfigurationsdaten von Blockgeräten und Gerätestatistiken zurück.

Eigenschaften:

- **name** - Name des Blockgeräts.
- **devid** - Kennung des Blockgeräts.
- **type** - Typ des Blockgeräts (`disk`, `partition`, `rom`).
- **size_bytes** - Gerätegröße in Bytes.
- **stats** - I/O-Statistiken des Blockgeräts. Enthält:
  - **reads_completed** - Anzahl der abgeschlossenen Leseoperationen;
  - **writes_completed** - Anzahl der abgeschlossenen Schreiboperationen;
  - **bytes_read** - Anzahl der gelesenen Bytes;
  - **bytes_written** - Anzahl der geschriebenen Bytes;
  - **io_time_ms** - Gesamtzeit für I/O in Millisekunden.

Beispiel für Linux:

```
{
  "config": [
    {
      "name": "sda",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY",
      "type": "disk",
      "size_bytes": 61991813632
    },
    {
      "name": "sda1",
      "type": "partition",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY-part1",
      "size_bytes": 1234567890
    },
    {
      "name": "sda2",
      "type": "partition",
      "devid": "ata-ST2000LM015-2E8174_ZDZ3YHZY-part2",
      "size_bytes": 1234567890
    },
    {
      "name": "sdb",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0",
      "type": "disk",
      "size_bytes": 61991813632
    },
    {
      "name": "sdb1",
      "type": "partition",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0-part1",
      "size_bytes": 1234567890
    },
    {
```

```
      "name": "sdb2",
      "type": "partition",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0-part2",
      "size_bytes": 1234567890
    },
    {
      "name": "sdb3",
      "type": "partition",
      "devid": "usb-Kingston_DataTraveler_3.0_E0D55E6C382C15A158991CBD-0:0-part3",
      "size_bytes": 1234567890
    },
    {
      "name": "nvme0n1",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633",
      "type": "disk",
      "size_bytes": 256060514304
    },
    {
      "name": "nvme0n1p1",
      "type": "partition",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633-part1",
      "size_bytes": 1234567890
    },
    {
      "name": "nvme0n1p2",
      "type": "partition",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633-part2",
      "size_bytes": 1234567890
    },
    {
      "name": "nvme0n1p3",
      "type": "partition",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633-part3",
      "size_bytes": 1234567890
    },
    {
      "name": "nvme0n1p4",
      "type": "partition",
      "devid": "nvme-SAMSUNG_MZVLW256HEHP-000L7_S35ENX1K753633-part4",
      "size_bytes": 1234567890
    }
  ],
  "values": [
    {
      "name": "sda",
      "stats": {
        "reads_completed": 871,
        "writes_completed": 0,
        "bytes_read": 13045760,
        "bytes_written": 0,
        "io_time_ms": 1256
      }
    },
    {
      "name": "sda1",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
```

```
    },
    {
      "name": "sda2",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "sdb",
      "stats": {
        "reads_completed": 51017,
        "writes_completed": 10682,
        "bytes_read": 10514627,
        "bytes_written": 1530472,
        "io_time_ms": 284988
      }
    },
    {
      "name": "sdb1",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "sdb2",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "sdb3",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "nvme0n1",
      "stats": {
        "reads_completed": 1259,
        "writes_completed": 7,
        "bytes_read": 49698,
        "bytes_written": 6,
        "io_time_ms": 243
      }
    },
    {
```

```
      "name": "nvme0n1p1",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "nvme0n1p2",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "nvme0n1p3",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    },
    {
      "name": "nvme0n1p4",
      "stats": {
        "reads_completed": 123,
        "writes_completed": 123,
        "bytes_read": 123,
        "bytes_written": 123,
        "io_time_ms": 123
      }
    }
  ]
}
```


Hinweise zu system.cpu.util-Datenpunkten unter Windows

Der Datenpunkt system.cpu.util liefert den prozentualen CPU-Auslastungswert.

Wenn auf dem Zabbix Agent für Windows ein Collector-Prozess gestartet wird, wird ein Puffer für N Datenpunkte für N logische Prozessoren (Threads) initialisiert. Die Werte im Puffer werden periodisch mithilfe von Windows-Leistungsindikatoren aktualisiert. Diese Werte werden gemeldet, wenn `system.cpu.util[n]` angefordert wird, wobei 0 <= n < N gilt.

Ein weiterer Datenpunkt im Puffer wird für `system.cpu.util[total]` verwendet.

Zabbix wählt automatisch unterschiedliche Leistungsindikatoren aus, um sowohl NUMA-Systeme als auch ältere Windows-Versionen ohne Unterstützung für Prozessorguppen zu unterstützen.

Wenn die Anzahl der logischen Prozessoren (Threads) kleiner oder gleich 64 ist und die Anzahl der Prozessorgruppen gleich 1 ist, dann:

```
N ist die Anzahl der logischen Prozessoren (Threads).

\Processor(_Total)\% Processor Time
\Processor(0)\% Processor Time
\Processor(1)\% Processor Time
\Processor(2)\% Processor Time
```

```
...
\Processor(N-1)\% Processor Time
```

Andernfalls:

```
G ist die Anzahl der Prozessorgruppen und N ist die Anzahl der logischen Prozessoren (Threads) in der Grup

\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
```

Unterstützung für große JSON-Werte

Um eine zuverlässige Verarbeitung großer Werte mit dem JSON Datenformat (1 MiB oder mehr) sicherzustellen, überprüfen Sie die folgenden Konfigurationsänderungen und wenden Sie sie bei Bedarf an:

1. Wenn Sie MySQL oder MariaDB verwenden, erhöhen Sie die maximal zulässige Paketgröße in `/etc/mysql/my.cnf` (sie ist auf einigen Systemen möglicherweise standardmäßig auf 1 MB gesetzt):

```
[mysqld]
max_allowed_packet=128M

[mysqldump]
max_allowed_packet=1G
```

2. Passen Sie bei Bedarf zusätzliche MySQL/MariaDB Server-Systemvariablen und InnoDB-Systemvariablen an. Zum Beispiel:

```
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. Erhöhen Sie `HistoryCacheSize` in der Zabbix-Server-Konfigurationsdatei auf `2G`.

4. Wenn Sie Zabbix Proxy verwenden, erhöhen Sie `HistoryCacheSize` auf `2G` und `ProxyMemoryBufferSize` auf `2G` in der Proxy-Konfigurationsdatei.

5. Wenn Sie Zabbix Agent verwenden, erhöhen Sie die Kommunikations-Timeouts zwischen Zabbix Server/Proxy und Agent (z. B. Datenpunkt-Timeouts oder den Wert des Parameters `Timeout` in den Zabbix-Server/Proxy- und Agent-Konfigurationsdateien). Andernfalls werden Nachrichten zwischen ihnen möglicherweise nicht vollständig übertragen, und es können Fehler wie *"message length does not match expected length"* auftreten. Überprüfen Sie außerdem bei Bedarf weitere Timeouts, z. B. das Script-Timeout oder das Timeout für Datenpunkt-Test.

Unterstützte Funktionen

Siehe Auslöserausdruck und Funktionen.

Makros

Es ist möglich, die standardmäßig verfügbaren unterstützten Makros und vom Benutzer definierte Makros, die nach Ort unterstützt werden zu verwenden.

Von Standort unterstützte Makros

Übersicht

Die Tabelle nach Speicherort unterstützte Makros enthält eine Referenz aller Makros (integrierte, Ausdrucks- und benutzerdefinierbare), die am jeweiligen Speicherort unterstützt werden.

Wenn alle Makros aus einer zugehörigen Gruppe unterstützt werden, wird nur der Gruppenname angegeben. Klicken Sie auf einen Makronamen oder Gruppennamen, um die vollständigen Details der Makros anzuzeigen.

Die Liste Makrodetails enthält eine detaillierte Liste aller **integrierten** Makros, gruppiert nach Anwendungsbereich. Beachten Sie, dass Sie zur Anpassung von Makrowerten (zum Beispiel zum Kürzen oder Extrahieren bestimmter Teilzeichenfolgen) Makrofunktionen verwenden können.

Unterstützte Makros nach Ort

Ort	Unterstützte Makros	Kommentare
Aktion		
Bedingung für den Zeitbereich, Standarddauer des Operationsschritts, Schrittdauer	Benutzerdefinierbare Makros (nur global)	Ein einzelnes Makro muss das gesamte Feld ausfüllen; mehrere Makros bzw. eine Mischung mit Text werden nicht unterstützt
Benachrichtigungen und Befehle		
Auslöserbasiert	Makros für Aktion Makros für Datum/Uhrzeit {ESC.HISTORY} Makros für Ereignis, Makros für Ursache/Symptom-Ereignis Makros für Ereignisaktualisierung: {EVENT.UPDATE.HISTORY}, {EVENT.UPDATE.STATUS} Ausdrucksmakro: {?EXPRESSION} {FUNCTION.VALUE} Host-Makros: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Makros für Ziel-Host (nur Befehle) Makros für Host-Inventar Makros für Datenpunkt (außer {ITEM.STATE}, {ITEM.STATE.ERROR}) Makros für Proxy Auslöser-Makros: {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} Benutzerdefinierbare Makros	Siehe auch: Makros in Nachrichten verwenden

Ort	Unterstützte Makros	Kommentare
Problemaktualisierung	Makros für Aktion Makros für Datum/Uhrzeit {ESC.HISTORY} Makros für Ereignis, Makros für Ursache/Symptom-Ereignis, Makros für Wiederherstellungsereignis Makros für Ereignisaktualisierung: alle (außer {EVENT.UPDATE.NSEVERITY}, {EVENT.UPDATE.SEVERITY}) Ausdrucksmakro: {?EXPRESSION} Makros für Funktion Host-Makros: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Makros für Ziel-Host (nur Befehle) Makros für Host-Inventar Makros für Datenpunkt (außer {ITEM.STATE}, {ITEM.STATE.ERROR}) Makros für Proxy Auslöser-Makros: {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} Benutzerdefinierbare Makros	
Problemwiederherstellung	Makros für Wiederherstellungsereignis Makros für Ereignis {FUNCTION.RECOVERY.VALUE} {TRIGGER.EXPRESSION.RECOVERY.EXPLAIN}	
Erkennung	Makros für Aktion Makros für Datum/Uhrzeit Makros für Erkennung Makros für Ereignisse: {EVENT.AGE}, {EVENT.DATE}, {EVENT.ID}, {EVENT.OBJECT}, {EVENT.SOURCE}, {EVENT.TIME}, {EVENT.TIMESTAMP} Makros für Ziel-Host (nur Befehle) Makros für Proxy	
Autoregistrierung	Makros für Aktion Makros für Datum/Uhrzeit Makros für Ereignisse: {EVENT.AGE}, {EVENT.DATE}, {EVENT.ID}, {EVENT.OBJECT}, {EVENT.SOURCE}, {EVENT.TIME}, {EVENT.TIMESTAMP} Host-Makros: {HOST.HOST}, {HOST.IP}, {HOST.PORT}, {HOST.METADATA} Makros für Ziel-Host (nur Befehle) Makros für Proxy	
Servicebasiert	Makros für Aktion Makros für Datum/Uhrzeit {ESC.HISTORY} Makros für Ereignisse: alle außer {EVENT.ACK.STATUS}, {EVENT.OPDATA} Makros für Service Benutzerdefinierbare Makros	
Servicewiederherstellung	Makros für Ereignis (außer {EVENT.ACK.STATUS}, {EVENT.OPDATA}), Makros für Wiederherstellungsereignis	

Ort	Unterstützte Makros	Kommentare
Serviceaktualisierung	Makros für Aktion Makros für Datum/Uhrzeit {ESC.HISTORY} Makros für Ereignis (außer {EVENT.ACK.STATUS}, {EVENT.OPDATA}) Makros für Ereignisaktualisierung: {EVENT.UPDATE.DATE}, {EVENT.UPDATE.NSEVERITY}, {EVENT.UPDATE.SEVERITY}, {EVENT.UPDATE.STATUS}, {EVENT.UPDATE.TIME}, {EVENT.UPDATE.TIMESTAMP} Makros für Service Benutzerdefinierbare Makros	
Interne Benachrichtigungen	Makros für Aktion Makros für Datum/Uhrzeit {ESC.HISTORY} Makros für Ereignis (außer {EVENT.NSEVERITY}, {EVENT.SEVERITY}), Makros für Wiederherstellungsereignis Host-Makros: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Makros für Host-Inventar Datenpunkt-Makros: {ITEM.DESCRIPTION}, {ITEM.DESCRIPTION.ORIG}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, {ITEM.NAME}, {ITEM.NAME.ORIG}, {ITEM.VALUETYPE} Makros für Proxy	
Datenpunktbasiert	Datenpunkt-Makros: {ITEM.STATE}, {ITEM.STATE.ERROR}	
LLD-Regelbasiert	Makros für Low-Level-Discovery-Benachrichtigung	
Auslöserbasiert	Auslöser-Makros: {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} Benutzerdefinierbare Makros	
Parameter für Alert-Skripte	Makros für Alert Benutzerdefinierbare Makros (nur global)	
Connector		
Felder für URL, Benutzername, Passwort, Bearer-Token, HTTP-Proxy, SSL-Zertifikat, SSL-Schlüsseldatei, SSL-Schlüsselpasswort	Benutzerdefinierbare Makros (nur global)	
Timeout-Feld	Benutzerdefinierbare Makros (nur global)	Ein einzelnes Makro muss das gesamte Feld ausfüllen; mehrere Makros bzw. eine Mischung mit Text werden nicht unterstützt.

Ort	Unterstützte Makros	Kommentare
Name des Diagramms	Ausdrucksmakro: {?EXPRESSION}	In diesem Makro an dieser Stelle wird nur eine einzelne Funktion **avg**, **last**, **max** oder **min** mit Sekunden als Parameter unterstützt (Zeit-Suffixe können verwendet werden). Das Makro {HOST.HOST<1-9>} kann innerhalb des Makros als Host verwendet werden (siehe auch indizierte Makros). Beispielverwendung: {?avg(/{HOST.HOST}/item.key,1h)} Nicht unterstützte Verwendung: {?last(/host/item1)/last(/host/item2)} {?last(/host/item1)*10} {?count(/host/item1,5m)}
Host, Host-Prototyp		
Schnittstellen-IP/DNS	Host-Makros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME} Benutzerdefinierbare Makros [1]	[1] Für die IP-Adresse muss ein einzelnes Makro das gesamte Feld ausfüllen; mehrere Makros bzw. eine Mischung mit Text werden nicht unterstützt
Schnittstellenport	Benutzerdefinierbare Makros	Ein einzelnes Makro muss das gesamte Feld ausfüllen; mehrere Makros bzw. eine Mischung mit Text werden nicht unterstützt.
SNMP-Community *SNMPv3-Kontextname,* *Sicherheitsname,* *Authentifizierungs-Passphrase,* *Datenschutz-Passphrase* *IPMI-Benutzername,* *Passwort*	Benutzerdefinierbare Makros	
Datenpunkt, Datenpunkt-Prototyp, LLD-Regel		
Name	Benutzerdefinierbare Makros	In LLD-Regelnamen nicht unterstützt.
Beschreibung	Benutzerdefinierbare Makros	

Ort	Unterstützte Makros	Kommentare
Schlüsselparameter	Host-Makros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Benutzerdefinierbare Makros	Die unterstützten {HOST.*}-Makros werden auf die Schnittstelle aufgelöst, die für den Datenpunkt ausgewählt ist. Wenn sie in Datenpunkten ohne Schnittstellen verwendet werden, werden sie entweder auf die Zabbix-Agent-, SNMP-, JMX- oder IPMI-Schnittstelle des Hosts in dieser Prioritätsreihenfolge aufgelöst oder auf 'UNKNOWN', wenn der Host keine Schnittstelle hat. Makros werden im Datenpunktschlüssel von Zabbix-Trapper-Datenpunkten nicht unterstützt.
Felder für Aktualisierungsintervall, Benutzerdefinierte Intervalle, Timeout (verfügbar für unterstützte Datenpunkttypen), Speichern bis zu (für Verlauf/Trends)	Benutzerdefinierbare Makros	Ein einzelnes Makro muss das gesamte Feld ausfüllen; mehrere Makros bzw. eine Mischung mit Text werden nicht unterstützt.
Felder für Gelöschte Ressourcen löschen, Gelöschte Ressourcen deaktivieren	Benutzerdefinierbare Makros	Diese Felder werden nur für LLD-Regeln unterstützt. Ein einzelnes Makro muss das gesamte Feld ausfüllen; mehrere Makros bzw. eine Mischung mit Text werden nicht unterstützt.
Parameter für Vorverarbeitungsschritte (einschließlich benutzerdefinierter Skripte), *Parameter für benutzerdefinierte Fehlerbehandlung* (Felder *Wert setzen auf* und *Fehler setzen auf*)	Benutzerdefinierbare Makros	
Reguläre Ausdrücke für Filter	Host-Makros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Benutzerdefinierbare Makros	Diese Felder werden nur für LLD-Regeln unterstützt.
Reguläre Ausdrücke für Überschreibungsfilter	Benutzerdefinierbare Makros	Diese Felder werden nur für LLD-Regeln unterstützt.

Ort	Unterstützte Makros	Kommentare
Überschreibungsoperationen: *Aktualisierungsintervall*, *Speicherzeitraum für Verlauf*, *Speicherzeitraum für Trends* (für Datenpunkt-Prototypen)	Benutzerdefinierbare Makros	Diese Felder werden nur für LLD-Regeln unterstützt. Ein einzelnes Makro muss das gesamte Feld ausfüllen; mehrere Makros bzw. eine Mischung mit Text werden nicht unterstützt.
Datenpunkttypen		
Browser-Datenpunkt *Parameternamen und -werte*	Host-Makros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Makros für Host-Inventar Datenpunkt-Makros: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} Benutzerdefinierbare Makros	
Script-Datenpunkt *Parameternamen und -werte*		
Berechneter/aggregierter Datenpunkt (*Ausdruckskonstanten*, *Funktionsparameter*, *Datenpunktschlüsselparameter*, (nur aggregierter Datenpunkt) *Filterbedingungen* (Hostgruppenname und Tag-Name))	Benutzerdefinierbare Makros	
Datenbanküberwachung-Datenpunkt *Felder für Benutzername, Passwort, SQL-Abfrage*	Benutzerdefinierbare Makros	
HTTP-Agent-Datenpunkt *Felder für URL, Abfragefelder, Anforderungstext, Header (Namen und Werte), SSL-Zertifikatsdatei, SSL-Schlüsseldatei*	Host-Makros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Datenpunkt-Makros: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} Benutzerdefinierbare Makros	URLs, die ein geheimes Makro enthalten, funktionieren nicht, da das Makro darin als ”******” aufgelöst wird
HTTP-Agent-Datenpunkt *Feld HTTP-Proxy*	Datenpunkt-Makros: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} Benutzerdefinierbare Makros	
HTTP-Agent-Datenpunkt *Felder für erforderliche Statuscodes, HTTP-Authentifizierungsbenutzername/-passwort, SSL-Schlüsselpasswort*	Benutzerdefinierbare Makros	
HTTP-Agent-Datenpunkt *Feld Zulässige Hosts*	Host-Makros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME} Benutzerdefinierbare Makros	
JMX-Agent-Datenpunkt *Feld Endpunkt*	Host-Makros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.PORT} Benutzerdefinierbare Makros	

Ort	Unterstützte Makros	Kommentare
Trapper-Datenpunkt *Feld Zulässige Hosts*	Host-Makros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP} Benutzerdefinierbare Makros	
SNMP-Agent-Datenpunkt *Feld SNMP OID*	Benutzerdefinierbare Makros	
SSH- und Telnet-Datenpunkt *Skript*	Host-Makros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Benutzerdefinierbare Makros	
SSH-Agent-Datenpunkt *Felder für Benutzername, öffentliche Schlüsseldatei, private Schlüsseldatei, Passwort*	Benutzerdefinierbare Makros	
Telnet-Agent-Datenpunkt *Felder für Benutzername, Passwort*	Benutzerdefinierbare Makros	
Karte		
URL, URL-Name	Host-Makros: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} {HOSTGROUP.ID} Makros für Host-Inventar Makros für Karte {TRIGGER.ID}	
Elementbeschriftung	Ausdrucksmakro: {?EXPRESSION} [1] Host-Makros: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} {HOSTGROUP.ID} Makros für Host-Inventar Makros für Karte Auslöser-Makros: {TRIGGER.EVENTS.*}, {TRIGGER.ID}, {TRIGGER.PROBLEM.EVENTS.*}, {TRIGGERS.ACK}, {TRIGGERS.PROBLEM.ACK}, {TRIGGERS.PROBLEM.UNACK}, {TRIGGERS.UNACK}	[1] In diesem Makro an dieser Stelle wird nur eine einzelne Funktion **avg**, **last**, **max** oder **min** mit Sekunden als Parameter unterstützt (Zeit-Suffixe können verwendet werden). Das Makro {HOST.HOST<1-9>} kann innerhalb des Makros als Host verwendet werden. Beispielverwendung: {?avg(/{HOST.HOST}/item.key,1h)} Nicht unterstützte Verwendung: {?last(/host/item1)/last(/host/item2)} {?last(/host/item1)*10} {?count(/host/item1,5m)}
Link-Beschriftung	Ausdrucksmakro: {?EXPRESSION} [1]	
Formbeschriftung Textfeld in Kartenformen	Ausdrucksmakro: {?EXPRESSION} [1] {MAP.NAME}	
Netzwerkerkennung		
Aktualisierungsintervall-Feld	Benutzerdefinierbare Makros (nur global)	Ein einzelnes Makro muss das gesamte Feld ausfüllen; mehrere Makros bzw. eine Mischung mit Text werden nicht unterstützt.

Ort	Unterstützte Makros	Kommentare
SNMP-Community, *SNMP OID*-Felder *SNMPv3-Kontextname*, *Sicherheitsname*, *Authentifizierungs-Passphrase*, *Datenschutz-Passphrase*-Felder	Benutzerdefinierbare Makros (nur global)	
Proxy		
Adresse für aktive Agenten > *Port* (wenn der Proxy zu einer Gruppe gehört) Für den passiven Proxy-Modus: Schnittstellenfelder *Adresse* und *Port* Felder *Timeouts für Datenpunkttypen* im Überschreibungsmodus	Benutzerdefinierbare Makros (nur global)	Ein einzelnes Makro muss das gesamte Feld ausfüllen; mehrere Makros bzw. eine Mischung mit Text werden nicht unterstützt.
Proxy-Gruppe		
Failover-Zeitraum, *Mindestanzahl von Proxys* Felder	Benutzerdefinierbare Makros (nur global)	Ein einzelnes Makro muss das gesamte Feld ausfüllen; mehrere Makros bzw. eine Mischung mit Text werden nicht unterstützt.
Skripte		
Befehle, *Bestätigungstext* (Typ: Skript, manuelle **Host-Aktion**)	Host-Makros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID} [1], {HOST.IP}, {HOST.NAME}, {HOST.PORT} Makros für Host-Inventar [2] {MANUALINPUT} Makros für Benutzername Benutzerdefinierbare Makros	[1] Nur Bestätigungstext [2] Unterstützt für Zabbix-Server und Zabbix-Proxy

Ort	Unterstützte Makros	Kommentare
Befehle (Typ: Skript, manuelle **Ereignisaktion**)	Makros für Datum/Uhrzeit Makros für Ereignis, Makros für Ursache/Symptom-Ereignis, Makros für Wiederherstellungsereignis (falls eine Wiederherstellung stattgefunden hat) Makros für Ereignisaktualisierung: {EVENT.UPDATE.HISTORY}, {EVENT.UPDATE.STATUS} {FUNCTION.VALUE} Host-Makros: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Makros für Host-Inventar Makros für Datenpunkt (außer {ITEM.STATE}, {ITEM.STATE.ERROR}) {MANUALINPUT} Makros für Proxy Auslöser-Makros: {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} Makros für Benutzername Benutzerdefinierbare Makros	
Bestätigungstext (Typ: Skript, manuelle **Ereignisaktion**)	Makros für Ereignis: {EVENT.ID}, {EVENT.NAME}, {EVENT.NSEVERITY}, {EVENT.SEVERITY}, {EVENT.STATUS}, {EVENT.VALUE} Host-Makros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Makros für Host-Inventar {MANUALINPUT} Makros für Benutzername Benutzerdefinierbare Makros	
URL, Bestätigungstext (Typ: URL, manuelle **Host-/Ereignisaktion**)		
Tag-Name und -Wert	Host-Makros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Makros für Host-Inventar Datenpunkt-Makros: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Auslöser-Makros: {TRIGGER.ID} (nur Trigger-Tag-Wert) Benutzerdefinierbare Makros	Tag-Name- und -Wert-Makros werden nur während der Ereigniserzeugung aufgelöst.
Auslöser		
Name	Host-Makros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Datenpunkt-Makros: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Positionsmakros/-referenzen: $1...$9 Benutzerdefinierbare Makros	
Ereignisname	Makros für Datum/Uhrzeit: {TIME}, {TIMESTAMP} Ausdrucksmakro: {?EXPRESSION} Makros für Funktion Host-Makros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Datenpunkt-Makros: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Auslöser-Makros: {TRIGGER.EXPRESSION.EXPLAIN}	

Ort	Unterstützte Makros	Kommentare
Betriebsdaten	Host-Makros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Datenpunkt-Makros: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Benutzerdefinierbare Makros	
Ausdruck	{TRIGGER.VALUE} Benutzerdefinierbare Makros [1]	[1] Nur in Konstanten und Funktionsparametern; geheime Makros werden nicht unterstützt
Felder für Tag zum Abgleich, Name des Menüeintrags, URL des Menüeintrags	Benutzerdefinierbare Makros	URLs, die ein geheimes Makro enthalten, funktionieren nicht, da das Makro darin als "******" aufgelöst wird.
Beschreibung	Host-Makros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Datenpunkt-Makros: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Benutzerdefinierbare Makros	
URL	{EVENT.ID} Host-Makros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Datenpunkt-Makros: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} {TRIGGER.ID}	
Benutzerparameter-Befehle	Positionsmakros/-referenzen: $1...$9	
Web-Szenario		
Name, Agent, HTTP-Proxy Felder	Benutzerdefinierbare Makros	
Aktualisierungsintervall-Feld	Benutzerdefinierbare Makros	Ein einzelnes Makro muss das gesamte Feld ausfüllen; mehrere Makros bzw. eine Mischung mit Text werden nicht unterstützt.
Variablen, Header, SSL-Zertifikat, SSL-Schlüsseldatei Felder	Host-Makros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Benutzerdefinierbare Makros	
Benutzer, Passwort, SSL-Schlüsselpasswort Felder	Benutzerdefinierbare Makros	
Web-Szenario-Schritt *Name, Variablen (nur Werte)* Felder	Benutzerdefinierbare Makros	
Web-Szenario-Schritt *Timeout, erforderliche Statuscodes* Felder	Benutzerdefinierbare Makros	Ein einzelnes Makro muss das gesamte Feld ausfüllen; mehrere Makros bzw. eine Mischung mit Text werden nicht unterstützt.
Web-Szenario-Schritt *URL, Header (Namen und Werte), erforderliche Zeichenfolge* Felder	Host-Makros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Benutzerdefinierbare Makros	

Ort	Unterstützte Makros	Kommentare
Web-Szenario-Schritt *Post*-Feld	Host-Makros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT}	
webhook-Medientyp		
Parameterwerte	Makros für Alert Alle Makros, die in auslöserbasierten Problembenachrichtigungen unterstützt werden (siehe oben)	Ein einzelnes Makro muss das gesamte Feld ausfüllen; mehrere Makros bzw. eine Mischung mit Text werden nicht unterstützt.
JavaScript-Skript, *JavaScript*-Skriptparametername	Benutzerdefinierbare Makros (nur global)	
Menüeintragsname, Menüeintrags-URL	{EVENT.TAGS.<tag name>}	
Widgets		
Gauge-Widget *Beschreibung* Feld	Host-Makros: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Makros für Host-Inventar Makros für Datenpunkt (außer {ITEM.STATE.ERROR}) Benutzerdefinierbare Makros (nur global)	
Honeycomb-Widget *Primäres/sekundäres Beschriftungsfeld*		
Datenpunktwert-Widget *Beschreibung* Feld		
Top-Hosts-Widget *Spalte Textdaten*	Host-Makros: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Makros für Host-Inventar Benutzerdefinierbare Makros (nur global)	
URL-Widget (dynamisch) *URL* Feld	Host-Makros: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Benutzerdefinierbare Makros	
*__Spalte "Benutzerdefinierter Text"__		
Überwachung > Probleme	Makros für Host (außer {HOST.METADATA}) Makros für Host-Inventar Makros für Datenpunkt (außer {ITEM.STATE.ERROR}) Auslöser-Makros: {TRIGGER.ID} Makros für Ereignisse: {EVENT.ID} Benutzerdefinierbare Makros	
Überwachung > Hosts	Makros für Host (außer {HOST.METADATA}) Makros für Host-Inventar Benutzerdefinierbare Makros	
Überwachung > Neueste Daten	Makros für Host (außer {HOST.METADATA}) Makros für Host-Inventar Makros für Datenpunkt (außer {ITEM.STATE.ERROR}) Benutzerdefinierbare Makros	
Datenerfassung > Vorlagen	Host-Makros: {HOST.DESCRIPTION}, {HOST.HOST}, {HOST.ID}, {HOST.NAME}	
Datenerfassung > Hosts	Makros für Host (außer {HOST.METADATA}) Makros für Host-Inventar Benutzerdefinierbare Makros	
Andere Orte		

Ort	Unterstützte Makros	Kommentare
Arbeitszeit-Feld	Benutzerdefinierbare Makros (nur global)	Ein einzelnes Makro muss das gesamte Feld ausfüllen; mehrere Makros bzw. eine Mischung mit Text werden nicht unterstützt.
Feld Zeitüberschreitungen für Datenpunkttypen (globale Einstellung)		
Benutzer-Medien *Wenn aktiv* Feld	Benutzerdefinierbare Makros (nur global)	
E-Mail-Medientyp *Benutzername, Passwort* Felder	Benutzerdefinierbare Makros (nur global)	Geheime Makros werden empfohlen.
Geplante Berichte *Betreff, Nachricht* Felder	Makros für Datum/Uhrzeit: {TIME}, {TIMESTAMP}	

Makrodetails

Diese Liste enthält Details zu jedem integrierten Makro, gruppiert nach Anwendungsbereich:

- Aktionsmakros
- Alarmmakros
- Datums-/Uhrzeitmakros
- Erkennungsmakros
- Ereignismakros
  - Makros für Ereignisaktualisierungen
  - Makros für Ursache-/Symptomereignisse
  - Makros für Wiederherstellungsereignisse
- Eskalationsmakros
- Funktionsmakros
- Host-Makros
  - Makros für Ziel-Hosts
- Hostgruppen-Makros
- Hostinventar-Makros
- Datenpunkt-Makros
- Benachrichtigungsmakros für Low-Level-Discovery
- Kartenmakros
- Proxy-Makros
- Skriptmakros
- Service-Makros
- Auslöser-Makros
- Benutzernamen-Makros

Aktionsmakros

{ACTION.ID}

Die numerische ID der ausgelösten Aktion.

{ACTION.NAME}

Der Name der ausgelösten Aktion.

Alarmmakros

{ALERT.MESSAGE}

Der Wert *Standardnachricht* aus der Aktionskonfiguration.

{ALERT.SENDTO}

Der *Send to*-Wert aus der Medienkonfiguration des Benutzers.

{ALERT.SUBJECT}

Der Wert *Standardbetreff* aus der Aktionskonfiguration.

Datums- und Zeitmakros

{DATE}

Das aktuelle Datum im Format `yyyy.mm.dd`.

{TIME}

Die aktuelle Uhrzeit im Format `hh:mm:ss`.

{TIMESTAMP}

Die aktuelle Zeit im UNIX-Zeitstempelformat.

Discovery-Makros

{DISCOVERY.DEVICE.IPADDRESS}

Die IP-Adresse des erkannten Geräts.
 Immer verfügbar, hängt nicht davon ab, ob der Host hinzugefügt wurde.

{DISCOVERY.DEVICE.DNS}

Der DNS-Name des erkannten Geräts.
 Immer verfügbar, hängt nicht davon ab, ob der Host hinzugefügt wurde.

{DISCOVERY.DEVICE.STATUS}

Der Status des erkannten Geräts (UP/DOWN).

{DISCOVERY.DEVICE.UPTIME}

Die Zeit seit der letzten Änderung des Discovery-Status für ein bestimmtes Gerät.
 Zum Beispiel: 1h 29m 01s.
 Bei Geräten mit dem Status DOWN ist dies die Dauer ihrer Ausfallzeit.

{DISCOVERY.RULE.NAME}

Der Name der Discovery-Regel, die das Vorhandensein oder Fehlen eines Geräts/Services erkannt hat.

{DISCOVERY.SERVICE.NAME}

Der Name des erkannten Dienstes.
 Zum Beispiel: HTTP.

{DISCOVERY.SERVICE.PORT}

Der Port des erkannten Dienstes.
 Zum Beispiel: 80.

{DISCOVERY.SERVICE.STATUS}

Der Status des erkannten Dienstes (UP/DOWN).

{DISCOVERY.SERVICE.UPTIME}

Die Zeit seit der letzten Änderung des Discovery-Status für einen bestimmten Service.
 Zum Beispiel: 1h 29m 01s.
 Für Services mit dem Status DOWN ist dies die Dauer ihrer Ausfallzeit.

Eskalationsmakros

{ESC.HISTORY}

Das Protokoll der zuvor gesendeten Benachrichtigungen, ihres Eskalationsschritts und ihres Status (*gesendet*, *in Bearbeitung* oder *fehlgeschlagen*).

Ereignismakros

{EVENT.ACK.STATUS}

Der Bestätigungsstatus des Ereignisses (Ja/Nein).

{EVENT.AGE}

Das Alter des Ereignisses, das eine Aktion ausgelöst hat, mit einer Genauigkeit bis auf eine Sekunde.
 Nützlich in eskalierten Nachrichten. Zum Beispiel: 9m 13s.

{EVENT.DATE}

Das Datum des Ereignisses, das eine Aktion ausgelöst hat, im Format `yyyy.mm.dd`.
 Zum Beispiel: 2025.04.14.

{EVENT.DURATION}

Die Dauer des Ereignisses (Zeitdifferenz zwischen Problem- und Wiederherstellungsereignissen), mit einer Genauigkeit bis auf eine Sekunde.
 Nützlich in Problem-Wiederherstellungsmeldungen.

{EVENT.ID}

Die numerische ID des Ereignisses, das eine Aktion ausgelöst hat.

{EVENT.NAME}

Der Name des Ereignisses, das eine Aktion ausgelöst hat.

{EVENT.NSEVERITY}

Der numerische Wert des Ereignisschweregrads.
 Mögliche Werte: *0* - Nicht klassifiziert, *1* - Information, *2* - Warnung, *3* - Durchschnittlich, *4* - Hoch, *5* - Katastrophe.

{EVENT.OBJECT}

Der numerische Wert des Ereignisobjekts.
 Mögliche Werte: *0* - Auslöser, *1* - Entdeckter Host, *2* - Entdeckter Dienst, *3* - Autoregistrierung, *4* - Datenpunkt, *5* - Low-level-Discovery-Regel.

{EVENT.OPDATA}

Die Betriebsdaten des zugrunde liegenden Auslösers des Problems.

{EVENT.SEVERITY}

Der Name des Ereignisschweregrads.

{EVENT.SOURCE}

Der numerische Wert der Ereignisquelle.
 Mögliche Werte: *0* - Auslöser, *1* - Discovery, *2* - Autoregistrierung, *3* - Intern, *4* - Service.

{EVENT.STATUS}

Der verbale Wert des Ereignisses, das eine Aktion ausgelöst hat.

{EVENT.TAGS}

Eine durch Kommas getrennte Liste von Event-Tags.
Wird zu einer leeren Zeichenkette erweitert, wenn keine Tags vorhanden sind.

{EVENT.TAGSJSON}

Ein JSON-Array, das Ereignis-Tag-Objekte enthält.
Wird zu einem leeren Array erweitert, wenn keine Tags vorhanden sind.

{EVENT.TAGS.<tag name>}

Der Wert des Ereignis-Tags, auf den über den Tag-Namen verwiesen wird.
 Ein Tag-Name, der nicht alphanumerische Zeichen enthält (einschließlich nicht-englischer Multibyte-UTF-Zeichen), muss in doppelte Anführungszeichen gesetzt werden. Anführungszeichen und Backslashes innerhalb eines in Anführungszeichen gesetzten Tag-Namens müssen mit einem Backslash maskiert werden.

{EVENT.TIME}

Die Uhrzeit des Ereignisses, das eine Aktion ausgelöst hat, im Format `hh:mm:ss`.
 Zum Beispiel: 12:57:53.

{EVENT.TIMESTAMP}

Der UNIX-Zeitstempel des Ereignisses, das eine Aktion ausgelöst hat.

{EVENT.VALUE}

Der numerische Wert des Ereignisses, das eine Aktion ausgelöst hat.
 Mögliche Werte: *1* - Problem, *0* - Wiederherstellung.

Makros für Ursache-/Symptomereignisse

{EVENT.CAUSE.ACK.STATUS}

Der Bestätigungsstatus des Ursache-Ereignisses (Ja/Nein).
 Wird im Kontext eines Symptomereignisses verwendet.

{EVENT.CAUSE.AGE}

Das Alter des Ursache-Ereignisses, mit einer Genauigkeit bis auf eine Sekunde.
 Nützlich in eskalierten Nachrichten.
 Wird im Kontext eines Symptom-Ereignisses verwendet.

{EVENT.CAUSE.DATE}

Das Datum des verursachenden Ereignisses im Format `yyyy.mm.dd`.
 Wird im Kontext eines Symptomereignisses verwendet.

{EVENT.CAUSE.DURATION}

Die Dauer des verursachenden Ereignisses (Zeitdifferenz zwischen Problem- und Wiederherstellungsereignissen), mit einer Genauigkeit bis auf eine Sekunde.
 Nützlich in Problem-Wiederherstellungsmeldungen.
 Wird im Kontext eines Symptomereignisses verwendet.

{EVENT.CAUSE.ID}

Die numerische ID des verursachenden Ereignisses.
 Wird im Kontext eines Symptomereignisses verwendet.

{EVENT.CAUSE.NAME}

Der Name des verursachenden Ereignisses.
 Wird im Kontext eines Symptomereignisses verwendet.

{EVENT.CAUSE.NSEVERITY}

Der numerische Wert des Schweregrads des Ursache-Ereignisses.
 Mögliche Werte: *0* - Nicht klassifiziert, *1* - Information, *2* - Warnung, *3* - Durchschnitt, *4* - Hoch, *5* - Katastrophe.
 Wird im Kontext eines Symptom-Ereignisses verwendet.

{EVENT.CAUSE.OBJECT}

Der numerische Wert des Objekttyps des verursachenden Ereignisses.
 Mögliche Werte: *0* - Auslöser, *1* - Erkannter Host, *2* - Erkannter Dienst, *3* - Autoregistrierung, *4* - Datenpunkt, *5* - Low-level-Discovery-Regel.
 Wird im Kontext eines Symptomereignisses verwendet.

{EVENT.CAUSE.OPDATA}

Die Betriebsdaten des zugrunde liegenden Auslösers des verursachenden Problems.
 Wird im Kontext eines Symptom-Ereignisses verwendet.

{EVENT.CAUSE.SEVERITY}

Der Name des Schweregrads des Ursache-Ereignisses.
 Mögliche Werte: *Nicht klassifiziert*, *Information*, *Warnung*, *Durchschnittlich*, *Hoch*, *Katastrophe*.
 Wird im Kontext eines Symptom-Ereignisses verwendet.

{EVENT.CAUSE.SOURCE}

Der numerische Wert der Quelle des verursachenden Ereignisses.
 Mögliche Werte: *0* - Auslöser, *1* - Discovery, *2* - Autoregistrierung, *3* - Intern.
 Wird im Kontext eines Symptomereignisses verwendet.

{EVENT.CAUSE.STATUS}

Der verbale Wert des verursachenden Ereignisses.
 Wird im Kontext eines Symptomereignisses verwendet.

{EVENT.CAUSE.TAGS}

Eine durch Kommas getrennte Liste von Tags des verursachenden Ereignisses.
Wird zu einer leeren Zeichenfolge erweitert, wenn keine Tags vorhanden sind.
 Wird im Kontext eines Symptomereignisses verwendet.

{EVENT.CAUSE.TAGSJSON}

Ein JSON-Array, das Tag-Objekte des verursachenden Ereignisses enthält.
Wird zu einem leeren Array erweitert, wenn keine Tags vorhanden sind.
 Wird im Kontext eines Symptomereignisses verwendet.

{EVENT.CAUSE.TAGS.<tag name>}

Der Wert des Ursache-Ereignis-Tags, auf den über den Tag-Namen verwiesen wird.
 Ein Tag-Name, der nicht alphanumerische Zeichen enthält (einschließlich nicht-englischer Multibyte-UTF-Zeichen), muss in doppelte Anführungszeichen gesetzt werden. Anführungszeichen und Backslashes innerhalb eines in Anführungszeichen gesetzten Tag-Namens müssen mit einem Backslash maskiert werden.
 Wird im Kontext eines Symptom-Ereignisses verwendet.

{EVENT.CAUSE.TIME}

Die Uhrzeit des verursachenden Ereignisses im Format `hh:mm:ss`.
 Wird im Kontext eines Symptomereignisses verwendet.

{EVENT.CAUSE.TIMESTAMP}

Der UNIX-Zeitstempel des Ursache-Ereignisses.
 Wird im Kontext eines Symptom-Ereignisses verwendet.

{EVENT.CAUSE.UPDATE.HISTORY}

Das Protokoll der Aktualisierungen des Ursacheproblems (Bestätigungen usw.).
 Wird im Kontext eines Symptomereignisses verwendet.

{EVENT.CAUSE.VALUE}

Der numerische Wert des Ursache-Ereignisses.
 Mögliche Werte: *1* - Problem, *0* - Wiederherstellung.
 Wird im Kontext eines Symptom-Ereignisses verwendet.

{EVENT.SYMPTOMS}

Die Liste der Symptomereignisse. Enthält die folgenden Details: Hostname, Ereignisname, Schweregrad, Alter, Servicetags und Werte.
 Dieses Makro wird im Kontext des Ursache-Ereignisses verwendet und gibt Informationen zu Symptomereignissen zurück.

Makros für Wiederherstellungsereignisse

{EVENT.RECOVERY.DATE}

Das Datum des Wiederherstellungsereignisses im Format `yyyy.mm.dd`.

{EVENT.RECOVERY.ID}

Die numerische ID des Wiederherstellungsereignisses.

{EVENT.RECOVERY.NAME}

Der Name des Wiederherstellungsereignisses.

{EVENT.RECOVERY.STATUS}

Der verbale Wert des Wiederherstellungsereignisses.

{EVENT.RECOVERY.TAGS}

Eine durch Kommas getrennte Liste von Tags des Wiederherstellungsereignisses. Wird zu einer leeren Zeichenfolge erweitert, wenn keine Tags vorhanden sind.

{EVENT.RECOVERY.TAGSJSON}

Ein JSON-Array, das Wiederherstellungs-Ereignis-Tag-Objekte enthält. Wird zu einem leeren Array erweitert, wenn keine Tags vorhanden sind.

{EVENT.RECOVERY.TIME}

Die Uhrzeit des Wiederherstellungsereignisses im Format `hh:mm:ss`.

{EVENT.RECOVERY.TIMESTAMP}

Der UNIX-Zeitstempel des Wiederherstellungsereignisses.

{EVENT.RECOVERY.VALUE}

Der numerische Wert des Wiederherstellungsereignisses.

Makros zur Ereignisaktualisierung

{EVENT.UPDATE.ACTION}

Lesbarer Name der während der Problemaktualisierung ausgeführten Aktion(en).
 Löst sich in die folgenden Werte auf: *acknowledged*, *unacknowledged*, *commented*, *changed severity from (original severity) to (updated severity)* und *closed* (abhängig davon, wie viele Aktionen in einer Aktualisierung ausgeführt werden).

{EVENT.UPDATE.ACTIONJSON}

Ein JSON-Array mit Details zu den während der Problemaktualisierung ausgeführten Aktion(en).
 Mögliche JSON-Property-Werte:
- true (für die Properties `acknowledge`, `unacknowledge`, `close`, `unsuppress`, `cause` und `symptom`);
- <message string> (für die Property `message`);
- Zeitstempel (für die Properties `suppress_until` und `timestamp`) oder 0 (für die Property `suppress_until`, wenn die Unterdrückung unbegrenzt ist);
- 0, 1, 2, 3, 4, 5 (für die Schweregrad-Properties `old` und `new`).

 Zum Beispiel: `{"acknowledge":true,"message":"Monthly maintenance.","severity":{"old":2,"new":1},'`

{EVENT.UPDATE.DATE}

Das Datum der Aktualisierung des Ereignisses (Bestätigung usw.) im Format `yyyy.mm.dd`.

{EVENT.UPDATE.HISTORY}

Das Protokoll der Problemaktualisierungen (Bestätigungen usw.).

{EVENT.UPDATE.MESSAGE}

Die Problemaktualisierungsnachricht.

{EVENT.UPDATE.NSEVERITY}

Der numerische Wert des neuen Ereignisschweregrads, der während der Problemaktualisierung festgelegt wurde.

{EVENT.UPDATE.SEVERITY}

Der Name des neuen Ereignisschweregrads, der während des Problemaktualisierungsvorgangs festgelegt wurde.

{EVENT.UPDATE.STATUS}

Der numerische Wert des Aktualisierungsstatus des Problems.
 Mögliche Werte: *0* - webhook wurde aufgrund eines Problem-/Wiederherstellungsereignisses aufgerufen, *1* - Aktualisierungsvorgang.

{EVENT.UPDATE.TIME}

Die Uhrzeit der Aktualisierung des Ereignisses (Bestätigung usw.) im Format `hh:mm:ss`.

{EVENT.UPDATE.TIMESTAMP}

Der UNIX-Zeitstempel der Aktualisierung des Ereignisses (Bestätigung usw.).

Funktionsmakros

{FUNCTION.VALUE}

Der Wert der N-ten auf einem Datenpunkt basierenden Funktion im Auslöserausdruck zum Zeitpunkt des Ereignisses.
 Es werden nur Funktionen mit `/host/key` als erstem Parameter gezählt.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {FUNCTION.VALUE<1-9>}, um auf die erste, zweite, dritte usw. Funktion in einem Auslöserausdruck zu verweisen. Siehe indexierte Makros.

{FUNCTION.RECOVERY.VALUE}

Der Wert der N-ten auf einem Datenpunkt basierenden Funktion in der Wiederherstellungsausdruck zum Zeitpunkt des Ereignisses.*
 Es werden nur Funktionen mit `/host/keys` als erstem Parameter gezählt.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {FUNCTION.RECOVERY.VALUE<1-9>}, um auf die erste, zweite, dritte usw. Funktion in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

Host-Makros

{HOST.CONN}

Die IP-Adresse oder der DNS-Name des Hosts, abhängig von den Host-Einstellungen.
 In globalen Skripten, in den Feldern für Interface-IP/DNS und in Web-Szenarien wird das Makro zum Haupt-Agent-Interface aufgelöst. Wenn kein Agent-Interface definiert ist, wird das Haupt-SNMP-Interface verwendet. Wenn auch kein SNMP-Interface definiert ist, wird das Haupt-JMX-Interface verwendet. Wenn auch kein JMX-Interface definiert ist, wird das Haupt-IPMI-Interface verwendet. Wenn der Host über kein Interface verfügt, wird das Makro zu 'UNKNOWN' aufgelöst.

 Dieses Makro kann mit einem numerischen Index als {HOST.CONN<1-9>} verwendet werden, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{HOST.DESCRIPTION}

Die Host-Beschreibung.

 Dieses Makro kann mit einem numerischen Index als {HOST.DESCRIPTION<1-9>} verwendet werden, um auf den ersten, zweiten, dritten usw. Host in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{HOST.DNS}

Der DNS-Name des Hosts.
 In globalen Skripten, in den Feldern für Schnittstellen-IP/DNS und in Web-Szenarien wird das Makro auf die primäre Agent-Schnittstelle aufgelöst. Wenn keine Agent-Schnittstelle definiert ist, wird die primäre SNMP-Schnittstelle verwendet. Wenn auch keine SNMP-Schnittstelle definiert ist, wird die primäre JMX-Schnittstelle verwendet. Wenn auch keine JMX-Schnittstelle definiert ist, wird die primäre IPMI-Schnittstelle verwendet. Wenn der Host keine Schnittstelle hat, wird das Makro in 'UNKNOWN' aufgelöst.

 Dieses Makro kann mit einem numerischen Index als {HOST.DNS<1-9>} verwendet werden, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{HOST.HOST}

Der technische Hostname.
 Makrofunktionen werden für dieses Makro nicht unterstützt, wenn es als Platzhalter im ersten Parameter einer History-Funktion verwendet wird, zum Beispiel `last(/{HOST.HOST}/{ITEM.KEY})`.

 Dieses Makro kann mit einem numerischen Index als {HOST.HOST<1-9>} verwendet werden, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{HOST.ID}

Die Host-ID.

 Dieses Makro kann mit einem numerischen Index als {HOST.ID<1-9>} verwendet werden, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{HOST.IP}

Die IP-Adresse des Hosts.
 In globalen Skripten, in den Feldern für Schnittstellen-IP/DNS und in Webszenarien wird das Makro zur Haupt-Agent-Schnittstelle aufgelöst. Wenn keine Agent-Schnittstelle definiert ist, wird die Haupt-SNMP-Schnittstelle verwendet. Wenn auch keine SNMP-Schnittstelle definiert ist, wird die Haupt-JMX-Schnittstelle verwendet. Wenn auch keine JMX-Schnittstelle

nicht definiert ist, wird die Haupt-IPMI-Schnittstelle verwendet. Wenn der Host keine Schnittstelle hat, wird das Makro zu 'UNKNOWN' aufgelöst.

 Dieses Makro kann mit einem numerischen Index als {HOST.IP<1-9>} verwendet werden, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{HOST.METADATA}

Die Host-Metadaten.

 Wird nur für die aktive Agent-Autoregistrierung verwendet.

{HOST.NAME}

Der sichtbare Hostname.

 Dieses Makro kann mit einem numerischen Index als {HOST.NAME<1-9>} verwendet werden, um auf den ersten, zweiten, dritten usw. Host in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{HOST.PORT}

Der Port des Hosts (Agent).
 In globalen Skripten, in den Feldern für die Schnittstellen-IP/DNS und in Webszenarien wird das Makro zur Hauptschnittstelle des Agent aufgelöst. Wenn keine Agent-Schnittstelle definiert ist, wird die Hauptschnittstelle von SNMP verwendet. Wenn auch keine SNMP-Schnittstelle definiert ist, wird die Hauptschnittstelle von JMX verwendet. Wenn auch keine JMX-Schnittstelle definiert ist, wird die Hauptschnittstelle von IPMI verwendet. Wenn der Host keine Schnittstelle hat, wird das Makro zu 'UNKNOWN' aufgelöst.

 Dieses Makro kann mit einem numerischen Index als {HOST.PORT<1-9>} verwendet werden, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

Makros des Ziel-Hosts

{HOST.TARGET.CONN}

Die IP-Adresse oder der DNS-Name des Ziel-Hosts.

{HOST.TARGET.DNS}

Der DNS-Name des Ziel-Hosts.

{HOST.TARGET.HOST}

Der technische Name des Ziel-Hosts.

{HOST.TARGET.IP}

Die IP-Adresse des Ziel-Hosts.

{HOST.TARGET.NAME}

Der sichtbare Name des Ziel-Hosts.

Host-Gruppen-Makros

{HOSTGROUP.ID}

Die Hostgruppen-ID.

Host-Inventar-Makros

{INVENTORY.ALIAS}

Das Feld *Alias* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.ALIAS<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.ASSET.TAG}

Das Feld *Asset tag* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.ASSET.TAG<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.CHASSIS}

Das Feld *Chassis* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.CHASSIS<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.CONTACT}

Das Feld *Contact* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.CONTACT<1-9>}, um auf den ersten, zweiten, dritten usw. Host in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{INVENTORY.CONTRACT.NUMBER}

Das Feld *Vertragsnummer* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.CONTRACT.NUMBER<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.DEPLOYMENT.STATUS}

Das Feld *Deployment status* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.DEPLOYMENT.STATUS<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.HARDWARE}

Das Feld *Hardware* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.HARDWARE<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.HARDWARE.FULL}

Das Feld *Hardware (vollständige Details)* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.HARDWARE.FULL<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.HOST.NETMASK}

Das Feld *Host-Subnetzmaske* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.HOST.NETMASK<1-9>}, um auf den ersten, zweiten, dritten usw. Host in einem Auslöser-Ausdruck zu verweisen. Siehe indizierte Makros.

{INVENTORY.HOST.NETWORKS}

Das Feld *Host networks* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.HOST.NETWORKS<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.HOST.ROUTER}

Das Feld *Host router* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.HOST.ROUTER<1-9>}, um auf den ersten, zweiten, dritten usw. Host in einem Auslöser-Ausdruck zu verweisen. Siehe indizierte Makros.

{INVENTORY.HW.ARCH}

Das Feld *Hardwarearchitektur* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.HW.ARCH<1-9>}, um auf den ersten, zweiten, dritten usw. Host in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{INVENTORY.HW.DATE.DECOMM}

Das Feld *Datum der Außerbetriebnahme der Hardware* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.HW.DATE.DECOMM<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.HW.DATE.EXPIRY}

Das Feld *Ablaufdatum der Hardware-Wartung* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.HW.DATE.EXPIRY<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.HW.DATE.INSTALL}

Das Feld *Datum der Hardware-Installation* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.HW.DATE.INSTALL<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.HW.DATE.PURCHASE}

Das Feld *Datum des Hardwarekaufs* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.HW.DATE.PURCHASE<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.INSTALLER.NAME}

Das Feld *Installer name* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.INSTALLER.NAME<1-9>}, um auf den ersten, zweiten, dritten usw. Host in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{INVENTORY.LOCATION}

Das Feld *Location* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.LOCATION<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.LOCATION.LAT}

Das Feld *Standortbreitengrad* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.LOCATION.LAT<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.LOCATION.LON}

Das Feld *Längengrad des Standorts* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.LOCATION.LON<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.MACADDRESS.A}

Das Feld *MAC address A* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.MACADDRESS.A<1-9>}, um auf den ersten, zweiten, dritten usw. Host in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{INVENTORY.MACADDRESS.B}

Das Feld *MAC address B* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.MACADDRESS.B<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.MODEL}

Das Feld *Model* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.MODEL<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.NAME}

Das Feld *Name* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.NAME<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.NOTES}

Das Feld *Notes* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.NOTES<1-9>}, um auf den ersten, zweiten, dritten usw. Host in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{INVENTORY.OOB.IP}

Das Feld *OOB-IP-Adresse* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.OOB.IP<1-9>}, um auf den ersten, zweiten, dritten usw. Host in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{INVENTORY.OOB.NETMASK}

Das Feld *OOB-Subnetzmaske* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.OOB.NETMASK<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.OOB.ROUTER}

Das Feld *OOB router* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.OOB.ROUTER<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.OS}

Das Feld *OS* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.OS<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.OS.FULL}

Das Feld *OS (vollständige Details)* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.OS.FULL<1-9>}, um auf den ersten, zweiten, dritten usw. Host in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{INVENTORY.OS.SHORT}

Das Feld *OS (Kurz)* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.OS.SHORT<1-9>}, um auf den ersten, zweiten, dritten usw. Host in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{INVENTORY.POC.PRIMARY.CELL}

Das Feld *Primary POC cell* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.POC.PRIMARY.CELL<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.POC.PRIMARY.EMAIL}

Das Feld *Primary POC email* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.POC.PRIMARY.EMAIL<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.POC.PRIMARY.NAME}

Das Feld *Name des primären Ansprechpartners* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.POC.PRIMARY.NAME<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.POC.PRIMARY.NOTES}

Das Feld *Notizen zum primären Ansprechpartner* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.POC.PRIMARY.NOTES<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.POC.PRIMARY.PHONE.A}

Das Feld *Primary POC phone A* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.POC.PRIMARY.PHONE.A<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.POC.PRIMARY.PHONE.B}

Das Feld *Primary POC phone B* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.POC.PRIMARY.PHONE.B<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.POC.PRIMARY.SCREEN}

Das Feld *Name des primären POC-Bildschirms* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.POC.PRIMARY.SCREEN<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.POC.SECONDARY.CELL}

Das Feld *Secondary POC cell* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.POC.SECONDARY.CELL<1-9>}, um auf den ersten, zweiten, dritten usw. Host in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{INVENTORY.POC.SECONDARY.EMAIL}

Das Feld *Secondary POC email* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.POC.SECONDARY.EMAIL<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.POC.SECONDARY.NAME}

Das Feld *Name des sekundären Ansprechpartners* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.POC.SECONDARY.NAME<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.POC.SECONDARY.NOTES}

Das Feld *Secondary POC notes* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.POC.SECONDARY.NOTES<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.POC.SECONDARY.PHONE.A}

Das Feld *Secondary POC phone A* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.POC.SECONDARY.PHONE.A<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.POC.SECONDARY.PHONE.B}

Das Feld *Secondary POC phone B* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.POC.SECONDARY.PHONE.B<1-9>}, um auf den ersten, zweiten, dritten usw. Host in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{INVENTORY.POC.SECONDARY.SCREEN}

Das Feld *Name des sekundären POC-Bildschirms* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.POC.SECONDARY.SCREEN<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.SERIALNO.A}

Das Feld *Seriennummer A* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.SERIALNO.A<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.SERIALNO.B}

Das Feld *Seriennummer B* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.SERIALNO.B<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.SITE.ADDRESS.A}

Das Feld *Site address A* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.SITE.ADDRESS.A<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.SITE.ADDRESS.B}

Das Feld *Site address B* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.SITE.ADDRESS.B<1-9>}, um auf den ersten, zweiten, dritten usw. Host in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{INVENTORY.SITE.ADDRESS.C}

Das Feld *Site address C* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.SITE.ADDRESS.C<1-9>}, um auf den ersten, zweiten, dritten usw. Host in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{INVENTORY.SITE.CITY}

Das Feld *Site city* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.SITE.CITY<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.SITE.COUNTRY}

Das Feld *Site country* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.SITE.COUNTRY<1-9>}, um auf den ersten, zweiten, dritten usw. Host in einem Ausdrück für einen Auslöser zu verweisen. Siehe indizierte Makros.

{INVENTORY.SITE.NOTES}

Das Feld *Site notes* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.SITE.NOTES<1-9>}, um auf den ersten, zweiten, dritten usw. Host in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{INVENTORY.SITE.RACK}

Das Feld *Standort des Racks* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.SITE.RACK<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.SITE.STATE}

Das Feld *Bundesland/Provinz des Standorts* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.SITE.STATE<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.SITE.ZIP}

Das Feld *Site ZIP/postal* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.SITE.ZIP<1-9>}, um auf den ersten, zweiten, dritten usw. Host in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{INVENTORY.SOFTWARE}

Das Feld *Software* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.SOFTWARE<1-9>}, um auf den ersten, zweiten, dritten usw. Host in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{INVENTORY.SOFTWARE.APP.A}

Das Feld *Softwareanwendung A* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.SOFTWARE.APP.A<1-9>}, um auf den ersten, zweiten, dritten usw. Host in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{INVENTORY.SOFTWARE.APP.B}

Das Feld *Softwareanwendung B* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.SOFTWARE.APP.B<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.SOFTWARE.APP.C}

Das Feld *Softwareanwendung C* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.SOFTWARE.APP.C<1-9>}, um auf den ersten, zweiten, dritten usw. Host in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{INVENTORY.SOFTWARE.APP.D}

Das Feld *Softwareanwendung D* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.SOFTWARE.APP.D<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.SOFTWARE.APP.E}

Das Feld *Softwareanwendung E* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.SOFTWARE.APP.E<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.SOFTWARE.FULL}

Das Feld *Software (vollständige Details)* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.SOFTWARE.FULL<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.TAG}

Das Feld *Tag* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.TAG<1-9>}, um auf den ersten, zweiten, dritten usw. Host in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{INVENTORY.TYPE}

Das Feld *Type* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.TYPE<1-9>}, um auf den ersten, zweiten, dritten usw. Host in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{INVENTORY.TYPE.FULL}

Das Feld *Typ (vollständige Details)* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.TYPE.FULL<1-9>}, um auf den ersten, zweiten, dritten usw. Host in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{INVENTORY.URL.A}

Das Feld *URL A* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.URL.A<1-9>}, um auf den ersten, zweiten, dritten usw. Host in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{INVENTORY.URL.B}

Das Feld *URL B* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.URL.B<1-9>}, um in einem Auslöserausdruck auf den ersten, zweiten, dritten usw. Host zu verweisen. Siehe indizierte Makros.

{INVENTORY.URL.C}

Das Feld *URL C* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.URL.C<1-9>}, um auf den ersten, zweiten, dritten usw. Host in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{INVENTORY.VENDOR}

Das Feld *Vendor* im Host-Inventar.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {INVENTORY.VENDOR<1-9>}, um auf den ersten, zweiten, dritten usw. Host in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

Datenpunkt-Makros

{ITEM.DESCRIPTION}

Die Beschreibung des N-ten Datenpunkts in der Auslöserausdruck, der eine Benachrichtigung ausgelöst hat.

 Dieses Makro kann mit einem numerischen Index als {ITEM.DESCRIPTION<1-9>} verwendet werden, um auf den ersten, zweiten, dritten usw. Host in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.DESCRIPTION.ORIG}

Die ursprüngliche Beschreibung (mit nicht aufgelösten Makros) des N-ten Datenpunkts im Auslöserausdruck, der eine Benachrichtigung ausgelöst hat.

 Dieses Makro kann mit einem numerischen Index als {ITEM.DESCRIPTION.ORIG<1-9>} verwendet werden, um auf den ersten, zweiten, dritten usw. Host in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.ID}

Die numerische ID des N-ten Datenpunkts in der Auslöser-Ausdruck, der eine Benachrichtigung ausgelöst hat.

 Dieses Makro kann mit einem numerischen Index als {ITEM.ID<1-9>} verwendet werden, um auf den ersten, zweiten, dritten usw. Host in einem Auslöser-Ausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.KEY}

Der Schlüssel des n-ten Datenpunkts in der Auslöser-Ausdruck, der eine Benachrichtigung ausgelöst hat.
 Makrofunktionen werden für dieses Makro nicht unterstützt, wenn es als Platzhalter im ersten Parameter einer History-Funktion verwendet wird, zum Beispiel `last(/{HOST.HOST}/{ITEM.KEY})`.

 Dieses Makro kann mit einem numerischen Index als {ITEM.KEY<1-9>} verwendet werden, um auf den ersten, zweiten, dritten usw. Host in einem Auslöser-Ausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.KEY.ORIG}

Der ursprüngliche Schlüssel (mit nicht aufgelösten Makros) des N-ten Datenpunkts in dem Auslöserausdruck, der eine Benachrichtigung ausgelöst hat.

 Dieses Makro kann mit einem numerischen Index als {ITEM.KEY.ORIG<1-9>} verwendet werden, um auf den ersten, zweiten, dritten usw. Host in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.LASTVALUE}

Der neueste Wert des N-ten Datenpunkts in dem Auslöserausdruck, der eine Benachrichtigung ausgelöst hat.
 Er wird im Frontend zu *UNKNOWN* aufgelöst, wenn der neueste Historienwert vor mehr als dem Zeitraum *Max history display period* erfasst wurde (festgelegt im Menübereich *Administration > General*).
 Wenn er im Problemnamen verwendet wird, wird das Makro beim Anzeigen von Problemereignissen nicht zum neuesten Datenpunktwert aufgelöst; stattdessen behält es den Datenpunktwert vom Zeitpunkt bei, als das Problem auftrat.
Wenn es in Benachrichtigungen verwendet wird, kann das Makro in einigen Fällen zum Zeitpunkt, an dem der Auslöser ausgelöst wurde, nicht zum neuesten Datenpunktwert aufgelöst werden. Wenn ein Datenpunkt beispielsweise schnell zwei Werte, ”A” und ”B”, empfängt und der Auslöser für ”A” ausgelöst wird, können Benachrichtigungen aufgrund einer leichten Verarbeitungsverzögerung ”B” als neuesten Wert anzeigen - der neueste Datenpunktwert hat sich zwischen dem Auslösen des Auslösers und der Erstellung der Benachrichtigung geändert. Um dies zu vermeiden, können Sie das Makro {ITEM.VALUE} verwenden, das sich auf den Wert zum Zeitpunkt des Auslösens des Auslösers auflöst und so sicherstellt, dass in der Benachrichtigung der richtige Wert verwendet wird.
 Es ist ein Alias für `last(/{HOST.HOST}/{ITEM.KEY})`.
 Der aufgelöste Wert für Text-/Log-Datenpunkte wird an den folgenden Stellen vom Frontend auf 20 Zeichen gekürzt:
- Betriebsdaten;
- Auslöserbeschreibung;
- Auslöser-URLs;
- Auslöser-URL-Beschriftungen;
- Beschreibung des Widgets für den Datenpunktwert.
Um den vollständigen Wert aufzulösen, können Sie Makrofunktionen verwenden, da vom Server keine Werte gekürzt werden. Zum Beispiel: `{{ITEM.LASTVALUE}.regsub("(.*)", \1)}`.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {ITEM.LASTVALUE<1-9>}, um auf den ersten, zweiten, dritten usw. Datenpunkt in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.LASTVALUE.AGE}

Die Zeit, die zwischen der letzten Erfassung des Datenpunktwerts und der Makroauswertung verstrichen ist.
 Wird in einem für Menschen lesbaren Format angezeigt (z. B. 1m 45s).

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {ITEM.LASTVALUE.AGE<1-9>}, um auf den ersten, zweiten, dritten usw. Datenpunkt in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.LASTVALUE.DATE}

Das Datum, an dem der letzte Datenpunktwert erfasst wurde.
 Wird im Format `YYYYMMDD` angezeigt.
 Die Server-Zeitzone wird in Auslösernamen (in der Liste *Monitoring* > *Probleme*), Ereignisnamen sowie in Tag-Namen und -Werten verwendet. In allen anderen Fällen wird die Zeitzone des Benutzers verwendet.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {ITEM.LASTVALUE.DATE<1-9>}, um auf den ersten, zweiten, dritten usw. Datenpunkt in einem Ausdrück eines Auslösers zu verweisen. Siehe indizierte Makros.

{ITEM.LASTVALUE.TIME}

Die Uhrzeit, zu der der letzte Datenpunktwert erfasst wurde.
 Wird im Format `HHMMSS` angezeigt.
 Die Server-Zeitzone wird in Auslösernamen (in der Liste *Monitoring* > *Probleme*), Ereignisnamen sowie in Tag-Namen und -Werten verwendet. In allen anderen Fällen wird die Zeitzone des Benutzers verwendet.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {ITEM.LASTVALUE.TIME<1-9>}, um auf den ersten, zweiten, dritten usw. Datenpunkt in einem Ausdrück für einen Auslöser zu verweisen. Siehe indizierte Makros.

{ITEM.LASTVALUE.TIMESTAMP}

Der UNIX-Zeitstempel, zu dem der letzte Datenpunktwert erfasst wurde.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {ITEM.LASTVALUE.TIMESTAMP<1-9>}, um auf den ersten, zweiten, dritten usw. Datenpunkt in einem Auslöser-Ausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.LOG.AGE}

Das Alter des Protokollereignisses.
Mit einer Genauigkeit bis auf eine Sekunde.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {ITEM.LOG.AGE<1-9>}, um auf den ersten, zweiten, dritten usw. Datenpunkt in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.LOG.DATE}

Das Datum, an dem der Logeintrag in das Log geschrieben wurde.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {ITEM.LOG.DATE<1-9>}, um auf den ersten, zweiten, dritten usw. Datenpunkt in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.LOG.EVENTID}

Die ID des Ereignisses im Ereignisprotokoll.
 Nur für die Überwachung des Windows-Ereignisprotokolls.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {ITEM.LOG.EVENTID<1-9>}, um auf das erste, zweite, dritte usw. Element in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.LOG.NSEVERITY}

Die numerische Schwere des Ereignisses im Ereignisprotokoll.
 Nur für die Überwachung des Windows-Ereignisprotokolls.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {ITEM.LOG.NSEVERITY<1-9>}, um auf das erste, zweite, dritte usw. Element in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.LOG.SEVERITY}

Die verbale Schwere des Ereignisses im Ereignisprotokoll.
 Nur für die Überwachung des Windows-Ereignisprotokolls.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {ITEM.LOG.SEVERITY<1-9>}, um auf den ersten, zweiten, dritten usw. Datenpunkt in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.LOG.SOURCE}

Die Quelle des Ereignisses im Ereignisprotokoll.
 Nur für die Überwachung des Windows-Ereignisprotokolls.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {ITEM.LOG.SOURCE<1-9>}, um auf den ersten, zweiten, dritten usw. Datenpunkt in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.LOG.TIME}

Die Zeit, zu der der Logeintrag in das Log geschrieben wurde.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {ITEM.LOG.TIME<1-9>}, um auf den ersten, zweiten, dritten usw. Datenpunkt in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.LOG.TIMESTAMP}

Der UNIX-Zeitstempel, zu dem der Logeintrag in das Log geschrieben wurde.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {ITEM.LOG.TIMESTAMP<1-9>}, um auf das erste, zweite, dritte usw. Element in einem Auslöser-ausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.NAME}

Der Name des Datenpunkts, bei dem alle Makros aufgelöst wurden.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {ITEM.NAME<1-9>}, um auf den ersten, zweiten, dritten usw. Datenpunkt in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.NAME.ORIG}

Der ursprüngliche Name des Datenpunkts (ohne aufgelöste Makros).

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {ITEM.NAME.ORIG<1-9>}, um auf den ersten, zweiten, dritten usw. Datenpunkt in einem Auslöser-Ausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.STATE}

Der aktuelle Zustand des N-ten Datenpunkts in der Auslöser-Ausdruck, der eine Benachrichtigung ausgelöst hat.
 Mögliche Werte: *Not supported*, *Normal*.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {ITEM.STATE<1-9>}, um auf den ersten, zweiten, dritten usw. Datenpunkt in einem Auslöser-Ausdruck zu verweisen. Siehe indexed macros.

{ITEM.STATE.ERROR}

Die Fehlermeldung mit Details dazu, warum ein Datenpunkt nicht mehr unterstützt wurde.
 Wenn ein Datenpunkt in den Status „nicht unterstützt“ wechselt und dann sofort wieder unterstützt wird, kann das Fehlerfeld leer sein.

{ITEM.VALUE}

Wird entweder aufgelöst zu:

- Dem historischen Wert (zum Zeitpunkt des Ereignisses) des N-ten Datenpunkts in der Auslöserausdruck, wenn er im Kontext einer Änderung des Auslöserstatus verwendet wird, zum Beispiel beim Anzeigen von Ereignissen oder beim Senden von Benachrichtigungen;
- Dem neuesten Wert des N-ten Datenpunkts im Auslöserausdruck, wenn er ohne den Kontext einer Änderung des Auslöser-status verwendet wird, zum Beispiel beim Anzeigen einer Liste von Auslösern in einem Popup-Auswahlfenster. In diesem Fall funktioniert er genauso wie {ITEM.LASTVALUE}.

Im ersten Fall wird er zu *UNKNOWN* aufgelöst, wenn der Historienwert bereits gelöscht wurde oder nie gespeichert wurde.
Im zweiten Fall und nur im Frontend wird er zu *UNKNOWN* aufgelöst, wenn der neueste Historienwert vor mehr als dem Zeitraum *Max history display period* erfasst wurde (eingestellt im Menübereich *Administration > General*).
Der aufgelöste Wert für Text-/Log-Datenpunkte wird vom Frontend an den folgenden Stellen auf 20 Zeichen gekürzt:
- Operative Daten;
- Auslöserbeschreibung;
- Auslöser-URLs;
- Auslöser-URL-Labels;
- Beschreibung des Datenpunktwert-Widgets.
Um den vollständigen Wert aufzulösen, können Sie Makrofunktionen verwenden, da vom Server keine Werte gekürzt werden. Zum Beispiel: `{{ITEM.VALUE}.regsub("(.*)", \1)}`.

Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {ITEM.VALUE<1-9>}, um auf den ersten, zweiten, dritten usw. Datenpunkt in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.VALUE.AGE}

Die Zeit, die zwischen der Erfassung des Datenpunktwerts und der Makroauswertung vergangen ist.
 Wird in einem menschen-lesbaren Format angezeigt (z. B. 1m 45s).

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {ITEM.VALUE.AGE<1-9>}, um auf den ersten, zweiten, dritten usw. Datenpunkt in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.VALUE.DATE}

Das Datum, an dem der Wert des Datenpunkts erfasst wurde.
 Wird im Format `YYYYMMDD` angezeigt.
 Die Server-Zeitzone wird in Auslösernamen (in der Liste *Monitoring* > *Probleme*), Ereignisnamen sowie Tag-Namen und -Werten verwendet. In allen anderen Fällen wird die Zeitzone des Benutzers verwendet.

Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {ITEM.VALUE.DATE<1-9>}, um auf den ersten, zweiten, dritten usw. Datenpunkt in einem Ausdrück eines Auslösers zu verweisen. Siehe indizierte Makros.

{ITEM.VALUE.TIME}

Die Uhrzeit, zu der der Wert des Datenpunkts erfasst wurde.
 Wird im Format `HHMMSS` angezeigt.
 Die Server-Zeitzone wird in Auslösernamen (in der Liste *Monitoring* > *Probleme*), Ereignisnamen sowie in Tag-Namen und -Werten verwendet. In allen anderen Fällen wird die Zeitzone des Benutzers verwendet.

Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {ITEM.VALUE.TIME<1-9>}, um auf den ersten, zweiten, dritten usw. Datenpunkt in einem Ausdrück eines Auslösers zu verweisen. Siehe indizierte Makros.

{ITEM.VALUE.TIMESTAMP}

Der UNIX-Zeitstempel, zu dem der Datenpunktwert erfasst wurde.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {ITEM.VALUE.TIMESTAMP<1-9>}, um auf den ersten, zweiten, dritten usw. Datenpunkt in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{ITEM.VALUETYPE}

Der Werttyp des N-ten Datenpunkts in der Auslöser-Ausdrucks, der eine Benachrichtigung ausgelöst hat.
 Mögliche Werte: *0* - Numerisch Gleitkomma, *1* - Zeichen, *2* - Log, *3* - Numerisch ohne Vorzeichen, *4* - Text.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {ITEM.VALUETYPE<1-9>}, um auf den ersten, zweiten, dritten usw. Datenpunkt in einem Auslöser-Ausdruck zu verweisen. Siehe indexierte Makros.

Benachrichtigungsmakros für Low-Level-Discovery

{LLDRULE.DESCRIPTION}

Die Beschreibung der Low-Level-Discovery-Regel, die eine Benachrichtigung ausgelöst hat.

{LLDRULE.DESCRIPTION.ORIG}

Die ursprüngliche Beschreibung (mit nicht aufgelösten Makros) der Low-Level-Discovery-Regel, die eine Benachrichtigung ausgelöst hat.

{LLDRULE.ID}

Die numerische ID der Low-Level-Discovery-Regel, die eine Benachrichtigung ausgelöst hat.

{LLDRULE.KEY}

Der Schlüssel der Low-Level-Discovery-Regel, die eine Benachrichtigung ausgelöst hat.

{LLDRULE.KEY.ORIG}

Der ursprüngliche Schlüssel (mit nicht aufgelösten Makros) der Low-Level-Discovery-Regel, die eine Benachrichtigung ausgelöst hat.

{LLDRULE.NAME}

Der Name der Low-Level-Discovery-Regel, die eine Benachrichtigung ausgelöst hat.

{LLDRULE.NAME.ORIG}

Der ursprüngliche Name (mit nicht aufgelösten Makros) der Low-Level-Discovery-Regel, die eine Benachrichtigung ausgelöst hat.

{LLDRULE.STATE}

Der aktuelle Zustand der Low-Level-Discovery-Vorlage.
 Mögliche Werte: *Nicht unterstützt*, *Normal*.

 Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {LLDRULE.STATE<1-9>}, um auf den ersten, zweiten, dritten usw. Datenpunkt in einem Auslöserausdruck zu verweisen. Siehe indizierte Makros.

{LLDRULE.STATE.ERROR}

Die Fehlermeldung mit Details dazu, warum die Low-Level-Discovery-Regel in den nicht unterstützten Zustand gewechselt ist.
 Wenn eine Low-Level-Discovery-Regel in den nicht unterstützten Zustand wechselt und dann sofort wieder unterstützt wird, kann das Fehlerfeld leer sein.

Makros in Karten

{MAP.ID}

Die Netzwerkplan-ID.

{MAP.NAME}

Der Name der Netzwerkkarte.

Proxy-Makros

{PROXY.DESCRIPTION}

Die Proxy-Beschreibung.
 Löst sich entweder auf in:

- Proxy des N-ten Datenpunkts im Ausdrück des Auslösers (bei benachrichtigungen auf Basis von Auslösern). Sie können hier indizierte Makros verwenden;
- Proxy, der die Erkennung ausgeführt hat (bei Erkennungsbenachrichtigungen). Verwenden Sie hier {PROXY.DESCRIPTION} ohne Indexierung;

- Proxy, bei dem sich ein aktiver Agent registriert hat (bei Autoregistrierungsbenachrichtigungen). Verwenden Sie hier {PROXY.DESCRIPTION} ohne Indexierung.

Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {PROXY.DESCRIPTION<1-9>}, um auf den ersten, zweiten, dritten usw. Eintrag in einem Ausdrück des Auslösers zu verweisen. Siehe indizierte Makros.

{PROXY.NAME}

Der Name des Proxy.
 Löst sich entweder auf in:

- Proxy des N-ten Datenpunkts im Auslöser-Ausdruck (bei auslöserbasierten Benachrichtigungen). Sie können hier indizierte Makros verwenden;
- Proxy, der die Erkennung ausgeführt hat (bei Erkennungsbenachrichtigungen). Verwenden Sie hier {PROXY.NAME} ohne Indizierung;
- Proxy, bei dem sich ein aktiver Agent registriert hat (bei Autoregistrierungsbenachrichtigungen). Verwenden Sie hier {PROXY.NAME} ohne Indizierung.

Dieses Makro kann mit einem numerischen Index verwendet werden, z. B. {PROXY.NAME<1-9>}, um auf den ersten, zweiten, dritten usw. Datenpunkt in einem Auslöser-Ausdruck zu verweisen. Siehe indizierte Makros.

Skript-Makros

{MANUALINPUT}

Der vom Benutzer zum Zeitpunkt der Skriptausführung angegebene manuelle Eingabewert.

Service-Makros

{SERVICE.DESCRIPTION}

Die Servicebeschreibung mit aufgelösten Makros.

{SERVICE.ID}

Die numerische ID des Service, der eine Aktion ausgelöst hat.

{SERVICE.NAME}

Der Dienstname mit aufgelösten Makros.

{SERVICE.ROOTCAUSE}

Die Liste der Problemereignisse von Auslösern, die zum Ausfall eines Service geführt haben, sortiert nach Schweregrad und Host-Name.
 Enthält die folgenden Details: Host-Name, Ereignisname, Schweregrad, Alter, Service-Tags und -Werte.

{SERVICE.TAGS}

Eine durch Kommas getrennte Liste von Service-Ereignis-Tags.
 Service-Ereignis-Tags können im Abschnitt für Konfigurations-Tags des Services definiert werden. Wird zu einer leeren Zeichenfolge erweitert, wenn keine Tags vorhanden sind.

{SERVICE.TAGSJSON}

Ein JSON-Array, das Objekte für Service-Ereignis-Tags enthält.
 Service-Ereignis-Tags können im Abschnitt für Konfigurations-Tags des Service definiert werden. Wird zu einem leeren Array erweitert, wenn keine Tags vorhanden sind.

{SERVICE.TAGS.<tag name>}

Der Wert des Service-Ereignis-Tags, auf den über den Tag-Namen verwiesen wird.
 Die Service-Ereignis-Tags können im Abschnitt für Konfigurations-Tags des Service definiert werden.
 Ein Tag-Name, der nicht alphanumerische Zeichen enthält (einschließlich nicht-englischer Multibyte-UTF-Zeichen), muss in doppelte Anführungszeichen gesetzt werden. Anführungszeichen und Backslashes innerhalb eines in Anführungszeichen gesetzten Tag-Namens müssen mit einem Backslash maskiert werden.

Auslöser-Makros

{TRIGGER.DESCRIPTION}

Die Auslöser-Beschreibung.
 Alle Makros, die in einer Auslöser-Beschreibung unterstützt werden, werden erweitert, wenn {TRIGGER.DESCRIPTION} im Benachrichtigungstext verwendet wird.

{TRIGGER.EVENTS.ACK}

Die Anzahl der bestätigten Ereignisse für ein Kartenelement in Karten oder für den Auslöser, der das aktuelle Ereignis in Benachrichtigungen erzeugt hat.

{TRIGGER.EVENTS.PROBLEM.ACK}

Die Anzahl der bestätigten Problemereignisse für alle Auslöser, unabhängig von ihrem Status.

{TRIGGER.EVENTS.PROBLEM.UNACK}

Die Anzahl der nicht bestätigten Problemereignisse für alle Auslöser, unabhängig von ihrem Status.

{TRIGGER.EVENTS.UNACK}

Die Anzahl der nicht bestätigten Ereignisse für ein Kartenelement in Karten oder für den Auslöser, der das aktuelle Ereignis in Benachrichtigungen erzeugt hat.

{TRIGGER.EXPRESSION}

Der Auslöser-Ausdruck.

{TRIGGER.EXPRESSION.EXPLAIN}

Ein teilweise ausgewerteter Auslöser-Ausdruck.
 Auf Datenpunkt basierende Funktionen werden zum Zeitpunkt der Ereigniserzeugung ausgewertet und durch die Ergebnisse ersetzt, während alle anderen Funktionen so angezeigt werden, wie sie im Ausdruck geschrieben sind. Kann zum Debuggen von Auslöser-Ausdrücken verwendet werden.

{TRIGGER.EXPRESSION.RECOVERY}

Der Auslöser-Wiederherstellungsausdruck, wenn die *OK-Ereigniserzeugung* in der Auslöser-Konfiguration auf „Wiederherstellungsausdruck“ gesetzt ist; andernfalls wird eine leere Zeichenkette zurückgegeben.

{TRIGGER.EXPRESSION.RECOVERY.EXPLAIN}

Ein teilweise ausgewerteter Auslöser-Wiederherstellungsausdruck.
 Auf Datenpunkt basierende Funktionen werden zum Zeitpunkt der Ereigniserzeugung ausgewertet und durch die Ergebnisse ersetzt, während alle anderen Funktionen so angezeigt werden, wie sie im Ausdruck geschrieben sind. Kann zum Debuggen von Auslöser-Wiederherstellungsausdrücken verwendet werden.

{TRIGGER.HOSTGROUP.NAME}

Eine sortierte (nach SQL-Abfrage), durch Komma und Leerzeichen getrennte Liste von Host-Gruppen, in denen der Auslöser definiert ist.

{TRIGGER.ID}

Die numerische Auslöser-ID, die diese Aktion ausgelöst hat.

{TRIGGER.NAME}

Der Auslösername mit aufgelösten Makros.
 Beachten Sie, dass `{EVENT.NAME}` in Aktionen verwendet werden kann, um den Namen des ausgelösten Ereignisses/Problems mit aufgelösten Makros anzuzeigen.

{TRIGGER.NAME.ORIG}

Der ursprüngliche Auslösername (mit nicht aufgelösten Makros).

{TRIGGER.NSEVERITY}

Der numerische Auslöser-Schweregrad.
 Mögliche Werte: *0* - Nicht klassifiziert, *1* - Information, *2* - Warnung, *3* - Durchschnitt, *4* - Hoch, *5* - Katastrophe.

{TRIGGER.PROBLEM.EVENTS.PROBLEM.ACK}

Die Anzahl der bestätigten Problemereignisse für Auslöser im Problemzustand.

{TRIGGER.PROBLEM.EVENTS.PROBLEM.UNACK}

Die Anzahl der nicht bestätigten Problemereignisse für Auslöser im Problemzustand.

{TRIGGER.SEVERITY}

Der Name des Auslöser-Schweregrads.
 Kann unter *Administration* > *General* > *Trigger displaying options* definiert werden.

{TRIGGER.STATE}

Der letzte Status des Auslöser-Ausdrucks.
 Mögliche Werte: *Unbekannt*, *Normal*.

{TRIGGER.STATE.ERROR}

Die Fehlermeldung mit Details dazu, warum ein Auslöser nicht mehr unterstützt wurde.
 Wenn ein Auslöser in den Status „nicht unterstützt“ wechselt und dann sofort wieder unterstützt wird, kann das Fehlerfeld leer sein.

{TRIGGER.STATUS}

Der Auslöser-Wert zum Zeitpunkt der Ausführung des Operationsschritts.
 Mögliche Werte: *Problem*, *OK*.

{TRIGGER.TEMPLATE.NAME}

Eine sortierte (nach SQL-Abfrage), durch Komma und Leerzeichen getrennte Liste von Vorlagen, in denen der Auslöser definiert ist, oder *UNKNOWN*, wenn der Auslöser in einem Host definiert ist.

{TRIGGER.URL}

Die Auslöser-URL.

{TRIGGER.URL.NAME}

Die Bezeichnung für die Auslöser-URL.

{TRIGGER.VALUE}

Der aktuelle numerische Auslöser-Wert.
 Mögliche Werte: *0* - Auslöser befindet sich im Status OK, *1* - Auslöser befindet sich im Status Problem.

{TRIGGERS.ACK}

Die Anzahl der bestätigten Auslöser für ein Kartenelement, unabhängig vom Auslöserstatus.
 Ein Auslöser gilt als bestätigt, wenn alle zugehörigen Problemereignisse bestätigt sind.

{TRIGGERS.PROBLEM.ACK}

Die Anzahl der bestätigten Problem-Auslöser für ein Kartenelement.
 Ein Auslöser gilt als bestätigt, wenn alle seine Problemereignisse bestätigt sind.

{TRIGGERS.PROBLEM.UNACK}

Die Anzahl der nicht bestätigten Problem-Auslöser für ein Kartenelement.
 Ein Auslöser gilt als nicht bestätigt, wenn mindestens eines seiner Problemereignisse nicht bestätigt ist.

{TRIGGERS.UNACK}

Die Anzahl der nicht bestätigten Auslöser für ein Kartenelement, unabhängig vom Auslöserstatus.
 Ein Auslöser gilt als nicht bestätigt, wenn mindestens eines seiner Problemereignisse nicht bestätigt ist.

Benutzernamen-Makros

{USER.FULLNAME}

Der Vorname, Nachname und Benutzername des Benutzers, der die Ereignisbestätigung hinzugefügt oder das Skript gestartet hat.

{USER.NAME}

Der Name des Benutzers, der das Skript gestartet hat.

{USER.SURNAME}

Der Nachname des Benutzers, der das Skript gestartet hat.

{USER.USERNAME}

Der Benutzername des Benutzers, der das Skript gestartet hat.

Indizierte Makros

Die Syntax für indizierte Makros **{MACRO<1-9>}** kann nur auf das N-te Element, die N-te Funktion oder den N-ten Host im Feld *Ausdruck* eines Auslösers verweisen:

- **{HOST.IP1}, {HOST.IP2}, {HOST.IP3}** werden zu den IP-Adressen des ersten, zweiten und dritten Hosts im Auslöserausdruck aufgelöst (falls vorhanden).
- **{ITEM.VALUE1}, {ITEM.VALUE2}, {ITEM.VALUE3}** werden zum Zeitpunkt des Ereignisses zu den Werten des ersten, zweiten und dritten Datenpunkts im Auslöserausdruck aufgelöst (falls vorhanden).
- **{FUNCTION.VALUE1}, {FUNCTION.VALUE2}, {FUNCTION.VALUE3}** werden zum Zeitpunkt des Ereignisses zu den Werten der ersten, zweiten und dritten datenpunktbasierten Funktion aufgelöst (falls vorhanden).

Im Kontext von Auslösern beziehen sich indizierte Makros immer auf das Feld *Ausdruck* der Auslöserkonfiguration, nicht auf den *Wiederherstellungsausdruck*. Beispielsweise wird bei einem Wiederherstellungsereignis **{ITEM.VALUE2}** zum Zeitpunkt der Wiederherstellung auf den Wert des zweiten Datenpunkts aus dem Problemausdruck aufgelöst.

Das Makro **{HOST.HOST<1-9>}** wird auch innerhalb des Ausdrucksmakros `{?func(/host/key,param)}` in **Diagrammnamen** unterstützt. Beispielsweise wird `{?func(/{HOST.HOST2}/key,param)}` in einem Diagrammnamen auf den Host des zweiten Datenpunkts im Diagramm aufgelöst.

Warning:
Indizierte Makros werden in keinem anderen Kontext als den hier genannten aufgelöst. Verwenden Sie für andere Kontexte Makros ohne Index (**{HOST.HOST}, {HOST.IP}** usw.).

Von Standort unterstützte Benutzermakros

Übersicht

Dieser Abschnitt enthält eine Liste der Stellen, an denen benutzerdefinierte Makros unterstützt werden.

Note:
Für *Aktionen*, *Netzwerkerkennung*, *Proxys* und alle unter dem Abschnitt *Andere Stellen* auf dieser Seite aufgeführten Stellen werden nur Benutzermakros auf globaler Ebene unterstützt. An den genannten Stellen werden Makros auf Host-Ebene und Vorlagen-Ebene nicht aufgelöst.

Note:
Um Makrowerte anzupassen (zum Beispiel bestimmte Teilzeichenfolgen zu kürzen oder zu extrahieren), können Sie Makrofunktionen verwenden.

Aktionen

In Aktionen können Benutzermakros in den folgenden Feldern verwendet werden:

Position		Mehrere Makros/Mischung mit Text[1]
Auslöser-basierte Benachrichtigungen und Befehle		ja
Auslöser-basierte interne Benachrichtigungen		ja
Benachrichtigungen zu Problemaktualisierungen		ja
Service-basierte Benachrichtigungen und Befehle		ja
Benachrichtigungen zu Serviceaktualisierungen		ja
Bedingung für Zeitperiode		nein
Operationen		
	Standarddauer des Operationsschritts	nein
	Schrittdauer	nein

Hosts/Host-Prototypen

In der Konfiguration eines Host und eines Host-Prototyps können Benutzermakros in den folgenden Feldern verwendet werden:

Ort		Mehrere Makros/Mischung mit Text[1]
Schnittstellen-IP/DNS		nur DNS
Schnittstellen-Port		nein
SNMP v1, v2		
	SNMP-Community	ja
SNMP v3		
	Kontextname	ja
	Sicherheitsname	ja
	Authentifizierungs-Passphrase	ja
	Privacy-Passphrase	ja
IPMI		
	Benutzername	ja
	Passwort	ja
Tags[2]		
	Tag-Namen	ja
	Tag-Werte	ja

Datenpunkte / Datenpunkt-Prototypen

In der Konfiguration eines Datenpunkts oder eines Datenpunkt-Prototyps können Benutzermakros in den folgenden Feldern verwendet werden:

Speicherort	Mehrere Makros/Textkombinationen[1]
Datenpunktname	ja
Datenpunktschlüsselparameter	ja[4]
Aktualisierungsintervall	nein
Benutzerdefinierte Intervalle	nein
Zeitüberschreitung (verfügbar für unterstützte Datenpunkttypen)	nein
Speichern bis (für Verlauf und Trends)	nein
Beschreibung	ja
Berechneter/aggregierter Datenpunkt	
Formel (Ausdruckskonstanten und Funktionsparameter; Datenpunktschlüsselparameter; (*nur aggregierter Datenpunkt*) Filterbedingungen (Name der Hostgruppe und Tag-Name))	ja
Datenbankmonitor	
Benutzername	ja
Passwort	ja
SQL-Abfrage	ja
HTTP-Agent	
URL[3]	ja
Abfragefelder	ja
Anfragetext	ja
Header (Namen und Werte)	ja
Erforderliche Statuscodes	ja
HTTP-Proxy	ja
Benutzername der HTTP-Authentifizierung	ja
Passwort der HTTP-Authentifizierung	ja
Datei des SSL-Zertifikats	ja
SSL-Schlüsseldatei	ja
Passwort des SSL-Schlüssels	ja
Zulässige Hosts	ja
JMX-Agent	
JMX-Endpunkt	ja
Skript-Datenpunkt	
Parametername und -werte	ja
Browser-Datenpunkt	
Parametername und -werte	ja

Speicherort	Mehrere Makros/Textkombinationen[1]
Telemetrieabfrage-Datenpunkt	
Zeitverschiebung	nein
Rückblicklimit	nein
Granularität	nein
SNMP-Agent	
SNMP-OID	ja
SSH-Agent	
Benutzername	ja
Datei mit öffentlichem Schlüssel	ja
Datei mit privatem Schlüssel	ja
Passwort	ja
Skript	ja
TELNET-Agent	
Benutzername	ja
Passwort	ja
Skript	ja
Zabbix-Trapper	
Zulässige Hosts	ja
Tags[2]	
Tag-Namen	ja
Tag-Werte	ja
Vorverarbeitungsschritte	
Parameter (einschließlich benutzerdefinierter Skripte)	ja
Benutzerdefinierte Parameter für die Fehlerbehandlung (Felder *Wert setzen auf* und *Fehler setzen auf*)	ja

Low-level-Discovery

In einer Low-level-Discovery- Regel können Benutzermakros in den folgenden Feldern verwendet werden:

Position	Mehrere Makros/Mischung mit Text[1]
Schlüsselparameter	ja
Aktualisierungsintervall	nein
Benutzerdefiniertes Intervall	nein
Timeout (verfügbar für unterstützte Datenpunkttypen)	nein
Verlorene Ressourcen löschen	nein

Position	Mehrere Makros/Mischung mit Text[1]
Verlorene Ressourcen deak-tivieren	nein
Beschreibung	ja
SNMP-Agent	
SNMP-OID	ja
SSH-Agent	
Benutzername	ja
Datei mit öffentlichem Schlüssel	ja
Datei mit privatem Schlüssel	ja
Passwort	ja
Skript	ja
TELNET-Agent	
Benutzername	ja
Passwort	ja
Skript	ja
Zabbix-Trapper	
Erlaubte Hosts	ja
Datenbankmonitor	
Benutzername	ja
Passwort	ja
SQL-Abfrage	ja
JMX-Agent	
JMX-Endpunkt	ja
HTTP-Agent	
URL[3]	ja
Abfragefelder	ja
Request-Body	ja
Header (Namen und Werte)	ja
Erforderliche Statuscodes	ja
Benutzername für HTTP-Authentifizierung	ja
Passwort für HTTP-Authentifizierung	ja
Filter	
Regulärer Ausdruck	ja
Überschreibungen	
Filter: regulärer Ausdruck	ja
Operationen: Aktualisierungsintervall (für Datenpunktprototypen)	nein
Operationen: Speicherzeitraum für Verlauf (für Datenpunktprototypen)	nein
Operationen: Speicherzeitraum für Trends (für Datenpunktprototypen)	nein

Netzwerkerkennung

In einer Netzwerkerkennungsregel können Benutzermakros in den folgenden Feldern verwendet werden:

Position		Mehrere Makros/Mischung mit Text[1]
Aktualisierungsintervall		nein
SNMP v1, v2		
	SNMP-Community	ja
	SNMP-OID	ja
SNMP v3		
	Kontextname	ja
	Sicherheitsname	ja
	Authentifizierungs-Passphrase	ja

Position		Mehrere Makros/Mischung mit Text[1]
	Privacy-Passphrase	ja
	SNMP-OID	ja

Proxies

In einer proxy Konfiguration können Benutzermakros in den folgenden Feldern verwendet werden:

Location	Multiple macros/mix with text[1]
Address for active agents > *Port* (if the proxy belongs to a group)	no
Interface address and port (for passive proxy)	no
Timeouts for item types	no

Proxy-Gruppen

In einer Proxy-Gruppe-Konfiguration können Benutzermakros in den folgenden Feldern verwendet werden:

Ort	Mehrere Makros/Mischung mit Text[1]
Failover-Zeitraum	nein
Minimale Anzahl von Proxys	nein

Vorlagen

In einer Vorlage-Konfiguration können Benutzer-Makros in den folgenden Feldern verwendet werden:

Ort		Mehrere Makros/Mischung mit Text[1]
Tags[2]		
	Tag-Namen	ja
	Tag-Werte	ja

Auslöser

In der Konfiguration eines Auslösers können Benutzermakros in den folgenden Feldern verwendet werden:

Position	Mehrere Makros/Mischung mit Text[1]
Name	ja
Betriebsdaten	ja
Ausdruck (nur in Konstanten und Funktionsparametern; geheime Makros werden nicht unterstützt)	ja

Position	Mehrere Makros/Mischung mit Text[1]
Tag für den Abgleich	ja
Name des Menüeintrags	ja
URL des Menüeintrags[3]	ja
Beschreibung	ja
Tags[2]	
Tag-Namen	ja
Tag-Werte	ja

Web-Szenario

In der Konfiguration eines Web-Szenarios können Benutzermakros in den folgenden Feldern verwendet werden:

Position		Mehrere Makros/Mischung mit Text[1]
Name		ja
Aktualisierungsintervall		nein
Agent		ja
HTTP-Proxy		ja
Variablen (nur Werte)		ja
Header (Namen und Werte)		ja
Schritte		
	Name	ja
	URL[3]	ja
	Variablen (nur Werte)	ja
	Header (Namen und Werte)	ja
	Timeout	nein
	Erforderliche Zeichenfolge	ja
	Erforderliche Statuscodes	nein
Authentifizierung		
	Benutzer	ja
	Passwort	ja
	SSL-Zertifikat	ja
	SSL-Schlüsseldatei	ja
	SSL-Schlüsselpasswort	ja
Tags[2]		
	Tag-Namen	ja
	Tag-Werte	ja

Andere Orte

Zusätzlich zu den hier aufgeführten Orten können Benutzermakros in den folgenden Feldern verwendet werden:

Ort	Mehrere Makros/Mischung mit Text[1]
Globale Skripte (URL, Skript, SSH, Telnet, IPMI), einschließlich Bestätigungstext	yes
Webhooks	
JavaScript-Skript	no
Name des JavaScript-Skriptparameters	no
Wert des JavaScript-Skriptparameters	yes
Dashboards	
Spalte des Datentyps *Text* im Dashboard-Widget *Top hosts*	yes
Parameter *Description* im Dashboard-Widget *Item value* und *Gauge*	yes
Parameter *Text* der primären/sekundären Beschriftung im Dashboard-Widget *Honeycomb*	yes
Parameter *URL*[3] im Dashboard-Widget *URL*	yes
Monitoring > *Problems*, *Hosts*, *Latest data*	
Spalte *Custom text*	yes
Data collection > *Templates*, *Hosts*	
Spalte *Custom text*	yes
Alerts > *Media types* > *Message templates*	
Betreff	yes
Nachricht	yes

Ort	Mehrere Makros/Mischung mit Text[1]
Alerts > *Media types* > *Script*	
Skriptparameter	yes
Alerts > *Media types* > *Media type*	
Felder *Username* und *Password* für den Medientyp *Email* (wenn *Authentication* auf "Username and password" gesetzt ist; Secret-Makros werden empfohlen)	yes
Users > *Users* > *Media*	
Wenn aktiv	yes
Administration > *General* > *GUI*	
Arbeitszeit	no
Administration > *General* > *Timeouts*	
Timeouts für Datentypen	no
Administration > *General* > *Connectors*	
URL	yes
Benutzername	yes
Passwort	yes
Bearer-Token	yes
Timeout	no
HTTP-Proxy	yes
SSL-Zertifikatsdatei	yes
SSL-Schlüsseldatei	yes
SSL-Schlüsselpasswort	yes

Eine vollständige Liste aller in Zabbix unterstützten Makros finden Sie unter unterstützte Makros.

Fußnoten

[1] Wenn mehrere Makros in einem Feld oder Makros zusammen mit Text für den jeweiligen Ort nicht unterstützt werden, muss ein einzelnes Makro das gesamte Feld ausfüllen.

[2] Makros, die in Tag-Namen und -Werten verwendet werden, werden nur während des Ereigniserstellungsprozesses aufgelöst.

[3] URLs, die ein geheimes Makro enthalten, funktionieren nicht, da das Makro darin als "******" aufgelöst wird.

[4] Makros werden im item key von Zabbix trapper items nicht unterstützt.

Versionskompatibilität

Unterstützte Agents

Um mit Zabbix 8.0 kompatibel zu sein, darf der Zabbix Agent nicht älter als Version 1.6 und nicht neuer als 8.0 sein.

Möglicherweise müssen Sie die Konfiguration älterer Agents überprüfen, da einige Parameter geändert wurden, beispielsweise Parameter im Zusammenhang mit der Protokollierung für Versionen vor 3.0.

Um die neuesten Funktionen, Metriken, eine verbesserte Leistung und einen geringeren Speicherverbrauch vollständig zu nutzen, verwenden Sie den neuesten unterstützten Agent.

Hinweise für Windows XP

- Verwenden Sie unter 32-Bit-Windows XP keine Zabbix Agents, die neuer als 6.0.x sind.
- Verwenden Sie unter Windows XP/Server 2003 keine Agent-Vorlagen, die neuer als Zabbix 4.0.x sind. Die neueren Vorlagen verwenden englische Performance Counter, die erst ab Windows Vista/Server 2008 unterstützt werden.

Unterstützte Agents 2

Ältere Zabbix Agents 2 ab Version 4.4 sind mit Zabbix 8.0 kompatibel; Zabbix Agent 2 darf nicht neuer als 8.0 sein.

Beachten Sie, dass bei Verwendung von Zabbix Agent 2 in den Versionen 4.4 und 5.0 das Standardintervall von 10 Minuten zum Aktualisieren nicht unterstützter Datenpunkte verwendet wird.

Um die neuesten Funktionen, Metriken, die verbesserte Leistung und den reduzierten Speicherverbrauch voll auszuschöpfen, verwenden Sie den neuesten unterstützten Agent 2.

Unterstützte Zabbix-Proxys

Für die vollständige Kompatibilität mit Zabbix 8.0 müssen Proxys mit der Hauptversion des Servers übereinstimmen. Nur Zabbix 8.0.x-Proxys sind vollständig mit einem Zabbix 8.0.x-Server kompatibel.

Veraltete Proxys werden teilweise unterstützt: Sie können weiterhin Daten erfassen und Skripte ausführen, können jedoch keine Konfigurationsaktualisierungen empfangen, wie z. B. neue Datenpunkte.

In Bezug auf den Zabbix Server können Proxys folgende Zustände haben:

- *Aktuell* (Proxy und Server haben dieselbe Hauptversion);
- *Veraltet* (die Proxy-Version ist älter als die Server-Version, wird aber teilweise unterstützt);
- *Nicht unterstützt* (die Proxy-Version ist älter als die Version des vorherigen LTS-Release des Servers *oder* die Proxy-Version ist neuer als die Hauptversion des Servers).

Beispiele:

Server-Version	*Aktuelle* Proxy-Version	*Veraltete* Proxy-Version	*Nicht unterstützte* Proxy-Version
6.4	6.4	6.0, 6.2	Älter als 6.0; neuer als 6.4
7.0	7.0	6.0, 6.2, 6.4	Älter als 6.0; neuer als 7.0
7.2	7.2	7.0	Älter als 7.0; neuer als 7.2
7.4	7.4	7.0	Älter als 7.0; neuer als 7.4
8.0	8.0	7.0, 7.2, 7.4	Älter als 7.0; neuer als 8.0

Von Proxys unterstützte Funktionalität:

Proxy version	Datenaktualisierung	Konfigurationsaktualisierung	Aufgaben
Current	Ja	Ja	Ja

Proxy version	Datenaktualisierung	Konfigurationsaktualisierung	Aufgaben
Outdated	Ja	Nein	Remote-Befehle (z. B. Shell-Skripte); Unmittelbare Prüfungen von Datenpunktwerten (d. h. *Jetzt ausführen*); Hinweis: Vorverarbeitung Tests mit einem echten Wert werden nicht unterstützt.
Unsupported	Nein	Nein	Nein

Warnungen zur Verwendung inkompatibler Zabbix-Daemon-Versionen werden protokolliert.

Unterstützte XML-Dateien

XML-Dateien, die nicht älter als Version 1.8 sind, werden für den Import in Zabbix 8.0 unterstützt.

Attention:
Im XML-Exportformat werden Auslöser-Abhängigkeiten nur nach Namen gespeichert. Wenn es mehrere Auslöser mit demselben Namen gibt (zum Beispiel mit unterschiedlichen Schweregraden und Ausdrücken), zwischen denen eine Abhängigkeit definiert ist, können diese nicht importiert werden. Solche Abhängigkeiten müssen manuell aus der XML-Datei entfernt und nach dem Import erneut hinzugefügt werden.

Syntax des Zeitraums

Übersicht

Um einen Zeitraum festzulegen, muss das folgende Format verwendet werden:

`d-d,hh:mm-hh:mm`

Dabei stehen die Symbole für Folgendes:

Symbol	Beschreibung
d	Wochentag: 1 - Montag, 2 - Dienstag, ... , 7 - Sonntag
hh	Stunden: 00-24
mm	Minuten: 00-59

Sie können mit einem Semikolon (;) als Trennzeichen mehr als einen Zeitraum angeben:

`d-d,hh:mm-hh:mm;d-d,hh:mm-hh:mm...`

Wenn der Zeitraum leer gelassen wird, entspricht dies 1-7,00:00-24:00, was der Standardwert ist.

Attention:
Die Obergrenze eines Zeitraums ist nicht eingeschlossen. Wenn Sie also 09:00-18:00 angeben, ist die letzte im Zeitraum enthaltene Sekunde 17:59:59.

Beispiele

Arbeitszeiten. Montag - Freitag von 9:00 bis 18:00:

`1-5,09:00-18:00`

Arbeitszeiten plus Wochenende. Montag - Freitag von 9:00 bis 18:00 Uhr und Samstag, Sonntag von 10:00 bis 16:00 Uhr:

`1-5,09:00-18:00;6-7,10:00-16:00`

Einheitensymbole

Übersicht

Die Arbeit mit großen Werten wie 86400, 104857600 oder 1000000 kann schwierig sein und zu Fehlern führen. Um die Konfiguration zu vereinfachen und die Lesbarkeit zu verbessern, unterstützt Zabbix Einheitensymbole (Suffixe), die als Wertmultiplikatoren fungieren.

Auslöser-Ausdrücke ohne Suffixe:

```
last(/host/system.uptime)<86400
avg(/host/system.cpu.load,600s)<10
last(/host/vm.memory.size[available])<20971520
```

Auslöser-Ausdrücke mit Suffixen:

```
last(/host/system.uptime)<1d
avg(/host/system.cpu.load,10m)<10
last(/host/vm.memory.size[available])<20M
```

Suffixe können auch die Konfiguration anderer Entitäten wie Datenpunkten, Widgets usw. vereinfachen und dabei helfen, Datenpunktwerte in einem menschenlesbaren Format anzuzeigen.

> **Note:**
> Ob ein Konfigurationsfeld einer Entität Suffixe unterstützt, entnehmen Sie immer der entsprechenden Seite für die zu konfigurierende Entität.

Zeitsuffixe

Das Zabbix-Frontend unterstützt in der Entitätskonfiguration die folgenden Zeitsuffixe:
 s - Sekunden *(bei Verwendung wirkt es genauso wie der Rohwert)*;
 m - Minuten;
 h - Stunden;
 d - Tage;
 w - Wochen;
 M - Monate *(nur Trendfunktionen)*;
 y - Jahre *(nur Trendfunktionen)*.

Zeitsuffixe werden nur mit ganzen Zahlen unterstützt. Beispiel: 1h wird unterstützt, 1,5h oder 1.5h jedoch nicht; verwenden Sie stattdessen 90m.

Speichergrößen-Suffixe

Zabbix unterstützt die folgenden Speichergrößen-Suffixe:
 K - Kilobytes
 M - Megabytes
 G - Gigabytes
 T - Terabytes

Item-Wert-Suffixe

Suffixe können auch verwendet werden, um numerische Item-Werte in einem für Menschen lesbaren Format anzuzeigen.

Um dies zu aktivieren, verwenden Sie eines der folgenden Suffixe im Feld *Units*, wenn Sie ein Item konfigurieren:
 B - Bytes;
 Bps - Bytes pro Sekunde;
 s - Sekunden, angezeigt mit bis zu drei größten ungleich null Zeit-Einheiten;
 uptime - verstrichene Zeit in hh:mm:ss oder N Tage, hh:mm:ss;
 unixtime - Unix-Zeitstempel, formatiert als yyyy.mm.dd hh:mm:ss.

Für die Interpretation und Anzeige dieser Suffixe gelten die folgenden zusätzlichen Regeln:

- Für B und Bps verwendet Zabbix die Basis-2-Umrechnung (1K = 1024B) gemäß dem JEDEC-Standard.
- Für andere Einheiten (z. B. Hz, W usw.) verwendet Zabbix die Basis-10-Umrechnung (1K = 1000).
- Für s (Sekunden):
  - Das Format umfasst yyy mmm ddd hhh mmm sss ms; es werden nur bis zu drei größte ungleich null Zeit-Einheiten angezeigt (z. B. 1M 10d 4h).
  - Wenn eine Einheit null ist und zwischen zwei ungleich null Einheiten liegt, wird sie ausgelassen (z. B. 10d 56m statt 10d 0h 56m).

Wenn *Units* verwendet werden, werden die folgenden Multiplikator-Suffixe automatisch auf Item-Werte angewendet:
 K, **M**, **G**, **T** - Kilo, Mega, Giga, Tera;
 P, **E**, **Z**, **Y** - Peta, Exa, Zetta, Yotta *(diese werden nur im Frontend angewendet).*

Um die Einheitenumrechnung zu verhindern, verwenden Sie das Präfix ! (z. B. !B oder !s).

Die folgenden Beispiele zeigen, wie empfangene Item-Werte basierend auf den angegebenen Einheiten umgerechnet werden:

```
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:
Vor Zabbix 4.0 gab es eine fest kodierte Stoppliste für Einheiten, bestehend aus `ms`, `rpm`, `RPM`, `%`. Diese Stoppliste ist veraltet; daher ist die korrekte Methode, die Umrechnung solcher Einheiten zu verhindern, `!ms`, `!rpm`, `!RPM`, `!%`.

Befehlsausführung

Zabbix verwendet eine gemeinsame Funktionalität für externe Prüfungen, Benutzerparameter, system.run-Datenpunkte, benutzerdefinierte Alarmskripte, Remote-Befehle und globale Skripte.

Ausführungsschritte

Note:
Standardmäßig werden alle Skripte in Zabbix mit der *sh*-Shell ausgeführt, und es ist nicht möglich, die Standard-Shell zu ändern. Um eine andere Shell zu verwenden, können Sie einen Workaround nutzen: Erstellen Sie eine Skriptdatei und rufen Sie dieses Skript bei der Befehlsausführung auf.

Der Befehl/das Skript wird auf Unix- und Windows- Plattformen auf ähnliche Weise ausgeführt:

1. Zabbix (der übergeordnete Prozess) erstellt eine Pipe für die Kommunikation
2. Zabbix setzt die Pipe als Ausgabe für den zu erstellenden Child- Prozess
3. Zabbix erstellt den Child-Prozess (führt den Befehl/das Skript aus)
4. Für den Child-Prozess wird eine neue Prozessgruppe (unter Unix) bzw. ein Job (unter Windows) erstellt
5. Zabbix liest aus der Pipe, bis ein Timeout eintritt oder niemand mehr auf das andere Ende schreibt (ALLE Handles/Dateideskriptoren wurden geschlossen). Beachten Sie, dass der Child-Prozess weitere Prozesse erstellen und beendet werden kann, bevor diese beendet werden oder den Handle/Dateideskriptor schließen.
6. Wenn das Timeout noch nicht erreicht wurde, wartet Zabbix, bis der ursprüngliche Child-Prozess beendet wird oder ein Timeout eintritt
7. Wenn der ursprüngliche Child-Prozess beendet wurde und das Timeout noch nicht erreicht wurde, prüft Zabbix den Exit-Code des ursprünglichen Child-Prozesses und vergleicht ihn mit 0 (ein Wert ungleich null wird als Ausführungsfehler betrachtet, nur für benutzerdefinierte Alarmierungsskripte, Remote-Befehle und Benutzerskripte, die auf Zabbix Server und Zabbix Proxy ausgeführt werden)
8. An diesem Punkt wird angenommen, dass alles abgeschlossen ist und der gesamte Prozessbaum (d. h. die Prozessgruppe oder der Job) beendet wird

Attention:
Zabbix geht davon aus, dass ein Befehl/Skript die Verarbeitung abgeschlossen hat, wenn der ursprüngliche Child-Prozess beendet wurde UND kein anderer Prozess den Ausgabe-Handle/Dateideskriptor noch offen hält. Wenn die Verarbeitung abgeschlossen ist, werden ALLE erstellten Prozesse beendet.

Alle doppelten Anführungszeichen und Backslashes im Befehl werden mit Backslashes maskiert, und der Befehl wird in doppelte Anführungszeichen eingeschlossen.

Exit-Code-Prüfung

Exit-Codes werden anhand der folgenden Bedingungen geprüft:

- Nur für benutzerdefinierte Alarm-Skripte, Remote-Befehle und Benutzerskripte, die auf dem Zabbix Server und dem Zabbix Proxy ausgeführt werden.
- Jeder Exit-Code, der ungleich 0 ist, wird als Ausführungsfehler betrachtet.
- Inhalte von Standardfehler und Standardausgabe fehlgeschlagener Ausführungen werden erfasst und im Frontend verfügbar gemacht (wo das Ausführungsergebnis angezeigt wird).
- Ein zusätzlicher Logeintrag kann für Remote-Befehle erstellt werden, die auf Zabbix Agent/Proxy ausgeführt werden, indem der Parameter LogRemoteCommands auf agent/proxy aktiviert wird.

Mögliche Frontend-Meldungen und Logeinträge für fehlgeschlagene Befehle/Skripte:

- Inhalte von Standardfehler und Standardausgabe fehlgeschlagener Ausführungen (falls vorhanden).
- "Process exited with code: N." (bei leerer Ausgabe und Exit-Code ungleich 0).
- "Process killed by signal: N." (wenn der Prozess durch ein Signal beendet wurde, nur unter Linux).
- "Process terminated unexpectedly." (wenn der Prozess aus unbekannten Gründen beendet wurde).

Siehe auch

- Externe Prüfungen
- Benutzerparameter
- system.run-Datenpunkte
- Benutzerdefinierte Alarmierungsskripte
- Remote-Befehle
- Globale Skripte

Beispiele für Escaping

Übersicht

Diese Seite enthält Beispiele für die korrekte Escaping-Verwendung bei regulären Ausdrücken in verschiedenen Kontexten.

> **Note:**
> Bei Verwendung des Konstruktors für Auslöser-Ausdrücke wird das korrekte Escaping in regulären Ausdrücken automatisch hinzugefügt.

Beispiele

Benutzermakro mit Kontext

Regulärer Ausdruck: `\.+\"[a-z]+<br>` Benutzermakro mit Kontext: `{$MACRO:regex:"\.+\\"[a-z]+"}`

Beachten Sie:

- Backslashes werden nicht maskiert;
- Anführungszeichen werden maskiert.

Makrofunktion innerhalb eines Datenpunktschlüssel-Parameters

Regulärer Ausdruck: `.+:(\d+)$<br>` Datenpunktschlüssel: `net.tcp.service[tcp,,"{{$ENDPOINT}.regsub(\".+:(\d+)$\",\1)}`

Beachten Sie:

- der reguläre Ausdruck innerhalb der Makrofunktion `regsub` steht in doppelten Anführungszeichen (weil er eine schließende Klammer enthält);
- die Anführungszeichen um den regulären Ausdruck werden maskiert (weil der gesamte dritte Datenpunkt-Parameter in doppelten Anführungszeichen steht);
- der dritte Datenpunktschlüssel-Parameter steht in doppelten Anführungszeichen, weil er ein Komma enthält.

LLD-Makrofunktion

Regulärer Ausdruck: `\.+\"([a-z]+)<br>` LLD-Makro: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}`

Beachten Sie:

- Backslashes werden nicht maskiert;
- Anführungszeichen werden maskiert.

LLD-Makrofunktion innerhalb eines Benutzermakro-Kontexts

Regulärer Ausdruck: `\.+\"([a-z]+)<br>` LLD-Makro: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}<br>` Benutzermakro mit Kontext: `{$MACRO:"{{#MACRO}.iregsub(\"\.+\\\"[a-z]+\", \1)}"}`

Beachten Sie:

- die Backslash-Maskierung für LLD ändert sich nicht;
- beim Einfügen des LLD-Makros in den Benutzermakro-Kontext müssen wir es in eine Zeichenkette setzen:

1. Anführungszeichen werden um den Makroausdruck hinzugefügt;
2. Anführungszeichen werden maskiert; insgesamt werden 3 neue Backslashes eingefügt.

Zeichenkettenparameter einer Funktion (beliebig)

`concat` wird als Beispiel verwendet.

Zeichenketteninhalt: `\.+\"[a-z]+<br>` Ausdruck: `concat("abc", "\\.\\\"[a-z]+")`

Beachten Sie:

- Zeichenkettenparameter erfordern die Maskierung sowohl von Backslashes als auch von Anführungszeichen.

LLD-Makrofunktion innerhalb eines Zeichenkettenparameters einer Funktion

Regulärer Ausdruck: `\.+\"([a-z]+)<br>` LLD-Makro: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}<br>` Ausdruck: `concat("abc, "{{#MACRO}.iregsub(\"\\.+\\\\\"([a-z]+)\", \\1)}")`

Beachten Sie:

- Zeichenkettenparameter erfordern die Maskierung sowohl von Backslashes als auch von Anführungszeichen;
- eine weitere Maskierungsebene wird hinzugefügt, weil das Makro erst aufgelöst wird, nachdem die Zeichenkette ohne Anführungszeichen vorliegt;

Benutzermakro mit Kontext innerhalb eines Zeichenkettenparameters einer Funktion

Regulärer Ausdruck: `\.+\"[a-z]+<br>` Benutzermakro mit Kontext: `{$MACRO:regex:"\.+\\"[a-z]+"}<br>` Ausdruck: `concat("abc, "{$MACRO:regex:\"\\.+\\\\\"[a-z]+\"}")`

Beachten Sie:

- wie im vorherigen Beispiel ist eine zusätzliche Maskierungsebene erforderlich;
- Backslashes und Anführungszeichen werden nur für die Maskierung auf oberster Ebene maskiert (da es sich um einen Zeichenkettenparameter handelt).

LLD-Makrofunktion innerhalb eines Benutzermakro-Kontexts innerhalb einer Funktion

Regulärer Ausdruck: `\.+\"([a-z])+<br>` LLD-Makro: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}<br>` Benutzermakro mit Kontext: `{$MACRO:"{{#MACRO}.iregsub(\"\.+\\\"([a-z]+)\", \1)}"}<br>` Ausdruck: `concat("abc, "{$MACRO:\"{{#MACRO}.iregsub(\\\"\.+\\\\\\\"([a-z]+)\\\", \\1)}\"}")`

Beachten Sie die drei Maskierungsebenen:

1. Für die LLD-Makrofunktion, ohne Maskierung von Backslashes;
2. Für das Benutzermakro mit Kontext, ohne Maskierung von Backslashes;
3. Für den Zeichenkettenparameter einer Funktion, mit Maskierung von Backslashes.

Benutzermakro mit Kontext direkt innerhalb einer Zeichenkette

Regulärer Ausdruck: `\.+\"[a-z]+<br>` Benutzermakro mit Kontext: `{$MACRO:regex:"\.+\\"[a-z]+"}<br>` Innerhalb der Zeichenkette eines beliebigen Ausdrucks, zum Beispiel: `func(arg1, arg2, arg3)="{$MACRO:regex:\"\\.+\\\\\"[a-z]+\"}"`

Beachten Sie:

- Zeichenketten erfordern ebenfalls die Maskierung von Backslashes;
- Zeichenketten erfordern ebenfalls die Maskierung von Anführungszeichen;
- erneut ein Fall mit 2 Maskierungsebenen:

1. Maskierung für den Benutzermakro-Kontext ohne Backslash-Maskierung;
2. Maskierung dafür, dass es sich um eine Zeichenkette mit Backslash-Maskierung handelt.

Was ist Zabbix

Zabbix wurde von Alexei Vladishev entwickelt und wird derzeit aktiv von Zabbix SIA weiterentwickelt und unterstützt.

Zabbix ist eine verteilte Open-Source-Monitoring-Lösung der Enterprise-Klasse.

Zabbix ist eine Software, die zahlreiche Parameter eines Netzwerks sowie den Zustand und die Integrität von Servern, virtuellen Maschinen, Anwendungen, Diensten, Datenbanken, Websites, der Cloud und vielem mehr überwacht. Zabbix verwendet einen flexiblen Benachrichtigungsmechanismus, mit dem Benutzer E-Mail-basierte Warnungen für praktisch jedes Ereignis konfigurieren können. Dies ermöglicht eine schnelle Reaktion auf Serverprobleme. Zabbix bietet hervorragende Funktionen für Berichte und Datenvisualisierung auf Grundlage der gespeicherten Daten. Dadurch eignet sich Zabbix ideal für die Kapazitätsplanung.

Zabbix unterstützt sowohl Polling als auch Trapping. Auf alle Zabbix-Berichte und -Statistiken sowie Konfigurationsparameter wird über ein webbasiertes Frontend zugegriffen. Ein webbasiertes Frontend stellt sicher, dass der Status Ihres Netzwerks und der Zustand Ihrer Server von jedem Standort aus beurteilt werden können. Bei korrekter Konfiguration kann Zabbix eine wichtige Rolle bei der Überwachung der IT-Infrastruktur spielen. Dies gilt gleichermaßen für kleine Organisationen mit wenigen Servern und für große Unternehmen mit einer Vielzahl von Servern.

Zabbix ist kostenlos verfügbar. Zabbix wurde unter der AGPL-3.0-Lizenz entwickelt und wird unter dieser Lizenz verbreitet. Das bedeutet, dass der Quellcode frei verbreitet wird und der Allgemeinheit zur Verfügung steht.

Kommerzieller Support ist verfügbar und wird von der Zabbix Company und ihren Partnern weltweit angeboten.

Erfahren Sie mehr über Zabbix-Funktionen.

Benutzer von Zabbix Viele Organisationen unterschiedlicher Größe auf der ganzen Welt verlassen sich auf Zabbix als primäre Monitoring-Plattform.

Architektur Zabbix besteht aus mehreren wichtigen Softwarekomponenten. Ihre Zuständigkeiten sind unten aufgeführt.

Server

Zabbix Server ist die zentrale Komponente, an die Agents Verfügbarkeits- und Integritätsinformationen sowie Statistiken melden. Der Server ist das zentrale Repository, in dem alle Konfigurations-, Statistik- und Betriebsdaten gespeichert werden.

Datenbankspeicher

Alle Konfigurationsinformationen sowie die von Zabbix gesammelten Daten werden in einer Datenbank gespeichert.

Web-Interface

Für einen einfachen Zugriff auf Zabbix von überall und von jeder Plattform aus wird die webbasierte Oberfläche bereitgestellt. Die Oberfläche ist Teil des Zabbix Server und läuft in der Regel (aber nicht zwingend) auf demselben physischen Rechner wie der Server.

Proxy

Zabbix proxy kann Leistungs- und Verfügbarkeitsdaten im Auftrag des Zabbix Server erfassen. Ein Proxy ist ein optionaler Bestandteil einer Zabbix-Installation; er kann jedoch sehr nützlich sein, um die Last eines einzelnen Zabbix Server zu verteilen.

Agent

Zabbix Agenten werden auf Überwachungszielen bereitgestellt, um lokale Ressourcen und Anwendungen aktiv zu überwachen und die gesammelten Daten an den Zabbix Server zu melden. Seit Zabbix 4.4 sind zwei Arten von Agenten verfügbar: der Zabbix agent (ressourcenschonend, auf vielen Plattformen unterstützt, in C geschrieben) und der Zabbix agent 2 (besonders flexibel, leicht durch Plugins erweiterbar, in Go geschrieben).

Datenfluss Darüber hinaus ist es wichtig, einen Schritt zurückzutreten und sich den gesamten Datenfluss innerhalb von Zabbix anzusehen. Um einen Datenpunkt zu erstellen, der Daten sammelt, müssen Sie zunächst einen Host erstellen. Am anderen Ende des Zabbix-Spektrums müssen Sie zunächst einen Datenpunkt haben, um einen Auslöser zu erstellen. Sie müssen einen Auslöser haben, um eine Aktion zu erstellen. Wenn Sie also eine Benachrichtigung erhalten möchten, dass Ihre CPU-Auslastung auf *Server X* zu hoch ist, müssen Sie zunächst einen Host-Eintrag für *Server X* erstellen, anschließend einen Datenpunkt zur Überwachung der CPU, dann einen Auslöser, der aktiviert wird, wenn die CPU zu hoch ist, und danach eine Aktion, die Ihnen eine E-Mail sendet. Auch wenn das nach vielen Schritten klingt, ist es mit der Verwendung von Vorlagen in Wirklichkeit nicht so. Durch dieses Design ist jedoch eine sehr flexible Konfiguration möglich.

Funktionen Zabbix ist eine hochintegrierte Lösung zur Netzwerküberwachung, die eine Vielzahl von Funktionen in einem einzigen Paket bietet.

Datenerfassung

- Verfügbarkeits- und Leistungsprüfungen
- Unterstützung für SNMP (sowohl Trapping als auch Polling), IPMI, JMX, VMware-Überwachung
- benutzerdefinierte Prüfungen
- Erfassung gewünschter Daten in benutzerdefinierten Intervallen
- durchgeführt von Server/Proxy und von Agenten

Flexible Schwellenwertdefinitionen

- Sie können sehr flexible Problem-Schwellenwerte definieren, sogenannte Auslöser, die auf Werte aus der Backend-Datenbank verweisen

Hochgradig konfigurierbare Benachrichtigungen

- das Senden von Benachrichtigungen kann für den Eskalationsplan, den Empfänger und den Medientyp angepasst werden
- Benachrichtigungen können mithilfe von Makrovariablen aussagekräftig und hilfreich gestaltet werden
- automatische Aktionen umfassen Remote-Befehle

Diagramme in Echtzeit

- überwachte Datenpunkte werden sofort mit der integrierten Diagrammfunktion dargestellt

Web-Überwachungsfunktionen

- Zabbix kann einem Pfad simulierter Mausklicks auf einer Website folgen und Funktionalität sowie Antwortzeit prüfen

Umfangreiche Visualisierungsoptionen

- Möglichkeit, benutzerdefinierte Diagramme zu erstellen, die mehrere Datenpunkte in einer einzigen Ansicht kombinieren können
- Netzwerkkarten
- Diashows in einer Dashboard-ähnlichen Übersicht
- Berichte
- Übersicht auf hoher Ebene (Geschäftsansicht) der überwachten Ressourcen

Speicherung historischer Daten

- Daten werden in einer Datenbank gespeichert
- konfigurierbare Historie
- integrierte Housekeeping-Prozedur

Einfache Konfiguration

- überwachte Geräte als Hosts hinzufügen
- Hosts werden zur Überwachung übernommen, sobald sie in der Datenbank vorhanden sind
- Vorlagen auf überwachte Geräte anwenden

Verwendung von Vorlagen

- Prüfungen in Vorlagen gruppieren
- Vorlagen können andere Vorlagen erben

Netzwerkerkennung

- automatische Erkennung von Netzwerkgeräten
- automatische Registrierung von Agenten
- Erkennung von Dateisystemen, Netzwerkschnittstellen und SNMP-OIDs

Schnelle Weboberfläche

- eine webbasierte Frontend in PHP
- von überall aus zugänglich
- Sie können sich per Klick durch die Oberfläche bewegen
- Audit-Log

Zabbix API

- Die Zabbix API bietet eine programmierbare Schnittstelle zu Zabbix für Massenänderungen, die Integration von Drittanbietersoftware und andere Zwecke.

Berechtigungssystem

- sichere Benutzerauthentifizierung
- bestimmte Benutzer können auf bestimmte Ansichten beschränkt werden

Voll ausgestatteter und leicht erweiterbarer Agent

- auf Überwachungszielen installiert
- kann sowohl unter Linux als auch unter Windows eingesetzt werden

Binäre Daemons

- in C geschrieben, für hohe Leistung und geringen Speicherbedarf
- leicht portierbar

Bereit für komplexe Umgebungen

- Fernüberwachung wird durch die Verwendung eines Zabbix Proxy vereinfacht

Glossar In diesem Abschnitt erfahren Sie die Bedeutung einiger häufig in Zabbix verwendeter Begriffe.

Host

- jedes physische oder virtuelle Gerät, jede Anwendung, jeder Dienst oder jede andere logisch zusammengehörige Sammlung überwachter Parameter.

Hostgruppe

- eine logische Gruppierung von Hosts. Hostgruppen werden verwendet, wenn verschiedenen Benutzergruppen Zugriffsrechte auf Hosts zugewiesen werden.

Datenpunkt

- ein bestimmtes Datenelement, das Sie von einem Host empfangen möchten, also eine Datenmetrik.

Wertvorverarbeitung

- eine Transformation des empfangenen Metrikwerts, bevor dieser in der Datenbank gespeichert wird.

Auslöser

- ein logischer Ausdruck, der einen Problemschwellenwert definiert und zum „Auswerten“ der in Datenpunkten empfangenen Daten verwendet wird.

Wenn die empfangenen Daten über dem Schwellenwert liegen, wechseln Auslöser vom Status *OK* in den Status *Problem*. Wenn die empfangenen Daten unter dem Schwellenwert liegen, bleiben Auslöser im Status *OK* oder kehren in diesen zurück.

Vorlage

- eine Gruppe von Entitäten (Datenpunkte, Auslöser, Diagramme, Regeln für die Low-Level-Discovery und Web-Szenarien), die auf einen oder mehrere Hosts angewendet werden kann.

Vorlagen dienen dazu, die Bereitstellung von Überwachungsaufgaben auf einem Host zu beschleunigen und Massenänderungen an Überwachungsaufgaben zu vereinfachen. Vorlagen werden direkt mit einzelnen Hosts verknüpft.

Vorlagengruppe

- eine logische Gruppierung von Vorlagen. Vorlagengruppen werden verwendet, wenn verschiedenen Benutzergruppen Zugriffsrechte auf Vorlagen zugewiesen werden.

Ereignis

- das einmalige Auftreten eines Ereignisses, das Aufmerksamkeit verdient, beispielsweise eine Statusänderung eines Auslösers oder die Durchführung einer Discovery bzw. einer automatischen Registrierung eines Agents.

Ereignis-Tag

- eine vordefinierte Markierung für das Ereignis. Es kann unter anderem für die Ereigniskorrelation und die differenzierte Vergabe von Berechtigungen verwendet werden.

Ereigniskorrelation

- eine Methode, um Probleme flexibel und präzise ihrer Lösung zuzuordnen.

Beispielsweise können Sie festlegen, dass ein von einem Auslöser gemeldetes Problem durch einen anderen Auslöser gelöst werden kann, der möglicherweise sogar eine andere Datenerfassungsmethode verwendet.

Problem

- ein Auslöser, der sich im Status „Problem“ befindet.

Problemaktualisierung

- von Zabbix bereitgestellte Optionen zur Problemverwaltung, beispielsweise das Hinzufügen eines Kommentars, die Bestätigung, die Änderung des Schweregrads oder das manuelle Schließen.

Aktion

- eine vordefinierte Möglichkeit, auf ein Ereignis zu reagieren.

Eine Aktion besteht aus Vorgängen (z. B. dem Senden einer Benachrichtigung) und Bedingungen (*wann* der Vorgang ausgeführt wird).

Eskalation

- ein benutzerdefiniertes Szenario zur Ausführung von Vorgängen innerhalb einer Aktion; eine Abfolge zum Senden von Benachrichtigungen bzw. Ausführen von Remote-Befehlen.

Makro

- ein Platzhalter für eine Variable, der abhängig vom Kontext in einen bestimmten Wert aufgelöst wird.

Medium

- eine Möglichkeit zur Zustellung von Benachrichtigungen; ein Zustellungskanal.

Benachrichtigung

- eine Nachricht über ein Ereignis, die über den ausgewählten Medienkanal an einen Benutzer gesendet wird.

Remote-Befehl

- ein vordefinierter Befehl, der bei Eintreten einer bestimmten Bedingung automatisch auf einem überwachten Host ausgeführt wird.

Web-Szenario

- eine oder mehrere HTTP-Anfragen zur Überprüfung der Verfügbarkeit einer Website.

Frontend

- die mit Zabbix bereitgestellte Weboberfläche.

Dashboard

- ein anpassbarer Bereich der Weboberfläche, der Zusammenfassungen und Visualisierungen wichtiger Informationen in visuellen Einheiten, den sogenannten Widgets, anzeigt.

Widget

- eine visuelle Einheit, die Informationen einer bestimmten Art und Quelle anzeigt (eine Zusammenfassung, eine Karte, ein Diagramm, die Uhr usw.) und im Dashboard verwendet wird.

Zabbix-API

- die Zabbix-API ermöglicht die Verwendung des JSON-RPC-Protokolls zum Erstellen, Aktualisieren und Abrufen von Zabbix-Objekten (z. B. Hosts, Datenpunkten, Diagrammen und anderen Objekten) sowie zum Ausführen beliebiger anderer Aufgaben.

Zabbix-Server

- ein zentraler Prozess der Zabbix-Software, der die Überwachung durchführt, mit Zabbix-Proxys und Agents interagiert, Auslöser berechnet, Benachrichtigungen sendet und als zentrales Daten-Repository dient.

Zabbix-Proxy

- ein Prozess, der Daten im Auftrag des Zabbix-Servers erfassen und dadurch den Server bei der Verarbeitung entlasten kann.

Zabbix-Agent

- ein Prozess, der auf Überwachungszielen bereitgestellt wird, um lokale Ressourcen und Anwendungen aktiv zu überwachen.

Zabbix-Agent 2

- eine neue Generation des Zabbix-Agents zur aktiven Überwachung lokaler Ressourcen und Anwendungen, die die Verwendung benutzerdefinierter Plugins für die Überwachung ermöglicht.

Attention:
Da der Zabbix-Agent 2 viele Funktionen mit dem Zabbix-Agent gemeinsam nutzt, steht der Begriff „Zabbix-Agent“ in der Dokumentation für beide - den Zabbix-Agent und den Zabbix-Agent 2 -, wenn das funktionale Verhalten identisch ist. Der Zabbix-Agent 2 wird nur dann ausdrücklich genannt, wenn sich seine Funktionalität unterscheidet.

Verschlüsselung

- *Unterstützung verschlüsselter Kommunikation zwischen Zabbix-Komponenten (Server, Proxy, Agent, zabbix_sender- und zabbix_get-Dienstprogramme)* unter Verwendung des Transport Layer Security (TLS)-Protokolls.

Automatische Registrierung des Agents

- *ein automatisierter Prozess, bei dem ein Zabbix-Agent selbst als Host registriert* und die Überwachung gestartet wird.

Netzwerk-Discovery

- *die automatisierte Erkennung von Netzwerkgeräten.*

Low-Level-Discovery

- *die automatisierte Erkennung von Entitäten niedriger Ebene auf einem bestimmten Gerät* (z. B. Dateisysteme, Netzwerkschnittstellen usw.).

Regel für die Low-Level-Discovery

- *eine Gruppe von Definitionen für die automatisierte Erkennung von Entitäten niedriger Ebene* auf einem Gerät.

Datenpunkt-Prototyp

- *eine Metrik mit bestimmten Parametern als Variablen, die für die Low-Level-Discovery bereitsteht.* Nach der Low-Level-Discovery werden die Variablen automatisch durch die tatsächlich erkannten Parameter ersetzt, und die Metrik beginnt automatisch mit der Datenerfassung.

Auslöser-Prototyp

- *ein Auslöser mit bestimmten Parametern als Variablen, der für die Low-Level-Discovery bereitsteht.* Nach der Low-Level-Discovery werden die Variablen automatisch durch die tatsächlich erkannten Parameter ersetzt, und der Auslöser beginnt automatisch mit der Auswertung der Daten.

Prototypen einiger anderer Zabbix-Entitäten werden ebenfalls bei der Low-Level-Discovery verwendet - Diagramm-Prototypen, Host-Prototypen und Hostgruppen-Prototypen.

Zabbix Cloud

Zabbix Cloud

Entdecken Sie die verfügbaren Konfigurationsoptionen und Zugriffskontrollen für Organisationen und Knoten in Zabbix Cloud.

Bitte beachten Sie, dass Zabbix-Cloud-Knoten derzeit mit **Zabbix 7.0** ausgeführt werden.
 Ausführliche Informationen zu Zabbix 7.0 finden Sie in der offiziellen Dokumentation.

Zabbix Cloud einrichten und verwalten

Erste Schritte mit Zabbix Cloud Geführte Anleitungen, um Ihre Zabbix Cloud-Instanz schnell in Betrieb zu nehmen, einschließlich der Erstkonfiguration und der Verbindung Ihrer überwachten Geräte.

Node-Konfiguration Erfahren Sie, wie Sie Ihren Zabbix Cloud-Knoten einrichten, mit Schwerpunkt auf Zugriff, Datenverarbeitung und Ressourceneinstellungen.

Benutzer hinzufügen Erfahren Sie, wie Sie Benutzer innerhalb einer Zabbix Cloud-Organisation und ihrer Knoten hinzufügen und ihnen Rollen zuweisen.

Erfahren Sie mehr über Zabbix Cloud

Zabbix Cloud vs. On-Premises Vergleichen Sie die Unterschiede bei Verwaltung, Skalierbarkeit und Wartung zwischen den beiden Bereitstellungen und erfahren Sie, welche Ihren Workflow am nahtlosesten unterstützen kann.

Zabbix Cloud erkunden Verschaffen Sie sich einen Überblick über Zabbix Cloud, einschließlich der wichtigsten Funktionen, Vorteile und dessen, was in einem Abonnement enthalten ist. Antworten auf häufige Fragen zur Einrichtung und Nutzung finden Sie in den FAQ.

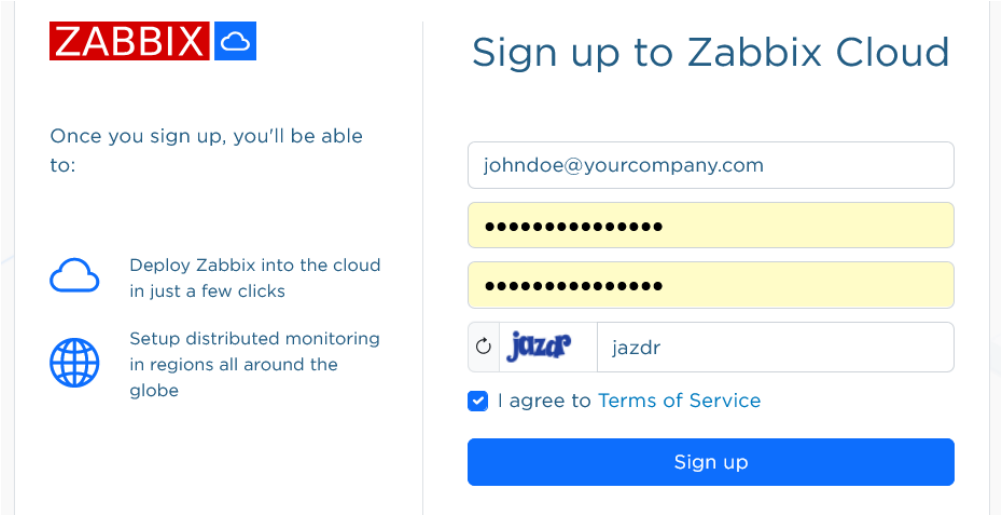
Zabbix in der Cloud bereitstellen

Um Zabbix in der Cloud zu starten, gehen Sie wie folgt vor:

1. Melden Sie sich bei cloud.zabbix.com an oder registrieren Sie sich.
2. Erstellen Sie einen Cloud-Knoten.
3. Beginnen Sie mit der Nutzung von Zabbix in der Cloud.

Anmelden oder registrieren Sie können sich bei cloud.zabbix.com mit einem bestehenden Google-, GitHub- oder Microsoft-Konto anmelden.

Alternativ können Sie die Seite Registrierung aufrufen, um sich zu registrieren.



Nach der Registrierung erhalten Sie eine Bestätigung Ihres neuen Kontos per E-Mail.

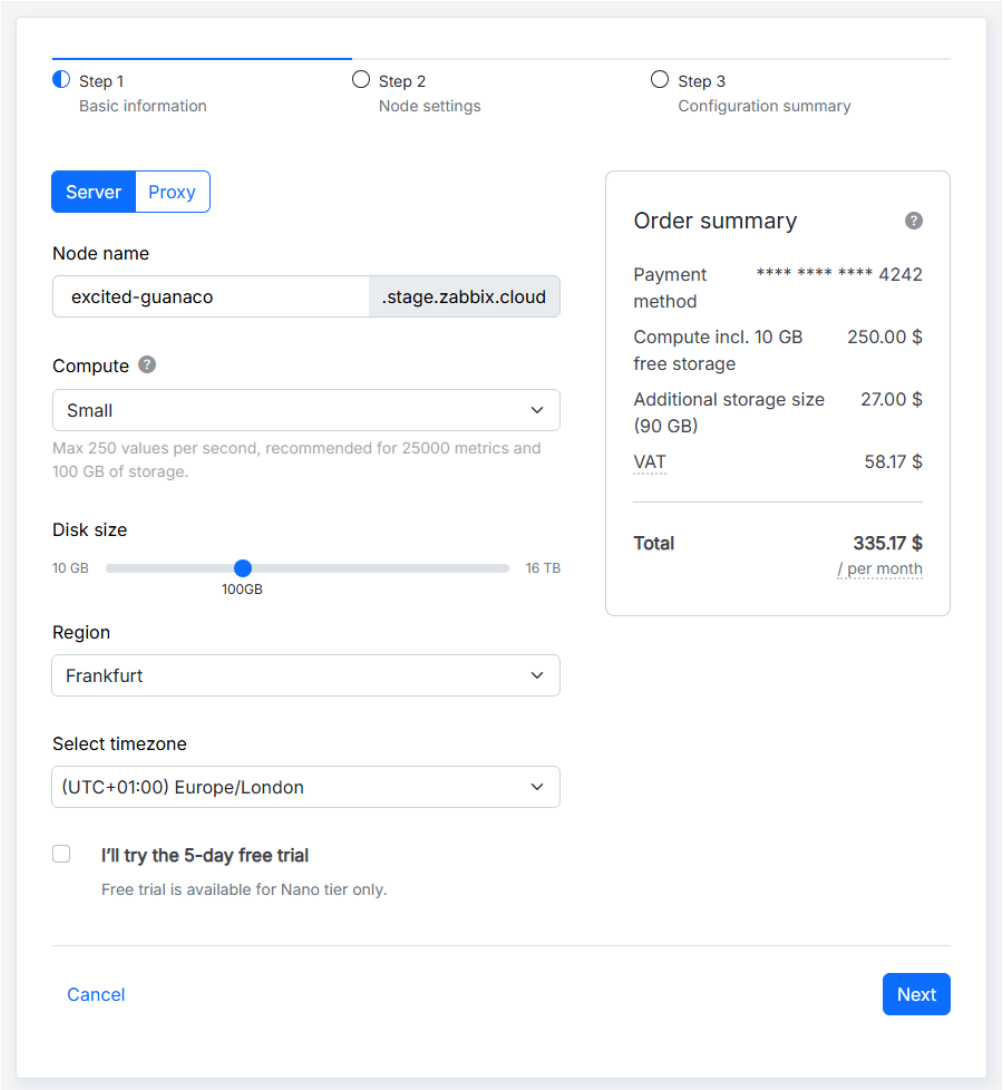
Rufen Sie cloud.zabbix.com auf, um sich anzumelden. Bei der Anmeldung werden Sie nach einem Einmalkennwort gefragt. Dieses Kennwort wird Ihnen per E-Mail zugesendet.

Cloud-Knoten erstellen Ein Cloud-Knoten ist eine in Zabbix Cloud bereitgestellte Zabbix-Instanz. Er kann als Server-Knoten oder Proxy-Knoten bereitgestellt werden.

Nach der Anmeldung sehen Sie, dass standardmäßig bereits eine *Private* Organisation für Ihr Konto vorhanden ist. Sie können die Standardorganisation später umbenennen.

Klicken Sie auf *Create New Node*. Ein dreistufiges Formular zur Knotenkonfiguration wird geöffnet.

Im ersten Schritt wählen Sie grundlegende Informationen aus.



Parameter	Beschreibung
Server/Proxy	Wählen Sie aus, ob eine Zabbix-Server-Instanz oder eine Zabbix-Proxy-Instanz bereitgestellt werden soll.
Node name	Der Knotenname bildet eine eindeutige Subdomain im Format `<your-name>.zabbix.cloud`, über die auf die Dienste des Knotens zugegriffen wird.
Compute	Wählen Sie die Rechenleistung für den Knoten aus. Informationen dazu, welche Ressourcenstufe Sie benötigen, finden Sie unter Preisinformationen. Sie können später jederzeit ein Upgrade durchführen, wenn Sie mehr Leistung benötigen. Vor der Erstellung eines Knotens, der kein Testknoten ist, muss eine Zahlungsmethode hinterlegt werden. Fügen Sie die Zahlungsinformationen über das Menü *Billing* hinzu. Zum Testen eines Knotens der Stufe *Nano* steht eine **kostenlose Testversion** mit einer Laufzeit von 5 Tagen zur Verfügung. Während des Testzeitraums fallen keine Kosten an. Nach Ablauf des Testzeitraums wird Ihr Knoten entfernt, sofern er nicht in einen kostenpflichtigen Knoten umgewandelt wurde.
Disk size	Der Schieberegler ist standardmäßig auf ein empfohlenes Minimum festgelegt, das auf der Stufe basiert. Erhöhen Sie den Wert, wenn Sie große Mengen an Verlaufsdaten oder Trends speichern. Beachten Sie, dass größere Festplatten höhere monatliche Kosten verursachen und dass Sie eine Festplatte nicht selbst verkleinern können. Für eine Verkleinerung müssen Sie den Support kontaktieren.
Region	Wählen Sie die Region aus, in der der Knoten bereitgestellt werden soll.
Select timezone	Wählen Sie die Zeitzone des Zabbix-Servers aus. Diese Zeitzone wird vom Zabbix-Server verwendet und ist unabhängig von den Zeitzonen für Wartungszeiträume und Benutzerprofile.

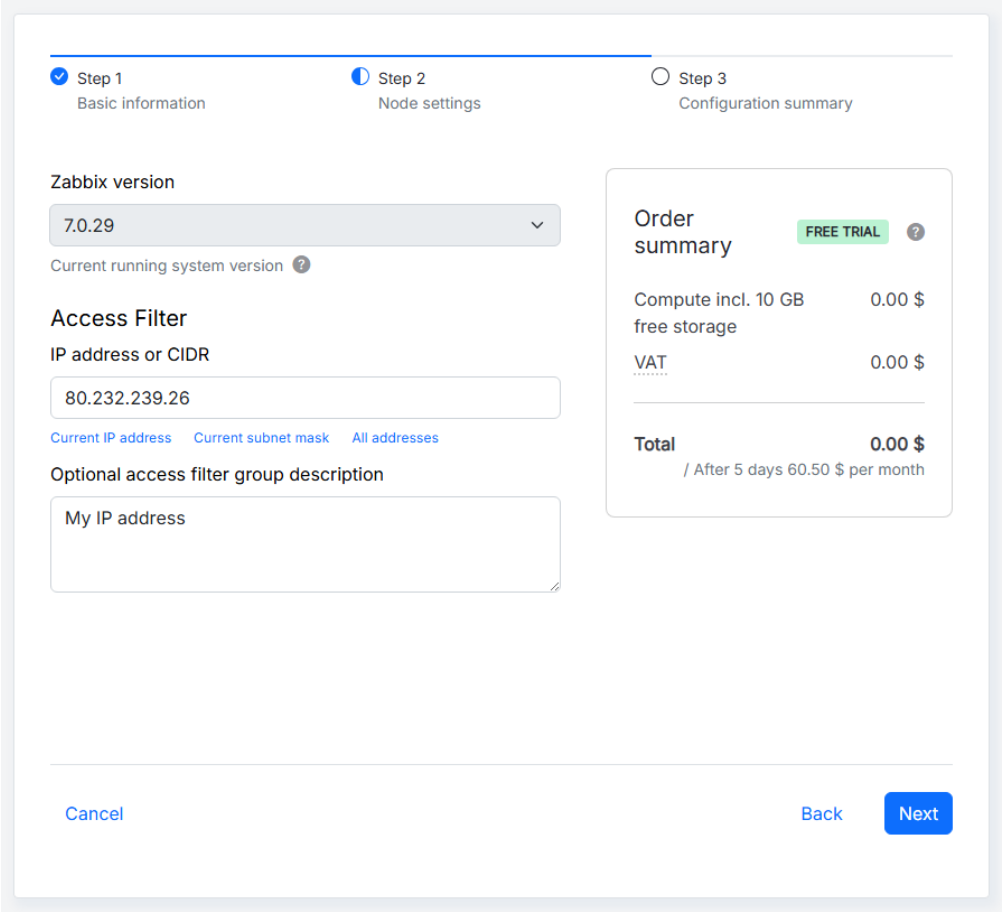
Nachdem Sie die Konfiguration abgeschlossen haben, klicken Sie auf *Next*.

Im zweiten Schritt wählen Sie die Knoteneinstellungen aus.

Die Konfigurationsoptionen im zweiten Schritt hängen davon ab, ob Sie einen Server- oder Proxy-Knoten erstellen.

Server-Knoten konfigurieren

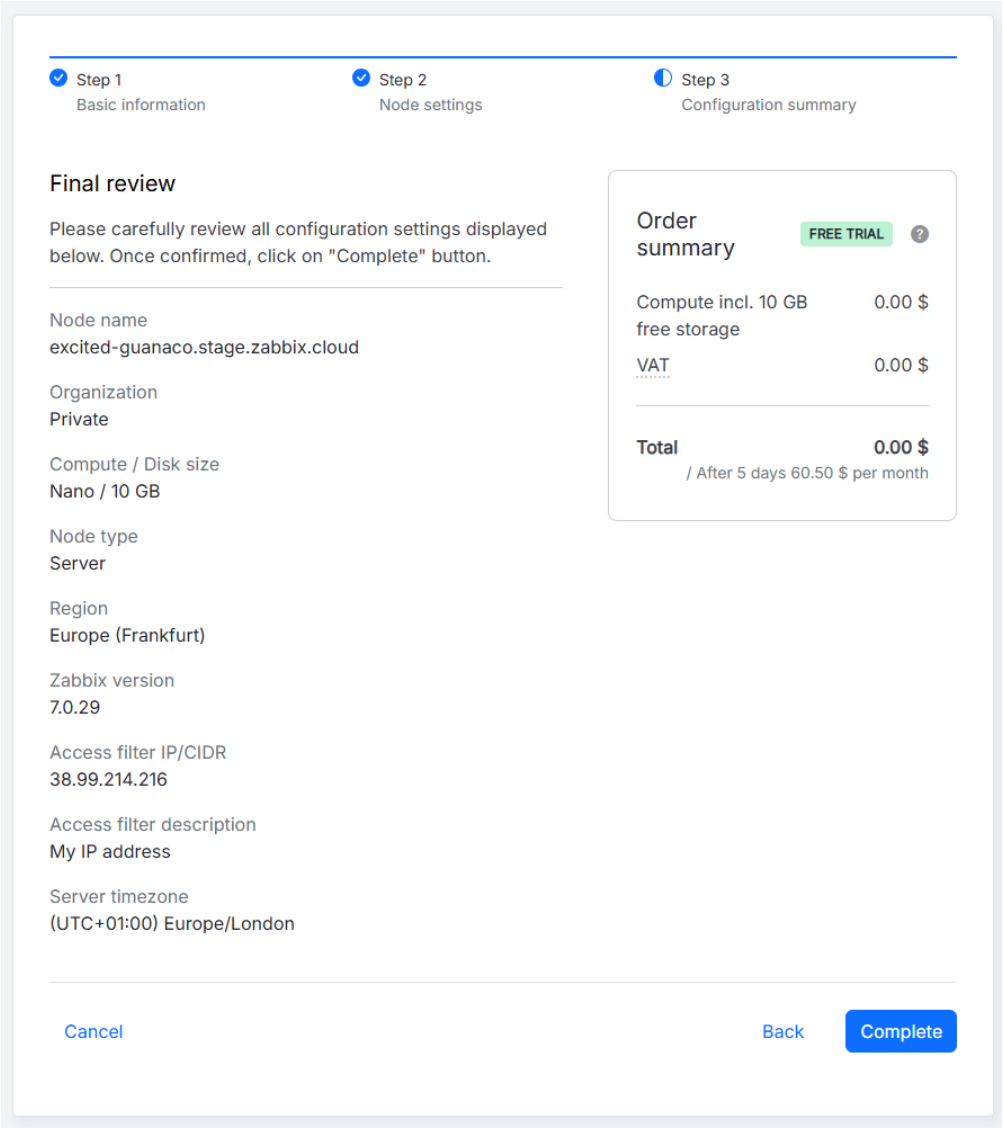
Bei der Erstellung eines Server-Knotens stehen die folgenden Parameter zur Verfügung:



Parameter	Beschreibung
Zabbix version	Wählen Sie die Zabbix-Version aus. Zabbix Cloud unterstützt die neueste LTS-Version und wendet automatisch Aktualisierungen von Nebenversionen an.
IP address or CIDR	Geben Sie den anfänglichen Zugriffsfilter für den Knoten an. Geben Sie die IP-Adresse oder CIDR manuell ein oder füllen Sie das Feld automatisch aus, indem Sie die entsprechende Option auswählen. Bei Server-Knoten gilt dieser Filter für Verbindungen zum Server, zur API und zum Frontend. Der Filter kann später im Tab „Access filters“ geändert werden.
Optional access filter group description	Fügen Sie optional eine Beschreibung für die Zugriffsfiltergruppe hinzu, die mehrere IP-Adressen oder CIDRs enthalten kann.

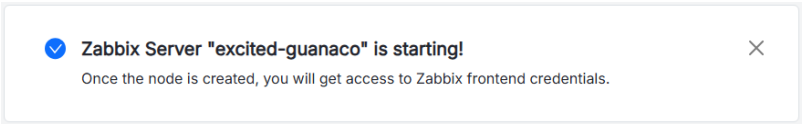
Warning:
Bei der Erstellung des Knotens wird das Feld *IP address or CIDR* mit der IP-Adresse vorausgefüllt, die für den Zugriff auf Zabbix Cloud verwendet wird. Sie können diesen Wert ändern, um einen anderen anfänglichen Zugriffsfilter festzulegen. Nach der Erstellung des Knotens können Sie Zugriffsfilter im Tab *Access filters* hinzufügen, bearbeiten oder löschen.

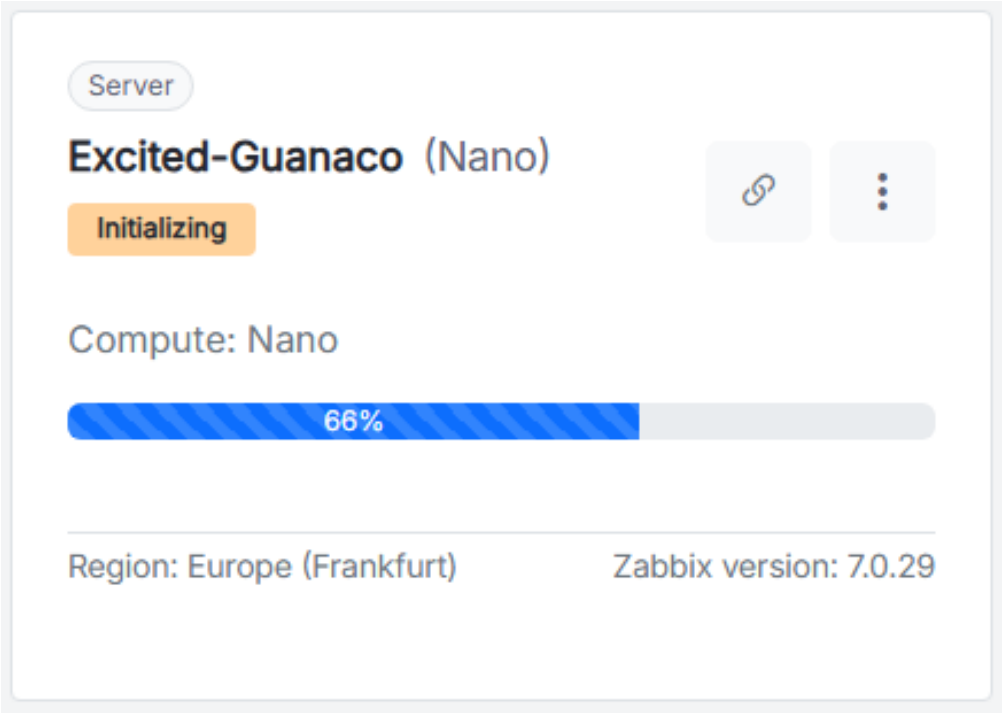
Im dritten Schritt wird eine Zusammenfassung der Knotenkonfiguration angezeigt.



Nachdem Sie die Konfigurationseinstellungen überprüft und bestätigt haben, dass alles korrekt ist, klicken Sie bei einem Testknoten auf *Complete* oder bei einem kostenpflichtigen Knoten auf *Proceed to payment*. Andernfalls können Sie durch Klicken auf die Schaltfläche *Back* zu einem vorherigen Schritt zurückkehren oder mit *Cancel* von vorne beginnen.

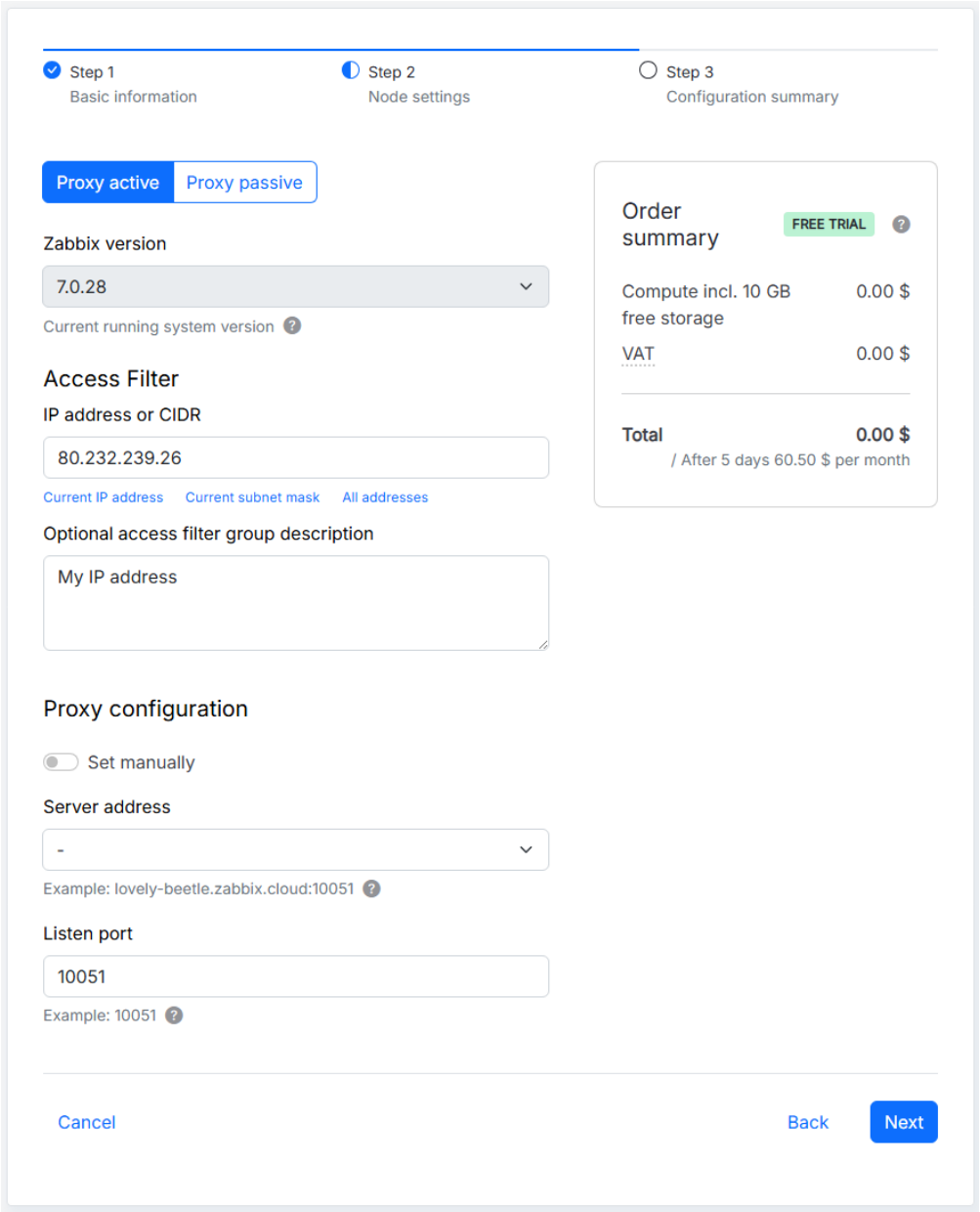
Während der Knoten initialisiert wird, wird eine entsprechende Meldung angezeigt, und der Knoten erscheint bereits unter *Organizations*.





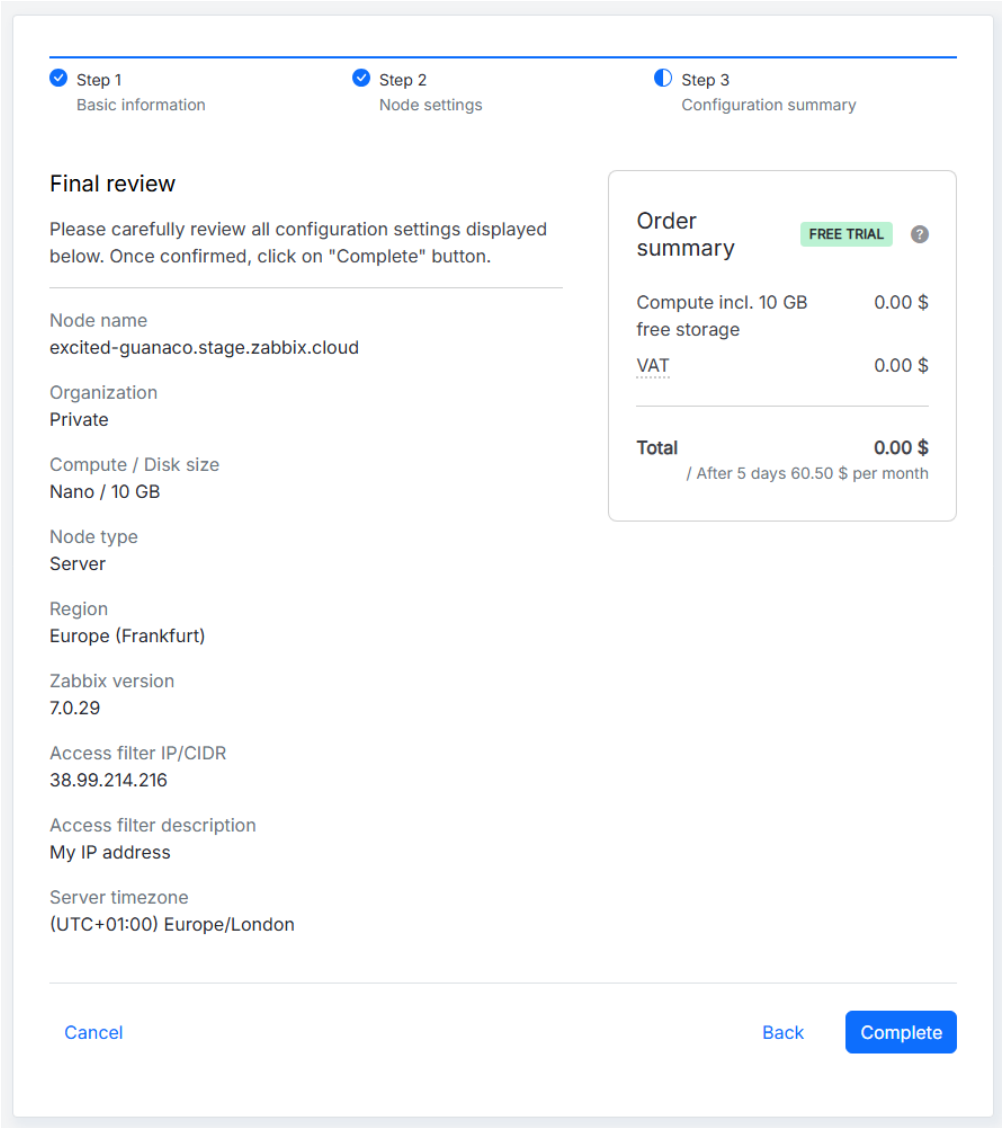
Proxy-Knoten konfigurieren

Bei der Erstellung eines Proxy-Knotens stehen die folgenden Parameter zur Verfügung:



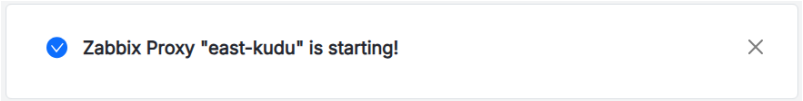
Parameter	Beschreibung
Proxy active/Proxy passive	Wählen Sie den Proxy-Modus aus. Im aktiven Modus initiiert der Proxy Verbindungen zum Zabbix-Server. Im passiven Modus initiiert der Zabbix-Server Verbindungen zum Proxy.
Zabbix version	Wählen Sie die Zabbix-Version aus. Zabbix Cloud unterstützt die neueste LTS-Version und wendet automatisch Aktualisierungen von Nebenversionen an.
IP address or CIDR	Geben Sie den anfänglichen eingehenden IP-Zugriffsfilter für den Proxy an. Geben Sie die IP-Adresse oder CIDR manuell ein oder füllen Sie das Feld automatisch aus, indem Sie die entsprechende Option auswählen. Der Filter legt fest, welche Netzwerkanforderungen den Proxy erreichen dürfen, und kann später im Tab „Access filters“ geändert werden.
Optional access filter group description	Fügen Sie optional eine Beschreibung für die Zugriffsfiltergruppe hinzu, die mehrere IP-Adressen oder CIDRs enthalten kann.
Set manually	Wählen Sie aus, wie der Zabbix-Server angegeben werden soll. Wenn diese Option deaktiviert ist, wählen Sie einen Server-Knoten aus der Liste der für Sie verfügbaren Server-Knoten aus. Wenn sie aktiviert ist, geben Sie die Serveradressen manuell an.
Server address	Geben Sie die Adresse des Zabbix-Servers an. Wenn *Set manually* deaktiviert ist, wählen Sie einen Server-Knoten aus der Liste aus. Wenn die Option aktiviert ist, können Sie bis zu 10 Serveradressen angeben. Geben Sie bei einem aktiven Proxy die IP-Adresse oder den DNS-Namen des Zabbix-Servers an. Geben Sie bei einem passiven Proxy die IP-Adresse, den DNS-Namen oder die CIDR des Zabbix-Servers an. Zugriffsfilter werden separat konfiguriert und bei der Angabe einer Serveradresse nicht automatisch aktualisiert.
Listen port	Geben Sie den Port an, an dem der Proxy auf eingehende Verbindungen wartet. Der Standardport ist 10051. Der Überwachungsport kann nach der Erstellung des Knotens nicht geändert werden.

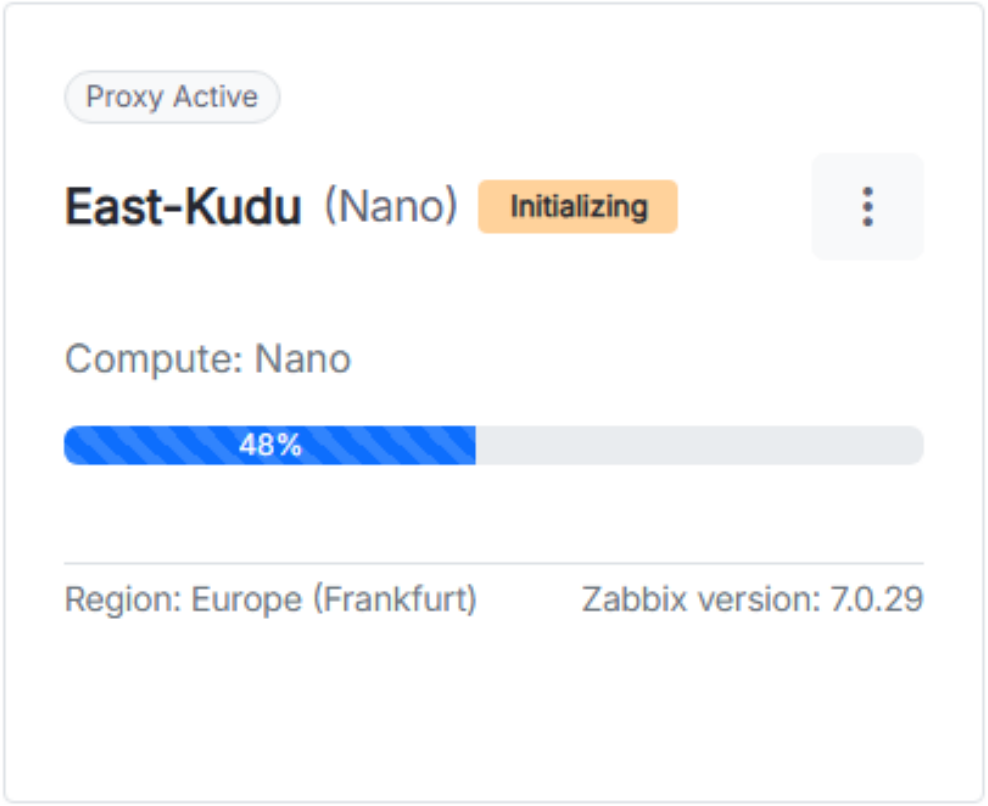
Im dritten Schritt wird eine Zusammenfassung der Knotenkonfiguration angezeigt.



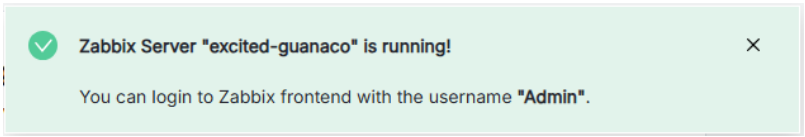
Nachdem Sie die Konfigurationseinstellungen überprüft und bestätigt haben, dass alles korrekt ist, klicken Sie bei einem Testknoten auf *Complete* oder bei einem kostenpflichtigen Knoten auf *Proceed to payment*. Andernfalls können Sie durch Klicken auf die Schaltfläche *Back* zu vorherigen Schritten zurückkehren oder mit *Cancel* von vorne beginnen.

Während der Knoten initialisiert wird, wird eine entsprechende Meldung angezeigt, und der Knoten erscheint bereits unter *Organizations*.





Informationen zur Konfiguration eines Zabbix-Cloud-Proxys finden Sie unter Zabbix-Cloud-Proxy einrichten.

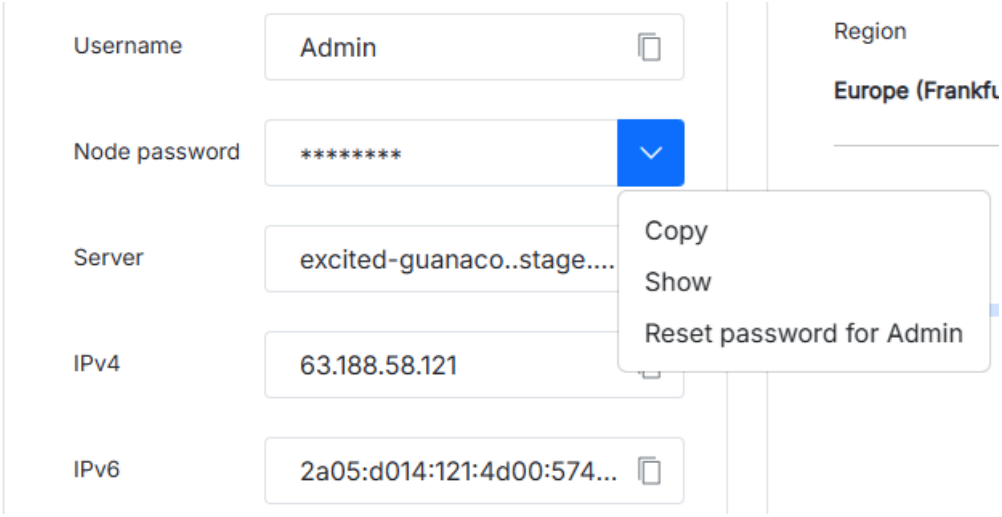


Zabbix verwenden

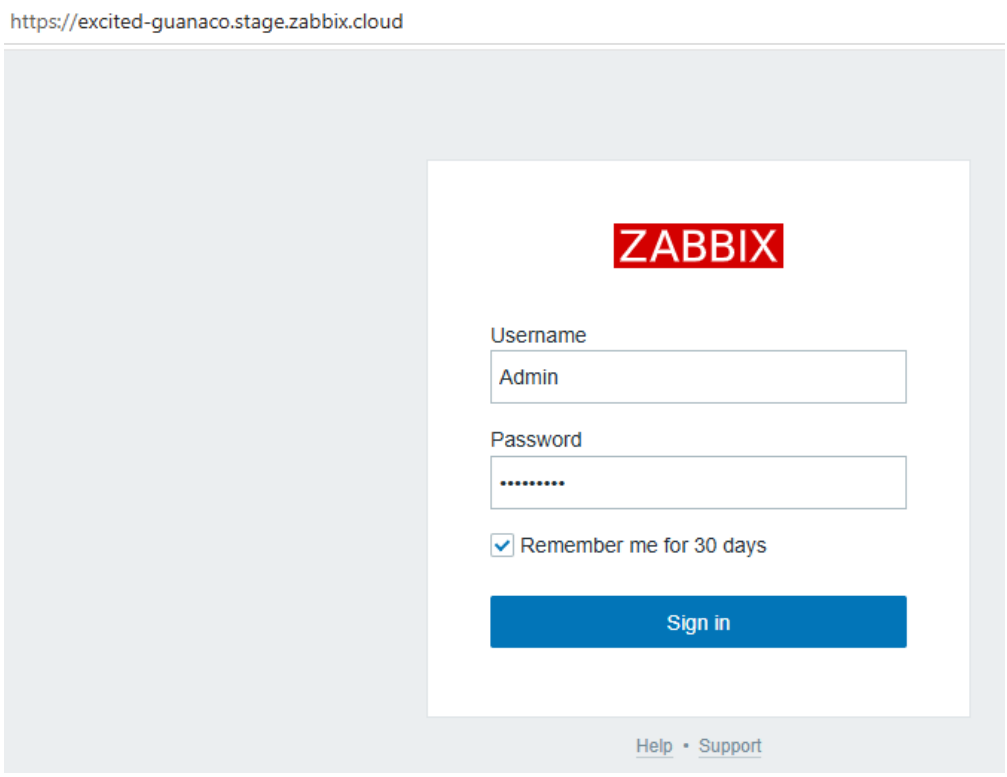
Wenn sich der Status des Serverknotens von *Initializing* zu *Running* ändert, können Sie sich beim Zabbix Frontend anmelden.

Um auf das anfängliche Frontend-Passwort zuzugreifen, klicken Sie auf den erstellten Knoten.

Sie werden zu den Einstellungen des Knotens weitergeleitet (Übersicht, Zugriffsfilter, Verschlüsselung, Backups usw.). Suchen Sie unter **Verbindungsinformationen** das Feld *Knotenpasswort* und wählen Sie im Dropdown-Menü *Passwort für Admin zurücksetzen* aus. Nachdem das Passwort zurückgesetzt wurde, klicken Sie auf dasselbe Dropdown-Menü und wählen Sie *Kopieren* aus.



Klicken Sie auf die Frontend-URL, um das Zabbix Frontend zu öffnen (`excited-guanaco.zabbix.cloud`).



Melden Sie sich mit den folgenden Anmeldedaten an:

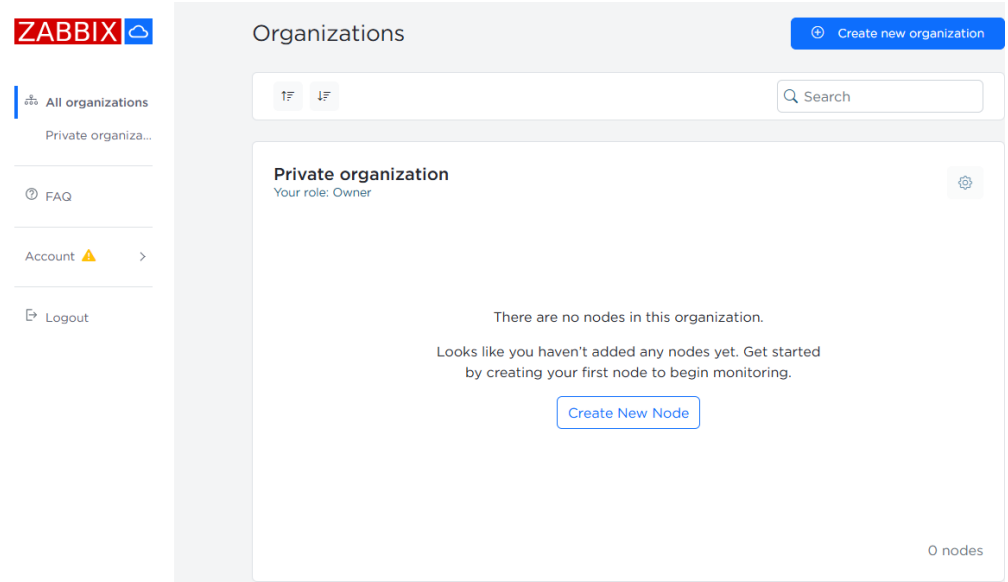
- *Benutzername* - `Admin`
- *Passwort* - geben Sie das kopierte Passwort ein

Um die Überwachung zu starten, müssen Sie Zugriffsfilter definieren - IP-Adresse oder IP-Adressen, die eine Verbindung zum Zabbix Cloud Server herstellen dürfen. Mindestens eine muss definiert werden.

Organisationsverwaltung Eine Organisation ist ein Container für einen oder mehrere Knoten. Organisationen helfen Ihnen außerdem dabei, Ihre Ressourcen zu strukturieren und Verantwortlichkeiten zu delegieren, zum Beispiel:

- Knotenverwaltung
- Abrechnung

Wenn Sie sich bei Zabbix Cloud anmelden, wird die Seite *Alle Organisationen* angezeigt. Standardmäßig wird für Ihr Konto eine private Organisation erstellt.

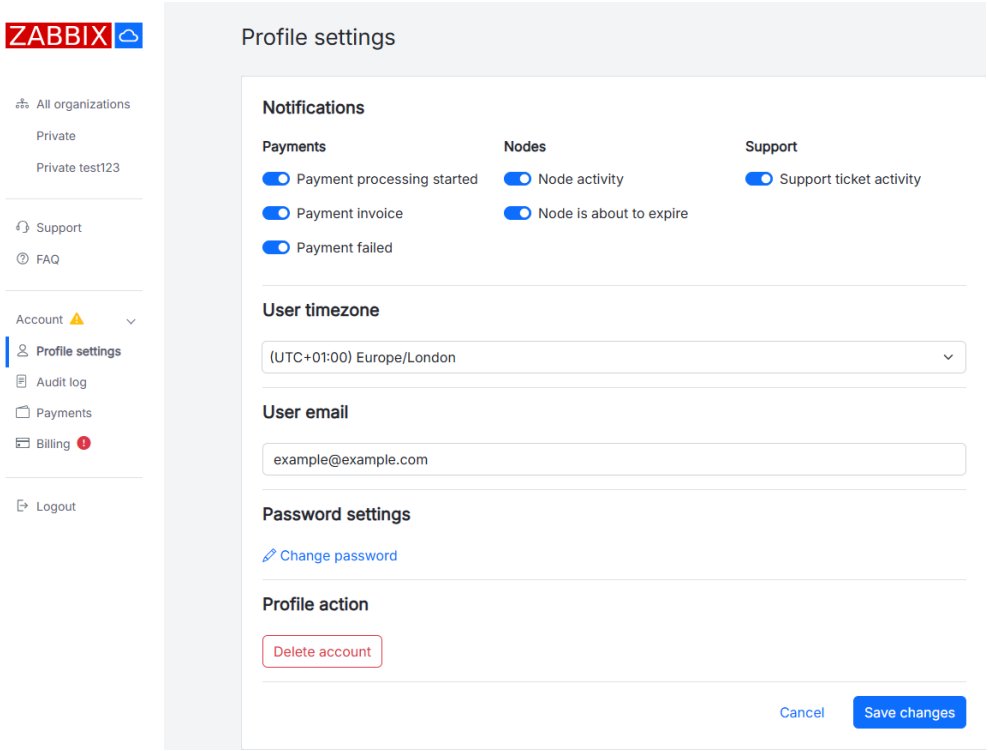


Als Eigentümer dieser Organisation können Sie Benutzer, Administratoren, Knoten und den Organisationsnamen verwalten. Klicken Sie auf die Schaltfläche  für Einstellungen, um auf die Verwaltungsoptionen zuzugreifen.

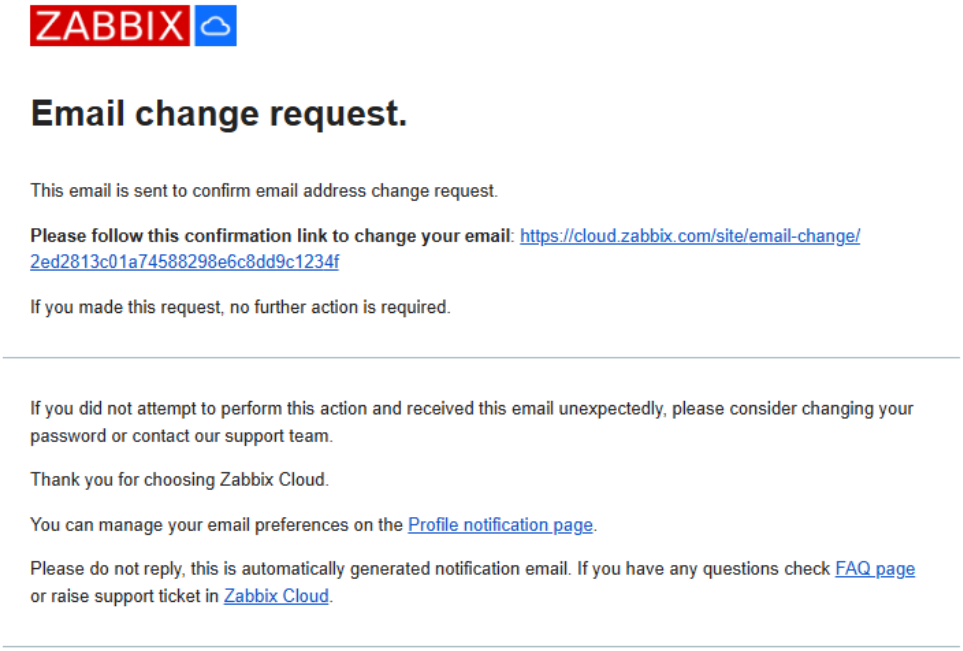
Kontoverwaltung Ändern der E-Mail-Adresse des Kontos

Um die E-Mail-Adresse des Kontos zu ändern, navigieren Sie zu *Konto* > *Profileinstellungen*.

Geben Sie im Feld *Benutzer-E-Mail* die neue E-Mail-Adresse ein und klicken Sie auf *Änderungen speichern*.



Eine Bestätigungsnachricht wird an die neue E-Mail-Adresse gesendet. Folgen Sie dem Link in der Nachricht, um die Änderung der E-Mail-Adresse zu bestätigen.



Der Bestätigungslink öffnet die Seite *Profileinstellungen* mit einer Bestätigungsnachricht und der aktualisierten E-Mail-Adresse im Feld „Benutzer-E-Mail“.

Nachdem die Änderung bestätigt wurde, wird eine E-Mail zur Bestätigung der Änderung sowohl an die neue als auch an die vorherige E-Mail-Adresse gesendet. Die an die vorherige Adresse gesendete Nachricht enthält die vorherige und die neue E-Mail-Adresse.

Node-Konfiguration

Übersicht

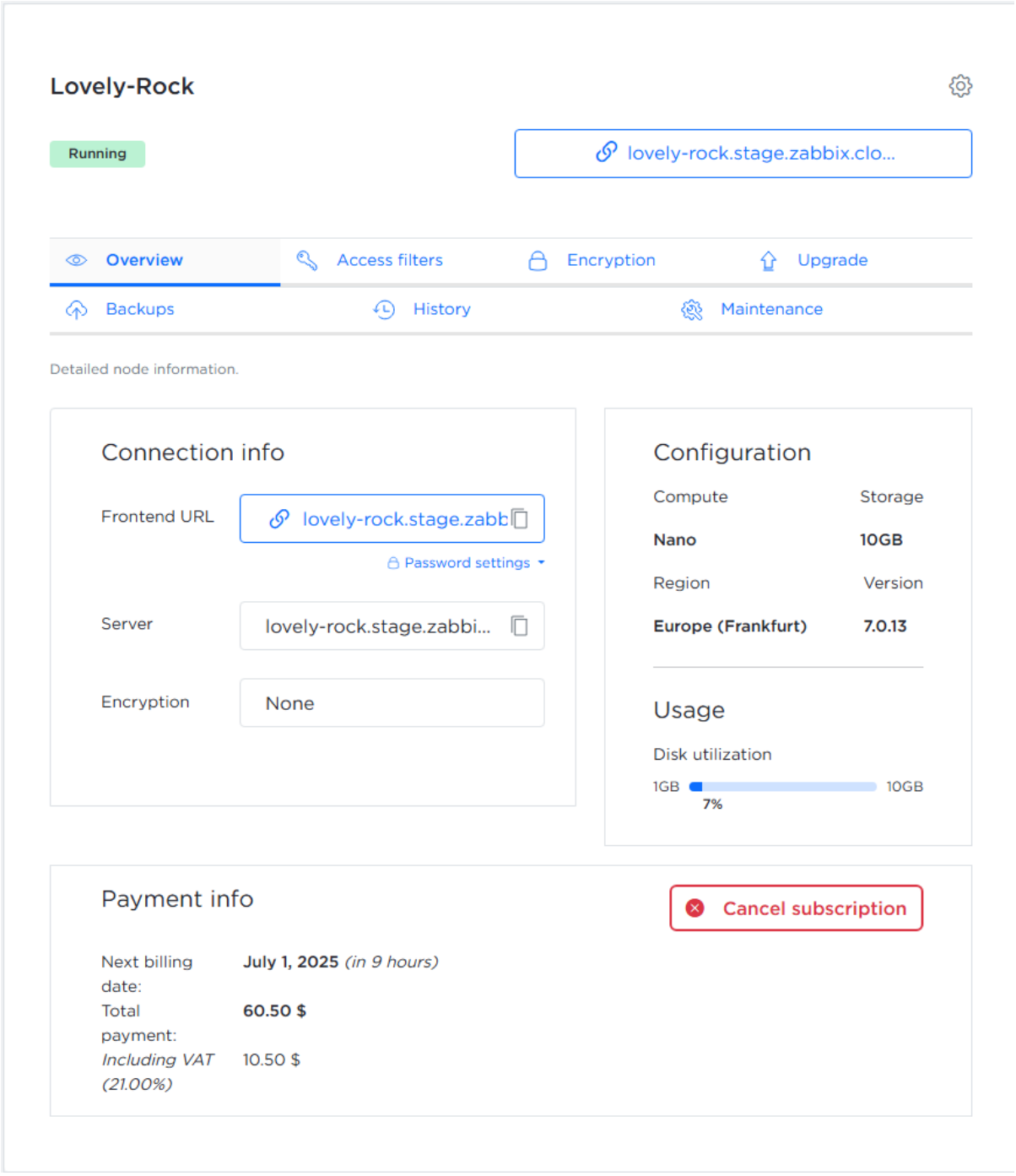
Ein **Cloud-Knoten** ist eine in der Cloud bereitgestellte Zabbix-Komponente. Server-Knoten umfassen den Zabbix-Server, die Datenbank und das Web-Frontend, während Proxy-Knoten den Zabbix-Proxy und die Datenbank umfassen. Ein Cloud-Knoten verfügt über dedizierte Rechenressourcen (CPU, Arbeitsspeicher) und Festplattenspeicher.

Benutzer können einen oder mehrere Cloud-Knoten erstellen. Cloud-Knoten können zu **Organisationen** (strukturell, regional usw.) gruppiert werden, wodurch eine verteilte Überwachungsumgebung entsteht.

Cloud-Knoten und Organisationen werden über die Benutzeroberfläche von Zabbix Cloud erstellt. Um die Konfigurationsseite eines Knotens aufzurufen, klicken Sie auf den Knoten unter *Alle Organisationen*, in Ihrem Organisationsmenü oder auf den Link *Konfigurationsseite des Knotens*, der nach dem Starten des Knotens angezeigt wird.

In der Konfigurationsansicht des Knotens können Sie auf jeder Registerkarte einen bestimmten Aspekt Ihres Knotens verwalten:

- Übersicht - Knotendetails anzeigen, das Frontend öffnen und Knotenbenutzer verwalten.
- Zugriffsfilter - Zugriff nach IP-Adresse oder Subnetz steuern.
- Verschlüsselung - TLS durch Hochladen von Zertifikaten, Schlüsseln und optionalen CRLs konfigurieren.
- Backups - Knotensnapshots erstellen, wiederherstellen und löschen; das nächste geplante Backup anzeigen.
- Verlauf - Aufbewahrungszeiträume für Auditprotokolle, Verlauf und Trends mit voreingestellten Einheiten und einem Aktualisierungslimit von 3-mal täglich festlegen.
- Wartung - Wöchentliche Wartungsfenster nach Zeitzone, Tag und Stunde definieren.
- Upgrade - Rechen- und Speicherressourcen skalieren oder den Status Ihrer Testphase/Zahlung verwalten.

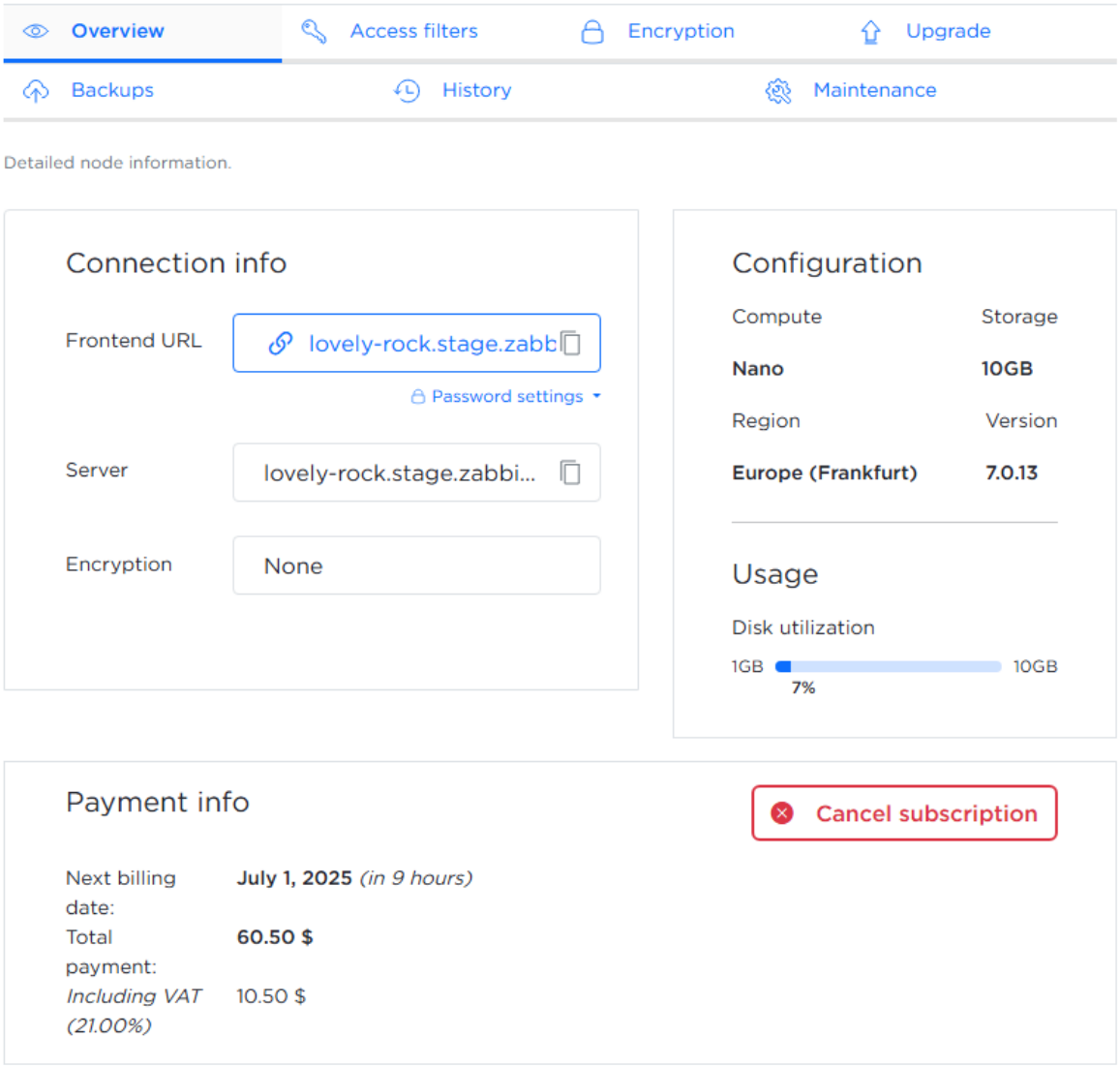


Note:
Einige Registerkarten sind möglicherweise abhängig vom Knotentyp (Server oder Proxy), vom Benutzertyp, der die Seite aufruft, und vom Knotenstatus (Testphase/keine Testphase) nicht verfügbar.

Registerkarte „Übersicht“

Die Registerkarte *Übersicht* bietet detaillierte Informationen zum Knoten. Die angezeigten Informationen hängen davon ab, ob es sich bei dem Knoten um einen Server oder einen Proxy handelt.

Serverknoten Die Registerkarte *Übersicht* für den Serverknoten enthält die folgenden Abschnitte:



Verbindungsinformationen:

- Anklickbare Frontend-URL mit Option zum Kopieren in die Zwischenablage
- Benutzername
- Passwort des Knotens und dessen Einstellungen
- Server-Hostname mit Option zum Kopieren in die Zwischenablage
- IPv4-Adresse mit Option zum Kopieren in die Zwischenablage
- IPv6-Adresse mit Option zum Kopieren in die Zwischenablage
- Aktuelle Verschlüsselungsmethode

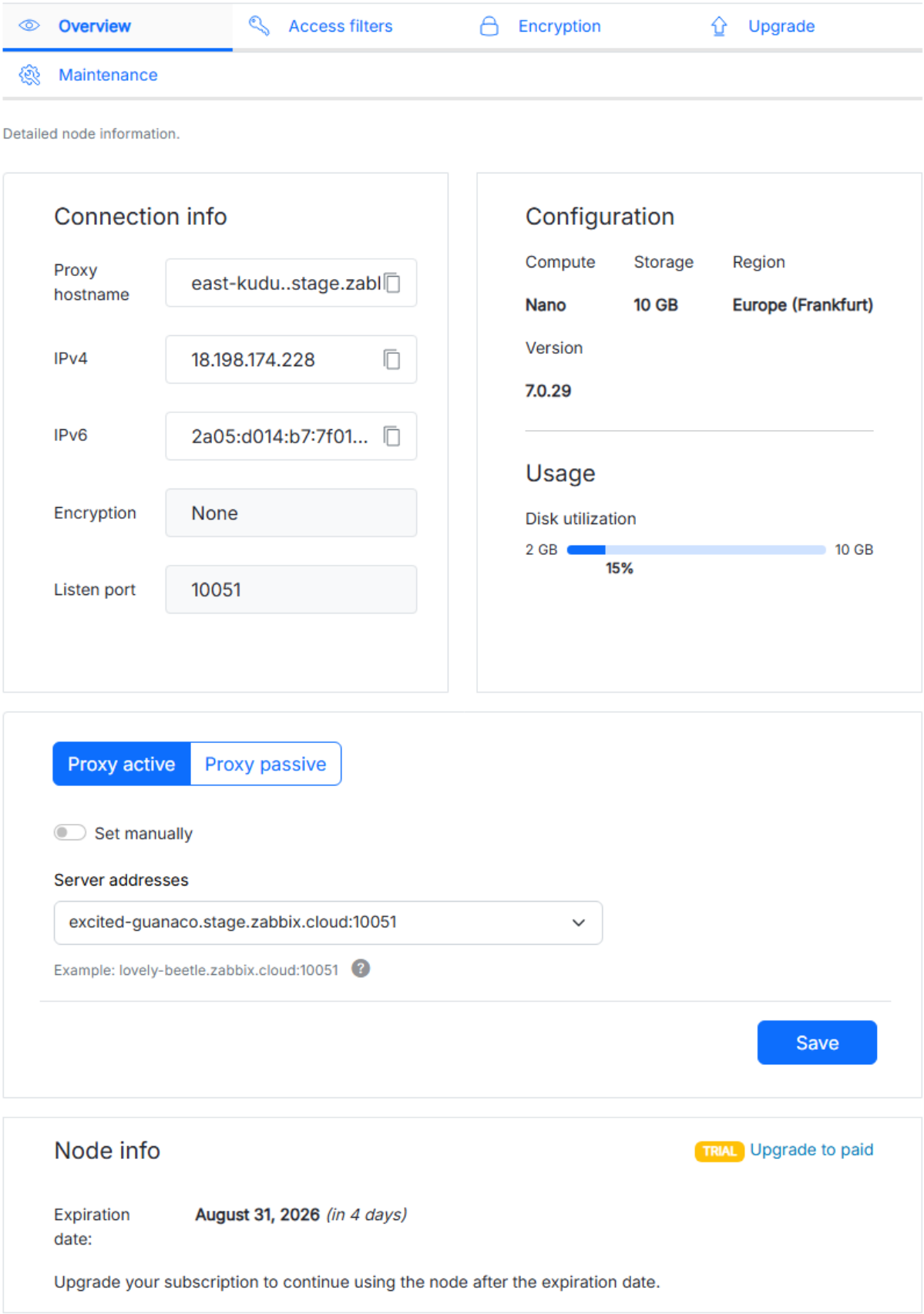
Konfiguration und Nutzung:

- Compute-Tier, maximaler Speicher, Region
- Version der Zabbix-Komponenten
- Festplattenauslastung

Knoteninformationen/Zahlungsinformationen:

- Ablaufdatum (für Test- und Prepaid-Knoten)
- Nach Abschluss der Initialisierung des Knotens wird abhängig vom Knotentyp eine kontextspezifische Schaltfläche angezeigt: - *Auf kostenpflichtigen Tarif upgraden* - verfügbar für Knoten in einer kostenlosen Testphase - *Abonnement kündigen* - verfügbar für kostenpflichtige Knoten mit einem aktiven Abonnement - *Erneut kostenpflichtig abonnieren* - wird angezeigt, wenn ein Abonnement gekündigt wurde - Für Prepaid-Knoten wird keine Aktionsschaltfläche angezeigt, da diese eine lange Gültigkeitsdauer haben und nicht gekündigt werden können

Proxy-Knoten Die Registerkarte *Übersicht* für den Proxy-Knoten enthält die folgenden Abschnitte:



Verbindungsinformationen:

- Hostname des Proxy mit einer Option zum Kopieren in die Zwischenablage
- IPv4-Adresse mit einer Option zum Kopieren in die Zwischenablage
- IPv6-Adresse mit einer Option zum Kopieren in die Zwischenablage
- Aktuelle Verschlüsselungsmethode
- Aktueller Überwachungsport

Konfiguration und Nutzung:

- Compute-Tier, maximaler Speicher, Region
- Version der Zabbix-Komponenten
- Festplattenauslastung

Proxy-Modus:

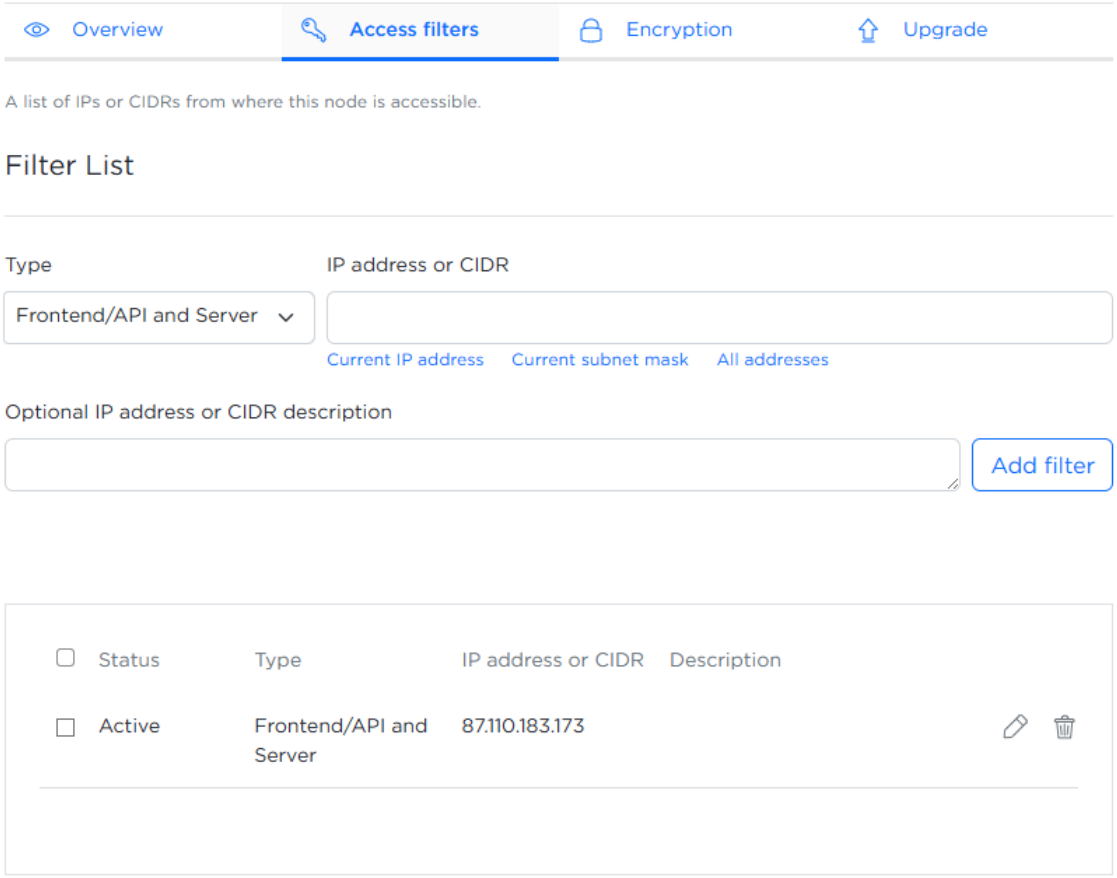
- Zwischen aktivem und passivem Proxy-Modus wechseln
- Die Serveradresse aus dem Dropdown-Menü *Serveradressen* auswählen oder *Manuell festlegen* aktivieren, um die Serveradresse manuell einzugeben

Knoteninformationen/Zahlungsinformationen:

- Ablaufdatum (für Test- und Prepaid-Knoten)
- Nach Abschluss der Initialisierung des Knotens wird abhängig vom Knotentyp eine kontextspezifische Schaltfläche angezeigt: - *Auf kostenpflichtig upgraden* – verfügbar für Knoten in einer kostenlosen Testphase - *Abonnement kündigen* – verfügbar für kostenpflichtige Knoten mit einem aktiven Abonnement - *Erneut kostenpflichtig abonnieren* – wird angezeigt, wenn ein Abonnement gekündigt wurde - Für Prepaid-Knoten wird keine Aktionsschaltfläche angezeigt, da diese eine lange Gültigkeitsdauer haben und nicht gekündigt werden können

Registerkarte „Zugriffsfilter“

Auf der Registerkarte *Zugriffsfilter* können Sie den Zugriff auf Ihren Zabbix-Knoten anhand einer IP-Adresse oder eines Subnetzes einschränken.



Um einen Zugriffsfilter zu konfigurieren, füllen Sie die folgenden Felder aus:

- *Typ* – Wählen Sie bei Serverknoten über die Dropdown-Liste aus, welche Zabbix-Komponente eingeschränkt werden soll: nur Frontend/API, nur Server oder beide. Bei Proxy-Knoten ist nur *Eingehende IP* verfügbar. Damit wird festgelegt, welche eingehenden Netzwerkanforderungen den Proxy erreichen dürfen.
- *IP-Adresse oder CIDR* – Geben Sie die IP-Adresse oder CIDR manuell ein oder füllen Sie das Feld automatisch aus, indem Sie die entsprechende Option auswählen.
- *Optionale Beschreibung der IP-Adresse oder des CIDR* – Fügen Sie für jede IP-Adresse oder jedes Subnetz eine optionale Beschreibung hinzu.

Nachdem Sie die Felder ausgefüllt haben, klicken Sie auf *Filter hinzufügen*. Hinzugefügte Filter werden in der Tabelle darunter angezeigt, einschließlich Status, Typ, IP-Adresse oder Subnetz und Beschreibung.

Warning:
Während der Initialisierung des Knotens wird automatisch ein Zugriffsfilter für die von Ihnen verwendete IP-Adresse erstellt. Wenn Sie später eine Verbindung von einer anderen IP-Adresse oder einem anderen Netzwerk herstellen, müssen Sie diese Adresse hier hinzufügen, andernfalls wird der Zugriff blockiert.

Sie können:

-  – einen Filter bearbeiten.
-  – einen Filter oder mehrere Filter mithilfe der Kontrollkästchen löschen.

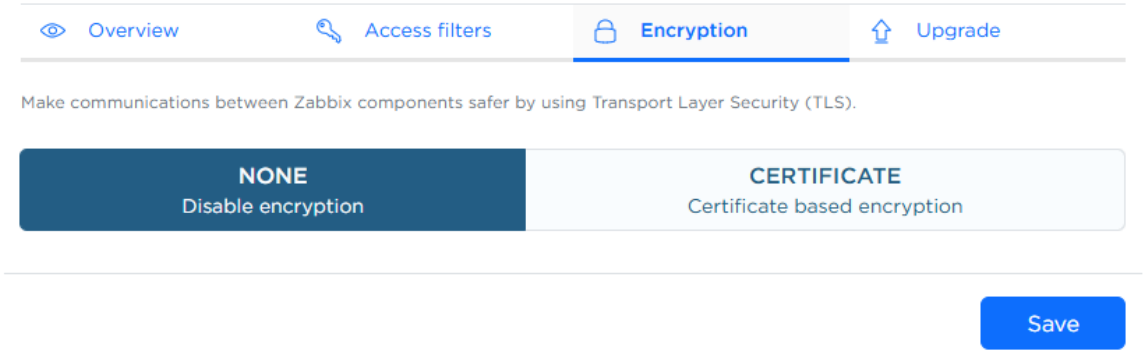
Registerkarte „Verschlüsselung“

Auf der Registerkarte *Verschlüsselung* können Sie die TLS-Verschlüsselung für die Kommunikation zwischen Zabbix-Komponenten konfigurieren, indem Sie die erforderlichen Zertifikatsdateien hochladen.

Um eine sichere Kommunikation mithilfe von Zertifikaten zu aktivieren, wählen Sie *CERTIFICATE* aus und laden Sie die folgenden Dateien von Ihrem Computer hoch:

- *Stammzertifikate der CA* – vertrauenswürdige CA-Zertifikate, die zur Überprüfung von Peer-Zertifikaten verwendet werden.
  Klicken Sie auf *Datei auswählen* und wählen Sie die Datei mit den vertrauenswürdigen Stammzertifikaten der CA aus (typischerweise eine `.crt`- oder `.pem`-Datei im PEM-Format).
- *Zertifikatskette* – stellt die Vertrauenskette für Ihr eigenes Zertifikat dar.
  Klicken Sie auf *Datei auswählen* und wählen Sie die PEM-codierte Datei (z. B. `.crt` oder `.pem`) aus, die Ihr Zertifikat und alle Zwischenzertifikate der CA enthält.
- *Privater Schlüssel* – der private Schlüssel, der Ihrem Zertifikat entspricht.
  Klicken Sie auf *Datei auswählen* und wählen Sie die PEM-codierte `.key`-Datei aus.
- *Widerrufene Zertifikate* (optional) – wenn Sie eine Zertifikatsperrliste (CRL) verwalten, klicken Sie auf *Datei auswählen* und wählen Sie die CRL-Datei aus, um Zertifikate anzugeben, denen nicht mehr vertraut werden soll.

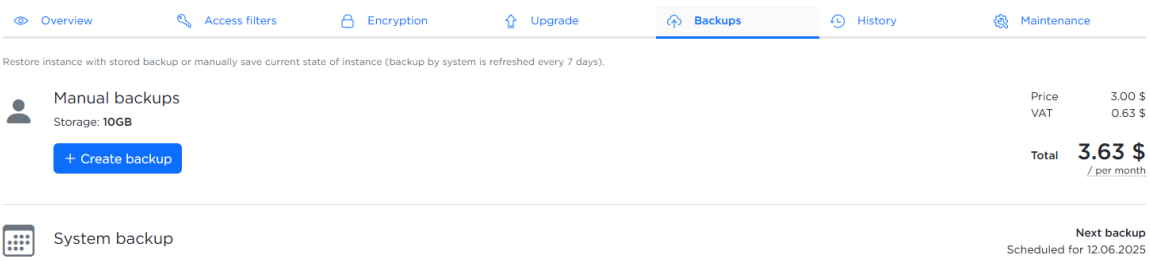
Nachdem Sie die Dateien ausgewählt haben, klicken Sie auf *Speichern*.
Die Dateien werden hochgeladen, validiert und angewendet, um die TLS-Verschlüsselung zwischen den Zabbix-Komponenten zu konfigurieren.



Registerkarte „Backups“

Auf der Registerkarte *Backups* können Sie Backups Ihres Zabbix-Knotens anzeigen, erstellen, wiederherstellen und löschen.

Sie können Ihre Instanz jederzeit aus einem gespeicherten Backup wiederherstellen oder ein neues manuelles Backup erstellen. Die Registerkarte zeigt außerdem das Datum des nächsten geplanten Backups an.



So erstellen Sie ein Backup:

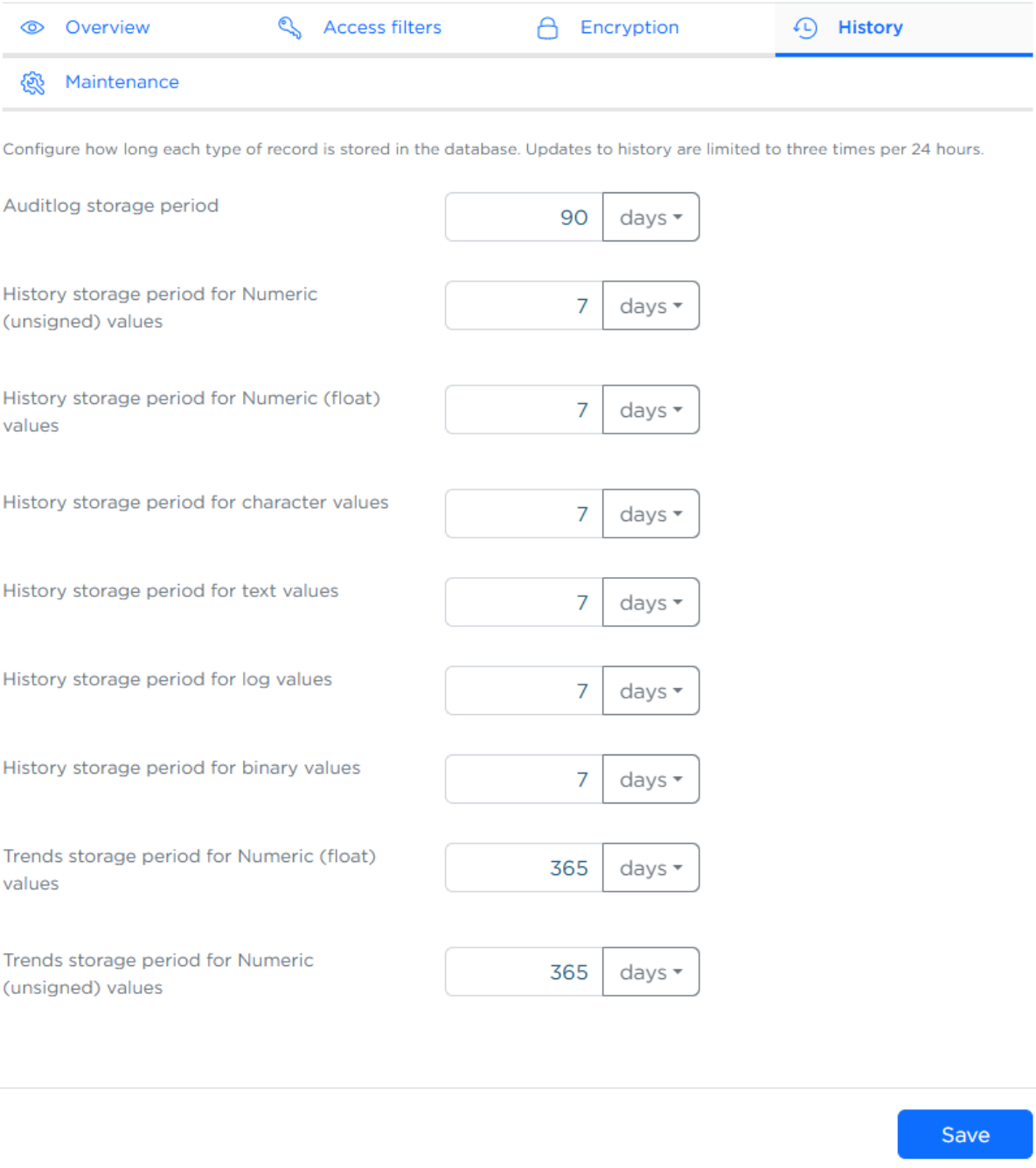
1. Klicken Sie auf die Schaltfläche *Backup erstellen*.
2. Geben Sie im Popup-Fenster optional eine Beschreibung ein.

3. Klicken Sie zur Bestätigung auf *Erstellen*.

Warning:
Zusätzliche Backups werden monatlich auf Grundlage der Speichernutzung berechnet.

Registerkarte „Historie“

Auf der Registerkarte *Historie* können Sie konfigurieren, wie lange verschiedene Arten historischer Daten gespeichert werden.



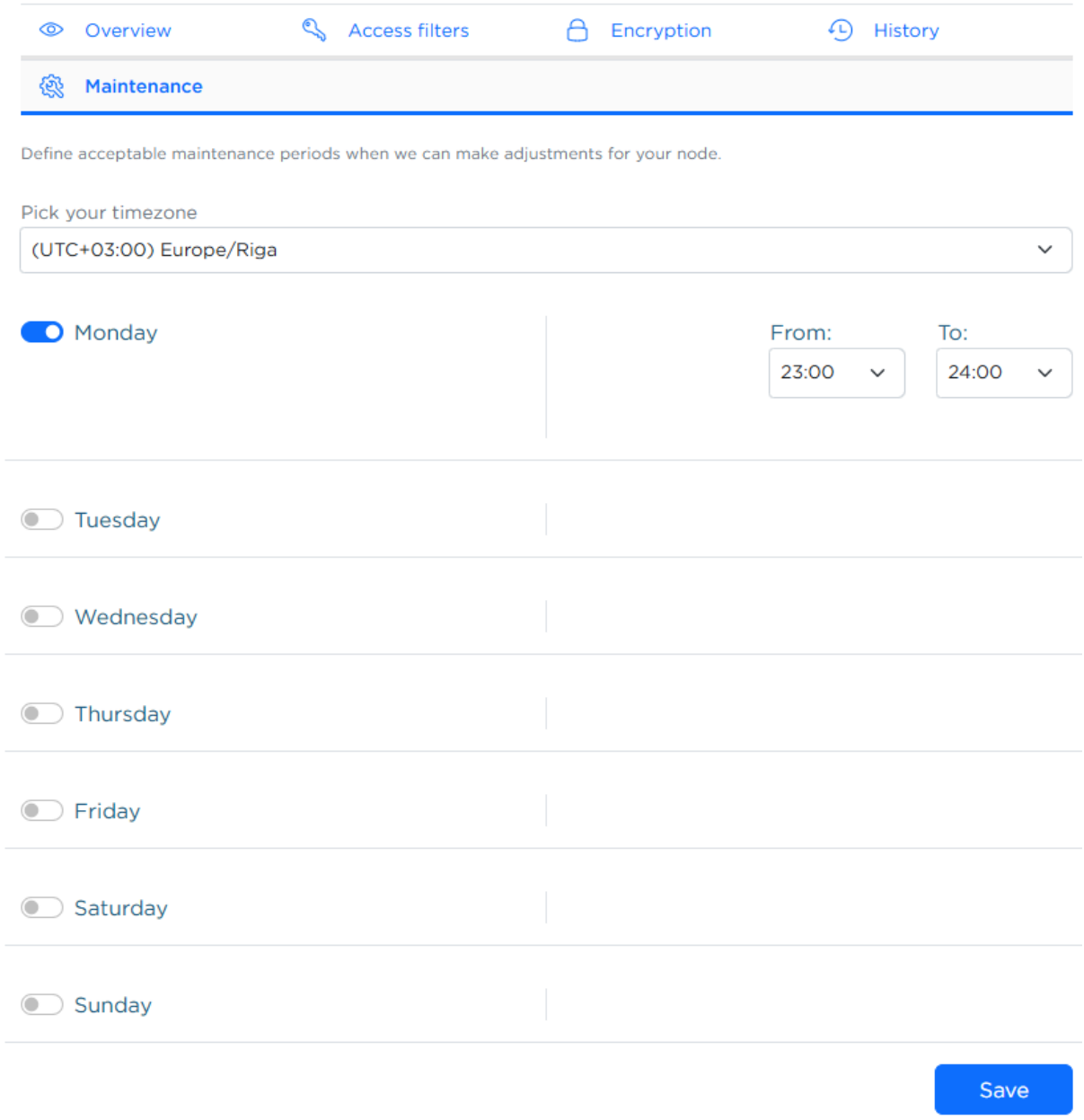
Sie können den Aufbewahrungszeitraum für jeden Datentyp (z. B. numerische Daten, Text und Protokolle) festlegen und die Einheit (Tage, Monate, Jahre) auswählen. Änderungen können bis zu dreimal innerhalb von 24 Stunden vorgenommen werden.

Nachdem Sie die Werte aktualisiert haben, klicken Sie auf *Speichern*, um die Änderungen anzuwenden.

Warning:
Längere Aufbewahrungszeiträume können den Speicherplatzbedarf erhöhen und die Leistung beeinträchtigen. Es wird empfohlen, die Aufbewahrungszeiträume entsprechend Ihren tatsächlichen Daten- und Analyseanforderungen zu konfigurieren.
 Zabbix Cloud verwendet die Datenbankpartitionierung zur Verwaltung der Datenspeicherung anstelle der herkömmlichen Zabbix-Housekeeping-Funktion. Dies ermöglicht eine schnellere Leistung und eine besser vorhersehbare Verwaltung der Aufbewahrungszeiträume.

Registerkarte „Wartung“

Auf der Registerkarte *Wartung* können Sie zulässige Wartungsfenster festlegen, in denen Aktualisierungen oder Anpassungen an Ihrem Knoten durchgeführt werden können.

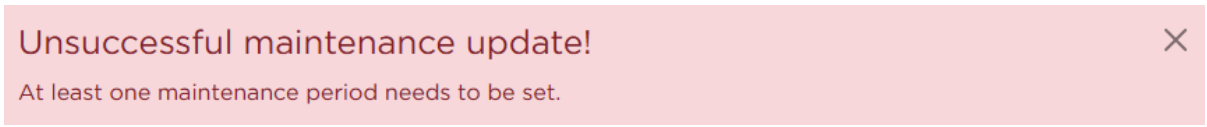


Sie müssen:

- Ihre Zeitzone aus der Dropdown-Liste auswählen.
- Mindestens einen Tag und ein stündliches Wartungsfenster auswählen.

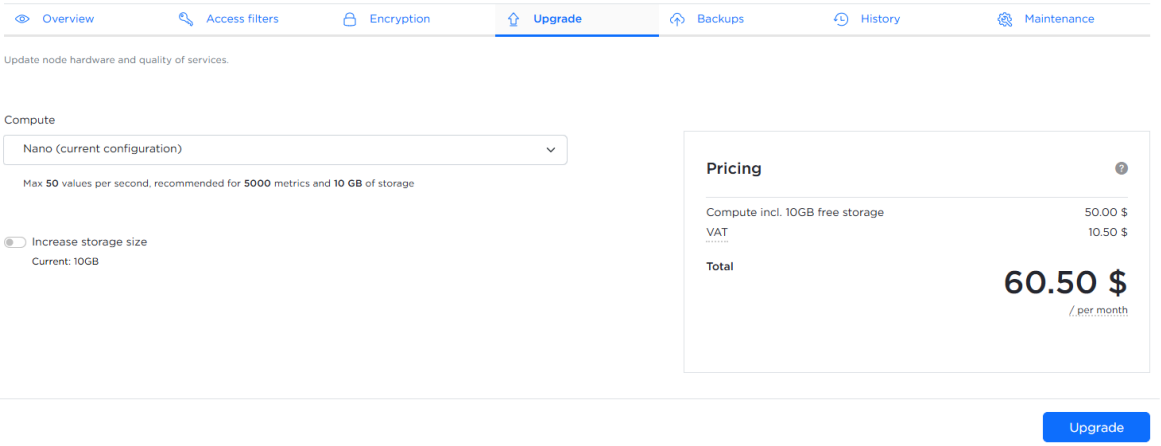
Klicken Sie auf *Speichern*, um die Änderungen anzuwenden.

Wenn kein Tag oder kein Zeitfenster ausgewählt ist, wird ein Fehler angezeigt.



Registerkarte „Upgrade“

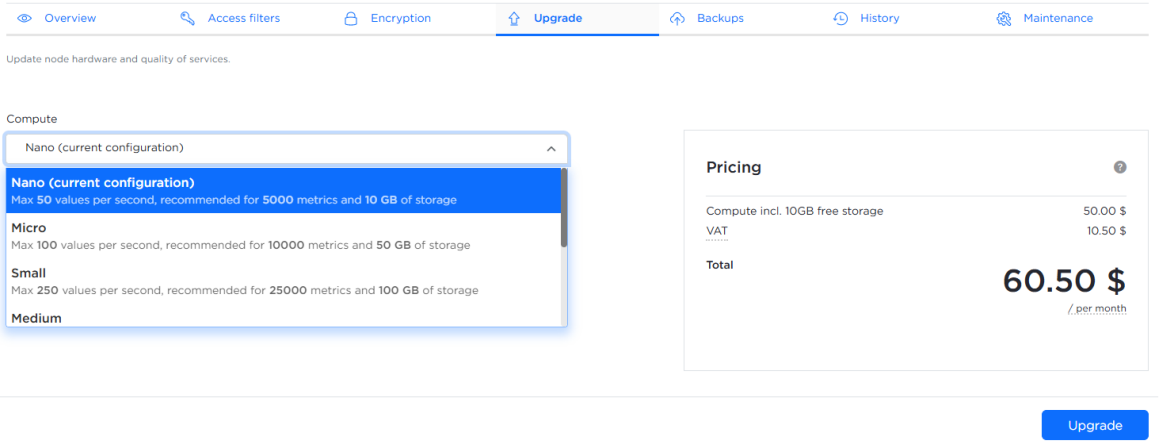
Auf der Registerkarte *Upgrade* können Sie die Ressourcen Ihres Knotens skalieren oder sein Abonnement verwalten.



Sie können die Compute-Stufe oder die Festplattengröße erhöhen. Beachten Sie, dass die Festplattengröße nur einmal innerhalb von 24 Stunden erhöht werden kann und über die Benutzeroberfläche nicht reduziert werden kann (wenden Sie sich für Anfragen zur Verkleinerung an den Zabbix-Support). Knoten mit einer kostenlosen Testversion können in kostenpflichtige Abonnements umgewandelt werden.

So erhöhen Sie die Compute-Stufe:

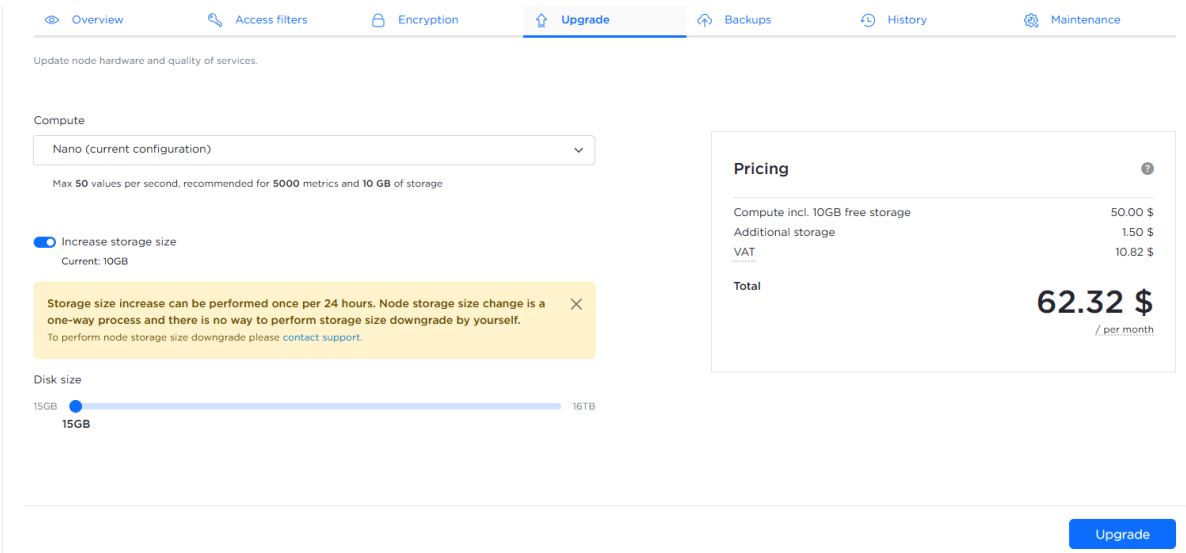
1. Klicken Sie auf das Dropdown-Menü *Compute*.



2. Wählen Sie die gewünschte Konfigurationsstufe aus.
3. Klicken Sie auf die Schaltfläche *Upgrade*.

So erhöhen Sie die Festplattengröße:

1. Klicken Sie auf das Optionsfeld *Increase storage size*.
2. Passen Sie den Schieberegler *Disk size* auf den gewünschten Wert an.



3. Klicken Sie auf *Upgrade*.

Note:
Der aktualisierte monatliche Gesamtpreis wird vor der Bestätigung der Änderungen in der Tabelle *Pricing* angezeigt.

Benutzer hinzufügen

Einführung

Die folgenden Beispiele beschreiben, wie Sie Benutzer zu Ihrer Organisation und zu Knoten hinzufügen.

Note:
Das Kopieren, Löschen und Zurücksetzen des Frontend-Passworts kann nur vom Eigentümer durchgeführt werden.

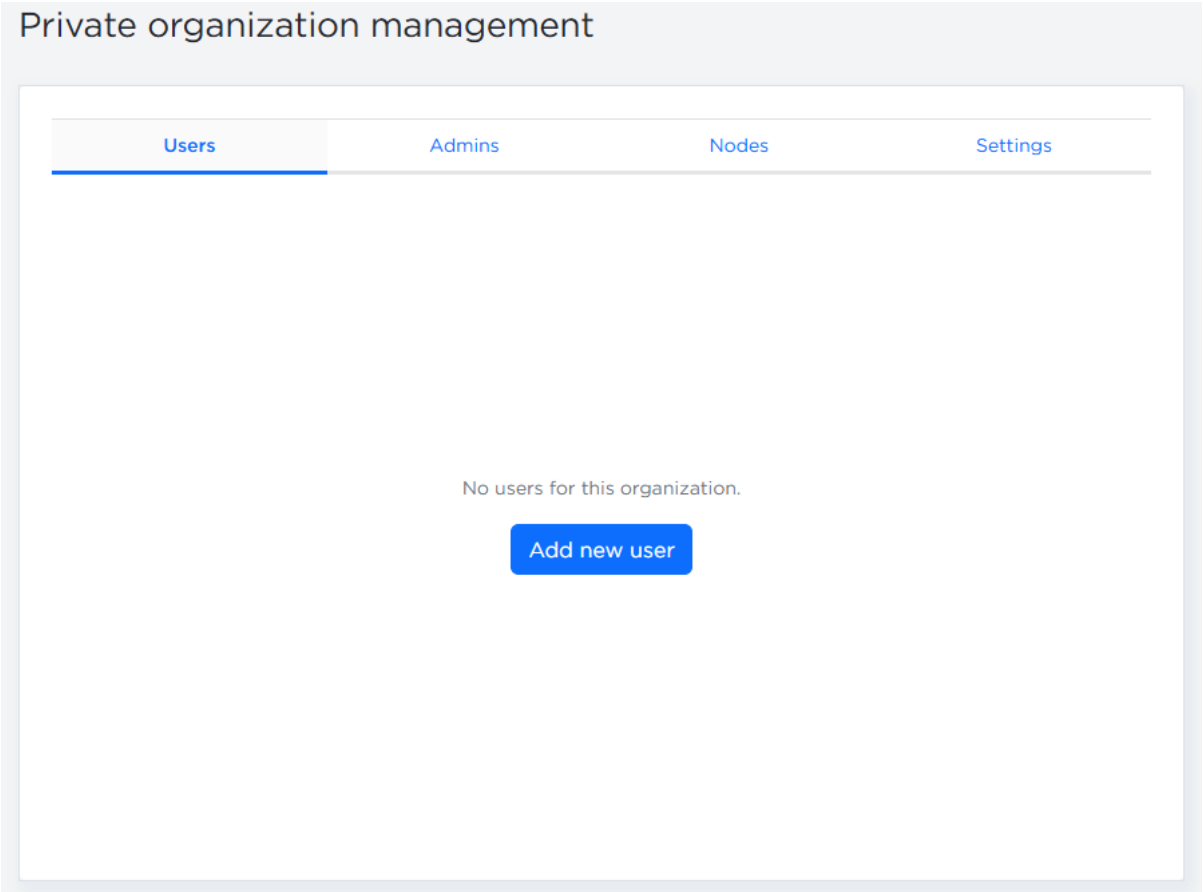
Benutzer zu Organisationen hinzufügen

Ähnlich wie bei Benutzern in Zabbix können Sie in Zabbix Cloud Benutzer erstellen und ihnen Rollen zuweisen, um zu steuern, was sie innerhalb einer Organisation und ihrer Nodes tun können.

So fügen Sie Ihrer Organisation Benutzer hinzu:

1. Öffnen Sie die Seite *All organizations*.

2. Klicken Sie auf die Schaltfläche  Einstellungen der jeweiligen Organisation, um deren Verwaltungsmenü zu öffnen.



3. Klicken Sie auf der Registerkarte *Users* oder *Admins* auf *Add new user*.

4. Geben Sie die E-Mail-Adresse des Benutzers ein und wählen Sie eine Rolle aus: *Administrator* oder *User*.

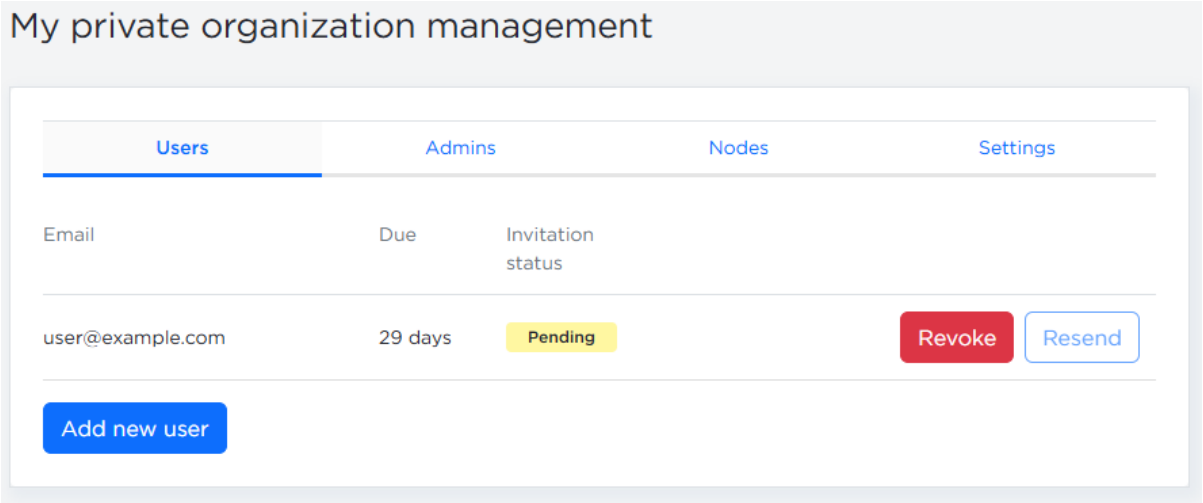
Beachten Sie, dass die Auswahl der Rolle *User* den Benutzer zur Organisation hinzufügt, ihm jedoch keinen Zugriff auf Nodes gewährt.

Jede Rolle gewährt eine andere Zugriffsstufe:

- **User** – kann auf die ihm zugewiesenen Nodes zugreifen und Konfigurationsaufgaben ausführen, die keine Kosten verursachen, zum Beispiel:
  - Zugriffsfilter verwalten
  - Verschlüsselung konfigurieren
  - Aufbewahrungszeiträume für Verlauf/Trends festlegen
  - Wartung planen
- **Administrator** – verfügt über alle Berechtigungen eines Users und kann außerdem Aktionen ausführen, die Kosten verursachen können, zum Beispiel:
  - Backups erstellen und wiederherstellen
  - Node-Ressourcen (Compute-Tier und Festplattengröße) hoch- oder herabstufen
  - Abonnements kündigen
- **Owner** – hat die vollständige Kontrolle über die Organisation. Zusätzlich zu allen Berechtigungen eines Administrators kann der Owner:
  - Nodes erstellen und löschen
  - Users und Administrators hinzufügen oder entfernen
  - Die Organisation umbenennen oder löschen
  - Rechnungsdaten verwalten

> **Attention:**
> Die Zahlungsmethode des Owners wird für alle Dienste belastet. Wenn eine Bankbestätigung erforderlich ist, muss der Owner die Transaktion autorisieren, auch wenn sie von einem Administrator initiiert wurde.

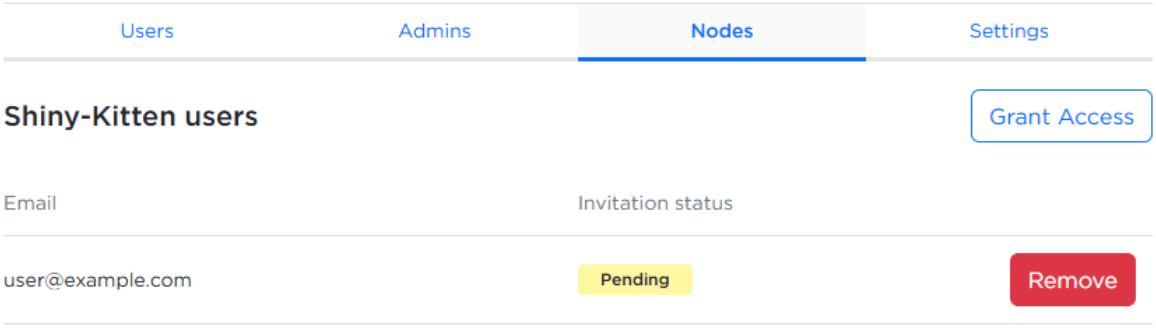
Der eingeladene Benutzer wird mit dem Status *Pending* hinzugefügt.
Sie können die Einladung widerrufen oder erneut senden (ein erneutes Senden ist nur einmal alle 10 Minuten möglich – die Schaltfläche wird deaktiviert, wenn dieses Limit erreicht ist).



Benutzer zu Nodes hinzufügen

So fügen Sie den Benutzer zum Node hinzu:

1. Wechseln Sie zur Registerkarte *Nodes*.
2. Klicken Sie auf *Grant Access*.
3. Wählen Sie einen zur Organisation hinzugefügten Benutzer aus der Dropdown-Liste aus.
4. Klicken Sie auf *Grant access*.



Alternativ können Sie den Zugriff im Fenster *Node configuration* gewähren:

1. Klicken Sie auf die Schaltfläche  Einstellungen.
2. Wählen Sie *Add user*.
3. Wählen Sie einen Benutzer aus und klicken Sie auf *Grant access*.

Eingeladene Benutzer müssen die Einladung annehmen, indem sie mit derselben E-Mail-Adresse, an die die Einladung gesendet wurde, ein Zabbix Cloud-Konto erstellen. Wenn der Benutzer unter dieser E-Mail-Adresse bereits ein Zabbix Cloud-Konto hat, muss er sich anmelden und die Einladung bestätigen. Einladungen sind an eine bestimmte E-Mail-Adresse gebunden und können nicht von einem anderen Konto aus angenommen werden.

Hauptunterschiede von Zabbix Cloud

Einführung

Zabbix Cloud ist eine cloudbasierte Plattform, die es ihren Benutzern ermöglicht, eigenständige Zabbix-Instanzen in der Cloud (Cloud-Knoten) auszuführen.

Im Vergleich zu einer regulären lokalen Zabbix-Bereitstellung gilt:

- Es sind keine Installation oder Wartung erforderlich.
- Cloud-Knoten werden vollständig von Zabbix verwaltet.
- Cloud-Knoten können innerhalb weniger Minuten gestartet werden.

Dieser Dienst wird auf Pay-as-you-go-Basis angeboten (eine **kostenlose Testversion** ist verfügbar).

Vergleichstabelle Obwohl sich die meisten Funktionen zwischen lokalen Zabbix-Installationen und Cloud-Knoten überschneiden, gibt es auch Unterschiede.
Eine zusammenfassende Vergleichstabelle finden Sie unten.
In den folgenden Abschnitten werden ausgewählte Punkte ausführlich beschrieben.

Funktion / Aspekt	Zabbix Cloud	Zabbix lokal
Release-Zyklus	Im Einklang mit den LTS-Releases	Sie wählen Ihre Zabbix-Version (stabile Version, Beta, benutzerdefinierte Builds)
Benutzerfreundlichkeit	Einsatzbereite Plattform, vollständig von Zabbix verwaltet	Erfordert Installation, Konfiguration und Wartung
Behebung von Sicherheitslücken	Automatische Betriebssystem-/Zabbix-Patches	Manuelles Patchen durch den Benutzer
HTTPS-Zertifikat	Gültiges Zertifikat standardmäßig vorhanden (kein Let's Encrypt)	Standardmäßig selbstsigniert (Let's Encrypt oder eine andere Zertifizierungsstelle kann konfiguriert werden)
Datenbankleistung (INSERT/UPDATE/SELECT)	Automatische Partitionierung für maximale Geschwindigkeit; keine lang laufenden DELETEs	Manuelle Partitionierung und Bereinigung (lange DELETE-Transaktionen möglich)
Leistungsoptimierung	Vom Zabbix-Cloud-Team vorkonfiguriert (keine Änderungsprotokolle)	Vollständige Transparenz und Kontrolle über Optimierungsparameter

Funktion / Aspekt	Zabbix Cloud	Zabbix lokal
Firewall-Steuerung	Vom Zabbix-Cloud-Team verwaltet; GUI-/Trapper-Ports sind nicht öffentlich zugänglich	Vollständige Kontrolle über Firewall-Regeln und die Freigabe von Ports
Ausgehender Datenverkehr über Port 25	Standardmäßig blockiert, um Spam und Missbrauch zu verhindern; verwenden Sie stattdessen den SMTP-Port 587 oder 465	Von Zabbix nicht eingeschränkt; abhängig von Ihrem Netzwerk und Hosting-Anbieter
Datenbankisolierung und -schutz	Stark isoliert, kein direkter Datenbank-/SSH-Zugriff	Abhängig von Ihrem Netzwerk und der Sicherheit Ihres Hosts
Hochverfügbarkeit (HA)	Wird auf einer hochverfügbaren Cloud-Plattform ausgeführt, Zabbix-Proxys können für eine höhere Ausfallsicherheit bei der Datenerfassung verwendet werden	Vom Benutzer verwaltete HA
SNMP-Traps	Nur über einen dedizierten Proxy (kein HA-/lastverteilter Trap-Empfänger)	Direkt oder über einen Proxy, HA/Lastverteilung möglich
SNMP-Abfragen	Ohne Proxys schwierig (jedes Gerät benötigt NAT/einen benutzerdefinierten Port)	Native SNMP-Abfragen; Proxy optional
Aufbewahrungseinstellungen	Historie/Trends/Audit nur über die Benutzeroberfläche (keine API oder Konfigurationsdatei)	Konfigurierbar in der Server-Konfigurationsdatei von Zabbix oder über die API
Benutzerdefinierte Skripte	`AlertScriptsPath`, `ExternalScripts`, Frontend- und Community-Module werden nicht unterstützt	Vollständig unterstützt (Pfad für Skripte, Module, Integrationen)
ODBC-Überwachung	Nur PostgreSQL (Treiber `{postgresql}`); MariaDB-Plugin als Platzhalter vorhanden	ODBC für PostgreSQL, MySQL, Oracle usw. - vollständig konfigurierbar
Begrenzung von ODBC-Aufrufen	Nicht möglich (nur `StartODBCPollers=1`); umfangreiche Synchronisierungsabfragen blockieren andere Aufgaben)	Anzahl und Zeitplanung der Poller vollständig anpassbar
Upload von SAML-Zertifikaten	Nicht unterstützt	Über die Benutzeroberfläche oder API unterstützt
Geplante Berichte	Sie müssen ein eigenes E-Mail-Medium erstellen (Skriptmedien werden nicht unterstützt)	Unterstützt standardmäßig sowohl Skript- als auch E-Mail-Medien
Host-Schnittstelle für aktive Prüfungen	Von der Plattform bereitgestellte IP-Adresse (automatisch erstellt)	Schnittstellen werden von Ihnen verwaltet; IP-Adressen liegen unter Ihrer Kontrolle

Funktionsunterschiede SNMP-Traps

SNMP-Traps werden nur über einen dedizierten Zabbix Proxy unterstützt. Wenn die Überwachung von SNMP-Traps erforderlich ist, können automatischer Lastausgleich oder Hochverfügbarkeit für den Proxy nicht verwendet werden, da SNMP-Traps an eine einzelne IP-Adresse gesendet werden müssen.

SNMP-Abfrage

Ohne Proxys erfordert die SNMP-Abfrage, dass jedes Gerät über NAT und benutzerdefinierte Ports erreichbar ist.

Aufbewahrungseinstellungen

Die Aufbewahrungszeiträume für Verlauf, Trends und Audit-Log können in der Cloud nur über die Weboberfläche konfiguriert werden. Diese Einstellungen können nicht über `zabbix_server.conf` oder die API konfiguriert werden.

Manuelle Überschreibungen des Verlaufs pro Datenpunkt werden nicht unterstützt (die Partitionierung wird global gesteuert). Cloud-Node-URLs akzeptieren nicht dieselben Abfrageparameter wie On-Premise-Installationen.

Benutzerdefinierte Skripte

Die folgenden Typen benutzerdefinierter Skripte werden von Zabbix Cloud nicht unterstützt:

- Warnskripte (`AlertScriptsPath`)
- Externe Skripte (`ExternalScripts`)
- Frontend-Skripte auf dem Zabbix Server

Von der Community entwickelte Frontend-Module können nicht installiert werden.

ODBC-Monitoring

Zabbix Cloud unterstützt ODBC-Monitoring nur für PostgreSQL. Verwenden Sie die offizielle Zabbix-ODBC-Vorlage und definieren Sie die Verbindungszeichenfolge in der Vorlage mit:

```
Driver={postgresql}
```

Ein MariaDB-Plugin ist ebenfalls installiert, funktioniert derzeit jedoch nicht. Falls verwendet, definieren Sie:

```
Driver={mysql}
```

Attention:
Es gibt keine bekannte funktionierende Konfiguration für Oracle MySQL 8.0 in Zabbix Cloud. Während einfache Abfragen (wie `SELECT 1`) erfolgreich sein können, geben komplexere Abfragen `SQL_ERROR` zurück.

 Es ist nicht möglich, ODBC-Aufrufe zu begrenzen (`StartODBCPollers=1` only). Umfangreiche synchrone Berichterstellung kann die Leistung beeinträchtigen, und es kann jeweils nur ein einzelnes `SELECT` gleichzeitig ausgeführt werden.

Zugriff auf die Infrastruktur

Zabbix Cloud bietet keinen SSH-Zugriff auf die zugrunde liegenden Knoten und erlaubt auch keine direkten Datenbankverbindungen (zum Beispiel zu Port 3306). Alle Konfigurations-, Überwachungs- und Fehlerbehebungsaktivitäten müssen über die Cloud-Benutzeroberfläche oder API durchgeführt werden, sodass die Betriebssystem- und Datenbankebenen isoliert und abgesichert bleiben.

Ausgehender Datenverkehr über Port 25

Um Spam und Missbrauch zu verhindern, sind ausgehende Verbindungen von Zabbix-Cloud-Knoten zu TCP-Port 25 standardmäßig blockiert. Dies betrifft:

- E-Mail-Medientypen, die über einen SMTP-Server an Port 25 senden. Da 25 der standardmäßige *SMTP-Serverport* für einen neuen E-Mail-Medientyp ist, ändern Sie ihn je nach Unterstützung Ihres SMTP-Servers in 587 (mit auf STARTTLS gesetzter *Verbindungssicherheit*) oder 465 (SSL/TLS). Der E-Mail-Anbieter *Office365 relay* kann nicht verwendet werden, da er eine Verbindung zum Relay-Endpunkt von Microsoft an Port 25 herstellt.
- Geplante Berichte, die einen E-Mail-Medientyp erfordern.
- Prüfungen, die der Cloud-Knoten selbst an Port 25 durchführt, z. B. `net.tcp.service[smtp]`-einfachen Prüfungen oder SMTP-Prüfungen in Regeln für die Netzwerkerkennung. Diese melden den SMTP-Dienst auch dann als nicht verfügbar, wenn er ausgeführt wird. Führen Sie solche Prüfungen stattdessen über einen Zabbix Proxy oder Zabbix Agent in Ihrem eigenen Netzwerk durch.

Der standardmäßige Medientyp *Cloud Email* ist nicht betroffen.

Schnittstelle für aktive Prüfungen

Active checks in der Cloud erstellen automatisch eine Host-Schnittstelle mit einer IP, die nichts mit Ihrem On-Premise-Netzwerk zu tun hat. Standardmäßig kann diese IP extern sein. Um Konsistenz zu gewährleisten, können Sie die Host-Schnittstelle nach der Erstellung manuell löschen oder anpassen.

Hochladen des SAML-Zertifikats

Die SAML-Authentifizierung wird nicht unterstützt, da nur wenige Anbieter unverschlüsselte oder unsignierte SAML-Nachrichten akzeptieren.

Geplante Berichte

Der standardmäßige Medientyp *Cloud Email* verwendet einen Skript-Transport und kann nicht für geplante Berichte verwendet werden. So senden Sie Berichte per E-Mail:

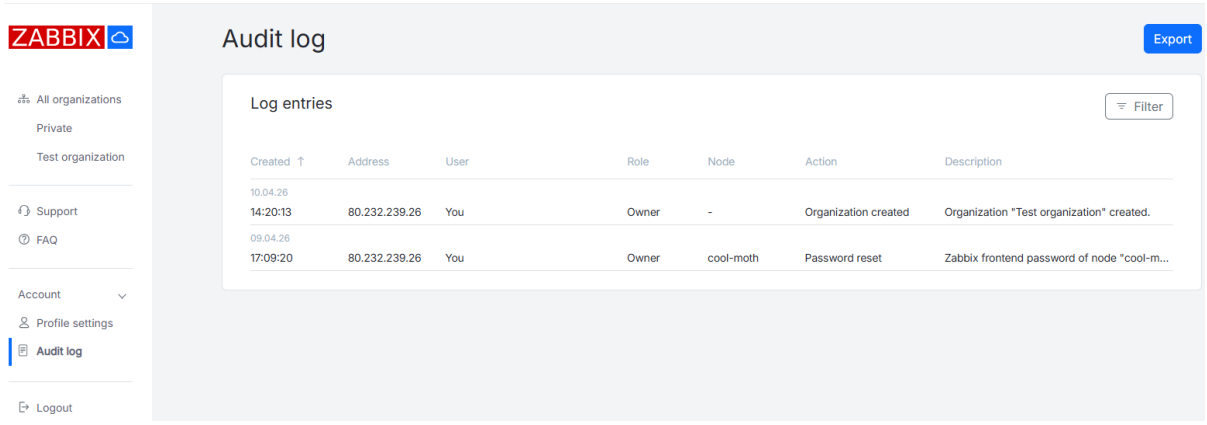
1. Erstellen Sie einen neuen E-Mail-Medientyp.
2. Weisen Sie Ihrem Benutzer unter *Benutzereinstellungen* > *Medien* das E-Mail-Medium zu.
3. Wählen Sie diesen Medientyp bei der Konfiguration geplanter Berichte aus.

Attention:
Verwenden Sie für diesen Medientyp nicht Port 25 als *SMTP-Serverport*; siehe Ausgehender Datenverkehr über Port 25.

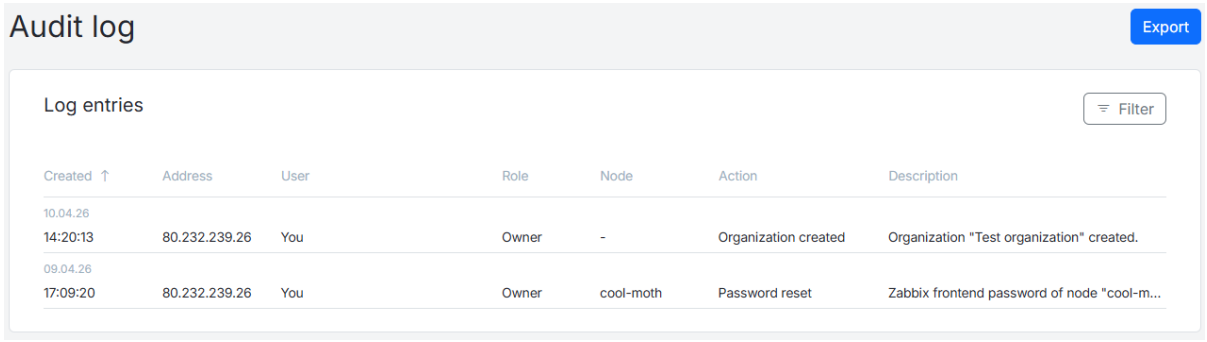
Auditprotokoll

Übersicht In Zabbix Cloud zeichnet das Auditprotokoll Benutzer- und Systemaktivitäten auf.

Nach dem Anmelden bei einem Zabbix Cloud-Konto wechseln Sie zu *Konto* > *Auditprotokoll*, um Auditprotokolleinträge anzuzeigen. Die für einen Benutzer verfügbaren Auditprotokolleinträge hängen von der Organisationsrolle des Benutzers ab.



Anzeigen von Audit-Log-Einträgen Audit-Log-Einträge werden in einer Tabelle angezeigt.



Die Tabelle enthält die folgenden Informationen:

Column	Description
Created	Datum und Uhrzeit, zu der der Logeintrag erstellt wurde. Die angezeigte Zeit entspricht der vom Benutzer im Dropdown-Menü *User timezone* unter *Account* > *Profile settings* ausgewählten Zeitzone.
User	Name des Organisationsbenutzers, der die Aktion ausgeführt hat.
Role	Die Rolle des Benutzers, der die Aktion ausgeführt hat.
Node	Name des Knotens, der mit der Aktion verknüpft ist.
Action	Name der ausgeführten Aktion.
Description	Beschreibung der ausgeführten Aktion.

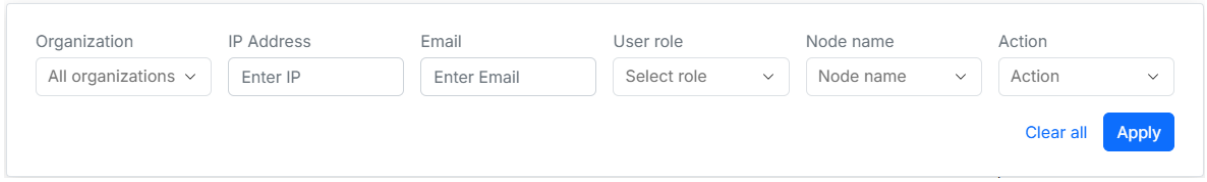
Auf die Spaltennamen kann geklickt werden, um die Einträge auf- oder absteigend zu sortieren.

Gefilterte Audit-Log-Einträge können mit der Schaltfläche *Export* im Format `.csv` exportiert werden.

Filter verwenden

Klicken Sie auf die Schaltfläche *Filter*, um den Filter zu öffnen.

Um die gefilterten Ergebnisse anzuzeigen, klicken Sie nach der Auswahl eines oder mehrerer Filterparameter auf *Anwenden*. Um den Filter zu löschen, klicken Sie auf *Alle löschen*.



Datensätze können nach einem oder mehreren Parametern gefiltert werden. Die folgenden Filterparameter sind verfügbar:

Parameter	Beschreibung
Organisation	Wählen Sie alle Organisationen oder eine bestimmte Organisation aus.
IP-Adresse	Geben Sie eine IP-Adresse an.
E-Mail	Geben Sie eine mit dem Benutzer verknüpfte E-Mail-Adresse an.
Benutzerrolle	Wählen Sie alle Rollen oder eine bestimmte Rolle aus: Owner, Admin oder User.

Parameter	Beschreibung
Knotenname	Wählen Sie alle Knoten oder einen bestimmten Knoten aus.
Aktion	Wählen Sie einen bestimmten Aktionstyp aus.

Das Audit-Protokoll erfasst die folgenden Aktionen:

- *Knoten erstellt*
- *Knoten gelöscht*
- *Knoten aktualisiert*
- *Knotenversion aktualisiert*
- *Passwort zurückgesetzt*
- *Wartung aktualisiert*
- *Verlaufseinstellung aktualisiert*
- *Datum des Knotenablaufs verlängert*
- *Proxy-Konfiguration aktualisiert*
- *Backup erstellt*
- *Backup gelöscht*
- *Backup wiederhergestellt*
- *Backup aktualisiert*
- *Zugriffsfilter hinzugefügt*
- *Zugriffsfilter aktualisiert*
- *Zugriffsfilter gelöscht*
- *Verschlüsselung aktiviert*
- *Verschlüsselung deaktiviert*
- *Abonnement gekündigt*
- *Abonnement verlängert*
- *Abonnementzahlung fehlgeschlagen*
- *Zahlung erfordert Bestätigung*
- *Organisation erstellt*
- *Organisation gelöscht*
- *Admin eingeladen*
- *Benutzer eingeladen*
- *Admin beigetreten*
- *Benutzer beigetreten*
- *Admin aus der Organisation entfernt*
- *Benutzer aus der Organisation entfernt*
- *Knoten zur Organisation hinzugefügt*
- *Knoten aus der Organisation entfernt*
- *Knotenzugriff gewährt*
- *Knotenzugriff entzogen*
- *Benutzereinstellungen aktualisiert*
- *Zahlung erstellt*
- *Zahlung abgeschlossen*
- *Zahlung fehlgeschlagen*
- *Mehrwertsteuer berechnet*
- *Mehrwertsteuerprüfung fehlgeschlagen*
- *Abrechnungsinformationen aktualisiert*
- *Zahlungsmethode aktualisiert*
- *Zahlungsmethode entfernt*
- *Prepaid-Knoten erstellt*
- *Prepaid-Knoten aktualisiert*
- *Vorfall gestartet*
- *Vorfall beendet*

Einen Zabbix Cloud Proxy einrichten

Übersicht Auf dieser Seite wird beschrieben, wie Sie einen auf Zabbix Cloud bereitgestellten Zabbix Proxy konfigurieren, um Überwachungsdaten von Hosts in einer entfernten Umgebung zu erfassen.

Das Verfahren ist in erster Linie für einen auf Zabbix Cloud erstellten Zabbix Server und Proxy vorgesehen.

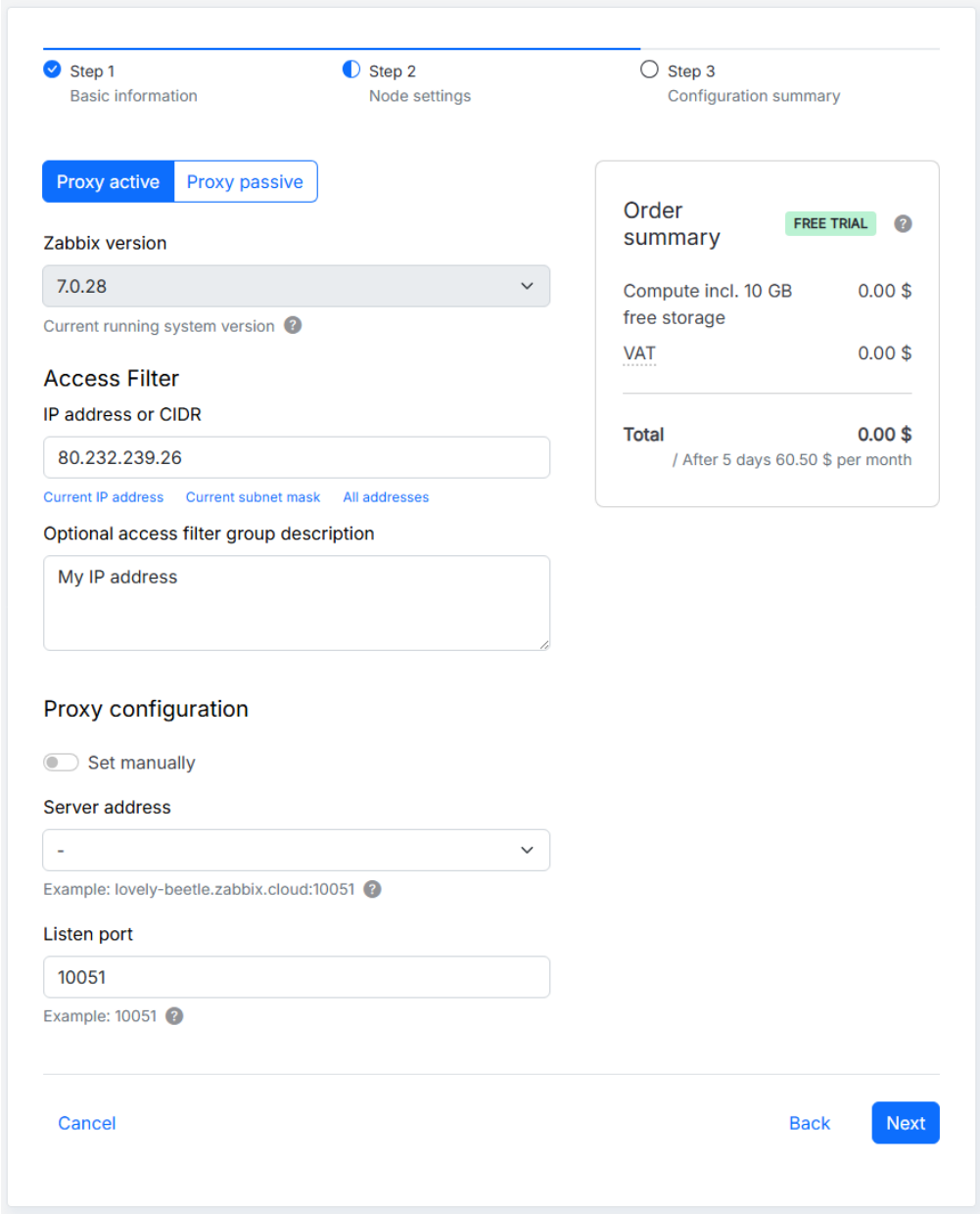
Das Verfahren beschreibt außerdem die erforderliche Konfiguration, wenn der Zabbix Server nicht auf Zabbix Cloud gehostet wird.

Voraussetzungen Ein Zabbix Server ist installiert, konfiguriert und wird ausgeführt, oder ein Zabbix Server-Knoten wird in Zabbix Cloud ausgeführt.

Einen Proxy-Knoten erstellen Erstellen Sie einen Zabbix Cloud-Proxy-Knoten.

Gehen Sie beim Erstellen des Knotens wie folgt vor:

1. Wählen Sie *Proxy* aus.
2. Wählen Sie *Proxy aktiv* oder *Proxy passiv* aus, je nachdem, wie der Proxy mit dem Zabbix Server kommunizieren soll.
3. Konfigurieren Sie die IP-Adresse oder die CIDR-Zugriffsfilter des Knotens (dies kann auch nach der Erstellung des Knotens erfolgen).
4. Geben Sie den Zabbix Server an, mit dem der Proxy eine Verbindung herstellen soll.



5. Schließen Sie die Erstellung des Knotens ab.

Weitere Informationen zum Erstellen eines Proxy-Knotens finden Sie unter Zabbix in der Cloud bereitstellen.

Zugriffsfilter konfigurieren Zabbix Cloud verwendet Zugriffsfilter, um zu steuern, welche IP-Adressen oder Subnetze eine Verbindung zu einem Cloud-Knoten herstellen können.

Während der Initialisierung des Knotens wird automatisch ein Zugriffsfilter für die IP-Adresse erstellt, die für den Zugriff auf den Knoten verwendet wird. Dies schließt nicht unbedingt die Adressen überwachter Hosts oder anderer Zabbix-Komponenten ein.

Fügen Sie die erforderlichen Adressen manuell hinzu.

Weitere Informationen finden Sie unter Knotenkonfiguration — Zugriffsfilter.

Proxy-Knoten

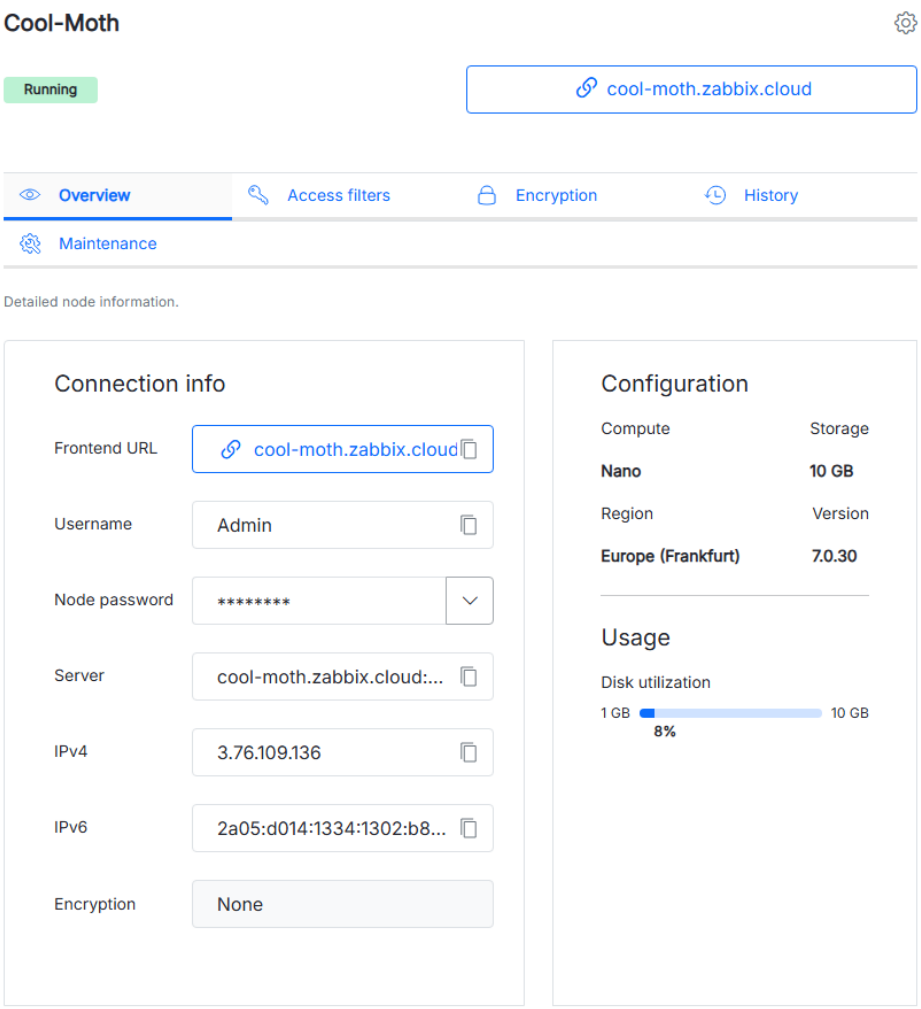
Fügen Sie die IP-Adressen oder Subnetze hinzu, von denen Verbindungen zum Proxy erwartet werden.

Wenn überwachte Hosts aktive Agent-Prüfungen verwenden, initiieren die Agents Verbindungen zum Proxy. Daher muss die IP-Adresse oder das Subnetz der überwachten Hosts durch die Zugriffsfilter des Proxy-Knotens zugelassen werden.

Wenn *Proxy Passive* als Proxy-Modus festgelegt wurde, stellt der Zabbix-Server eine Verbindung zum Proxy her. Daher muss die IP-Adresse oder das Subnetz des Zabbix-Servers durch die Zugriffsfilter des Proxy-Knotens zugelassen werden.

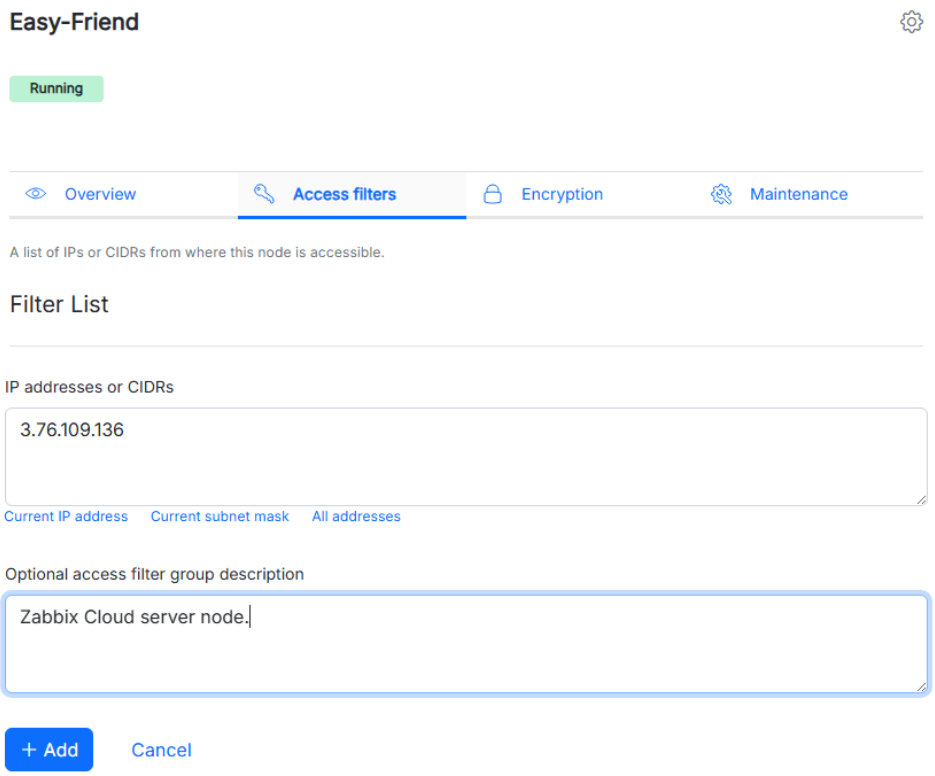
Wenn *Proxy passive* ausgewählt wurde und ein Serverknoten verwendet wird:

1. Navigieren Sie zur Registerkarte *Overview* Ihres Serverknotens und kopieren Sie die IPv4- oder IPv6-Adresse des Knotens.

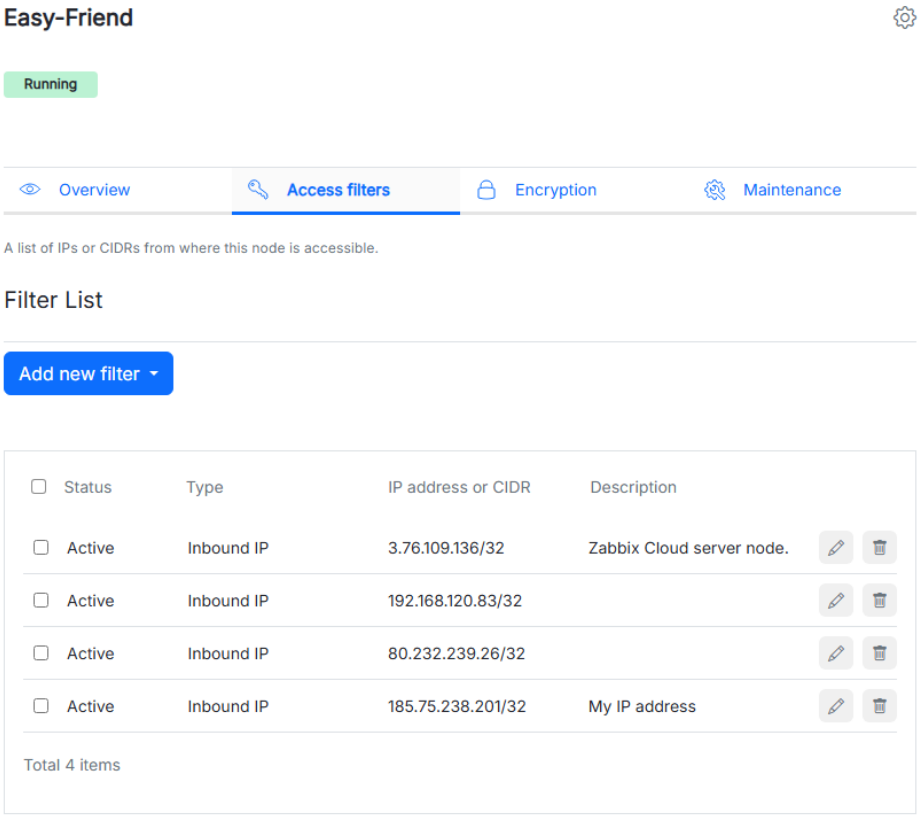


2. Navigieren Sie zur Registerkarte *Access filters* Ihres Proxy-Knotens und klicken Sie auf *Add new filter* > *Add*.

3. Fügen Sie die kopierte Adresse in das Feld *IP addresses or CIDRs* ein und geben Sie im Feld *Optional access filter group description* eine Beschreibung ein.



4. Klicken Sie auf *Add*.



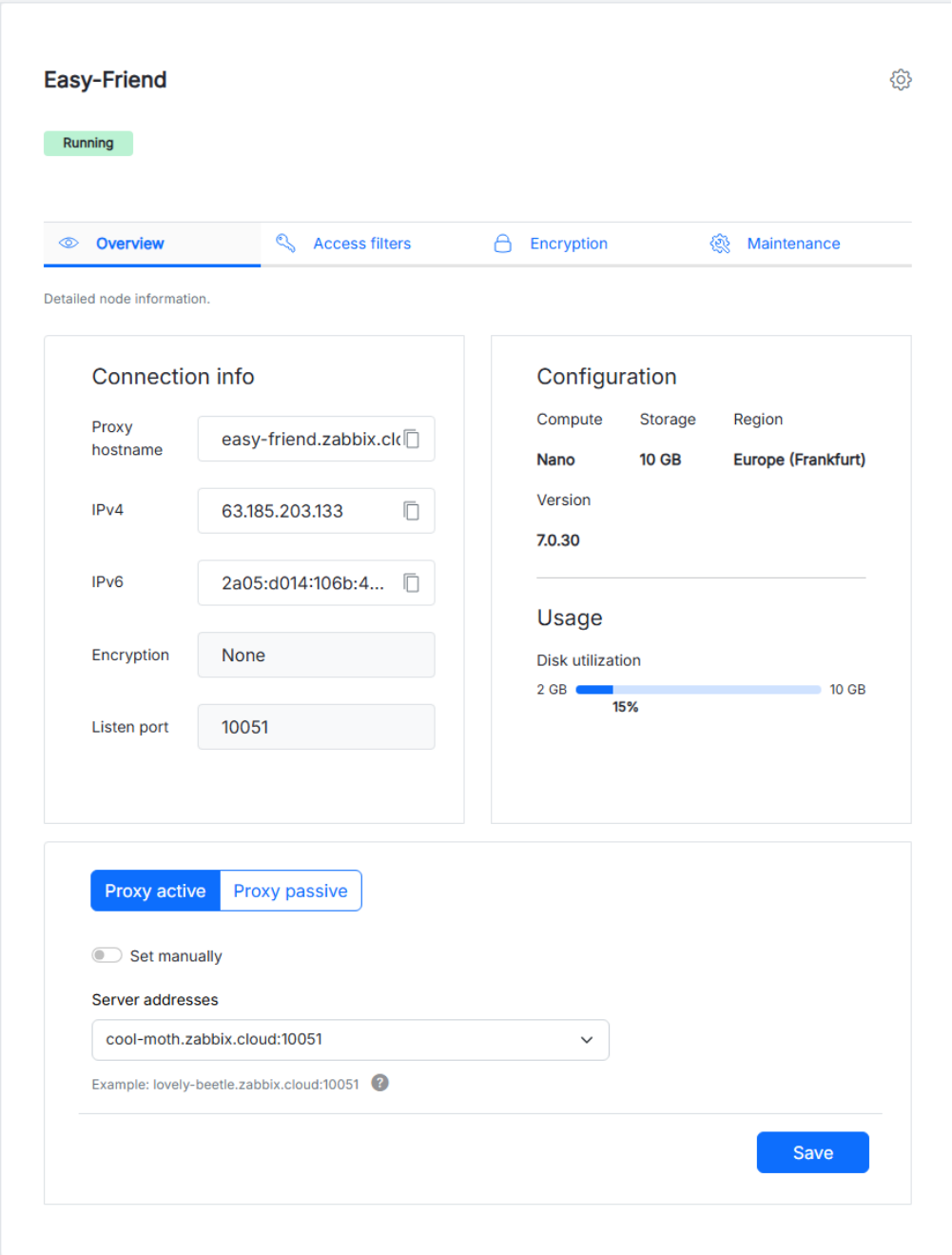
Führen Sie die Schritte 2-4 für die IP-Adresse eines Zabbix-Agents aus, der aktive Prüfungen verwendet.

Serverknoten

Wenn der Zabbix Server in der Zabbix Cloud gehostet wird, konfigurieren Sie die Zugriffsfilter auch auf dem Serverknoten.

Fügen Sie bei einem Proxy im aktiven Modus die IP-Adresse oder CIDR des Proxy-Knotens zu den Zugriffsfiltern des Serverknotens hinzu, da der Proxy die Verbindung zum Server initiiert.

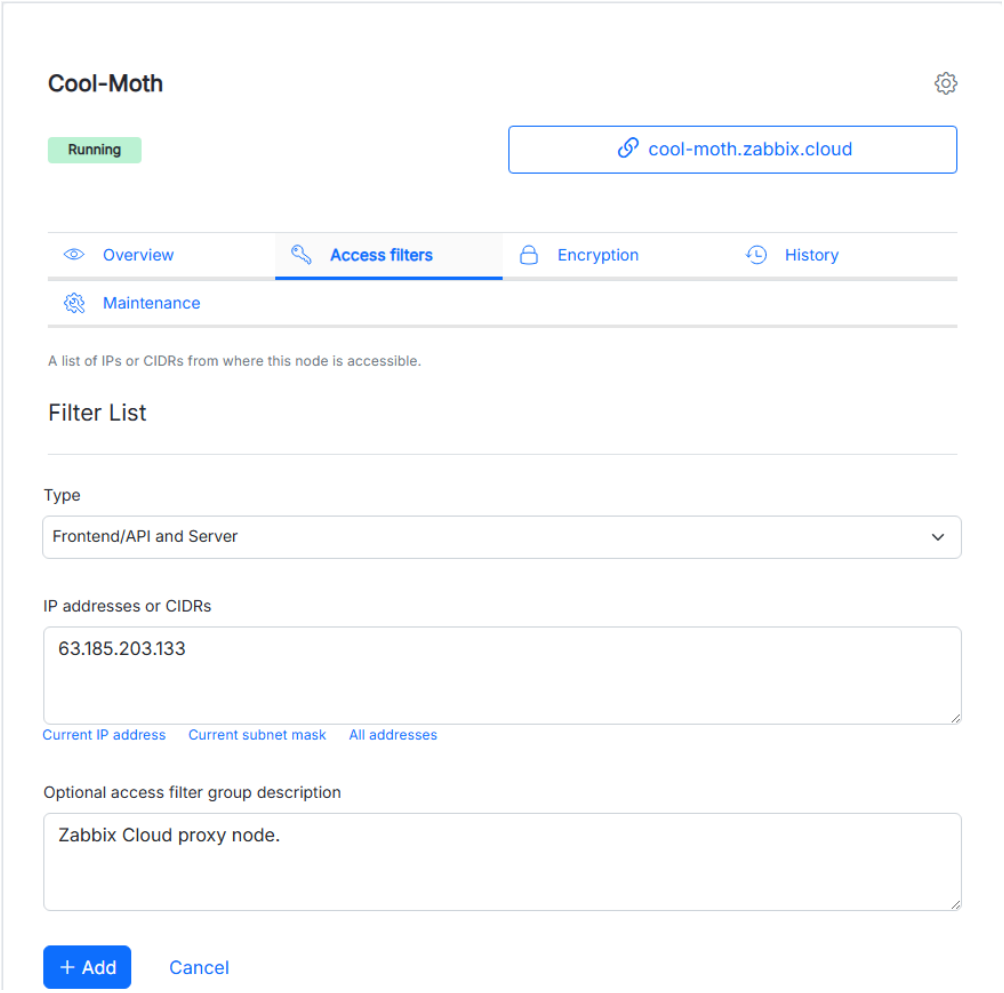
1. Navigieren Sie zur Registerkarte *Übersicht* Ihres Proxy-Knotens und kopieren Sie die IPv4- oder IPv6-Adresse des Knotens.



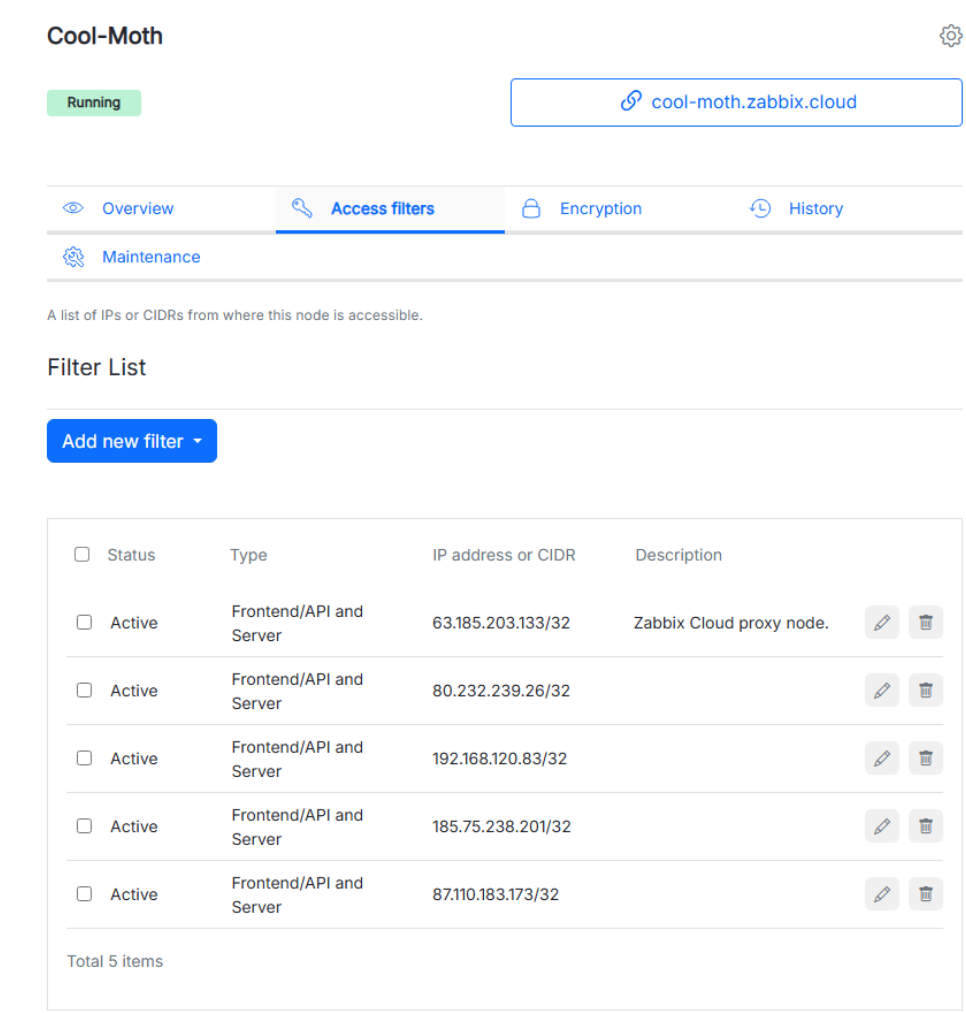
2. Navigieren Sie zur Registerkarte *Zugriffsfilter* Ihres Serverknotens und klicken Sie auf *Neuen Filter hinzufügen* > *Hinzufügen*.

3. Fügen Sie die kopierte Adresse in das Feld *IP-Adressen oder CIDRs* ein und geben Sie im Feld *Optionale Beschreibung der Zugriffsfiltergruppe* eine Beschreibung ein.

4. Wählen Sie den entsprechenden Filtertyp aus.



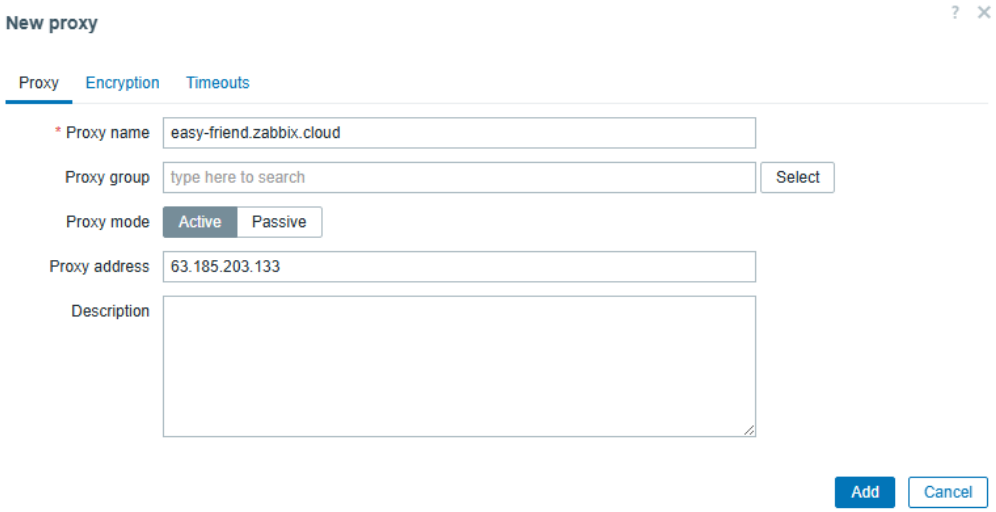
5. Klicken Sie auf *Hinzufügen*.



Proxy konfigurieren Der Proxy muss im Zabbix Frontend konfiguriert werden, damit er mit dem Zabbix Server kommunizieren kann.

Öffnen Sie im Zabbix Frontend *Administration* > *Proxies*, klicken Sie auf *Create proxy* und konfigurieren Sie den Proxy wie erforderlich:

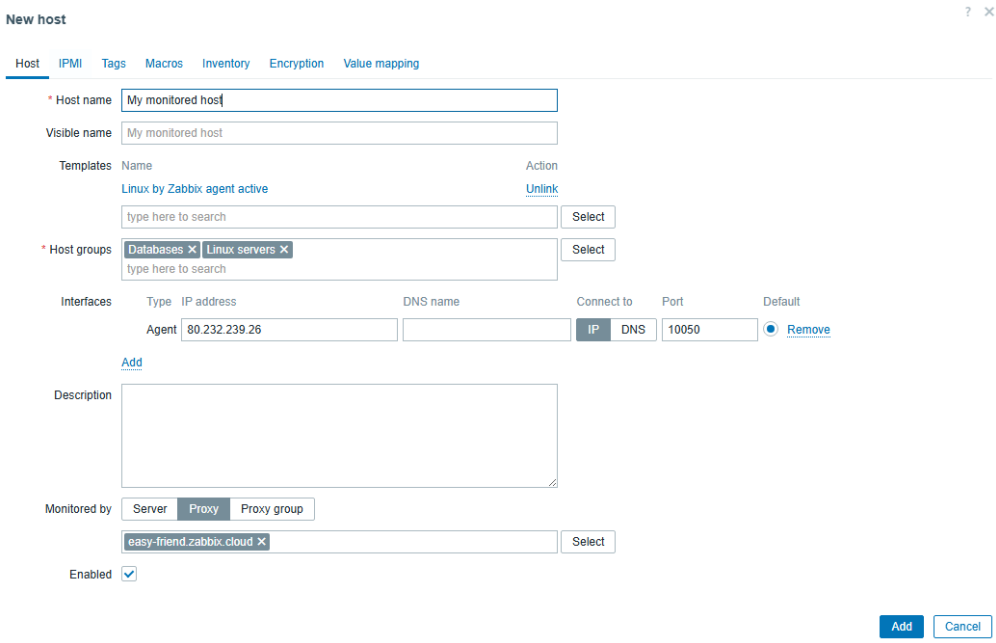
1. Fügen Sie den Namen des Proxys in das Feld *Proxy name* ein. Er muss mit dem für den Proxy konfigurierten Proxy-Hostnamen übereinstimmen. Geben Sie ihn ein oder kopieren Sie ihn aus der Registerkarte *Overview* Ihres Proxys.
2. Wählen Sie den Proxy-Modus aus, der dem bei der Proxy-Erstellung ausgewählten Modus entspricht.
3. Geben Sie bei einem aktiven Proxy die IP-Adresse oder den DNS-Namen des Proxys ein.
4. Klicken Sie auf *Add*.



Weitere Informationen finden Sie unter Verteiltes Monitoring mit Proxys.

Überwachte Hosts konfigurieren Fügen Sie die Hosts hinzu, die über den Proxy überwacht werden sollen.

1. Gehen Sie zu *Datenerfassung* > *Hosts*.
2. Erstellen Sie einen Host oder öffnen Sie einen vorhandenen Host.
3. Wählen Sie unter *Überwacht von* den *Proxy* aus. Geben Sie entweder den Namen des Proxys ein, um danach zu suchen und ihn auszuwählen, oder klicken Sie auf die Schaltfläche *Auswählen*, um eine Liste der verfügbaren Proxys zu öffnen.
4. Konfigurieren Sie die Host-Schnittstelle und weitere Parameter.
5. Klicken Sie auf *Hinzufügen* oder *Aktualisieren*.



Ein Host, der für die Überwachung durch einen Proxy konfiguriert ist, sendet oder empfängt Überwachungsdaten über diesen Proxy, abhängig vom Datentyp und dem Agent-Modus.

Zabbix Agent konfigurieren Wenn ein überwachter Host den Zabbix Agent oder Zabbix Agent 2 verwendet, konfigurieren Sie den Agent so, dass er mit der entsprechenden Zabbix-Komponente kommuniziert.

Die Konfiguration hängt davon ab, ob der Agent *passive* oder *aktive* Prüfungen verwendet.

Passive Agent checks

Bei passiven Checks initiiert der Zabbix Proxy Verbindungen zum Agent und fordert Überwachungsdaten an.

Öffnen Sie die Konfigurationsdatei des Zabbix Agent und aktualisieren Sie den Parameter `Server`, um Verbindungen von der IP-Adresse des Zabbix Cloud Proxy zuzulassen.

Beispiel:

```
Server=<proxy_IP>
```

Stellen Sie sicher, dass der überwachte Host eingehende TCP-Verbindungen vom Proxy am Agent-Port zulässt, der standardmäßig 10050 lautet.

Der Proxy muss den überwachten Host über das Netzwerk erreichen können.

Aktive Agent-Prüfungen

Bei aktiven Prüfungen initiiert der Agent Verbindungen zum Zabbix Proxy und sendet die erfassten Überwachungsdaten.

Öffnen Sie die Konfigurationsdatei des Zabbix Agent und aktualisieren Sie den Parameter `ServerActive` mit der Adresse des Zabbix Cloud Proxy.

Beispiel:

```
ServerActive=<proxy_IP>:<proxy_listen_port>
```

Ersetzen Sie `<proxy_IP>` durch die IP-Adresse des Zabbix Cloud Proxy und `<proxy_listen_port>` durch den beim Erstellen des Proxy-Knotens konfigurierten Empfangsport. Der standardmäßige Empfangsport ist 10051.

Wenn in `ServerActive` ein Proxy angegeben ist, ruft der Agent seine Konfiguration für aktive Prüfungen vom Proxy und nicht direkt vom Zabbix Server ab.

Stellen Sie sicher, dass der überwachte Host eine ausgehende TCP-Verbindung zum Proxy über den konfigurierten Empfangsport herstellen kann.

Weitere Informationen zu passiven und aktiven Agent-Prüfungen finden Sie unter Passive and active agent checks.

Externer Zabbix-Server Ein Zabbix-Cloud-Proxy kann auch mit einem außerhalb der Zabbix Cloud bereitgestellten Zabbix-Server kommunizieren.

In diesem Fall muss der externe Zabbix-Server entsprechend dem ausgewählten Proxy-Modus mit dem Cloud-Proxy kommunizieren können.

Bei einem Proxy im aktiven Modus initiiert der Proxy die Verbindung zum externen Zabbix-Server. Konfigurieren Sie den externen Zabbix-Server so, dass er Verbindungen vom Proxy akzeptiert, und stellen Sie sicher, dass der Proxy den Server über das Netzwerk erreichen kann.

Bei einem passiven Proxy initiiert der Zabbix-Server die Verbindung zum Proxy. Fügen Sie die IP-Adresse oder das Subnetz des externen Zabbix-Servers zu den Zugriffsfiltern des Proxy-Knotens hinzu und stellen Sie sicher, dass der Server den Proxy über das Netzwerk erreichen kann.

Der externe Zabbix-Server verwendet keine Zugriffsfilter der Zabbix Cloud. Konfigurieren Sie stattdessen die erforderlichen Firewall- und Netzwerkregeln in der externen Umgebung.

Die überwachten Hosts können sich in derselben Umgebung wie der externe Zabbix-Server, in einem anderen Netzwerk oder in einem über den Proxy erreichbaren Netzwerk befinden.

Wenn Agents aktive Prüfungen verwenden, konfigurieren Sie ihren Parameter `ServerActive` so, dass er auf den Zabbix-Cloud-Proxy verweist. Wenn Agents passive Prüfungen verwenden, konfigurieren Sie ihren Parameter `Server` so, dass Verbindungen vom Proxy zugelassen werden.

Konfiguration überprüfen Nachdem Sie den Proxy, die Agents, Zugriffsfilter und die Netzwerkkonnektivität konfiguriert haben, überprüfen Sie, ob der Proxy mit dem Zabbix-Server kommuniziert.

1. Gehen Sie zu *Administration* > *Proxies*.
2. Überprüfen Sie den *State* des Proxys.
3. Gehen Sie zu *Monitoring* > *Latest data* und überprüfen Sie, ob aktuelle Datenpunktwerte verfügbar sind.

Wenn der Proxy nicht mit dem Server kommuniziert, überprüfen Sie:

- den Proxy-Modus;
- die Serveradresse und den Port;
- die Proxy-Zugriffsfilter;
- die Server-Zugriffsfilter, wenn der Server in der Zabbix Cloud gehostet wird;
- die Agent-Konfiguration;
- Firewall- und Routingregeln;
- die in `Server` oder `ServerActive` konfigurierte Adresse.

Mobile-App

Mobile App

Mit der Zabbix Mobile App können Sie Ihre Zabbix-Umgebung von Ihrem Mobilgerät aus überwachen. Zeigen Sie Probleme an, prüfen Sie deren Details, aktualisieren Sie Probleme und erhalten Sie Push-Benachrichtigungen, wenn etwas Ihre Aufmerksamkeit erfordert.

App herunterladen (demnächst verfügbar)

Zabbix für die App einrichten

Zabbix-Bridge-Adapter einrichten Installieren Sie den Zabbix-Bridge-Adapter, damit mobile Geräte eine Verbindung zu Zabbix herstellen können.

Geräte hinzufügen Fügen Sie Ihr Mobilgerät im Zabbix-Frontend hinzu, um es mit Ihrem Zabbix-Server zu verbinden.

Push-Benachrichtigungen einrichten Erhalten Sie Benachrichtigungen auf Ihrem Mobilgerät, wenn ein Problem auftritt.

Informationen zur App

App-Übersicht Erfahren Sie die Grundlagen der Verwendung der mobilen Zabbix-App.

Sicherer mobiler Zugriff Sichern Sie die App, Ihre mobilen Geräte und den Zabbix bridge adapter oder entziehen Sie verlorenen oder kompromittierten Geräten den Zugriff.

Zabbix für die App einrichten

Um die Zabbix mobile app (demnächst verfügbar) zu verwenden, müssen Sie zunächst:

1. Den Zabbix bridge adapter einrichten.
2. Push-Benachrichtigungen einrichten.
3. Ihr Mobilgerät im Zabbix-Frontend hinzufügen.

Die folgenden Abschnitte behandeln jeden Schritt.

Nach der Einrichtung finden Sie unter Secure mobile access Informationen dazu, wie Sie die App und den Zabbix bridge adapter sicher halten.

Zabbix-Bridge-Adapter einrichten Damit Ihr Zabbix-Server mit der mobilen App kommunizieren kann, müssen Sie den Zabbix-Bridge-Adapter einrichten.

Der Bridge-Adapter kann auf demselben System wie Ihr Zabbix-Server oder mithilfe von Paketen auf einem anderen System installiert werden:

> **Attention:**
> Die folgenden Schritte gelten für Ubuntu 24.04. Für andere Betriebssysteme verwenden Sie die Befehle zur Repository- und Paketinstallation (z. B. `wget`, `dpkg`, `apt`) von der Seite Zabbix herunterladen für Ihr Betriebssystem und installieren Sie damit das Paket `zabbix-bridge-adapter`.

1. Starten Sie eine neue Shell-Sitzung mit Root-Berechtigungen:

```
sudo -s
```

2. Installieren Sie das Zabbix-Repository:

```
wget https://repo.zabbix.com/zabbix/8.0/release/ubuntu/pool/main/z/zabbix-release/zabbix-release_latest_8.
dpkg -i zabbix-release_latest_8.0+ubuntu24.04_all.deb
apt update
```

3. Installieren Sie den Zabbix-Bridge-Adapter:

```
apt install zabbix-bridge-adapter
```

4. Bearbeiten Sie die Konfigurationsdatei des Zabbix-Servers:

```
EnableMobileDevices=1
BridgeAdapterURL=https://bridge-adapter.example.com:10054/rpc
```

5. Starten Sie den Zabbix-Server neu, zum Beispiel (siehe weitere Optionen):

```
systemctl restart zabbix-server
```

6. Kopieren Sie die Server-ID des Zabbix-Servers aus *Reports* > *System information*; Sie verwenden sie im nächsten Schritt.

7. Bearbeiten Sie die Konfigurationsdatei des Zabbix-Bridge-Adapters:

```
ZabbixAPIURL=https://example.com/zabbix/api_jsonrpc.php
ServerId=019fb1f9-22bd-74ea-bc03-51efb8b76aba
DataDir=/var/lib/zabbix-bridge-adapter
```

Attention:
Das in DataDir angegebene Verzeichnis muss über die Berechtigungen 0700 verfügen, und alle darin gespeicherten Dateien müssen über die Berechtigungen 0600 verfügen.

8. Lesen Sie die Endbenutzer-Lizenzvereinbarung (EULA) des Zabbix-Bridge-Adapters und akzeptieren Sie sie:

```
zabbix_bridge_adapter license show
sudo -u zabbix-bridge-adapter zabbix_bridge_adapter -c /etc/zabbix/zabbix_bridge_adapter.conf license acce
```

9. Starten Sie den Zabbix-Bridge-Adapter:

```
systemctl start zabbix-bridge-adapter
```

Attention:
Beim Start verwendet der Zabbix-Bridge-Adapter die Server-ID, um Ihren Zabbix-Server beim zentral gehosteten Zabbix Bridge zu registrieren. Der Registrierungsstatus wird in einer Statusdatei in DataDir gespeichert. Wenn Sie den Bridge-Adapter neu installieren und die Statusdatei verloren geht, schlägt die Registrierung desselben Zabbix-Servers möglicherweise fehl. Um dies zu vermeiden, sichern Sie die Statusdatei, bevor Sie den Bridge-Adapter neu installieren, und stellen Sie sie anschließend wieder her.

Push-Benachrichtigungen einrichten So erhalten Sie Benachrichtigungen auf Ihrem Mobilgerät, wenn in Ihrer überwachten Umgebung etwas Aufmerksamkeit erfordert:

1. Erstellen Sie einen Medientyp für Push-Benachrichtigungen (oder aktivieren Sie den standardmäßigen).
2. Erstellen Sie eine Aktion (oder aktivieren Sie die standardmäßige Auslöseraktion *Probleme an Zabbix-Administratoren melden*).
3. Fügen Sie ein Gerät hinzu, um den Medientyp automatisch zu Ihrem Benutzer hinzuzufügen (oder fügen Sie Benutzermedien manuell hinzu).

Geräte hinzufügen Um die mobile Zabbix-App auf Ihrem Gerät zu verwenden, müssen Sie dieses Gerät im Zabbix-Frontend hinzufügen. Anweisungen finden Sie unter:

- *Benutzer* > *Geräte* (als *Super-Admin* können Sie Ihre eigenen mobilen Geräte oder die mobiler anderer Benutzer hinzufügen)
- *Benutzereinstellungen* > *Geräte* (als beliebiger Benutzer können Sie Ihre eigenen mobilen Geräte hinzufügen)

App-Übersicht

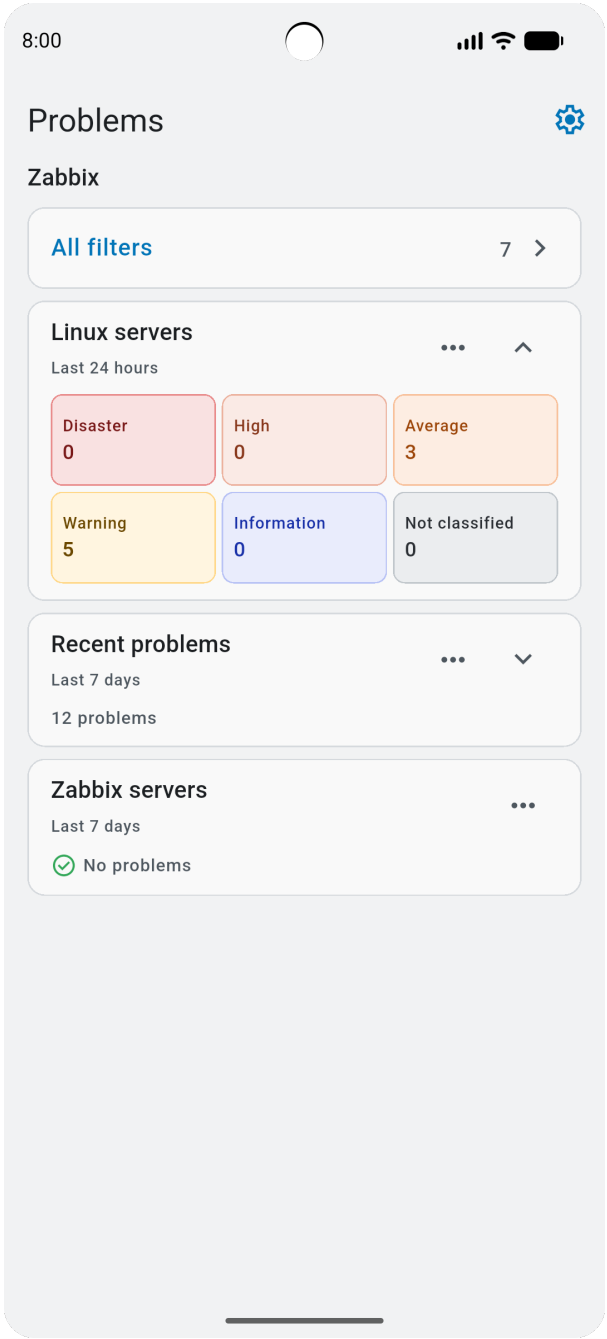
Mit der Zabbix Mobile App (demnächst verfügbar) können Sie Ihre Zabbix-Umgebung von Ihrem Mobilgerät aus überwachen. Zeigen Sie Probleme an, prüfen Sie deren Details, aktualisieren Sie Probleme und erhalten Sie Push-Benachrichtigungen, wenn etwas Ihre Aufmerksamkeit erfordert.

Diese Seite bietet Ihnen einen Überblick über die Mobile App und ihre Verwendung.

Probleme Der Bildschirm *Probleme* bietet Ihnen eine Übersicht der Probleme basierend auf Ihren angehefteten Filtern. Er enthält:

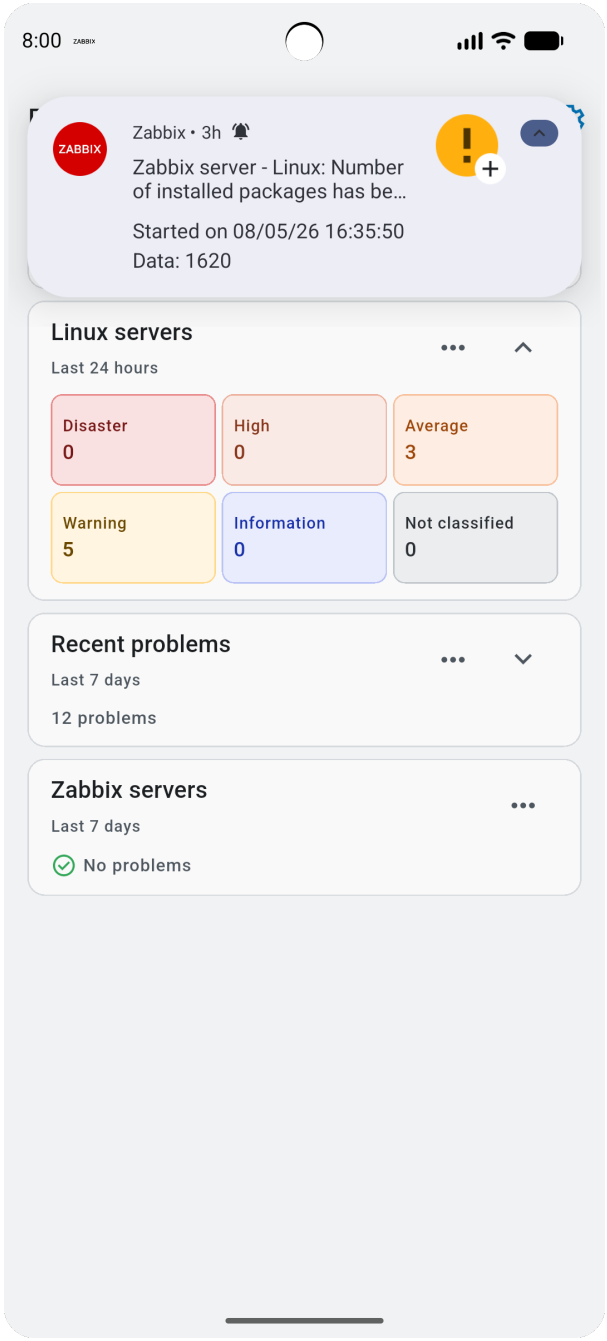
- Symbol *Einstellungen* – tippen Sie darauf, um die App-Einstellungen anzuzeigen.
- *Alle Filter* – tippen Sie darauf, um Filter anzuzeigen und zu verwalten.
- Name des Zabbix Servers (oder *Zabbix*, wenn bei der Installation der Zabbix-Weboberfläche kein Name konfiguriert wurde).

- Angeheftete Filter – tippen Sie auf den Namen oder einen Schweregrad, um den Bildschirm *Problemliste* zu öffnen. Tippen Sie auf das Drei-Punkte-Menü, um den Filter zu verwalten (zu bearbeiten, zu duplizieren, die Anheftung aufzuheben oder ihn zu löschen), oder tippen Sie auf den Pfeil, um ihn auf- oder zuzuklappen. Beachten Sie, dass angeheftete Filter alphabetisch sortiert sind und der Filter *Aktuelle Probleme* nicht bearbeitet oder entfernt werden kann.



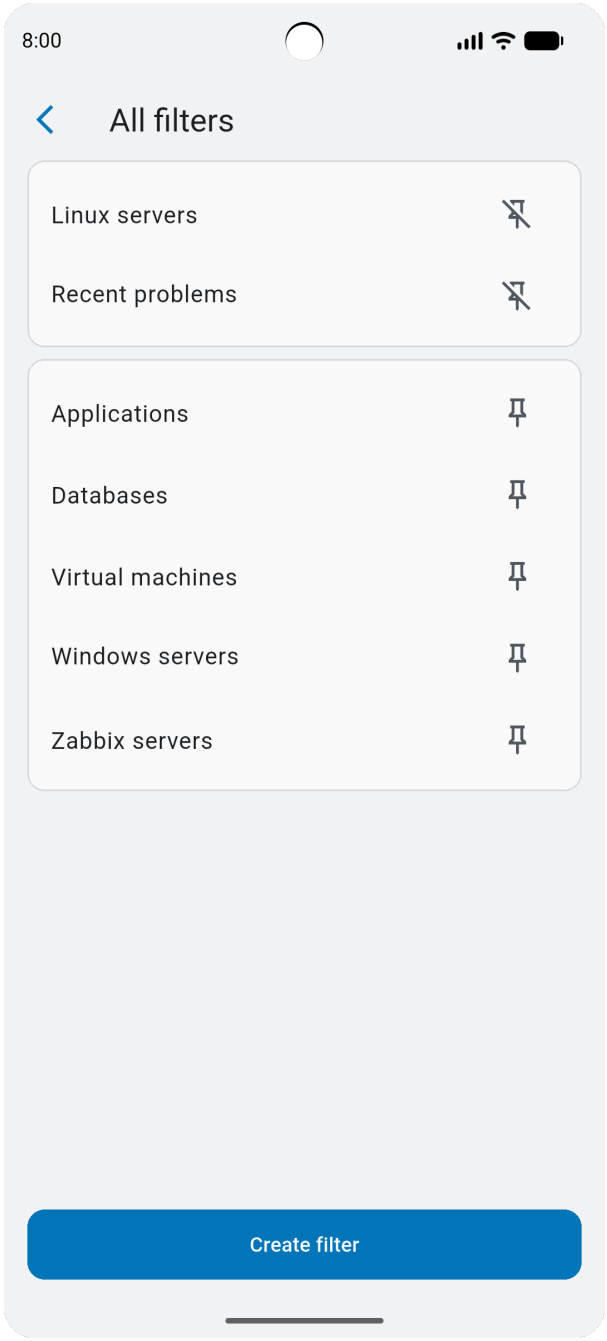
Benachrichtigungen Wenn der Zabbix Server ein Problem erkennt oder ein Problem aktualisiert bzw. behoben wird, kann er eine Benachrichtigung an Ihr Mobilgerät senden. Die Benachrichtigung enthält den Host des Problems, den Namen und eine aus den für den Medientyp Push-Benachrichtigung konfigurierten Nachrichtenvorlagen generierte Nachricht.

Tippen Sie auf die Benachrichtigung, um den Bildschirm *Problemdetails* zu öffnen.



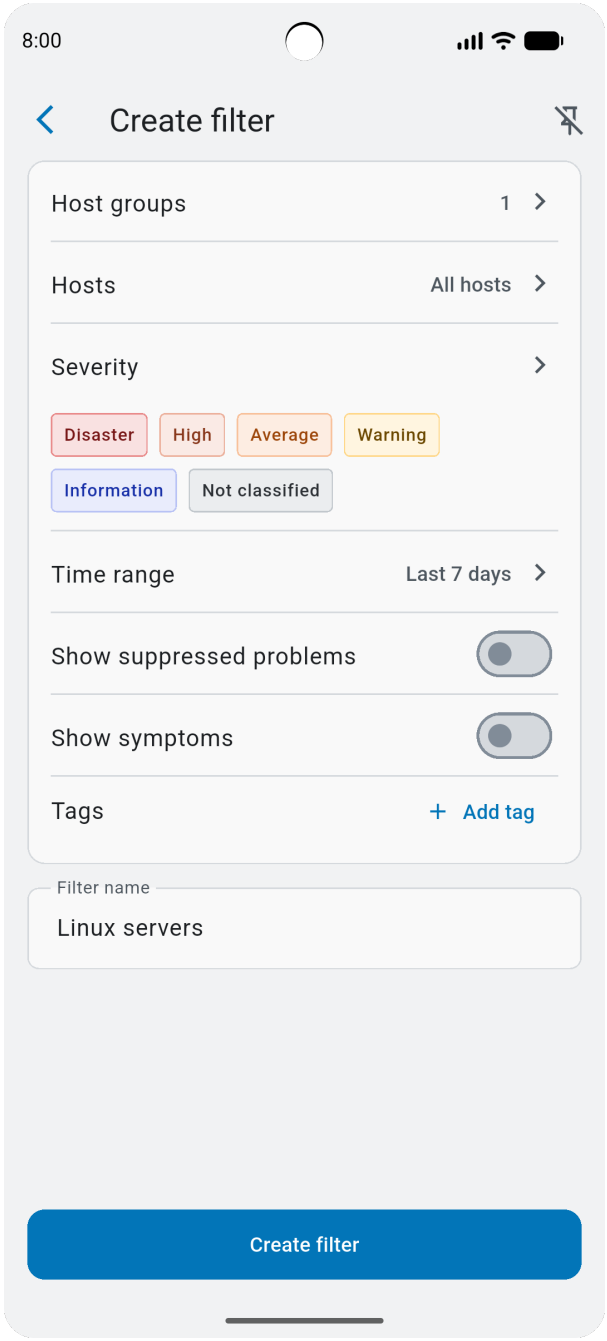
Alle Filter Auf dem Bildschirm *Alle Filter* können Sie Filter verwalten. Er enthält:

- Filternamen – tippen Sie darauf, um den Filter zu bearbeiten.
- Stecknadelsymbole – tippen Sie darauf, um den Filter auf dem Bildschirm *Probleme* anzuheften oder davon zu lösen.
- *Filter erstellen* – tippen Sie darauf, um einen neuen Filter zu erstellen.



Filter erstellen Auf dem Bildschirm *Filter erstellen* können Sie einen Filter mit Optionen erstellen, die denen unter *Monitoring* > *Probleme* ähneln (siehe Details). Sie können außerdem einen *Zeitraum* festlegen, für den ungelöste Probleme angezeigt werden sollen.

Tippen Sie auf *Filter erstellen*, um den Filter zu speichern. Neue Filter werden auf dem Bildschirm *Probleme* angeheftet. Tippen Sie auf das Stecknadelsymbol, um dies zu ändern.



Problemliste Der Bildschirm *Problemliste* zeigt Probleme an, die dem Filter entsprechen (z. B. *Aktuelle Probleme*).

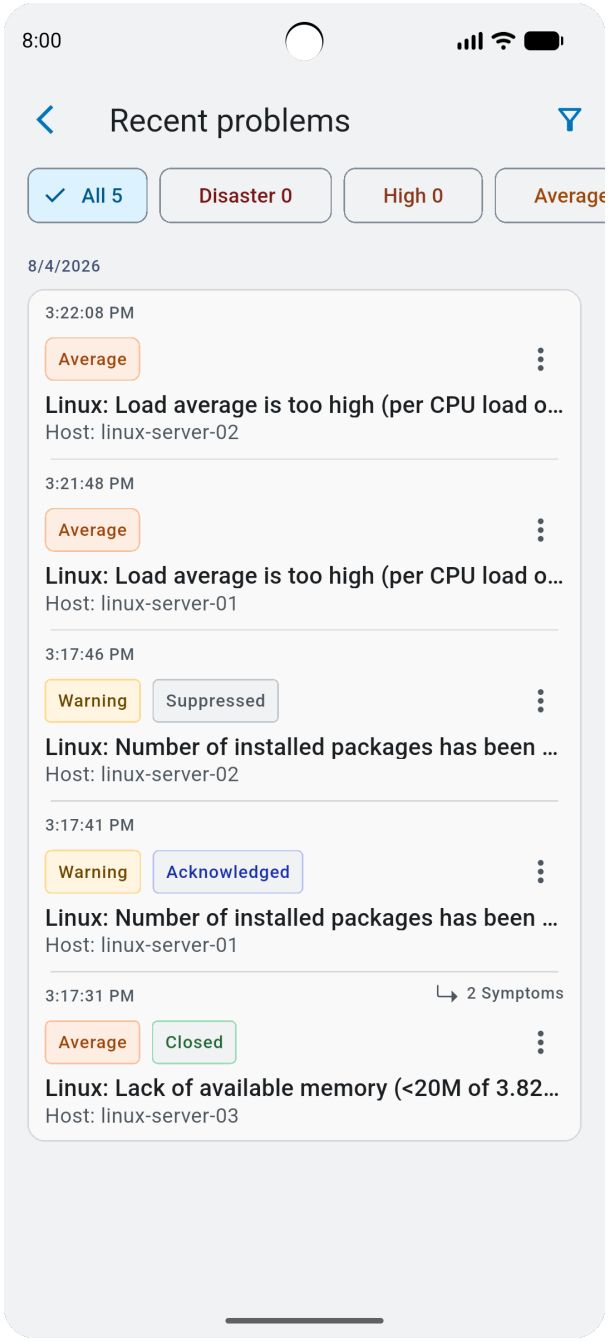
Tippen Sie auf das Filtersymbol oder einen Problemschweregrad, um die Liste weiter einzugrenzen.

Für jedes Problem werden folgende Informationen angezeigt:

- Startzeit und -datum.
- Schweregrad, Name und Host.
- Ob es unterdrückt wurde (wird angezeigt, wenn *Unterdrückte Probleme anzeigen* aktiviert ist).
- Ob es bestätigt oder kürzlich geschlossen wurde (automatisch gelöst oder manuell geschlossen).
- Ob es Symptome gibt – tippen Sie darauf, um die Problemliste der Symptome anzuzeigen.

Tippen Sie auf ein Problem, um den Bildschirm *Problemdetails* zu öffnen.

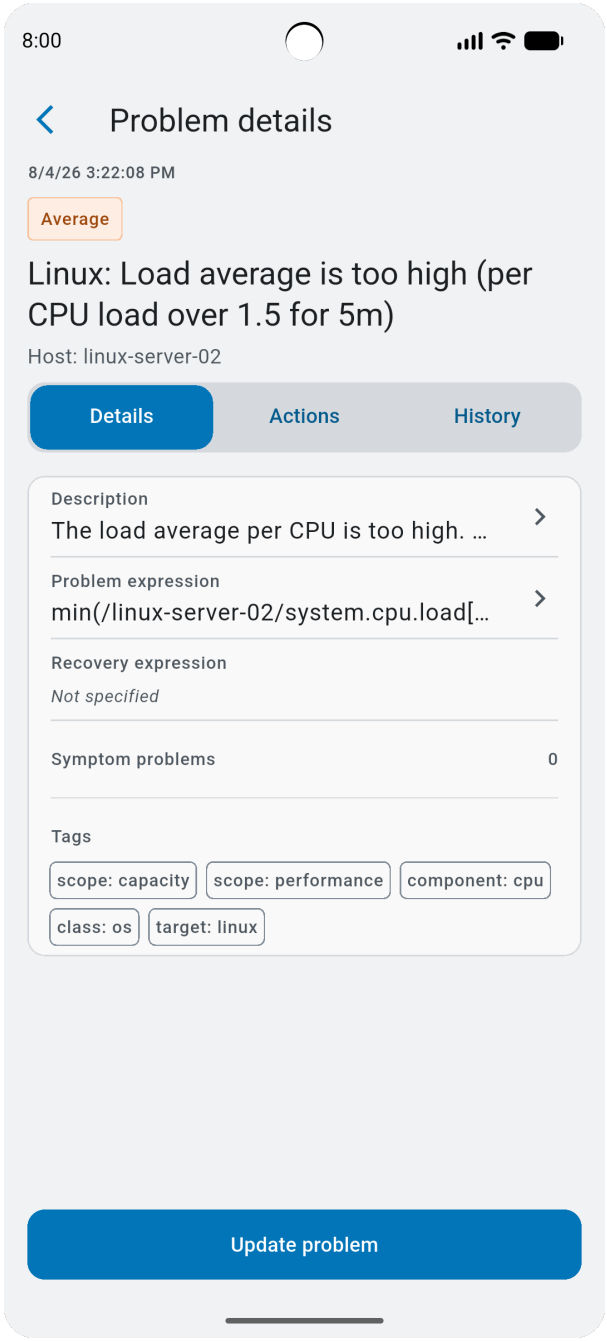
Tippen Sie auf das Drei-Punkte-Menü, um das Problem schnell zu bestätigen oder für einen von Ihnen ausgewählten Zeitraum zu unterdrücken.



Problemdetails Der Bildschirm *Problemdetails* zeigt Details zum Problem, Aktionen und den Verlauf.

- Registerkarte *Details* – zeigt die Problembeschreibung, Ausdrücke, Symptomprobleme (tippen, um die Problemliste anzuzeigen) und Tags.
- Registerkarte *Aktionen* – zeigt die Zeitleiste des Problems: wann es begonnen oder bestätigt wurde, ob eine Benachrichtigung gesendet wurde und weitere Aktionen.
- Registerkarte *Verlauf* – zeigt das Alter, die Dauer und die Wiederherstellungszeit des Problems (oder *Nicht verfügbar*, wenn das Problem weiterhin besteht).

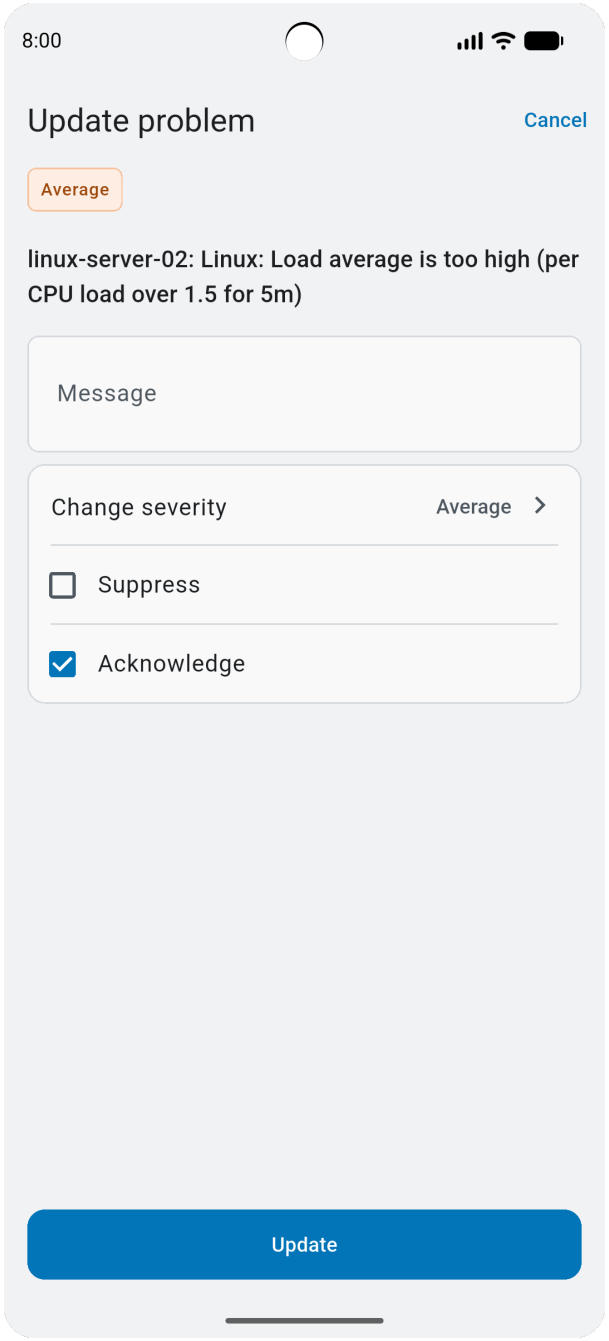
Tippen Sie auf *Problem aktualisieren*, um das Problem zu aktualisieren.



Problem aktualisieren Auf dem Bildschirm *Problem aktualisieren* können Sie ein Problem aktualisieren:

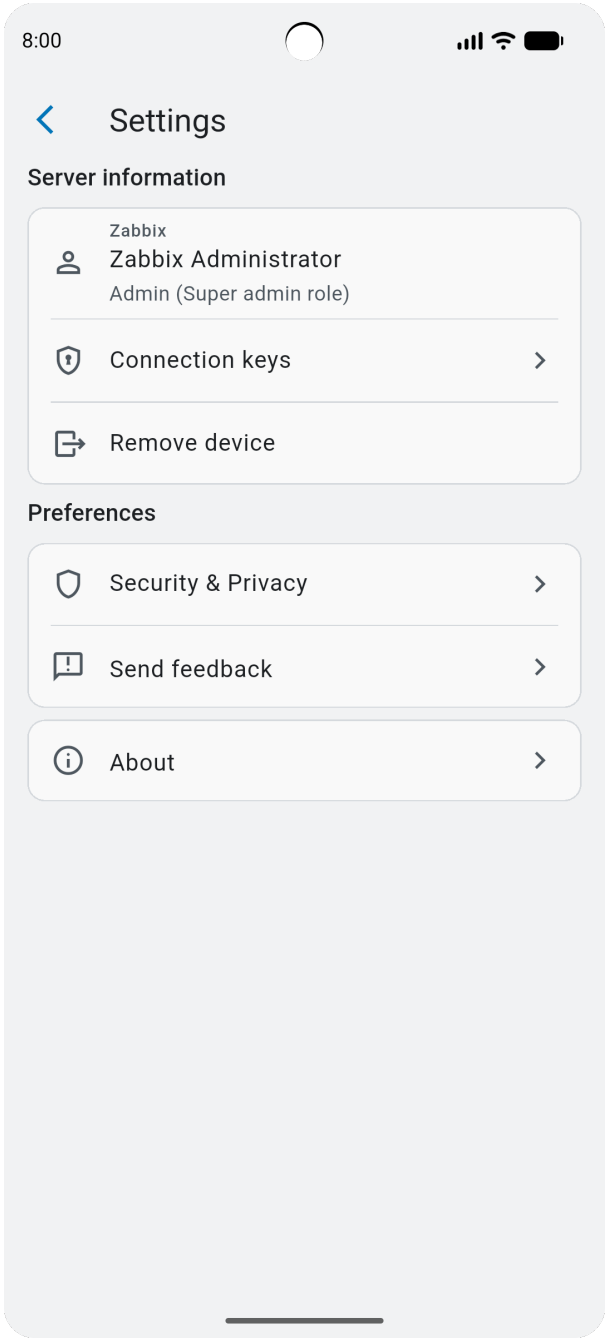
- Eine Nachricht hinzufügen.
- Den Schweregrad des Problems ändern.
- Das Problem für einen von Ihnen festgelegten Zeitraum unterdrücken.
- Das Problem bestätigen oder die Bestätigung aufheben.
- Das Problem schließen.

Tippen Sie auf *Aktualisieren*, um Ihre Änderungen zu speichern.



Einstellungen Der Bildschirm *Einstellungen* ermöglicht den Zugriff auf die App-Einstellungen. Er enthält:

- Den Namen des Zabbix-Servers (oder *Zabbix*, wenn bei der Installation des Zabbix-Webinterfaces kein Name konfiguriert wurde).
- Den Namen, Typ und die Rolle des Benutzers, dem das Mobilgerät hinzugefügt wurde.
- *Verbindungsschlüssel* – öffnen, um die Server-ID, Geräte-ID und weitere Details anzuzeigen und zu kopieren.
- *Gerät entfernen* – antippen, um das Mobilgerät aus Zabbix zu entfernen.
- *Sicherheit und Datenschutz* – öffnen, um die App-Sperre zu aktivieren und die Weitergabe von Diagnosedaten zu verwalten.
- *Feedback senden* – öffnen, um Feedback an Zabbix zu senden.
- *Über* – öffnen, um die App-Version und weitere Details zur App anzuzeigen.



Sicherer mobiler Zugriff

Diese Seite enthält Best Practices für die Sicherheit der mobilen Zabbix-App und des Zabbix-Bridge-Adapters.

Mobile Geräte schützen Um zu verhindern, dass andere Personen die App auf Ihrem mobilen Gerät öffnen, aktivieren Sie die App-Sperre in den *Einstellungen* der App.

Wenn ein Gerät verloren gegangen ist oder kompromittiert wurde, entfernen Sie es im Zabbix-Frontend. Dadurch werden seine Zugangsdaten widerrufen und entfernt, sodass das Gerät nicht mehr auf Zabbix zugreifen kann. Abhängig von Ihren Berechtigungen und Ihrer Benutzerrolle können Sie Geräte in den folgenden Bereichen entfernen:

- *Benutzereinstellungen > Geräte* - Jeder Benutzer kann seine eigenen Geräte entfernen (erfordert die Berechtigung *Eigene Geräte verwalten*).
- *Benutzer > Geräte* - Ein *Super-Admin* kann die Geräte beliebiger Benutzer entfernen (erfordert die Berechtigung *Benutzergeräte verwalten*).

Sie können außerdem einschränken, auf welche Teile von Zabbix von einem mobilen Gerät aus zugegriffen werden kann. Weisen Sie Benutzern dazu eine Benutzerrolle zu, die nur die von der App benötigten Berechtigungen gewährt. Die minimal erforderlichen Berechtigungen sind:

Bereich	Konfiguration
Zugriff auf UI-Elemente	Deaktivieren Sie alle Kontrollkästchen.
Zugriff auf Services	Setzen Sie *Lese-/Schreibzugriff auf Services* und *Nur-Lesezugriff auf Services* auf `None`.
Zugriff auf Module	Deaktivieren Sie alle Kontrollkästchen.
Zugriff auf API	Deaktivieren Sie das Kontrollkästchen *Aktiviert*.
Zugriff auf Aktionen	Deaktivieren Sie alle Kontrollkästchen. Damit Benutzer in der App Probleme aktualisieren können, markieren Sie zusätzlich die benötigten Aktionen: - *Problemkommentare hinzufügen* - *Schweregrad ändern* - *Probleme bestätigen* - *Probleme unterdrücken* - *Probleme schließen* - *Eigene Medien erstellen und bearbeiten* Beachten Sie, dass die Probleme, die Benutzer in der App sehen können, auch von ihrem Zugriff auf Hosts abhängen.
Geräte	Markieren Sie das Kontrollkästchen *Aktiviert*. Damit Benutzer ihre eigenen mobilen Geräte hinzufügen und entfernen können, markieren Sie zusätzlich das Kontrollkästchen *Eigene Geräte verwalten*. Andernfalls muss ein *Super-Admin* ihre Geräte unter *Benutzer > Geräte* verwalten.

Zabbix bridge adapter schützen Der Zabbix bridge adapter speichert seine Daten (Statusdatei und Verschlüsselungsschlüssel) in dem Verzeichnis, das im Parameter `DataDir` der Konfigurationsdatei des Zabbix bridge adapters angegeben ist. Stellen Sie sicher, dass das Verzeichnis über die Berechtigungen 0700 verfügt und alle darin gespeicherten Dateien die Berechtigungen 0600 haben.

Der Zabbix bridge adapter leitet Anfragen von mobilen Geräten an die Zabbix API weiter. Um einzuschränken, welche API-Methoden über den Zabbix bridge adapter aufgerufen werden können, verwenden Sie den Parameter `AllowedAPIMethods`. Erlauben Sie beispielsweise nur die minimale Menge an Methoden, die die mobile App benötigt, oder sogar weniger, wenn bestimmte Funktionen der mobilen App nicht verwendet werden sollen (z. B. `event.acknowledge` nicht erlauben, wenn Benutzer der mobilen App Probleme nicht aktualisieren sollen).

Entwicklerzentrum

Entwicklerzentrum

Beginnen Sie mit der Entwicklung benutzerdefinierter Erweiterungen für Zabbix mit Anleitungen zu Plugins, Modulen und Widgets.

Auf Zabbix aufbauen

Plugins Ein Überblick darüber, wie Zabbix-Plugins entwickelt und verwaltet werden, um die Funktionalität zu erweitern oder die Integration mit externen Systemen zu ermöglichen.

Python-Bibliothek für Zabbix Ressourcen zur Verwendung der offiziellen Bibliothek `zabbix_utils`, um mit der Zabbix-API zu interagieren und Integrationen zu erstellen.

Frontend-Module Anleitungen und Referenzen zum Erstellen benutzerdefinierter Module, die das Zabbix-Frontend erweitern oder anpassen, um spezifische Anwendungsfälle zu unterstützen.

Widgets Eine Aufschlüsselung der Widget-Struktur und -Logik mit Anleitungen zum Erstellen benutzerdefinierter Dashboard-Elemente, die auf Ihre Anforderungen zugeschnitten sind.

Erstellen eines Plugins

Diese Seite richtet sich an alle, die daran interessiert sind, ein eigenes Plugin (einen unabhängigen Datensammler) für Zabbix Agent 2 zu entwickeln.

Ein benutzerdefiniertes Plugin erweitert die Funktionalität von Zabbix Agent 2 über die bereitgestellten integrierten Plugins und ladbaren Plugins hinaus.

Jedes Plugin ist ein *Go*-Paket, das die Struktur definiert und eine oder mehrere Schnittstellen (*Exporter*, *Configurator*, *Runner*) implementiert. Weitere Informationen finden Sie unter Plugin-Schnittstellen und Verbindungsdiagramm.

Diese Anleitung unterstützt Sie bei der Erstellung eines benutzerdefinierten ladbaren Plugins.

Weitere Hinweise finden Sie auch in den Repositorys für:

- Beispiel-Plugin
- Vorhandene ladbare Plugins

Was Sie erstellen werden Dies ist eine Schritt-für-Schritt-Anleitung zum Erstellen eines einfachen ladbaren Plugins mit dem Namen **MyIP**. Dieses Plugin implementiert einen einzelnen Datenpunktschlüssel (`myip`), der die externe IP-Adresse des Hosts des Zabbix Agent zurückgibt.

Schritt 1: Einrichtung

1. Ein Plugin ist ein standardmäßiges Go-Modul. Beginnen Sie damit, die Datei `go.mod` im Plugin-Verzeichnis zu initialisieren, um Plugin-Abhängigkeiten zu verwalten:

```
cd path/to/plugins/myip # Wechseln Sie in Ihr Plugin-Verzeichnis
go mod init myip
```

2. Installieren Sie die erforderliche Abhängigkeit Zabbix Go SDK (`golang.zabbix.com/sdk`):

```
go get golang.zabbix.com/sdk@$LATEST_COMMIT_HASH
```

Ersetzen Sie `$LATEST_COMMIT_HASH` durch den neuesten `HEAD`-Commit-Hash aus dem repository `golang.zabbix.com/sdk` im entsprechenden Release-Branch. Zum Beispiel:

```
go get golang.zabbix.com/sdk@af85407
```

Beachten Sie, dass die Versionierung von `golang.zabbix.com/sdk` derzeit nicht unterstützt wird, sich dies jedoch in Zukunft ändern kann.

Zusätzliche Abhängigkeiten können bei Bedarf mit `go get` installiert werden.

3. Erstellen Sie eine leere Datei `main.go` für den Plugin-Quellcode:

```
touch main.go
```

Damit ist die anfängliche Einrichtung abgeschlossen, und das Plugin ist bereit für die Entwicklung.

Schritt 2: Plugin-Struktur Das im vorherigen Schritt installierte Modul `golang.zabbix.com/sdk` stellt das erforderliche Framework für die Plugin-Entwicklung bereit und sorgt dafür, dass alle Plugins eine einheitliche Struktur haben.

1. Richten Sie den grundlegenden Ausführungsablauf ein.

Definieren Sie zunächst den grundlegenden Ausführungsablauf des Plugins. Fügen Sie den folgenden Code zu `main.go` hinzu:

```
package main

func main() {
    err := run()
    if err != nil {
        panic(err)
    }
}

func run() error {
    return nil
}
```

Damit wird der grundlegende Ausführungsablauf für das Plugin eingerichtet. Die Funktion `run` wird später die Kernlogik des Plugins enthalten.

2. Untersuchen Sie die Plugin-Schnittstellen.

Ein Zabbix Agent 2-Plugin muss durch eine Struktur dargestellt werden, die Schnittstellen aus dem Paket `golang.zabbix.com/sdk/plugin` implementiert:

- *Accessor* – definiert grundlegende Methoden, die alle Plugins implementieren müssen, z. B. das Festlegen des Plugin-Namens und die Verarbeitung von Zeitüberschreitungen für Datenpunktschlüssel.
- Eine oder mehrere der folgenden funktionalen Plugin-Schnittstellen:
  - *Exporter* – führt eine Abfrage aus und gibt einen oder mehrere Werte, nichts oder einen Fehler zurück; wird häufig zusammen mit der Schnittstelle *Collector* verwendet.
  - *Collector* – verwaltet die regelmäßige Erfassung von Daten.
  - *Runner* – definiert die Start- und Beendigungsverfahren des Plugins.
  - *Watcher* – ermöglicht eine unabhängige Abfrage von Metriken unter Umgehung des internen Schedulers des Agents; nützlich für trapbasierte oder ereignisgesteuerte Überwachung.
  - *Configurator* – definiert, wie das Plugin seine Konfigurationseinstellungen liest und anwendet.

Sie können diese Schnittstellen selbst implementieren oder die vom Zabbix Go SDK bereitgestellten Standardimplementierungen verwenden und bei Bedarf anpassen. In diesem Tutorial werden die Standardimplementierungen verwendet.

3. Erstellen Sie die Plugin-Struktur.

Importieren Sie nun das Paket *plugin* und erstellen Sie eine Struktur `myIP`, die die Struktur `plugin.Base` einbettet:

```
import "golang.zabbix.com/sdk/plugin"

type myIP struct {
    plugin.Base
}
```

Die Struktur `myIP` erfüllt derzeit die Schnittstelle *Accessor*. Eine Methode zur Implementierung einer der funktionalen Plugin-Schnittstellen, *Exporter*, wird später in diesem Tutorial hinzugefügt.

Schritt 3: Datenpunktschlüssel definieren Ihr Plugin benötigt die Datenpunktschlüssel, um Daten zu erfassen und sie an den Zabbix Server oder Proxy bereitzustellen.

1. Importieren Sie das Paket *errs* für die Fehlerbehandlung:

```
import "golang.zabbix.com/sdk/errs"
```

2. Registrieren Sie Datenpunktschlüssel mit der Funktion `plugin.RegisterMetrics()` innerhalb der Funktion `run()`:

```
func run() error {
    p := &myIP{}

    // Register the `myip` item key.
    err := plugin.RegisterMetrics(
        p,
        "MyIP",                          // Plugin name
        "myip",                          // Item key name
        "Returns the host's IP address.", // Item key description
    )
    if err != nil {
        return errs.Wrap(err, "failed to register metrics")
    }

    return nil
}
```

Um mehrere Datenpunktschlüssel zu registrieren, wiederholen Sie die Parameter *Metrikname* und *Beschreibung* für jede Metrik. Zum Beispiel:

```
plugin.RegisterMetrics(&impl, "Myip", "metric.one", "Metric one description.", "metric.two", "Metric two d
```

Schritt 4: Den Handler einrichten Der Handler erleichtert die Kommunikation zwischen dem Agent und dem Plugin.

1. Importieren Sie das Paket *container*:

```
import "golang.zabbix.com/sdk/plugin/container"
```

2. Fügen Sie in der Funktion `run()` Code hinzu, um einen Handler zu erstellen und einzurichten:

```
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
}
```

Schritt 5: Datenerfassung implementieren Die Datenerfassung erfolgt über die Exporter-Schnittstelle, die die Methode Export beschreibt:

```
func Export(
  key string,             // Der zu erfassende Datenpunktschlüssel.
  params []string,        // An den Datenpunktschlüssel übergebene Argumente (`myip[arg1, arg2]`).
  context ContextProvider // Metadaten zur Datenerfassung des Datenpunktschlüssels.
) (any, error)
```

1. Importieren Sie die erforderlichen Pakete für HTTP-Anfragen und das Lesen von Antworten:

```
import (
    "io"
    "net/http"
)
```

2. Implementieren Sie die Methode Export für die Struktur myIP:

```
func (p *myIP) Export(
    key string, params []string, context plugin.ContextProvider,
) (any, error) {
    // Das Plugin kann abhängig vom Parameter `key` unterschiedliche Logik zur Datenerfassung verwenden.
    // Diese Implementierung prüft nur, ob der angegebene `key` unterstützt wird.
    if key != "myip" {
        return nil, errs.Errorf("unknown item key %q", key)
    }

    // Das Protokoll wird an das Agent-2-Protokoll weitergeleitet.
    p.Infof(
        "received request to handle %q key with %d parameters",
        key,
        len(params),
    )
```

```
    // Erfassen Sie die Daten und geben Sie sie zurück.

    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
}
```

Schritt 6: Das Plugin erstellen und konfigurieren

1. Um das Plugin zu erstellen, führen Sie aus:

```
go mod tidy
go build
```

Dadurch sollte im aktuellen Verzeichnis ein ausführbares `myip` erstellt werden.

2. Konfigurieren Sie Zabbix Agent 2 so, dass das Plugin verwendet wird:

```
echo "Plugins.MyIP.System.Path=$PATH_TO_THE_MYIP_PLUGIN_EXECUTABLE" > /etc/zabbix/zabbix_agent2.d/plugins.
```

Ersetzen Sie `$PATH_TO_THE_MYIP_PLUGIN_EXECUTABLE` durch den Pfad zu dem in Schritt 5 erstellten `myip`.

Der Plugin-Name im Namen des Konfigurationsparameters (*MyIP* in diesem Tutorial) muss mit dem in der Funktion *plugin.RegisterMetrics()* definierten Plugin-Namen übereinstimmen.

3. Um das Plugin und seinen `myip`-Datenpunkt zu testen, führen Sie aus:

```
zabbix_agent2 -c /etc/zabbix/zabbix_agent2.conf -t myip
```

Die Ausgabe sollte eine externe IP-Adresse Ihres Hosts enthalten und etwa so aussehen:

```
myip                                            [s|192.0.2.1]
```

Damit haben Sie ein einfaches ladbares Plugin für Zabbix Agent 2 erstellt. Glückwunsch!

```
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{}

    // Registriert den `myip`-Datenpunktschlüssel.
    err := plugin.RegisterMetrics(
        p,
        "MyIP",                              // Plugin-Name
        "myip",                              // Name des Datenpunktschlüssels
        "Gibt die IP-Adresse des Hosts zurück.", // Beschreibung des Datenpunktschlüssels
    )
    if err != nil {
        return errs.Wrap(err, "failed to register metrics")
    }

    // Erstellt einen neuen Handler.
    h, err := container.NewHandler("MyIP") // Plugin-Name
    if err != nil {
        return errs.Wrap(err, "failed to create new handler")
    }

    // Richtet die Protokollierung so ein, dass Protokolle vom Plugin an den Agent weitergeleitet werden.
    // Verfügbar über p.Logger.Infof, p.Logger.Debugf usw.
    p.Logger = h

    // Startet die Plugin-Ausführung.
    // Blockiert, bis eine Beendigungsanforderung vom Agent empfangen wird.
    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) {
    // Das Plugin kann abhängig vom Parameter `key` unterschiedliche Datenerfassungslogik verwenden.
    // Diese Implementierung prüft nur, ob der angegebene `key` unterstützt wird.
    if key != "myip" {
        return nil, errs.Errorf("unknown item key %q", key)
    }

    // Das Protokoll wird an Agent 2 weitergeleitet.
    p.Infof(
        "received request to handle %q key with %d parameters",
        key,
        len(params),
    )

    // Erfasst die Daten und gibt sie zurück.

    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
}
```

Vollständiger Quellcode

Plugin-Schnittstellen

Dieser Abschnitt beschreibt die verfügbaren Plugin-Schnittstellen.

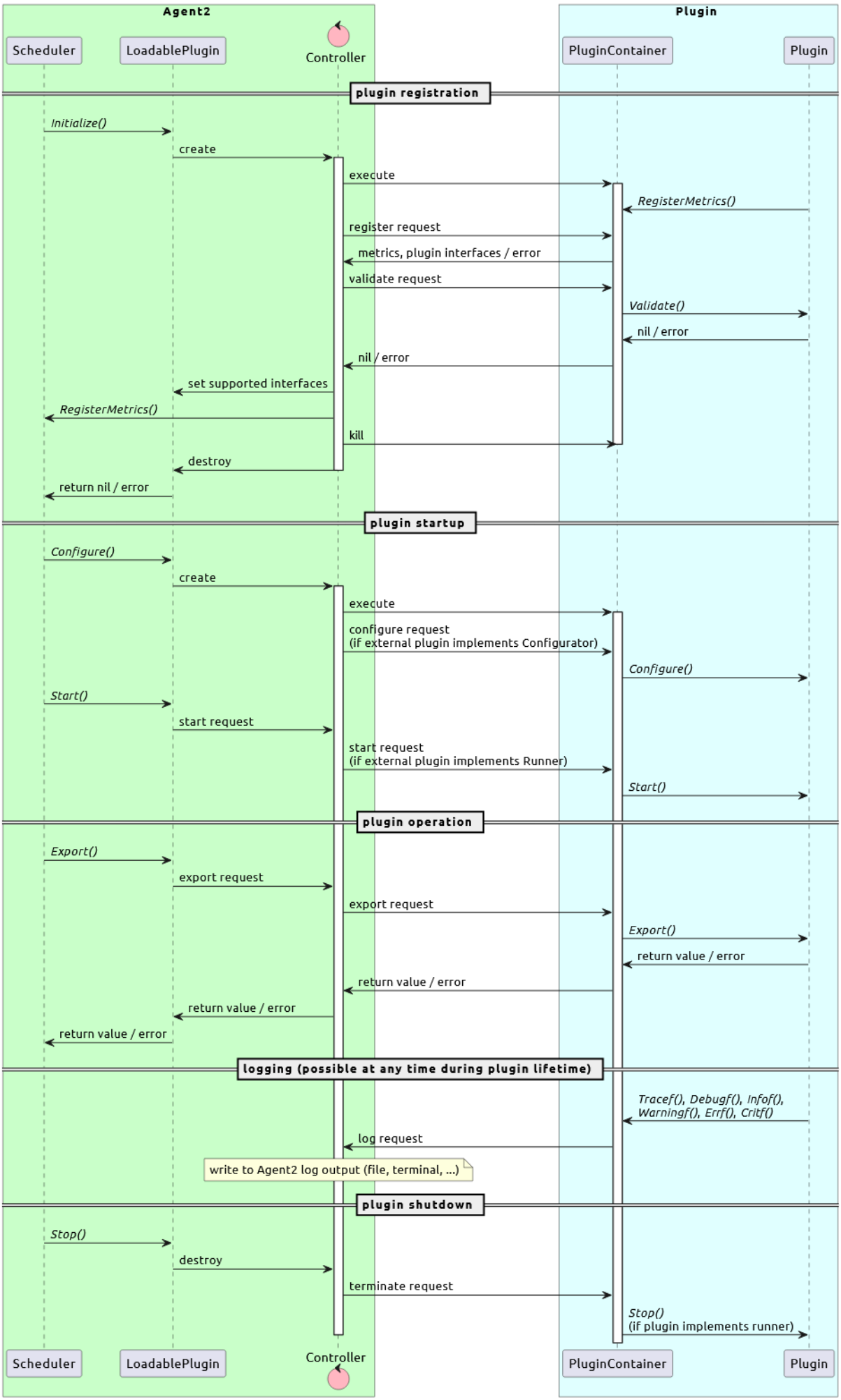
plugin.Exporter *Exporter* ist die einfachste Schnittstelle, die eine Abfrage durchführt und einen Wert (Werte), nichts oder einen Fehler zurückgibt. Sie akzeptiert einen vorbereiteten Elementschlüssel, Parameter und Kontext. Der Zugriff auf alle anderen Plugin-Schnittstellen ist exklusiv und keine Methode kann aufgerufen werden, wenn ein Plugin bereits eine Aufgabe ausführt. Außerdem gibt es eine Grenze von maximal 100 gleichzeitigen *Export()* Aufrufen pro Plugin, die je nach den Anforderungen für jedes Plugin reduziert werden kann.

plugin.Configurator Das *Configurator*-Interface stellt Plugin-Konfigurationsparameter von Zabbix Agent 2 Konfigurationsdateien bereit.

plugin.Runner Das *Runner*-Interface bietet die Mittel um die Initialisierung durchzuführen wenn ein Plugin gestartet (aktiviert) wird und um die Deinitialisierung durchzuführen wenn ein Plugin gestoppt (deaktiviert) wird. Zum Beispiel kann ein Plugin eine Hintergrund-Aufgabe *goroutine* durch die Implementierung des Runner-Iinterfaces starten/stoppen.

Verbindungsdiagramm Zabbix Agent 2 verbindet sich bidirektional mit den Plugins, wobei unter Linux UNIX-Sockets und unter Windows Named Pipes verwendet werden.

Das folgende Verbindungsdiagramm veranschaulicht den Kommunikationsprozess zwischen Zabbix Agent 2 und einem ladbaren Plugin sowie den Prozess der Metrikerfassung.


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

Python-Bibliothek für Zabbix

zabbix_utils ist die offizielle Python-Bibliothek für Zabbix.

Mit zabbix_utils können Sie:

- die Zabbix-API verwenden, um Zabbix-Objekte zu verwalten (Hosts erstellen, Datenpunkte aktualisieren, Ereignisse abrufen usw.).
- Daten vom Zabbix Agent erfassen (ähnlich wie mit Zabbix get).
- Daten an den Zabbix Server oder Proxy senden (ähnlich wie mit Zabbix sender).

Die Bibliothek unterstützt sowohl die synchrone als auch die asynchrone Ausführung von Skripten und kann dadurch sowohl einfache Workflows als auch parallele Operationen in großem Maßstab verarbeiten, etwa die Massendatenerfassung, Bulk-Exporte, die Verarbeitung von Warnmeldungen und eine effiziente Systemskalierung.

Die Bibliothek ist mit unterstützten Versionen des Zabbix Server und modernen Python-Versionen kompatibel.

Um zu beginnen, installieren Sie die Bibliothek und folgen Sie dann der Anleitung Schnellstart.

Installation

zabbix_utils kann mit jeder der unten beschriebenen Methoden installiert werden.

Voraussetzungen:

- Zabbix 6.0 oder höher (getestet mit 6.0, 7.0, 7.2, 7.4)
- Python 3.8 oder höher (getestet mit 3.8–3.13)
- aiohttp asynchrones HTTP-Framework (nur für den asynchronen Modus erforderlich)

Python Package Index (PyPI)

Dies ist die gängigste Methode für die meisten Umgebungen:

```
pip install zabbix_utils
pip install zabbix_utils[async] # Nur für async-Unterstützung
```

Zabbix-Repository

Verwenden Sie diese Methode, wenn Sie Abhängigkeiten lieber über den Paketmanager Ihres Systems verwalten möchten.

Laden Sie die Zabbix-Pakete für Ihre Linux-Distribution herunter und führen Sie dann die folgenden Befehle aus.

Unter RHEL und Derivaten:

```
dnf install python3-zabbix-utils
dnf install epel-release # Nur für Async-Unterstützung
dnf install python3-aiohttp # Nur für Async-Unterstützung
```

Unter Debian/Ubuntu und Derivaten:

```
apt install python3-zabbix-utils
apt install python3-aiohttp # Nur für Async-Unterstützung
```

Quelle (GitHub)

Verwenden Sie diese Methode, wenn Sie die neueste Entwicklungsversion bevorzugen:

```
git clone https://github.com/zabbix/python-zabbix-utils
cd python-zabbix-utils/
pip install -r requirements.txt # Nur für asynchrone Unterstützung
python3 setup.py install
```

Um die für den asynchronen Modus erforderlichen Abhängigkeiten zu installieren, können Sie auch eine der oben beschriebenen Methoden verwenden.

Schnellstartanleitung

Nachdem Sie zabbix_utils installiert haben, können Sie es in Ihrem Skript verwenden.

Diese Schnellstartanleitung zeigt Ihnen, wie Sie:

- den Hostnamen vom Zabbix Agent abrufen.
- einen Host und einen Trapper-Datenpunkt mit der Zabbix API erstellen.
- einen Wert an den Datenpunkt senden.

Die Anleitung stellt das Skript Schritt für Schritt vor und erklärt jeden Teil, sobald er eingeführt wird. Das vollständige Skript wird am Ende der Seite bereitgestellt.

Die Anleitung geht außerdem davon aus, dass Ihr Zabbix Server, Agent und die API lokal ausgeführt werden.

Hostnamen vom Zabbix Agent abrufen

Beginnen Sie damit, den Hostnamen des Systems abzurufen, auf dem der Zabbix Agent ausgeführt wird. Sie benötigen diesen Hostnamen, um mithilfe der Zabbix API einen Host zu erstellen.

1. Importieren Sie die Klasse `Getter` aus zabbix_utils. Diese Klasse funktioniert wie Zabbix get und ermöglicht es Ihnen, Daten vom Zabbix Agent anzufordern.

2. Erstellen Sie eine `Getter`-Instanz, die eine Verbindung zum lokalen Zabbix Agent unter `127.0.0.1` auf Port 10050 herstellt.

3. Rufen Sie die Methode `get()` für die `Getter`-Instanz auf, um den Hostnamen vom Zabbix Agent anzufordern.

- Die Methode `get()` verwendet einen Datenpunktschlüssel als Parameter und sendet eine Anfrage an den Zabbix Agent für diesen Datenpunkt.
- Die Methode `get()` gibt ein Objekt zurück, und der Hostname ist im Attribut `value` dieses Objekts gespeichert.
- `system.hostname` ist ein integrierter Schlüssel des Zabbix Agent, der den Namen des Hosts zurückgibt.

```
from zabbix_utils import Getter

agent = Getter(host='127.0.0.1', port=10050)
hostname = agent.get('system.hostname').value
```

Mit der Zabbix-API verbinden und anmelden

Stellen Sie als Nächstes eine Verbindung zur Zabbix-API her. Dadurch kann Ihr Skript Zabbix-Objekte wie Hosts und Datenpunkte verwalten.

1. Importieren Sie die Klasse `ZabbixAPI` aus zabbix_utils.

2. Erstellen Sie eine `ZabbixAPI`-Instanz und geben Sie die URL Ihres Zabbix-Web-Interface an.

3. Rufen Sie die Methode `login()` für die `ZabbixAPI`-Instanz auf und geben Sie Ihren Benutzernamen und Ihr Passwort an. Das Konto muss über die Berechtigung verfügen, auf die Zabbix-API zuzugreifen.

4. Rufen Sie die Methode `logout()` auf, um die Sitzung nach den API-Operationen zu schließen.

```
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()
```

Host in Zabbix erstellen

Nachdem Sie nun den Hostnamen vom Zabbix Agent erhalten haben und mit der Zabbix API verbunden sind, können Sie einen neuen Host in Zabbix erstellen.

1. Rufen Sie die API-Methode `host.create()` auf der `ZabbixAPI`-Instanz auf und geben Sie die Host-Details an:

- `host` - auf die Variable `hostname` setzen, die den Hostnamen enthält, den Sie vom Zabbix Agent abgerufen haben.
- `interfaces` - enthält die Verbindungsdetails für den Zabbix Agent, der auf dem Host ausgeführt wird.
- `groups` - enthält mindestens eine Hostgruppe, zu der der Host gehören soll.

2. Da die Zabbix API die ID des neu erstellten Hosts zurückgibt, speichern Sie diese ID zur späteren Verwendung in der Variable `host_id`.

3. Geben Sie eine Meldung aus, um zu bestätigen, dass der Host erstellt wurde.

```
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}")
```

Neuen Datenpunkt in Zabbix erstellen

Nachdem Sie einen Host erstellt haben, können Sie ihm einen Datenpunkt hinzufügen.

1. Definieren Sie einen eindeutigen Schlüssel für den zu erstellenden Datenpunkt.

2. Rufen Sie die Methode `item.create()` auf der `ZabbixAPI`-Instanz auf und geben Sie die Details des Datenpunkts an:

- `hostid` - auf die Variable `host_id` setzen, die die ID des soeben erstellten Hosts enthält.
- `name` - ein Name für den Datenpunkt.
- `key_` - auf die Variable `item_key` setzen, die Sie gerade definiert haben.
- `type` - auf 2 setzen (Trapper-Datenpunkt); erforderlich, um Werte zu empfangen, die in den folgenden Schritten von Ihrem Skript gesendet werden.
- `value_type` - auf 3 setzen (numerisch ohne Vorzeichen), der Datentyp, den dieser Datenpunkt speichert.

3. Da die Zabbix API die ID des neu erstellten Datenpunkts zurückgibt, speichern Sie diese ID zur späteren Verwendung in der Variable `item_id`.

4. Geben Sie eine Meldung aus, um zu bestätigen, dass der Datenpunkt erstellt wurde.

```
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}")
```

Wert an Host senden

Nachdem Sie Ihren Host und Datenpunkt erstellt haben, können Sie Daten dorthin senden.

1. Importieren Sie die Klasse `time` und warten Sie einige Sekunden, bevor Sie die Daten senden. Dadurch wird sichergestellt, dass Zabbix Ihren neuen Host und Datenpunkt vollständig verarbeitet hat.

2. Importieren Sie die Klasse `Sender` aus zabbix_utils. Diese Klasse funktioniert wie Zabbix sender und ermöglicht es Ihrem Skript, Daten an Zabbix zu senden.

3. Erstellen Sie eine `Sender`-Instanz, die sich mit dem lokalen Zabbix Server unter `127.0.0.1` auf Port 10051 verbindet.

4. Rufen Sie die Methode `send_value()` für die `Sender`-Instanz auf, um einen Wert an den Datenpunkt Ihres Hosts zu senden, und geben Sie dabei folgende Details an:

- `hostname` - auf die Variable `hostname` setzen, die den Hostnamen enthält, den Sie vom Zabbix Agent abgerufen haben.
- `item_key` - auf die Variable `item_key` setzen, die Sie zuvor in Ihrem Skript definiert haben.
- `1` - der zu sendende Wert.

4. Geben Sie eine Meldung aus, um zu bestätigen, dass der Wert gesendet wurde.

```
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}")
```

Nachdem dieser Code erfolgreich ausgeführt wurde, sollten Sie eine Erfolgsmeldung sehen, die angibt, dass Zabbix Ihren Wert empfangen hat:

```
Sender response: {"processed": 1, "failed": 0, "total": 1, "time": "0.000151", "chunk": 1}
```

Nun können Sie auch in Ihrer Zabbix-Weboberfläche (*Monitoring* > *Aktuelle Daten*) nachsehen, um den Wert zu sehen.

Vollständiges Skript

Nachfolgend finden Sie das vollständige Skript, das alle Schritte kombiniert: Abrufen des Hostnamens, Erstellen des Hosts und des Datenpunkts mithilfe der Zabbix API sowie Senden von Daten an Zabbix.

```
import time
from zabbix_utils import Getter, ZabbixAPI, Sender

### Hostnamen vom Zabbix Agent abrufen
agent = Getter(host='127.0.0.1', port=10050)
hostname = agent.get('system.hostname').value

### Verbindung zur Zabbix API herstellen und anmelden
api = ZabbixAPI(url='127.0.0.1/zabbix')
api.login(user='Admin', password='zabbix')

### Host in Zabbix erstellen
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"Host '{hostname}' created with ID {host_id}")

### Datenpunkt in Zabbix erstellen
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}")

### Von der API abmelden
api.logout()

### Warten, bis Zabbix den neuen Host verarbeitet hat
time.sleep(10)

### Wert an den Host senden
sender = Sender(server='127.0.0.1', port=10051)
response = sender.send_value(hostname, item_key, 1)

print(f"Sender response: {response}")
```

Zabbix API verwenden

zabbix_utils ermöglicht Ihnen die Verwendung der Zabbix API zur Verwaltung von Zabbix-Objekten, einschließlich des Erstellens von Hosts, des Aktualisierens von Datenpunkten, des Abrufens von Ereignissen und mehr.

API-Anfragen können im synchronen oder asynchronen Modus ausgeführt werden:

- Im synchronen Modus sendet Ihr Python-Skript eine Anfrage und wartet auf eine Antwort, bevor es fortfährt. Dies eignet sich für einfache, sequenzielle und vorhersehbare Vorgänge.
- Im asynchronen Modus sendet das Skript Anfragen, ohne auf jede einzelne Antwort zu warten, sodass andere Vorgänge parallel fortgesetzt werden können; dies ist effizienter bei langsamen Anfragen oder großen Datenmengen.

Die Beispiele auf dieser Seite konzentrieren sich auf den synchronen Modus, obwohl der asynchrone Modus ähnlichen Mustern folgt. Zusätzliche Beispiele sind im GitHub-Repository zabbix_utils verfügbar.

Import

Um zabbix_utils für die Arbeit mit der Zabbix-API zu verwenden, importieren Sie die Klasse `ZabbixAPI` in Ihr Skript:

```
from zabbix_utils import ZabbixAPI
```

> **Note:**
> Wenn Sie bereits eine Python-Bibliothek aus der Community verwenden, können Sie diesen Import in der Regel durch `ZabbixAPI` aus zabbix_utils ersetzen.

Anmelden

Bevor Sie API-Anfragen stellen, müssen Sie eine `ZabbixAPI`-Instanz erstellen und sich bei der Zabbix API anmelden.

Wählen Sie die Methode, die am besten dazu passt, wie Sie Anmeldedaten verwalten. Sie können Ihren Benutzernamen und Ihr Passwort oder API-Tokens verwenden.

Während der Initialisierung

Beim Erstellen einer `ZabbixAPI`-Instanz können Sie die URL der Zabbix-Weboberfläche und Ihre Zugangsdaten auf einmal angeben:

```
### Benutzername und Passwort:
api = ZabbixAPI(url="127.0.0.1/zabbix", user="Admin", password="zabbix")

### API-Token:
api = ZabbixAPI(url="127.0.0.1/zabbix", token="your_api_token")
```

Wenn Sie Verbindungsparameter lieber gruppieren möchten, können Sie sie als Python-Dictionary übergeben und mithilfe der Python-Syntax für Schlüsselwortargumente entpacken:

```
ZABBIX_AUTH = {
    "url": "127.0.0.1/zabbix",
```

```
    "user": "Admin",
    "password": "zabbix"
}

api = ZabbixAPI(**ZABBIX_AUTH)
```

Verwendung von `login()`

Sie können zunächst eine ZabbixAPI-Instanz nur mit der URL der Zabbix-Weboberfläche erstellen und später die Methode `login()` mit Ihren Zugangsdaten verwenden:

```
### Benutzername und Passwort:
api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(user="Admin", password="zabbix")

### API-Token:
api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(token="your_api_token")
```

Verwendung von Umgebungsvariablen

Sie können Ihre Zugangsdaten zunächst als Umgebungsvariablen speichern, zum Beispiel über ein Terminal:

```
### Benutzername und Passwort:
export ZABBIX_USER="Admin"
export ZABBIX_PASSWORD="zabbix"

### Token:
export ZABBIX_TOKEN="your_api_token"
```

Wenn Sie dann Ihr Skript aus derselben Terminal-Sitzung starten, benötigt die ZabbixAPI-Instanz im Skript nur die URL der Zabbix-Weboberfläche:

```
api = ZabbixAPI(url="127.0.0.1/zabbix")
```

Sie können die URL auch in einer Umgebungsvariablen speichern:

```
### Benutzername und Passwort:
export ZABBIX_USER="Admin"
export ZABBIX_PASSWORD="zabbix"
export ZABBIX_URL="https://127.0.0.1/zabbix"

### Token:
export ZABBIX_TOKEN="your_api_token"
export ZABBIX_URL="https://127.0.0.1/zabbix"
```

Wenn URL und Zugangsdaten vollständig als Umgebungsvariablen gesetzt sind, benötigt die ZabbixAPI-Instanz im Skript überhaupt keine Argumente:

```
api = ZabbixAPI()
```

Abmelden

Wenn Sie sich mit einem Benutzernamen und Passwort angemeldet haben, rufen Sie nach Abschluss Ihrer API-Operationen die Methode `logout()` auf:

```
from zabbix_utils import ZabbixAPI

api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(user="Admin", password="zabbix")

### API-Aktionen kommen hier hin

api.logout()
```

Note:
Beim Verwenden von API-Tokens ist das Aufrufen der Methode `logout()` nicht erforderlich.

API-Anfragen

Nachdem Sie sich angemeldet haben, können Sie beliebige API-Anfragen stellen, indem Sie Methoden aufrufen, die in der Zabbix-API-Methodenreferenz beschrieben sind.

API-Methoden werden im folgenden Format aufgerufen:

```
api_instance.zabbix_object.method(parameters)
```

Note:
Einige Methoden- oder Objektnamen der Zabbix-API verwenden Wörter, die in Python reservierte Schlüsselwörter sind (z. B. `import`). Um Python-Fehler zu vermeiden, fügen Sie beim Verwenden in Python einen Unterstrich an den Methoden- oder Objektnamen an (z. B. `api.configuration.import_`).

Zum Beispiel, um mit `host.get` eine Liste von Hosts anzuzeigen:

```
### 1. Import ZabbixAPI from zabbix_utils:
from zabbix_utils import ZabbixAPI

### 2. Create the ZabbixAPI instance and log in:
api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(user="Admin", password="zabbix")

### 3. Retrieve hosts matching a filter:
hosts = api.host.get(
    filter={
        "host": [
            "Zabbix server",
            "Linux server"
        ]
    }
)

### 4. Iterate over returned hosts and print each host's details:
for host in hosts:
    print(host)

### 5. Log out from the API to close the session:
api.logout()
```

Asynchroner Modus

Der asynchrone Modus ermöglicht es Ihrem Skript, API-Anfragen zu senden, ohne auf den Abschluss jeder einzelnen Anfrage zu warten. Dadurch kann Ihr Skript effizienter werden, wenn es viele API-Anfragen ausführen muss oder wenn einige Anfragen lange dauern.

Bei der Verwendung des asynchronen Modus gibt es im Vergleich zum synchronen Modus wichtige Unterschiede:

- Importieren Sie das Python-Modul `asyncio` (zuvor müssen Sie die erforderlichen Abhängigkeiten installieren).
- Importieren Sie `AsyncZabbixAPI` anstelle von `ZabbixAPI`.
- Schreiben Sie Ihren Code innerhalb einer `async`-Funktion.
- Verwenden Sie `await` beim Aufruf von API-Methoden, einschließlich `login()` und `logout()`.

Attention:
Beim Erstellen einer `AsyncZabbixAPI`-Instanz können Sie sich nicht während der Initialisierung anmelden.

Zum Beispiel, um im asynchronen Modus mit `host.get` eine Liste von Hosts anzuzeigen:

```
### 1. Importieren Sie asyncio für den asynchronen Modus und AsyncZabbixAPI aus zabbix_utils:
import asyncio
from zabbix_utils import AsyncZabbixAPI

### 2. Definieren Sie die Haupt-async-Funktion, in der alle API-Operationen ausgeführt werden:
async def main():

    # 3. Erstellen Sie eine AsyncZabbixAPI-Instanz und melden Sie sich an (await ist erforderlich):
```

```
    api = AsyncZabbixAPI(url="127.0.0.1")
    await api.login(user="User", password="zabbix")

    # 4. Rufen Sie Hosts ab, die einem Filter entsprechen (await ist erforderlich):
    hosts = await api.host.get(
        filter={
            "host": [
                "Zabbix server",
                "Linux server"
            ]
        }
    )

    # 5. Iterieren Sie über die zurückgegebenen Hosts und geben Sie die Details jedes Hosts aus:
    for host in hosts:
        print(host)

    # 6. Melden Sie sich von der API ab, um die Sitzung zu schließen (await ist erforderlich):
    await api.logout()

### 7. Führen Sie die async-Hauptfunktion main() mit der Ereignisschleife von asyncio aus:
asyncio.run(main())
```

Beispiele

Die folgenden Beispiele zeigen häufige Aufgaben mit der Zabbix-API unter Verwendung der Bibliothek zabbix_utils.

Weitere Beispiele sind im Verzeichnis `examples` des GitHub-Repositorys zabbix_utils verfügbar.

Daten vom Zabbix Agent erfassen

Überblick

zabbix_utils ermöglicht es Ihnen, Daten vom Zabbix Agent zu erfassen (ähnlich wie bei Zabbix get).

Daten können im synchronen oder asynchronen Modus erfasst werden:

- Im synchronen Modus fordert Ihr Python-Skript Daten an und wartet vor dem Fortfahren auf die Antwort, was sich für einfache, sequenzielle und vorhersehbare Vorgänge eignet.
- Im asynchronen Modus fordert das Skript Daten an, ohne auf jede einzelne Antwort zu warten, sodass andere Vorgänge parallel fortgesetzt werden können; dies ist bei langsamen Anfragen oder großen Datenmengen effizienter.

Die Beispiele auf dieser Seite konzentrieren sich auf den synchronen Modus, obwohl der asynchrone Modus ähnlichen Mustern folgt. Zusätzliche Beispiele sind im GitHub-Repository zabbix_utils verfügbar.

Import

Um zabbix_utils zum Sammeln von Datenpunkt-Werten zu verwenden, importieren Sie die Klasse `Getter` in Ihr Python-Skript:

```
from zabbix_utils import Getter
```

Anfragedaten

Um einen Datenpunktwert anzufordern:

1. Erstellen Sie eine `Getter`-Instanz und geben Sie die IP-Adresse und den Port Ihres Zabbix Agent an.
2. Rufen Sie die Methode `get()` für die `Getter`-Instanz auf und geben Sie den Schlüssel des Datenpunkts an, den Sie abrufen möchten.

Um beispielsweise Daten für den Datenpunkt `system.uname` anzufordern:

```
agent = Getter(host='192.0.2.1', port=10050)
response = agent.get('system.uname')
```

Verwendung einer nicht standardmäßigen IP

Wenn der Server, auf dem Ihr Skript ausgeführt wird, mehrere IP-Adressen hat, können Sie für den `Getter` eine `source_ip` angeben, die beim Verbinden mit dem Zabbix Agent verwendet werden soll:

```
agent = Getter(
    host='192.0.2.1',
    port=10050,
    source_ip='10.10.7.1'
)
```

Verwenden von Timeout

Sie können für den `Getter` ein Antwort-`timeout` festlegen, um zu steuern, wie lange Ihr Skript auf eine Antwort vom Zabbix Agent warten soll, bevor es aufgibt:

```
agent = Getter(
    host='192.0.2.1',
    port=10050,
    timeout=30
)
```

Verwendung von Verschlüsselung

Der `Getter` enthält keine integrierte Unterstützung für Verschlüsselung, aber Sie können diese bereitstellen, indem Sie mithilfe von Bibliotheken von Drittanbietern einen Wrapper erstellen:

```
def psk_wrapper(sock, tls):
    # ...
    # Implementierung des TLS-PSK-Wrappers für den Socket
    # ...

agent = Getter(
    host='192.0.2.1',
    port=10050,
    socket_wrapper=psk_wrapper
)
```

Antwort

Die Antwort des Zabbix Agent wird von der Bibliothek verarbeitet und als `AgentResponse`-Objekt zurückgegeben:

```
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
```

Asynchroner Modus

Der asynchrone Modus ermöglicht es Ihrem Skript, Werte zu erfassen, ohne auf das Eintreffen jedes einzelnen Werts warten zu müssen. Dadurch kann Ihr Skript effizienter arbeiten, wenn es viele Werte erfassen muss oder wenn die Erfassung mancher Werte lange dauert.

Bei der Verwendung des asynchronen Modus gibt es wichtige Unterschiede im Vergleich zum synchronen Modus:

- Importieren Sie das Python-Modul `asyncio` (Sie müssen zuvor die erforderlichen Abhängigkeiten installieren).
- Importieren Sie `AsyncGetter` anstelle von `Getter`.
- Schreiben Sie Ihren Code innerhalb einer `async`-Funktion.
- Verwenden Sie `await`, wenn Sie die Methode `get()` aufrufen.

Zum Beispiel, um einen einzelnen Wert im asynchronen Modus zu erfassen:

```
### 1. Import asyncio for asynchronous mode, and AsyncGetter from zabbix_utils:
import asyncio
from zabbix_utils import AsyncGetter
```

```
### 2. Define the main async function where all data requests will be executed:
async def main():
    agent = AsyncGetter(host='192.0.2.1', port=10050)

    # 3. Fetch the system.uname value from Zabbix agent (must await):
    response = await agent.get('system.uname')

    # 4. Print the value returned by Zabbix agent:
    print(response.value)

### 5. Run the async main() function using asyncio's event loop:
asyncio.run(main())
```

Daten an Zabbix-Server oder -Proxy senden

Mit zabbix_utils können Sie Datenpunkt-Werte an einen Trapper-Datenpunkt auf dem Zabbix-Server oder -Proxy senden (ähnlich wie mit Zabbix sender).

Sie können einen einzelnen Wert, mehrere Werte oder sogar mehrere Zabbix-Cluster als Ziel angeben.

Daten können im synchronen oder asynchronen Modus gesendet werden:

- Im synchronen Modus sendet Ihr Python-Skript Werte und wartet auf eine Antwort, bevor es fortfährt; dies eignet sich für einfache, sequenzielle und vorhersehbare Vorgänge.
- Im asynchronen Modus sendet das Skript Werte, ohne auf jede Antwort zu warten, sodass andere Vorgänge parallel fortgesetzt werden können; dies ist bei langsamen Anfragen oder großen Datenmengen effizienter.

Die Beispiele auf dieser Seite konzentrieren sich auf den synchronen Modus, obwohl der asynchrone Modus einem ähnlichen Muster folgt. Zusätzliche Beispiele sind im GitHub-Repository zabbix_utils verfügbar.

Import

Um zabbix_utils zum Senden von Datenpunkt-Werten zu verwenden, importieren Sie die Klasse `Sender` in Ihr Skript:

```
from zabbix_utils import Sender
```

Zum Senden mehrerer Werte können Sie auch die Klasse `ItemValue` importieren:

```
from zabbix_utils import Sender, ItemValue
```

Einzelnen Wert senden

So senden Sie einen Datenpunktwert:

1. Erstellen Sie eine `Sender`-Instanz und geben Sie dabei die IP-Adresse und den Port Ihres Zabbix-Servers oder -Proxys an.
2. Rufen Sie die Methode `send_value()` der `Sender`-Instanz im folgenden Format auf:

```
sender_instance.send_value('host', 'item.key', 'value', optional_timestamp, optional_nanoseconds)
```

Um beispielsweise 1 an den Trapper-Datenpunkt `service.status` auf dem Host `Linux server` zu senden:

```
sender = Sender(server='127.0.0.1', port=10051)
response = sender.send_value('Linux server', 'service.status', 1)
```

Verwendung einer nicht standardmäßigen IP

Wenn der Server, auf dem Ihr Skript ausgeführt wird, mehrere IP-Adressen hat, können Sie eine `source_ip` angeben, die der `Sender` beim Senden von Werten an den Zabbix Server oder Proxy verwenden soll:

```
sender = Sender(
    server='127.0.0.1',
    port=10051,
    source_ip='10.10.7.1'
)
```

Timeout verwenden

Sie können für den `Sender` einen Antwort-`timeout` festlegen, um zu steuern, wie lange Ihr Skript auf eine Antwort vom Zabbix Server oder Proxy warten soll, bevor es aufgibt:

```
sender = Sender(
    server='127.0.0.1',
    port=10051,
    timeout=30
)
```

Agent-Konfigurationsdatei verwenden

Sie können zabbix_utils die Parameter Server oder ServerActive aus einer lokalen Zabbix-Agent- oder Agent-2-Konfigurationsdatei lesen lassen. In diesem Fall müssen Sie beim Erstellen einer Sender-Instanz keine Verbindungsparameter angeben:

```
sender = Sender(
    use_config=True,
    config_path='/etc/zabbix/zabbix_agent2.conf'
)
```

Attention:
Wenn ServerActive einen oder mehrere Zabbix-Cluster mit mehreren Serverinstanzen enthält, sendet Sender Daten an den ersten verfügbaren Server in jedem Cluster. Wenn ServerActive nicht festgelegt ist, wird die Adresse aus Server mit dem Standardport (10051) verwendet.

Verwendung von Verschlüsselung

Der Sender enthält keine integrierte Unterstützung für Verschlüsselung, aber Sie können diese durch die Erstellung eines Wrappers mit Bibliotheken von Drittanbietern bereitstellen:

```
def psk_wrapper(sock, tls):
    # ...
    # Implementierung eines TLS-PSK-Wrappers für den Socket
    # ...

sender = Sender(
    server='127.0.0.1',
    port=10051,
    socket_wrapper=psk_wrapper
)
```

Antwort für einen einzelnen Wert

Die vom Zabbix Server oder Proxy zurückgegebene Antwort wird von der Bibliothek verarbeitet und als TrapperResponse-Objekt zurückgegeben:

```
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
```

Asynchroner Modus

Der asynchrone Modus ermöglicht es Ihrem Python-Skript, Werte zu senden, ohne auf eine Antwort vom Zabbix Server oder Proxy zu warten. Dadurch kann Ihr Skript effizienter werden, wenn es viele Werte senden muss oder wenn das Senden einiger Werte lange dauert.

Bei der Verwendung des asynchronen Modus gibt es im Vergleich zum synchronen Modus einige wichtige Unterschiede:

- Importieren Sie das Python-Modul asyncio (Sie müssen zuerst die erforderlichen Abhängigkeiten installieren).
- Importieren Sie AsyncSender anstelle von Sender.
- Schreiben Sie Ihren Code innerhalb einer async-Funktion.
- Verwenden Sie await, wenn Sie die Methode send_value() aufrufen.

Zum Beispiel, um einen einzelnen Wert im asynchronen Modus zu senden:

```
### 1. Importieren Sie asyncio für den asynchronen Modus und AsyncSender aus zabbix_utils:
import asyncio
from zabbix_utils import AsyncSender

### 2. Definieren Sie die Hauptfunktion async, in der alle Datensendevorgänge (müssen await verwenden) aus
async def main():
    sender = AsyncSender(server='127.0.0.1', port=10051)
    response = await sender.send_value('Linux server', 'service.status', 1)

    # 3. Geben Sie die vom Zabbix Server oder Proxy zurückgegebene Antwort aus:
    print(response)

### 4. Führen Sie die asynchrone Funktion main() mit der Ereignisschleife von asyncio aus:
asyncio.run(main())
```

Mehrere Werte senden

Um mehrere Werte zu senden:

1. Bereiten Sie ein Array von `ItemValue`-Objekten vor, wobei jedes dasselbe Format wie die Methode `send_value()` verwendet.
2. Erstellen Sie eine `Sender`-Instanz und geben Sie dabei die IP-Adresse und den Port Ihres Zabbix-Server oder -Proxy an.
3. Rufen Sie für die `Sender`-Instanz die Methode `send()` auf (anstelle von `send_value()`) und geben Sie dabei das Array von Objekten mit den zu sendenden Werten an.

Zum Beispiel, um fünf Werte an verschiedene Hosts zu senden:

```
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)
```

Benutzerdefinierte Chunk-Größe verwenden

Wenn Sie mehr Werte senden müssen, als ein Trapper-Datenpunkt in einer einzelnen Anfrage akzeptieren kann, können Sie diese in Chunks aufteilen.

Standardmäßig beträgt die Chunk-Größe 250 Werte. Sie können sie ändern, indem Sie beim Erstellen einer `Sender`-Instanz den Parameter `chunk_size` festlegen.

Um beispielsweise fünf Werte in drei Chunks (2-2-1) zu senden, setzen Sie den Parameter `chunk_size` auf 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)
```

Werte an mehrere Zabbix-Cluster senden

Um Werte an mehrere Zabbix-Cluster zu senden:

1. Bereiten Sie ein Array von Zabbix-Clustern vor. Wenn ein Cluster mehrere Knoten hat, wird der Wert an den ersten **verfügbaren** Knoten jedes Clusters gesendet.
2. Erstellen Sie einen `Sender` und geben Sie dabei Ihr Array von Zabbix-Clustern an.

3. Rufen Sie die Methode `send_value()` für die Sender-Instanz auf und verwenden Sie dabei dasselbe Format wie bei der Methode `send_value()`.

Zum Beispiel, um einen Wert an den ersten verfügbaren Knoten in jedem Cluster zu senden:

```
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)
```

Antwort für mehrere Werte

Standardmäßig gibt Sender ein aggregiertes Ergebnis für das Senden von Werten über alle Hosts oder Cluster hinweg zurück:

```
print(response)
### {"processed": 2, "failed": 0, "total": 2, "time": "0.000108", "chunk": 2}
```

Wenn Sie detailliertere Informationen benötigen, können Sie die Ergebnisse für jeden Cluster und jeden Chunk über das Attribut `response.details` prüfen:

```
print(response)
### {"processed": 2, "failed": 0, "total": 2, "time": "0.000108", "chunk": 2}

if response.failed == 0:
    print(f"Wert erfolgreich in {response.time} gesendet")
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"{resp.processed} von {resp.total} verarbeitet bei {node.address}:{node.port}")
            # 1 von 1 verarbeitet bei 127.0.0.1:10051
            # 1 von 1 verarbeitet bei zabbix.example.local:10051
```

Debug-Protokollierung

Um Probleme bei der Arbeit mit der Zabbix API, beim Sammeln von Daten vom Zabbix Agent oder beim Senden von Daten an den Zabbix Server oder Proxy zu beheben, können Sie die Debug-Protokollierung aktivieren.

Die Bibliothek zabbix_utils verwendet das standardmäßige Python-Modul `logging`.

So zeigen Sie detaillierte Debug-Meldungen aus der Bibliothek an:

1. Importieren Sie das Python-Modul `logging` in Ihr Skript.
2. Konfigurieren Sie das Logging-System so, dass Meldungen in einem lesbaren Format ausgegeben werden, und setzen Sie die Logging-Stufe auf 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)
```

Wenn die Debug-Protokollierung aktiviert ist, werden detaillierte Informationen über Anfragen und Antworten ausgegeben, zum Beispiel:

```
[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
```

Module

Was ist ein PHP-Frontend-Modul?

- Ein Modul ist eine Einheit mit einer eindeutigen ID, einem Namen, einer Beschreibung, einem Autor und anderen Feldern, die in seiner Manifestdatei definiert sind, zusammen mit PHP, Javascript und anderen Dateien, die sich in einem einzigen Unterverzeichnis von */ui/modules* befinden.
- Ein Modul sollte einfachen Regeln entsprechen, um einen korrekten Betrieb zu gewährleisten.
- Ein Modul muss vom Administrator installiert (entpackt) und im Frontend aktiviert werden.

Wozu ein Modul verwendet werden kann

- Hinzufügen neuer Funktionen über benutzerdefinierte Frontend-Abschnitte;
- Erstellen von benutzerdefinierten Dashboard-Widget-Typen (siehe widget modules);
- Überschreiben oder Erweitern der bestehenden Funktionalität.

Wofür ein Modul nicht verwendet werden kann

- Die Registrierung einer neuen API-Methode oder die Änderung einer bestehenden Methode.

Wie Module funktionieren

- Ein aktiviertes Modul wird bei jeder HTTP-Anfrage gestartet, bevor der Aktionscode ausgeführt wird.
- Das Modul registriert neue Aktionen oder definiert die bestehenden um.
- Das Modul fügt neue Frontend-Abschnitte hinzu und entfernt oder definiert die vorhandenen neu.
- Falls erforderlich greift das Modul auf Frontend-Ereignisse wie onBeforeAction und onTerminate zu.
- Die angeforderte Aktion wird schließlich ausgeführt, indem der Aktionscode ausgeführt wird - entweder der Standardcode oder der vom Modul definierte.

Wie geht es weiter? Egal, ob Sie es vorziehen, zu lernen, indem Sie etwas tun, oder zuerst die Anleitungen lesen, diese Seiten enthalten alle Informationen und Schritte, die erforderlich sind, um Ihre eigenen Module zu erstellen:

- Schritt-für-Schritt-Tutorials zum Schreiben Ihres ersten Moduls
- Struktur der Moduldatei
- Besonderheiten der Widget-Module](/devel/modules/widgets)
- Modulbeispiele zur Wiederverwendung](/devel/modules/examples)

Struktur der Moduldateien

Der gesamte Code eines Moduls wird in einem einzigen Verzeichnis innerhalb von **zabbix/ui/modules/** gespeichert.

```
example_module_directory/          (erforderlich)
   manifest.json                   (erforderlich)  Metadaten und Action-Definition.
   Module.php                                      Modulinitialisierung und Event Handling.
   actions/                                        Action Controller Dateien.
       SomethingView.php
       SomethingCreate.php
       SomethingDelete.php
       data_export/
           ExportAsXml.php
           ExportAsExcel.php
   views/                                          View Dateien.
       example.something.view.php
       example.something.delete.php
   assets/                                         Alle zusätzlichen Dateien, die in Views verwendet werden so
       js/                                         JavaScript Dateien, die in Views verwendet werden.
           example.something.view.js.php
       css/                                        CSS-Dateien, die in den Views verwendet werden.
           example.something.css
        image.png                                  Bilder, die in Views verwendet werden.
        example.something.file                     Beliebige Dateien zur Verwendung in Views.
```

Modul Dateistruktur

Schreiben eines Moduls Ein Beispiel für den Prozess zum Schreiben eines Moduls besteht aus den folgenden Schritten (falls verfügbar, klicken Sie auf den Datei- oder Ordnernamen, um zusätzliche Details zum Schritt anzuzeigen):

1. Erstellen Sie ein neues Verzeichnis für das Modul innerhalb von **zabbix/ui/modules/**.
2. Fügen Sie die Datei manifest.json mit den Metadaten des Moduls hinzu.
3. Erstellen Sie den Ordner *views* und definieren Sie eine oder mehrere Modulansichten.
4. Erstellen Sie den Ordner *actions* und definieren Sie eine oder mehrere Modulaktionen.
5. Erstellen Sie die Datei Module.php (oder Widget.php für Dashboard-Widgets) und definieren Sie die Initialisierungs- und Ereignisbehandlungsregeln.
6. Erstellen Sie den Ordner *assets* für JavaScript-Dateien (legen Sie diese unter *assets/js* ab), CSS-Styles (unter *assets/css*) oder andere zusätzliche Dateien.
7. Stellen Sie sicher, dass die erforderlichen Ansichten, Aktionen und Asset-Dateien in manifest.json angegeben sind.
8. Registrieren Sie das Modul im Zabbix Frontend und beginnen Sie mit der Nutzung.

> **Note:**
> Sie können ein Modul registrieren und aktivieren, sobald Sie die Datei manifest.json erstellt haben. Sobald das Modul aktiviert ist, können Sie alle an den Moduldateien vorgenommenen Änderungen sofort durch Aktualisieren des Zabbix Frontend in der Vorschau anzeigen.

manifest.json

Jedes Modul benötigt die Datei manifest.json. Die Datei muss sich im primären Verzeichnis des Moduls befinden (zum Beispiel *zabbix/ui/modules/module_name/manifest.json*).

Mindestens sollte manifest.json diese Felder angeben:

```
{
    "manifest_version": 2.0,
    "id": "my_ip_address",
    "name": "My IP Address",
    "namespace": "MyIPAddress",
    "version": "1.0"
}
```

Unterstützte Parameter in **manifest.json** (klicken Sie auf den Parameternamen für eine detaillierte Beschreibung):

Parameter	Beschreibung	Erforderlich
manifest_version	Manifest-Version des Moduls.	Ja
id	Eindeutige Modul-ID.	
name	Modulname, der im Bereich Administration angezeigt wird.	
namespace	PHP-Namespace für Modulklassen.	
version	Modulversion.	
type	Typ des Moduls. Für Widgets muss *widget* gesetzt werden	Ja für Widgets, sonst nein
widget	Widget-Konfiguration. Nur für Widgets verwendet.	
actions	Mit dem Modul zu registrierende Aktionen.	
assets	CSS-Styles und JavaScript-Dateien, die eingebunden werden sollen.	Nein
author	Autor des Moduls.	
config	Standardwerte für benutzerdefinierte Moduloptionen.	
description	Modulbeschreibung.	
url	Ein Link zur Modulbeschreibung.	

manifest_version

Manifestversion des Moduls. Derzeit wird die Version **2.0** unterstützt.

Typ: Double

Beispiel:

```
"manifest_version": 2.0
```

id

Modul-ID. Muss eindeutig sein. Um zukünftige Namenskonflikte zu vermeiden, wird empfohlen, ein Präfix für Module zu verwenden (Autor- oder Firmenname oder ein anderes). Wenn ein Modul beispielsweise ein Beispiel für Lektionen ist und der Modulname „Mein Modul“ lautet, lautet die ID „example_my_module“.

Typ: Zeichenfolge

Beispiel:

```
„id“: „example_my_module“
```

Name

Modulname, der im Abschnitt „Administration“ angezeigt wird.

Typ: Zeichenfolge

Beispiel:

```
"name": "Mein Modul"
```

Namespace

PHP-Namespace für Modulklassen.

Typ: String

Beispiel:

```
"namespace": "ClockWidget"
```

Version

Modulversion. Die Version wird im Bereich „Administration“ angezeigt.

Typ: Zeichenfolge

Beispiel:

```
"version": "1.0"
```

Typ

Typ des Moduls. Erforderlich für Widgets und muss ”Widget” entsprechen.

Typ: Zeichenfolge

Standard: ”Modul”

Beispiel:

```
"Typ": "Widget"
```

Aktionen

Aktionen, die mit dem Modul registriert werden sollen.

Die Definition des Objektschlüssels *class* für jede Aktion ist erforderlich, andere Aktionsschlüssel sind optional.

Type: Objekt

Unterstützte Objektschlüssel, wenn *type* auf *module* gesetzt ist:

- **dein.eigener.aktions.name** (Objekt) – Aktionsname, sollte in Kleinbuchstaben [a-z] geschrieben und Wörter durch Punkte getrennt werden. Unterstützt die Schlüssel:
- **class** (Zeichenfolge; erforderlich) – Name der Aktionsklasse.
- **layout** (Zeichenfolge) – Aktionslayout. Unterstützte Werte: *layout.json*, *layout.htmlpage* (Standard), *null*.
- **view** (Zeichenfolge) – Aktionsansicht.

Beispiel:

```
"actions": {
"module.example.list": {
"class": "ExampleList",
"view": "example.list",
"layout": "layout.htmlpage"
}
}
```

Unterstützte Objektschlüssel, wenn *type* auf *widget* gesetzt ist:

- **widget.{id}.view** (Objekt) - Datei- und Klassenname für die Widget-Ansicht. Ersetzen Sie **{id}** durch den id-Wert des Widgets (z. B. *widget.example_clock.view*). Unterstützt die Schlüssel:
- **class** (Zeichenfolge; erforderlich) - Aktionsklassenname für den Widget-Ansichtsmodus, um die Standardklasse CController-DashboardWidgetView zu erweitern. Die Klassenquelldatei muss sich im Verzeichnis *actions* befinden.
- **view** (Zeichenfolge) - Widget-Ansicht. Muss sich im Verzeichnis *views* befinden. Wenn die Ansichtsdatei *widget.view.php* ist, was standardmäßig erwartet wird, kann dieser Parameter weggelassen werden. Wenn Sie einen anderen Namen verwenden, geben Sie ihn hier an.
- **widget.{id}.edit** (Objekt) – Dateiname für die Widget-Konfigurationsansicht. Ersetzen Sie **{id}** durch den id-Wert des Widgets (z. B. *widget.example_clock.edit*). Unterstützt die Schlüssel:
- **class** (Zeichenfolge; erforderlich) – Name der Aktionsklasse für den Widget-Konfigurationsansichtsmodus. Die Klassenquell-datei muss sich im Verzeichnis *actions* befinden.
- **view** (Zeichenfolge) – Widget-Konfigurationsansicht. Muss sich im Verzeichnis *views* befinden. Wenn die Ansichtsdatei *widget.edit.php* ist, was standardmäßig erwartet wird, kann dieser Parameter weggelassen werden. Wenn Sie einen anderen Namen verwenden, geben Sie ihn hier an.

Beispiel:

```
"actions": {
"widget.tophosts.view": {
"class": "WidgetView"
},
"widget.tophosts.column.edit": {
"class": "ColumnEdit",
"view": "column.edit",
"layout": "layout.json"
}
}
```

Assets

Einzuschließende CSS-Stile und JavaScript-Dateien.

Typ: Objekt

Unterstützte Objektschlüssel:

- **css** (Array) – einzuschließende CSS-Dateien. Die Dateien müssen sich in *assets/css* befinden.
- **js** (Array) – einzuschließende JavaScript-Dateien. Die Dateien müssen sich in *assets/js* befinden.

Beispiel:

```
"assets": {
"css": ["widget.css"],
"js": ["class.widget.js"]
}
```

Autor

Modulautor. Der Autor wird im Bereich ”Administration” angezeigt.

Typ: Zeichenfolge

Beispiel:

```
"author": "John Smith"
```

Konfiguration

Standardwerte für die Moduloptionen. Das Objekt kann beliebige benutzerdefinierte Schlüssel enthalten. Falls angegeben, werden diese Werte bei der Modulregistrierung in die Datenbank geschrieben. Später hinzugefügte neue Variablen werden beim ersten Aufruf geschrieben. Danach können die Variablenwerte nur noch direkt in der Datenbank geändert werden.

Typ: Objekt

Beispiel:

```
"config": {
"username": "Admin",
"password": "",
"auth_url": "https://example.com/auth"
}
```

Beschreibung

Modulbeschreibung.

Typ: Zeichenfolge

Beispiel:

```
"Beschreibung": "Dies ist ein Uhr-Widget."
```

widget

Widget-Konfiguration. Wird verwendet, wenn *type* auf *widget* eingestellt ist.

Typ: Objekt

Unterstützte Objektschlüssel:

- **name** (Zeichenfolge) – wird in der Widgetliste und als Standardkopfzeile verwendet. Wenn leer, wird der Parameter „name“ aus dem Modul verwendet.
- **size** (Objekt) – Standard-Widgetabmessungen. Unterstützte Schlüssel:
- *width* (Ganzzahl) – Standard-Widgetbreite.
- *height* (Ganzzahl) – Standard-Widgethöhe.
- **form_class** (Zeichenfolge) – Klasse mit Widgetfeldformular. Muss sich im Verzeichnis *includes* befinden. Wenn die Klasse *WidgetForm.php* ist, was standardmäßig erwartet wird, kann dieser Parameter weggelassen werden. Wenn Sie einen anderen Namen verwenden, geben Sie ihn hier an.
- **js_class** (Zeichenfolge) – Name einer JavaScript-Klasse für den Widget-Ansichtsmodus, um die Standardklasse CWidget zu erweitern. Die Klasse wird mit dem Dashboard geladen. Die Klassenquelldatei muss sich im Verzeichnis „assets/js“ befinden. Siehe auch: assets.
- **use_time_selector** (Boolesch) – bestimmt, ob das Widget einen Dashboard-Zeitselektor benötigt.

Unterstützte Werte: *true*, *false* (Standard).

- **refresh_rate** (Ganzzahl) – Aktualisierungsrate des Widgets in Sekunden (Standard: 60).

Beispiel:

```
"widget": {
"name": "",
"size": {
"width": 12,
"height": 5
},
"form_class": "WidgetForm",
"js_class": "CWidget",
"use_time_selector": false,
"refresh_rate": 60
}
```

url

Ein Link zur Modulbeschreibung. Bei Widgets wird dieser Link geöffnet, wenn im Fenster *Widget hinzufügen* oder *Widget bearbeiten* auf das Hilfesymbol ? geklickt wird. Wenn **url** nicht angegeben ist, wird beim Klicken auf das Hilfesymbol die allgemeine Seite Dashboard-Widgets geöffnet.

Typ: String

Beispiel:

```
"url": "http://example.com"
```

Aktionen (Actions)

Aktionen sind für die 'Geschäftslogik' des Moduls verantwortlich. Eine Aktion besteht in der Regel aus einem controller und einer action view.

Ein Modul kann:

- Aktionen aufrufen, die bereits im Zabbix-Frontend definiert sind.
- Standardaktionen mit eigenen Aktionen überschreiben.
- Völlig neue Aktionen definieren.

Um das Verhalten einer Standardaktion durch ein benutzerdefiniertes Verhalten zu überschreiben, definieren Sie eine Aktion mit demselben Namen in der Modulkonfiguration. Wenn die Aktion aufgerufen wird, wird die Modulaktion anstelle der Standardaktion von Zabbix ausgeführt.

Aktionsdateien sollten im Ordner *actions* gespeichert werden. Die Aktionen müssen in der manifest.json angegeben werden.

Controller

Workflow des Action-Controllers:

1. Prüfen, dass alle in einer HTTP-Anfrage übergebenen Parameter gültig sind:
   - Die Methode *checkInput()* des Controllers aufrufen;
   - Die in CNewValidator.php definierten Validierungsregeln verwenden;
   - Die Methode *validateInput()* aufrufen.
2. Benutzerberechtigungen prüfen.
3. Die Daten entsprechend den übergebenen Parametern vorbereiten: Wenn *checkInput()* true zurückgibt, ruft Zabbix die Methode *doAction()* des Controllers auf.
4. Das Array **$data** für die Ansicht vorbereiten. Verwenden Sie *CControllerResponseData* und die Methode *setResponse()*, um die Antwort im Array **$data** zu speichern.

Beispiel:

```
/**
 * Eingabeparameter validieren.
 *
 * @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;
}

/**
 * Benutzerberechtigungen prüfen.
 *
 * @return bool
 */
protected function checkPermissions() {
    return $this->getUserType() >= USER_TYPE_ZABBIX_ADMIN;
}

/**
 * Action ausführen und Antwortobjekt erzeugen.
 */
protected function doAction(): void {
    $data = [
        'hosts_count' => API::Host()->get([
            'countOutput' => true,
            'filter' => [
                'status' => $this->getInput('status')
            ]
        ])
    ];

    $this->setResponse(new CControllerResponseData($data));
}
```

Note:
Die vollständige Liste der verfügbaren Controller-Klassen können Sie im Zabbix-Quellcode einsehen.

Ansichten

Eine Ansichtsdatei empfängt die Daten von einem Controller und bereitet daraus die HTML-Darstellung vor.

Note:
Das Definieren von Ansicht(en) für ein Frontend-Modul ist optional, es sei denn, das Modul ist ein Widget.
 Dashboard-Widgets benötigen mindestens zwei Ansichten: eine für den Bearbeitungsmodus und eine für den Ansichtsmodus (sie sollten im Verzeichnis *views* gespeichert werden).

Es ist möglich, in der Ansicht vordefinierte Zabbix-HTML-Klassen (aus */zabbix/ui/include/classes/html*) zu verwenden sowie neue HTML- und CSS-Klassen hinzuzufügen. Neue Klassen sollten im Ordner *assets* des Moduls gespeichert werden.

Beispiel:

```
...
(new CColHeader(_('Name')))
```

Dadurch wird ein neuer Spaltenname *Name* hinzugefügt und die oberste Tabellenzeile wie auf anderen Zabbix-Seiten gestaltet.

Aktionsansicht

Dies ist eine Referenzdatei zum Definieren einer Aktionsansicht.

```
<?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();
```

Assets

Der Ordner *assets* kann beliebige Dateien und Unterordner enthalten, die nicht zu anderen Verzeichnissen gehören. Sie können ihn verwenden für:

- JavaScript-Stile (müssen sich in *assets/js* befinden);
- CSS-Stile (müssen sich innerhalb von *assets/css* befinden);
- Bilder;
- Schriftarten;
- Alles, was Sie sonst noch einfügen müssen.

assets/js

Das Verzeichnis *assets/js* ist reserviert und sollte nur JavaScript-Dateien enthalten. Um vom Widget verwendet zu werden, geben Sie diese Dateien in der *manifest.json* an.

Zum Beispiel:

```
"assets": {
    "js": ["class.widget.js"]
}
```

assets/css

assets/css ist reserviert und sollte nur CSS-Stil-Dateien enthalten. Um vom Widget verwendet zu werden, geben Sie diese Dateien in der *manifest.json* an.

Zum Beispiel:

```
"assets": {
    "css": ["mywidget.css"]
}
```

CSS-Stile

CSS-Dateien können ein benutzerdefiniertes Attribut `theme` enthalten, um unterschiedliche Stile für bestimmte Frontend-Themes zu definieren.

Verfügbare Themes und ihre Attributwerte:

- **Blau** - [theme='blue-theme']
- **Blau (klassisch)** - [theme='blue-classic-theme']
- **Dunkel** - [theme='dark-theme']
- **Dunkelblau** - [theme='dark-blue-theme']
- **Dunkel (klassisch)** - [theme='dark-classic-theme']
- **Hoher Kontrast, hell** - [theme='hc-light']
- **Hoher Kontrast, dunkel** - [theme='hc-dark']

Beispiel:

```
.widget {
    background-color: red;
}

[theme='dark-theme'] .widget {
    background-color: green;
}
```

Registrierung eines neuen Moduls

In diesem Abschnitt wird erklärt, wie Sie ein neues Modul zum Zabbix-Frontend hinzufügen können.

Vorraussetzungen Bevor Sie fortfahren, stellen Sie sicher, dass: - Das Modul befindet sich im Verzeichnis *modules* Ihrer Zabbix-Frontend-Installation (zum Beispiel *zabbix/ui/modules*). - Das Modul hat zumindest eine Basisversion der Datei manifest.json. - Sie haben Zugriff auf den Menüabschnitt ”Administration” in Zabbix (erfordert den Rollentyp ”Super admin user”)

Hinzufügen eines Moduls Öffnen Sie die Seite *Administration* > *Allgemein* > *Module* und klicken Sie auf *Verzeichnis scannen*.



Suchen Sie Ihr Modul in der Liste und aktivieren Sie es.

Um ein Modul zu aktivieren, klicken Sie auf den Hyperlink *Deaktiviert* - der Status des Moduls wird auf *Aktiviert* geändert.

Klicken Sie auf den Modulnamen, um zusätzliche Informationen zum Modul anzuzeigen, z. B. Autor, Version oder eine kurze Beschreibung (falls im Manifest definiert).

Widget-Vorschau Sobald Widget-Module hinzugefügt wurden, sind sie sofort in der Widget-Liste des Dashboards sichtbar.

Sie können ein Dashboard öffnen, in den Bearbeitungsmodus wechseln und das Widget wie gewohnt zum Dashboard hinzufügen.

Wenn Sie Änderungen am Widget vornehmen, aktualisieren Sie das Dashboard, um zu sehen, wie das Widget mit den neuesten Aktualisierungen aussieht.

Widgets

Widgets sind Zabbix-Frontend-Module, die für die Dashboards genutzt werden. Sofern nicht anders angegeben, gelten alle Modul-richtlinien auch für Widgets.

Ein Widget unterscheidet sich jedoch deutlich von einem Modul. Um ein Widget zu erstellen:

- den Typ „widget“ in der Datei manifest.json angeben („type“: „widget“);
- mindestens zwei Ansichten einfügen: eine für den widget presentation mode und eine für den widget configuration mode (example.widget.view.php und example.widget.edit.php);
- und ein controller für die Widget-Präsentation (WidgetView.php);
- Verwendung und Erweiterung der Standard-widget classes.

Konfiguration

Auf dieser Seite werden Klassen beschrieben, die zum Erstellen einer Widget-Konfigurationsansicht mit benutzerdefinierten Konfigurationsfeldern verwendet werden können. Die Widget-Konfigurationsansicht ist der Teil des Widgets, der es dem Benutzer ermöglicht, Widget-Parameter für Präsentation zu konfigurieren.

Widget

Primäre Widget-Klasse, erweitert die Basisklasse aller Dashboard-Widgets - *CWidget*. Erforderlich, um das Standardverhalten des Widgets zu überschreiben.

Die Klasse *Widget* sollte sich im Stammverzeichnis des Widgets befinden (zum Beispiel, *zabbix/ui/modules/my_custom_widget*).

Widget.php Beispiel

```
<?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

Die Klasse *WidgetForm* erweitert die Standardklasse *CWidgetForm* und enthält einen Satz von *CWidgetField* Feldern welche für die Definition der Speicherstruktur der Widget-Konfiguration in der Datenbank und für die Eingabevalidierung erforderlich sind.

Die Klasse *WidgetForm* sollte sich im Verzeichnis *includes* befinden. Wenn die Klasse einen anderen Namen hat, sollte der Name im Parameter *widget/form_class* in der Datei *manifest.json* angegeben werden.

includes/WidgetForm.php Beispiel

```
<?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

Die Klasse *CWidgetFormView* wird benötigt, um die Darstellungslogik der in der Klasse *WidgetForm* definierten Felder festzulegen und ihr Aussehen und Verhalten bei der Darstellung in der Konfigurationsansicht zu bestimmen.

Die Klasse *CWidgetFormView* unterstützt die folgenden Methoden:

- *addField()* – erhält eine Instanz der Klasse *CWidgetFieldView* als Parameter. Für jede Klasse *CWidgetField* gibt es eine entsprechende Klasse *CWidgetFieldView*, die in der Widget-Konfigurationsansicht verwendet wird.
- *addFieldset()* – erhält eine Instanz der Klasse *CWidgetFieldsGroupView*, die Felder in einem ausklappbaren Container zusammenfasst.
- *addFieldsGroup()* – erhält eine Instanz von *CWidgetFormFieldsetCollapsibleView*, die Felder visuell (mit einem Rahmen) zu einer Gruppe zusammenfasst.
- *includeJsFile()* – ermöglicht das Hinzufügen einer JavaScript-Datei zur Widget-Konfigurationsansicht.
- *addJavaScript()* – ermöglicht das Hinzufügen von Inline-JavaScript, das ausgeführt wird, sobald die Widget-Konfigurationsansicht geladen ist.

Die Klasse *CWidgetFormView* sollte sich im Verzeichnis *views* befinden.

Beispiel für views/widget.edit.php

```php
<?php

/**
 * My custom widget form view.
 *
 * @var CView $this
 * @var array $data
 */

use Modules\MyCustomWidget\Includes\WidgetForm;

(new CWidgetFormView($data))
    ->addField(
        (new CWidgetFieldMultiSelectItemView($data['fields']['itemid']))->setPopupParameter('numeric', tru
    )
    ->addFieldset(
        (new CWidgetFormFieldsetCollapsibleView(_('Advanced configuration')))
            ->addField(
                new CWidgetFieldTextBoxView($data['fields']['description'])
            )
            ->addField(
                new CWidgetFieldColorView($data['fields']['chart_color'])
            )
    )
    ->includeJsFile('widget.edit.js.php')
    ->addJavaScript('my_custom_widget_form.init('.json_encode([
        'color_palette' => WidgetForm::DEFAULT_COLOR_PALETTE
    ]).');')
    ->show();
```

JavaScript

Eine JavaScript-Klasse kann verwendet werden, um der Widget-Konfigurationsansicht dynamisches Verhalten und Interaktivität hinzuzufügen. Sie können zum Beispiel einen Farbwähler initialisieren, der in der Klasse *CWidgetFormView* definiert ist.

Die JavaScript-Klasse sollte mit dem Formular geladen werden, daher sollte sie in der Klasse *CWidgetFormView* mit den Methoden *includeJsFile()* und *addJavaScript()* referenziert werden.

Im folgenden Beispiel wird sofort eine Singleton-Klasseninstanz erstellt und unter dem Namen *window.my_custom_widget_form* gespeichert. Wenn Sie also das Formular zum zweiten Mal öffnen, wird die Instanz neu erstellt.

Die JavaScript-Klasse sollte sich im Verzeichnis *views* befinden.

views/widget.edit.js.php Beispiel

```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

Die Klasse *CWidgetField* ist eine Basisklasse, von der alle Formularfeldklassen (*CWidgetFieldCheckBox*, *CWidgetFieldTextArea*, *CWidgetFieldRadioButtonList* usw.) abgeleitet werden. Von *CWidgetField* abgeleitete Klassen sind für das Empfangen, Speichern und Validieren von Konfigurationswerten des Widgets verantwortlich.

Die folgenden *CWidgetField*-Klassen sind verfügbar.

CWidgetField class	Database field type	Description
CWidgetFieldCheckBox	int32	Einzelnes Kontrollkästchen.
CWidgetFieldCheckBoxList	array of int32	Mehrere Kontrollkästchen unter einem einzelnen Konfigurationsfeld.
CWidgetFieldColor	string	Feld zur Farbauswahl.
CWidgetFieldDatePicker	string	Feld zur Auswahl eines Datums.
CWidgetFieldIntegerBox	int32	Feld zur Eingabe einer Ganzzahl. Kann zur Konfiguration von Minimal- und Maximalwerten verwendet werden.
CWidgetFieldLatLng	string	Textfeld zur Eingabe von durch Kommas getrennten Angaben für Breitengrad, Längengrad und Kartenzoomstufe.
CWidgetFieldMultiSelect	-	Basisklasse für Mehrfachauswahlfelder, von der alle `CWidgetFieldMultiSelect*`-Klassen abgeleitet werden.
CWidgetFieldMultiSelectAction	ID	Mehrfachauswahlfeld zur Auswahl von Aktionen aus der Liste der unter *Alerts > Actions* definierten Aktionen.
CWidgetFieldMultiSelectGraph	ID	Mehrfachauswahlfeld zur Auswahl benutzerdefinierter Graphen.
CWidgetFieldMultiSelectGraphPrototype	ID	Mehrfachauswahlfeld zur Auswahl benutzerdefinierter Graph-Prototypen.
CWidgetFieldMultiSelectGroup	ID	Mehrfachauswahlfeld zur Auswahl von Hostgruppen.
CWidgetFieldMultiSelectHost	ID	Mehrfachauswahlfeld zur Auswahl von Hosts.
CWidgetFieldMultiSelectHostInventory	array of int32	Mehrfachauswahlfeld zur Auswahl von Hostinventarfeldern.
CWidgetFieldMultiSelectItem	ID	Mehrfachauswahlfeld zur Auswahl von Datenpunkten.
CWidgetFieldMultiSelectItemPrototype	ID	Mehrfachauswahlfeld zur Auswahl von Datenpunkt-Prototypen.
CWidgetFieldMultiSelectMap	ID	Mehrfachauswahlfeld zur Auswahl von Karten.
CWidgetFieldMultiSelectMediaType	ID	Mehrfachauswahlfeld zur Auswahl von Medientypen.
CWidgetFieldMultiSelectOverrideHost	ID	Mehrfachauswahlfeld zur Auswahl einer Datenquelle für Hosts aus einem Dashboard oder einem anderen Widget.
CWidgetFieldMultiSelectService	ID	Mehrfachauswahlfeld zur Auswahl von Services.
CWidgetFieldMultiSelectSla	ID	Mehrfachauswahlfeld zur Auswahl von SLAs.
CWidgetFieldMultiSelectUser	ID	Mehrfachauswahlfeld zur Auswahl von Benutzern.
CWidgetFieldNavTree	array of (multiple mixed)	Speichert eine Navigationsbaumstruktur von Karten zur Verwendung im Widget *Map navigation tree*.
CWidgetFieldNumericBox	string	Feld zur Eingabe einer Gleitkommazahl.
CWidgetFieldPatternSelect	-	Basisklasse für Musterauswahlfelder, von der alle `CWidgetPatternSelect*`-Klassen abgeleitet werden.
CWidgetFieldPatternSelectHost	string	Mehrfachauswahlfeld zur Auswahl von Hosts. Unterstützt Hostnamensmuster.
CWidgetFieldPatternSelectItem	string	Mehrfachauswahlfeld zur Auswahl von Datenpunkten. Unterstützt Datenpunktnamensmuster.
CWidgetFieldRadioButtonList	int32	Optionsfeldgruppe, die aus einem oder mehreren Optionsfeldern besteht.
CWidgetFieldRangeControl	int32	Schieberegler zur Auswahl eines Werts vom Typ Ganzzahl.

CWidgetField class	Database field type	Description
CWidgetFieldReference	string	Speichert eine eindeutige Kennung für dieses Widget im Dashboard; wird für die Kommunikation zwischen Widgets verwendet.
CWidgetFieldSelect	int32	Dropdown-Auswahlfeld.
CWidgetFieldSeverities	array of int32	Mit Auslöser-Schweregraden voreingestelltes *CWidgetFieldCheckBoxList*.
CWidgetFieldSparkline	array of (multiple mixed)	Ermöglicht die Konfiguration eines Sparkline-Diagramms.
CWidgetFieldTags	array of (string, int32, string)	Ermöglicht die Konfiguration einer oder mehrerer Zeilen für Tag-Filter.
CWidgetFieldTextArea	string	Textbereich zur Eingabe von mehrzeiligem Text.
CWidgetFieldTextBox	string	Textfeld zur Eingabe von einzeiligem Text.
CWidgetFieldThresholds	array of (string, string)	Ermöglicht die Konfiguration von Farb- und Zahlenpaaren.
CWidgetFieldTimePeriod	array of string	Feld zur Auswahl eines Zeitraums.
CWidgetFieldTimeZone	string	Dropdown-Liste mit Zeitzonen.
CWidgetFieldUrl	string	Textfeld zur Eingabe von URLs.

Note:
Die maximale Länge für `int32`-Felder beträgt 10 Zeichen, für ID-Felder 20 Zeichen und für `string`-Felder 65535 Zeichen. Bei zusammengesetzten Typen (z. B. `array of (string, int32, string)`) gilt die Begrenzung für jedes Element einzeln. Feldspezifische Längenbeschränkungen können mithilfe von Methoden wie `setMaxLength()` festgelegt werden.

Präsentation

Diese Seite beschreibt die Komponenten, die zur Erstellung einer Widget-Präsentationsansicht verwendet werden können. Die Widget-Präsentationsansicht ist der Teil des Widgets, der die Daten gemäß seiner Konfiguration empfängt und sie auf dem Dashboard in einem Container anzeigt.

Die Präsentationsansicht besteht aus drei Teilen:

- Widget-Aktion
- Widget-Ansicht
- JavaScript

Widget-Aktion

Die Widget-Aktionsklasse (*WidgetView*) enthält Methoden für Operationen mit Widgets im Präsentationsansichtsmodus. Die meisten Widget-Aktionen verwenden und/oder erweitern die Standard-Controllerklasse *CControllerDashboardWidgetView*.

Die Widget-Aktionsklasse sollte sich im Verzeichnis *actions* befinden und im Parameter *actions* (*actions/widget.{id}.view/class*) in der Datei *manifest.json* angegeben sein.

actions/WidgetView.php Beispiel (implementiert im Zabbix-eigenen *Systeminformationen* widget)

```
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()
            ]
        ]));
    }
}
```

Widget-Ansicht

Die Widget-Ansichtsklasse (*CWidgetView*) ist für den Aufbau der Widget-Darstellungsansicht verantwortlich.

Die Widget-Ansichtsklasse sollte sich im *views*-Verzeichnis befinden.
Wenn die Datei, die die Widget-Ansichtsklasse enthält, einen anderen Namen als den Standardnamen (*widget.view.php*) hat, muss sie im *manifest.json*-Datei-*actions*-Parameter (*actions/widget.{id}.view/view*) angegeben werden.

views/widget.view.php Beispiel

```
<?php

/**
 * Meine benutzerdefinierte Widget-Ansicht.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem(
    new CTag('h1', true, $data['name'])
    )
    ->show();
```

JavaScript

Die JavaScript-Klasse ist für das Verhalten des Widgets verantwortlich, wie das Aktualisieren von Widget-Daten, das Anpassen der Größe des Widgets, das Anzeigen von Widget-Elementen usw.

Alle JavaScript-Operationen verwenden und/oder erweitern die JavaScript-Basisklasse für alle Dashboard-Widgets - *CWidget*.
Die *CWidget*-Klasse enthält eine Reihe von Methoden mit der Standardimplementierung für das Widget-Verhalten.
Abhängig von der Komplexität des Widgets können diese Methoden so verwendet oder erweitert werden.

Die *CWidget*-Klasse enthält die folgenden Methoden:

- Methoden, die den Lebenszyklus eines Widgets definieren: *onInitialize()*, *onStart()*, *onActivate()*, *onDeactivate()*, *onDestroy()*, *onEdit()*.
- Methoden, die das Aktualisieren und Anzeigen von Widget-Daten steuern: *promiseUpdate()*, *getUpdateRequestData()*, *processUpdateResponse(response)*, *processUpdateErrorResponse(error)*, *setContents(response)*.
- Methoden, die das Aussehen des Widgets anpassen: *onResize()*, *hasPadding()*.

Die JavaScript-Klasse sollte sich im Verzeichnis *assets/js* befinden und im *assets*-Parameter (*assets/js*) in der *manifest.json*-Datei angegeben werden.

Lebenszyklusmethoden

Die Lebenszyklusmethoden des Widgets werden vom Dashboard und in verschiedenen Stadien des Lebenszyklus des Widgets während seiner Existenz im Dashboard aufgerufen.

Die Methode ***onInitialize()*** definiert den anfänglichen Zustand und/oder die Werte des Widgets, ohne HTML- oder Datenmanipulationen durchzuführen.
Diese Methode wird aufgerufen, wenn ein Widget erstellt wird (ein Widget-Objekt instanziiert wird), typischerweise durch das Hinzufügen des Widgets zu einer Dashboard-Seite oder durch das Laden der Dashboard-Seite.

Beispiel:

```
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;
}
```

Die Methode ***onStart()*** definiert die HTML-Struktur des Widgets, ohne dabei Daten zu verarbeiten. Diese Methode wird vor der ersten Aktivierung der Dashboard-Seite aufgerufen, also bevor das Dashboard und seine Widgets dem Benutzer vollständig angezeigt werden.

Beispiel:

```
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`
        );
    }
}
```

Die Methode ***onActivate()*** macht das Widget aktiv und interaktiv, indem sie benutzerdefinierte Event-Listener aktiviert (um auf Benutzeraktionen zu reagieren) und den Aktualisierungszyklus des Widgets startet (um seinen Inhalt auf dem neuesten Stand zu halten). Diese Methode wird aufgerufen, wenn die Dashboard-Seite aktiviert wird, das heißt, wenn sie in der Benutzeroberfläche vollständig angezeigt wird.

Beachten Sie, dass sich das Widget vor dem Aufruf der Methode *onActivate()* im inaktiven Zustand (`WIDGET_STATE_INACTIVE`) befindet. Nach erfolgreichem Aufruf wechselt das Widget in den aktiven Zustand (`WIDGET_STATE_ACTIVE`). Im aktiven Zustand reagiert das Widget, lauscht auf Ereignisse, aktualisiert seinen Inhalt periodisch und kann mit anderen Widgets interagieren.

Beispiel:

```
onActivate() {
    this._startClock();

    this._resize_observer = new ResizeObserver(this._events.resize);
    this._resize_observer.observe(this._target);
}
```

Die Methode ***onDeactivate()*** beendet jede Aktivität und Interaktivität des Widgets, indem benutzerdefinierte Event-Listener deaktiviert und der Aktualisierungszyklus des Widgets gestoppt werden. Diese Methode wird aufgerufen, wenn die Dashboard-Seite deaktiviert wird, d. h. wenn zu einer anderen Seite gewechselt wird oder sie gelöscht wird, oder wenn das Widget von der Dashboard-Seite gelöscht wird.

Beachten Sie, dass sich das Widget vor dem Aufruf der Methode *onDeactivate()* im aktiven Zustand (`WIDGET_STATE_ACTIVE`) befindet. Nach erfolgreichem Aufruf wechselt das Widget in den inaktiven Zustand (`WIDGET_STATE_INACTIVE`).

Beispiel:

```
onDeactivate() {
    this._stopClock();
    this._resize_observer.disconnect();
}
```

Die Methode ***onDestroy()*** führt Bereinigungsaufgaben aus, bevor das Widget aus dem Dashboard gelöscht wird. Dazu kann gehören, eine Datenbankverbindung zu schließen, die während der Widget-Initialisierung aufgebaut wurde, temporäre Daten zu bereinigen, um Systemspeicher freizugeben und Ressourcenlecks zu vermeiden, sowie Event-Listener im Zusammenhang mit Größenänderungsereignissen oder Button-Klicks abzumelden, um unnötige Ereignisverarbeitung und Speicherlecks zu verhindern usw. Diese Methode wird aufgerufen, wenn das Widget oder die Dashboard-Seite, die es enthält, gelöscht wird.

Beachten Sie, dass ein Widget im aktiven Zustand (`WIDGET_STATE_ACTIVE`) immer zuerst durch Aufruf der Methode *onDeactivate()* deaktiviert wird, bevor die Methode *onDestroy()* aufgerufen wird.

Beispiel:

```
onDestroy() {
    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);
    }
}
```

Die Methode ***onEdit()*** definiert das Erscheinungsbild und Verhalten des Widgets, wenn das Dashboard in den Bearbeitungsmodus wechselt. Diese Methode wird aufgerufen, wenn das Dashboard in den Bearbeitungsmodus wechselt, typischerweise wenn ein

Benutzer mit der Schaltfläche *Bearbeiten* des Widgets oder der Schaltfläche *Dashboard bearbeiten* des Dashboards interagiert.

Beispiel:

```
onEdit() {
    this._deactivateGraph();
}
```

Methoden des Aktualisierungsprozesses

Die Methoden des Widget-Aktualisierungsprozesses sind dafür verantwortlich, aktualisierte Daten vom Zabbix-Server oder einer anderen Datenquelle abzurufen und im Widget anzuzeigen.

Die Methode ***promiseUpdate()*** startet den Datenaktualisierungsprozess, indem sie Daten abruft, typischerweise über Webanfragen oder API-Aufrufe. Diese Methode wird aufgerufen, wenn eine Dashboard-Seite angezeigt wird, und danach in regelmäßigen Abständen, bis zu einer anderen Dashboard-Seite gewechselt wird.

Im Folgenden sehen Sie ein Beispiel für die Standardimplementierung der Methode *promiseUpdate()*, die von den meisten nativen Zabbix-Widgets verwendet wird. In der Standardimplementierung folgt die Methode *promiseUpdate()* einem allgemeinen Muster zum Abrufen von Daten vom Server. Sie erstellt ein neues `Curl`-Objekt mit der entsprechenden URL und den Anfrageparametern, sendet mit der Methode *fetch()* eine POST-Anfrage unter Verwendung des Datenobjekts, das von der Methode *getUpdateRequestData()* erstellt wird, und verarbeitet die Antwort (oder eine Fehlerantwort) entsprechend mit *processUpdateResponse(response)* oder *processUpdateErrorResponse(error)*. Diese Implementierung eignet sich für die meisten Widgets, da sie Daten in der Regel im JSON-Format abrufen und diese auf konsistente Weise verarbeiten.

```
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);
        });
    }
```

Die Methode ***getUpdateRequestData()*** bereitet die Server-Anfragedaten für die Aktualisierung des Widgets vor, indem sie verschiedene Eigenschaften und ihre entsprechenden Werte (Widget-Kennungen, Filtereinstellungen, Zeitbereiche usw.) aus dem Zustand und der Konfiguration des Widgets sammelt und ein Datenobjekt erstellt, das die erforderlichen Informationen darstellt, die in der Aktualisierungsanfrage an den Server gesendet werden sollen. Diese Methode wird nur als Teil der Standardmethode *promiseUpdate()* aufgerufen, also während des Widget-Aktualisierungsprozesses.

Standardimplementierung:

```
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 : undefined,
        view_mode: this._view_mode,
        edit_mode: this._is_edit_mode ? 1 : 0,
        dynamic_hostid: this._dashboard.templateid !== null || this.supportsDynamicHosts()
            ? (this._dynamic_hostid ?? undefined)
            : undefined,
```

```
            ...this._contents_size
        };
}
```

Die Methode ***processUpdateResponse(response)*** verarbeitet die Antwort, die nach der Aktualisierungsanfrage vom Server empfangen wird, und löscht, wenn der Aktualisierungsprozess erfolgreich und fehlerfrei war, die Widget-Daten und zeigt mit der Methode *setContents()* neue Inhalte an. Diese Methode wird nur als Teil der Standardmethode *promiseUpdate()* aufgerufen, also während des Widget-Aktualisierungsprozesses.

Standardimplementierung:

```
processUpdateResponse(response) {
    this._setHeaderName(response.name);

    this._updateMessages(response.messages);
    this._updateInfo(response.info);
    this._updateDebug(response.debug);

    this.setContents(response);
}
```

Die Methode ***processUpdateErrorResponse(error)*** verarbeitet die vom Server nach der Aktualisierungsanfrage empfangene Antwort, wenn die Antwort ein Fehler ist, und zeigt die Fehlermeldung/-en an. Diese Methode wird nur als Teil der Standardmethode *promiseUpdate()* aufgerufen, also während des Widget-Aktualisierungsprozesses.

Standardimplementierung:

```
processUpdateErrorResponse(error) {
    this._updateMessages(error.messages, error.title);
}
```

Die Methode ***setContents(response)*** zeigt den Inhalt des Widgets an, wenn der Widget-Aktualisierungsprozess erfolgreich und ohne Fehler abgeschlossen wurde, was unter anderem die Bearbeitung von DOM-Elementen, das Aktualisieren von UI-Komponenten, das Anwenden von Stilen oder Formatierungen usw. umfassen kann. Diese Methode wird nur als Teil der Standardmethode *processUpdateResponse(response)* aufgerufen, also während der Verarbeitung der Antwort, die nach der Aktualisierungsanfrage vom Server empfangen wurde.

Standardimplementierung:

```
setContents(response) {
    this._body.innerHTML = response.body ?? '';
}
```

Methoden zur Änderung der Darstellung

Die Methoden zur Änderung der Widget-Präsentation sind für die Änderung des Erscheinungsbildes der Widgets zuständig.

Die Methode ***onResize()*** ist dafür verantwortlich, die visuellen Elemente des Widgets an die neue Widget-Größe anzupassen, was unter anderem das Neuanordnen von Elementen, das Anpassen von Elementabmessungen, das Kürzen von Text, die Implementierung von Lazy Loading zur Verbesserung der Reaktionsfähigkeit während der Größenänderung usw. umfassen kann. Diese Methode wird aufgerufen, wenn die Größe des Widgets geändert wird, zum Beispiel wenn der Benutzer die Größe des Widgets manuell ändert oder wenn die Größe des Browserfensters geändert wird.

Beispiel:

```
onResize() {
    if (this.getState() === WIDGET_STATE_ACTIVE) {
        this._startUpdating();
    }
}
```

Die Methode ***hasPadding()*** ist dafür verantwortlich, am unteren Rand des Widgets einen vertikalen Innenabstand von 8 px anzuwenden, wenn es so konfiguriert ist, dass sein Header angezeigt wird. Diese Methode wird aufgerufen, wenn die Dashboard-Seite aktiviert wird, also wenn sie zur in der Benutzeroberfläche angezeigten Seite wird.

Standardimplementierung:

```
hasPadding() {
    return this.getViewMode() !== ZBX_WIDGET_VIEW_MODE_HIDDEN_HEADER;
}
```

Bei einigen Widgets ist es erforderlich, den gesamten verfügbaren Widget-Bereich zu nutzen, um beispielsweise eine benutzerdefinierte Hintergrundfarbe zu konfigurieren. Im Folgenden sehen Sie ein Beispiel für die Implementierung der Methode *hasPadding()*, die im nativen Zabbix-Widget *Datenpunktwert* verwendet wird.

```
hasPadding() {
    return false;
}
```

Tutorials

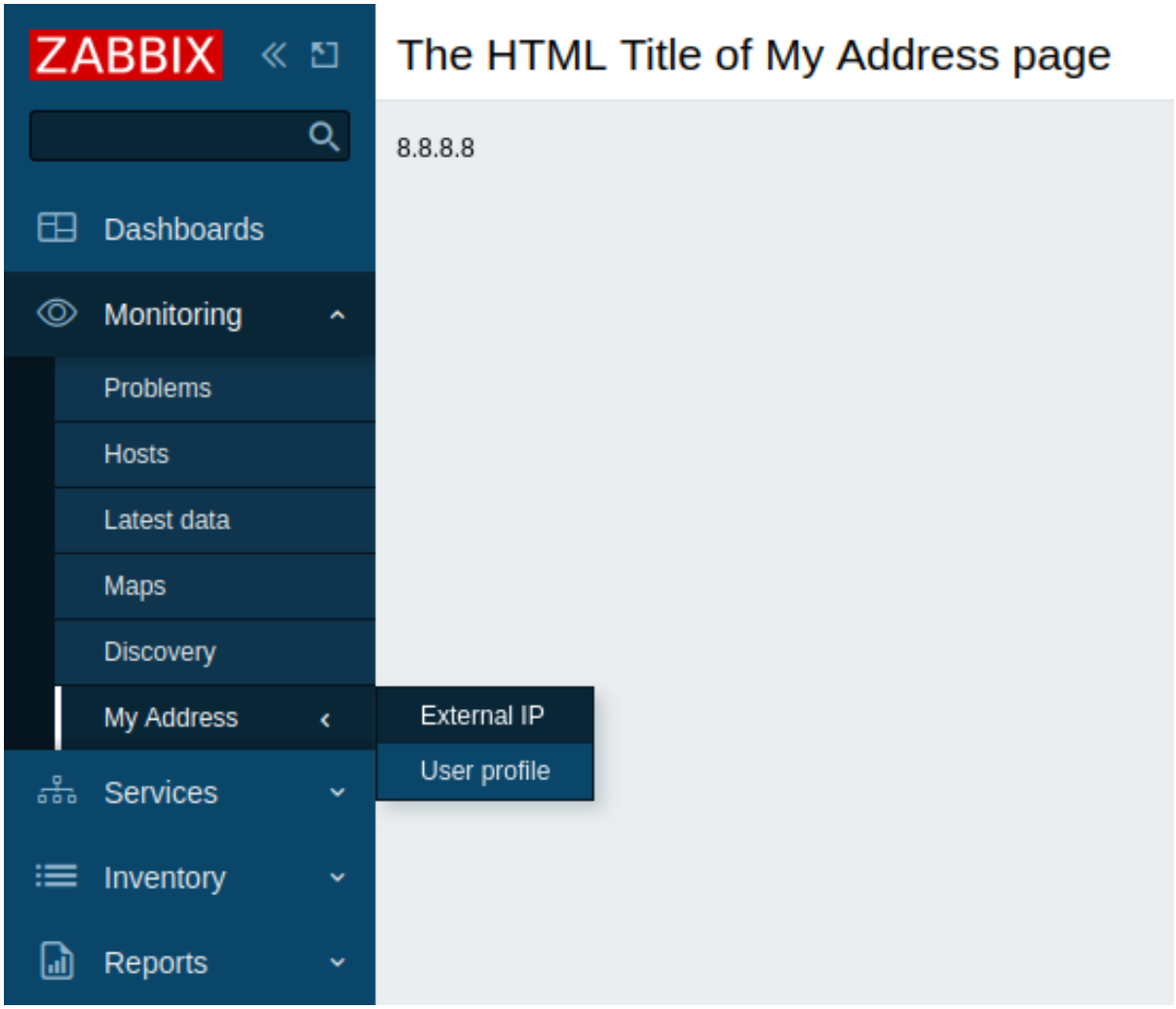
Dieser Abschnitt enthält praktische Schritt-für-Schritt-Tutorials, die veranschaulichen, wie man in Zabbix ein benutzerdefiniertes Modul und ein Widget erstellt.

Erstellen eines Moduls (Tutorial)

Dies ist ein Schritt-für-Schritt-Tutorial, das zeigt, wie man ein einfaches Zabbix-Frontend-Modul erstellt. Sie können alle Dateien dieses Moduls als ZIP-Archiv herunterladen: MyAddress.zip.

Was Sie erstellen werden

In diesem Tutorial erstellen Sie zunächst ein Frontend-Modul, das einen neuen Menüabschnitt *My Address* hinzufügt, und wandeln es anschließend in ein fortgeschritteneres Frontend-Modul um, das eine HTTP-Anfrage an *https://api.seeip.org* sendet und die Antwort – die IP-Adresse Ihres Computers – auf einer neuen Seite im neu erstellten Menüabschnitt *My Address* anzeigt. So wird das fertige Modul aussehen:



Teil I - Neuer Menübereich

Ein leeres Modul zum Zabbix-Frontend hinzufügen

1. Erstellen Sie ein Verzeichnis *MyAddress* im Verzeichnis *modules* Ihrer Zabbix-Frontend-Installation (zum Beispiel *zabbix/ui/modules*).

2. Erstellen Sie eine Datei *manifest.json* mit grundlegenden Metadaten des Moduls (siehe die Beschreibung der unterstützten Parameter).

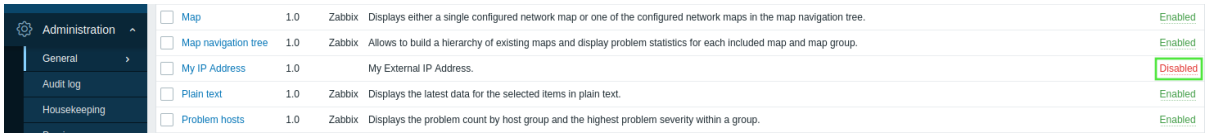
ui/modules/MyAddress/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "my-address",
    "name": "My IP Address",
    "version": "1.0",
    "namespace": "MyAddress",
    "description": "My External IP Address."
}
```

3. Wechseln Sie im Zabbix-Frontend zu *Administration* > *Allgemein* > *Module* und klicken Sie auf die Schaltfläche *Verzeichnis scannen*.



4. Suchen Sie in der Liste nach dem neuen Modul *My IP Address* und klicken Sie auf den Hyperlink "Disabled", um den Status des Moduls von "Disabled" auf "Enabled" zu ändern (falls das Modul nicht aufgeführt ist, siehe den Abschnitt Fehlerbehebung).



Das Modul ist nun im Frontend registriert. Es ist jedoch noch nicht sichtbar, da Sie die Funktionalität des Moduls noch definieren müssen. Sobald Sie Inhalte zum Modulverzeichnis hinzufügen, sehen Sie die Änderungen im Zabbix-Frontend sofort nach dem Aktualisieren der Seite.

Erstellen eines Menüabschnitts

1. Erstellen Sie eine Datei *Module.php* im Verzeichnis *MyAddress*.

Diese Datei implementiert eine neue Klasse *Module*, die von der Standardklasse *CModule* erbt. Die Klasse *Module* fügt einen neuen Menüabschnitt *My Address* in das Hauptmenü ein.

Die Methode *setAction()* legt eine Aktion fest, die beim Klicken auf den Menüabschnitt ausgeführt wird. Zunächst können Sie die vordefinierte Aktion *userprofile.edit* verwenden, die die Seite *User profile* öffnet. In Teil III dieses Tutorials erfahren Sie, wie Sie eine benutzerdefinierte Aktion erstellen.

ui/modules/MyAddress/Module.php

```
<?php

namespace Modules\MyAddress;

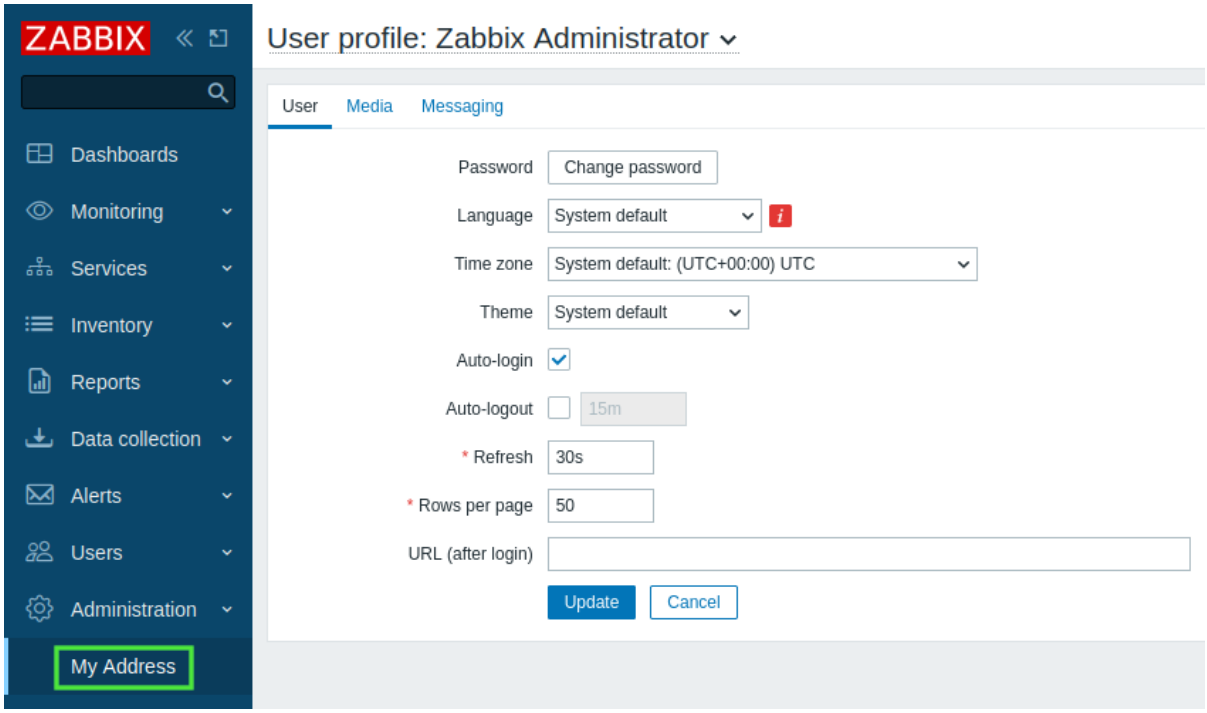
use Zabbix\Core\CModule,
    APP,
    CMenuItem;

class Module extends CModule {

    public function init(): void {
        APP::Component()->get('menu.main')
            ->add((new CMenuItem(_('My Address')))
            ->setAction('userprofile.edit'));
    }
}
```

Note:
Sie können `'userprofile.edit'` durch andere Aktionen ersetzen, zum Beispiel durch `'charts.view'` (öffnet benutzerdefinierte Diagramme), `'problems.view'` (öffnet *Monitoring* > *Problems*) oder `'report.status'` (öffnet den Bericht *System information*).

3. Aktualisieren Sie das Zabbix Frontend. Unten im Zabbix-Hauptmenü gibt es jetzt einen neuen Abschnitt *My Address*. Klicken Sie auf *My Address*, um die Seite *User profile* zu öffnen.



Teil II - Änderung der Position des Menüabschnitts

In diesem Teil verschieben Sie den Menüabschnitt *My Address* in den Abschnitt *Monitoring* und fügen ihm anschließend ein verschachteltes Menü hinzu. Dadurch können Benutzer über den Menüabschnitt *Monitoring* > *My Address* auf zwei Untermenüseiten zugreifen.

1. Öffnen und bearbeiten Sie die Datei *Module.php*.

ui/modules/MyAddress/Module.php

```
<?php

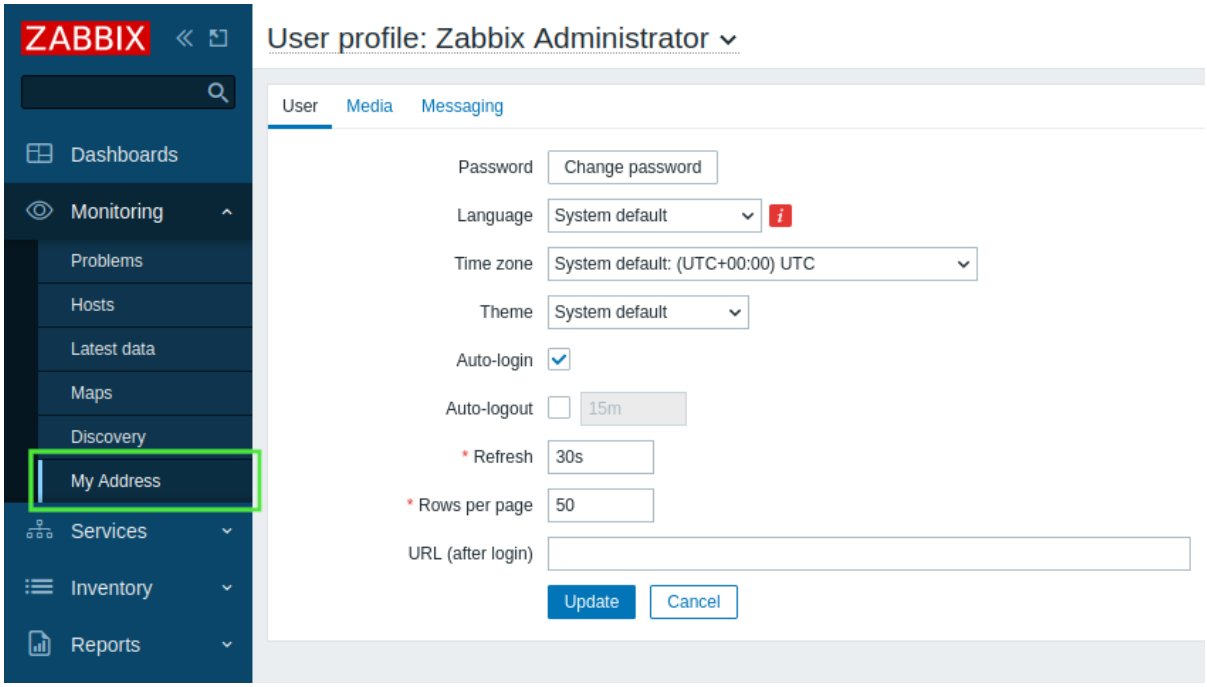
namespace Modules\MyAddress;

use Zabbix\Core\CModule,
    APP,
    CMenuItem;

class Module extends CModule {

    public function init(): void {
        APP::Component()->get('menu.main')
            ->findOrAdd(_('Monitoring'))
            ->getSubmenu()
            ->insertAfter(_('Discovery'),
                (new CMenuItem(_('My Address')))->setAction('userprofile.edit')
            );
    }
}
```

2. Aktualisieren Sie das Zabbix Frontend. Erweitern Sie den Menüabschnitt *Monitoring* und beobachten Sie, dass sich der Abschnitt *My address* nun unterhalb des Abschnitts *Discovery* befindet.



3. Um dem Menüabschnitt *My Address* verschachtelte Seiten hinzuzufügen, öffnen und bearbeiten Sie die Datei *Module.php* erneut.

Dieser Schritt erstellt zwei Unterabschnitte:

- *External IP*, das eine neue Aktion 'my.address' ausführt, die in den nächsten Schritten definiert wird;
- *User profile*, das die vordefinierte Aktion 'userprofile.edit' ausführt, um die Seite *User profile* zu öffnen.

Beachten Sie, dass Sie für das verschachtelte Menü zusätzlich zu den in den vorherigen Schritten verwendeten Klassen die Klasse *CMenu* verwenden müssen.

ui/modules/MyAddress/Module.php

```
<?php

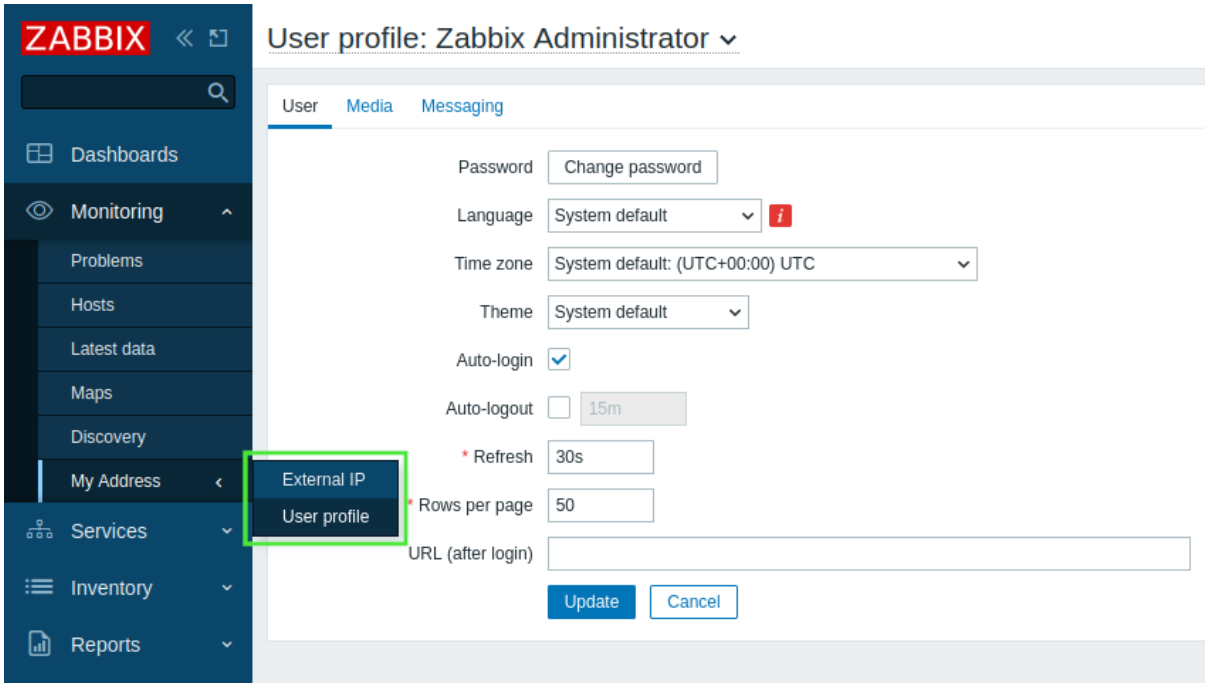
namespace Modules\MyAddress;

use Zabbix\Core\CModule,
    APP,
    CMenu,
    CMenuItem;

class Module extends CModule {

    public function init(): void {
        APP::Component()->get('menu.main')
            ->findOrAdd(_('Monitoring'))
            ->getSubmenu()
            ->insertAfter(_('Discovery'),
                (new CMenuItem(_('My Address')))->setSubMenu(
                    new CMenu([
                        (new CMenuItem(_('External IP')))->setAction('my.address'),
                        (new CMenuItem(_('User profile')))->setAction('userprofile.edit')
                    ])
                )
            );
    }
}
```

4. Aktualisieren Sie das Zabbix Frontend. Beobachten Sie, dass der Menüabschnitt *My address* nun ein Menü der dritten Ebene mit zwei Seiten enthält - *External IP* und *User profile*.



Teil III - Modulaktion

Eine Aktion wird in zwei Dateien implementiert - *actions/MyAddress.php* und *views/my.address.php*. Die Datei ***actions/MyAddress.php*** übernimmt die Implementierung der Geschäftslogik, während die Datei ***views/my.address.php*** für die Ansicht verantwortlich ist.

1. Erstellen Sie im Verzeichnis *MyAddress* ein Verzeichnis *actions*.
2. Erstellen Sie im Verzeichnis *actions* eine Datei *MyAddress.php*.

Die Aktionslogik wird in der Klasse *MyAddress* definiert. Diese Aktionsklasse implementiert vier Funktionen: *init()*, *checkInput()*, *checkPermissions()* und *doAction()*. Das Zabbix Frontend ruft die Funktion *doAction()* auf, wenn die Aktion angefordert wird. Diese Funktion ist für die Geschäftslogik des Moduls verantwortlich.

> **Attention:**
> Die Daten müssen als assoziatives Array organisiert sein. Das Array kann mehrdimensional sein und beliebige Daten enthalten, die von der Ansicht erwartet werden.

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. Erstellen Sie im Verzeichnis *MyAddress* ein neues Verzeichnis *views*.
4. Erstellen Sie im Verzeichnis *views* eine Datei *my.address.php* und definieren Sie die Modulansicht.

Beachten Sie, dass die Variable `$data` in der Ansicht verfügbar ist, ohne dass sie speziell definiert werden muss. Das Framework übergibt das assoziative Array automatisch an die Ansicht.

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();
```

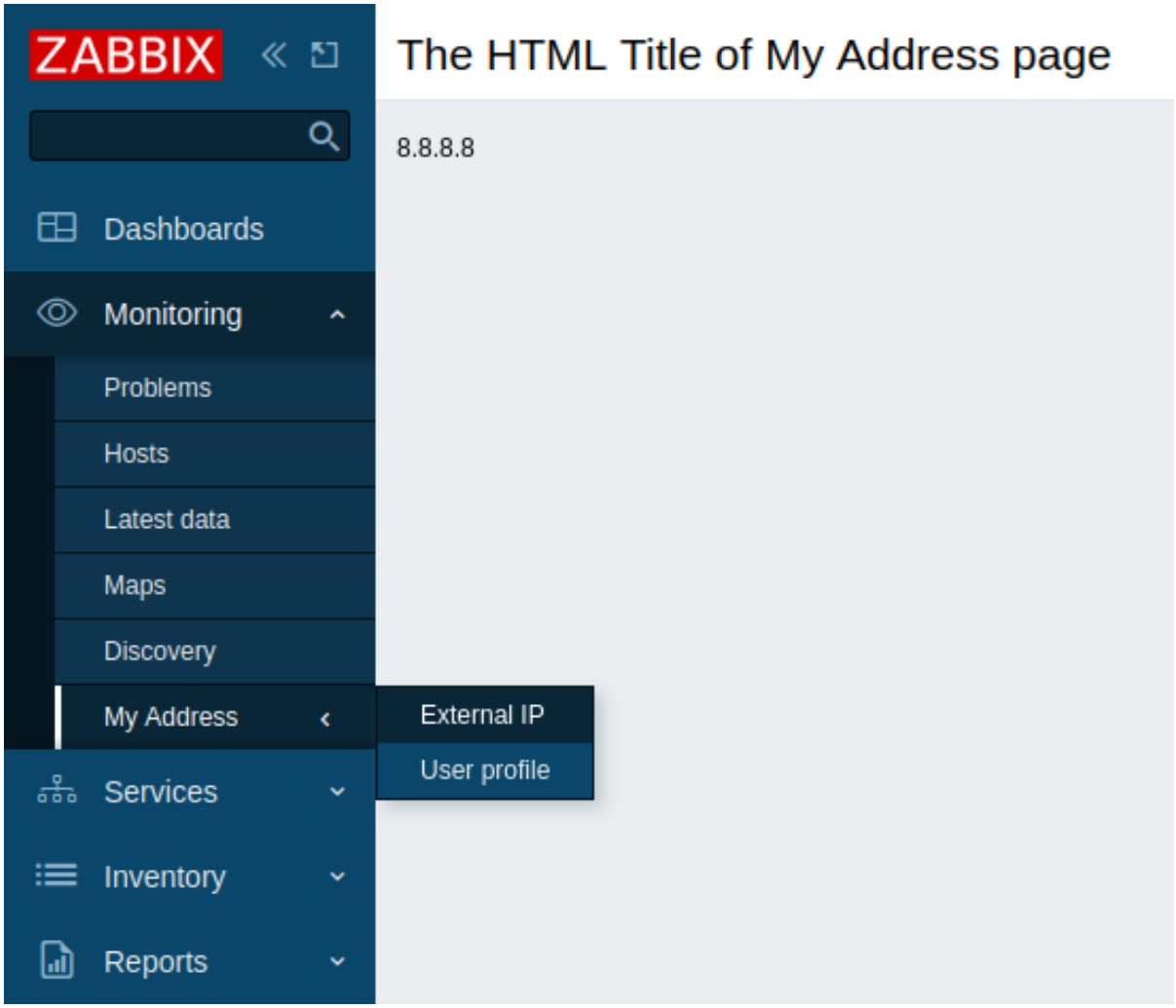
5. Die Modulaktion muss in der Datei *manifest.json* registriert werden. Öffnen Sie *manifest.json* und fügen Sie ein neues Objekt `actions` hinzu, das Folgendes enthält:

- den Aktionsschlüssel mit dem Aktionsnamen in Kleinbuchstaben (a-z) und mit durch Punkte getrennten Wörtern (zum Beispiel `my.address`);
- den Klassennamen der Aktion (`MyAddress`) als Wert für den Schlüssel `class` des Objekts `my.address`;
- den Ansichtsname der Aktion (`my.address`) als Wert für den Schlüssel `view` des Objekts `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"
        }
    }
}
```

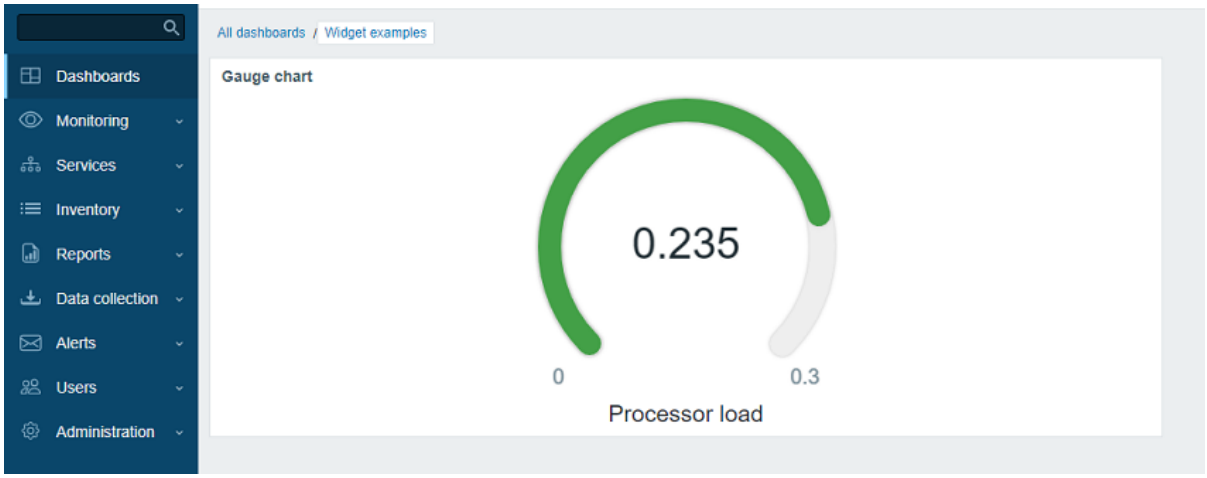
6. Aktualisieren Sie das Zabbix Frontend. Klicken Sie auf *My address* > *External IP*, um die IP-Adresse Ihres Computers anzuzeigen.



Erstellen eines Widgets (Tutorial)

Dies ist ein Schritt-für-Schritt Tutorial, welches dir zeigt, wie man ein einfaches Zabbix-Frontend-Modul erstellt. Sie können alle Dateien dieses Moduls als ZIP-Archiv herunterladen: MyAddress.zip.

Was Sie erstellen werden In diesem Tutorial erstellen Sie zunächst ein einfaches „Hello, world!“-Widget und wandeln es anschließend in ein fortgeschritteneres Widget um, das einen Datenpunktwert als Tachodiagramm anzeigt. So wird das fertige Widget aussehen:



Teil I – ”Hallo Welt!“ In diesem Abschnitt erfahren Sie, wie Sie die minimal erforderlichen Widget-Elemente erstellen und ein neues Widget zum Zabbix-Frontend hinzufügen.

Ein leeres Widget zum Zabbix-Frontend hinzufügen

1. Erstellen Sie ein Verzeichnis `lesson_gauge_chart` im Verzeichnis `modules` Ihrer Zabbix-Frontend-Installation (zum Beispiel *zabbix/ui/modules*).

> **Note:**
> Alle benutzerdefinierten Widgets werden als externe Module behandelt und müssen dem Verzeichnis `modules` Ihrer Zabbix-Frontend-Installation hinzugefügt werden (zum Beispiel *zabbix/ui/modules*). Das Verzeichnis `zabbix/ui/widgets` ist für die integrierten Zabbix-Widgets reserviert und wird zusammen mit der Zabbix-UI aktualisiert.

2. Erstellen Sie eine Datei `manifest.json` mit grundlegenden Widget-Metadaten (siehe die Beschreibung der unterstützten Parameter).

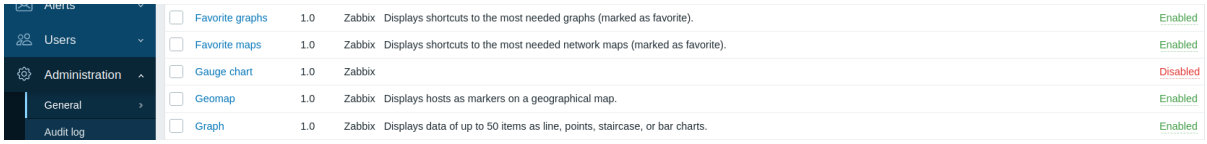
ui/modules/lesson_gauge_chart/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
    "type": "widget",
    "name": "Gauge chart",
    "namespace": "LessonGaugeChart",
    "version": "1.1",
    "author": "Zabbix"
}
```

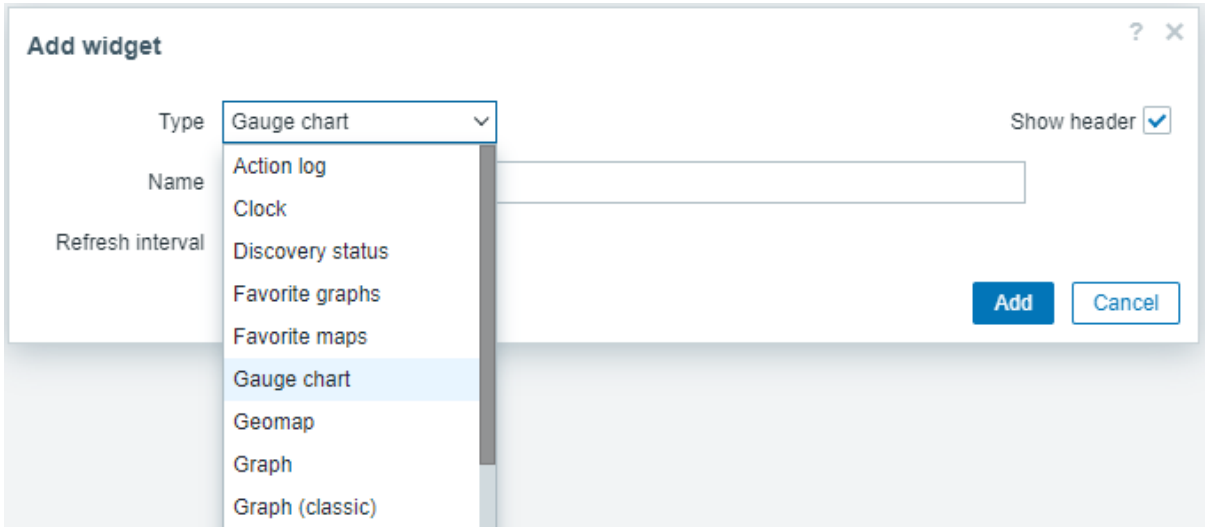
3. Gehen Sie im Zabbix-Frontend zu *Administration* > *Allgemein* > *Module* und klicken Sie auf die Schaltfläche *Verzeichnis scannen*.



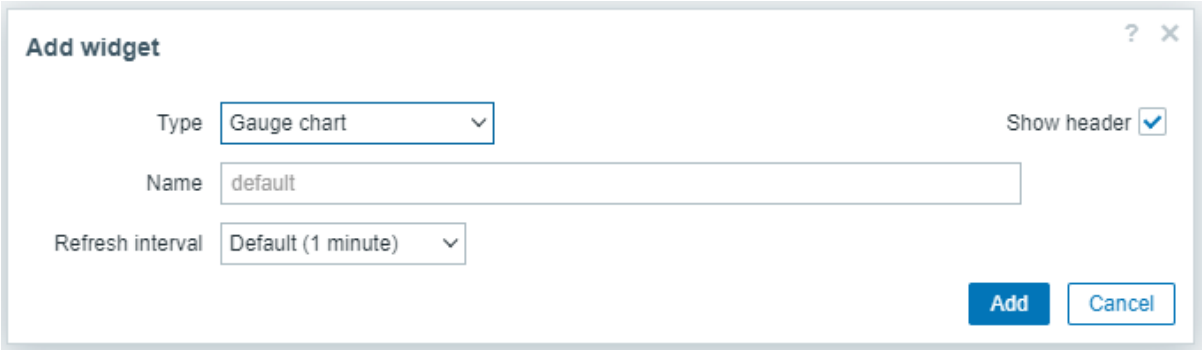
4. Suchen Sie in der Liste nach dem neuen Modul *Gauge chart* und klicken Sie auf den Hyperlink "Disabled", um den Status des Moduls von "Disabled" auf "Enabled" zu ändern (falls das Modul nicht aufgeführt ist, siehe Abschnitt Fehlerbehebung).



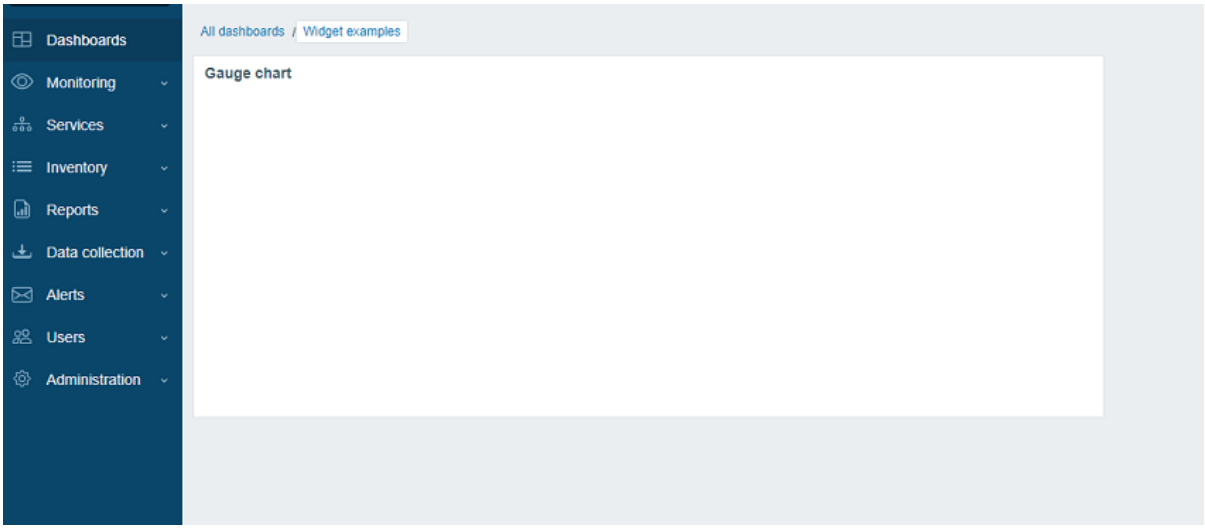
5. Öffnen Sie ein Dashboard, wechseln Sie in den Bearbeitungsmodus und fügen Sie ein neues Widget hinzu. Wählen Sie im Feld *Type* "Gauge chart" aus.



6. Zu diesem Zeitpunkt enthält die Konfiguration des Widgets *Gauge chart* nur die allgemeinen Widget-Felder *Name* und *Aktualisierungsintervall*. Klicken Sie auf *Add*, um das Widget dem Dashboard hinzuzufügen.



7. Ein leeres Widget sollte auf dem Dashboard erscheinen. Klicken Sie oben rechts auf *Save changes*, um das Dashboard zu speichern.



Eine Widget-Ansicht hinzufügen

Note:
Die **view**-Datei des Widgets sollte sich im Verzeichnis `views` befinden (für dieses Tutorial: `ui/modules/lesson_gauge_chart/views/`). Wenn die Datei den Standardnamen `widget.view.php` hat, müssen Sie sie nicht in der Datei `manifest.json` registrieren. Wenn die Datei einen anderen Namen hat, geben Sie sie im Abschnitt `actions/widget.lesson_gauge_chart.view` der Datei `manifest.json` an.

1. Erstellen Sie im Verzeichnis `lesson_gauge_chart` ein Verzeichnis `views`.
2. Erstellen Sie im Verzeichnis `views` eine Datei `widget.view.php`.

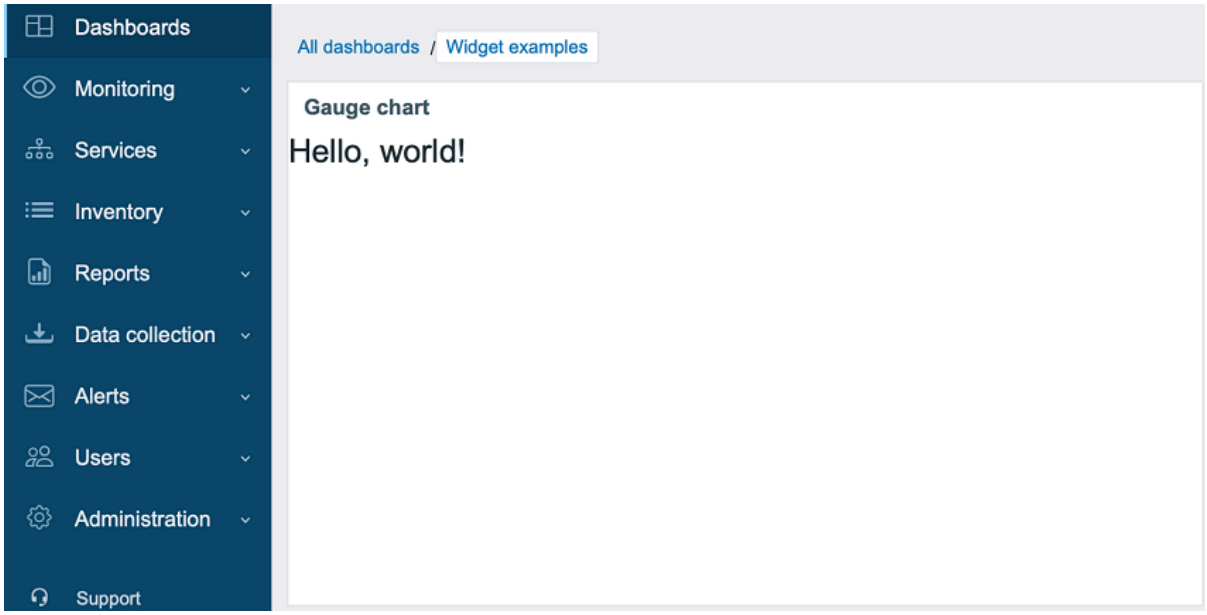
`ui/modules/lesson_gauge_chart/views/widget.view.php`

```
<?php

/**
 * Gauge chart widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem(
        new CTag('h1', true, 'Hello, world!')
    )
    ->show();
```

3. Aktualisieren Sie das Dashboard. Das Widget *Gauge chart* zeigt jetzt "Hello, world!" an.



Teil II - Gauge Chart (Messgerätetabelle) Einstellungen zu einer Konfigurationsansicht hinzufügen und in einer Widget-Ansicht verwenden

In diesem Abschnitt erfahren Sie, wie Sie ein Widget-Konfigurationsfeld hinzufügen und den eingegebenen Wert in der Widget-Ansicht als Text anzeigen.

Die Widget-Konfiguration besteht aus einem Formular (`Zabbix\Widgets\CWidgetForm`) und einer Widget-Formularansicht (`widget.edit.php`). Um Felder (`Zabbix\Widgets\CWidgetField`) hinzuzufügen, müssen Sie eine `WidgetForm`-Klasse erstellen, die von `Zabbix\Widgets\CWidgetForm` erbt.

Das Formular enthält eine Reihe von Feldern (`Zabbix\Widgets\CWidgetField`) verschiedener Typen, die zur Validierung der vom Benutzer eingegebenen Werte verwendet werden. Das Formularfeld (`Zabbix\Widgets\CWidgetField`) für jeden Eingabetyp konvertiert den Wert in ein einheitliches Format, um ihn in der Datenbank zu speichern.

> **Note:**
> Die **Formular**-Datei des Widgets sollte sich im Verzeichnis `includes` befinden (für dieses Tutorial: `ui/modules/lesson_gauge_chart/includes/`). Wenn die Datei den Standardnamen `WidgetForm.php` hat, müssen Sie sie nicht in der Datei `manifest.json` registrieren. Wenn die Datei einen anderen Namen hat, geben Sie sie im Abschnitt `widget/form_class` der Datei `manifest.json` an.

1. Erstellen Sie im Verzeichnis `lesson_gauge_chart` ein neues Verzeichnis `includes`.
2. Erstellen Sie im Verzeichnis `includes` eine Datei `WidgetForm.php`.

`ui/modules/lesson_gauge_chart/includes/WidgetForm.php`

```
<?php

namespace Modules\LessonGaugeChart\Includes;

use Zabbix\Widgets\CWidgetForm;

class WidgetForm extends CWidgetForm {
}
```

3. Fügen Sie dem Widget-Konfigurationsformular ein Feld *Description* hinzu. Dies ist ein reguläres Textfeld, in das ein Benutzer beliebige Zeichen eingeben kann. Dafür können Sie die Klasse `CWidgetFieldTextBox` verwenden.

`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. Erstellen Sie im Verzeichnis `views` eine Widget-Konfigurationsansichtsdatei `widget.edit.php` und fügen Sie eine Ansicht für das neue Feld *Description* hinzu. Für die Feldklasse `CWidgetFieldTextBox` lautet die Ansicht `CWidgetFieldTextBoxView`.

`ui/modules/lesson_gauge_chart/views/widget.edit.php`

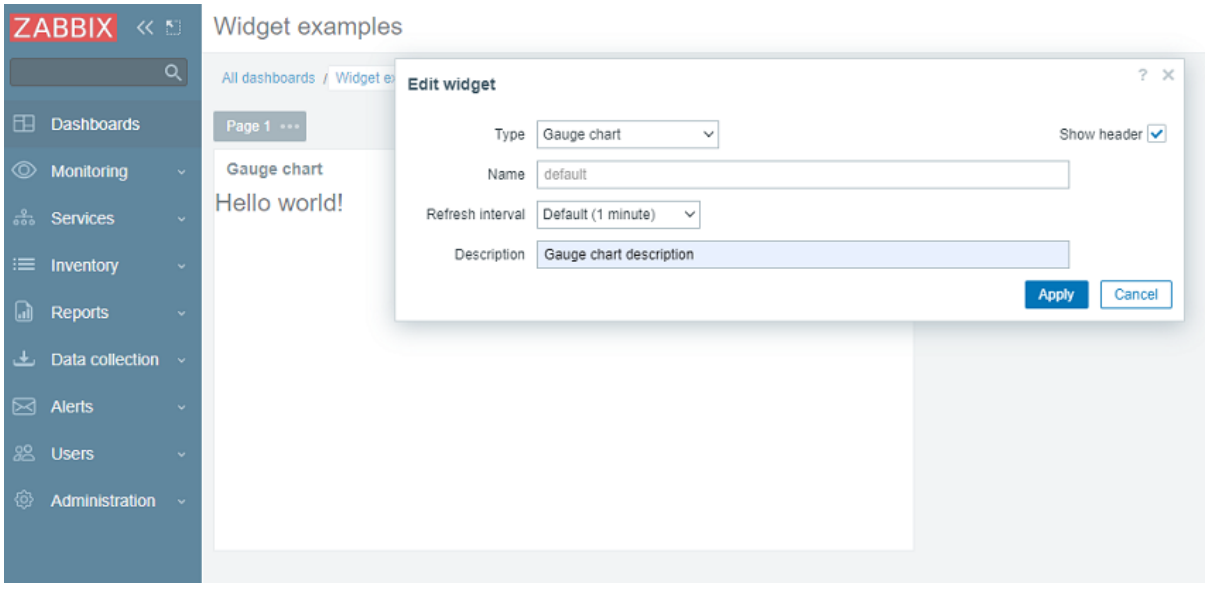
```
<?php

/**
 * Gauge chart widget form view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetFormView($data))
    ->addField(
        new CWidgetFieldTextBoxView($data['fields']['description'])
    )
    ->show();
```

5. Gehen Sie zum Dashboard und klicken Sie auf das Zahnradsymbol im Widget, um das Widget-Konfigurationsformular zu öffnen.

6. Das Widget-Konfigurationsformular enthält jetzt ein neues Textfeld *Description*. Geben Sie einen beliebigen Wert ein, zum Beispiel *Gauge chart description*.



7. Klicken Sie im Widget-Konfigurationsformular auf *Apply*. Klicken Sie anschließend oben rechts auf *Save changes*, um das Dashboard zu speichern. Beachten Sie, dass die neue Beschreibung nirgendwo sichtbar ist und das Widget weiterhin "Hello, world!" anzeigt.

Damit die neue Beschreibung im Widget angezeigt wird, muss der Wert des Feldes *Description* aus der Datenbank abgerufen und an die Widget-Ansicht übergeben werden. Dafür müssen Sie eine Aktionsklasse erstellen.

8. Erstellen Sie im Verzeichnis `lesson_gauge_chart` ein neues Verzeichnis `actions`.

9. Erstellen Sie im Verzeichnis `actions` eine Datei `WidgetView.php`. Die Aktionsklasse `WidgetView` erbt von der Klasse `CControllerDashboardWidgetView`.

Die Werte der Widget-Konfigurationsfelder werden in der Eigenschaft `$fields_values` der Aktionsklasse gespeichert.

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. Öffnen Sie `manifest.json` und registrieren Sie `WidgetView` als Aktionsklasse im Abschnitt `actions/widget.lesson_gauge_char`

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. Jetzt können Sie den Wert des Feldes description, der in `$data['description']` enthalten ist, in der Widget-Ansicht verwenden. Öffnen Sie `views/widget.view.php` und ersetzen Sie den statischen Text "Hello, world!" durch `$data['description']`.

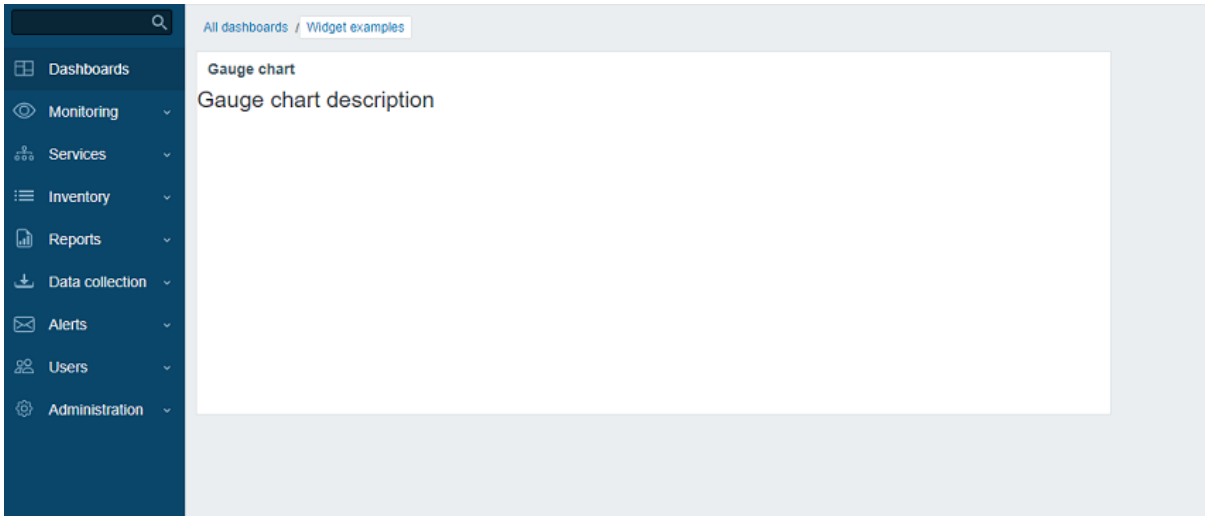
ui/modules/lesson_gauge_chart/views/widget.view.php

```
<?php

/**
 * Gauge chart widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem(
        new CTag('h1', true, $data['description'])
    )
    ->show();
```

12. Aktualisieren Sie die Dashboard-Seite. Sie sollten nun den Beschreibungstext des Widgets anstelle von "Hello, world!" sehen.



Einen Datenpunktwert über die API abrufen

Das Widget sollte den letzten Wert eines Datenpunkts nach Wahl des Benutzers anzeigen. Dafür müssen Sie die Möglichkeit hinzufügen, Datenpunkte in der Widget-Konfiguration auszuwählen.

In diesem Abschnitt erfahren Sie, wie Sie dem Widget-Formular ein Feld zur Auswahl eines Datenpunkts hinzufügen und wie Sie den visuellen Teil dieses Felds in der Konfigurationsansicht einbinden. Anschließend kann der Widget-Controller Datenpunktdaten und deren Wert über eine API-Anfrage abrufen. Nach dem Empfang kann der Wert in der Widget-Ansicht angezeigt werden.

1. Öffnen Sie `includes/WidgetForm.php` und fügen Sie das Feld `CWidgetFieldMultiSelectItem` hinzu. Dadurch kann in der Konfigurationsmaske ein Datenpunkt ausgewählt werden.

`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. Öffnen Sie `views/widget.edit.php` und fügen Sie die visuelle Komponente des Felds in die Konfigurationsansicht ein.

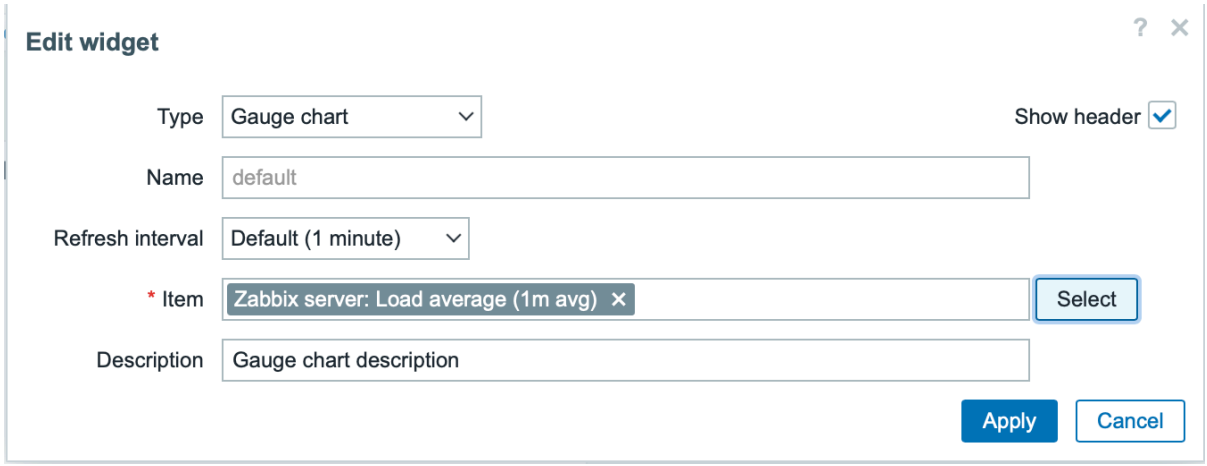
`ui/modules/lesson_gauge_chart/views/widget.edit.php`

```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. Kehren Sie zum Dashboard zurück und klicken Sie auf das Zahnradsymbol im Widget, um das Konfigurationsformular zu öffnen.
4. Das Widget-Konfigurationsformular enthält jetzt ein neues Eingabefeld *Item*. Wählen Sie den Host "Zabbix server" und den Datenpunkt "Load average (1m avg)" aus.



5. Klicken Sie im Widget-Konfigurationsformular auf *Apply*. Klicken Sie anschließend oben rechts auf *Save changes*, um das Dashboard zu speichern.
6. Öffnen und ändern Sie `actions/WidgetView.php`.

Ab jetzt ist die Datenpunkt-ID im Widget-Controller in `$this->fields_values['itemid']` verfügbar. Die Controller-Methode `doAction()` ermittelt die Datenpunktdaten (Name, Werttyp, Einheiten) mithilfe der API-Methode `item.get` und den letzten Wert des Datenpunkts mithilfe der API-Methode `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. Öffnen Sie `views/widget.view.php` und fügen Sie den Datenpunktwert zur Widget-Ansicht hinzu.

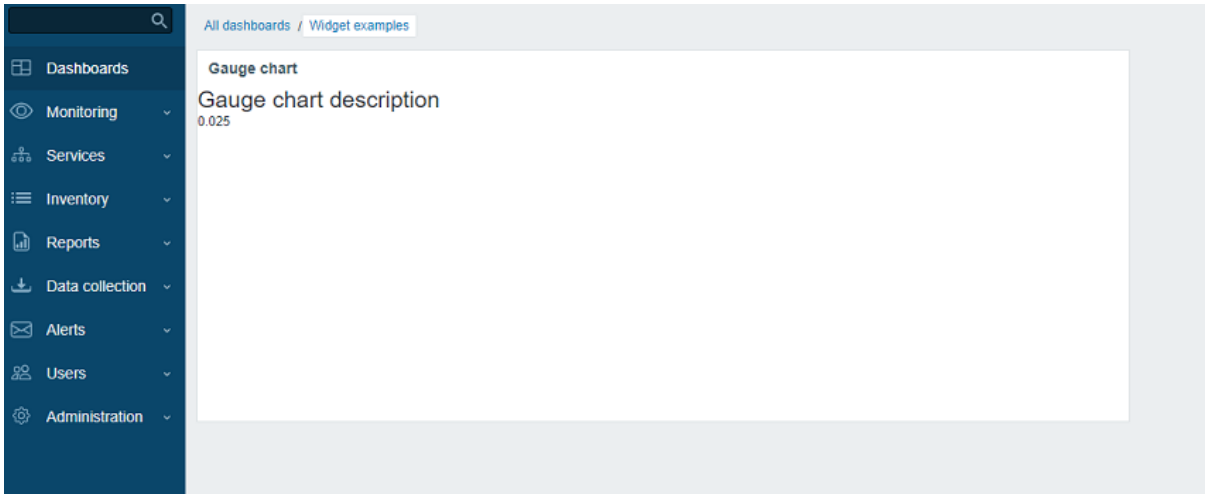
`ui/modules/lesson_gauge_chart/views/widget.view.php`

```
<?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. Aktualisieren Sie die Dashboard-Seite. Das Widget zeigt nun den neuesten Datenpunktwert an.



Erweiterte Konfigurationseinstellungen zu einer Konfigurationsansicht hinzufügen

In diesem Abschnitt erfahren Sie, wie Sie einen ein- und ausklappbaren Abschnitt *Erweiterte Konfiguration* mit optionalen Parametern hinzufügen, z. B. Farbe, Mindest- und Höchstwerte, Einheiten und das zuvor erstellte Feld *Description*.

1. Erstellen Sie im Hauptverzeichnis des Widgets `lesson_gauge_chart` eine Datei `Widget.php`, um eine neue Klasse `Widget` zu erstellen.

Die Klasse `Widget` erweitert die Basisklasse `CWidget`, um die standardmäßigen Widget-Einstellungen hinzuzufügen bzw. zu überschreiben (in diesem Fall - Übersetzungen). Das unten bereitgestellte JavaScript zeigt die Zeichenfolge "No data" an, wenn keine Daten vorhanden sind. Die Zeichenfolge "No data" ist in den Zabbix-UI-Übersetzungsdateien vorhanden.

Wenn es Widget-Konstanten gibt, wird empfohlen, diese ebenfalls in der Klasse `Widget` anzugeben.

`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. Öffnen Sie `includes/WidgetForm.php` und fügen Sie die neuen Felder *Color* (Farbauswahl), *Min* (numerisches Feld), *Max* (numerisches Feld) und *Units* (Auswahl) hinzu, und definieren Sie die Standard-Farbpalette für die Farbauswahl, damit sie in den nächsten Schritten verwendet werden kann.

`ui/modules/lesson_gauge_chart/includes/WidgetForm.php`

```
<?php

namespace Modules\LessonGaugeChart\Includes;

use Modules\LessonGaugeChart\Widget;

use Zabbix\Widgets\{
    CWidgetField,
    CWidgetForm
```

```
};

use Zabbix\Widgets\Fields\{
    CWidgetFieldColor,
    CWidgetFieldMultiSelectItem,
    CWidgetFieldNumericBox,
    CWidgetFieldSelect,
    CWidgetFieldTextBox
};

/**
 * Gauge chart widget form.
 */
class WidgetForm extends CWidgetForm {

    public const DEFAULT_COLOR_PALETTE = [
        'FF465C', 'B0AF07', '0EC9AC', '524BBC', 'ED1248', 'D1E754', '2AB5FF', '385CC7', 'EC1594', 'BAE37D',
        '6AC8FF', 'EE2B29', '3CA20D', '6F4BBC', '00A1FF', 'F3601B', '1CAE59', '45CFDB', '894BBC', '6D6D6D'
    ];

    public function addFields(): self {
        return $this
            ->addField(
                (new CWidgetFieldMultiSelectItem('itemid', _('Item')))
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
                    ->setMultiple(false)
            )
            ->addField(
                (new CWidgetFieldColor('chart_color', _('Color')))->setDefault('FF0000')
            )
            ->addField(
                (new CWidgetFieldNumericBox('value_min', _('Min')))
                    ->setDefault(0)
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
            )
            ->addField(
                (new CWidgetFieldNumericBox('value_max', _('Max')))
                    ->setDefault(100)
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
            )
            ->addField(
                (new CWidgetFieldSelect('value_units', _('Units'), [
                    Widget::UNIT_AUTO => _x('Auto', 'history source selection method'),
                    Widget::UNIT_STATIC => _x('Static', 'history source selection method')
                ]))->setDefault(Widget::UNIT_AUTO)
            )
            ->addField(
                (new CWidgetFieldTextBox('value_static_units'))
            )
            ->addField(
                new CWidgetFieldTextBox('description', _('Description'))
            );
    }
}
```

3. Öffnen Sie `views/widget.edit.php` und fügen Sie die visuellen Komponenten der Felder zur Konfigurationsansicht hinzu.

`ui/modules/lesson_gauge_chart/views/widget.edit.php`

```
<?php

/**
 * Gauge chart widget form view.
 *
```

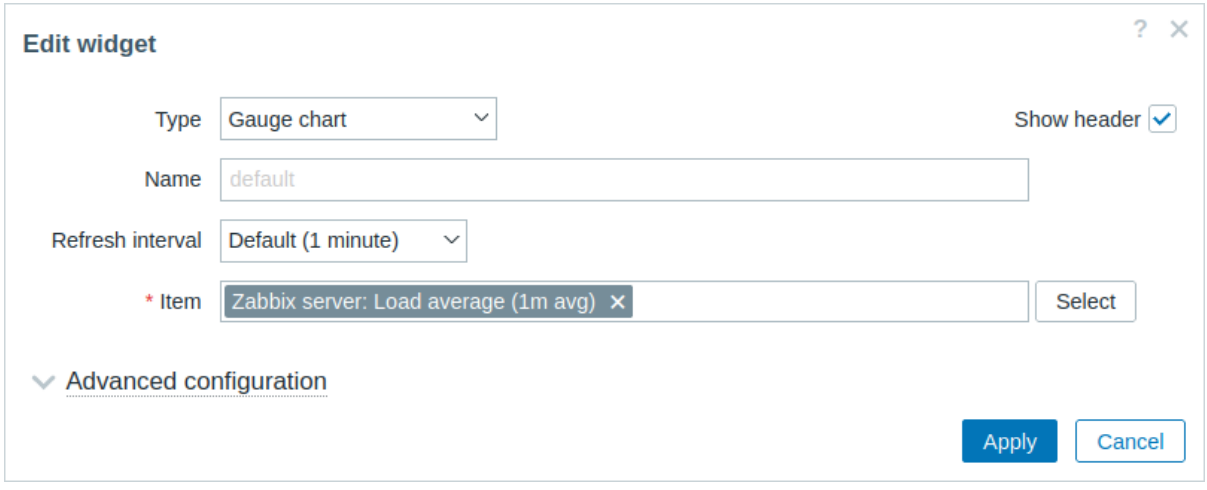
```
 * @var CView $this
 * @var array $data
 */

$lefty_units = new CWidgetFieldSelectView($data['fields']['value_units']);
$lefty_static_units = (new CWidgetFieldTextBoxView($data['fields']['value_static_units']))
    ->setPlaceholder(_('value'))
    ->setWidth(ZBX_TEXTAREA_TINY_WIDTH);

(new CWidgetFormView($data))
    ->addField(
        (new CWidgetFieldMultiSelectItemView($data['fields']['itemid']))
            ->setPopupParameter('numeric', true)
    )
    ->addFieldset(
        (new CWidgetFormFieldsetCollapsibleView(_('Advanced configuration')))
            ->addField(
                new CWidgetFieldColorView($data['fields']['chart_color'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_min'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_max'])
            )
            ->addItem([
                $lefty_units->getLabel(),
                (new CFormField([
                    $lefty_units->getView()->addClass(ZBX_STYLE_FORM_INPUT_MARGIN),
                    $lefty_static_units->getView()
                ]))
            ])
            ->addField(
                new CWidgetFieldTextBoxView($data['fields']['description'])
            )
    )
    ->show();
```

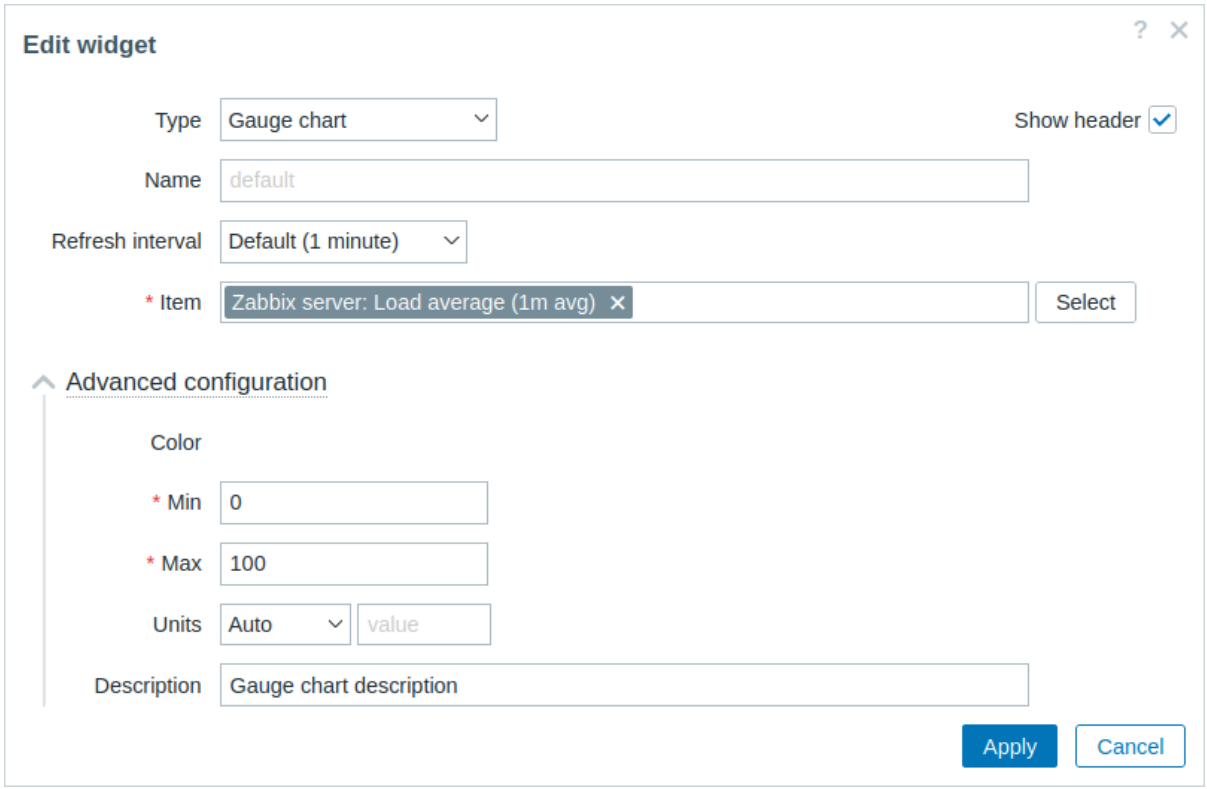
Note:
Die Methode `addField()` der Klasse `CWidgetFormView` nimmt als zweiten Parameter eine CSS-Klassenzeichenfolge entgegen.

4. Kehren Sie zum Dashboard zurück, wechseln Sie in den Bearbeitungsmodus und klicken Sie auf das Zahnradsymbol im Widget, um das Widget-Konfigurationsformular zu öffnen. Das Widget-Konfigurationsformular enthält nun einen neuen ein- und ausklappbaren Abschnitt *Erweiterte Konfiguration*.



5. Klappen Sie den Abschnitt *Erweiterte Konfiguration* auf, um zusätzliche Widget-Konfigurationsfelder anzuzeigen. Beachten

Sie, dass das Feld *Color* noch keine Farbauswahl hat. Dies liegt daran, dass die Farbauswahl mit JavaScript initialisiert werden muss, das im nächsten Abschnitt hinzugefügt wird - *JavaScript zum Widget hinzufügen*.



JavaScript zum Widget hinzufügen

In diesem Abschnitt lernen Sie, wie Sie ein Gauge-Diagramm hinzufügen - erstellt mit JavaScript -, das anzeigt, ob der letzte Wert normal oder zu hoch/zu niedrig ist.

1. Erstellen Sie eine Datei `widget.edit.js.php` im Verzeichnis `views`.

JavaScript ist für die Initialisierung des Farbwählers in der Konfigurationsansicht verantwortlich.

`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. Öffnen Sie `views/widget.edit.php` und fügen Sie die Datei `widget.edit.js.php` mit dem JavaScript zur Konfigurationsansicht hinzu. Verwenden Sie dazu die Methode `includeJsFile()`. Um Inline-JavaScript hinzuzufügen, verwenden Sie die Methode `addJavaScript()`.

`ui/modules/lesson_gauge_chart/views/widget.edit.php`

```
<?php

/**
 * Gauge chart widget form view.
 *
 * @var CView $this
 * @var array $data
 */

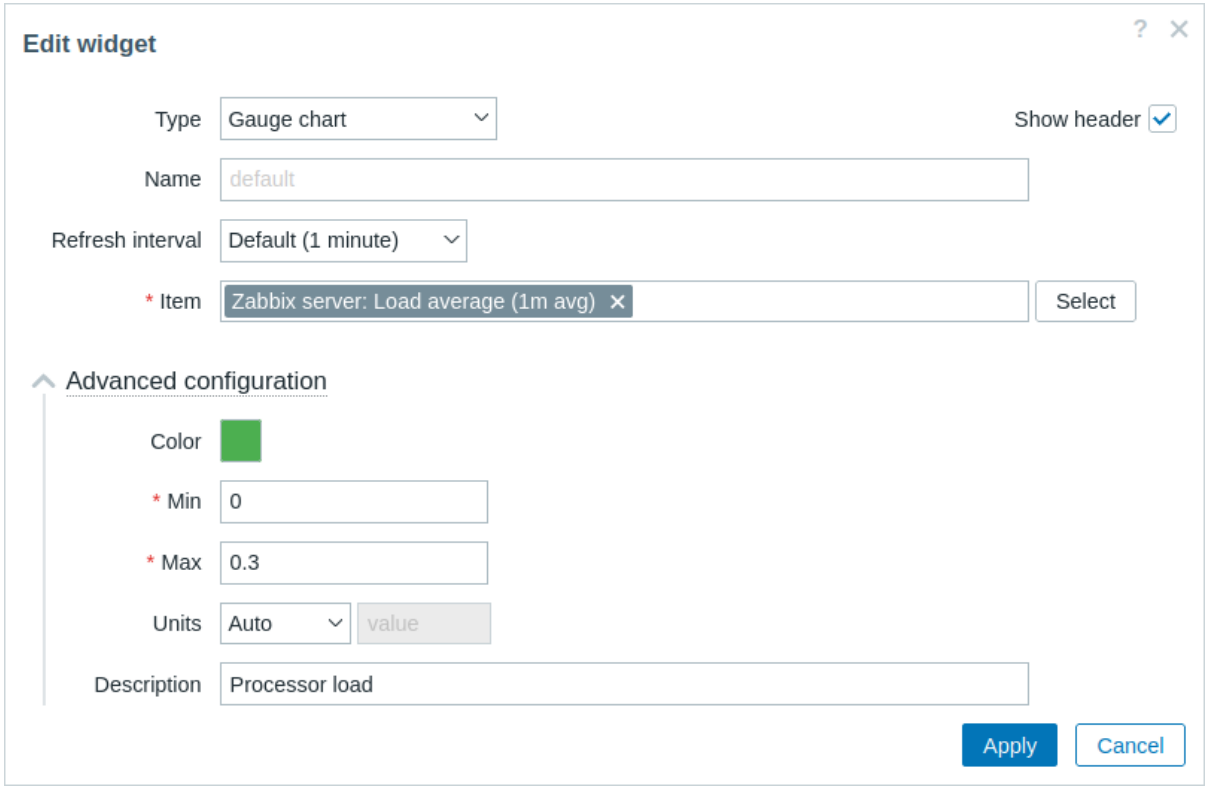
use Modules\LessonGaugeChart\Includes\WidgetForm;

$lefty_units = new CWidgetFieldSelectView($data['fields']['value_units']);
$lefty_static_units = (new CWidgetFieldTextBoxView($data['fields']['value_static_units']))
    ->setPlaceholder(_('value'))
    ->setWidth(ZBX_TEXTAREA_TINY_WIDTH);

(new CWidgetFormView($data))
    ->addField(
        (new CWidgetFieldMultiSelectItemView($data['fields']['itemid']))
            ->setPopupParameter('numeric', true)
    )
    ->addFieldset(
        (new CWidgetFormFieldsetCollapsibleView(_('Advanced configuration')))
            ->addField(
                new CWidgetFieldColorView($data['fields']['chart_color'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_min'])
            )
            ->addField(
                new CWidgetFieldNumericBoxView($data['fields']['value_max'])
            )
            ->addItem([
                $lefty_units->getLabel(),
                (new CFormField([
                    $lefty_units->getView()->addClass(ZBX_STYLE_FORM_INPUT_MARGIN),
                    $lefty_static_units->getView()
                ]))
            ])
            ->addField(
                new CWidgetFieldTextBoxView($data['fields']['description'])
            )
    )
    ->includeJsFile('widget.edit.js.php')
    ->addJavaScript('widget_lesson_gauge_chart_form.init('.json_encode([
        'color_palette' => WidgetForm::DEFAULT_COLOR_PALETTE
    ], JSON_THROW_ON_ERROR).');')
    ->show();
```

3. Kehren Sie zum Dashboard zurück und klicken Sie auf das Zahnradsymbol im Widget, um das Widget-Konfigurationsformular zu öffnen. Erweitern Sie nun den Abschnitt *Advanced configuration*, um den initialisierten Farbwähler anzuzeigen. Füllen Sie

die Felder mit Werten aus und wählen Sie eine Farbe für das Gauge-Diagramm aus.



4. Klicken Sie im Widget-Konfigurationsformular auf *Apply*. Klicken Sie anschließend oben rechts auf *Save changes*, um das Dashboard zu speichern.
5. Öffnen Sie `actions/WidgetView.php` und aktualisieren Sie den Controller.

Die Eigenschaft `$this->fields_values` enthält nun die Werte aller Felder aus *Advanced configuration*. Vervollständigen Sie den Controller, damit die Konfiguration und der ausgewählte Item-Wert an die Widget-Ansicht übergeben werden können.

`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. Öffnen und ändern Sie `views/widget.view.php`.

Sie müssen einen Container für das Gauge-Diagramm erstellen, das Sie in den nächsten Schritten zeichnen werden, sowie einen Container für die Beschreibung.

Um Werte als JSON-Objekt an JavaScript zu übergeben, verwenden Sie die Methode `setVar()`.

`ui/modules/lesson_gauge_chart/views/widget.view.php`

```
<?php

/**
 * Gauge chart widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem([
        (new CDiv())->addClass('chart'),
        $data['fields_values']['description']
            ? (new CDiv($data['fields_values']['description']))->addClass('description')
            : null
    ])
    ->setVar('history', $data['history'])
    ->setVar('fields_values', $data['fields_values'])
    ->show();
```

7. Erstellen Sie im Verzeichnis `lesson_gauge_chart` ein neues Verzeichnis `assets`. Dieses Verzeichnis wird zum Speichern von JavaScript, CSS und möglicherweise weiteren Assets wie Schriftarten oder Bildern verwendet.
8. Erstellen Sie für JavaScript der Widget-Ansicht im Verzeichnis `assets` ein Verzeichnis `js`.
9. Erstellen Sie im Verzeichnis `assets/js` eine Datei `class.widget.js`.

Diese JavaScript-Widget-Klasse erweitert die Basis-JavaScript-Klasse aller Dashboard-Widgets - `CWidget`.

Das Dashboard ist auf eine korrekte Implementierung eines Widgets angewiesen und kommuniziert alle relevanten Informationen an das Widget über die entsprechenden JavaScript-Methoden. Außerdem erwartet das Dashboard, dass das Widget Ereignisse erzeugt, wenn eine Interaktion stattfindet. Daher enthält die Klasse `CWidget` eine Reihe von Methoden mit der Standardimplementierung des Widget-Verhaltens, die durch Vererbung angepasst werden können.

In diesem Fall ist eine gewisse Anpassung erforderlich, daher wird die benutzerdefinierte Logik für das folgende Widget-Verhalten implementiert:

- Widget-Initialisierung, die für die Definition des Anfangszustands des Widgets verantwortlich ist (siehe die Methode `onInitialize()`);
- Anzeige des Widget-Inhalts (also das Zeichnen des Gauge-Diagramms), wenn der Widget-Aktualisierungsvorgang erfolgreich und ohne Fehler war (siehe die Methode `processUpdateResponse(response)` und die zugehörigen Methoden `_resizeChart()` und `_updatedChart()`)
- Größenänderung des Widgets (siehe die Methode `onResize()` und die zugehörige Methode `_resizeChart()`)

Für andere Aspekte des Gauge-Diagramm-Widgets wird die Standardimplementierung für das Widget-Verhalten verwendet. Weitere Informationen zu den JavaScript-Methoden der Klasse `CWidget` finden Sie unter: JavaScript.

Da dieses JavaScript für die Widget-Ansicht erforderlich ist, sollte es mit der Dashboard-Seite geladen werden. Um das Laden von JavaScript zu aktivieren, müssen Sie die Parameter `assets/js` und `js_class` in der Datei `manifest.json` wie in Schritt 10 gezeigt aktualisieren.

`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. Öffnen Sie `manifest.json` und fügen Sie hinzu:

- den Dateinamen (`class.widget.js`) zum Array im Abschnitt `assets/js`;
- den Klassennamen (`WidgetLessonGaugeChart`) zum Parameter `js_class` im Abschnitt `widget`.

Die Klasse `WidgetLessonGaugeChart` wird nun automatisch mit dem Dashboard geladen.

`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"]
    }
}
```

CSS-Stile zum Widget hinzufügen

In diesem Abschnitt erfahren Sie, wie Sie benutzerdefinierte CSS-Stile hinzufügen, damit das Widget ansprechender aussieht.

1. Erstellen Sie für die Widget-Stile im Verzeichnis `assets` ein neues Verzeichnis `css`.
2. Erstellen Sie im Verzeichnis `assets/css` die Datei `widget.css`. Um Widget-Elemente zu gestalten, verwenden Sie den Selektor `div.dashboard-widget-{widget id}`. Um CSS für das gesamte Widget zu konfigurieren, verwenden Sie den Selektor `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;
}

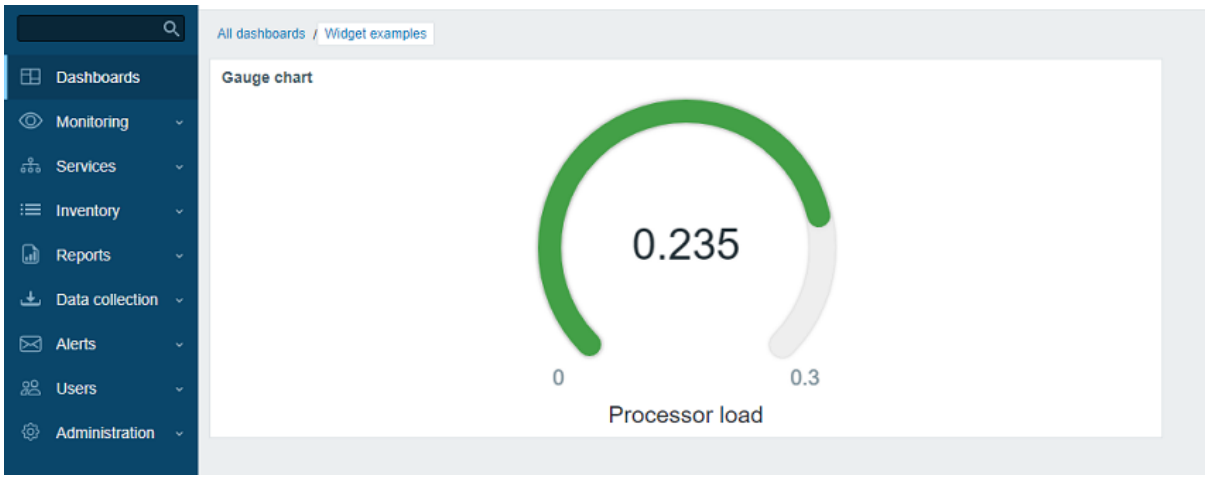
.dashboard-grid-widget-hidden-header div.dashboard-widget-lesson_gauge_chart .chart {
    margin-top: 8px;
}
```

3. Öffnen Sie `manifest.json` und fügen Sie den Namen der CSS-Datei (`widget.css`) dem Array im Abschnitt `assets/css` hinzu. Dadurch können die in `widget.css` definierten CSS-Stile zusammen mit der Dashboard-Seite geladen werden.

`ui/modules/lesson_gauge_chart/manifest.json`

```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
    "type": "widget",
    "name": "Gauge chart",
    "namespace": "LessonGaugeChart",
    "version": "1.0",
    "author": "Zabbix",
    "actions": {
        "widget.lesson_gauge_chart.view": {
            "class": "WidgetView"
        }
    },
    "widget": {
        "js_class": "WidgetLessonGaugeChart"
    },
    "assets": {
        "css": ["widget.css"],
        "js": ["class.widget.js"]
    }
}
```

4. Aktualisieren Sie die Dashboard-Seite, um die fertige Version des Widgets anzuzeigen.



Beispiele

Dieser Abschnitt enthält Dateien mit Beispielmodulen und Widgets, die Sie als Grundlage für Ihre eigenen Module verwenden können.

Um ein Modul zu verwenden:

1. Laden Sie das ZIP-Archiv herunter.
2. Entpacken Sie den Inhalt in ein separates Verzeichnis im Ordner */zabbix/ui/modules*.
3. Registrieren Sie das Modul im Zabbix-Frontend.

Modulbeispiel

- Erteilen Sie bei der Erstellung einer Host-Gruppe Leseberechtigungen für konfigurierte Benutzergruppen - hg_auto_perm.zip

Beispiele für Widgets

- Minimales Widget - widget_min.zip
- "Hallo, Welt"-Widget nur mit CSS - hello_world_css.zip
- Hallo, Welt"-Widget nur mit JavaScript - hello_world_js.zip
- "Hallo, Welt"-Widget mit PHP - hello_world_php.zip

Note:
Sie können auch Zabbix native widgets als Beispiel verwenden.

Änderungen an der Erweiterungsentwicklung

Änderungen von 7.4 zu 8.0

Zabbix-Manpages

Referenzdokumentation für Zabbix-Befehlszeilenwerkzeuge und -Dienste mit einer Auflistung der verfügbaren Befehle, Optionen und Anwendungsbeispiele.

zabbix_agent2

```
ZABBIX_AGENT2(8)            Systemadministrator-Handbuch           ZABBIX_AGENT2(8)

NAME
       zabbix_agent2 - Zabbix Agent 2

ÜBERSICHT
       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

BESCHREIBUNG
       zabbix_agent2  ist  eine  Anwendung zur Überwachung von Parametern ver-
       schiedener Dienste.

OPTIONEN
       -c, --config config-file
              Die alternative config-file anstelle der Standarddatei verwenden.

       -R, --runtime-control runtime-option
              Administrative Funktionen entsprechend runtime-option ausführen.

           Laufzeitsteuerungsoptionen
               userparameter_reload
                   Benutzerparameter aus der Konfigurationsdatei neu laden

               log_level_increase
                   Log-Level erhöhen

               log_level_decrease
                   Log-Level verringern

               help
                   Verfügbare Laufzeitsteuerungsoptionen auflisten

               metrics
                   Verfügbare Metriken auflisten
               version
                   Version anzeigen
```

-p, --print
Bekannte Datenpunkte ausgeben und beenden. Für jeden Datenpunkt werden entweder allgemeine Standardwerte verwendet oder spezifische Standardwerte für Tests bereitgestellt. Diese Standardwerte werden in eckigen Klammern als Parameter des Datenpunktschlüssels aufgeführt. Zurückgegebene Werte werden in eckige Klammern eingeschlossen und mit dem Typ des zurückgegebenen Werts versehen, getrennt durch ein Pipe-Zeichen. Bei Benutzerparametern ist der Typ immer t, da der Agent nicht alle möglichen Rückgabewerte bestimmen kann. Datenpunkte, die als funktionierend angezeigt werden, funktionieren nicht zwangsläufig vom Zabbix Server oder von zabbix_get aus bei der Abfrage eines laufenden Agent-Daemons, da Berechtigungen oder die Umgebung unterschiedlich sein können. Die Typen der zurückgegebenen Werte sind:

d Zahl mit Dezimalteil.

m Nicht unterstützt. Dies kann durch die Abfrage eines Datenpunkts verursacht werden, der nur im aktiven Modus funktioniert, wie ein Datenpunkt zur Log-Überwachung oder ein Datenpunkt, der mehrere erfasste Werte benötigt. Berechtigungsprobleme oder fehlerhafte Benutzerparameter können ebenfalls zum Status „nicht unterstützt“ führen.

s Text. Maximale Länge nicht begrenzt.

t Text. Gleich wie s.

u Vorzeichenlose Ganzzahl.

-t, --test item-key
Einzelnen Datenpunkt testen und beenden. Siehe --print für die Beschreibung der Ausgabe.

-v, --verbose
Ausführliche Ausgabe aktivieren beim Testen eines Datenpunkts oder beim Ausgeben bekannter Datenpunkte.

-T, --test-config
Konfigurationsdatei validieren und beenden.

-h, --help
Diese Hilfe anzeigen und beenden.

-V, --version
Versionsinformationen ausgeben und beenden.

DATEIEN
/usr/local/etc/zabbix_agent2.conf
Standardpfad der Konfigurationsdatei von Zabbix Agent 2 (falls nicht während der Kompilierung geändert).

SIEHE AUCH
Dokumentation

zabbix_agentd(8), zabbix_get(1), zabbix_proxy(8), zabbix_sender(1), zabbix_server(8), zabbix_js(1), zabbix_web_service(8)

zabbix_agentd

```
ZABBIX_AGENTD(8)            Systemadministrator-Handbuch           ZABBIX_AGENTD(8)

NAME
       zabbix_agentd - Zabbix-Agent-Daemon

ÜBERSICHT
       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

BESCHREIBUNG
       zabbix_agentd ist ein Daemon zur Überwachung verschiedener Server-Parameter.

OPTIONEN
       -c, --config config-file
              Die alternative config-file anstelle der Standarddatei verwenden.

       -f, --foreground
              Zabbix-Agent im Vordergrund ausführen.

       -R, --runtime-control runtime-option
              Administrative Funktionen entsprechend runtime-option ausführen.

           Laufzeitsteuerungsoptionen
               userparameter_reload
                   Benutzerparameter aus der Konfigurationsdatei neu laden

               log_level_increase[=target]
                   Log-Level erhöhen; betrifft alle Prozesse, wenn target nicht
                   angegeben ist

               log_level_decrease[=target]
                   Log-Level verringern; betrifft alle Prozesse, wenn target nicht
                   angegeben ist

           Ziele der Log-Level-Steuerung
               process-type
                   Alle Prozesse des angegebenen Typs (aktive Prüfungen, collector,
                   listener)

               process-type,N
                   Prozesstyp und Nummer (z. B. listener,3)
               pid Prozesskennung, bis zu 65535. Bei größeren Werten target
                   als "process-type,N" angeben

       -p, --print
              Bekannte Datenpunkte ausgeben und beenden. Für jeden Datenpunkt werden entweder allgemeine
              Standardwerte verwendet oder spezifische Standardwerte für Tests bereitgestellt.
              Diese Standardwerte werden in eckigen Klammern als Datenpunktschlüsselparameter
              aufgeführt. Zurückgegebene Werte werden in eckige Klammern eingeschlossen und mit
              dem Typ des zurückgegebenen Werts versehen, getrennt durch ein Pipe-Zeichen.
              Für Benutzerparameter ist der Typ immer t, da der Agent nicht alle möglichen
              Rückgabewerte bestimmen kann. Als funktionsfähig angezeigte Datenpunkte sind keine
              Garantie dafür, dass sie vom Zabbix-Server oder von zabbix_get bei der Abfrage
              eines laufenden Agent-Daemons funktionieren, da Berechtigungen oder Umgebung
```

```
              unterschiedlich sein können. Die Typen der zurückgegebenen Werte sind:

           d   Zahl mit Dezimalteil.

           m   Nicht unterstützt. Dies kann dadurch verursacht werden, dass ein Datenpunkt
               abgefragt wird, der nur im aktiven Modus funktioniert, wie ein Datenpunkt zur
               Log-Überwachung, oder ein Datenpunkt, der mehrere gesammelte Werte benötigt.
               Berechtigungsprobleme oder fehlerhafte Benutzerparameter können ebenfalls zum
               Status "nicht unterstützt" führen.

           s   Text. Maximale Länge nicht begrenzt.

           t   Text. Gleich wie s.

           u   Vorzeichenlose Ganzzahl.

       -t, --test item-key
              Einzelnen Datenpunkt testen und beenden. Siehe --print für die Ausgabebeschreibung.

       -T, --test-config
              Konfigurationsdatei validieren und beenden.

       -h, --help
              Diese Hilfe anzeigen und beenden.

       -V, --version
              Versionsinformationen ausgeben und beenden.

DATEIEN
       /usr/local/etc/zabbix_agentd.conf
              Standardpfad der Zabbix-Agent-Konfigurationsdatei (falls nicht zur
              Kompilierzeit geändert).

SIEHE AUCH
       Dokumentation

       zabbix_get(1),  zabbix_proxy(8),  zabbix_sender(1),   zabbix_server(8),
       zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix                            2022-01-06                  ZABBIX_AGENTD(8)
```

zabbix_get

```
ZABBIX_GET(1)               Allgemeines Befehlshandbuch          ZABBIX_GET(1)

NAME
       zabbix_get - Zabbix-get-Dienstprogramm

ÜBERSICHT
       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-sub-
       ject  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

BESCHREIBUNG

zabbix_get ist ein Befehlszeilenprogramm zum Abrufen von Daten vom Zabbix Agent.

OPTIONEN

-s, --host host-name-or-IP
Geben Sie den Hostnamen oder die IP-Adresse eines Hosts an.

-p, --port port-number
Geben Sie die Portnummer des auf dem Host laufenden Agent an. Der Standardwert ist 10050.

-I, --source-address IP-address
Geben Sie die Quell-IP-Adresse an.

-t, --timeout seconds
Geben Sie das Timeout an. Gültiger Bereich: 1-600 Sekunden (Standard: 30)

-k, --key item-key
Geben Sie den Schlüssel des Datenpunkts an, dessen Wert abgerufen werden soll.

-P, --protocol value
Das für die Kommunikation mit dem Agent verwendete Protokoll. Werte:

auto — Verbindung über das JSON-Protokoll herstellen, bei Fehlschlag auf das Klartextprotokoll zurückfallen und erneut versuchen (Standard)

json — Verbindung über das JSON-Protokoll herstellen

plaintext — Verbindung über das Klartextprotokoll herstellen, bei dem nur der Schlüssel des Datenpunkts gesendet wird (Versionen 6.4.x und älter)

--tls-connect value
Wie die Verbindung zum Agent hergestellt wird. Werte:

unencrypted — Verbindung ohne Verschlüsselung herstellen (Standard)

psk — Verbindung über TLS und einen Pre-Shared Key herstellen

cert — Verbindung über TLS und ein Zertifikat herstellen

--tls-ca-file CA-file
Vollständiger Pfadname zu einer Datei mit den Zertifikaten der obersten CA(s) zur Verifizierung des Peer-Zertifikats.

--tls-crl-file CRL-file
Vollständiger Pfadname zu einer Datei mit widerrufenen Zertifikaten.

--tls-agent-cert-issuer cert-issuer
Zulässiger Aussteller des Agent-Zertifikats.

```
       --tls-agent-cert-subject cert-subject
              Zulässiger Betreff des Agent-Zertifikats.

       --tls-cert-file cert-file
              Vollständiger Pfadname zu einer Datei, die das Zertifikat oder
              die Zertifikatskette enthält.

       --tls-key-file key-file
              Vollständiger Pfadname zu einer Datei mit dem privaten Schlüssel.

       --tls-psk-identity PSK-identity
              PSK-Identity-Zeichenfolge.

       --tls-psk-file PSK-file
              Vollständiger Pfadname zu einer Datei mit dem Pre-Shared Key.

       --tls-cipher13 cipher-string
              Chiffre-Zeichenfolge für OpenSSL 1.1.1 oder neuer für TLS 1.3.
              Überschreibt die Standardkriterien für die Auswahl der Chiffren-
              suite. Diese Option ist nicht verfügbar, wenn die OpenSSL-Version
              kleiner als 1.1.1 ist.

       --tls-cipher cipher-string
              GnuTLS-Prioritätszeichenfolge (für TLS 1.2 und höher) oder OpenSSL-
              Chiffre-Zeichenfolge (nur für TLS 1.2). Überschreibt die Standard-
              kriterien für die Auswahl der Chiffrensuite.

       -h, --help
              Diese Hilfe anzeigen und beenden.

       -V, --version
              Versionsinformationen ausgeben und beenden.

EXIT-STATUS
       Der Exit-Status ist 0, wenn zabbix_get erfolgreich einen Wert vom Zabbix
       Agent abruft. Wenn das Abrufen des Werts fehlschlägt, ist der Exit-Sta-
       tus 1.

BEISPIELE
       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  op-
       erations,O=Example  Corp,DC=example,DC=com"  --tls-cert-file /home/zab-
       bix/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

SIEHE AUCH
       Dokumentation

       zabbix_agentd(8), zabbix_proxy(8), zabbix_sender(1),  zabbix_server(8),
       zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix                          2022-01-06                    ZABBIX_GET(1)
```

zabbix_js

```
ZABBIX_JS(1)               Allgemeines Benutzerhandbuch              ZABBIX_JS(1)
```

NAME
       zabbix_js - Zabbix-JS-Dienstprogramm

ÜBERSICHT
       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

BESCHREIBUNG
       zabbix_js ist ein Befehlszeilen-Dienstprogramm, das zum Testen
       eingebetteter Skripte verwendet werden kann.

OPTIONEN
       -s, --script script-file
              Geben Sie den Dateinamen des auszuführenden Skripts an. Wenn '-'
              als Dateiname angegeben wird, wird das Skript von stdin gelesen.

       -p, --param input-param
              Geben Sie den an das Skript übergebenen Eingabeparameter an. Der
              Parameter wird unverändert als Zeichenkette übergeben und
              steht im Skript über die Variable 'value' zur Verfügung. Gegen-
              seitig ausschließend mit -i.

       -i, --input input-file
              Geben Sie eine Datei an, die den an das Skript übergebenen Ein-
              gabeparameter enthält. Der Dateiinhalt wird unverändert als Zei-
              chenkette übergeben und steht im Skript über die Variable 'value'
              zur Verfügung. Wenn '-' als Dateiname angegeben wird, wird die
              Eingabe von stdin gelesen. Gegenseitig ausschließend mit -p.

       -w, --webdriver url
              Geben Sie die webdriver-URL an.

       -l, --loglevel log-level
              Geben Sie die Protokollierungsstufe an (Bereich: 0-5).

       -t, --timeout timeout
              Geben Sie das Timeout in Sekunden an. Gültiger Bereich:
              1-600 Sekunden (Standard: 10)

       -h, --help
              Diese Hilfe anzeigen und beenden.

       -V, --version
              Versionsinformationen ausgeben und beenden.

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

SIEHE AUCH
       Dokumentation

       zabbix_agentd(8), zabbix_proxy(8), zabbix_sender(1), zabbix_server(8),
       zabbix_get(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix   2022-01-06   ZABBIX_JS(1)

zabbix_proxy

NAME

zabbix_proxy - Zabbix-Proxy-Daemon

SYNOPSIS

```
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 ist ein Daemon, der Monitoring-Daten von Geräten sammelt und an den Zabbix Server sendet.

OPTIONS

-c, --config config-file
: Verwendet die alternative config-file anstelle der Standarddatei.

-f, --foreground
: Führt den Zabbix-Proxy im Vordergrund aus.

-R, --runtime-control runtime-option
: Führt administrative Funktionen gemäß runtime-option aus.

Runtime control options

config_cache_reload
: Lädt den Konfigurationscache neu. Wird ignoriert, wenn der Cache derzeit geladen wird. Der aktive Zabbix-Proxy stellt eine Verbindung zum Zabbix Server her und fordert Konfigurationsdaten an. Die Standard-Konfigurationsdatei (sofern die Option -c nicht angegeben ist) wird verwendet, um die PID-Datei zu finden, und das Signal wird an den in der PID-Datei aufgeführten Prozess gesendet.

snmp_cache_reload
: Lädt den SNMP-Cache neu.

housekeeper_execute
: Führt den Housekeeper aus. Wird ignoriert, wenn der Housekeeper derzeit ausgeführt wird.

diaginfo[=section]
: Protokolliert interne Diagnoseinformationen des angegebenen Abschnitts. section kann historycache, preprocessing, locks sein. Standardmäßig werden Diagnoseinformationen aller Abschnitte protokolliert.

prof_enable[=target]
: Aktiviert Profiling, betrifft alle Prozesse, wenn target nicht angegeben ist

prof_disable[=target]
: Deaktiviert Profiling, betrifft alle Prozesse, wenn target nicht angegeben ist

log_level_increase[=target]
: Erhöht das Protokollierungsniveau, betrifft alle Prozesse, wenn target nicht angegeben ist.

log_level_decrease[=target]
: Verringert das Protokollierungsniveau, betrifft alle

Prozesse, wenn target nicht angegeben ist.

Log level control targets
process-type
Alle Prozesse des angegebenen Typs (availability manager, browser poller, configuration syncer, data sender, discovery manager, history syncer, housekeeper, http poller, icmp pinger, internal poller, ipmi manager, ipmi poller, java poller, odbc poller, poller, agent poller, http agent poller, snmp poller, preprocessing manager, preprocessing worker, self-monitoring, snmp trapper, supervisor, task manager, trapper, unreachable poller, vmware collector)

process-type,N
Prozesstyp und Nummer (z. B. poller,3)

pid Prozesskennung

Profiling control targets
process-type
Alle Prozesse des angegebenen Typs (availability manager, browser poller, configuration syncer, data sender, discovery manager, history syncer, housekeeper, http poller, icmp pinger, internal poller, 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
Prozesstyp und Nummer (z. B. history syncer,1)

pid Prozesskennung

scope
(rwlock, mutex, processing) kann mit process-type verwendet werden (z. B. history syncer,1,processing)

-T, --test-config
Validiert die Konfigurationsdatei und beendet sich.

-h, --help
Zeigt diese Hilfe an und beendet sich.

-V, --version
Gibt Versionsinformationen aus und beendet sich.

FILES
/usr/local/etc/zabbix_proxy.conf
Standardpfad der Zabbix-Proxy-Konfigurationsdatei (falls nicht zur Kompilierzeit geändert).

SEE ALSO
Documentation

zabbix_agentd(8), zabbix_get(1), zabbix_sender(1), zabbix_server(8), zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

zabbix_sender

```
ZABBIX_SENDER(1)            Allgemeine Befehle            ZABBIX_SENDER(1)

NAME
       zabbix_sender - Zabbix-Sender-Dienstprogramm

SYNOPSIS
       zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] -s
       host -k key -o value
       zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] [-s
       host] [-T] [-N] [-r] -i input-file
       zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
       [-t timeout] [-s host] -k key -o value
       zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
       [-t timeout] [-s host] [-T] [-N] [-r] -i input-file
       zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] -s
       host --tls-connect cert --tls-ca-file CA-file [--tls-crl-file CRL-file]
       [--tls-server-cert-issuer cert-issuer] [--tls-server-cert-subject
       cert-subject] --tls-cert-file cert-file --tls-key-file key-file
       [--tls-cipher13 cipher-string] [--tls-cipher cipher-string] -k key -o
       value
       zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] [-s
       host] --tls-connect cert --tls-ca-file CA-file [--tls-crl-file
       CRL-file] [--tls-server-cert-issuer cert-issuer]
       [--tls-server-cert-subject cert-subject] --tls-cert-file cert-file
       --tls-key-file key-file [--tls-cipher13 cipher-string] [--tls-cipher
       cipher-string] [-T] [-N] [-r] -i input-file
       zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
       [-t timeout] [-s host] --tls-connect cert --tls-ca-file CA-file
       [--tls-crl-file CRL-file] [--tls-server-cert-issuer cert-issuer]
       [--tls-server-cert-subject cert-subject] --tls-cert-file cert-file
       --tls-key-file key-file [--tls-cipher13 cipher-string] [--tls-cipher
       cipher-string] -k key -o value
       zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
       [-t timeout] [-s host] --tls-connect cert --tls-ca-file CA-file
       [--tls-crl-file CRL-file] [--tls-server-cert-issuer cert-issuer]
       [--tls-server-cert-subject cert-subject] --tls-cert-file cert-file
       --tls-key-file key-file [--tls-cipher13 cipher-string] [--tls-cipher
       cipher-string] [-T] [-N] [-r] -i input-file
       zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] -s
       host --tls-connect psk --tls-psk-identity PSK-identity --tls-psk-file
       PSK-file [--tls-cipher13 cipher-string] [--tls-cipher cipher-string] -k
       key -o value
       zabbix_sender [-v] -z server [-p port] [-I IP-address] [-t timeout] [-s
       host] --tls-connect psk --tls-psk-identity PSK-identity --tls-psk-file
       PSK-file [--tls-cipher13 cipher-string] [--tls-cipher cipher-string]
       [-T] [-N] [-r] -i input-file
       zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
       [-t timeout] [-s host] --tls-connect psk --tls-psk-identity PSK-iden-
       tity --tls-psk-file PSK-file [--tls-cipher13 cipher-string] [--tls-ci-
       pher cipher-string] -k key -o value
       zabbix_sender [-v] -c config-file [-z server] [-p port] [-I IP-address]
       [-t timeout] [-s host] --tls-connect psk --tls-psk-identity PSK-iden-
       tity --tls-psk-file PSK-file [--tls-cipher13 cipher-string] [--tls-ci-
       pher cipher-string] [-T] [-N] [-r] -i input-file
       zabbix_sender -h
       zabbix_sender -V

DESCRIPTION
       zabbix_sender ist ein Befehlszeilen-Dienstprogramm zum Senden von
```

Überwachungsdaten an den Zabbix-Server oder Proxy. Auf dem Zabbix-Server sollte ein Datenpunkt vom Typ „Zabbix trapper“ mit dem entsprechenden Schlüssel erstellt werden. Beachten Sie, dass eingehende Werte nur von Hosts akzeptiert werden, die im Feld „Allowed hosts“ dieses Datenpunkts angegeben sind.

OPTIONS

-c, --config config-file
: Verwendet config-file. Zabbix sender liest die Serverdetails aus der agentd-Konfigurationsdatei. Standardmäßig liest Zabbix sender keine Konfigurationsdatei. Unterstützt werden nur die Parameter Hostname, ServerActive, SourceIP, TLSConnect, TLSCAFile, TLSCRLFile, TLSServerCertIssuer, TLSServerCertSubject, TLSCertFile, TLSKeyFile, TLSPSKIdentity und TLSPSKFile. Ein über den Parameter HostnameItem definierter Hostname wird nicht übernommen. In diesem Fall muss der Hostname über die Befehlszeile angegeben werden (siehe Option -s). Alle in dem Agent-Konfigurationsparameter ServerActive definierten Adressen werden zum Senden von Daten verwendet. Wenn das Senden von Batchdaten an eine Adresse fehlschlägt, werden die folgenden Batches nicht an diese Adresse gesendet.

-z, --zabbix-server server
: Hostname oder IP-Adresse des Zabbix-Servers. Wenn ein Host von einem Proxy überwacht wird, sollte stattdessen der Hostname oder die IP-Adresse des Proxy verwendet werden. Bei Verwendung zusammen mit --config werden die Einträge des Parameters ServerActive in der agentd-Konfigurationsdatei überschrieben.

-p, --port port
: Gibt die Portnummer des auf dem Server ausgeführten Zabbix-Server-Trappers an. Der Standardwert ist 10051. Bei Verwendung zusammen mit --config werden die Porteinträge des Parameters ServerActive in der agentd-Konfigurationsdatei überschrieben.

-I, --source-address IP-address
: Gibt die Quell-IP-Adresse an. Bei Verwendung zusammen mit --config wird der Parameter SourceIP in der Agent-Konfigurationsdatei überschrieben.

-t, --timeout seconds
: Gibt das Timeout an. Gültiger Bereich: 1-300 Sekunden (Standardwert: 60).

-s, --host host
: Gibt den Hostnamen an, zu dem der Datenpunkt gehört (wie im Zabbix-Frontend registriert). Die IP-Adresse und der DNS-Name des Hosts funktionieren nicht. Bei Verwendung zusammen mit --config wird der in der Agent-Konfigurationsdatei angegebene Parameter Hostname überschrieben.

-k, --key key
: Gibt den Datenpunktschlüssel an, an den der Wert gesendet werden soll.

-o, --value value
: Gibt den Datenpunktwert an.

-i, --input-file input-file
: Lädt Werte aus der Eingabedatei. Geben Sie als <input-file> - an, um Werte aus der Standardeingabe zu lesen. Jede Zeile der Datei enthält durch Leerzeichen getrennt: <hostname> <key> <value>. Jeder Wert muss in einer eigenen Zeile angegeben werden. Jede

Zeile muss drei durch Leerzeichen getrennte Einträge enthalten: <hostname> <key> <value>, wobei „hostname“ der Name des überwachten Hosts ist, wie er im Zabbix-Frontend registriert ist, „key“ der Zieldatenpunktschlüssel und „value“ der zu sendende Wert. Geben Sie als <hostname> - an, um den Hostnamen aus der Agent-Konfigurationsdatei oder aus dem Argument --host zu verwenden.

Beispiel für eine Zeile einer Eingabedatei:

```
 "Linux DB3" db.connections 43
```

Der Werttyp muss in der Datenpunktkonfiguration des Zabbix-Frontends korrekt festgelegt sein. Zabbix sender sendet bis zu 250 Werte über eine Verbindung. Der Inhalt der Eingabedatei muss UTF-8-codiert sein. Alle Werte aus der Eingabedatei werden sequenziell von oben nach unten gesendet. Die Einträge müssen nach folgenden Regeln formatiert sein:

- Anführungszeichen und Einträge ohne Anführungszeichen werden unterstützt.
- Das doppelte Anführungszeichen ist das Anführungszeichen.
- Einträge mit Leerzeichen müssen in Anführungszeichen gesetzt werden.
- Doppelte Anführungszeichen und umgekehrte Schrägstriche innerhalb eines in Anführungszeichen gesetzten Eintrags müssen mit einem umgekehrten Schrägstrich maskiert werden.
- Maskierung wird in Einträgen ohne Anführungszeichen nicht unterstützt.
- Escape-Sequenzen für Zeilenumbrüche (\n) werden in Zeichenfolgen in Anführungszeichen unterstützt.
- Escape-Sequenzen für Zeilenumbrüche werden am Ende eines Eintrags entfernt.

-T, --with-timestamps
Diese Option kann nur zusammen mit der Option --input-file verwendet werden.

Jede Zeile der Eingabedatei muss vier durch Leerzeichen getrennte Einträge enthalten: <hostname> <key> <timestamp> <value>. Der Zeitstempel muss im Unix-Zeitstempelformat angegeben werden. Wenn der Zieldatenpunkt Auslöser enthält, die auf ihn verweisen, müssen alle Zeitstempel in aufsteigender Reihenfolge angegeben werden. Andernfalls wird die Ereignisberechnung nicht korrekt ausgeführt.

Beispiel für eine Zeile einer Eingabedatei:

```
 "Linux DB3" db.connections 1429533600 43
```

Weitere Informationen finden Sie unter der Option --input-file.

Wenn ein Wert mit Zeitstempel für einen Host gesendet wird, der sich in einer Wartungsperiode vom Typ „no data“ befindet, wird dieser Wert verworfen. Es ist jedoch möglich, einen Wert mit Zeitstempel für eine abgelaufene Wartungsperiode zu senden; dieser

wird akzeptiert.

-N, --with-ns
Diese Option kann nur zusammen mit der Option --with-timestamps verwendet werden.

Jede Zeile der Eingabedatei muss fünf durch Leerzeichen getrennte Einträge enthalten: <hostname> <key> <timestamp> <ns> <value>.

Beispiel für eine Zeile einer Eingabedatei:

"Linux DB3" db.connections 1429533600 7402561 43

Weitere Informationen finden Sie unter der Option --input-file.

-r, --real-time
Sendet Werte einzeln, sobald sie empfangen werden. Dies kann beim Lesen aus der Standardeingabe verwendet werden.

--tls-connect value
Gibt an, wie eine Verbindung zum Server oder Proxy hergestellt wird. Werte:

unencrypted
Verbindung ohne Verschlüsselung (Standardwert)

psk
Verbindung über TLS mit einem vorab gemeinsam vereinbarten Schlüssel

cert
Verbindung über TLS mit einem Zertifikat

--tls-ca-file CA-file
Vollständiger Pfadname einer Datei, die die Zertifikate der übergeordneten Zertifizierungsstelle(n) zur Überprüfung des Peer-Zertifikats enthält.

--tls-crl-file CRL-file
Vollständiger Pfadname einer Datei, die widerrufene Zertifikate enthält.

--tls-server-cert-issuer cert-issuer
Zulässiger Aussteller des Serverzertifikats.

--tls-server-cert-subject cert-subject
Zulässiger Betreff des Serverzertifikats.

--tls-cert-file cert-file
Vollständiger Pfadname einer Datei, die das Zertifikat oder die Zertifikatskette enthält.

--tls-key-file key-file
Vollständiger Pfadname einer Datei, die den privaten Schlüssel enthält.

--tls-psk-identity PSK-identity
PSK-Identitätszeichenfolge.

--tls-psk-file PSK-file
Vollständiger Pfadname einer Datei, die den vorab gemeinsam vereinbarten Schlüssel enthält.

--tls-cipher13 cipher-string
Verschlüsselungszeichenfolge für OpenSSL 1.1.1 oder höher für

TLS 1.3. Überschreibt die Kriterien für die Auswahl der standardmäßigen Verschlüsselungssuite. Diese Option ist nicht verfügbar, wenn die OpenSSL-Version älter als 1.1.1 ist.

--tls-cipher cipher-string
GnuTLS-Prioritätszeichenfolge (für TLS 1.2 und höher) oder OpenSSL-Verschlüsselungszeichenfolge (nur für TLS 1.2). Überschreibt die Kriterien für die Auswahl der standardmäßigen Verschlüsselungssuite.

-v, --verbose
Ausführlicher Modus; -vv liefert weitere Details.

-h, --help
Zeigt diese Hilfe an und beendet das Programm.

-V, --version
Gibt Versionsinformationen aus und beendet das Programm.

EXIT STATUS

Der Exit-Status ist 0, wenn die Werte gesendet und alle erfolgreich vom Server verarbeitet wurden. Wenn Daten gesendet wurden, die Verarbeitung mindestens eines Werts jedoch fehlgeschlagen ist, lautet der Exit-Status 2. Wenn das Senden der Daten fehlgeschlagen ist, lautet der Exit-Status 1.

EXAMPLES

zabbix_sender -c /etc/zabbix/zabbix_agentd.conf -k mysql.queries -o 342.45
Sendet 342.45 als Wert für den Datenpunkt mysql.queries des überwachten Hosts. Verwendet den überwachten Host und den Zabbix-Server, die in der Agent-Konfigurationsdatei definiert sind.

zabbix_sender -c /etc/zabbix/zabbix_agentd.conf -s "Monitored Host" -k mysql.queries -o 342.45
Sendet 342.45 als Wert für den Datenpunkt mysql.queries des Hosts „Monitored Host“ unter Verwendung des in der Agent-Konfigurationsdatei definierten Zabbix-Servers.

zabbix_sender -z 192.168.1.113 -i data_values.txt
Sendet Werte aus der Datei data_values.txt an den Zabbix-Server mit der IP-Adresse 192.168.1.113. Hostnamen und Schlüssel sind in der Datei definiert.

echo "- hw.serial.number 1287872261 SQ4321ASDF" | zabbix_sender -c /usr/local/etc/zabbix_agentd.conf -T -i -
Sendet einen Wert mit Zeitstempel über die Befehlszeile an den in der Agent-Konfigurationsdatei angegebenen Zabbix-Server. Der Bindestrich in den Eingabedaten gibt an, dass der Hostname ebenfalls aus derselben Konfigurationsdatei verwendet werden soll.

echo '"Zabbix server" trapper.item ""' | zabbix_sender -z 192.168.1.113 -p 10000 -i -
Sendet über die Befehlszeile einen leeren Wert eines Datenpunkts an den Zabbix-Server mit der IP-Adresse 192.168.1.113 am Port 10000. Leere Werte müssen durch leere doppelte Anführungszeichen angegeben werden.

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
Sendet 342.45 als Wert für den Datenpunkt mysql.queries des Hosts

„Monitored Host“ an den Server mit der IP-Adresse 192.168.1.113
unter Verwendung von TLS mit einem Zertifikat.

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
Sendet 342.45 als Wert für den Datenpunkt mysql.queries des Hosts
„Monitored Host“ an den Server mit der IP-Adresse 192.168.1.113
unter Verwendung von TLS mit einem vorab gemeinsam vereinbarten
Schlüssel (PSK).

SEE ALSO
Dokumentation

zabbix_agentd(8), zabbix_get(1), zabbix_proxy(8), zabbix_server(8),
zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix 2022-01-06 ZABBIX_SENDER(1)

zabbix_server

ZABBIX_SERVER(8) System Manager's Manual ZABBIX_SERVER(8)

NAME
zabbix_server - Zabbix Server-Daemon

SYNOPSIS
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 ist der zentrale Daemon der Zabbix-Software.

OPTIONS
-c, --config config-file
Verwendet die alternative config-file anstelle der Standarddatei.

-f, --foreground
Führt den Zabbix Server im Vordergrund aus.

-R, --runtime-control runtime-option
Führt administrative Funktionen gemäß runtime-option aus.

Runtime control options
config_cache_reload
Lädt den Konfigurationscache neu. Wird ignoriert, wenn der Cache derzeit geladen wird.

snmp_cache_reload
Lädt den SNMP-Cache neu.

housekeeper_execute
Führt den Housekeeper aus. Wird ignoriert, wenn der Housekeeper derzeit ausgeführt wird

trigger_housekeeper_execute
Führt den trigger housekeeper aus (entfernt Probleme für gelöschte Auslöser). Wird ignc

secrets_reload
Lädt Geheimnisse aus Vault neu. Wird ignoriert, wenn der Konfigurationscache oder Gehei

```
    diaginfo[=section]
        Protokolliert interne Diagnoseinformationen des angegebenen Abschnitts. Section kann hi

    prof_enable[=target]
        Aktiviert Profiling, betrifft alle Prozesse, wenn target nicht angegeben ist

    prof_disable[=target]
        Deaktiviert Profiling, betrifft alle Prozesse, wenn target nicht angegeben ist

    ha_status
        Zeigt den Status des Hochverfügbarkeitsclusters an. Kann nur auf dem aktiven Knoten aus

    ha_remove_node
        ha_remove_node[=name|ID] Entfernt einen Hochverfügbarkeitsknoten anhand seines in ha_st

    ha_set_failover_delay[=delay]
        Legt die Failover-Verzögerung für Hochverfügbarkeit fest. Die Verzögerung kann mit Zeit

    log_level_increase[=target]
        Erhöht den Protokollierungsgrad, betrifft alle Prozesse, wenn target nicht angegeben is

    log_level_decrease[=target]
        Verringert den Protokollierungsgrad, betrifft alle Prozesse, wenn target nicht angegebe

Log level control targets
    process-type
        Alle Prozesse des angegebenen Typs (alerter, alert manager,
        alert syncer, availability manager, browser poller, config-
        uration syncer, configuration syncer worker, connector man-
        ager, connector worker, discovery manager, escalator, ha m-
        anager, history poller, history syncer, housekeeper, h-
        ttp poller, http agent poller, icmp pinger, internal poll-
        er, ipmi manager, ipmi poller, java poller, lld manager,
        lld worker, odbc poller, poller, agent poller, preprocessi-
        ng manager, preprocessing worker, proxy poller, proxy group
        manager, report manager, report writer, self-monitoring,
        service manager, snmp poller, snmp trapper, supervisor,
        task manager, timer, trapper, trigger housekeeper, unreac-
        hable poller, vmware collector)

    process-type,N
        Prozess-Typ und Nummer (z. B. poller,3)

    pid Prozesskennung

Profiling control targets
    process-type
        Alle Prozesse des angegebenen Typs (alerter, alert manager,
        alert syncer, availability manager, browser poller, config-
        uration syncer, configuration syncer worker, connector man-
        ager, connector worker, discovery manager, escalator,
        ha manager, history poller, history syncer, housekeeper,
        http poller, http agent poller, icmp pinger, internal po-
        ller, ipmi manager, ipmi poller, java poller, lld manager,
        lld worker, odbc poller, poller, agent poller, preproc-
        essing manager, proxy poller, proxy group manager, repo-
        rt manager, report writer, self-monitoring, service mana-
        ger, snmp poller, snmp trapper, task manager, timer, trap-
        per, trigger housekeeper, unreachable poller, vmware colle-
        ctor)

    process-type,N
```

```
                Prozess-Typ und Nummer (z. B. history syncer,1)

            pid Prozesskennung

            scope
                (rwlock, mutex, processing) kann mit process-type verwendet
                werden (z. B. history syncer,1,processing)

    -T, --test-config
           Validiert die Konfigurationsdatei und beendet sich.

    -h, --help
           Zeigt diese Hilfe an und beendet sich.

    -V, --version
           Gibt Versionsinformationen aus und beendet sich.

FILES
    /usr/local/etc/zabbix_server.conf
           Standardpfad der Konfigurationsdatei des Zabbix Server (sofern
           dies bei der Kompilierung nicht geändert wurde).

SEE ALSO
    Dokumentation

    zabbix_agentd(8),  zabbix_get(1),  zabbix_proxy(8),   zabbix_sender(1),
    zabbix_js(1), zabbix_agent2(8), zabbix_web_service(8)

Zabbix                          2022-01-06                 ZABBIX_SERVER(8)
```

zabbix_web_service

```
ZABBIX_WEB_SERVICE(8)       Systemverwaltungs-Handbuch      ZABBIX_WEB_SERVICE(8)

NAME
    zabbix_web_service - Zabbix-Web-Service

SYNOPSIS
    zabbix_web_service [-c config-file]
    zabbix_web_service [-c config-file] -T
    zabbix_web_service -h
    zabbix_web_service -V

DESCRIPTION
    zabbix_web_service ist eine Anwendung zur Bereitstellung von Web-Services für Zab-
    bix-Komponenten.

OPTIONS
    -c, --config config-file
           Verwenden Sie die alternative config-file anstelle der Standarddatei.

    -T, --test-config
           Konfigurationsdatei prüfen und beenden.

    -h, --help
           Diese Hilfe anzeigen und beenden.

    -V, --version
           Versionsinformationen ausgeben und beenden.

FILES
```

```
       /usr/local/etc/zabbix_web_service.conf
              Standardpfad der Konfigurationsdatei des Zabbix-Web-Services (falls
              nicht während der Kompilierung geändert).

SEE ALSO
       Dokumentation

       zabbix_agentd(8),  zabbix_get(1),  zabbix_proxy(8),   zabbix_sender(1),
       zabbix_server(8), zabbix_js(1), zabbix_agent2(8)

Zabbix                            2022-01-06            ZABBIX_WEB_SERVICE(8)
```

Urheberrechtshinweis

Die Zabbix-Dokumentation wird NICHT unter der AGPL-3.0-Lizenz verbreitet. Die Nutzung der Zabbix-Dokumentation unterliegt den folgenden Bedingungen:

Sie dürfen eine gedruckte Kopie dieser Dokumentation ausschließlich für Ihren eigenen persönlichen Gebrauch erstellen. Die Konvertierung in andere Formate ist zulässig, solange der tatsächliche Inhalt in keiner Weise verändert oder bearbeitet wird. Sie dürfen diese Dokumentation in keiner Form und auf keinem Medium veröffentlichen oder verbreiten, es sei denn, Sie verbreiten die Dokumentation in einer Weise, die der Verbreitung durch Zabbix ähnelt (das heißt elektronisch zum Download auf einer Zabbix-Website) oder auf einem USB- oder ähnlichen Medium, vorausgesetzt jedoch, dass die Dokumentation zusammen mit der Software auf demselben Medium verbreitet wird. Jede andere Nutzung, wie etwa jede Verbreitung gedruckter Kopien oder die Nutzung dieser Dokumentation ganz oder teilweise in einer anderen Veröffentlichung, erfordert die vorherige schriftliche Zustimmung eines bevollmächtigten Vertreters von Zabbix. Zabbix behält sich alle Rechte an dieser Dokumentation vor, die oben nicht ausdrücklich eingeräumt wurden.